CePETel
Sindicato de los Profesionales
de las Telecomunicaciones

SECRETARÍA TÉCNICA

IPEI
Instituto Profesional de
Estudios e Investigación

Formación Profesional en CePETel 2023

Desde la Secretaria Técnica del Sindicato CePETel convocamos a participar en el siguiente curso de formación profesional:

Redes Definidas por Software (SDN)

Clases: 6 de 3hs c/u de 18:00 a 21:00 hs.
Días que se cursa: los días lunes, 30 octubre; 6, 13 y 27 de noviembre; 4 y 11 de diciembre.
Modalidad: a distancia (requiere conectarse a la plataforma Zoom en los días y horarios indicados precedentemente).
Docente: José Luis Pellegrino

La capacitación es:

- **Sin cargo para afiliados y su grupo familiar directo.**
- **Sin cargo para encuadrados con convenio CePETel.**
- **Con cargo al universo no contemplado en los anteriores.**

Informes: enviar correo a tecnico@cepetel.org.ar

Inscripción (hasta el 27 de octubre): ingresar al formulario (se recomienda realizar el registro por medio de una cuenta de correo personal y ***no utilizar dispositivos de la empresa para acceder al link***).

https://forms.gle/YScyrL1kJ313xhvJ7

Temario:

Introducción General: *Presentación del paradigma, discusión de las tecnologías relacionadas, resumen histórico*

Networking capa 2: *Repaso de networking necesario para desarrollar temas como VXLAN (Virtual eXtensible LAN)*

Networking capa 3: *Repaso de networking necesario para relacionar el uso de BGP (Border Gateway Protocol) en SDN*

Virtualización: *Analisis del impacto que introduce NFV en el mundo SDN*

Ing. Daniel Herrero – Secretario Técnico – CDC

Datacenters: *Características técnicas, concepto de RAID (Redundant Array of Intependent Disks), conectividad, Leaf&Spine*

NFV (Network Function Virtualization): *Relación entre NFV y SDN, como proveer conectividad a las funciones virtuales*

SRIOV (Single Root-I/O Virtualization) & OVS (Open Virtual Switch): *Mecanismos aceleradores que permiten mejorar la performance en términos de latencia*

Sistemas de Storage: *Descripción de un sistema típico de almacenamiento*

Nubes públicas: *Ejemplos de nubes comerciales, diferencias y similitudes con Telco Cloud*

Servidores: *Ejemplos de servidores DELL*

Introducción a Open Stack y SDN: *Relación entre OPEN STACK y SDN*

NETCONF (Network Configuration Protocol): *Descripción del NETCONF*

SDN en DC (Data Center): *Casos de aplicación de SDN*

SDN en SP (Service Provider): *Casos de aplicación de SDN*

SDN en WAN: *Casos de aplicación de SDN*

SDN en EMPRESAS: *Casos de aplicación de SDN*

Enfoques de despliegues de SDN: *Arquitecturas de despliegue centralizado, distribuído, híbrido: el rol de los vendors tradicionales de equipos de Networking*

Acerca del docente

José Luis Pellegrino es Ingeniero en Electrónica Universidad Nacional de La Plata (especialista en Telecomunicaciones), contando con más de 20 años de experiencia laboral. Es experto en redes fijas y móviles,y posee un amplio conocimiento sobre diferentes tecnologías tales como Conmutación C.S Y P.S, NGN, redes y protocolos de señalización S7 (ISUP/MAP/INAP) y señaización IP:H.248, SIP, Diameter, así como también en redes y arquitecturas IMS, C.S, SBC, P.S, LTE, CSFB, mVoLTE, fVoLTE, WiFi, WiFiCalling, WRTC. En el Sindicato Cepetel dictó el curso CORE IMS en el año 2020, mientras que en el 2021 hizo lo propio con Redes 5 G Nivel Inicial y en este 2022 dictó también Redes 5 G Nivel Avanzado al igual que el curso de Redes 6 G.

INTRODUCTION TO SDN

And here is the question:

What is SDN?

Can you define SDN in a few words?

DEBUNKING THE MYTHS OF SOFTWARE DEFINED NETWORKING

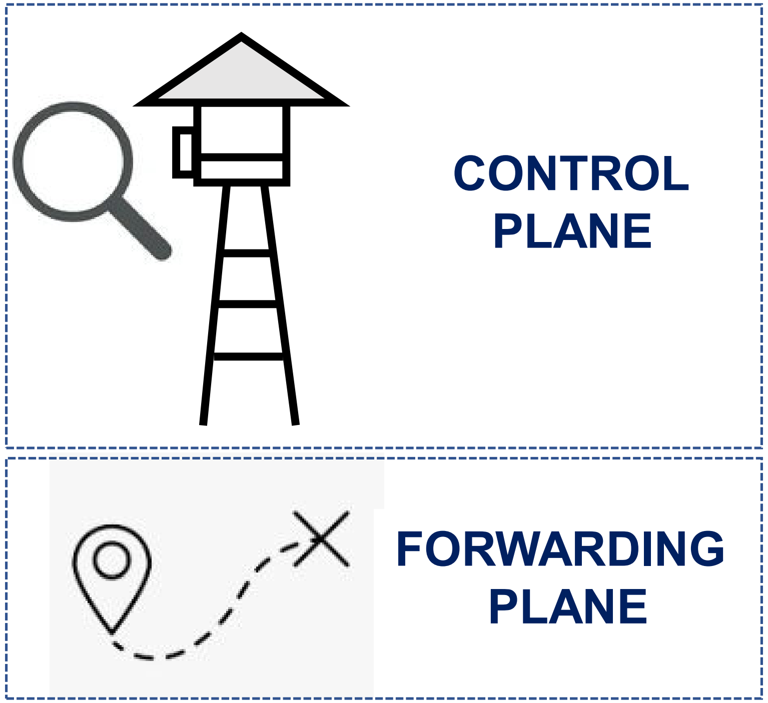
SDN as a “new” technology is surrounded by different type of myths. Discuss about these myths will help us to understand the evolution and the adopting curve/use cases of this technology

SDN MYTHS

During the Industrial revolution, some myths appeared.
SDN, is perhaps one of the “revolutions” of networking world, and same is happening

SDN MYTHS, FIRST WAVE, EARLY ADOPTERS

Software-defined networking (SDN) first came onto scene about 10 years ago.
Current Networks had been unchanged for many years, and now is time to introduce changes to get higher level of **automation and reduce costs.**



CONTROL PLANE

FORWARDING PLANE

Basic idea was to separate “Control plane” from “forwarding plane” in order to centralize the point where routing decisions take place.

During the fisrt years, not all promises had been realized, or were even realistic.

Such is commonly the case with new technologies: the initial hype exceeds the reality of the situation, but usually there are reasons to hold firm to the trend and take advantage of what the technology does deliver successfully.

First Driver: Reduce costs

Source: radware.com

SDN MYTHS

MYTH: SDN IS REALLY JUST ABOUT REDUCING CAPITAL EXPENDITURES

Reality: This same myth was associated with server virtualization a decade ago. While virtualization certainly reduced capitalization expenditures for servers by increasing the utilization of hardware resources, the greater benefits of automation, redundancy and granular manageability were obtained after implementation.

The same is true of SDN. While there is a significant potential cost savings for organizations in capitalizing their network infrastructures, the real cost savings lies in the transition to an automated data plane device management system as so much networking functionality is managed manually today.

Although all organizations would realize a resulting reduction in operating expenses, the reduction in operating services for large organizations could outpace any reduction in capital expenditures. While the compounding cost savings is what initially attracts managers to the concept of SDN, the degree of automation, agility and centralized control that will be enjoyed by both users and IT administrators alike may be justification enough for its implementation.

OPEX

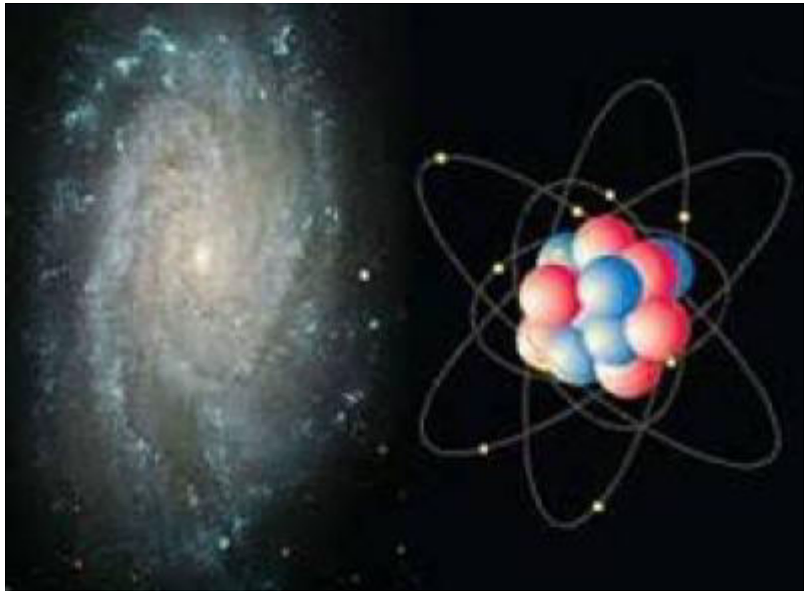
CAPEX

OPEX Saving could be more important than CAPEX saving
But….. Does SDN mean CAPEX saving?

Source: WEI

SDN MYTHS

MYTH: SDN IS ONLY MEANINGFUL FOR LARGE DATA DRIVEN DATACENTERS



Reality: While it’s true that it is primarily large enterprise organizations such as Facebook, Yahoo and Ebay that have been the primary benefactors of SDN, large datacenters are usually the early adopters of new technologies as was the case with virtualization. SDN simplifies the entire administrative process from configuration to management and monitoring for networks of all sizes. This reduces the burden levied upon IT departments which can be especially beneficial to smaller organizations whose IT staffs are overburdened due to staffing limitations

In Telco industry, some improvements and architecture changes as Network disaggregation are new drivers for SDN adoption.

As it will be seen, an interesting question is how to handle network inside the Datacenter and also outside the Datacenter

Source: WEI

SDN MYTHS

MYTH: SDN REQUIRES MY IT STAFF TO HAVE PROGRAMMING SKILLS

Reality: According to Dominic Wilde, vice president of global product line management at HP Networking, “The idea that all of the sudden you have to become programmers overnight is false.” (*)

Large enterprise based organizations such as Google have the luxury of in-house programmers that can customize code for SDN (this is primarily on the northbound interface which connects the control plane with the higher level administrative application or orchestrator). This luxury isn’t restricted to SDN, but is allocated for IT tasks across the board out of necessity as these organizations usually face unique challenges due to their size and intense traffic demands. The fact is that there are a number of turnkey, vendor-supported solutions available today that can be implemented without having to write a single line of code. These platforms commonly provide an intuitive web interface for users to utilize SDN capabilities.

In spite what is has been said, would Telco industry benefit of having in-house programmers? Think about RIC, just as an example

(*) Sources SDN: Programming Skills Needed - Or Not?, NetworkComputing.com, Marcia Savage, 4/02/2014

Source: WEI

SDN MYTHS

MYTH: SDN WILL BE AN IT JOB KILLER

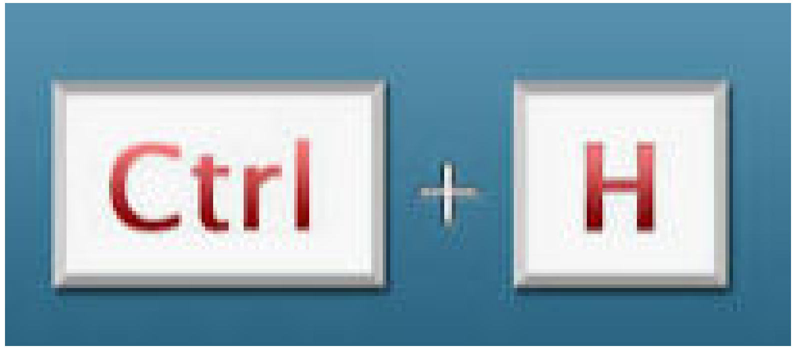
Reality: Automation has been a perceived threat to all industries ever since the Industrial Revolution. The IT industry has automated so many configuration tasks that were performed manually fifteen years ago. Today we can image fleets of computing devices simultaneously and enterprise level antivirus applications run without almost no human intervention at all. It is true that SDN-enabled environments will require less hands-on effort to keep their myriad of network devices up and running, however, the need for network administrators will not diminish. If history is an indicator, it is most probable that new job opportunities will be created as IT managers will actually have time to focus on strategic value added projects rather than constant day-to-day hands-on maintenance tasks.

Source: WEI

SDN MYTHS

MYTH: AN SDN IMPLEMENTATION WILL REQUIRE REPLACING MY NETWORK ALL AT ONCE

Reality: Yes, in order to experience the complete benefit of SDN for your entire network, including branch offices, all of your network hardware must be SDN compliant. But when has anyone done a complete upgrade at one time? No organization virtualized their entire server fleet at one time. Instead, they chose a transition strategy of either virtualizing vital servers that would garner the biggest impact or they chose lesser important servers to ease their way into the virtual process. Organizations will transition to SDN in the same manner. This can be accomplished by simply choosing SDN devices for your networking components as part of your existing hardware refresh plan or deploying SDN whenever new equipment is added for new projects or infrastructure growth. SDN devices can co-exist with traditional devices during the transition process, giving your staff the needed time to grow accustomed to this new technology and how to maximize its value.



Not a complete upgrade at one time!!
Will be a complete upgrade sometime at all?
How long can SDN devices co-exist with traditional devices?
Which is the traditional devices suppliers rol?

Source: WEI

SDN MYTHS

MYTH: SDN IS ONLY A THEORETICAL CONCEPT AND ISN'T REALITY

Reality: SDN is reality and is in production within some of the biggest enterprise organizations in the world. Some of the largest network hardware and software manufacturers in the world are releasing and shipping SDN ready devices, controllers and application software every day and collectively have tens of millions of SDN ports in production. Three of the leading SDN solutions are VMware NSX, Aruba/HPE OpenFlow and Cisco ACI, and enterprice are implementing them with clients today. SDN is already redefining the network as we know it and all organizations that aren't a part of this revolution should begin planning their SDN strategy sooner than later because their competitors most certainly are.

SUMMARY There are many myths and misconceptions about SDN as is the case with any new technology. SDN is more than just software, virtualization or network trafficking. SDN is delivering dynamic networks to organizations across the globe today, networks that can adjust rapidly to change in real time. It is this dynamic nature and ability that also delivers a competitive advantage to any organization today

Source: WEI

SDN MYTHS

MYTH: SDN IS ONLY MEANINGFUL FOR LARGE DATA DRIVEN DATACENTERS

→

It is only at the beginning. Early adopters are Large Data, but other users like Telcos are adopting it

MYTH: SDN REQUIRES MY IT STAFF TO HAVE PROGRAMMING SKILLS

→

The fact is that there are a number of turnkey, vendor-supported solutions available today that can be implemented without having to write a single line of code.

MYTH: SDN WILL BE AN IT JOB KILLER

→

If history is an indicator, it is most probable that new job opportunities will be created as IT managers will actually have time to focus on strategic value added projects rather than constant day-to-day hands-on maintenance tasks.

MYTH: AN SDN IMPLEMENTATION WILL REQUIRE REPLACING MY NETWORK ALL AT ONCE

→

Not a complete upgrade at one time

MYTH: SDN IS ONLY A THEORETICAL CONCEPT AND ISN'T REALITY

→

SDN is reality and is in production within some of the biggest enterprise organizations in the world

Source: WEI

SDN MYTHS, FIRST WAVE, EARLY ADOPTERS

Use Case #1: Save big money by commoditizing data center switching.

During the early days SDN adoption was promoted mainly by hyperscalers like Google. Their situation was clear, and they needed to reduce CAPEX in their networks.

By 2015, while some were enjoying real advantages, SDN was still a myth for others. The early hype was that ***OpenFlow-based*** "underlay" SDNs would turn the networking equipment world on its head by allowing all of the controls to be centralized into software controllers so that the actual switches themselves could be "**dumb packet forwarders**." This has indeed been a huge boon to web-scale operations, such as Google's poster-child implementation, where the traffic is fairly homogenous and there is an army of programmers available to build it and keep it running.

Service providers would have been able to enjoy similar success since they typically had the incentive and the resources to leverage such technology advances.

But it was a very knew and inmature technology, which has not inspired the majority of mainstream enterprises, which preferred stability and sophistication in their networking gear. They didn't have the resources or the risk tolerance to adopt a major architectural shift of this nature.

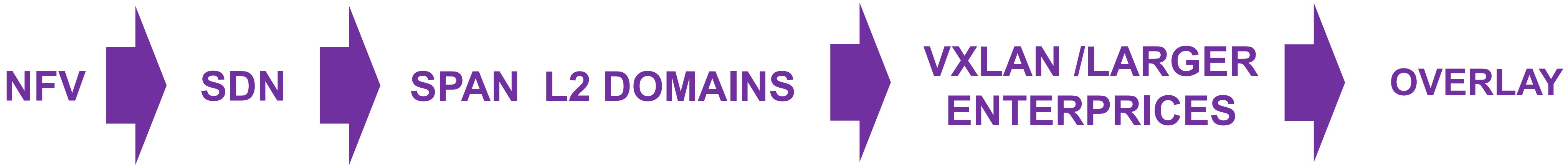
- ***Early adopters: hyperscalers, Underlay***
- ***OpenFlow was the first protocol developed for SDN***

Source: radware.com

SDN MYTHS, FIRST WAVE, EARLY ADOPTERS

Use Case #2: Stretch virtual networks (beyond the Datacenter limits).

Virtualization technologies were developed many years before that SDN was announced. It seemed a good idea to link both universes.
The “overlay” type of SDN appears to be the alternative reality of choice for mainstream enterprise. While tunneling and encapsulation predate the whole SDN movement by many, many years, a recent new crop of sophisticated choices for virtual network switching that can **span Layer 2 domains and eliminate VLAN scale limits** opens the door to the full promise of virtual server mobility. Further, these solutions can be implemented with little or no direct involvement by the networking team. The only myth remaining here is that although it was true, SDN overlays made the most sense for larger enterprises seeking to consolidate and leverage investments made in server virtualization and hybrid cloud infrastructures.



NFV → SDN → SPAN L2 DOMAINS → VXLAN /LARGER ENTERPRICES → OVERLAY

Source: radware.com

SDN MYTHS, FIRST WAVE, EARLY ADOPTERS

Use Case #3: Automate network functions and services.

This is the realm of network functions virtualization (NFV) and is made possible by the new levels of programmability and standardization that have come with SDN push. Service providers have latched onto this with a vengeance as they recognize an opportunity to expand service offerings while also stripping out both capital and operational costs. Enterprises understand the value too, but they **look at NFV as a means to apply the same types of optimizations** and controls with mixed hybrid infrastructures they have become reliant on in traditional, on-premise environments. While enterprises are lagging behind service providers in NFV adoption, this is clearly an eventual reality in both camps.

At those moment Telco industry had not massively adopted SDN. See later some new RAN architectures which push SDN adoption

ENTERPRISES WERE NOT YET ADOPTING → **IT WAS ONLY MATTER OF TIME**

Source: radware.com

SDN MYTHS, FIRST WAVE, EARLY ADOPTERS

For the most part, these three core SDN use cases have ended up proving to be more reality than hype. EMA research published in mid-2014 found that while very few enterprises (less than 20%) had deployed SDN of any type, the overwhelming majority (over 80%!) was at some stage of research, evaluation, test, or deployment. Such focus and attention would certainly drive more use and more validation of SDN technologies in the months and years to come.

What about Telco Industry?

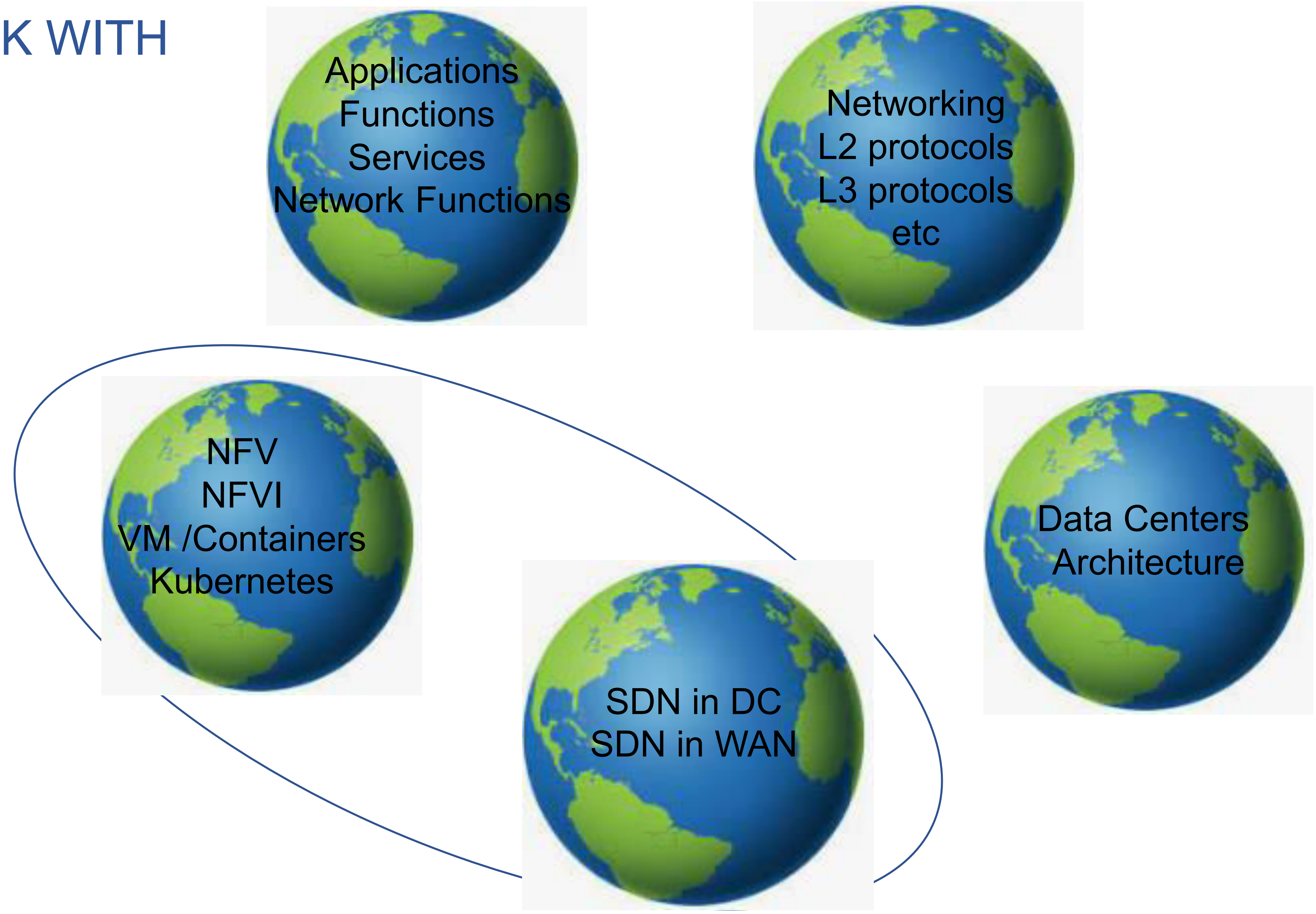
Source: radware.com

SDN USE EXAMPLES

DIFFERENT WORLD LINKED BY SDN

SDN need other technologies collaboration.
Understanding these technologies, will help us to
understand SDN

DIFFERENT WORLDS TO WORK WITH



Applications
Functions
Services
Network Functions

Networking
L2 protocols
L3 protocols
etc

NFV
NFVI
VM /Containers
Kubernetes

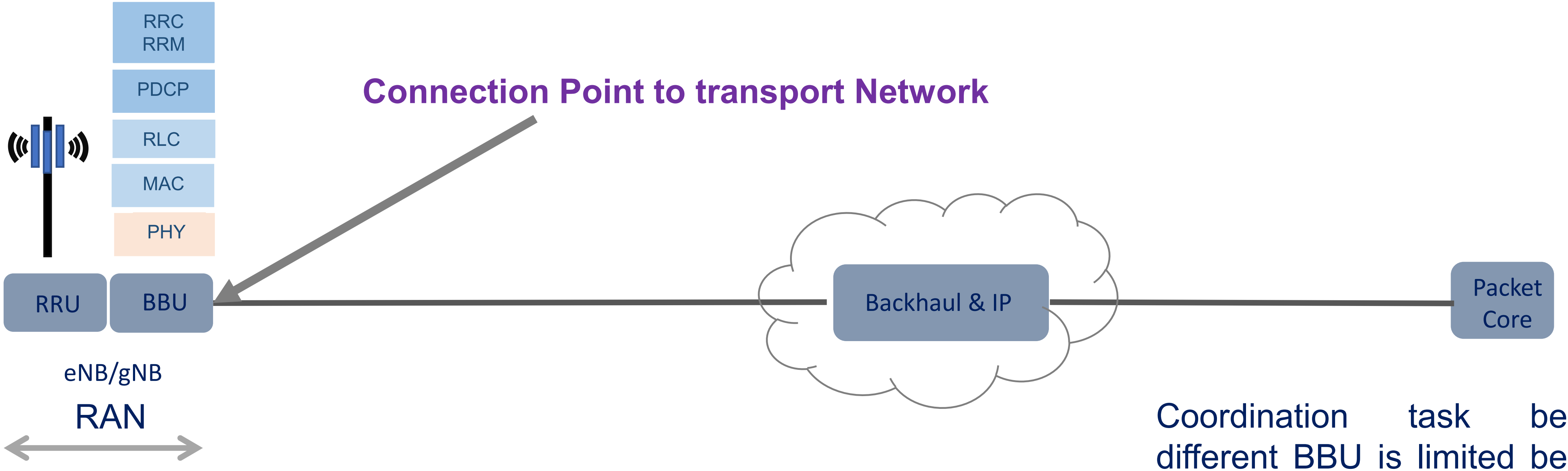
SDN in DC
SDN in WAN

Data Centers
Architecture

NETWORK EXAMPLES

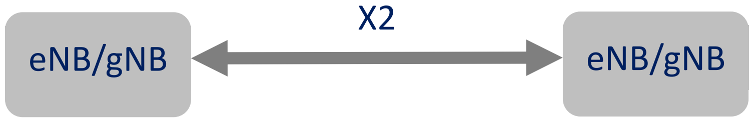
SDN IS NOT ONLY MATTER OF REDUCING COST.
CERTAIN DEPLOYMENTS ARE ALMOST IMPOSSIBLE
TO BE DONE WITHOUT SDN

NETWORKS EXAMPLES WHERE SDN AUTOMATION IS CRUCIAL
DISTRIBUTED RAN



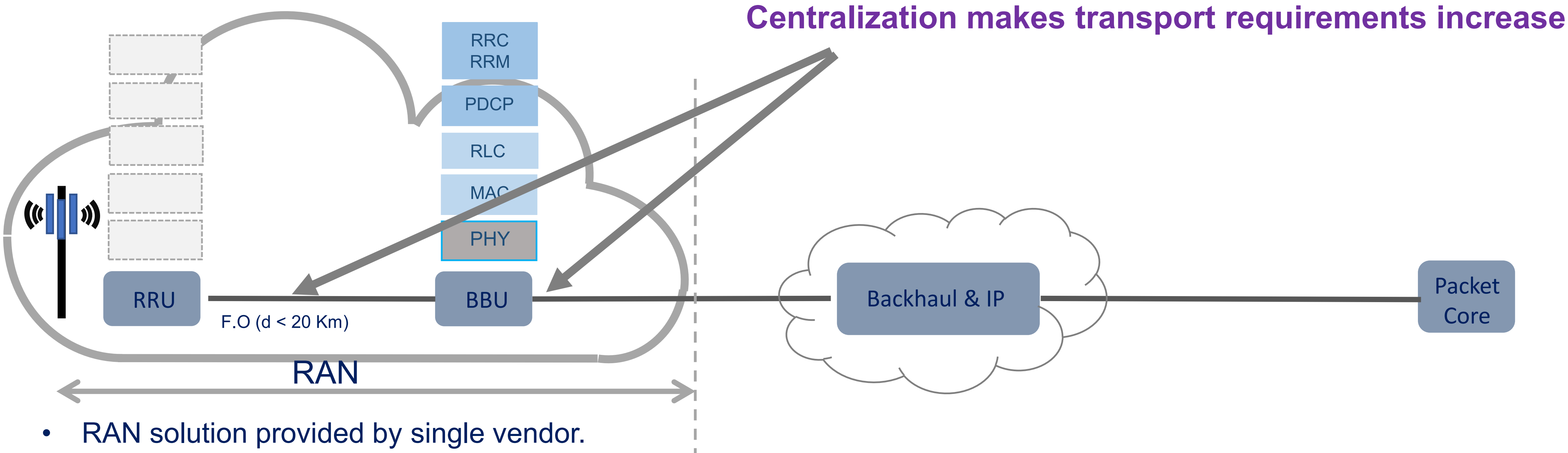
- Monolitic RAN provider by single vendor.

Coordination task between different BBU is limited because of distributed location of BBU (Interface X2 , LTE).



NETWORKS EXAMPLES WHERE SDN AUTOMATION IS CRUCIAL

CENTRALIZED RAN



- RAN solution provided by single vendor.
- Maximum desaggregation of RAN is RRU and BBU
- BBUs pool in same site
- RAN located in Edge and Packet Core centralized.
- Only Radio layer in site.
- All RRU located in area of < 20 kms distance (Fronthaul).
- Backhaul and synchronization complexiity is reduced
- Baremetal solution.
- Propietary CPRI

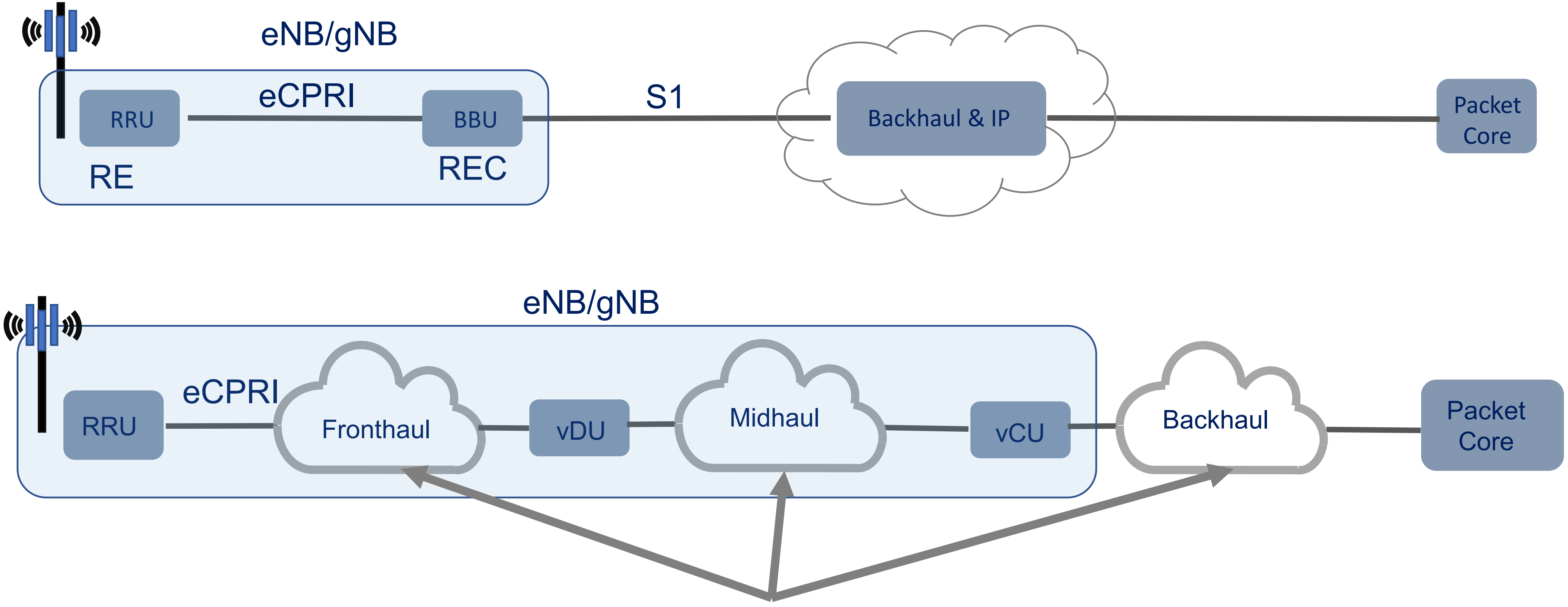
eNB = RRU +BBU

Still valid in 5G?

NETWORKS EXAMPLES WHERE SDN AUTOMATION IS CRUCIAL

DISAGGREGATION USE CASES

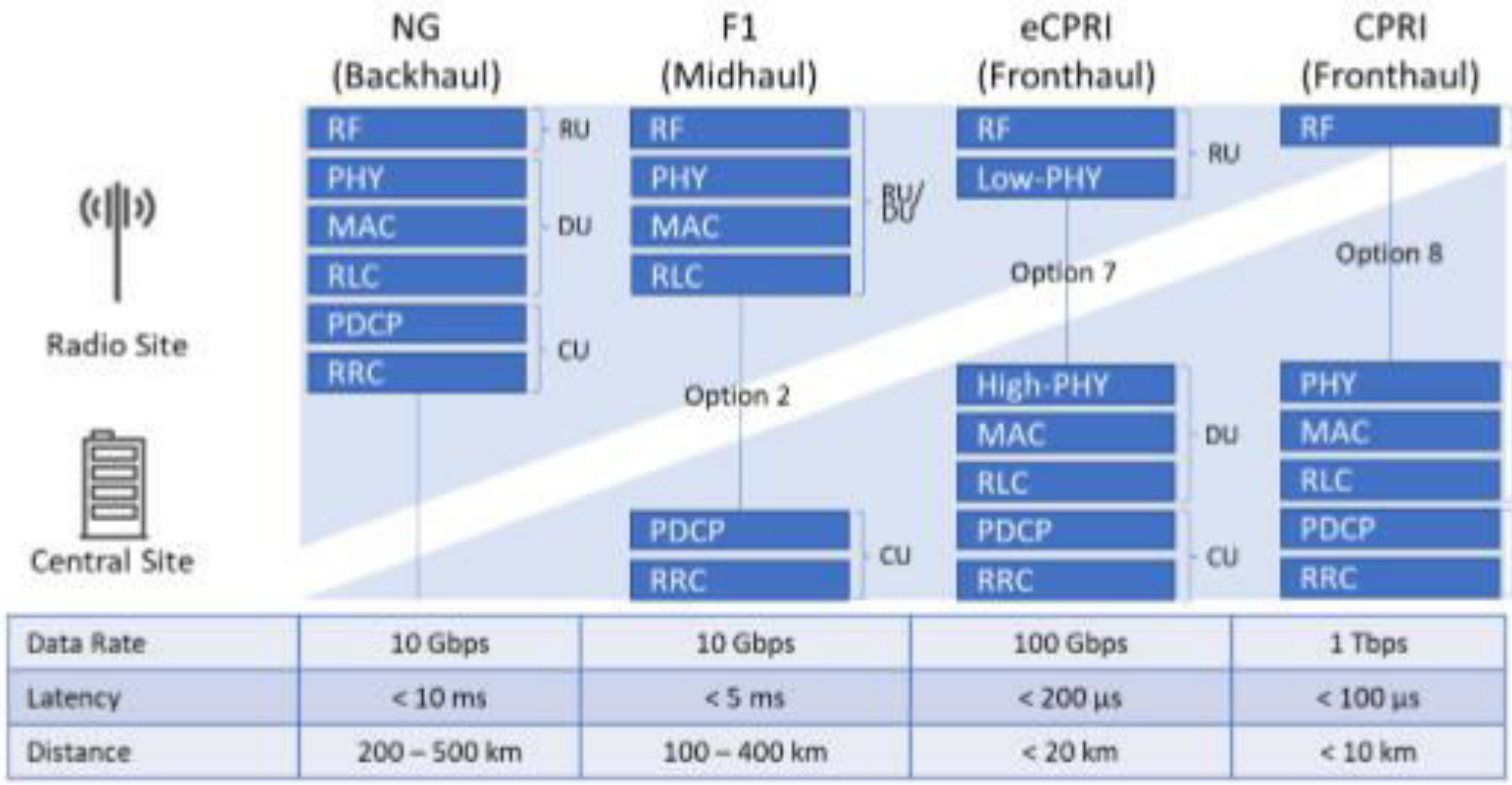
Different transport segments
Different locations for VNFs
Automation is crucial



Transport

NETWORKS EXAMPLES WHERE SDN AUTOMATION IS CRUCIAL

Latency requirements, data speed and distance limits for interfaces NG, F1, eCPRI y CPRI.
Example based in one Radio site with three sectors, 100 MHz bandwith channel, 64 T /64R, 256 QAM, 16 layers MIMO and multiuser MIMO (MU-MIMO).

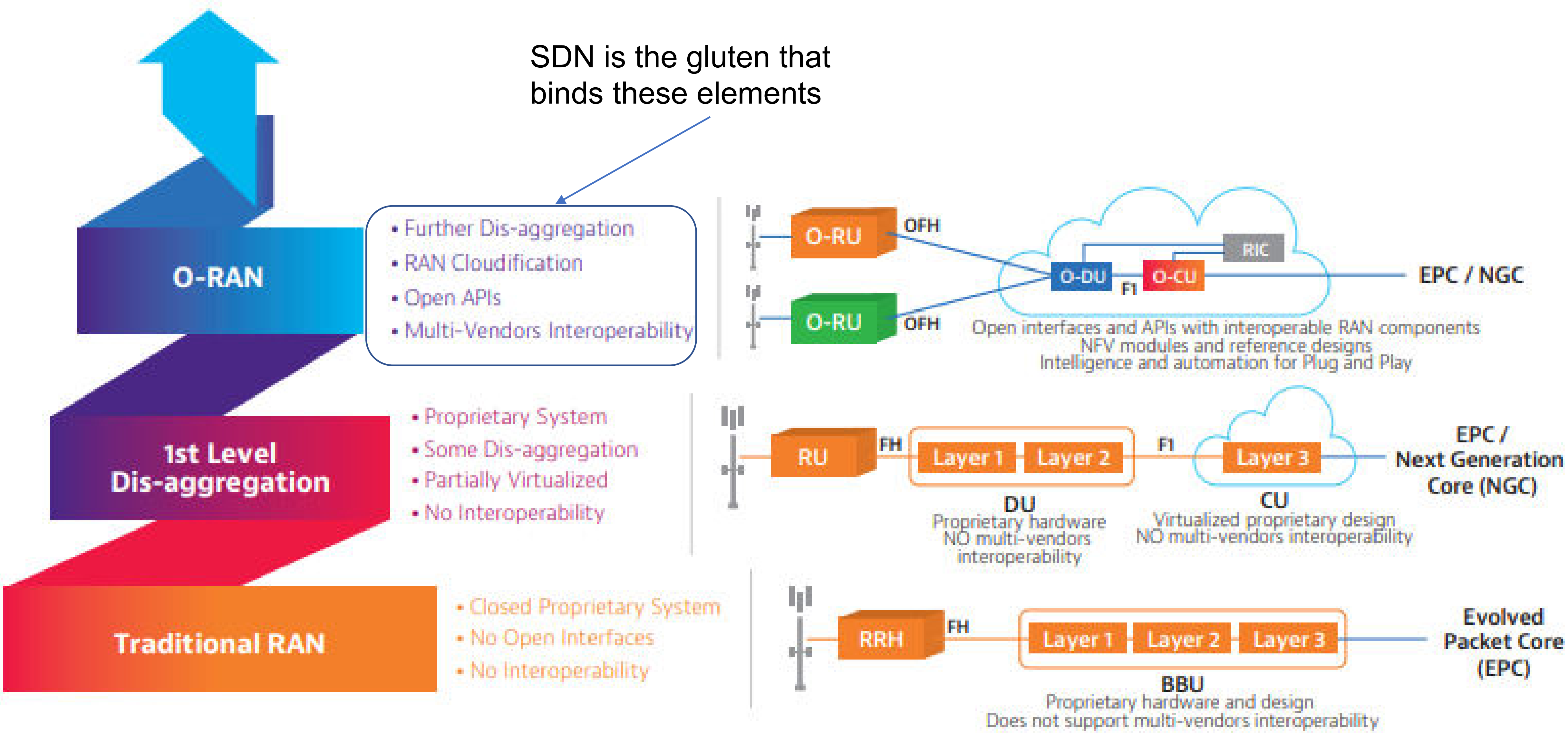


Data Rate	10 Gbps	10 Gbps	100 Gbps	1 Tbps
Latency	< 10 ms	< 5 ms	< 200 µs	< 100 µs
Distance	200 – 500 km	100 – 400 km	< 20 km	< 10 km

Disaggregation makes SDN essential

Source: NOKIA

NETWORKS EXAMPLES WHERE SDN AUTOMATION IS CRUCIAL



SDN is the gluten that binds these elements

O-RAN

- Further Dis-aggregation
- RAN Cloudification
- Open APIs
- Multi-Vendors Interoperability

O-RU OFH
O-RU OFH
RIC
O-DU F1 O-CU
EPC / NGC

Open interfaces and APIs with interoperable RAN components
NFV modules and reference designs
Intelligence and automation for Plug and Play

1st Level Dis-aggregation

- Proprietary System
- Some Dis-aggregation
- Partially Virtualized
- No Interoperability

RU FH Layer 1 Layer 2 F1 Layer 3
EPC / Next Generation Core (NGC)

DU
Proprietary hardware
NO multi-vendors interoperability

CU
Virtualized proprietary design
NO multi-vendors interoperability

Traditional RAN

- Closed Proprietary System
- No Open Interfaces
- No Interoperability

RRH FH Layer 1 Layer 2 Layer 3
Evolved Packet Core (EPC)

BBU
Proprietary hardware and design
Does not support multi-vendors interoperability

BACK TO THE GENESIS

SDN IS STILL NETWORKING

SDN is a new approach to the current world of networking, but it is still networking.

OSI model

The Open Systems Interconnection (OSI) model defines seven different layers of technology: physical, data link, network, transport, session, presentation, and application.

Switches

These devices operate at layer 2 of the OSI model and use logical local addressing to move frames across a network. Devices in this category include Ethernet in all its variations, VLANs, aggregates, and redundancies.

Routers

These devices operate at layer 3 of the OSI model and connect IP subnets to each other. Routers move packets across a network in a hop-by-hop fashion.

Ethernet

These broadcast domains connect multiple hosts together on a common infrastructure.
Hosts communicate with each other using layer 2 media access control (MAC) addresses.

IP addressing and subnetting

Hosts using IP to communicate with each other use 32-bit addresses. Humans often use a dotted decimal format to represent this address. This address notation includes a network portion and a host portion, which is normally displayed as 192.168.1.1/24.

SDN IS STILL NETWORKING

ICMP
Network engineers use this protocol to troubleshoot and operate a network, as it is the core protocol used (on some platforms) by the ping and traceroute programs. In addition, the Internet Control Message Protocol (ICMP) is used to signal error and other messages between hosts in an IP-based network.

Data center
A facility used to house computer systems and associated components, such as telecommunications and storage systems. It generally includes redundant or backup power supplies, redundant data communications connections, environmental controls (e.g., air conditioning and fire suppression), and security devices. Large data centers are industrial-scale operations that use as much electricity as a small town.

MPLS
Multiprotocol Label Switching (MPLS) is a mechanism in high-performance networks that directs data from one network node to the next based on short path labels rather than long network addresses, avoiding complex lookups in a routing table. The labels identify virtual links (paths) between distant nodes rather than endpoints. MPLS can encapsulate packets of various network protocols. MPLS supports a range of access technologies.

Northbound interface
An interface that conceptualizes the lower-level details (e.g., data or functions) used by, or in, the component. It is used to interface with higher-level layers using the southbound interface of the higher-level component(s). In architectural overview, the northbound interface is normally drawn at the top of the component it is defined in, hence the name northbound interface. Examples of a northbound interface are JSON or Thrift.

SDN IS STILL NETWORKING

Southbound interface
An interface that conceptualizes the opposite of a northbound interface. The southbound interface is normally drawn at the bottom of an architectural diagram.
Examples of southbound interfaces include I2RS, NETCONF, or a command-line interface.

Network topology
The arrangement of the various elements (links, nodes, interfaces, hosts, etc.) of a computer network. Essentially, it is the topological structure of a network and may be depicted physically or logically. Physical topology refers to the placement of the network's various components, including device location and cable installation, while logical topology shows how data flows within a network, regardless of its physical design. Distances between nodes, physical interconnections, transmission rates, and/or signal types may differ between two networks, yet their topologies may be identical.

Application programming interfaces
A specification of how some software components should interact with each other.
In practice, an API is usually a library that includes specification for variables, routines, object classes, and data structures. An API specification can take many forms, including an international standard (e.g., POSIX), vendor documentation (e.g., the JunOS SDK), or the libraries of a programming language.

SDN What is knew? Is there anything new?

A possible starting point year 2011

In 2011, the first organization dedicated to the growth and success of SDN began with the Open Networking Foundation (ONF). Among its stated missions was to evolve the OpenFlow protocol from its academic roots to a commercially viable substrate for building networks and networking products. Within two years, the ONF's membership had grown to approximately 100 entities, representing the diverse interest and expectations for SDN..

Some people as Tom Nadeau and Ken Gray realized that SDN was really about general network programmability and the associated interfaces, protocols, data models, and APIs. Using this insight, they helped to organize the SDN Birds of a Feather session at IETF 82, in Taipei, to investigate this more general SDN model. At that meeting, Tom Nadeau presented a framework for software-defined networks that envisioned SDN as a generalized mechanism for network programmability. This work encouraged the community to take a more general view of SDN and eventually led to the formation of the Interface to the Routing System (I2RS) Working Group in the IETF.

More about I2RS later on this course
Is Open Flow the only possible choice for SDN?

* SDN Software Defined Networks By Thomas D. Nadeau & Ken Gray

SDN, how technological shifts can affect the industry and people lifes

In 1996, Gartner coined the term “service-oriented architecture SOA.” By 2000, it had taken center stage with the core purpose of allowing for the easy cooperation of a large number of computers connected over a network to exchange information via services without human interaction. There was no need to make underlying changes to the program or application itself. Essentially, it took on the same role as a single operating system on one machine and applied it to the entire infrastructure of servers, allowing for more usable, flexible, and scalable applications and services to be built, tested, deployed, and managed. It introduced web services as the de facto way to make functional building blocks accessible over standard Internet protocols independent of platforms and languages— allowing for faster and easier development, testing, deployment, and manageability of IT infrastructures. SOA drastically changed the way developers, their managers, and the business looked at technology.

SDN is not so different. The network is the cornerstone of IT in that it can enable new architectures that in turn create new business opportunities. In essence, it allows IT to become more relevant than ever and the enabler of new business.

The network is now the largest business enabler if architected and utilized in the correct way—allowing for the network, server, and storage to be tied together to enable the principles of SOA to be executed at the network layer. SDN and APIs to the network change the accessibility to programming intent and receiving state from the network and services, thus overcoming the traditional view that the network has to be built and run by magicians.

SOA is to the entire infrastructure, what SO is to one machine

* SDN Software Defined Networks By Thomas D. Nadeau & Ken Gray

But, What is SDN?

SDN impacts in different fields, it can be used keeping in mind different pourposes…..so

A pragmatic definition could be this: SDN functionally enables the network to be accessed by operators programmatically, allowing for automated management and orchestration techniques; application of configuration policy across multiple routers, switches, and servers; and the decoupling of the application that performs these operations from the network device's operating system.

Historically, network configuration state has remained largely static, unchanged, and commonly untouchable. Manual configuration and CLI-based configuration on a device-by-device basis was the norm, and network management constituted the basic "screen scraping".

when you're dealing with multiple routers, switches, and servers working as a system (and services that are routing traffic across multiple domains with different users, permissions, and policies), control and management state needs to be applied across the network as an operation.

What's a bit different now is that one major functionality of the SDN architecture is the ability to write applications on top of a platform that customizes data from different sources or data bases into one network-wide operation.

But, What is SDN?

SDN impacts in different fields, it can be used keeping in mind different pourposes…..so

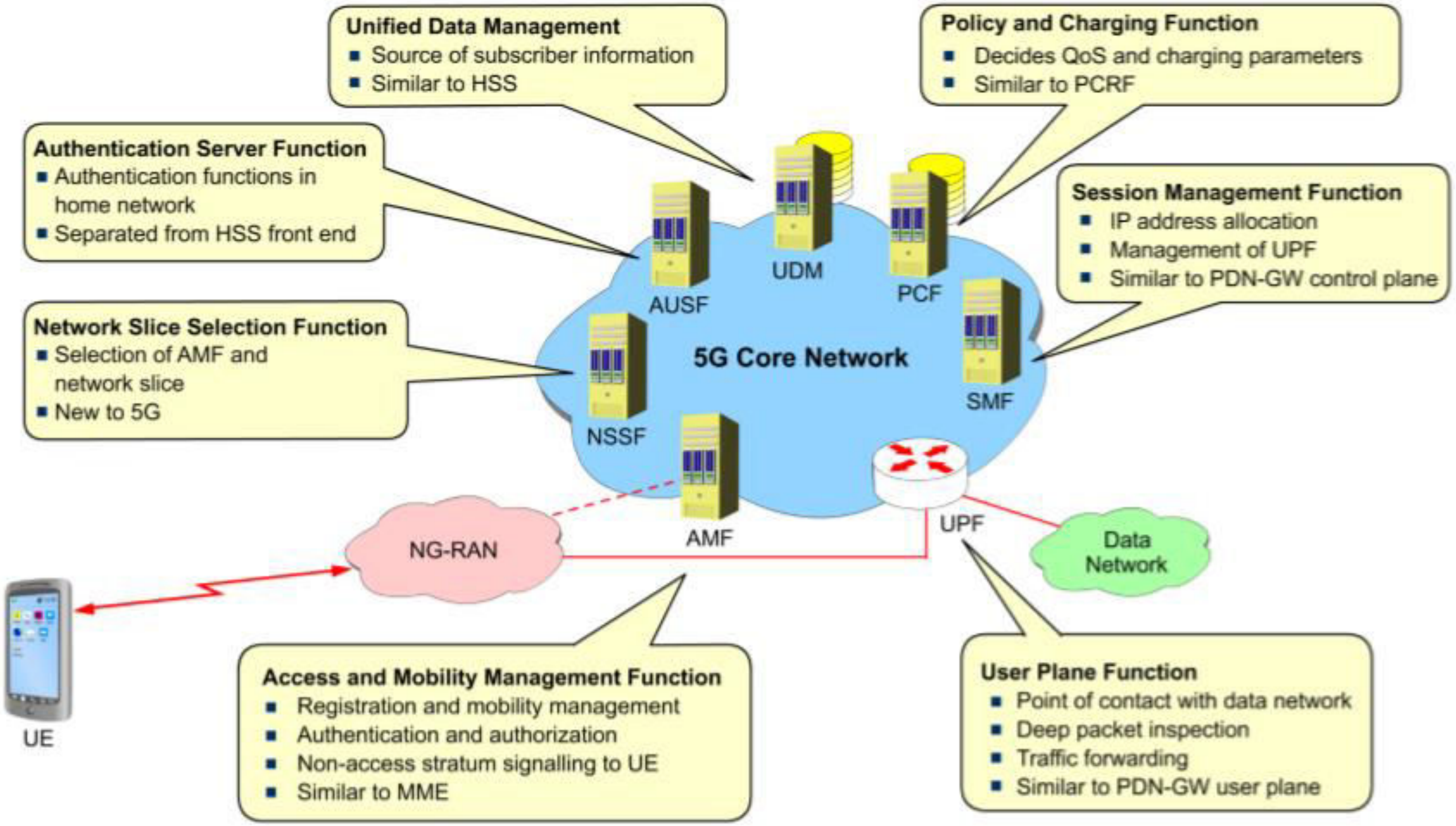
SDN is also an architecture that allows for a centrally managed and distributed control, management, and data plane, where policy that dictates the forwarding rules is centralized, while the actual forwarding rule processing is distributed among multiple devices. In this model, application policy calculation (e.g., QoS, access control lists, and tunnel creation) happens locally in real time and the quality, security, and monitoring of policies are managed centrally and then pushed (*) to the switching/routing nodes.

(*) Packet Forwarding Control Protocol (PFCP) adopted by 3GPP for User plane control in 5GC is quite similar

PFCP
-Defined by 3GPP
-TS 29.244
-One of the main protocols introduced in the 5GC
-Also used in the 4G/TE EPC for Control and User Plane Separation (CUPS). Attention: not used un EPC, but in EPC+.
Scope of PFCP is similar to that of OPenFlow (designed for SDN).
-Unlike OpenFlow, PFCP it was engineered to serve the particular use-case of 5GC, while OpenFlow is also applicable for fixed networks.
-PFCP is also used on the interface between the control plane and user plane functions of a disaggregated BNG. See TR-459 (Broadband Forum)

But, What is SDN?

SDN impacts in different fields, it can be used keeping in mind different purposes…..so



Unified Data Management
- Source of subscriber information
- Similar to HSS

Policy and Charging Function
- Decides QoS and charging parameters
- Similar to PCRF

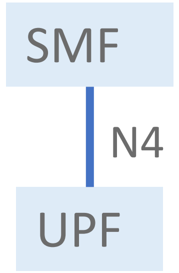
Authentication Server Function
- Authentication functions in home network
- Separated from HSS front end

Session Management Function
- IP address allocation
- Management of UPF
- Similar to PDN-GW control plane

Network Slice Selection Function
- Selection of AMF and network slice
- New to 5G

Access and Mobility Management Function
- Registration and mobility management
- Authentication and authorization
- Non-access stratum signalling to UE
- Similar to MME

User Plane Function
- Point of contact with data network
- Deep packet inspection
- Traffic forwarding
- Similar to PDN-GW user plane



N4 protocols. N4 is the interface binding the control plane and user plane of the 5G packet gateway. 3GPP defines the usage of the **Packet Forwarding Control Protocol (PFCP)** for the communication between the control and user plane elements affected by CUPS (Control and User Plane Separation).
In addition, it is planned to reuse PFCP, with some enhancements, in the interface N4. The user plane is done via extension of GTP U (over UDP).

But, What is SDN?

SDN doesn't replace the control plane on the router or switch. It augments them. How? By having a view of the entire network all at once versus only from one position in the topology (e.g., the router or switch).

However, OpenFlow is only one protocol and one element of SDN. There are many other protocols now. Some examples include I2RS, PCE-P, BGP-LS, FORCES, OMI, and NetConf/Yang. All of these are also open standards. What's important to remember is that SDN is not a protocol; SDN is an operational and programming architecture.

What do we get from SDN? The architecture brings the network and networking data closer to the application layer and the applications closer to the networking layer. As practiced in SOA, no longer is there the need for a human element or scripting languages to act as humans to distribute data and information bidirectionally because APIs and tooling now have evolved in a way that this can be delivered in a secure and scalable way via open interfaces and interoperability. The data in the network (e.g., stats, state, subscriber info, service state, security, peering, etc.) can be analyzed and used by an application to create policy intent and program the network into a new configuration. It can be programmed this way persistently or only ephemerally.

But, What is SDN?

Programmability (i.e., the ability to access the network via APIs and open interfaces) is central to SDN. The notion of removing the control and management planes to an off-switch/router application connected to the networking device by SDN protocols is equally important. This off-box application is really what software developers would call a “platform,” as it has its own set of APIs, logic, and the ability for an application to make requests to the network, receive events, and speak the SDN protocols. What’s key here is that programmers don’t need to know the SDN protocols because they write to the controller’s APIs. Programmers don’t need to know the different configuration syntax or semantics of different networking devices because they program to a set of APIs on the controller that can speak to many different devices.

Different vendors, eras of equipment, and classes of equipment (e.g., transport, simple switches, wireless base stations, subscriber termination gateways, peering routers, core routers, and servers) all are on the trajectory to be able to be programmed by the SDN protocols that plug into the bottom of the controller.

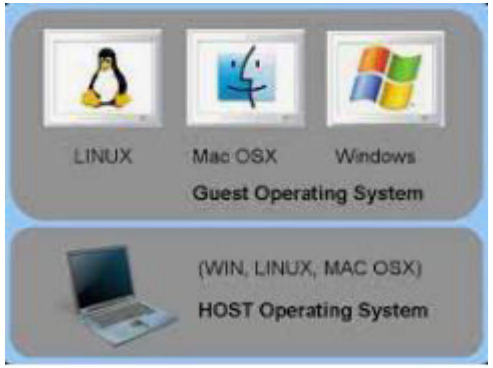
The programmer only uses the APIs on the top of the controller to automate, orchestrate, and operate the network. This doesn’t necessarily mean there is a grand unification theory of controllers and one to serve all layers and functions of networking, but what it does mean is that the network now has been abstracted and is being programmed off box. Thus, when integrated into an IaaS (Infrastructure as a Service) layer in a stack, OSS, or IT system, the network is being automated and orchestrated as fast as users log onto the net and as fast as workloads are being spun up on servers.

EVOLUTION PATH FROM ISOLATED RESOURCES TO MODERN D.C

Historically, computing and storage functions—which were executed on the often thousands (or more) of desktops within an enterprise organization—were handled by departmental servers that provided services dedicated only to local use.

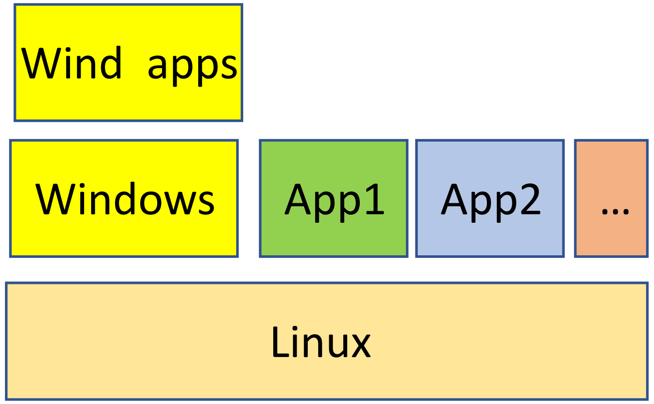
Data centers were originally designed to physically separate traditional computing elements (e.g., PC servers), their associated storage, and the networks that interconnected them with client users. The computing power that existed in these types of data centers became focused on specific server functionality—running applications such as mail servers, database servers, or other such widely used functionality in order to serve desktop clients

Latelly, the departmental servers migrated into the data center for a variety of reasons—first and foremost, to facilitate ease of management, and second, to enable sharing among the enterprise's users.



VMware had invented an interesting technology that allowed a host operating system such as one of the popular Linux distributions to execute one or more client operating systems (e.g., Windows). What VMware did was to create a small program that created a virtual environment that synthesized a real computing environment (e.g., virtual NIC, BIOS, sound adapter, and video). It then marshaled real resources between the virtual machines. This supervisory program was called a ***hypervisor*.**

THE IDEA BEHIND THE HYPERVISOR CONCEPT: THE ORIGEN


Wind apps

Windows

App1

App2

…

Linux

VMWARE engineers wanted to run Linux for most of computing needs (perhaps because they loved it) and Windows only for those situations that required that specific OS environment to execute (may be because of Corporation policies).

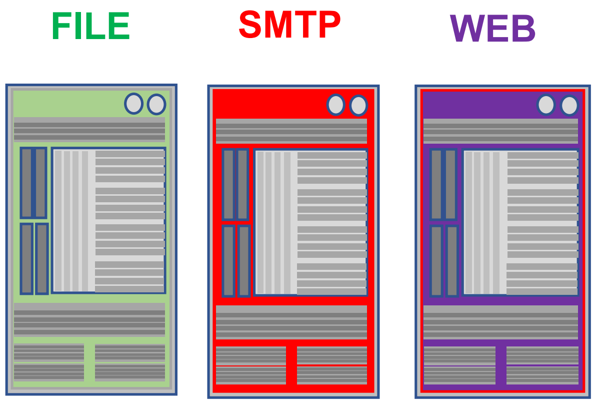
When not needed, Windows would be closed as if it were another program, and continue on with Linux.

All client operating systems can be treated as if it were just a program consisting of a file (albeit large) that existed on the hard disk.

That file could be manipulated as any other file could be (i.e., it could be moved or copied to other machines and executed there as if it were running on the machine on which it was originally installed).

Even more interestingly, the operating system could be paused without it knowing, essentially causing it to enter into a state of suspended animation.

THE DATA CENTER ENVIRONMENT



With the advent of operating system virtualization, the servers that typically ran a single, dedicated operating system, such as Microsoft Windows Server, and the applications specifically tailored for that operating system could now be viewed as a ubiquitous computing and storage platform. With further advances and increases in memory, computing, and storage, data center compute servers were increasingly capable of executing a variety of operating systems simultaneously in a virtual environment.

Vmware expanded its single-host version to a more data-center-friendly environment that was capable of executing and controlling many hundreds or thousands of virtual machines from a single console. Operating systems such as Windows Server that previously occupied an entire “bare metal” machine were now executed as virtual machines, each running whatever applications client users demanded. The only difference was that each was executing in its own self-contained environment that could be paused, relocated, cloned, or copied (i.e., as a backup). Thus began the age of *elastic computing*.

ELASTIC COMPUTING ENVIRONMENT

Within the elastic computing environment, operations departments were able to move servers to any physical data center location simply by pausing a virtual machine and copying a file. They could even spin up new virtual machines simply by cloning the same file and telling the hypervisor to execute it as a new instance.

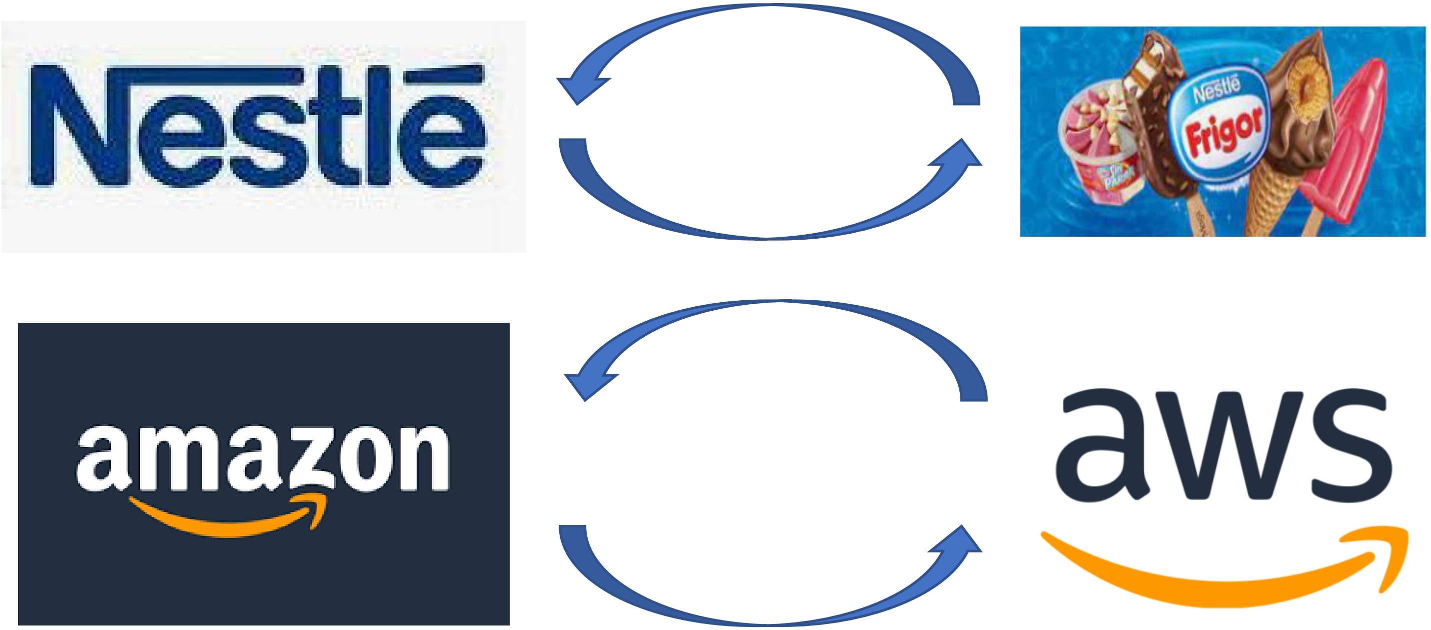
This flexibility allowed network operators to start optimizing the data center resource location and thus utilization based on metrics such as power and cooling. By packing together all active machines, an operator could turn down cooling in another part of a data center by sleeping or idling entire banks or rows of physical machines, thus optimizing the cooling load on a data center. Similarly, an operator could move or dynamically expand computing, storage, or network resources by geographical demand.

EQUIPMENT CONSUME POWER, EVEN IF THEY ARE NOT USED

As with all advances in technology, this newly discovered flexibility in operational deployment of computing, storage, and networking resources brought about a new problem: one not only of operational efficiency both in terms of maximizing the utilization of storage and computing power, but also in terms of power and cooling. As mentioned earlier, network operators began to realize that computing power demand in general increased over time. To keep up with this demand, IT departments (which typically budget on a yearly basis) would order all the equipment they predicted would be needed for the following year. However, once this equipment arrived and was placed in racks, it would consume power, cooling, and space resources—even if it was not yet used! This was the dilemma discovered first at Amazon.

* SDN Software Defined Networks By
Thomas D. Nadeau & Ken Gray

SWAP RESOURCE POOL FROM ONE SERVICE TO OTHER



At the time, Amazon's business was growing at the rate of a "hock nine months. As a result, growth had to stay ahead of demand for its computing services, which served its retail ordering, stock, and warehouse management systems, as well as internal IT systems. As a result, Amazon's IT department was forced to order large quantities of storage, network, and computing resources in advance, but faced the dilemma of having that equipment sit idle until the demand caught up with those resources. Amazon Web Services (AWS) was invented as a way to commercialize this unused resource pool so that it would be utilized at a rate closer to 100%. When internal resources needed more resources, AWS would simply push off retail users, and when it was not, retail compute users could use up the unused resources. Some call this elastic computing services, but Thomas D. Nadeau & Ken Gray call it *hyper virtualization*.

* SDN Software Defined Networks By Thomas D. Nadeau & Ken Gray

THE INEFFICIENCY

First hyperscalers as Amazon and Rackspace faced a huge problem
Buying storage and computing in huge quantities for pricing efficiency was not efficient at all.
They realized that their computing and storage were not being used in an efficient way.

They could resell their spare computing power and storage to external users in an effort to recoup some of their capital investments.

This gave them a new idea: Multitenant data center.

A new problem arised: How to separate thousands of potential tenants, whose resources needed to be spread arbitrarily across different physical data centers' virtual machines.

THE INEFFICIENCY

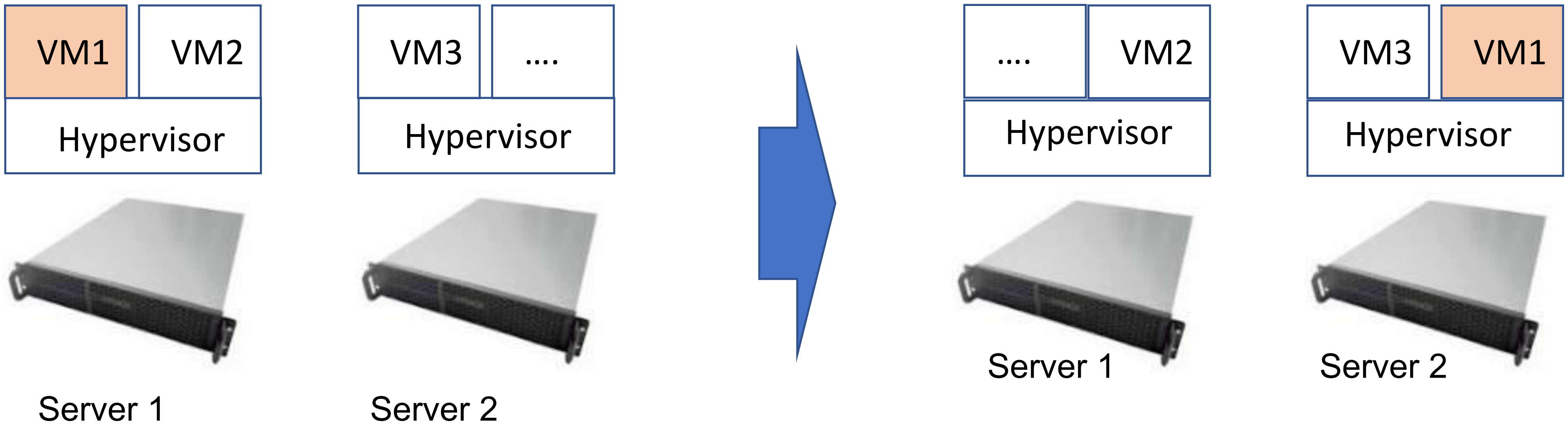
During the moving process to hyper virtualized environments, execution environments were generally run by a single enterprise or organization.

They typically owned and operated all of the computing and storage as if they were a single, flat local area network (LAN) interconnecting a large number of virtual or physical machines and network attached storage.

If the number of departments is small –fewer than 100- as it was the case for some type of networks, the problem was easily solved using existing tools such as layer 2 or layer 3 MPLS VPNs. In both cases, though, the network components that linked all of the computing and storage resources up until that point were rather simplistic.

It was generally a flat Ethernet LAN that connected all of the physical and virtual machines. Most of these environments assigned IP addresses to all of the devices (virtual or physical) in the network from a single network (perhaps with IP subnets), as a single enterprise owned the machines and needed access to them. This also meant that it was generally not a problem moving virtual machines between different data centers located within that enterprise because, again, they all fell within the same routed domain and could reach one another regardless of physical location.

THE MOVE OF A VM TO ANOTHER SERVER



A multitenant data center means that all resources can be offered in ISOLATED ways (like network slicing).
Those environment allowed for the execution of any number of operating systems and applications on top of those operating systems, but each needed a unique network address if it was to be accessed by its owner or other external users such as customer.

In the past, addresses could be assigned from a single, internal block of possibly private addresses and routed internally easily. Now, however, it was needed to assign unique addresses that are externally routable and accessible.

Each VM in question had a unique layer 2 address as well. When a router delivers a packet, it ultimately has to deliver a packet using Ethernet (not just IP). This is generally not an issue until you consider virtual machine mobility (*VM mobility*). In these cases, virtual machines are relocated for power, cooling, or computing compacting reasons.
But physical relocation means physical address relocation. It also possibly means changes to layer 3 routing in order to ensure packets previously destined for that machine in its original location can now be changed to its new location.

THE MOVE OF A VM TO ANOTHER SERVER

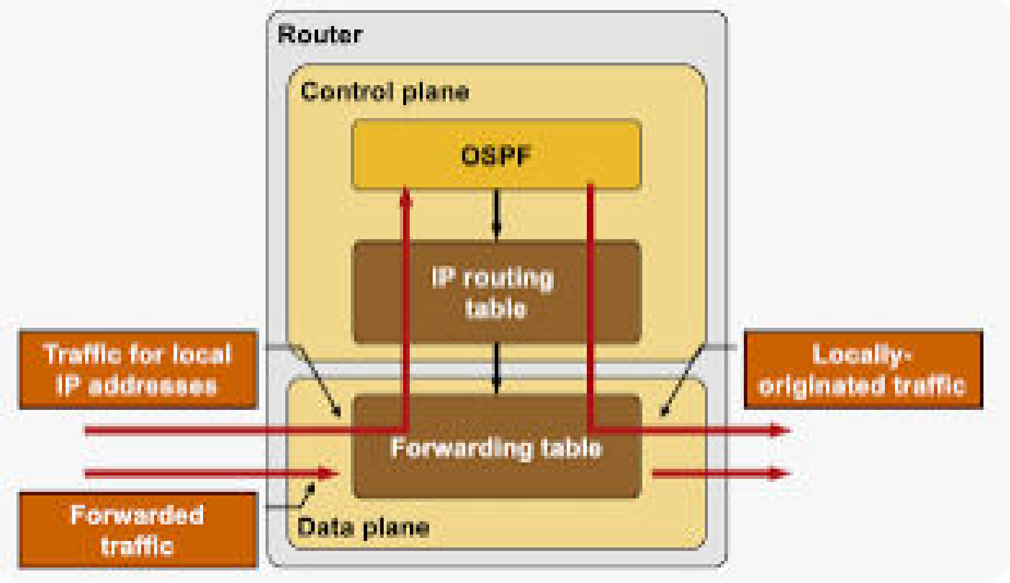
Network equipement –apart from some increase in switch fabric capacities-had not evolved much since the advent of IP, MPLS, and mobile technologies

IP and MPLS allowed a network operator to create networks and virtual network overlays on top of those base networks much in the way that data center operators were able to create virtual machines to run over physical ones with the advent of computing virtualization.

Network virtualization was generally referred to as *virtual private networks* (VPN) and came in a number of flavors, including point-to-point (e.g., a personal VPN as you might run on your laptop and connect to your corporate network); layer 3 (virtualizing an IP or routed network in cases such as to allow a network operator to securely host enterprise in a manner that isolated their traffic from other enterprise); and layer 2 VPNs (switched network virtualization that isolates similarly to a layer 3 VPN except that the addresses used are Ethernet).

THE CLASSICAL ROUTERS AND THEIR INTERFACES

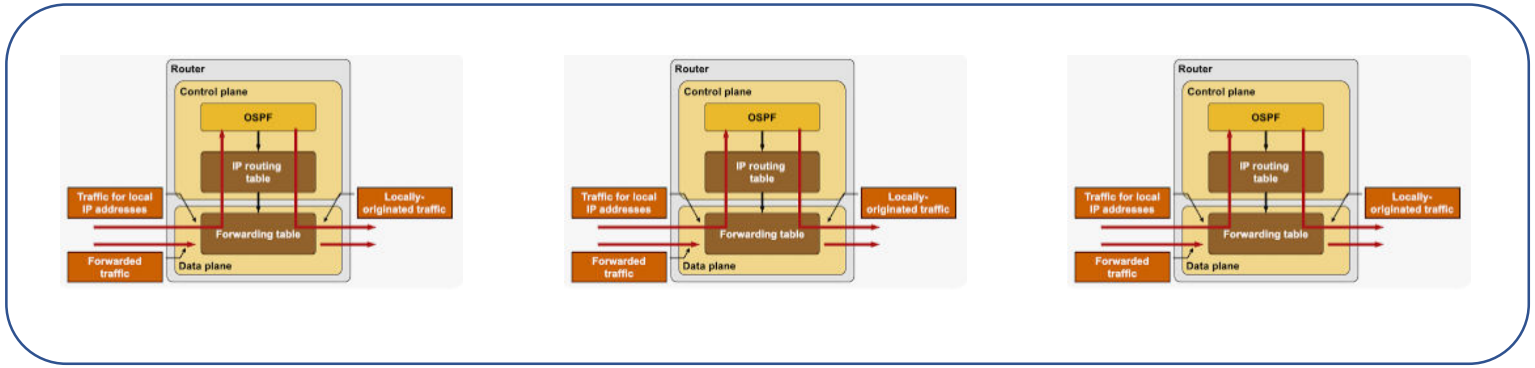
Commercial routers and switches typically come with management interfaces -command line interfaces, XML/Netconf, graphical user interfaces (GUIs), and the Simple Network Management Protocol (SNMP)- that allow a network operator to configure and otherwise manage these devices



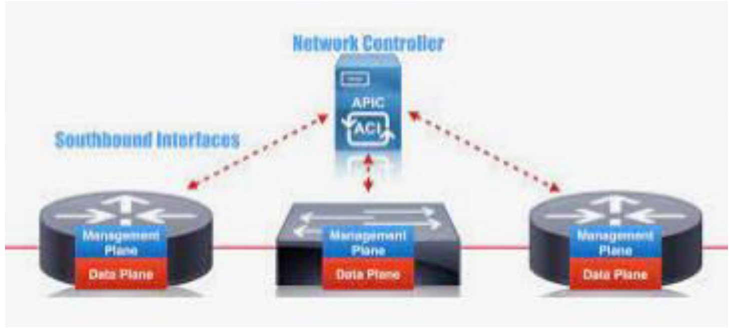
These interface allow an operator suitable access to a device's capabilities, but still often hide the lowest levels of details from the operator. There are no possibility to program them or to use a new routing protocol unless the customer makes a feature enhancement request of a device vendor, which means to wait some amount of time (several years in some cases, if the requirement is accepted)

THE DISTRIBUTED AND CENTRALIZED CONTROL PLANE

Each device has its forwarding plane (usually quite specialized), and also its control plane (usually general-purpose computing)



A network device is comprised of a data plane that is often a switch fabric connecting the various network ports on a device and a control plane that is the brains of a device. Each device in the network has a control plane that implements the protocol. Each node communicate with each other to coordinate network path construction



A centralized control plane paradigm involves one single (or at least logical) control plane (the north) which would push down commands to each device, thus commanding it to manipulate its physical switching and routing hardware.

THE MOTIVATION FOR SDN

- Network device vendors were not meeting customer needs, particularly in the feature development and innovation spaces.

- High-end routing and switching equipment highly overpriced for at least the control plane components of their devices.

- The cost of raw, elastic computing power diminishing rapidly to the point where having thousands of processors at one's disposal was a reality.

- Take advantage of this processing power to run a logically centralized control plane and potentially even use inexpensive, commodity-priced switching hardware.

- A few engineers from Stanford University created a protocol called OpenFlow that could be implemented in just such a configuration. OpenFlow was architected for a number of devices containing only data planes to respond to commands sent to them from a (logically) centralized controller that housed the single control plane for that network.

- Based on this basic architecture just described, one can now imagine how quickly and easily it was to devise a new networking protocol by simply implementing it within a data center on commodity priced hardware. Even better, one could implement it in an elastic computing environment in a virtual machine.

See: Open Networking Foundation (ONF)

SDN WILL CO-EXIST WITH CLASSICAL NETWORKS

software-driven networks is a slightly different view in comparison with SDN

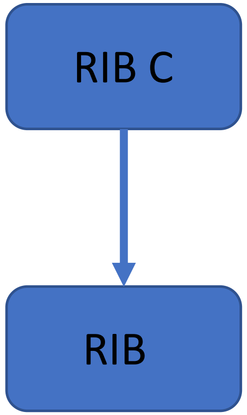
In the software-driven approach, OpenFlow and that architecture is seen as a distinct subset of functionality that is possible.
Rather than viewing the network as being comprised of logically centralized control planes with brainless network devices, one views the world as more of a hybrid of the old and the new.

As in other network domains, it is unrealistic to think that existing networks are going to be dismantled wholesale to make way for a new world proposed by the ONF and software-defined networks.

It is also unrealistic to discard all of the advances in network technology that exist today and are responsible for things like the Internet. Instead, there is more likely a hybrid approach whereby some portion of networks are operated by a logically centralized controller, while other parts would be run by the more traditional distributed control plane. This would also imply that those two worlds would need to interwork with each other (AGAIN!!!).

LOCATION AND PROGRAMABILITY (how and where)

It was said that one of major targets was greater and more flexible network device programmability. [HOW]



At the same time the location of the network control and data planes, what is a different issue, is solved by SDN. [WHERE]

Juniper, Cisco, Level3, and other vendors and service providers have spearheaded an effort around network programmability called the Interface to the Routing System (I2RS).
There were a lot of contributions and several IETF drafts, including the primary requirements and framework Drafts which lead to RFC 7921 “An Architecture for the Interface to the Routing System” by Alia Atlas Juniper Networks, J. Halpern Ericsson, S. Hares Huawei , David Ward Cisco Systems and T. Nadeau Brocade, June 2016.

The basic idea around I2RS is to create a protocol and components to act as a means of programming a network device’s routing information base (RIB) using a fast path protocol that allows for a quick cut-through of provisioning operations in order to allow for real-time interaction with the RIB and the RIB manager that controls it. Previously, the only access one had to the RIB was via the device’s configuration system (in Juniper’s case, Netconf or SNMP).

IR2S ADOPTION

I2RS is *not* just another provisioning protocol.
There are a number of other key concepts that comprise an entire solution to the overarching problem of speeding up the feedback loop between network elements, network programming, state and statistical gathering, and post-processing analytics. Today, this loop is painfully slow. Those involved in I2RS believe the key to the future of programmable networks lies within optimizing this loop.

I2RS provides varying levels of abstraction in terms of programmability of network paths, policies, and port configuration, but in all cases has the advantage of allowing for adult supervision of said programming as a means of checking the commands prior to committing them.

Some protocols exist today for programming at the hardware abstraction layer (HAL), which is far too granular or detailed for the network's efficiency and in fact places undue burden on its operational systems.

Another example is providing operational support systems (OSS) applications quick and optimal access to the RIB in order to quickly program changes and then witness the results, only to be able to quickly reprogram in order to optimize the network's behavior.

One key aspect around all of these examples is that the discourse between the applications and the RIB occur via the RIB manager. It allows operators to preserve their operational and workflow investment in routing protocol intelligence that exists in device operating systems such as Junos (Juniper) or IOS-XR (Cisco) while leveraging this new and useful programmability paradigm to allow additional levels of optimization in their networks.

IR2S ADOPTION

I2RS is normalized and abstracted topology (see RFC 7921).

This topology is represented by a common and extensible object model defined in the RFC.

The service also allows for multiple abstractions of topological representation to be exposed.

As a consequence of this model nonrouters (or routing protocol speakers) can more easily manipulate and change the RIB state going forward.
In the past (and also today in some networks) nonrouters have a major difficulty getting at this information at best.

Going forward, components of a network management/OSS, analytics, or other applications that we cannot yet envision will be able to interact quickly and efficiently with routing state and network topology.

SDN: SOME POSSIBLE DEFINITIONS

Software-defined networks (SDN): an architectural approach that optimizes and simplifies network operations by more closely binding the interaction (i.e., provisioning, messaging, and alarming) among applications and network services and devices, whether they be real or virtualized. It often is achieved by employing a point of logically centralized network control—which is often realized as an SDN controller—which then orchestrates, mediates, and facilitates communication between applications wishing to interact with network elements and network elements wishing to convey information to those applications. The controller then exposes and abstracts network functions and operations via modern, application-friendly and bidirectional programmatic interfaces.

* SDN Software Defined Networks By Thomas D. Nadeau & Ken Gray

Software-Defined Networking (SDN) is an approach to networking that uses software-based controllers or application programming interfaces (APIs) to communicate with underlying hardware infrastructure and direct traffic on a network.
This model differs from that of traditional networks, which use dedicated hardware devices (i.e., routers and switches) to control network traffic. SDN can create and control a virtual network – or control a traditional hardware – via software.

* VMWare

Software-defined networking (SDN) describes an architecture that separates the network control plane and the forwarding plane, aiming to simplify and improve network control. IT teams are better able to rapidly adapt to changing business requirements and application needs.2

* CIENA

SDN: A COMPLEX ECOSYSTEM

Different approaches

- Software-defined
- Software-driven
- Programmable networks
- Rich and complex set of historical lineage
- So many Challenges
- Variety of solutions (or aproaches) to those problems.

Only possible because of technologies that preceded software defined, software-driven, and programmable networks

- IP
- BGP
- MPLS
- Ethernet.

Virtualization technology today is based on the technologies started by VMware years ago and continues to be the basis on which it and other products are based. Network attached storage enjoys a similarly rich history.

CENTRALIZED AND DISTRIBUTED CONTROL AND DATA PLANE

Different approaches and a bit of controversial

Centralized or semi-centralized programmatic control.

Separation of a network device's control and data planes

How far away the control plane can be located from the data plane?

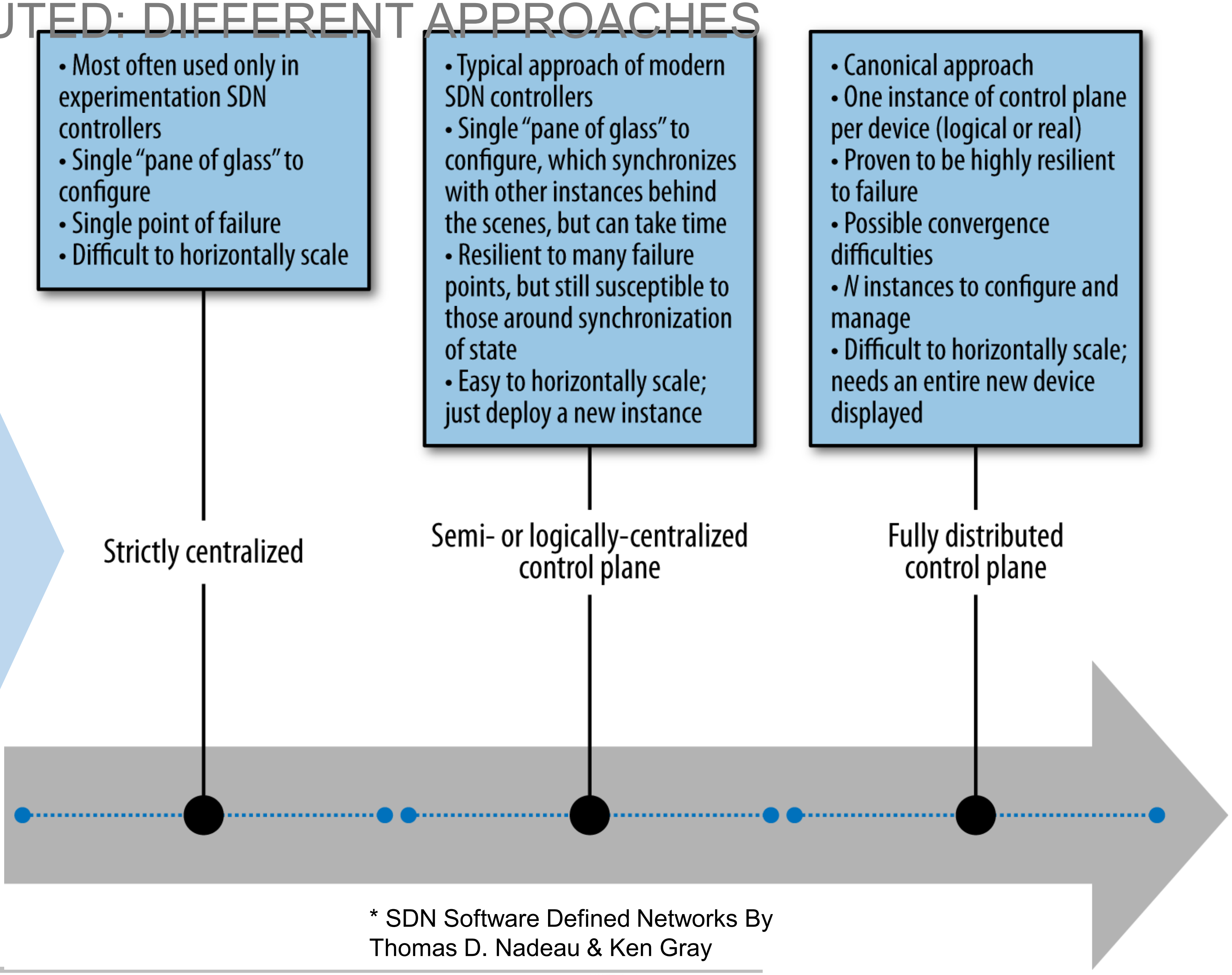
How many instances are needed to exist to satisfy resiliency and high-availability requirements

Less expensive, so-called commodity hardware.

100% of the control plane can be, in fact, relocated further away than a few inches?

CENTRALIZED AND DISTRIBUTED: DIFFERENT APPROACHES

- From the simplest, being the canonical fully distributed control plane,
- to the semi- or logically centralized control plane,
- to finally the strictly centralized control plane



Strictly centralized

- Most often used only in experimentation SDN controllers
- Single "pane of glass" to configure
- Single point of failure
- Difficult to horizontally scale

Semi- or logically-centralized control plane

- Typical approach of modern SDN controllers
- Single "pane of glass" to configure, which synchronizes with other instances behind the scenes, but can take time
- Resilient to many failure points, but still susceptible to those around synchronization of state
- Easy to horizontally scale; just deploy a new instance

Fully distributed control plane

- Canonical approach
- One instance of control plane per device (logical or real)
- Proven to be highly resilient to failure
- Possible convergence difficulties
- *N* instances to configure and manage
- Difficult to horizontally scale; needs an entire new device displayed

* SDN Software Defined Networks By Thomas D. Nadeau & Ken Gray

FULLY CENTRALIZED CONTROL

- *Revolutionary* proponents
- Clean slate approach
- Control plane of a network is completely centralized.

In most cases, this extreme approach has been tempered to be, in reality, a logically centralized approach due to either scale or high availability requirements that make a strictly centralized approach difficult.

- No control plane functions effectively exist at a device
- Device is a dumb (albeit fast) switching device under the *total* control of the remotely located, centralized control plane.
- Generally applies best to newly deployed networks rather than existing ones.

CENTRALIZED HIBRID CONTROL

Evolutionary proponents

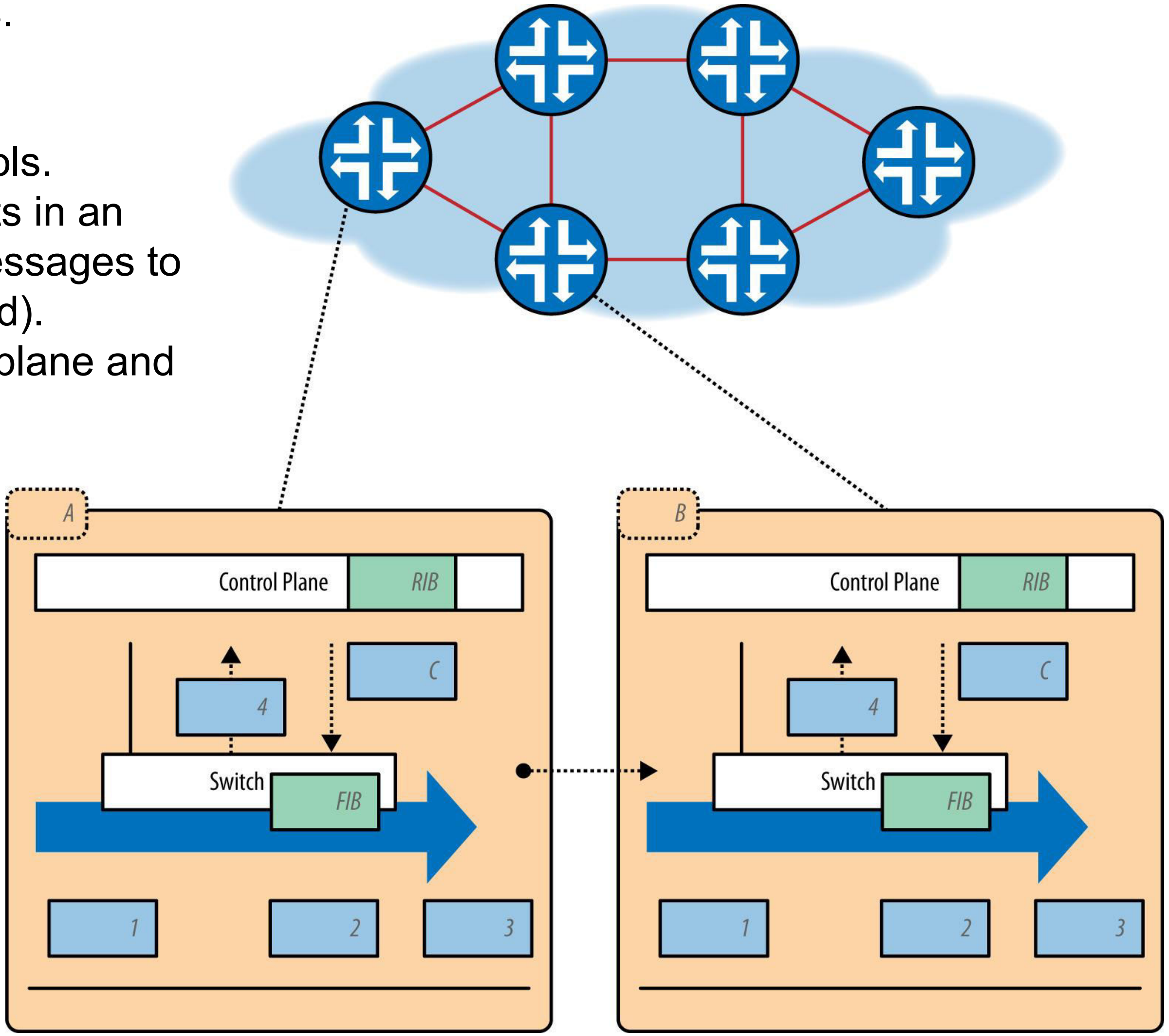
- General definition of networks in which a centralized control paradigm provides *some* new capabilities, but does *not* replace every capability nor does it completely remove the control plane from the device.
- This paradigm typically works in conjunction with a distributed control plane in some fashion, meaning that the device retains some classical control plane functions (e.g., ARP processing or MAC address learning), while allowing a centralized controller to manipulate other areas of functionality more convenient for that operational paradigm.
- Often characterized as the hybrid operation or as part of the underlay/overlay concept in which the distributed control plane provides the underlay and the centralized control plane provides a logical overlay that utilizes the underlay as a network transport.

Classic use of control planes

- Completely distributed.
- Every device runs a complete instance of a control plane in addition to at least one data plane.
- Each independent control plane must cooperate with the other control planes in order to support a cohesive and operational network.
- Nothing new and is neither revolutionary nor evolutionary.

CONTROL PLANE

- A packet is received that comes from an unknown MAC address.
- It is punted or redirected (4) to the control plane of the device.
- It is learned, processed, and later forwarded onward.
- Same treatment is given to control traffic such as routing protocols.
- Information contained in packet is processed and possibly results in an alteration of the RIB as well as the transmission of additional messages to its peers, alerting them of this update (i.e., a new route is learned).
- The RIB becomes stable, the FIB is updated in both the control plane and the data plane.
- Forwarding will be updated and reflect these changes.

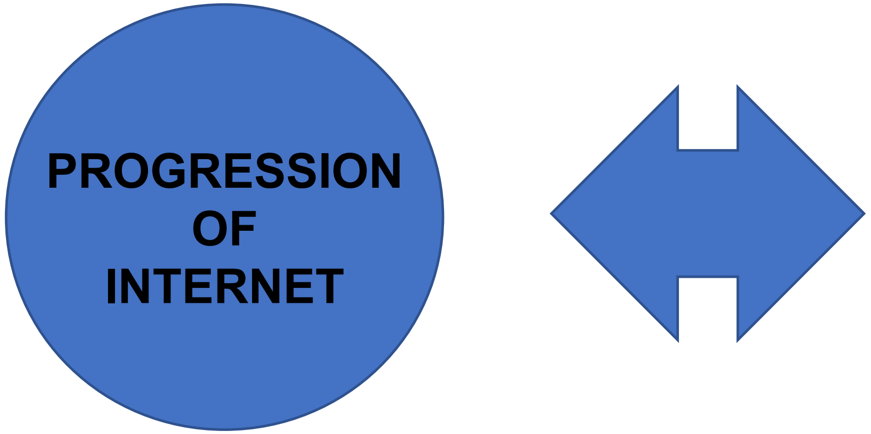


However, (the packet received was one of an unlearned MAC address), the control plane returns the packet (C) to the data plane (2), which forwards the packet accordingly (3). If additional FIB programming is needed, this also takes place in the (C) step, which would be the case for now the MAC addresses source has been learned.

* SDN Software Defined Networks By Thomas D. Nadeau & Ken Gray

Routing information base (RIB)
Forwarding information base (FIB).

CONTROL PLANE



- Evolution of control schemes for managing reachability information.
- Protocols for the distribution of reachability information.
- Increasing growth of the information base used (i.e., route table size growth).
- High rates of change in the network or even nonoperation.
- Diffusion of responsibility for advertising reachability to parts of the destination/target data

Progression of Internet is consequence of layer 2 and layer 3 evolution both in terms of functionality and hardware

Layer 2 control plane (hardware or physical layer addresses such as IEEE MAC addresses). Learning MAC addresses, the use of the mechanisms to guarantee an acyclic graph (as the Spanning Tree Protocol), and flooding of BUM (broadcast, unicast unknown, and multicast) traffic create their own scalability challenges and also reveal their scalability limitations.

In a layer 2 network, forwarding focuses on the reachability of MAC addresses. Layer 2 networks primarily deal with the storage of MAC addresses for forwarding purposes (enormous number of host).

CONTROL PLANE

- In a layer 3 network, forwarding focuses on the reachability of network addresses (a destination IP prefix)
- Layer 3 networking is used to segment or stitch together layer 2 domains in order to overcome layer 2 scale problems.
- Specifically, layer 2 bridges that represent some sets of IP subnetworks are typically connected together with a layer 3 router.
- Layer 3 routers are connected together to form larger networks—or really different subnetwork address ranges. Larger networks connect to other networks via *gateway* routers that often specialize in simply interconnecting large networks.
- The router routes traffic between networks at layer 3 and will only forward packets at layer 2 when it knows the packet has arrived at the final destination layer 3 network that must then be delivered to a specific host.

The Multiprotocol Label Switching (MPLS) protocol, the Ethernet Virtual Private Network (EVPN) protocol, and the Locator/ID Separation Protocol (LISP) blurs the separation line between L2 and L3.

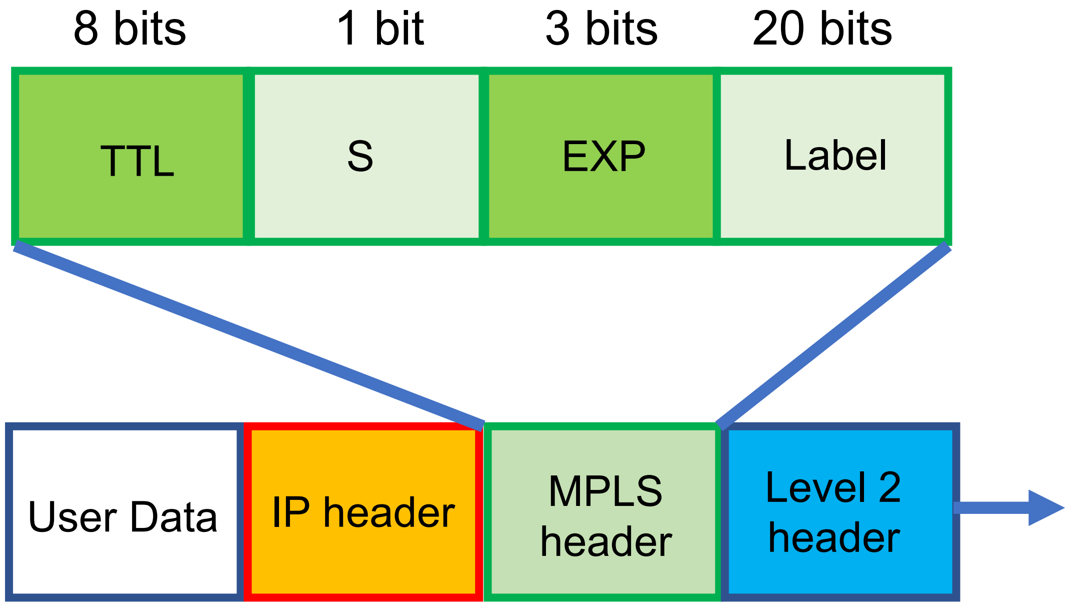
Can we say that there are L2-L3 devices?

CONTROL PLANE

How to deal with layer 2 scale problems.
MPLS

The suite MPLS includes a) the best parts of layer 2 forwarding (or switching) with b) the best parts of layer 3 IP routing to form a technology that shares the extremely fast-packet forwarding that ATM invented with the very flexible and complex path signaling techniques adopted from the IP world.

Multiprotocol label switching (MPLS) is a technology designed to improve the speed and efficiency of data forwarding over large networks or at network edge locations. It offers resiliency and low latency as it runs on a virtual private network (VPN) and can be integrated with any underlying infrastructure, such as Internet Protocol (IP), Ethernet, Frame Relay protocol, and asynchronous transfer mode (ATM).



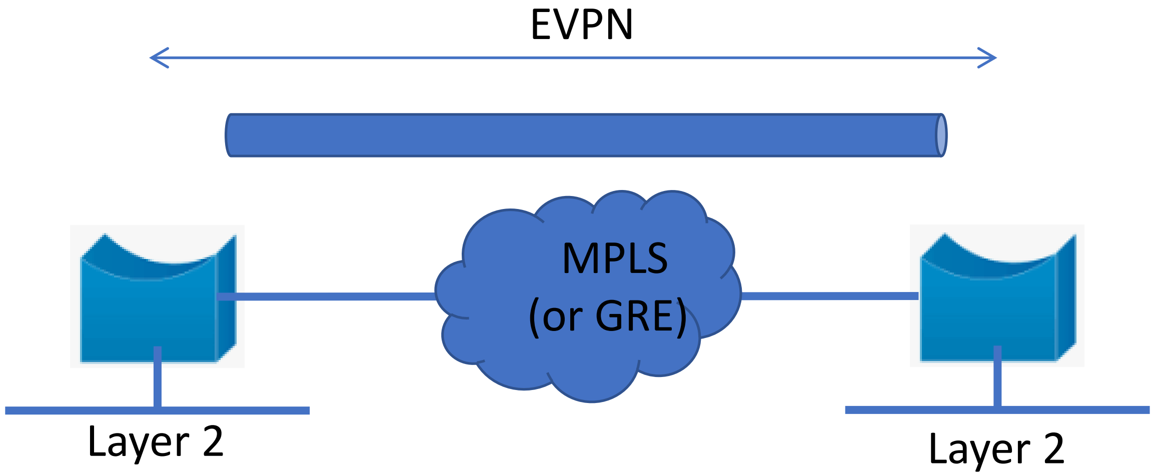
CONTROL PLANE

How to deal with layer 2 scale problems. How to overcome layer 2 scale problems

EVPN

Ethernet VPN (EVPN) is a standards-based technology that provides virtual bridged multipoint connectivity between different Layer 2 domains over an IP or IP/MPLS backbone.

Only layer 2 addressing and reachability information is exchanged over these tunnels.
Reachability information between distant bridges is exchanged as data inside a new BGP address family. This is a design that minimizes the need for broadcast and multicast.



We should be familiar with MPLS, address families, EVPN, etc

How to deal with layer 2 scale problems. How to overcome layer 2 scale problems

LISP

LISP (Locator/ID Separation Protocol, RFC 4984) attempts to solve some of the shortcomings of the general distributed control plane model as applied to multihoming, adding new addressing domains and separating the site address from the provider in a new map and encapsulation control and forwarding protocol.

Locator ID Separation Protocol (LISP) is a network architecture and protocol that implements the use of two namespaces instead of a single IP address:

Endpoint identifiers (EIDs)—assigned to end hosts.

Routing locators (RLOCs)—assigned to devices (primarily routers) that make up the global routing system.

How to deal with layer 2 scale problems. How to overcome layer 2 scale problems

LISP

Locator/ID Separation Protocol (LISP) is routing architecture that provides new semantics for IP addressing. The current IP routing and addressing architecture uses a single numbering space, the IP address, to express two pieces of information:
•Device identity
•The way the device attaches to the network
The LISP routing architecture design separates the device identity, or endpoint identifier (EID), from its location, or routing locator (RLOC), into two different numbering spaces. Splitting EID and RLOC functions yields several advantages.

Simplify Routing Operations LISP enables enterprises and service providers to:
•Simplify multihomed routing
•Facilitate scalable any-to-any WAN connectivity
•Support data center virtual machine mobility

Improve Scalability and Support LISP routing architecture also:
•Improves scalability of the routing system through greater aggregation of RLOCs
•Optimizes IP routing for both IPv4 and IPv6 hosts
•Reduces operational complexities
LISP can be gradually introduced into an existing IP network without affecting the network endpoints or hosts.
LISP is a Cisco innovation that is being promoted as an open standard. Cisco participates in standards bodies such as the IETF LISP Working Group to develop the LISP architecture.

Source: Cisco

CONTROL PLANE

At its heart, LISP attempts to solve some of the shortcomings of the general distributed control plane model as applied to multihoming, adding new addressing domains and separating the site address from the provider in a new map and encapsulation control and forwarding protocol.

- At lower level, there are adjunct control processes particular to certain network types.
- Used to augment the knowledge of the greater control plane.
- The services provided by these processes include:

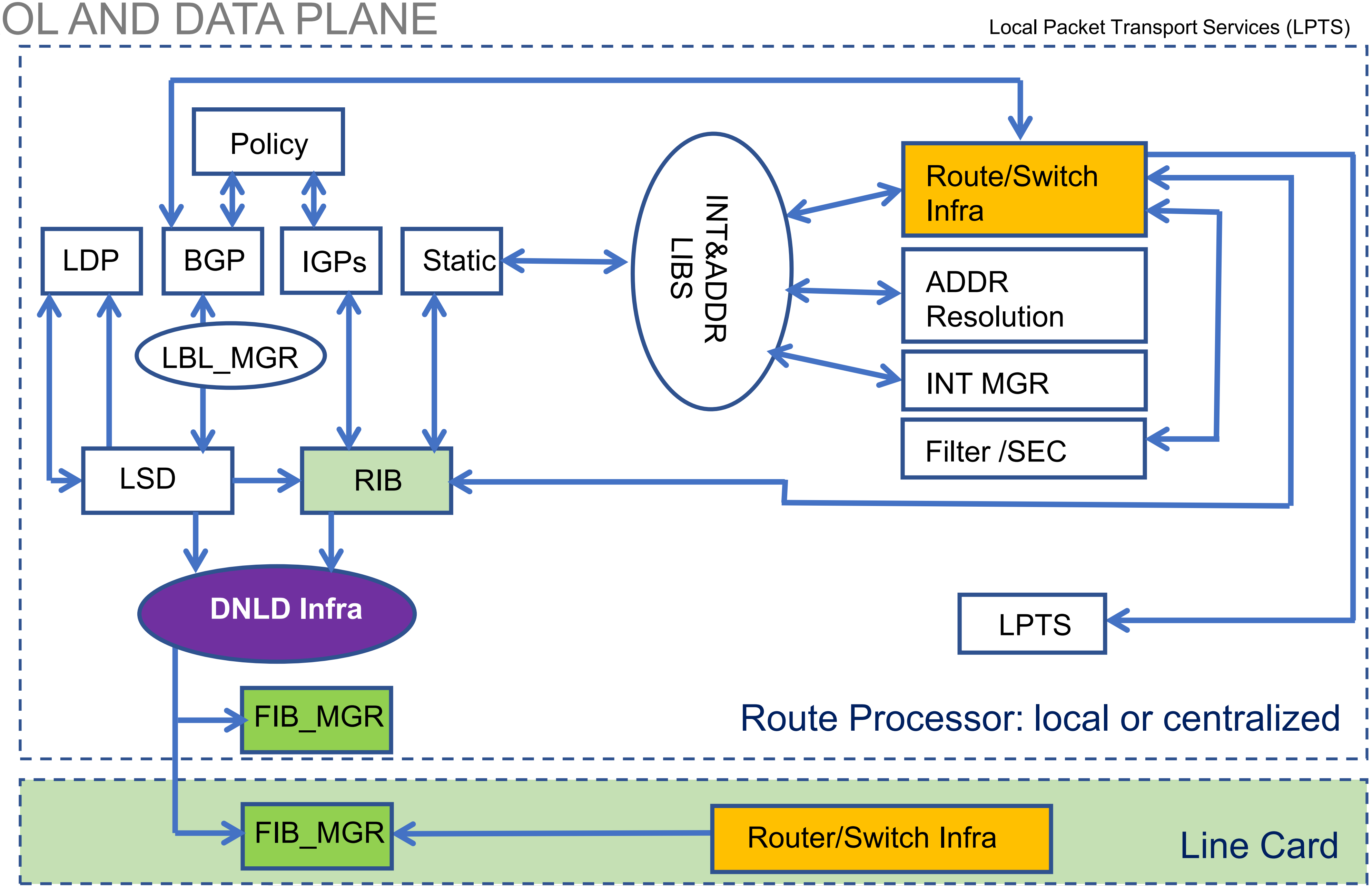
  verification/notification of link availability
  quality information
  neighbor discovery
  address resolution.

These services have very tight performance loops (for short event detection times)
Almost invariably local to the data plane (e.g., OAM)—regardless of the strategy chosen for the control plane.
Various routing protocols as well as RIB-to-FIB control that comprises the heart of the control plane.

Only the data plane resides on the line card.
The control plane is situated on the route processor
The rute processor may be distributed or centralized

Routing information base (RIB)
Forwarding information base (FIB).

CONTROL AND DATA PLANE



DATA PLANE

Normally, a well-formed (i.e., correct) datagram is processed in the data plane by performing lookups in the FIB table (or tables, in some implementations) that are programmed earlier by the control plane (fast path).

The one exception to this processing is when packets cannot be matched to those rules, such as when an unknown destination is detected, and these packets are sent to the route processor where the control plane can further process them using the RIB.

FIB tables could reside in a number of forwarding targets:

- Software
- Hardware-accelerated software (GPU/CPU, as exemplified by Intel or ARM).
- Commodity silicon (NPU, as exemplified by Broadcom, Intel, or Marvell, in the Ethernet switch market)
- FPGA and specialized silicon (ASICs like the Juniper Trio)
- or any combination

DATA PLANE

The *software path:* CPU-driven forwarding of the modern dedicated network element (e.g., router or switch), which trades off a processor intensive lookup.

It can be implemented in the kernel or in the user space, depending on the vendor-specific design (characteristics and infrastructure of the host operating system).

Its hypervisor-based switch or bridge counterpart (*) of the modern compute environment has many of the optimizations (and some of the limitations) of hardware forwarding models.

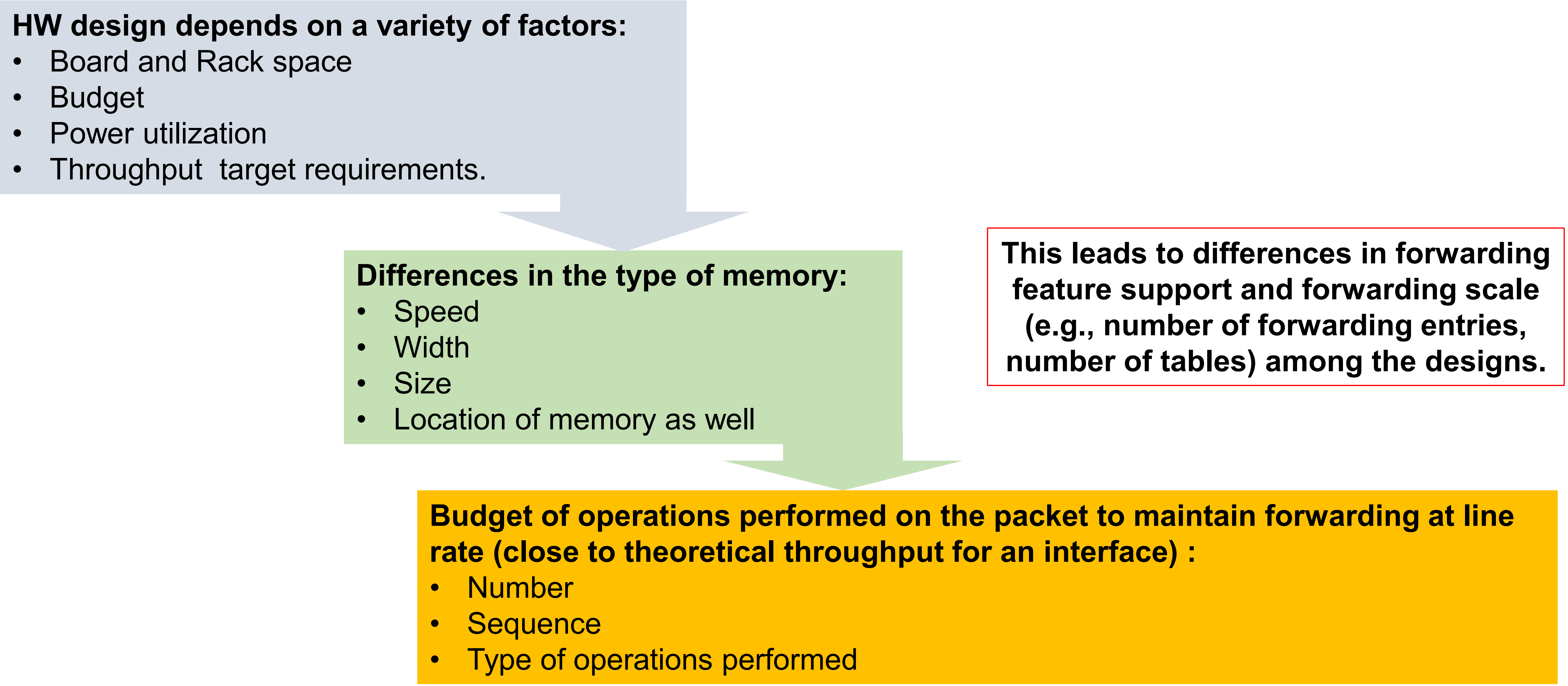
Historically, lookups in hardware tables have proven to result in much higher packet forwarding performance and therefore have dominated network element designs (**), particularly for higher bandwidth network elements. However, recent advances in the I/O processing of generic processors (***), spurred on by the growth and innovation in cloud computing, are giving purpose-built designs, particularly in the mid-to-low performance ranges, quite a run for the money.

(*) See SRIOV and OVS.

(**) A good example is a SBC (Session Border Controller, an IMS component which includes a P-CSCF and a media plane for real-time session.

(***) COTS

DATA PLANE – HARDWARE FORWARDING



HW design depends on a variety of factors:

- Board and Rack space
- Budget
- Power utilization
- Throughput target requirements.

Differences in the type of memory:

- Speed
- Width
- Size
- Location of memory as well

Budget of operations performed on the packet to maintain forwarding at line rate (close to theoretical throughput for an interface) :

- Number
- Sequence
- Type of operations performed

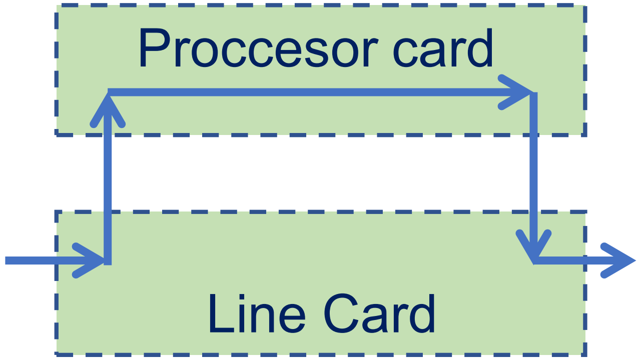
This leads to differences in forwarding feature support and forwarding scale (e.g., number of forwarding entries, number of tables) among the designs.

DATA PLANE – FORWARDING

Actions resulting from the data plane forwarding lookup

- *Forward*
- *Replicate* (in cases such as multicast
- *Drop*
- *Re-mark*
- *Count*
- *Queue (huge impact in time sensitive networks because of latency)*

Some of these actions may be combined or chained together.



In some cases, the forward decision returns a local port, indicating the traffic is destined for a locally running process such as OSPF or BGP6.

These datagrams leave the hardware-forwarding path and are forwarded to the route processor using an internal communications channel.

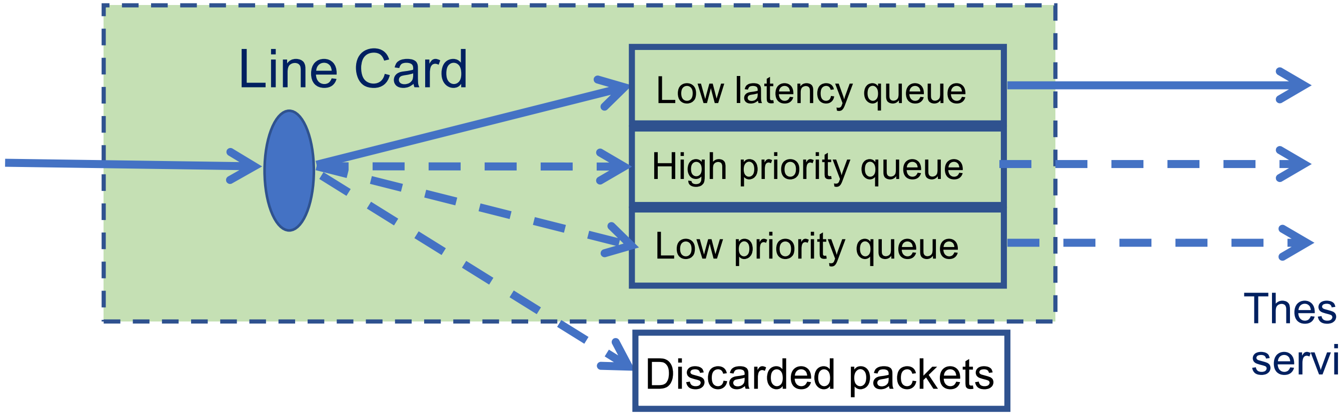
This path is generally a relatively low-throughput path, as it is not designed for high-throughput packet forwarding of normal traffic; however, some designs simply add an additional path to the internal switching fabric for this purpose, which can result in near-line rate forwarding within the box.

DATA PLANE – FORWARDING

The data plane can also implement some small services/ features commonly referred to as forwarding features. In some systems, these features use their own discrete tables, while others perform as extensions to the forwarding tables (increasing entry width).
Including these features, system can (to a small degree) locally alter or preempt the outcome of the forwarding lookup. For example:

- An access control list entry may specify a drop action for a specific matching flow (note that in the ACL, a wider set of parameters may be involved in the forwarding decision). In its absence, there may have been a legitimate forwarding entry and thus the packet would NOT be dropped.

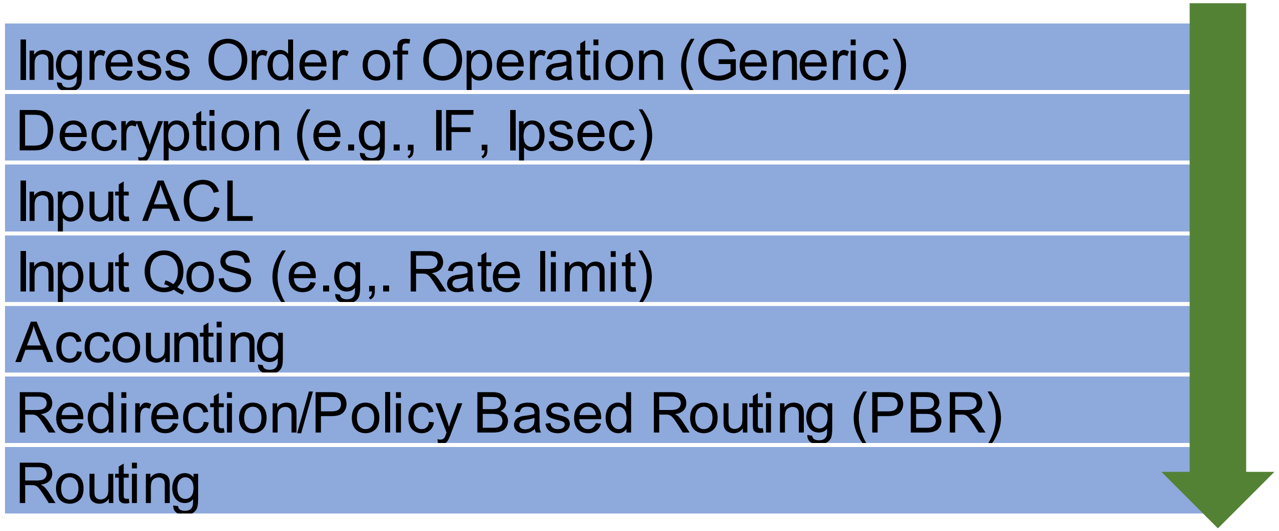
- A QOS policy can ultimately map a flow to a queue on egress or remark its TOS/COS to normalize service with policies across the network. And, like the ACL, it may mark the packet to be dropped (shaped) regardless of the existing forwarding entry for the destination/flow.



In some cases DSCP field is marked and used for forwarding purposes

These forwarding features overlap the definition of services. Arguably, a data plane and control plane component of these services exists

DATA PLANE – FORWARDING



Ingress Order of Operation (Generic)

Decryption (e.g., IF, Ipsec)

Input ACL

Input QoS (e.g,. Rate limit)

Accounting

Redirection/Policy Based Routing (PBR)

Routing

COMMUNICATION BETWEEN PLANES

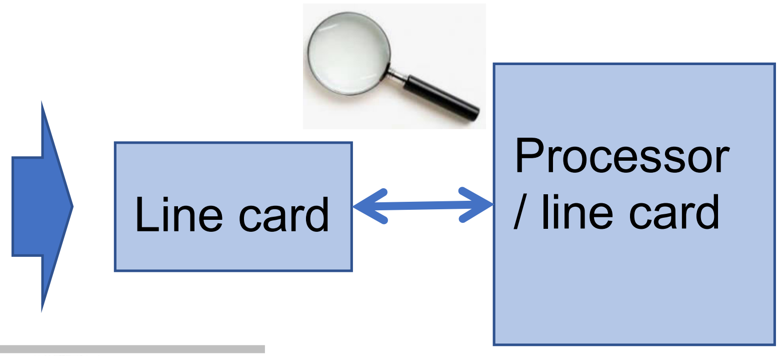
The internal function of larger, multislot/multicard (chassis-based) distributed forwarding systems of today mimic some of the behaviors of the logically centralized but physically distributed control mechanisms of SDN.

Particularly those aspects of the distribution of tables and their instantiation in hardware are of interest. An examination of the inner workings of a typical distributed switch reveals a number of functions and behaviors that mimic those of an externalized control plane.

For example, in systems where the control plane resides on an independent processor/line card and data planes exist on other, independent line cards, certain behaviors around the communication between these elements must exist for the system to be resilient and fault tolerant.

Are all of these behaviors needed if the control plane is removed from the chassis and relocated further away (i.e., logically or strictly centralized)?

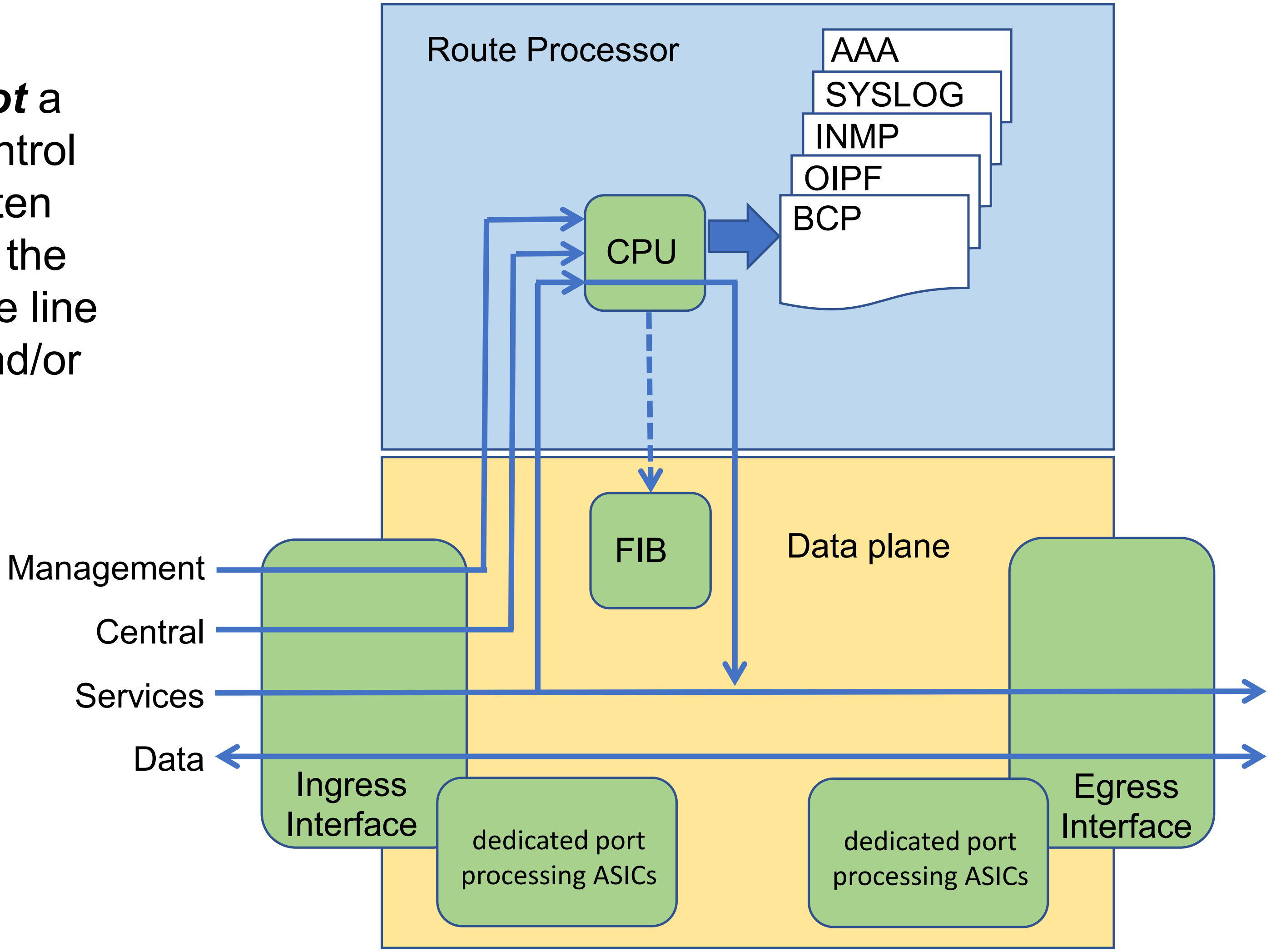
Similar in some aspects
to those of an
externalized control plane



SEPARATION IS IMPORTANT

The separation of the control and data planes is ***not*** a new concept. any multislot router/switch has its control plane executing on a dedicated processor/card (often two for redundancy) and the switching functions of the data plane executing independently on one or more line cards, each of which has a dedicated processor and/or packet processor.

Under normal operation, the ports have forwarding tables that dictate how they process inbound-to-outbound interface switching. These tables are populated and managed by the route processor's CPU/control plane program or programs. When control plane messages or unknown packets are received on these interfaces, they are generally pushed up to the route processor for further processing. Think of the route processor and line cards as being connected over a small but high-speed, internal network because in reality this is in fact how modern switches are built.



INMPp: internetwork Management Program
OIPF:Open IPTV Forum
Syslog is a de facto standard for sending log messages over an IP computer network.

SEPARATION IS IMPORTANT

Some issues derived from coupling planes interdependencies

SCALE

The service cards are limited to a certain amount of subscriber/flow/service state that they can support for a particular generation of the card. Unfortunatelly there is a significant lag between the availability of a new family of processors (or new processors within the family currently employed on the card) and a new service card that takes advantage of that innovation. It takes considerable time to do additional custom design. This unfortunately leads to added system cost.

Forwarding cards could support a certain scale of forwarding entries. Some of these cards have separate, local slave or peer processors to the control processor on the control board, and these in turn have local processing limitations of their own which can involve CPU processing budget.

The control card memories can handle a certain route scale or other state and have processing limitations based on the generation of the CPU complex on the card, but this memory is also used to store control protocol state and management such as BFD or SNMP. Another fundamental limitation to these designs is that this memory is, generally speaking, the fastest money can buy and thus the most expensive.

SEPARATION IS IMPORTANT

Some issues derived from coupling planes interdependencies

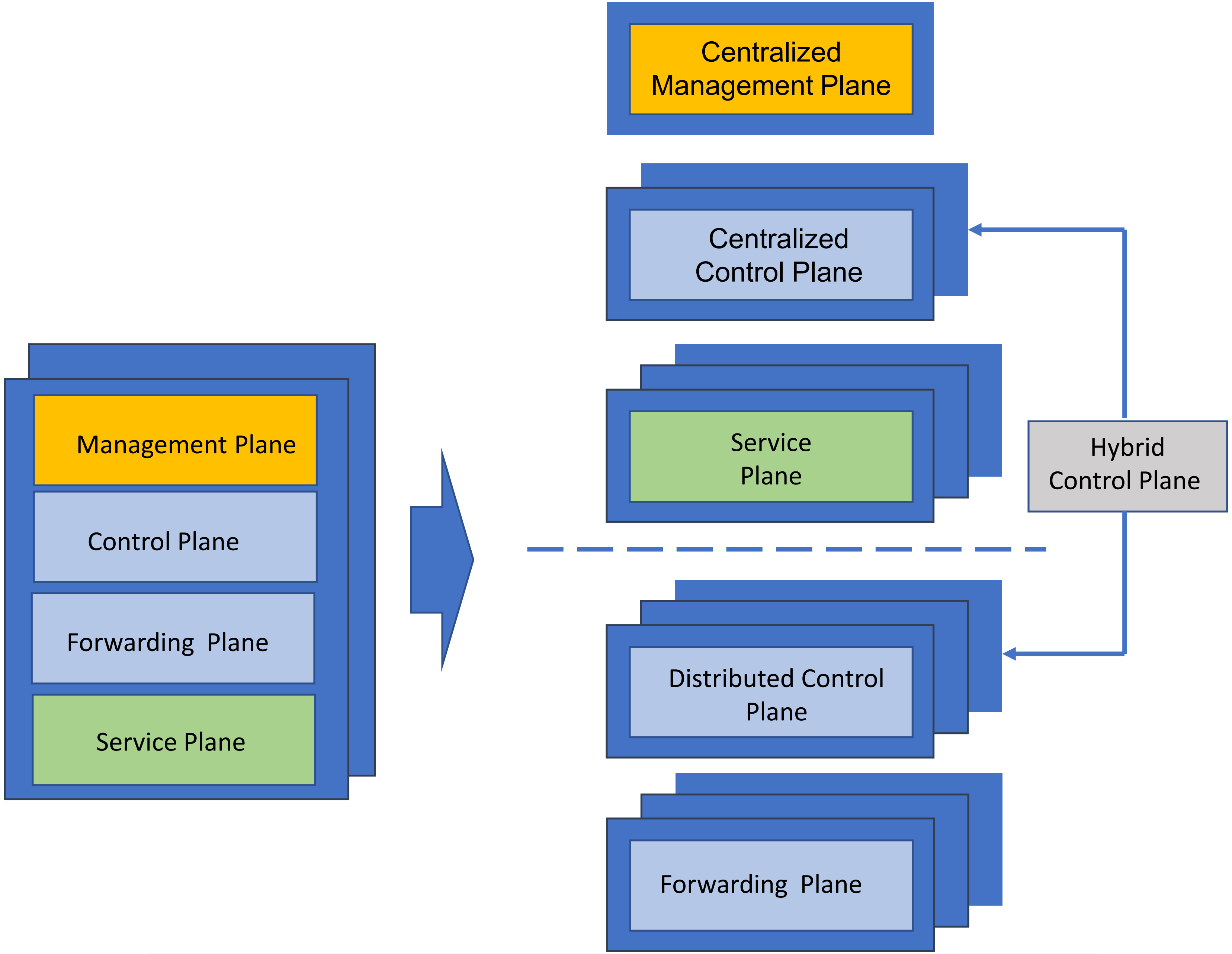
EVOLUTION

Data plane and control plane can evolve and scale independently from each other

So in the past, the network operator had to follow a hardware upgrade path to solve the scale or processing related problems of the control plane. While doing this, the operator had to keep an eye on the forwarding card scale as well as the price-to-performance numbers to pick just the right time to participate in an upgrade. Though it is more pertinent to the separation of the control plane discussion, in the highly specialized platform solutions, they might have to balance the ratio of service cards to forwarding cards, which could significantly reduce the overall forwarding potential of the device (giving up forwarding slots for service slots). One way equipment vendors tried to help this situation was by separating the control and data planes apart so that they could evolve and scale independently—or at least much better than if they were combined.

The SDN-driven twist on the typical equipment evolution is that while there may still exist a cycle of growth/scale and upgrade in the control (and service) plane to accommodate scale, this is much easier to pursue in a COTS compute environment. This is particularly true given the innovations in this environment being driven by cloud computing.
Further, dissecting the control plane from the management processes further provides some level of scale impact isolation by running those user-level processes on COTS hardware within the router/switch, or even remotely

Separation of data plane and control plane, allows for an independent scale and evolution


Centralized Management Plane
Centralized Control Plane
Management Plane
Control Plane
Forwarding Plane
Service Plane
Service Plane
Hybrid Control Plane
Distributed Control Plane
Forwarding Plane

SEPARATION IS IMPORTANT

Some issues derived from coupling planes interdependencies

COSTS

Cost is driven by its companions: scale (a CAPEX driver), complexity, and stability (OPEX drivers).

For many customers (particularly service providers or large enterprises with data center operations), the cost of processing power is very cheap on generic compute (COTS) in comparison to the cost of processing in their network elements. The integration costs associated with the integrated service and control cards drive some of this cost differential. Admittedly, some of this cost differential is also driven by a margin expectation of the vendor for the operating system (those control, management, and service processes), which are not always licensed separately.
It’s a way to recover their investment in their intellectual property and fund ongoing maintenance and development.

Integration can add new costs

INNOVATION

Theoretically, separation can benefit the consumer by changing the software release model in a way that enables innovations in either plane to proceed independently from each other (as compared to the current model in which innovations in either plane are gated by the build cycle of the multipurpose integrated monolith).
More relevant to the control/data separation would be the ability to support the introduction of new hardware in the forwarding plane without having to iterate the control plane (for example, the physical handling of the device would be innovation in the data plane component via new drivers).

SEPARATION IS IMPORTANT

Some issues derived from coupling planes interdependencies

STABILITY

By separating the control and data planes, the forwarding elements may become more stable by virtue of having a smaller and less volatile codebase. The premise that a smaller codebase is generally more stable is fairly common these days. For example, a related (and popular) SDN benefit claim comes from the clean slate proposition, which posits that the gradual development of features (not recommended) in areas like Multiprotocol Label Switching (MPLS) followed a meandering path of feature upgrades that naturally bloats the code bases of existing implementations.
This bloat leads to implementations that are overly complex and ultimately fragile.

The claim is that the implementation of the same functionality using centralized label distribution to emulate the functionality of the distributed LDP or RSVP and a centralized knowledge of network topology could be done with a codebase at least an order of magnitude smaller than currently available commercial codebases. The natural claim is that in a highly prescriptive and centralized control system, the network behavior can approach that of completely static forwarding, which is arguably stable.

Separation means smaller codebase, which in turn means more stability

SEPARATION IS IMPORTANT

Some issues derived from coupling planes interdependencies

COMPLEXITY AND ITS RESULTING FRAGILITY

The question of how many control planes and where these control planes are located directly impacts the scale, performance, and resiliency—or lack thereof, which we refer to as fragility—of a network.

Specifically, network operators plan on deploying enough devices within a network to handle some percentage of peak Demand. When the utilization approaches this, new devices must be deployed to satisfy the demand. In traditional routing and switching systems, it's important to understand how much localized forwarding throughput demand can be satisfied without increasing the number of managed devices and their resulting control protocol entities in the network.

The general paradigm of switch and router design is to use a firmly distributed control plane model, and that generally means that for each device deployed, a control plane instance will be brought up to control the data plane within that chassis.

The question then is this: how does this additional control plane impact the scale of the overall network control plane for such things as network convergence (i.e., the time it takes for the entirety of running control planes to achieve and agreed upon a loop-free state of the network)? The answer is that it does impact the resiliency and performance of the overall system, and the greater the number of control planes, the potential at least exists for additional fragility in the system. It does also increase the anti-fragility of the system if tuned properly, however, in that it creates a system that eventually becomes consistent regardless of the conditions. Simply put, the number of protocol speakers in distributed or eventual consistency control models can create management and operations complexity.

SEPARATION IS IMPORTANT

Some issues derived from coupling planes interdependencies

COMPLEXITY AND ITS RESULTING FRAGILITY

IDEA: TO REDUCE THE NUMBER OF CONTROL PLANE INSTANCES

Initially, an effort to curtail the growth of control planes was addressed by creating small clusters of systems from stand-alone elements. Each element of the cluster was bonded by a common inter-chassis data and control fabric that was commonly implemented as a small, dedicated switched Ethernet network.

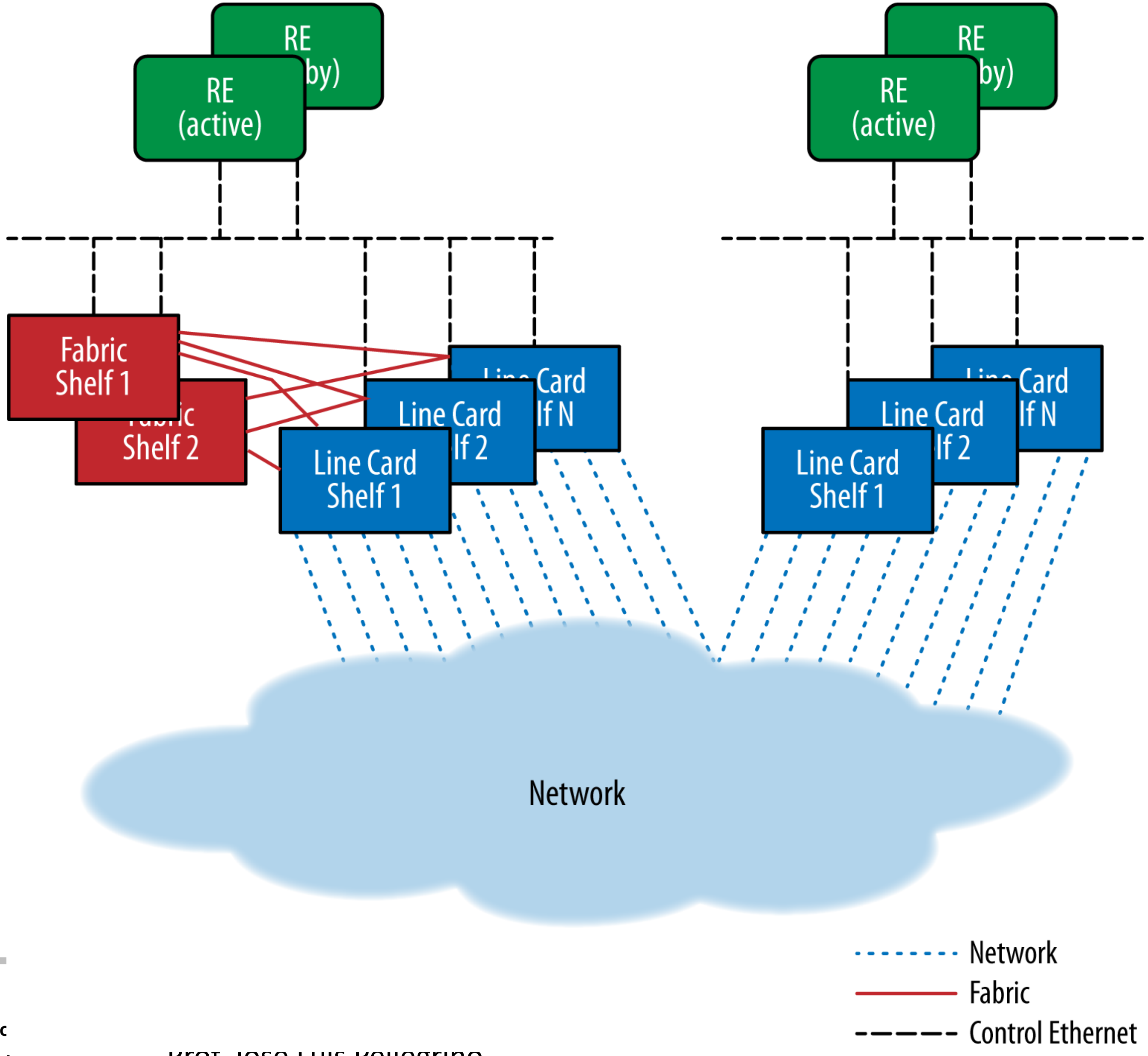
The multichassis system took this concept a step further by providing an interconnecting fabric between the shelves and thus behaved as a single logical system, controlled by a single control plane. Connectivity between the shelves was, however, implemented through external (network) ports, and the centralized control plane uses multiple virtual control plane instances—one per shelf. It was also managed as such in that it revealed a single IP address to the network operator, giving them one logical entity to manage.

SEPARATION IS IMPORTANT

Some issues derived from coupling planes interdependencies

COMPLEXITY AND ITS RESULTING FRAGILITY

IDEA: TO REDUCE THE NUMBER OF CONTROL PLANE INSTANCES

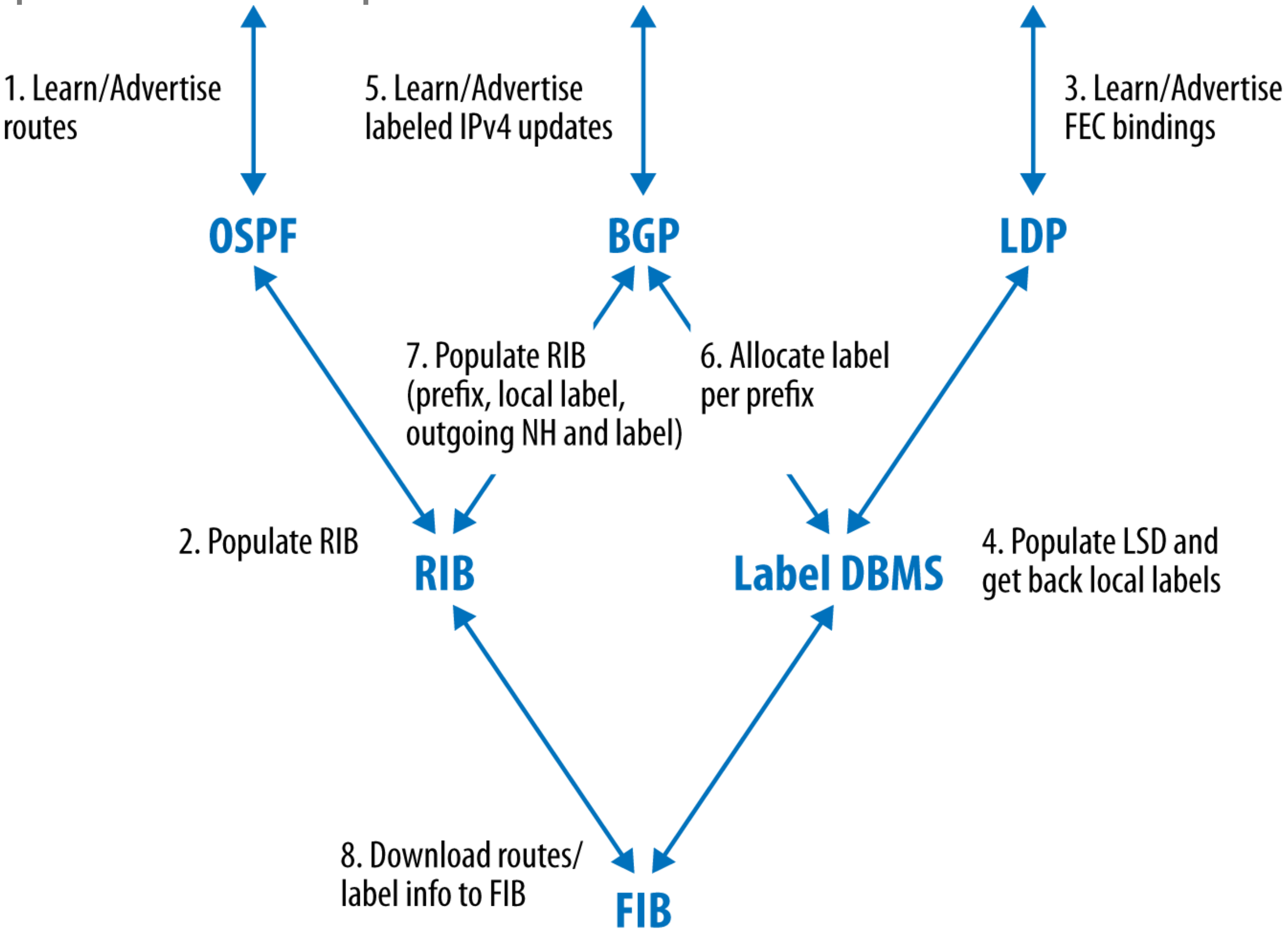


SEPARATION IS IMPORTANT

Some issues derived from coupling planes interdependencies

You should note that this latter view of multichassis or cluster systems approaches some of the characteristics of SDN (centralization and more independent scaling of the control plane), albeit without solving the programmability/flexibility problems of the control plane.

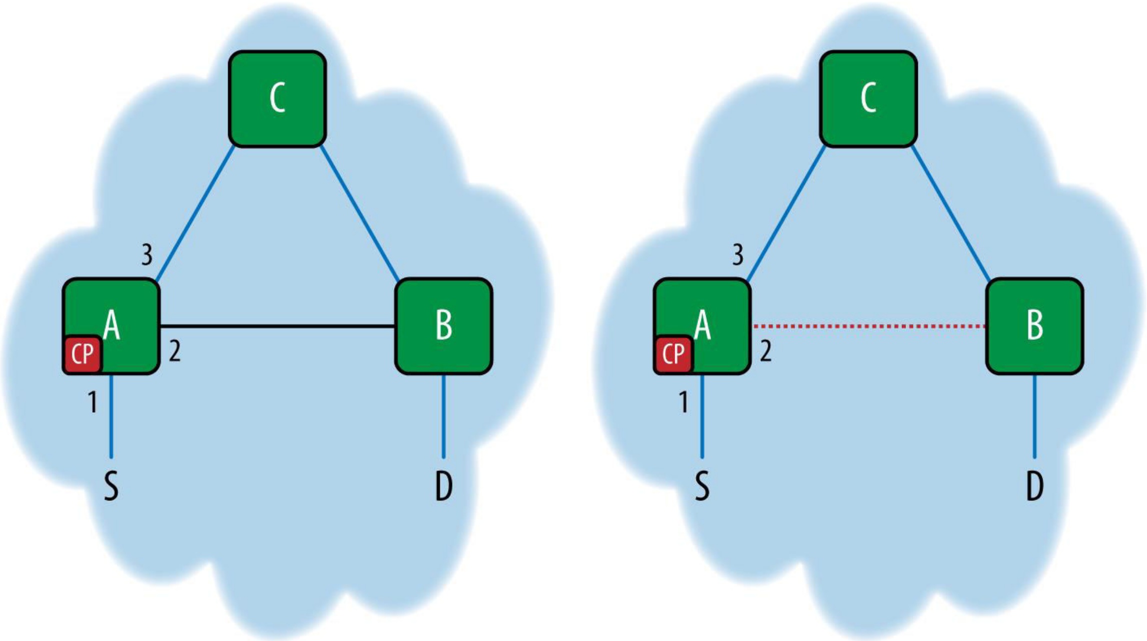
There is also the potential to reduce the number and interaction of protocols required to create forwarding state in the elements. Figure shows the process interaction in an IGP/BGP/MPLS network to learn/advertise prefixes and label bindings to populate forwarding in the data plane.


1. Learn/Advertise routes
5. Learn/Advertise labeled IPv4 updates
3. Learn/Advertise FEC bindings
OSPF
BGP
LDP
7. Populate RIB (prefix, local label, outgoing NH and label)
6. Allocate label per prefix
2. Populate RIB
RIB
Label DBMS
4. Populate LSD and get back local labels
8. Download routes/ label info to FIB
FIB

DBMS: Database Management System

DISTRIBUTED CONTROL PLANES

The control paradigm that has evolved with the Internet, which is our ultimate network scale problem to date, is a distributed, eventual consensus model. In this model, the individual elements or their proxies participate together to distribute reachability information in order to develop a localized view of a consistent, loop-free network. We “label” the model as one of eventual consensus because of the propagation delays of reachability updates, inherent in the distributed control plane model in anything beyond a small home network, forms a fairly complex network graph. By design, the model is of intermittent nonsynchronization that could lead to less optimal forwarding paths but (hopefully) avoiding or limiting transient cycles otherwise known as micro-loops in the overall path.



IP and MPLS forwarding are examples of a distributed control model. In these forwarding paradigms, routes and reachability information is exchanged that later results in data plane paths being programmed to realize those paths

CREATING THE IP UNDERLAY

The Network Layer Reachability Information (**NLRI**) is exchanged between **BGP** routers using UPDATE messages.

INTRODUCTION TO SDN

NETWORKING REVIEW PART A

Although NFV and SDN, have being adopted during last years, some “classical/underlay” technologies must be reviewed prior to move forward to the next step

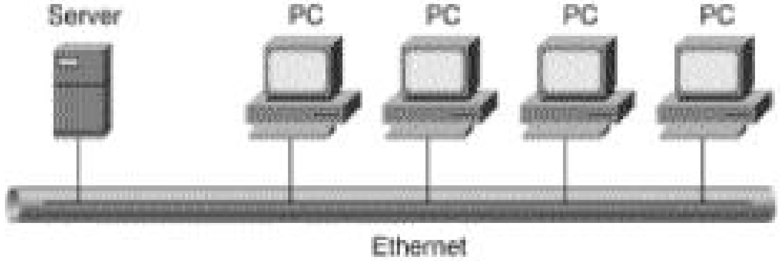
CePETel
Sindicato de los Profesionales
de las Telecomunicaciones

SECRETARÍA TÉCNICA IPEI

ETHERNET. INTRODUCTION

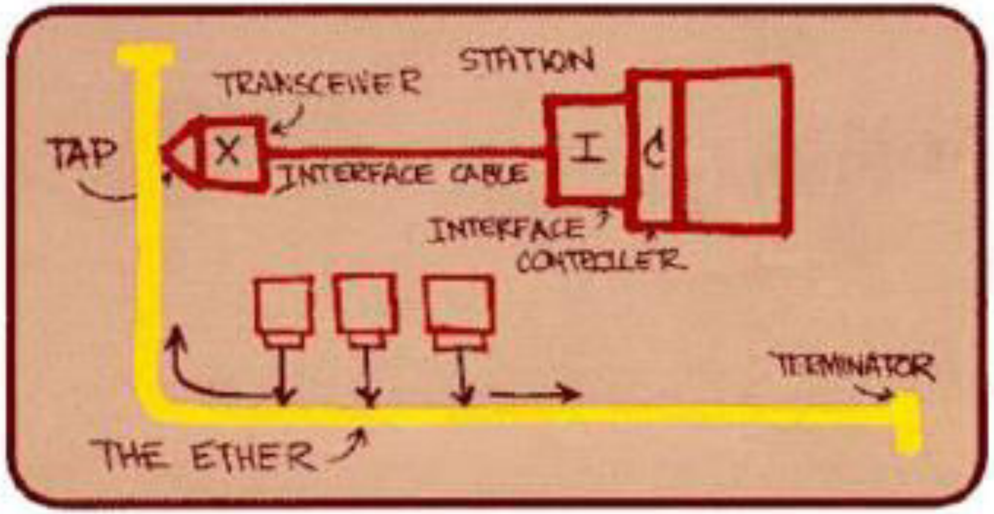
1973

Robert M. Metcalfe and David R. Boggs implemented the Alto Aloha Network at Xerox PARC



1976

The name Ethernet was first used



This diagram was hand drawn by Robert M. Metcalfe and photographed by Dave R. Boggs in 1976 to produce a 35mm slide used to present Ethernet to the National Computer Conference in June of that year. On the drawing are the original terms for describing Ethernet.

ETHERNET. INTRODUCTION

Ethernet has been around for over 30 years and has no serious competitors in sight, so it is likely to exist for a some more years. Few CPU architectures, operating systems, or Programming languages have been on the crest of the wave for more than two decades. Clearly, Ethernet did something right, but what was it? Probably the main reason for its longevity is that Ethernet is simple and flexible. In practice, simple translates to reliable and easy to maintain. Simple also translates as cheap. Thin Ethernet cabling and twisted pair cabling have a relatively low cost. Interface cards are also low cost. Only when hubs and switches were introduced, considerable investment was required, but by the time they came on the scene, Ethernet was already well established. Ethernet is easy to maintain. There is no software to install (just the drivers) and no configuration tables to manage (to mess with). Plus, adding new hosts is as simple as connecting them. Another point is that Ethernet easily interfaces with TCP/IP, which has become dominant. IP is a connectionless protocol, which fits perfectly with Ethernet, which is also not connection-oriented. IP didn't fit as well with ATM, which is connection-oriented. Finally, Ethernet has been able to evolve in important ways. Speeds have increased by a few magnitude levels and hubs and switches have been introduced, but these changes do not require software modifications. When FDDI, Fiber Channel, and ATM were introduced, they were faster than Ethernet, but they were also incompatible with Ethernet, much more complex, and unwieldy. Over time, Ethernet caught up with them in speed, so they no longer had advantages and gradually fell out of use.

Ethernet is simple. flexible, reliable, easy to maintain, NIC are cheap, easily interfaces with TCP/IP

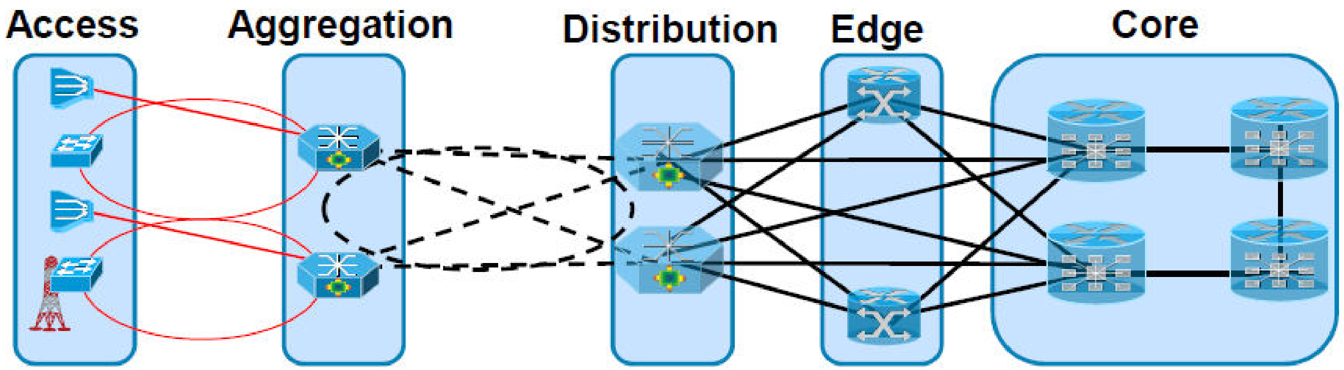
Until the appearance of SDN, the networks remained without major changes

ETHERNET TECHNOLOGY EVOLUTION THROUGH THE YEARS

Ethernet Technology Evolution through the years

▪ 1980’s	10 Mbps Ethernet IEEE 802.3
▪ 1992-1995	100 Mbps Ethernet IEEE 802.3u
▪ 1995-1999	1 Gbps Ethernet IEEE 802.3z
▪ 1998-2000	10/100/1000 Mbps Ethernet link Aggregation IEEE 802.3ad
▪ 1999-2002	10 Gbps IEEE 802.3 ae
▪ 2010	100 Gbps IEEE 802.3ba
▪ 2017	400 Gbps Ethernet over optical fiber using multiple lines at 25G/50G. (802.3bs)

ETHERNET. NETWORK ARCHITECTURE OF CONVERGENT SP

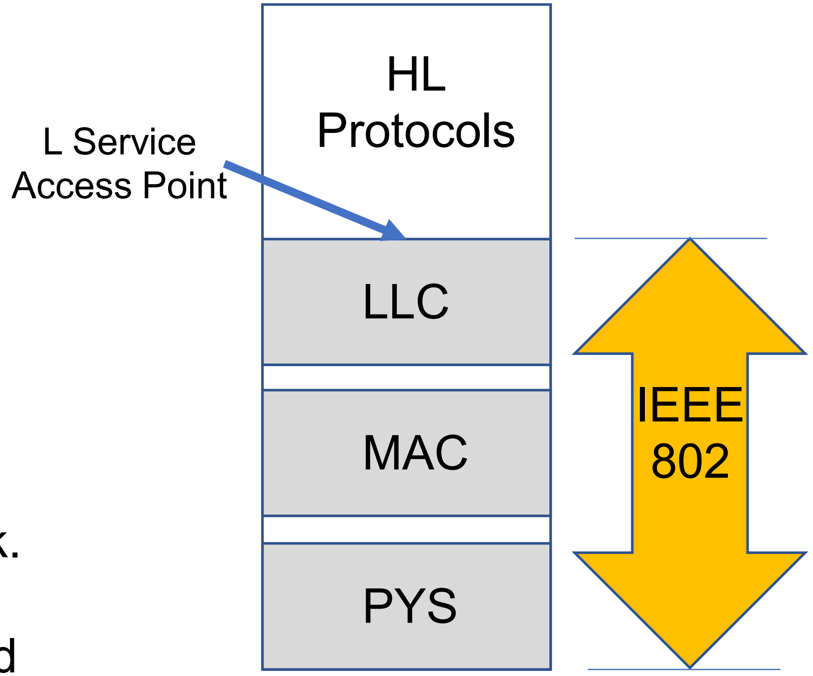


					Services
L2 MPLS Aggregation Service Model	802.1ad (Q-in-Q) Ethernet MST /LAG/ REP	EoMPLS	802.1ad	IP or IP-VPN	Residential: BTV/VoD, VoIP, HSI
		EoMPLS			Ethernet VPN: E-Line
		EoMPLS	H-VPLS	VPLS	Ethernet VPN: E-LAN/-Tree
		EoMPLS	802.1ad	IP or IP-VPN	IP and IP-VPN
L2+L3 MPLS Aggregation Service Model		IP or IP-VPN			Residential: BTV/VoD, VoIP, HSI
		EoMPLS			Ethernet VPN: E-Line
		EoMPLS	H-VPLS	VPLS	Ethernet VPN: E-LAN/-Tree
		IP or IP-VPN			IP and IP-VPN

PROTOCOL ARCHITECTURE IN LAN NETWORKS

LAN network protocol architecture

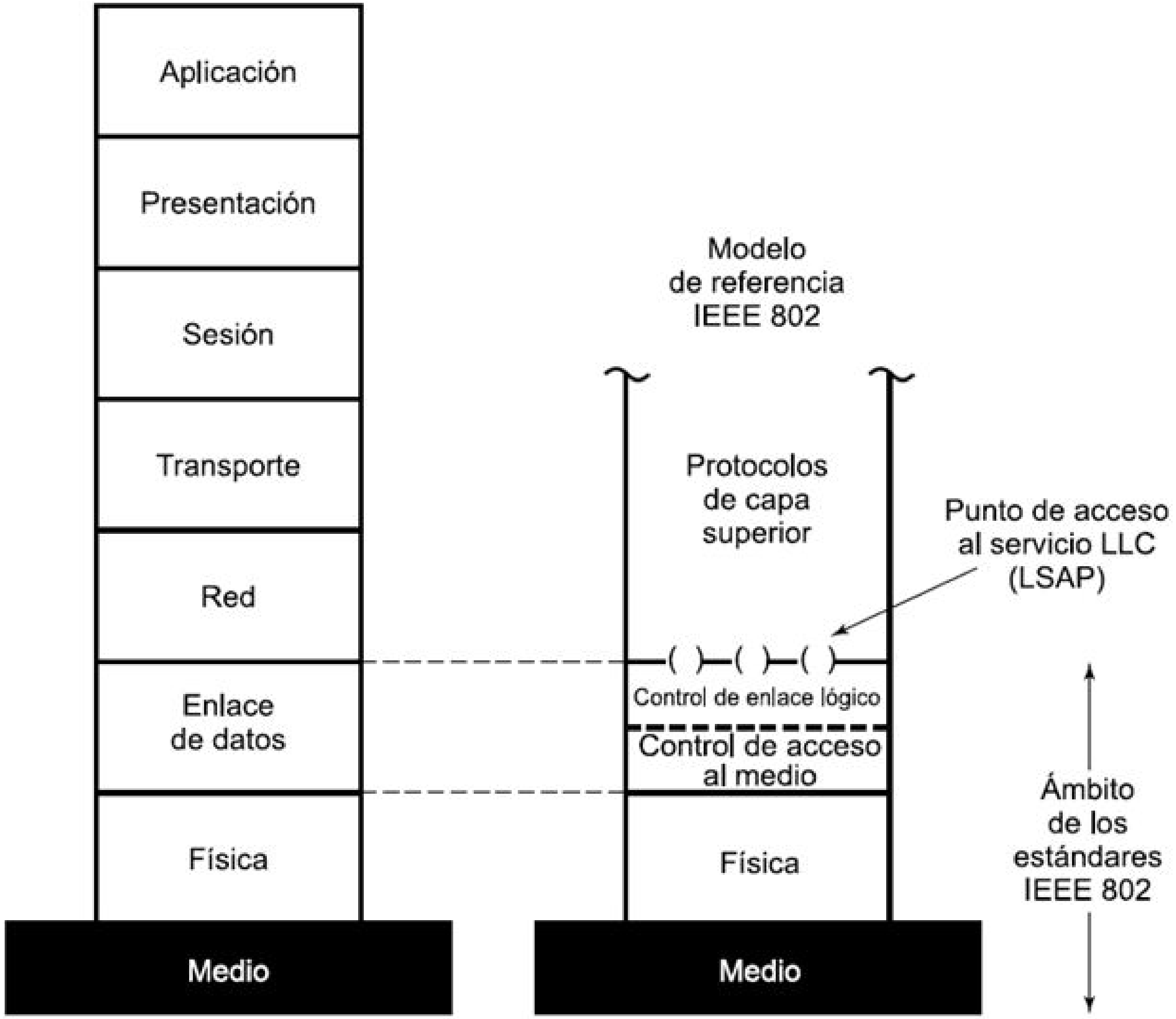
The architecture of a LAN is best described in terms of a hierarchy of protocols that organize the basic functions of the LAN. This section begins with a description of the standardized protocol architecture for LANs, which includes the physical, medium access control (MAC), and logical link control (LLC) layers.

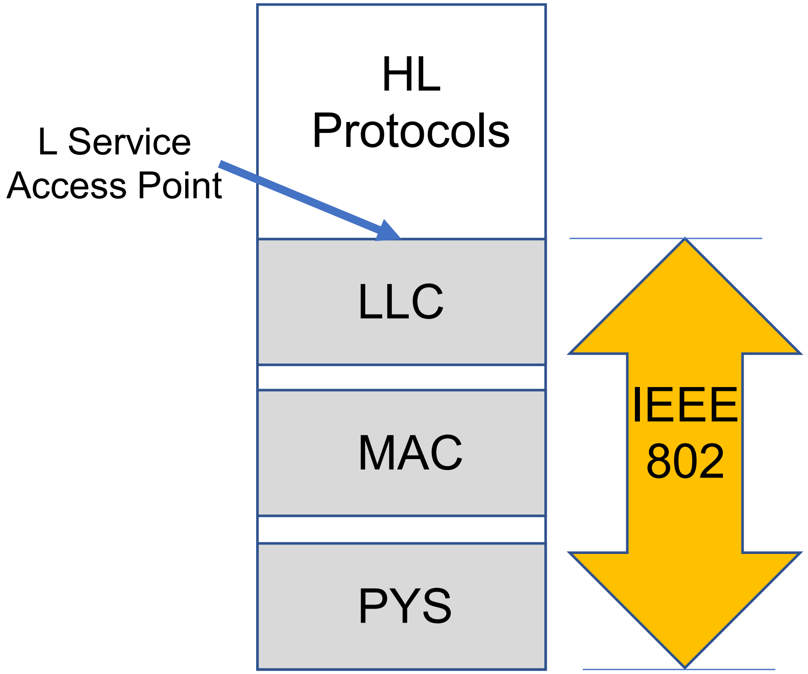


IEEE 802 Reference Model

Protocols defined specifically for transmission over LAN and MAN networks address issues related to the transmission of blocks of data over the network. According to OSI, higher layer protocols (layer 3 or 4 and higher) are independent of the network architecture and are applicable to LAN, MAN and WAN networks. Thus, the study of LAN protocols is related to the lower layers of the OSI model. The following figure relates the LAN protocols to those of the OSI architecture. This architecture was developed by the IEEE 802 committee and has been adopted by all organizations working on the specification of LAN standards; It is referred to as the IEEE 802 reference model.

PROTOCOL ARCHITECTURE IN LAN NETWORKS


Aplicación
Presentación
Sesión
Transporte
Red
Enlace de datos
Física
Medio
Modelo de referencia IEEE 802
Protocolos de capa superior
Punto de acceso al servicio LLC (LSAP)
Control de enlace lógico
Control de acceso al medio
Física
Medio
Ámbito de los estándares IEEE 802

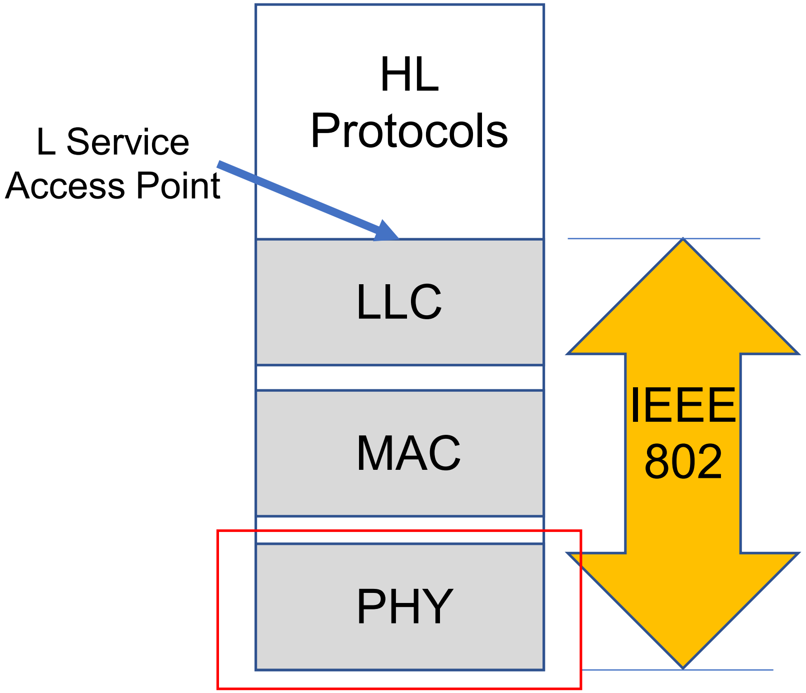


PROTOCOL ARCHITECTURE IN LAN NETWORKS- physical layer

Physical layer
From the bottom up, the bottom layer of the IEEE 802 reference model is the physical layer of the OSI model, and includes features such as:

- Encoding/decoding of signals (Manchester, others.
- Preamble generation/removal (for synchronization).
- Transmission/reception of bits.

There are several alternatives for the transmission medium that can be used in a LAN:



The physical layer for each transmission rate is divided into sublayers that are independent of the particular media type and sublayers that are specific to the media type or to the encoding of the signal.

PROTOCOL ARCHITECTURE IN LAN NETWORKS- physical layer

- The reconciliation sublayer and the optional medium-independent interface (MII on 10 Mbps and 100 Mbps Ethernet, GMII on Gigabit Ethernet) provide the logical connection between the MAC and the different sets of medium-dependent layers. The MII and GMII are defined with separate data transmission and reception paths that are bit-serial for 10 Mbps implementations, nibble-serial (4 bits wide) for 100 Mbps implementations, and byte-serial (8 bits wide) for 1000 Mbps implementations. The Media Independent Interfaces and Reconciliation Sublayer are common for their respective transmission rates and are configured for full duplex operation in 10Base-T and all Ethernet versions.

- The medium-dependent physical coding sublayer (PCS) provides the logic for coding, multiplexing, and synchronization of the output symbol streams, as well as symbol code alignment, demultiplexing, and decoding of the incoming data.

- The physical medium coupling (PMA) sublayer contains the signal transmitters and receivers (transceivers), as well as the clock recovery logic for the received data streams.

- The medium dependent interface (MDI) is the cable connector between the signal transceivers and the link

- The Auto-Negotiation sublayer allows the NICs at each end of the link to exchange information about their individual capabilities, and then negotiate and select the most favorable operating mode that they are both capable of supporting. Auto-negotiation is optional in early Ethernet implementations and is required in later versions.

PROTOCOL ARCHITECTURE IN LAN NETWORKS- physical layer

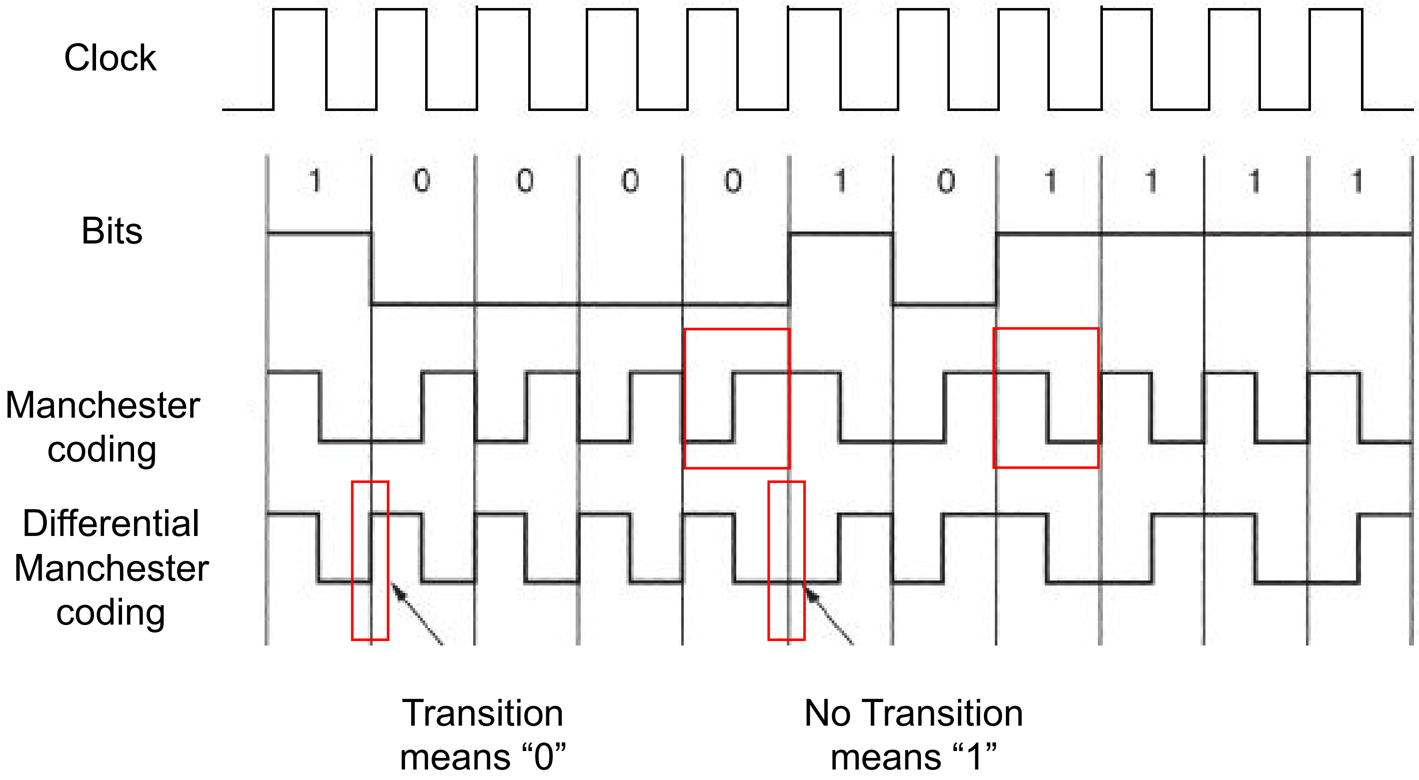
Timing and coding used in LAN networks

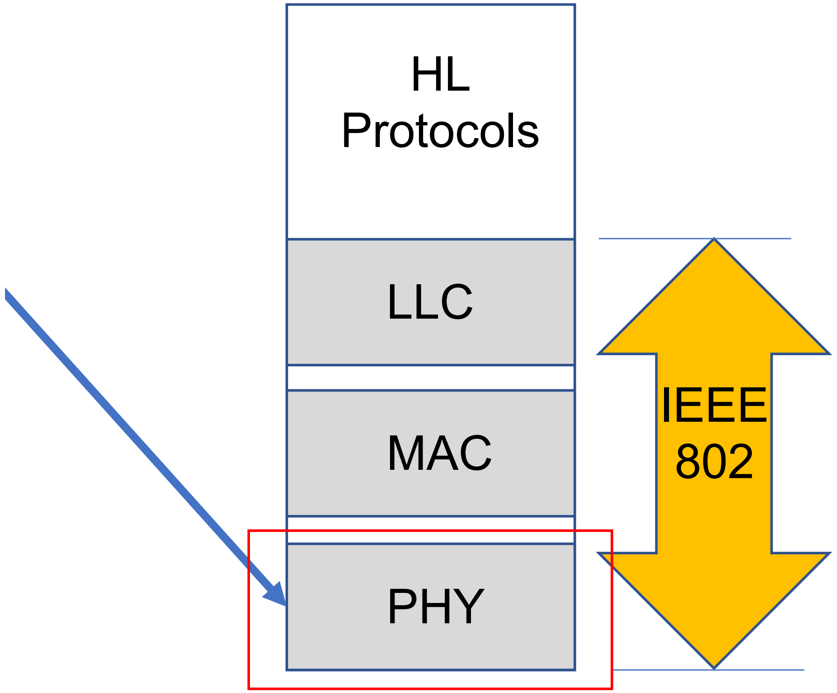
Regarding timing, different clock speeds can cause the receiver and sender to get out of sync with where the bit boundaries are, especially after a long series of consecutive 0s or a long series of consecutive 1s.What is needed is a mechanism for recipients to unambiguously determine the beginning, end or middle of each bit without reference to an external clock. Two such approaches are called Manchester coding and differential Manchester coding.

In Manchester coding, each bit period is divided into two equal intervals. A binary 1 bit is sent having a high voltage level during the first interval and a low voltage level during the second. A binary 0 is just the opposite: first low and then high. This scheme ensures that each bit period has a halfway transition, making it easier for the receiver to synchronize with the sender. A disadvantage of Manchester coding is that it requires twice as much bandwidth as direct binary coding, since the pulses are half as wide. For example, to send data at 10 Mbps, the signal has to change 20 million times/sec.

Differential Manchester coding is a variation of basic Manchester coding. Here, a bit 1 is indicated by the absence of a transition at the start of the interval. A bit 0 is indicated by the presence of a transition at the start of the interval. In both cases there is also a transition in the middle. The differential scheme requires more complex equipment, but offers better noise immunity. All Ethernet systems use Manchester coding due to its simplicity. The high signal is + 0.85 volts, and the low signal is − 0.85 volts, giving a DC value of 0 volts. Ethernet does not use differential Manchester coding, but other LANs (such as 802.5 token ring) do.

PROTOCOL ARCHITECTURE IN LAN NETWORKS- physical layer





PROTOCOL ARCHITECTURE IN LAN NETWORKS- Data link layer

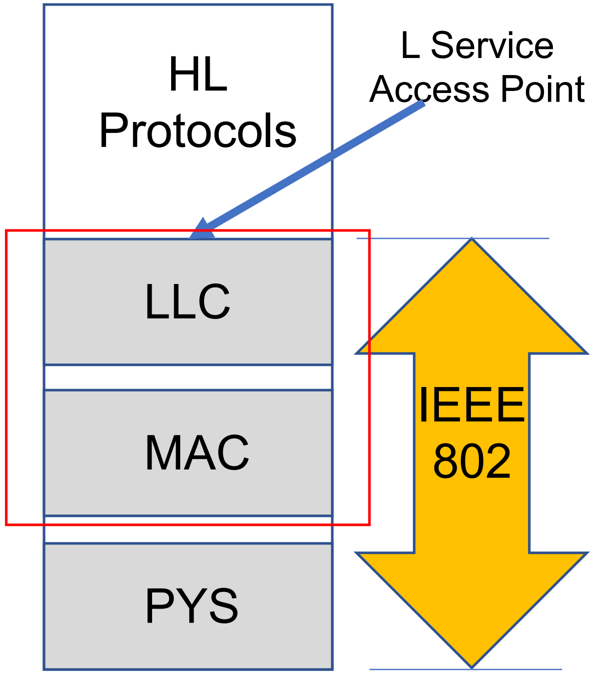
Data Link Layer (layer 2)
Above the physical layer are the functions associated with the services offered to LAN users. Among them we can describe the following:

MAC

- In transmission, assembly of data in frames with address and error detection fields.
- On reception, frame disassembly, address recognition and error detection
- Access control to the LAN transmission medium.

LLC

- Interface with higher layers and error and flow control.



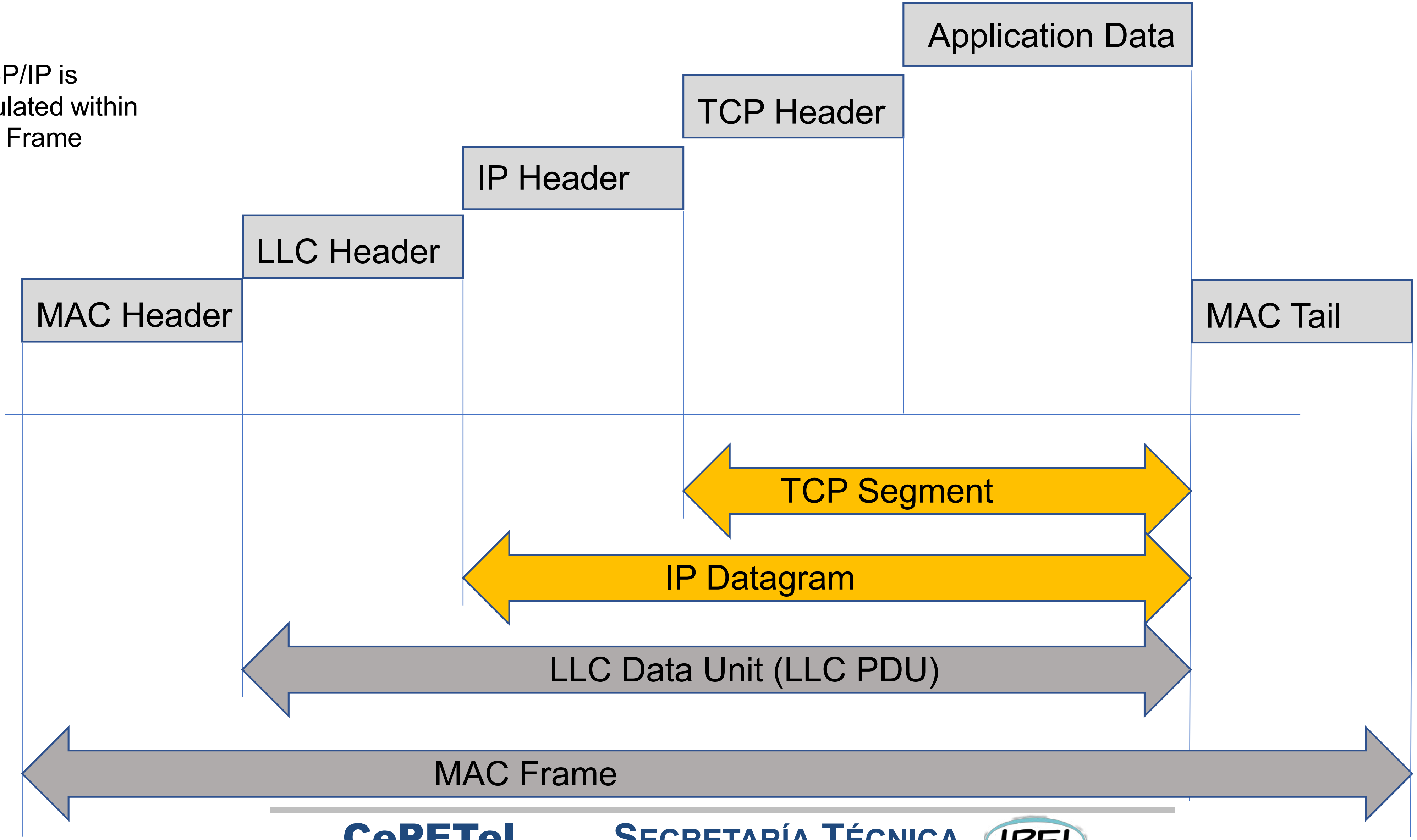
These functions are generally associated with OSI Layer 2. The set of functions in the last of the four points are grouped in the logical link control (LLC) layer, while the functions specified in the first three points are treated in a separate layer called medium access control (MAC).

This separation of functions is due to the following reasons:

- The logic necessary to manage access to a shared medium is not found in layer 2 of traditional data link control.
- Multiple MAC options can be offered for the same LLC.

PROTOCOL ARCHITECTURE IN LAN NETWORKS

TCP/IP is encapsulated within the Frame



CePETel

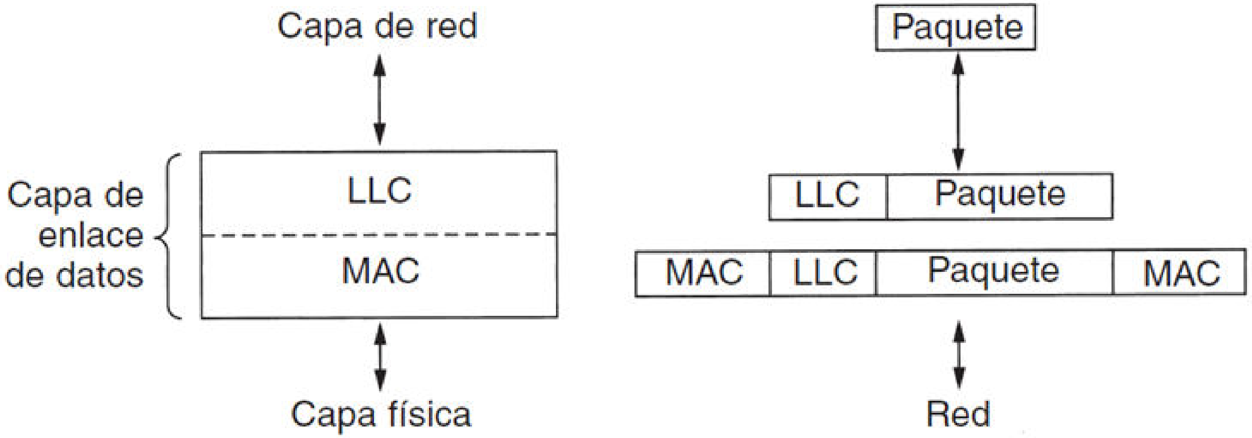
Sindicato de los Profesionales

de las Telecomunicaciones

SECRETARÍA TÉCNICA IPEI

PROTOCOL ARCHITECTURE IN LAN NETWORKS- LLC PROTOCOL

All Ethernet and 802 protocols offer is a best-effort datagram service
Sometimes best effort service is appropriate. For example, no guarantees are required or expected to transport IP packets. An IP packet can simply be entered into an 802 payload field and sent to its destination; If it is lost, so be it. However, there are also systems where a data link protocol with error control and flow control is desired. The IEEE has defined one that can operate on top of all Ethernet and 802 protocols. Additionally, this protocol, called LLC (Logical Link Control), hides the differences between the different types of 802 networks, providing a single format and interface with the network layer. This format, interface and protocol are closely based on HDLC. The LLC forms the top half of the data link layer, with the MAC sublayer below it, as shown in the figure below.



Note: In radio interfaces, there is also a layer 2 protocol for Link Control called RLC, which also is associate to MAC layer : It is also a matter of shared medium, but the nature of link (radio) is particular and there is in cellular networks an “agent” which has no equivalence in LAN networks.

PROTOCOL ARCHITECTURE IN LAN NETWORKS- LLC PROTOCOL

The typical use of the LLC is as follows. The network layer of the sending machine passes a packet to the LLC using the LLC's access primitives. The LLC sublayer then adds an LLC header containing the sequence and receipt numbers. The resulting structure is then entered into the payload field of a frame 802 and transmitted. In the receiver the reverse process occurs.

The LLC provides three service options: unreliable datagram service, acknowledged datagram service, and reliable connection-oriented service.

For the Internet, best-effort attempts to send IP packets are sufficient, so acknowledgments at the LLC level are not required.

The LLC header contains three fields: a destination access point, a source access point and a control field.

The LLC layer in LANs is similar in many ways to other commonly used link layers. Like all link layers, LLC is concerned with the transmission of a link layer protocol data unit (PDU) between two stations, without requiring a intermediate switching node.

LLC has two features not shared by most other link control protocols:

1. It must support multiple access, a consequence of the shared medium nature of the link (this differs from a multipoint line in that no parent node now exists).
2. The MAC layer offloads you from some link access details.

LLC addressing involves the specification of the source and destination LLC users. Usually ,a user is a higher layer protocol or network management function in the station. In keeping with OSI terminology for the user of a layer of the protocol architecture, these LLC user addresses are called Service Access Points (SAP).

PROTOCOL ARCHITECTURE IN LAN NETWORKS- LLC PROTOCOL

LLC Services
LLC specifies the mechanisms for addressing stations through the medium and for controlling the data exchange between two users.
The operation and format of this standard are based in HDLC. There are three possible services for connected devices that use LLC:

The non-connection-oriented service without confirmation:
This service is of the datagram type. It is very simple, since it does not include flow or error control mechanisms, so reception of the data is not guaranteed. In any case, in most devices there is some upper layer of software responsible for managing reliability issues. It requires minimal logic and is useful in two situations. Firstly, in those in which the software of the upper layers offers the necessary reliability and flow control mechanisms, avoiding duplication. For example, TCP could provide the mechanisms necessary to ensure reliable data reception. Second, there are situations in which the cost of establishing and maintaining the connection is unjustified and even counterproductive (for example, data acquisition activities that involve periodic sampling of data sources, such as sensors and automatic self-test reports. security of equipment or network components). In a monitoring application, occasional data loss may not cause problems as long as the next report arrives soon. Thus, in most cases, connectionless, unacknowledged services are preferable.

PROTOCOL ARCHITECTURE IN LAN NETWORKS- LLC PROTOCOL

The service in connection mode:

This service is similar to that offered by HDLC. A logical connection is established between two users who exchange data, with flow and error control. It can be used on very simple devices, such as terminal controllers, that have little software above this level. In these cases, the service provides flow control and reliability mechanisms, usually implemented in higher layers of the communications software.

The connectionless service confirmed:

it is a mix of the previous two.The datagrams are confirmed, but no previous logical connection is established. With the service in connection mode, the logical link control software must maintain some type of table containing the status of each active connection. It is useful in several situations.

PROTOCOL ARCHITECTURE IN LAN NETWORKS- LLC PROTOCOL

If the user needs to ensure reception, but there are a large number of destinations for the data, the connection-mode service is not practical given the large number of tables required. An example is a process control or automated enterprise where a central device is necessary to communicate with a large number of programmable processors and controllers. Another possible use of this service is the management of alarms or emergency control signals in a factory: given their importance, confirmation is necessary, so that the sender can be sure that the signal was received. On the other hand, given the urgency of the signal, the user may not want to waste time establishing a logical connection as a prior step to sending the data. Generally, a seller offers these services as options that the consumer can choose when he purchases the equipment. Another possibility is for the consumer to purchase a device that presents two or all three services, selecting each of them according to the application.

LLC Protocol

The basic LLC protocol was designed after HDLC and has similar functions and formats to it. The differences between the two protocols can be summarized as follows:

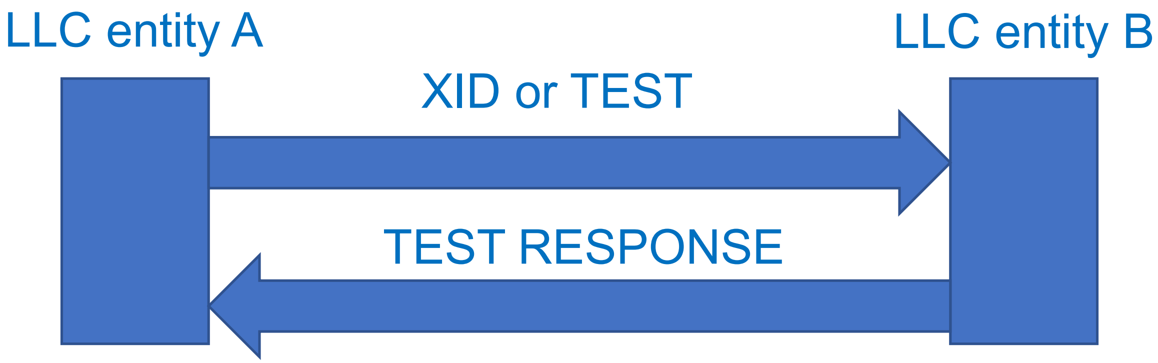
- LLC makes use of HDLC asynchronous balanced mode of operation to support the LLC service in connection mode. This is called type 2 operation, and the other HDLC modes are not used.
- LLC provides a non-connection-oriented service without confirmation using the non-information PDU numbered, which is known as a type 1 operation.
- LLC offers a confirmed connectionless service using two unnumbered PDUs new ones, which is called type 3 operation.
- LLC allows multiplexing through the use of LLC service access points (LSAP).

PROTOCOL ARCHITECTURE IN LAN NETWORKS- LLC PROTOCOL

For type 1 operation (unacknowledged connectionless service), the Unnumbered Information (UI) PDU is used to transmit user data. There is no acknowledgment, flow control, or error control, although there is error detection and rejection at the MAC level.

Two other PDUs are used to support the management functions associated with the three types of operations. Both PDUs are used as follows.

An LLC entity may issue a command (C/R bit=0) XID or TEST, with the receiving LLC entity sending the corresponding XID or TEST in response. The XID PDU is used to exchange two types of information: supported operation types and window size. For its part, the TEST PDU is used to carry out a closed-loop test of the transmission path between two LLC entities. Upon receipt of a TEST Command PDU, the destination LLC entity sends, as soon as possible, a TEST Response PDU.



UA: Unnumbered Acknowledgment
Dm: disconnected Mode
AC: Acknowledged Connectionless

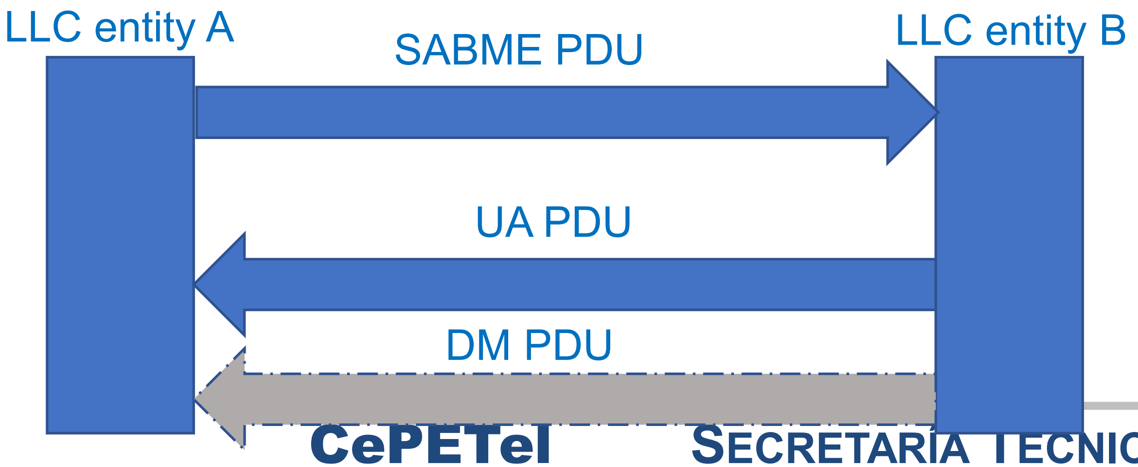
PROTOCOL ARCHITECTURE IN LAN NETWORKS- LLC PROTOCOL

In type 2 operation, a data link connection is established between two SAP LLCs prior to their exchange.

Connection establishment is attempted by the type 2 protocol in response to a request from a user. The LLC entity sends a SABME PDU to request a logical connection to the other LLC entity. If the LLC user specified in the DSAP field accepts the connection, the destination LLC entity returns an Unnumbered Acknowledgment (UA) PDU. The connection is uniquely identified by the user's SAP pair. If the destination LLC user rejects the connection request, its LLC entity returns a Disconnected Mode (DM) PDU.

Once the connection is established, data is exchanged, as in HDLC, using information PDUs. Information PDUs contain the sent and received sequence numbers for sequential order management and flow control. As in HDLC, supervisory PDUs are used for error and flow control. Either LLC entity can terminate a logical LLC connection by sending a Disconnect PDU (DISC).

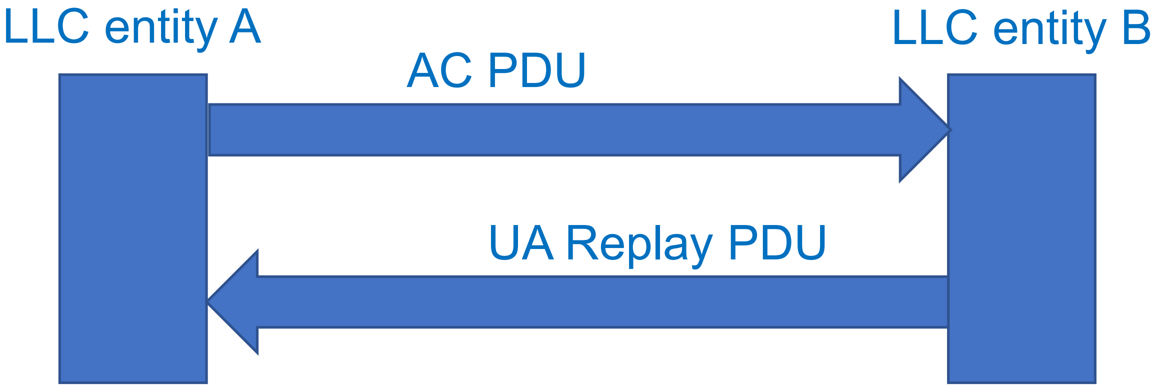
Establishment of the connection



UA: Unnumbered Acknowledgment
DM: Disconnected Mode
AC: Acknowledged Connectionless
SABME: set asynchronous balanced mode extended)

PROTOCOL ARCHITECTURE IN LAN NETWORKS- LLC PROTOCOL

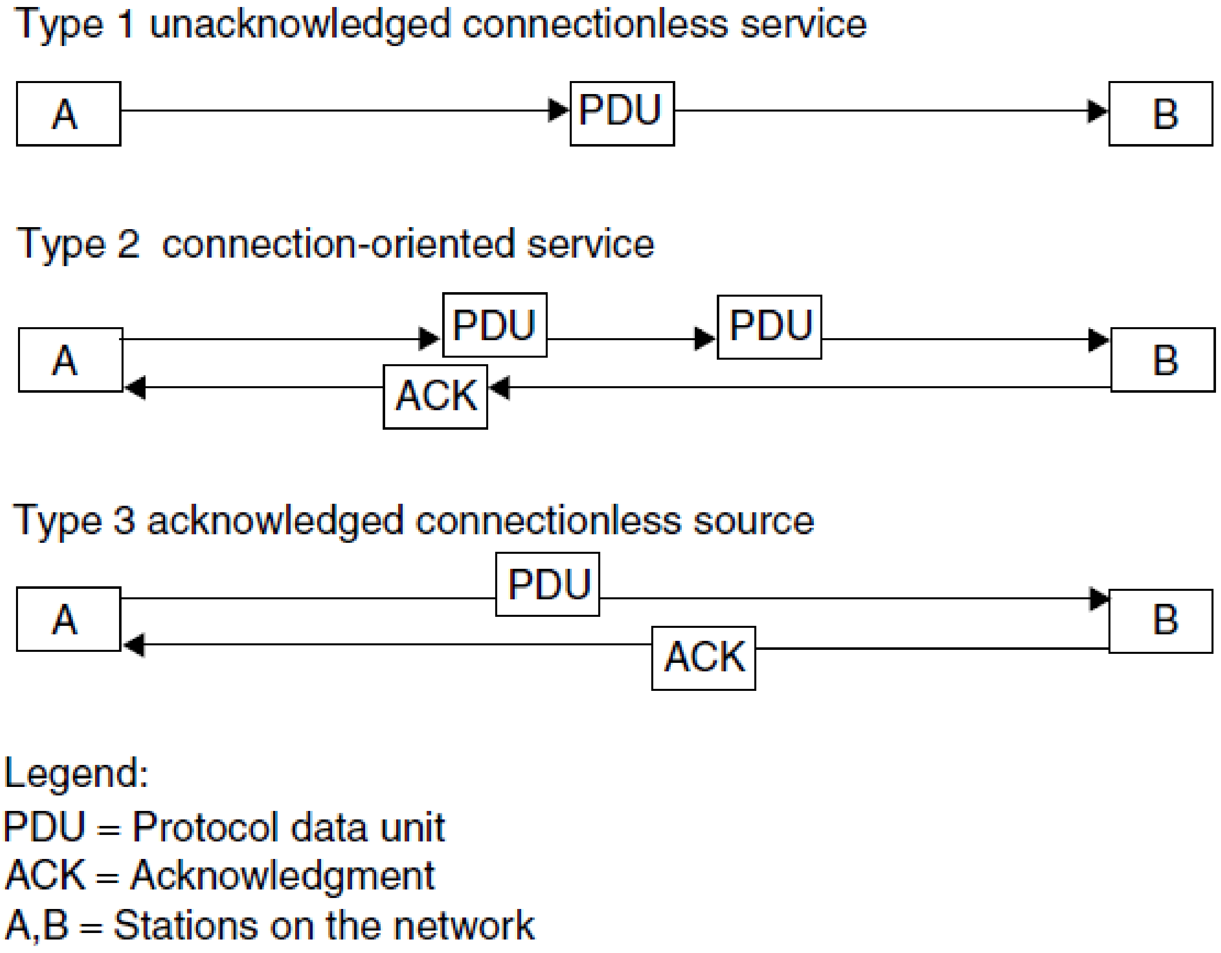
In type 3 operation, each PDU transmitted is acknowledged. A new unnumbered PDU is defined (not existing in HDLC): the Acknowledged Connectionless (AC) information PDU. User data is sent in successive AC Command PDUs, and must be acknowledged using an AC Reply PDU.



UA: Unnumbered Acknowledgment
Dm: disconnected Mode
AC: Acknowledged Connectionless

PROTOCOL ARCHITECTURE IN LAN NETWORKS- LLC PROTOCOL

To prevent PDU losses, a 1-bit sequence number is used, so that the sender alternates the use of 0 and 1 in its AC command PDUs and the receiver responds with an AC PDU with the opposite number of the command received. Only one PDU can be sent in each direction at a given time.



Type 1 unacknowledged connectionless service

Type 2 connection-oriented service

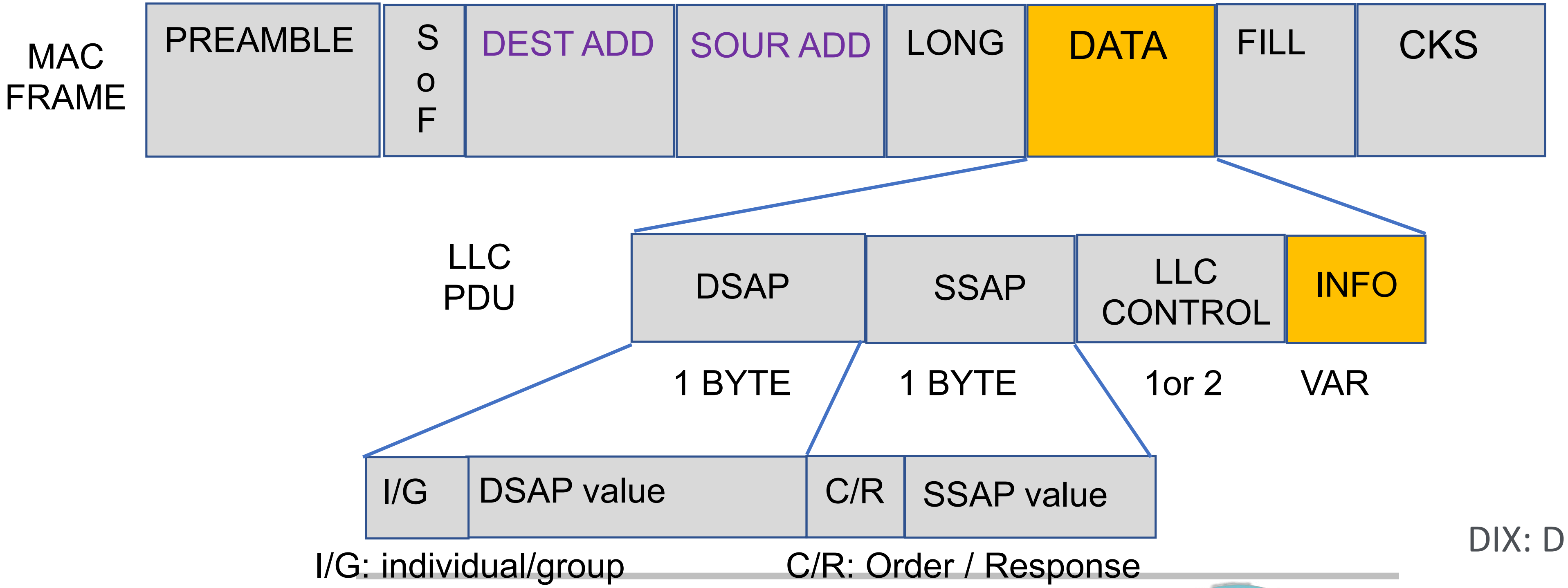
Type 3 acknowledged connectionless source

Legend:
PDU = Protocol data unit
ACK = Acknowledgment
A,B = Stations on the network

PROTOCOL ARCHITECTURE IN LAN NETWORKS- LLC PROTOCOL

The LLC header contains three fields: a destination access point, a source access point, and a control field.

Each of the Destination Service Access Point (DSAP) and Source Service Access Point (SSAP) fields contains a 7-bit address that specifies the destination and source LLC users. One bit of the DSAP field indicates whether the address is individual or group, while one bit of SSAP indicates whether the PDU is a command or a response (they replace the DIX Type field). The format of the LLC control field is identical to that of HDLC, making use of extended sequence numbers (7 bits).



DIX: DEC, Intel and Xerox

PROTOCOL ARCHITECTURE IN LAN NETWORKS- MAC

Media access control

All LANs and MANs consist of a set of devices that must share the transmission capacity of the network, so some method of access control to the medium is required in order to make efficient use of this capacity.

This is the function of the medium access control (MAC) protocol. The key parameters in any media access control technique are where and how.

Where refers to whether control is carried out centrally or distributed.

In a centralized scheme, a controller is designed with the authority to grant access to the network, so a station wishing to transmit must wait until permission is granted by the controller.

In a decentralized network, stations jointly perform the function of media access control to dynamically determine the order in which they will transmit.

A centralized scheme has certain advantages, among which are:

- Can improve access control by providing priorities, rejections and guaranteed capacity.
- Allows the use of relatively simple access logic at each station.
- Resolves distributed coordination problems between peer entities.

The main disadvantages of centralized schemes are:

- Generates a failure point; That is, there is a point in the network such that if a failure occurs at it, it will fail the entire network
- It can act as a bottleneck, reducing performance.The pros and cons of distributed schemes are the opposite of the previous points.

PROTOCOL ARCHITECTURE IN LAN NETWORKS- MAC

The second parameter, how, is imposed by the topology and is a compromise between factors such as cost, performance and complexity. In general, we can classify access control techniques as static or dynamic.

With static techniques, a given capacity is dedicated to a connection. This is the same approach used in circuit switching, frequency division multiplexing (FDM), and synchronous time division multiplexing (TDM). These techniques are not optimal in LAN and MAN networks since the needs of the stations are unpredictable. It is therefore preferable to have the ability to reserve capacity dynamically more or less in response to immediate requests.

The dynamic approach can be subdivided into three categories: circular rotation, reserve and containment.

Circular rotation

With the circular rotation technique, each station is given the opportunity to transmit, to which the station can decline the offer or can transmit subject to an upper limit, generally specified in terms of the amount of data to transmit or the time for it. In any case, when the station finishes it must hand over the transmission turn to the next station in the logical sequence. Sequence control can be centralized or distributed, with the polling method being an example of a centralized technique.

When several stations have data to transmit over a long period of time, circular rotation techniques can be very efficient. On the other hand, if only a few stations have data to transmit for an extended period of time, there will be a considerable cost in passing the turn between stations, since most of them do not transmit data but only give up the turn. In these circumstances other techniques may be preferable depending on whether the data traffic is bursty or continuous. Continuous traffic is characterized by long, reasonably continuous transmissions; Some examples are voice communication, telemetry, and large file transfer. On the other hand, bursty traffic is characterized by short and sporadic transmissions, as in the case of terminal-station interactive traffic.

PROTOCOL ARCHITECTURE IN LAN NETWORKS- MAC

Reservation
Reservation techniques are suitable for continuous traffic. Generally, in these techniques, time is divided into slots, as in the case of the synchronous TDM technique. A station that wishes to transmit reserves future slots for a long, even indefinite, period of time. Again, bookings can be carried out centrally or distributed.

Containment
Containment techniques are generally appropriate for bursty traffic. With these techniques, there is no control to determine whose turn it is, but rather all the stations compete in a way that can be, as we will see, quite rude and chaotic. These techniques are necessarily distributed in nature, their main advantage being the fact that they are simple to implement and efficient under low or moderate load conditions. However, for some of these techniques, performance tends to deteriorate under high load conditions. Although both centralized and distributed reservation techniques are implemented in some LAN products, the most common are circular rotation and containment techniques.

PROTOCOL ARCHITECTURE IN LAN NETWORKS- MAC

The media access control (MAC) sublayer
Networks can be divided into two categories: those that use point-to-point connections and those that use broadcast channels.
In any broadcast network, the key issue is how to determine who can use the channel when there is competition for it. When there is only one channel, determining who should have the turn is very complicated. Many protocols are known to solve this problem.
In the literature, broadcast channels are sometimes called multi-access channels or random access channels. The protocols used to determine who is next on a multi-access channel belong to a sublayer of the data link layer called the MAC (Medium Access Control) sublayer.
From a technical point of view, the MAC sublayer is the bottom part of the data link layer.

The channel assignment problem:
The central issue of this topic is how to assign a single broadcast channel between competing users.

- **Static channel assignment in LANs and MANs**

The traditional way of allocating a single channel, such as a telephone trunk, among several competing users is FDM (Frequency Division Multiplexing).If there are N users, the bandwidth is divided into N parts of equal size, and each user is assigned a part.When there are only a small fixed number of users, each of which has (buffered) a heavy traffic load (for example, the switching offices of a carrier company), FDM is a simple and efficient allocation mechanism. However, when the number of senders is large and varies continuously, or when traffic is bursty, FDM presents some problems. The basic problem is that when some users are idle, their bandwidth is simply wasted.

PROTOCOL ARCHITECTURE IN LAN NETWORKS- MAC

- Dynamic channel assignment in LANs and MANs

- CSMA with collision detection

The persistent and non-persistent CSMA protocols are certainly an improvement over ALOHA because they ensure that no station starts transmitting when it detects that the channel is busy. Another improvement is for stations to abort their transmissions as soon as they detect a collision. In other words, if two stations detect that the channel is idle and begin transmitting simultaneously, both will detect the collision almost immediately. Instead of finishing transmitting their frames, which are hopelessly corrupted anyway, they must abruptly stop transmission as soon as they detect the collision. Prompt termination of damaged frames saves time and bandwidth.This is achieved with a protocol, known as CSMA/CD (Carrier Sense Multiple Access and Collision Detection), and is widely used in LANs at the MAC sublayer. CSMA/CD, like many other LAN protocols, uses the conceptual model in the figure below. At the point marked t0, a station has finished transmitting its frame. Any other station that has a frame to send can now attempt to do so. If two or more stations decide to transmit simultaneously, there will be a collision. Collisions can be detected by comparing the power or pulse width of the received signal with that of the transmitted signal. Once a station detects a collision, it aborts the transmission, waits a random amount of time, and tries again, assuming that no other station has started transmitting during that time. Therefore, our CSMA/CD model will consist of alternating periods of contention and transmission, with periods of inactivity occurring when all stations are idle (for example, due to lack of work).

PROTOCOL ARCHITECTURE IN LAN NETWORKS- MAC

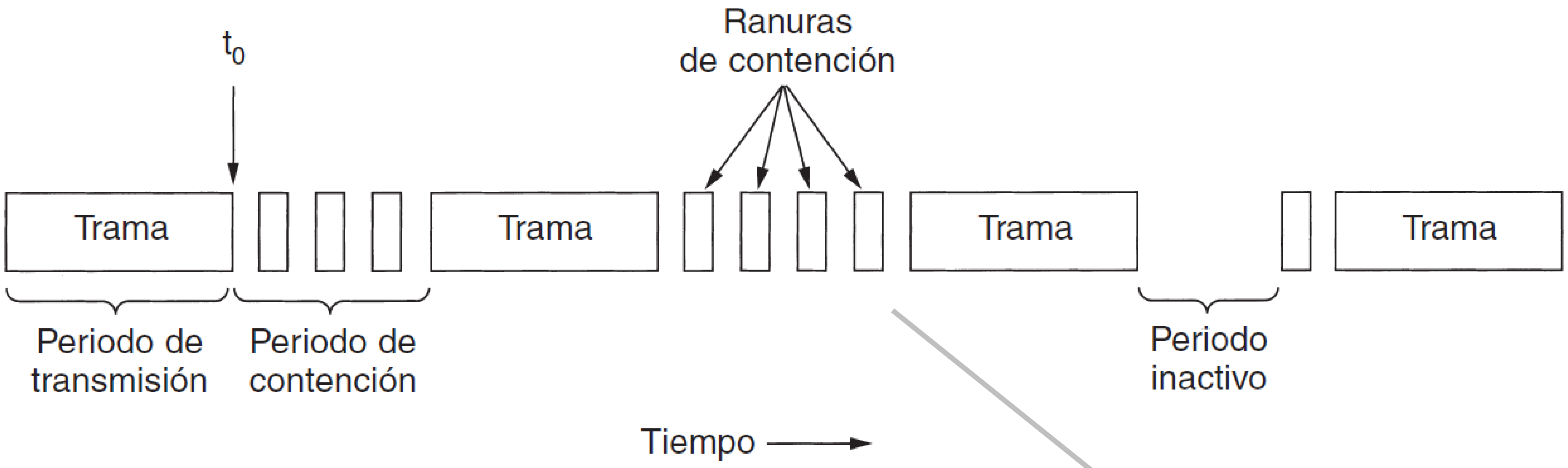
It is important to realize that collision detection is an analog process. The station's hardware must listen to the cable while transmitting. If what you read is different than what you put in it, you know a collision is occurring.

The implication is that the signal coding must allow collisions to be detected. For this reason, special encoding is usually used.

It is also worth mentioning that a broadcast station must continually monitor the channel for bursts of noise that could indicate a collision. For this reason, single-channel CSMA/CD is inherently a half-duplex system. It is impossible for a station to transmit and receive frames at the same time, because receive logic is in use, checking for collisions during each transmission.

To avoid any misunderstanding, it is important to note that no MAC sublayer protocol guarantees reliable delivery. Even in the absence of collisions, the receiver might not have correctly copied the frame for various reasons (for example, lack of buffer space or an undetected interrupt).

PROTOCOL ARCHITECTURE IN LAN NETWORKS- MAC



equal to the worst case round trip propagation time in the cable (2T)

Now let's see how the randomization process is carried out when a collision occurs. The model is the one in the figure above. After a collision, time is divided into discrete slots whose length is equal to the worst-case round-trip propagation time in the cable (2T). Taking into account the longest path allowed by Ethernet, the slot time was set to 512 bit times, or 51.2 μsec (*)

After the first collision, each station waits 0 or 1 slot times before trying again. If two stations collide and both choose the same random number, there will be a new collision. After the second collision, each one chooses 0, 1, 2, or 3 at random and waits that number of slot times. If a third collision occurs (the probability of this occurring is 0.25), then for the next time the number of slots to wait will be chosen at random from the interval 0 to 23 − 1. In general, after i collisions, a random number between 0 and 2i − 1 is chosen, and that number is skipped of slots. However, after 10 collisions have been reached, the randomization interval freezes in a maximum of 1023 slots.

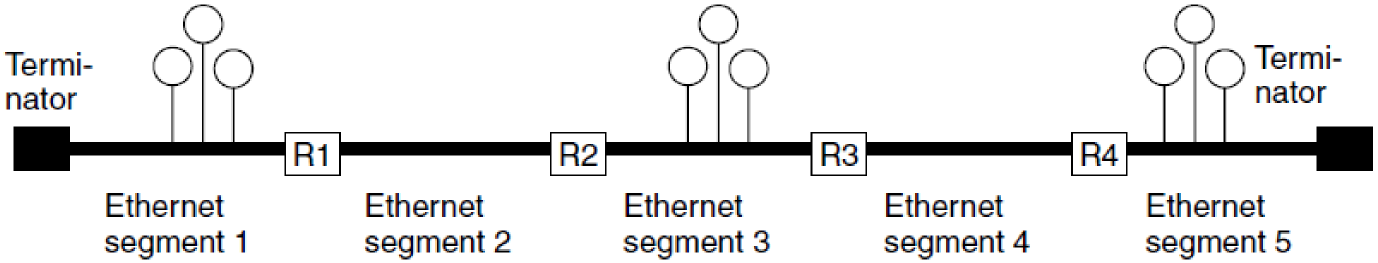
(*) Just an example for 10 Mbps

PROTOCOL ARCHITECTURE IN LAN NETWORKS- MAC

After 16 collisions, the controller abandons and reports acomputer failure. Subsequent recovery is the responsibility of the upper layers.This algorithm, called binary exponential backoff, was chosen to dynamically adapt the number of stations attempting to transmit. If the randomization interval for all collisions were 1023, the chance of two stations colliding a second time will be negligible, but the average wait after a collision will be hundreds of slot times, which introduces a significant delay. On the other hand, if each station is always delayed by 0 or 1 slots, then, by trying to transmit 100 stations at the same time, there would be collisions over and over again, until 99 of them chose 1 and the remaining station chose 0. This could take years. By making the randomization interval grow exponentially as more and more collisions occur, the algorithm ensures a small delay when only a few stations collide, but also ensures that the collision is resolved within a reasonable interval when there are collisions between many stations. Truncating the rollback to 1023 prevents the limit from growing too much.

LAN ETHERNET NETWORKS.

The Internet and ATM were designed for wide-area connectivity. However, many companies Universities and other organizations had a large number of computers that require interconnection. This need gave rise to the local area network. In this section we will say something about the most popular LAN - Ethernet.

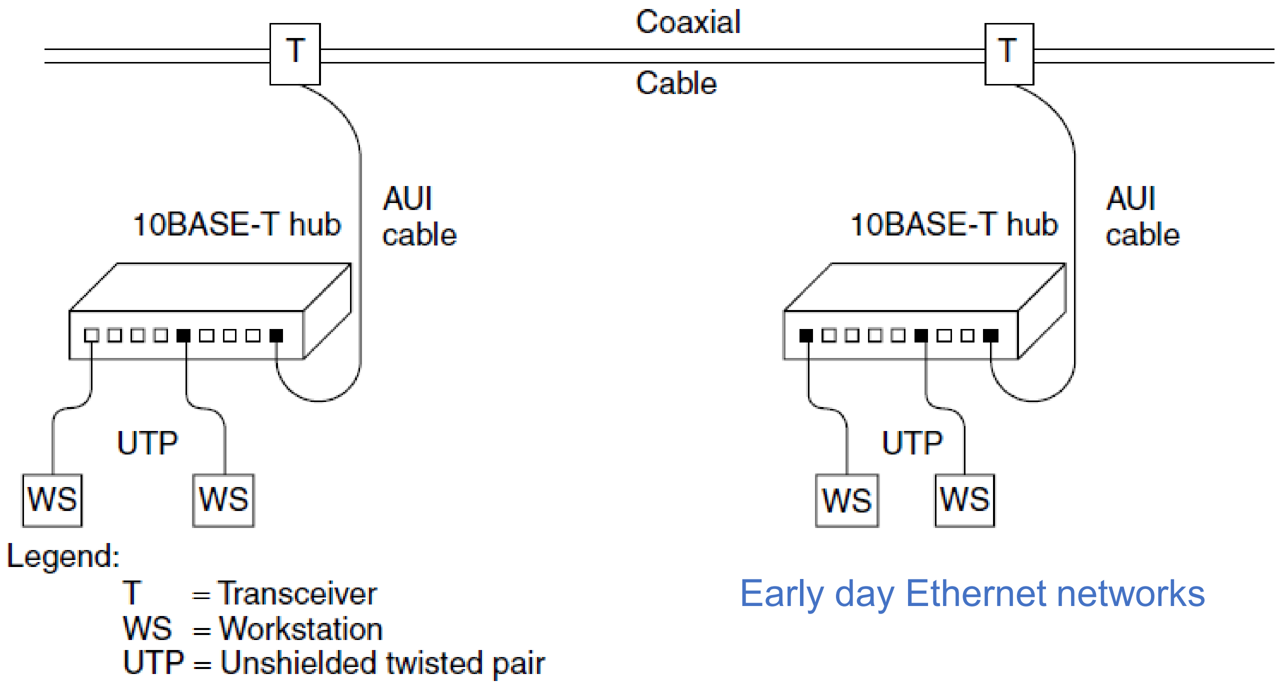


A student named Bob Metcalfe did his undergraduate degree at M.I.T. and then moved on to get his Ph.D. at Harvard. During his studies, he was introduced to Abramson's work. He became so interested in it that after graduating from Harvard he decided to spend the summer in Hawaii working with Abramson before going to work at Xerox's Palo Alto Research Center (PARC). When he arrived at PARC, he saw that the researchers there had designed and built what would later be called personal computers. But the machines were isolated. Applying his knowledge of Abramson's work, along with Abramson's colleague David Boggs, he designed and implemented the first local area network (Metcalfe and Boggs, 1976). They called the system Ethernet.   Xerox Ethernet was so successful that DELL, Intel, and Xerox designed a standard in 1978. for a 10 Mbps Ethernet, called the DIX standard. With two minor changes, in 1983 the DIX standard became the **IEEE 802.3 standard.**

LAN ETHERNET NETWORKS.

Ethernet continued its development and is still in development. New versions have been released at 100 Mbps,1 and 10 Gbps, and even higher. Cabling has also been improved and switching has been added and other features.

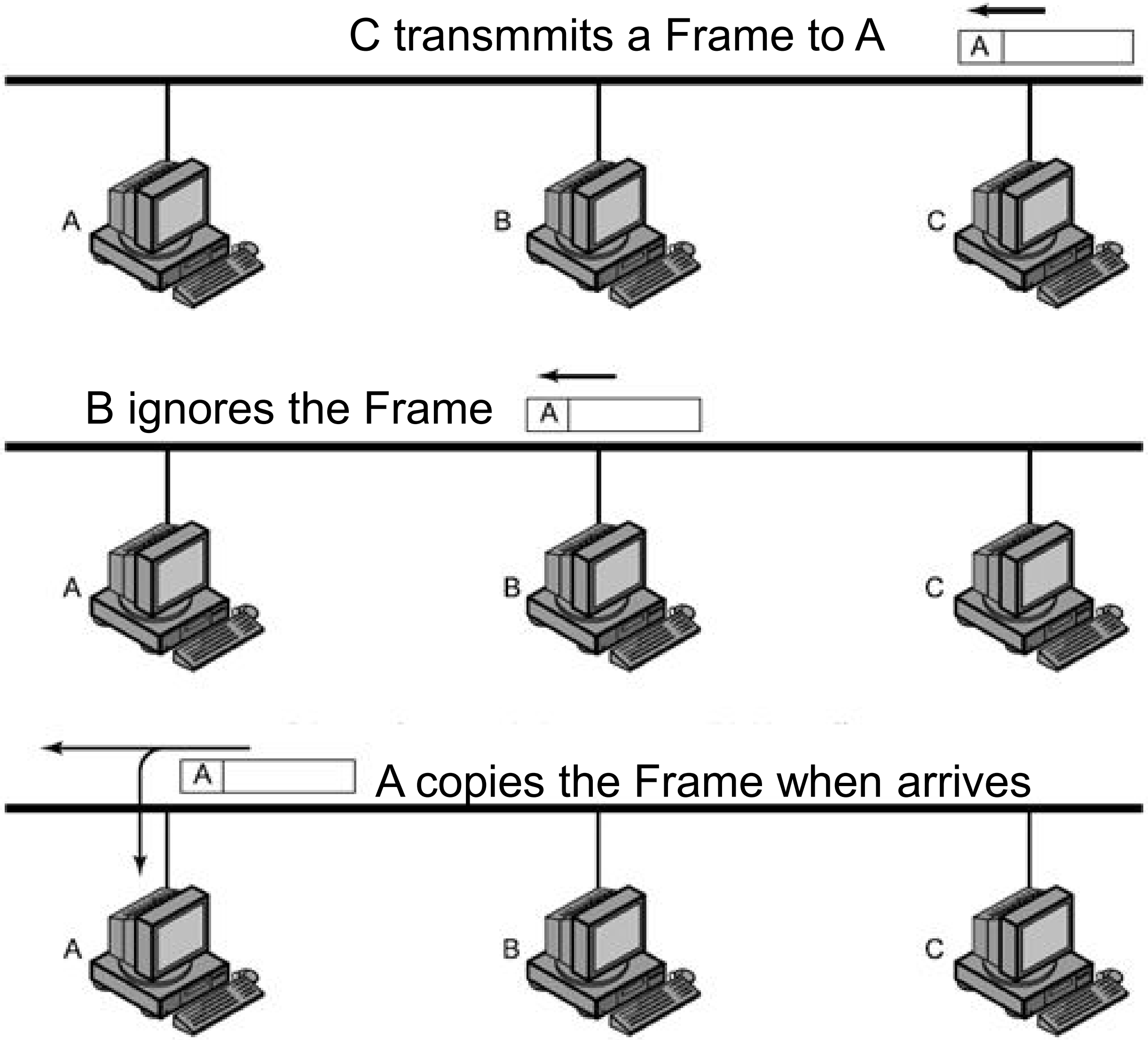
By the way, Ethernet (IEEE 802.3) is not the only LAN standard. The committee also standardized Token Bus (802.4) and Token Ring (802.5).



Early day Ethernet networks

Ethernet is a broadcast network. Broadcast networks have a single communication channel, so all machines on the network share it. If one machine sends a short message - in certain contexts known as a frame - all the others receive it. An address field within the frame specifies the recipient. When a machine receives the frame, it checks the address field. If the frame is destined for that machine, it processes it; If it is intended for someone else, he ignores it.

LAN ETHERNET NETWORKS.



Broadcast systems typically also allow addressing of a frame to all destinations using a special code in the address field.
When a frame with this code is transmitted, all machines on the network receive it and process it.
This mode of operation is known as broadcasting.
Some broadcast systems also support broadcasting to a subset of machines, known as multicasting.

LAN ETHERNET NETWORKS.

The Ethernet MAC Sublayer Protocol
The following figure shows the original DIX frame structure (DEC, Intel, Xerox) and the one standardized by the IEEE (802.3). Each frame starts with an 8-byte Preamble, each of which contains the bit pattern 10101010.The Manchester encoding of this pattern produces a 10 MHz square wave for 6.4 µsec (*) to allow the receiver's clock to synchronize with that of the sender. They are asked to remain synchronized for the rest of the frame, using Manchester encoding to keep track of bit boundaries.

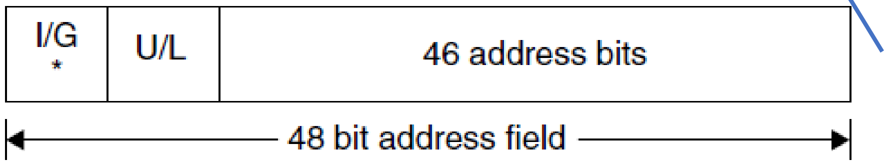
The frame contains two Addresses, one for the destination and one for the origin.
The standard allows for 2- and 6-byte addresses, but the parameters defined for the 10 Mbps baseband standard use only 6-byte addresses.
The high-order bit of the destination address is 0 for ordinary addresses and 1 for group addresses.
Group addresses allow multiple stations to listen in a single direction. When a frame is sent to a group address, all stations in the group receive it. Sending to a group of stations is called multicast. The address consisting only of 1 bits is reserved for broadcast.

A frame containing only bits 1 in the destination field is accepted at all stations on the network. The difference between broadcast and multicast is important enough to warrant repetition. A multicast frame is sent to a selected group of stations on the Ethernet; a broadcast frame is sent to all stations on the Ethernet.

(*): 10 MHz: 0,1 µsec per bit, 8 bits : 6,4 µsecs

LAN ETHERNET NETWORKS.

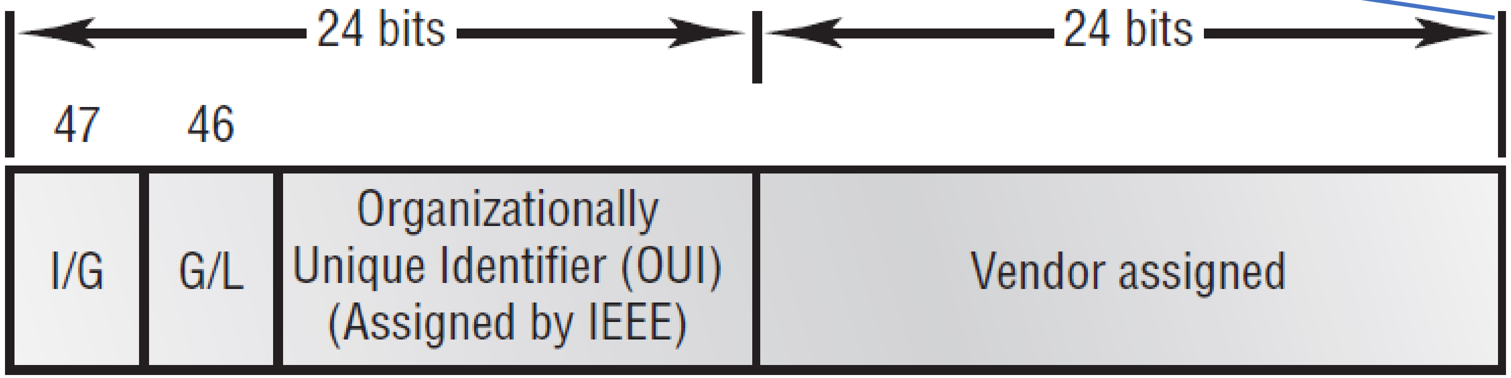
	8	6	6	2	0-1500	0-46	4
Eth DIX	PREAMBLE 01 01 01 01	DEST ADD	SOUR ADD	TYPE	DATA	FILL	CKS
IEEE 802.3	PREAMBLE SoF	DEST ADD	SOUR ADD	LONG	DATA	FILL	CKS



(b) 6 byte field (ethernet and IEEE 802.3)

bit subfield '0' = individual address '1' = group address
bit subfield '0' = universally administrated addressing
'1' = locally administrated addressing

* Set to '0' in source address field



LAN ETHERNET NETWORKS.

Multicast is more selective, but involves managing groups. Diffusion is less sophisticated but does not require group management.

Another interesting feature of addressing is the use of bit 46 (adjacent to the high order bit) to distinguish local from global addresses. Local addresses are assigned by each network administrator and have no meaning outside the local network. In contrast, global addresses are assigned by the IEEE to ensure that no two stations anywhere in the world have the same global address. With 48 − 2 = 46 bits available, there are about 7 × 1013 global addresses. The idea is that any station can uniquely address any other station by simply giving the correct 48-bit number. It is the job of the network layer to find a way to locate the destination.

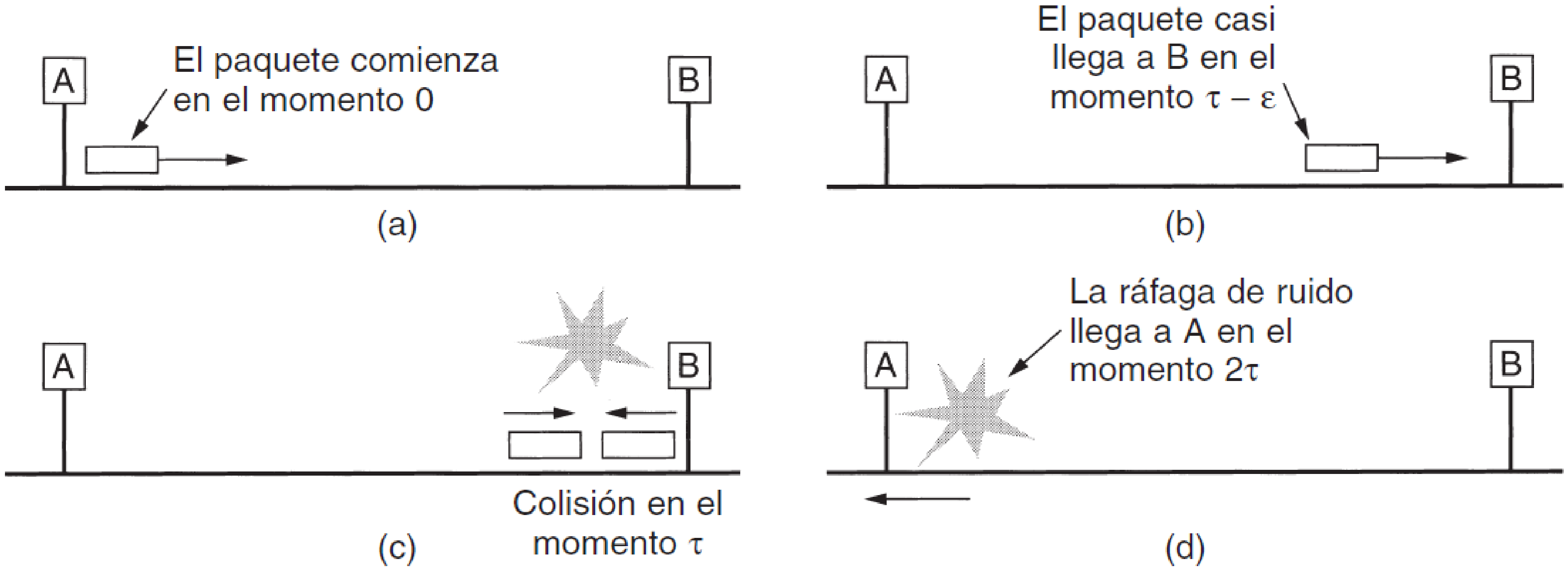
Next is the **Type** field, which tells the receiver what to do with the frame. It is possible to use multiple network layer protocols at the same time on the same machine, so when an Ethernet frame arrives, the kernel must know which one to deliver the frame to. The Type field specifies which process to give the frame to. When the IEEE standardized Ethernet, the committee made two changes to the DIX format. The first was to reduce the preamble to 7 bytes and use the last byte for a start-of-frame delimiter, for compatibility with 802.4 and 802.5. The second was to change the Type field into a Length field. Of course, there was now no way for the receiver to know what to do with the incoming frame, but that problem was solved by adding a small header to the data portion to provide this information.

Then there is the **Data**, up to **1500 bytes**. This limit was chosen somewhat arbitrarily when the DIX standard was established, mainly based on the fact that a transceiver needs enough RAM to hold an entire frame and RAM was very expensive in 1978.

The final Ethernet field is the Checksum (FCS). In fact, this is a 32-bit hash code of the data. If some data bits are received erroneously (due to noise in the cable), the checksum is almost certainly wrong, and the error will be detected. The checksum algorithm is a cyclic redundancy check (CRC). It simply performs error detection, not forward error correction (FEC).

LAN ETHERNET NETWORKS.

A higher limit could have meant more RAM and therefore a more expensive transceiver.In addition to there being a maximum frame length, there is also a minimum frame length. While sometimes a 0-byte data field is useful, it causes problems. When a transceiver detects a collision, it truncates the current frame, meaning that lost bits and pieces of frames appear all the time on the cable.For Ethernet to easily distinguish valid frames from garbage, it requires those frames to be at least 64 bytes long, from the destination address to the checksum, including both. If the data portion of a frame is less than 46 bytes, the Padding field is used to pad the frame to the minimum size. Another (more important) reason for having a minimum frame length is to prevent a station from completing the transmission of a short frame before the first bit reaches the far end of the cable, where it could have a collision with another frame. This problem is illustrated in the figure below.


A
El paquete comienza en el momento 0
B
(a)
A
El paquete casi llega a B en el momento τ – ε
B
(b)
A
B
Colisión en el momento τ
(c)
A
La ráfaga de ruido llega a A en el momento 2τ
B
(d)

LAN ETHERNET NETWORKS.

At time 0, station A, at one end of the network, sends a frame. Let's call Ƭ the time it takes for this frame to reach the other end. Just before the frame reaches the other end (i.e., at time Ƭ − Ɛ) the most distant station, B, begins transmitting. When B detects that it is receiving more power than it is sending, it knows that a collision has occurred, so it aborts its transmission and generates a 48-bit noise burst to warn the other stations. In other words, she clutters the cable to make sure the sender doesn't ignore the collision. At approximately time 2Ƭ, the sender sees the noise burst and also aborts its transmission; then wait a random time before retrying.

If a station attempts to transmit a very short frame, a collision is conceivable, but the transmission is completed before the noise burst arrives back, at time 2Ƭ. The sender will then incorrectly assume that the frame was sent successfully. To prevent this situation from occurring, all frames should take more than 2Ƭ to send, so that transmission is still taking place when the noise burst returns to the sender. For a 10 Mbps LAN with a maximum length of 2500 meters and four repeaters (from the 802.3 specification), the round trip time (including the propagation time through the four repeaters) has been determined to be approximately 50 µsec in the worst case, including the time to pass through the repeaters, which is certainly non-zero. Therefore, the minimum frame must take at least this long to transmit. At 10 Mbps, one bit takes 100 nsec, so 500 bits is the smallest frame guaranteed to work. To add some margin of safety, this number was rounded to 512 bits or 64 bytes. Frames with less than 64 bytes are padded to 64 bytes with the Padding field.

As network speed increases, the minimum frame length should increase, or the maximum cable length should decrease, proportionally. For a 2500 meter LAN operating at 1 Gbps, the minimum frame size would have to be 6400 bytes. Alternatively, the minimum frame size could be 640 bytes and the maximum distance between two stations 250 meters.

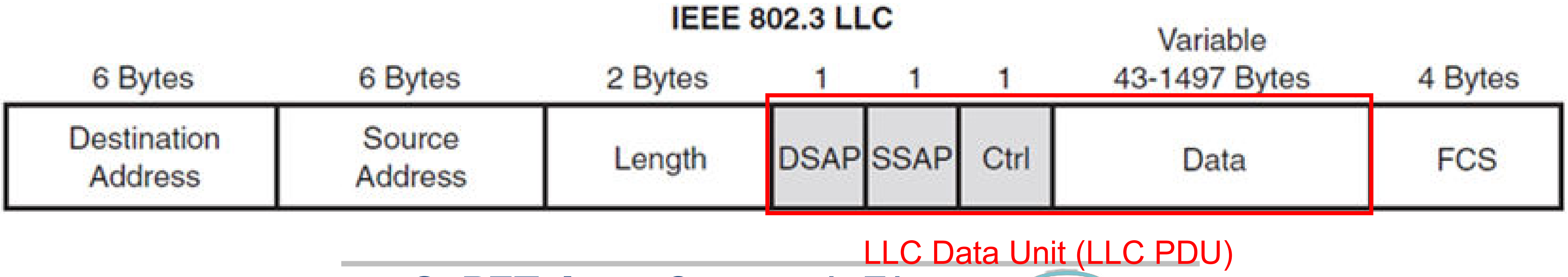
LAN ETHERNET NETWORKS.

As described so far, CSMA/CD does not provide confirmation of receipt. Since the simple absence of collisions does not guarantee that the bits were not altered by noise spikes on the cable, for reliable communication the destination must verify the checksum and, if correct, return a receipt confirmation frame to the source. . Typically, this acknowledgment would just be another frame, as far as the protocol is concerned, and would have to fight for channel time in the same way as a data frame. However, a simple modification of the contention algorithm allows rapid confirmation of receipt of a frame. All that will be needed is to reserve for the destination station the first contention slot following the next successful transmission. Unfortunately, the standard does not provide this possibility.

IEEE 802.2 STANDARD

IEEE 802.2 Standard: Logical Link Control

The IEEE has defined a protocol that can operate on top of all Ethernet and 802 protocols. Additionally, this protocol, called LLC (Logical Link Control), hides the differences between the different types of 802 networks, providing a single format and interface with the network layer.
This format, interface and protocol are closely based on HDLC. The LLC forms the top half of the data link layer, with the MAC sublayer below it.
The typical use of the LLC is as follows. The network layer of the sending machine passes a packet to the LLC using the LLC's access primitives. The LLC sublayer then adds an LLC header containing the sequence and receipt numbers. The resulting structure is then entered into the payload field of a frame 802 and transmitted. In the receiver the reverse process occurs.



IEEE 802.3 LLC

6 Bytes	6 Bytes	2 Bytes	1	1	1	Variable 43-1497 Bytes	4 Bytes
Destination Address	Source Address	Length	DSAP	SSAP	Ctrl	Data	FCS

LLC Data Unit (LLC PDU)

IEEE 802.2 STANDARD

When the IEEE standardized Ethernet, the committee made two changes to the DIX format. The first was to reduce the preamble to 7 bytes and use the last byte for a Start of Frame delimiter, for compatibility with 802.4 and 802.5.
The second was to change the **Type** field into a **Length** field.

XNS: 0x0600 : 0000 0110 0000 0000 : 1024 +512 : 1536   >   1500
DECNET : 0x6003 : 0110 0000 0000 0011 : 8192 + 4096 +3 :   12291   > 1500
IP: 0x0800 : 0000 1000 0000 0000 : 2048 > 1500

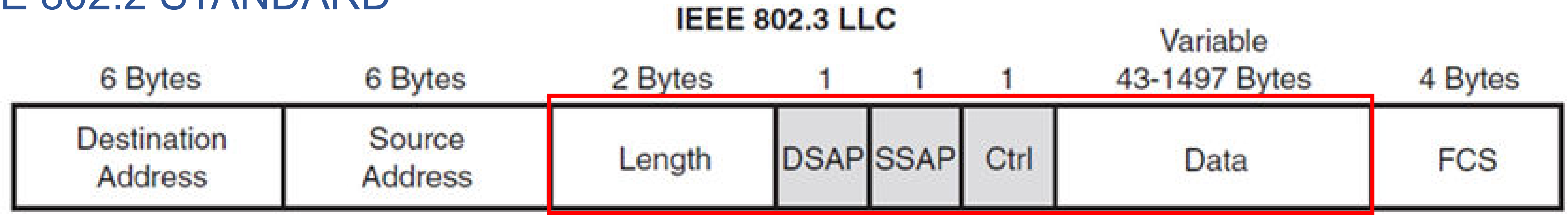
Of course, there was now no way for the receiver to know what to do with the incoming frame, but that problem was solved by adding a small header to the data portion to provide this information.

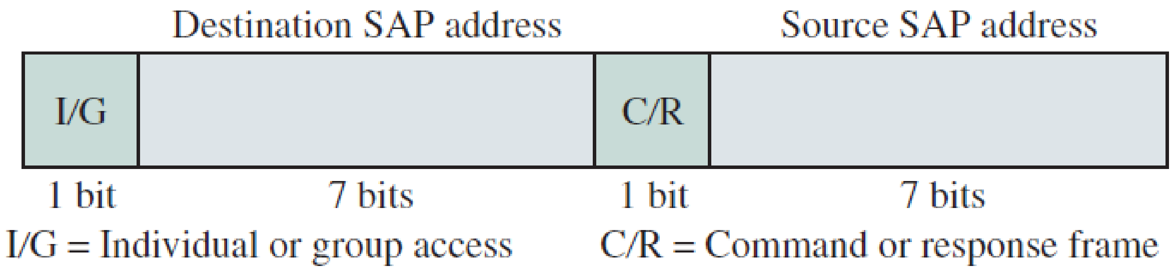
Unfortunately, by the time 802.3 was released, so much hardware and software was already in use for Ethernet DIX that very few manufacturers and users were willing to convert the Type field to a Length field.

In 1997, the IEEE dropped the fight and said the two forms were a good fit.

Fortunately, all Type fields in use before 1997 were larger than 1500 (in decimal), (0x0600 XNS [Xerox], 0x0800 IP [Internet Protocol], and 0x6003 [DECNET]). Consequently, any number less than or equal to 1500 can be interpreted as Length (802.3), and any number larger than 1500 can be interpreted as Type (Ethernet Type, Type II).
Additionally, a new form of packet type field was needed so a logical link control (LLC) header with destination and source service access points (DSAP and SSAP, respectively) and control fields follow the field Length for the top level protocol identification, as shown in previous figure:

IEEE 802.2 STANDARD





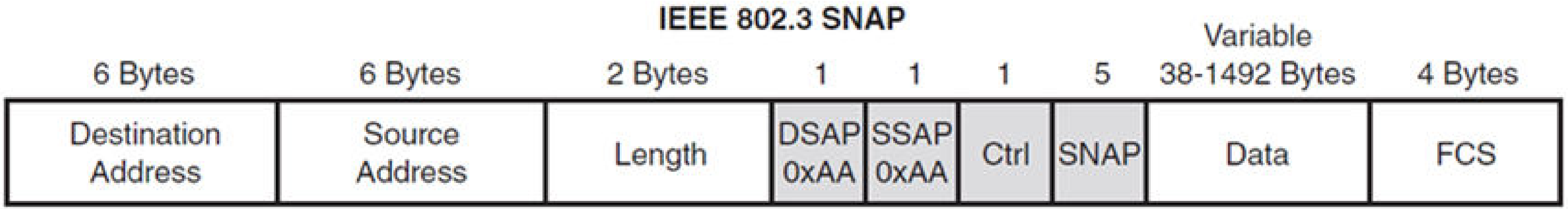
The control information is carried within the Data field of the MAC frame.

The new fields are the following:

- Length: this is the length of the frame, it includes the fields: FCS, addresses and length(excludes preamble)
- DSAP: a value of 0xAA indicates SNAP (Subnetwork Access Protocol)
- SSAP: a value of 0xAA indicates SNAP
- Control: the control field specifies the type of LLC frame

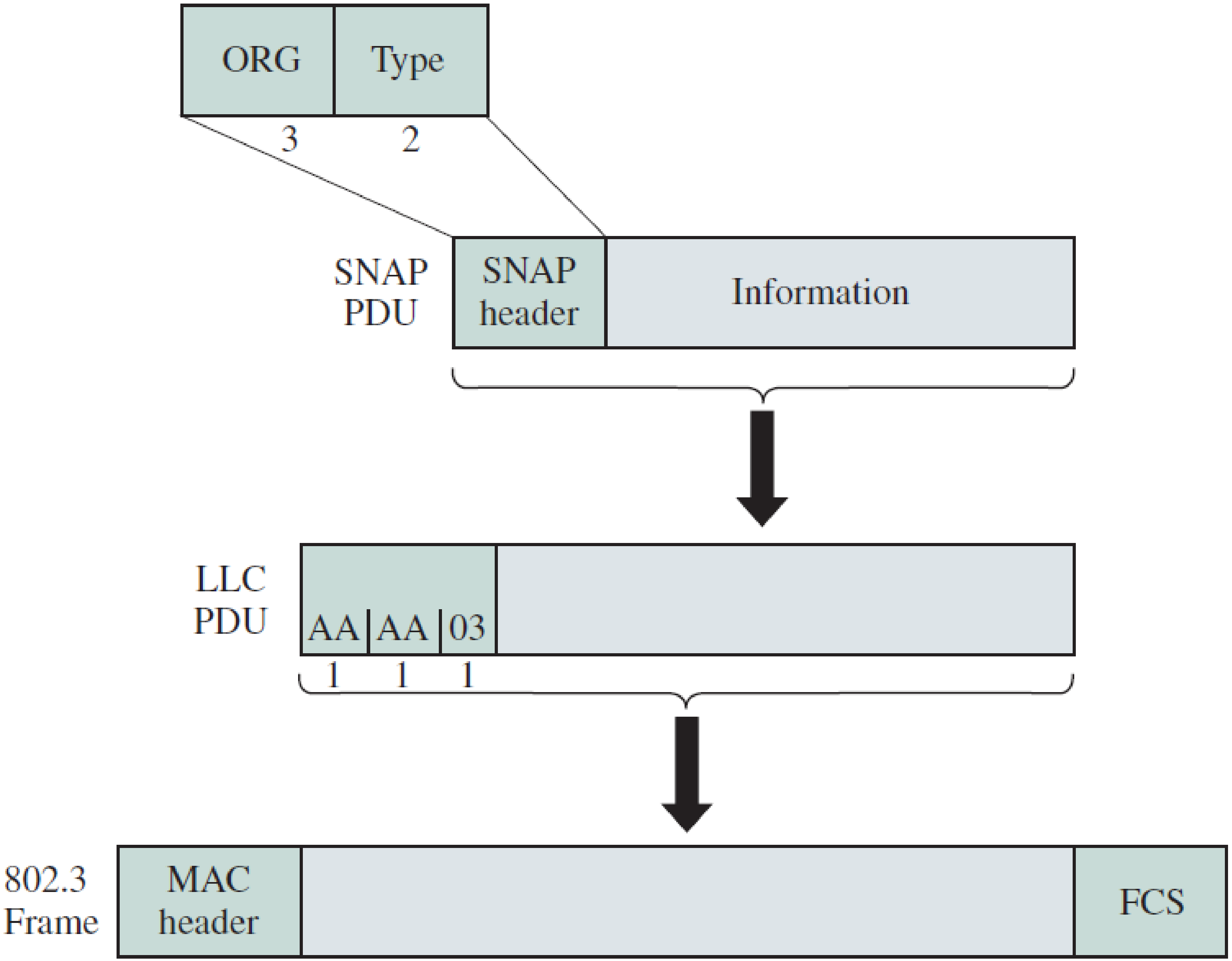
IEEE 802.2 STANDARD

The following figure shows the format of an IEEE 802.3 SNAP frame that is indicated by the DSAP and SSAP values and that includes the SNAP field. The SNAP header includes 3 bytes of vendor code and 2 bytes of local code. A vendor code of 0s (0x000000) indicates that the local code is an Ethernet Type II for backward compatibility. This new format moves the Ethernet Type field 8 bytes from its original location in Ethernet II.

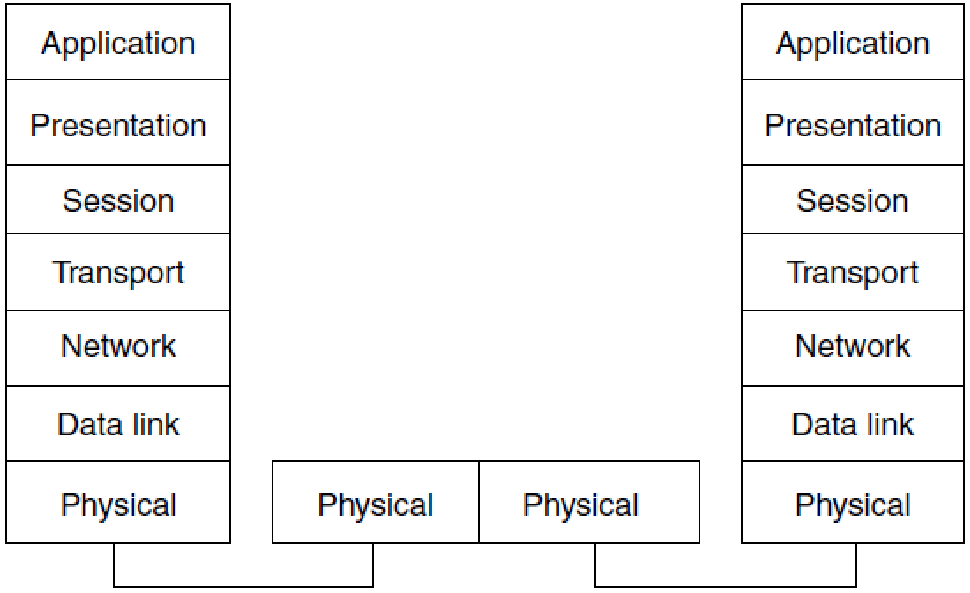


IEEE 802.3 SNAP

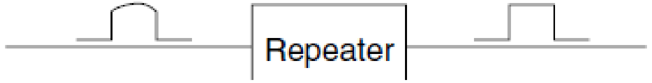
6 Bytes	6 Bytes	2 Bytes	1	1	1	5	Variable 38-1492 Bytes	4 Bytes
Destination Address	Source Address	Length	DSAP 0xAA	SSAP 0xAA	Ctrl	SNAP	Data	FCS


ORG
Type
3
2
SNAP PDU
SNAP header
Information
LLC PDU
AA
AA
03
1
1
1
802.3 Frame
MAC header
FCS

DEVICES- HUB

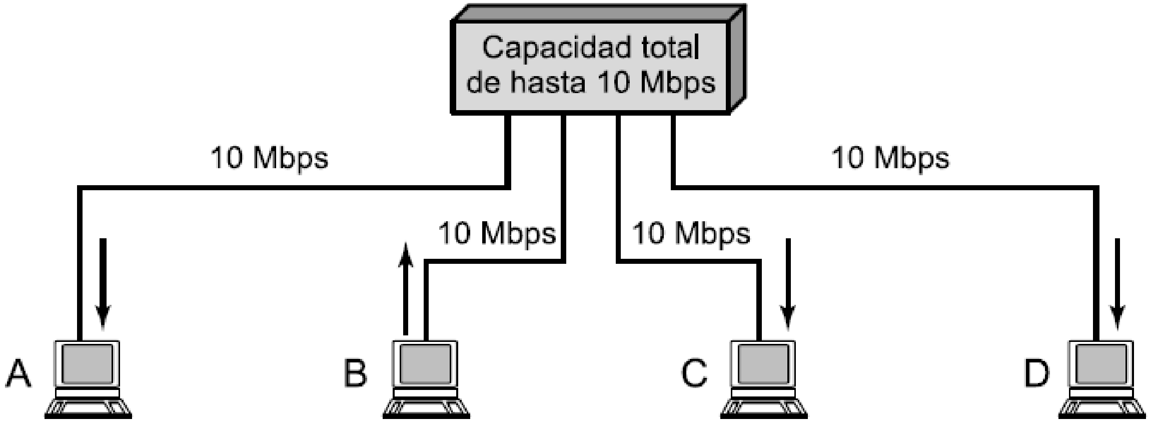


(a) OSI operation



(b) Signal regeneration process

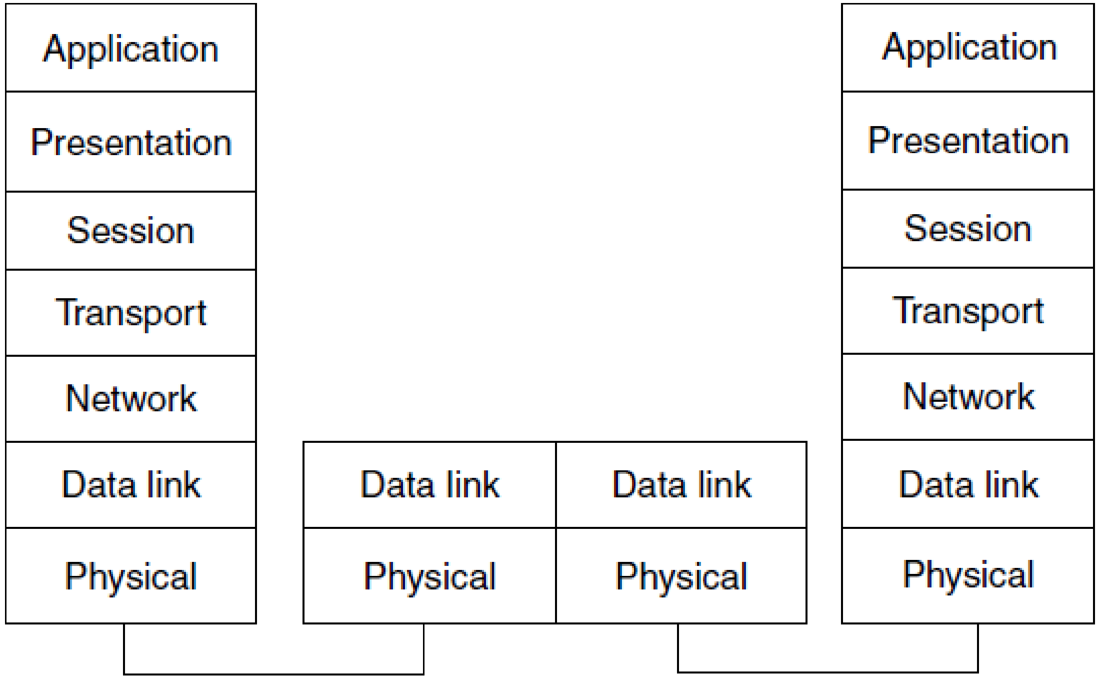
The concentrator (Hub) is an active element that acts as the central element of the star. Each station is connected to the hub by two links (transmit and receive). The hub acts as a repeater: when a single station transmits, the hub replicates the signal on the line out to each station.



A hub uses a star structure to connect stations to it, so that the transmission from one station is received at the hub and retransmitted over all output lines. Therefore, to avoid the occurrence of collision, only one station can transmit at a given time. Again, the total capacity of the LAN is 10 Mbps.

DEVICES- BRIDGE

OSI operation

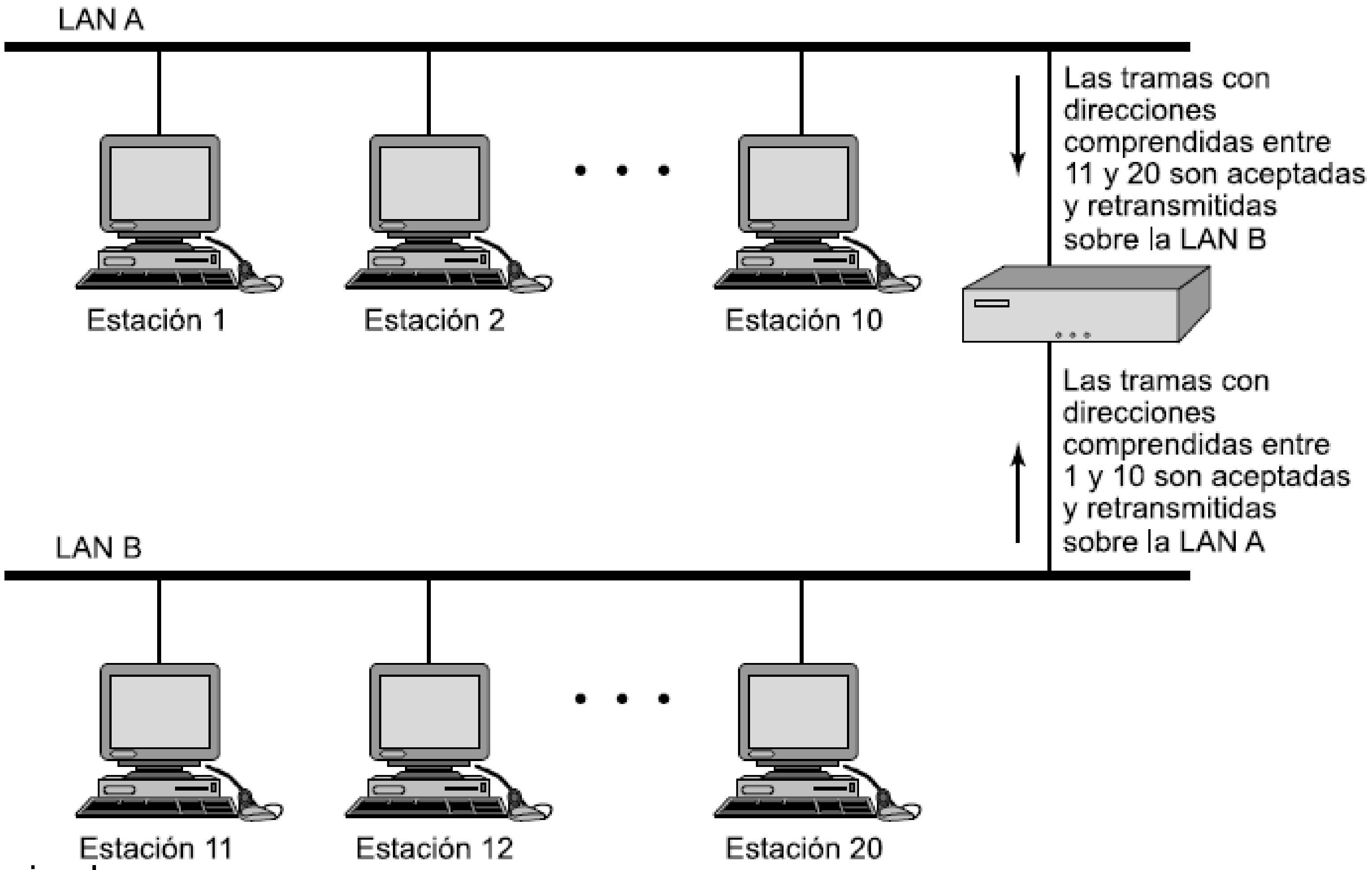


The bridge allows an extension of the LANs in such a way that it is not necessary to modify the communications software of the stations connected to them. From the point of view of each of the stations on the two (or more) LAN segments, it appears as if there is only a single LAN where each station has a unique address. Stations use that unique address and do not need to explicitly discriminate between stations on the same or different LANs; the bridge takes care of it.

The routing procedure for an incoming frame depends on the LAN it came from (the source LAN) and the LAN it is destined for (the destination LAN), as shown below:

1. If the destination and source LANs are the same, discard the frame.
2. If the destination and source LANs are different, forward the frame.
3. If the destination LAN is unknown, resort to flooding. This algorithm must be applied each time a frame arrives.

DEVICES. BRIDGE


LAN A
Estación 1
Estación 2
Estación 10
Las tramas con direcciones comprendidas entre 11 y 20 son aceptadas y retransmitidas sobre la LAN B
Las tramas con direcciones comprendidas entre 1 y 10 son aceptadas y retransmitidas sobre la LAN A
LAN B
Estación 11
Estación 12
Estación 20

The bridge functions are few and simple:

- Reading all the frames transmitted in A and accepting those addressed to stations in B (frames with a destination address in network A are discarded by the bridge)
- Retransmission to B of each of the frames, making use of the Medium Access Control protocol of this LAN.
- Same process for traffic from B to A.

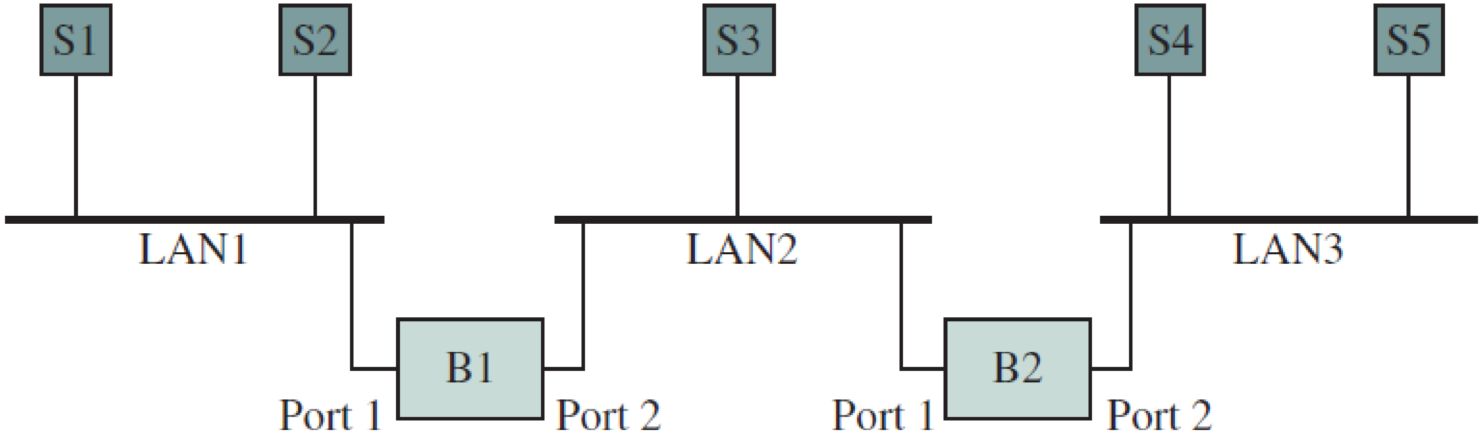
DEVICES- DESIGN ASPECTS OF A BRIDGE

It is worth highlighting several aspects of the bridge design:

- The bridge does not modify the content or format of the frames it receives nor does it encapsulate them with additional header.
- Each frame to be transmitted is simply copied from a LAN and repeated with, exactly the same bit pattern on the other LAN. This can be done this way since the two LANs use the same protocols.
- The bridge must have sufficient temporary memory to accept peak demands. For a small time period, frames can be received faster than they can be relay.
- The bridge must have addressing and routing capacity. At a minimum, you must know the addresses of each network to determine which frames should pass. Furthermore, there may be more than two LANs interconnected by several bridges, in which case it may be necessary route a frame across several bridges along its path from source to destination.
- A bridge can connect more than two LANs.

Structure of a bridge:
 - Ports
 - Memory buffers. To store frames
. - Address Table: MAC
Address/Port/Age
Learning logic.
- Search Logic. spanning tree

BRIDGE MAC LEARNING



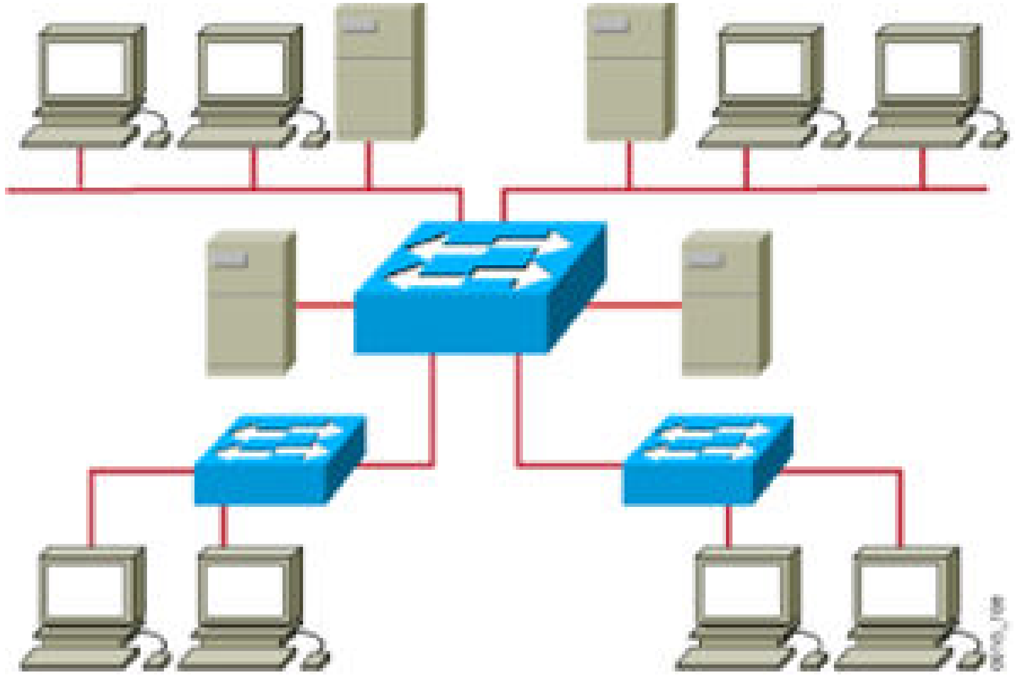
Address	Port

Address	Port

When the bridges are first connected, all tables are empty. None of the bridges know where the destinations are located, so they use a flooding algorithm: all frames arriving with an unknown destination are sent to all LANs to which the bridge is connected, except the one from which they come . Over time, bridges learn where destinations are located.

In SDN The concept of Linux Bridge will be introduced

DEVICES- SWITCH



Ethernet switches improve performance by reducing the number of devices that share the same resource within the segment. Ethernet switches also make intelligent decisions in sending frames by examining the source and destination MAC addresses of incoming frames. Ethernet switches operate at layer 2 of the OSI reference model. Due to their internal high-speed architecture and large number of ports, Ethernet switches offer high-speed flow. Greater availability than traditional bridges.

- An Ethernet switch learns the source MAC addresses of the devices it is connected to each of its ports listening for incoming traffic. Mapping MAC addresses to ports is stored in a MAC database, also known as the MAC address table.

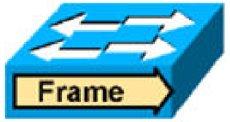
- When an Ethernet switch receives a frame, the switch consults the MAC database to determine which port the station identified as the destination of the frame can be reached. If the destination MAC address is found in the MAC database, the frame is transmitted only to the port identified as the destination of the frame. If the destination MAC address is not found in the MAC database, the frame is transmitted to all ports except the port where the frame entered.

-The Switch learns address
-Select port based on address
-Avoid loops

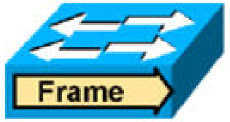
In SDN The concept of OPEN V Switch is introduced

SWITHC. FRAME TRANSMISSION

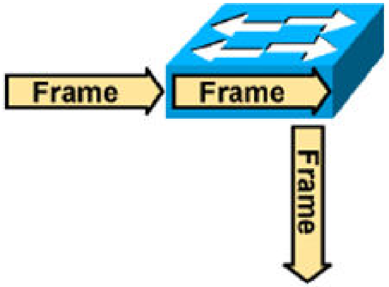
Cut–Through
 - The switch checks the address destination and immediately start sending frame

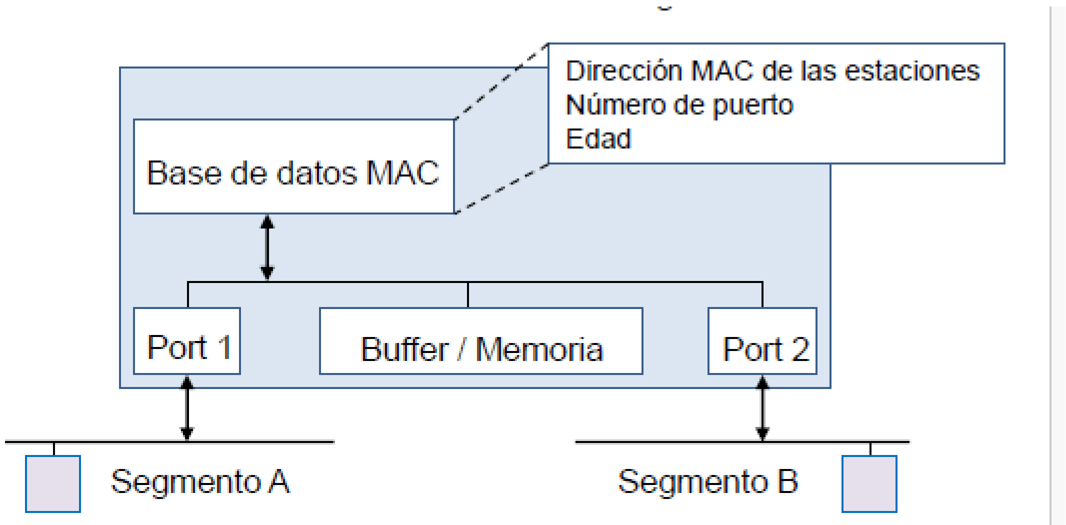


Fragment-free
 - The switch checks the first 64 bytes, then send the frame



Store and forward
 - The entire frame is received and checked before shipping

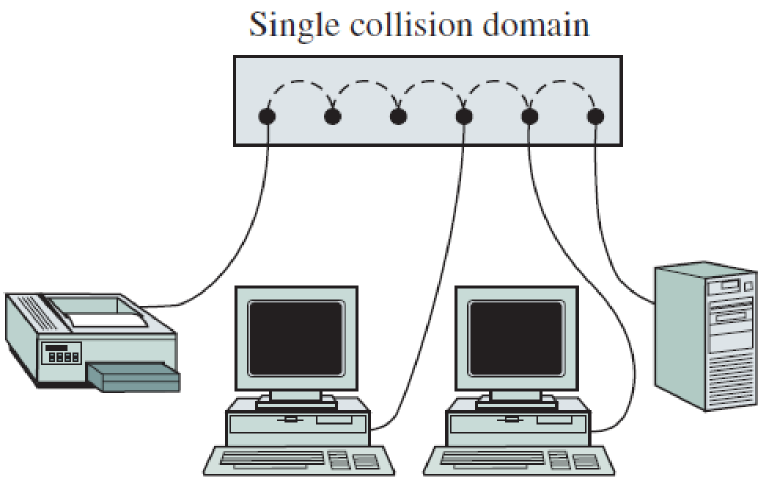


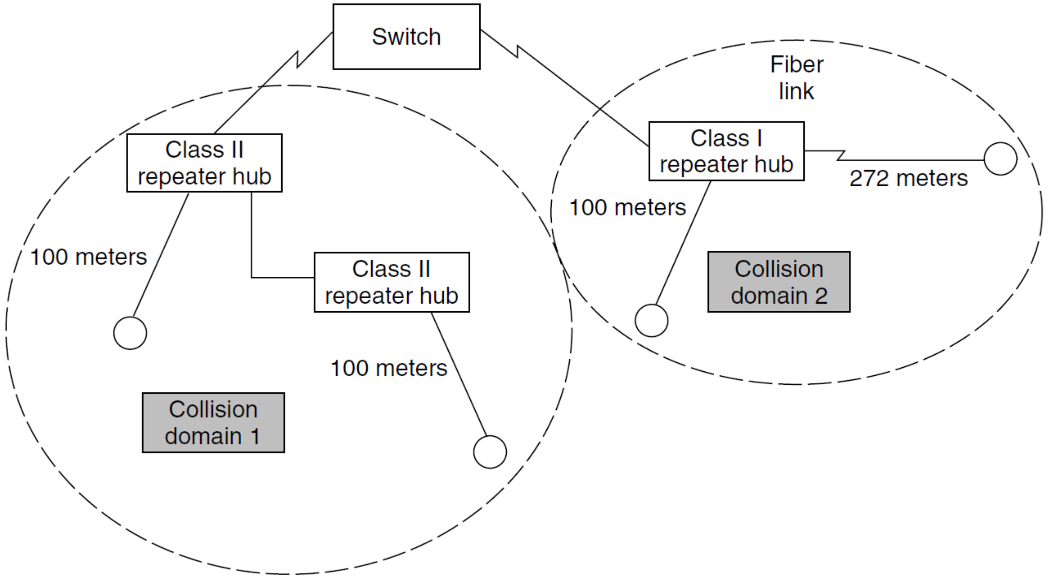


Since each switch port typically goes to only one computer, they must have room for many more line cards than bridges, which are intended only to connect LANs. Each line card provides buffer space for frames arriving at its ports. As each port is its own collision domain, the switches never lose frames due to collisions. However, if frames arrive faster than they can be retransmitted, the switch might run out of buffer space and proceed to drop frames. Switches are more sophysticated than bridges, they have more ports and some features are implemented in HW. **Both of them limit the collision domain to each port**.

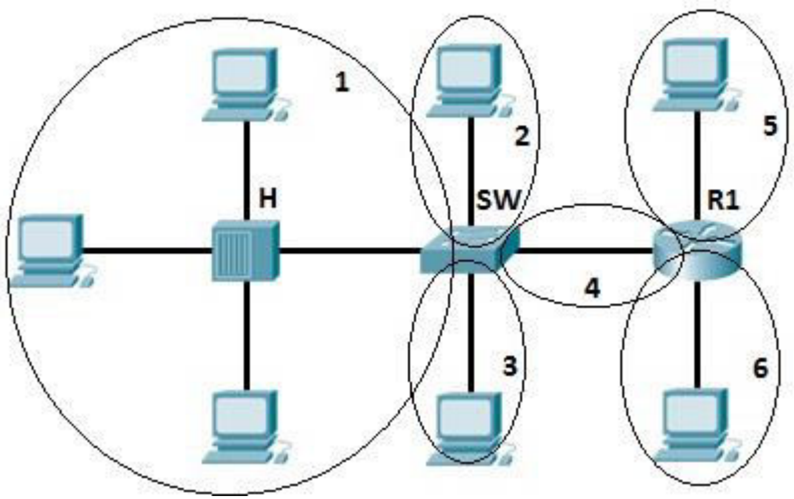
SWITCH COLLISION DOMAIN AND BROADCAST DOMAIN

Collision domain





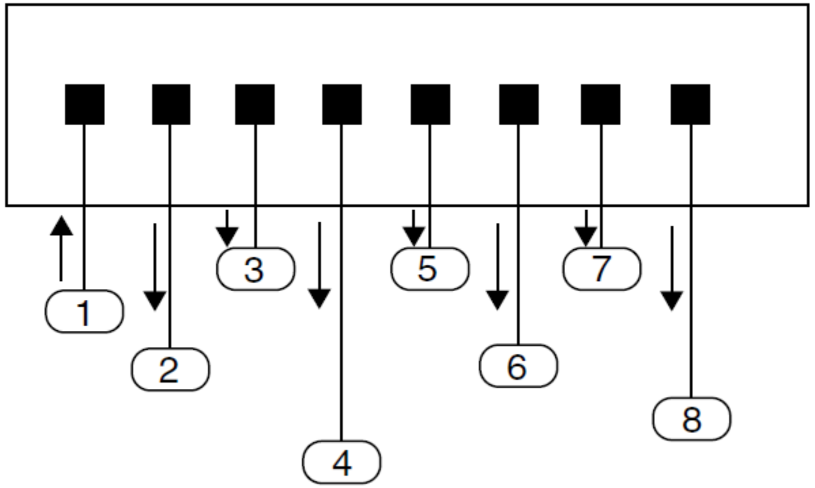
Switches isolate or divide Collision domains

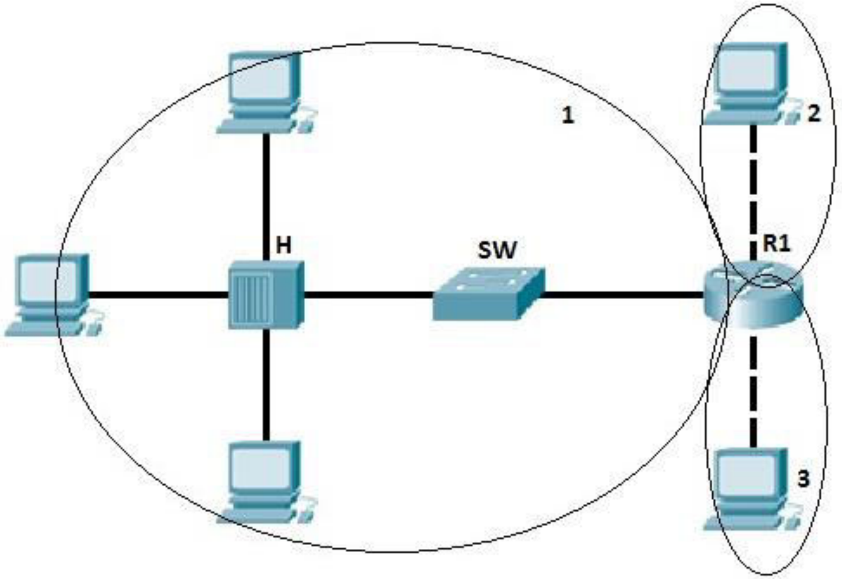


It can be extracted from this image that the hubs (device with the H) – which are obsolete – do not divide the collision domain, rather they expand it. Meanwhile, the ports of the Switches (SW) and the Routers (R1) do separate the collision domains.

SWITCH COLLISION DOMAIN AND BROADCAST DOMAIN

Broadcast domain



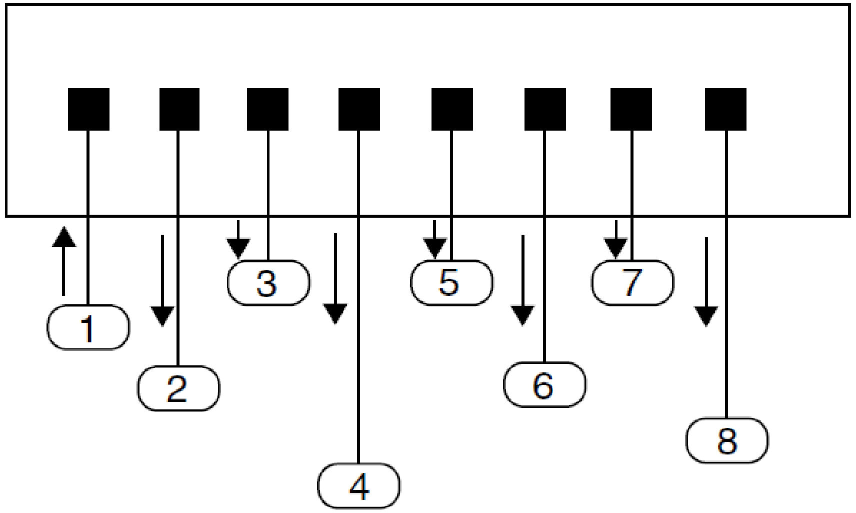


“FFFFFFFFFFFF” in hexadecimal is recognized as a broadcast address, and every station on the network will receive and accept frames with that destination address.

A broadcast domain is a logical separation within the computer network in which messages, normally layer 3 packets in the OSI model, can be broadcast - as long as they meet certain conditions - to that all devices within that space, logically defined, can receive them. The equipment that does not limit these domains, in addition to the Hub, are the switches and bridges—equipment with few ports and less 'intelligence' than the switches.

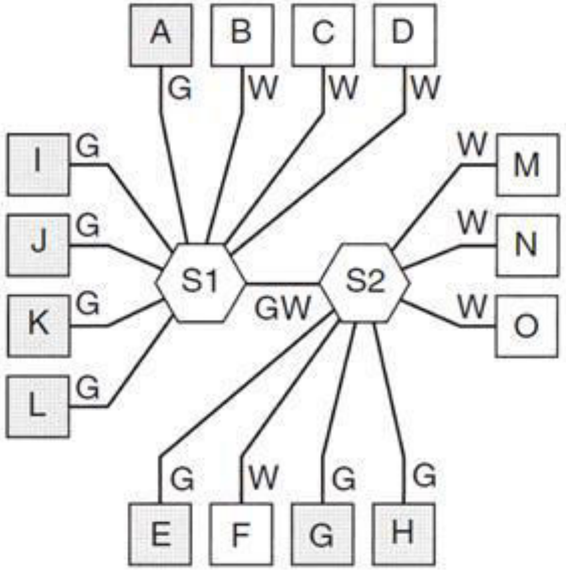
As can be seen in the figure, the equipment that limits the broadcast domains par excellence are the routers - although there are also switches that can limit this diffusion by creating VLANs. In the image you can see how the router ports are the ones that mark the division of the broadcast domains and to highlight this, as was done in the collision domain, ellipses have been used to indicate each of these broadcast domains.

VLAN



A switch, like a bridge or a hub, has a single broadcast domain.

The use of vlans allows dividing the only broadcast domain of a switch into several, partitioning logically to the switch (a broadcast domain per vlan), to the point that two machines that belong to different vlans on a switch cannot communicate with each other through it.



Broadcast domain can be separated by routers (L3) or By Switches using VLANs (L2)

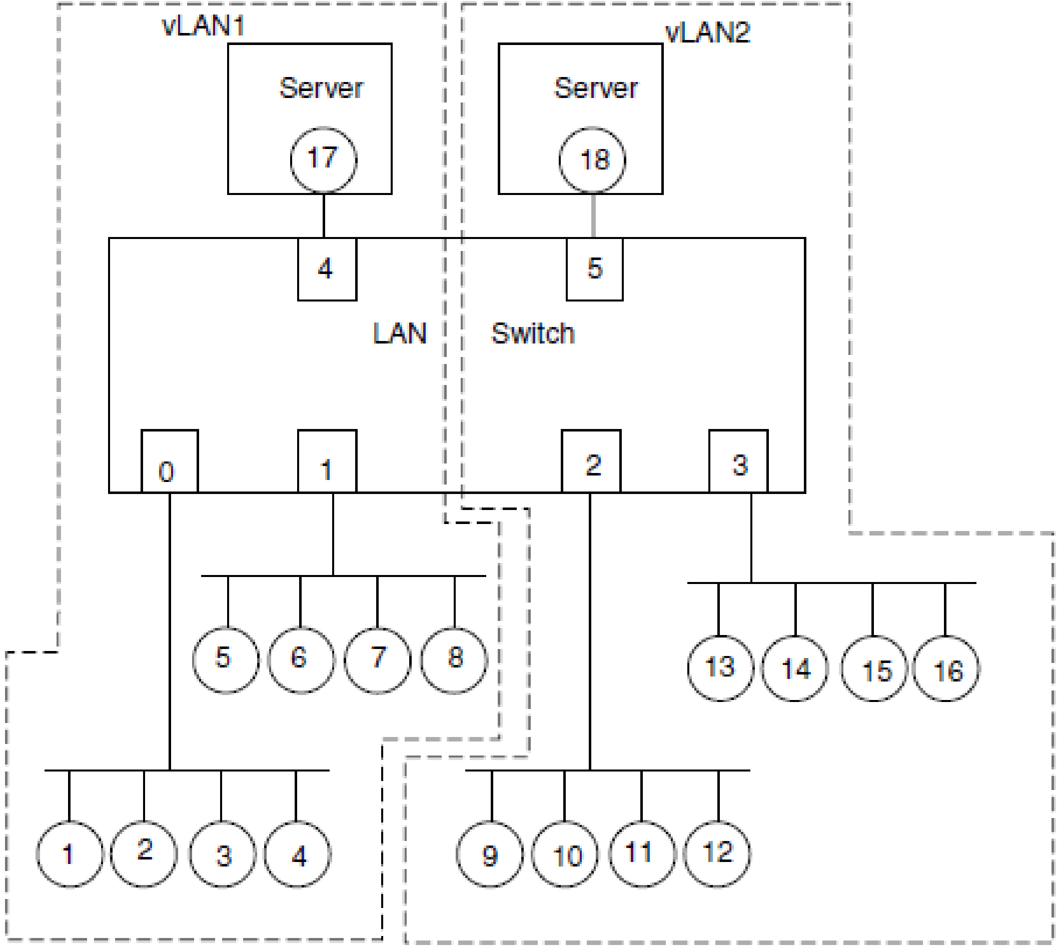
VLAN -

Using VLAN we can separate different broadcast domains within ONE switch.

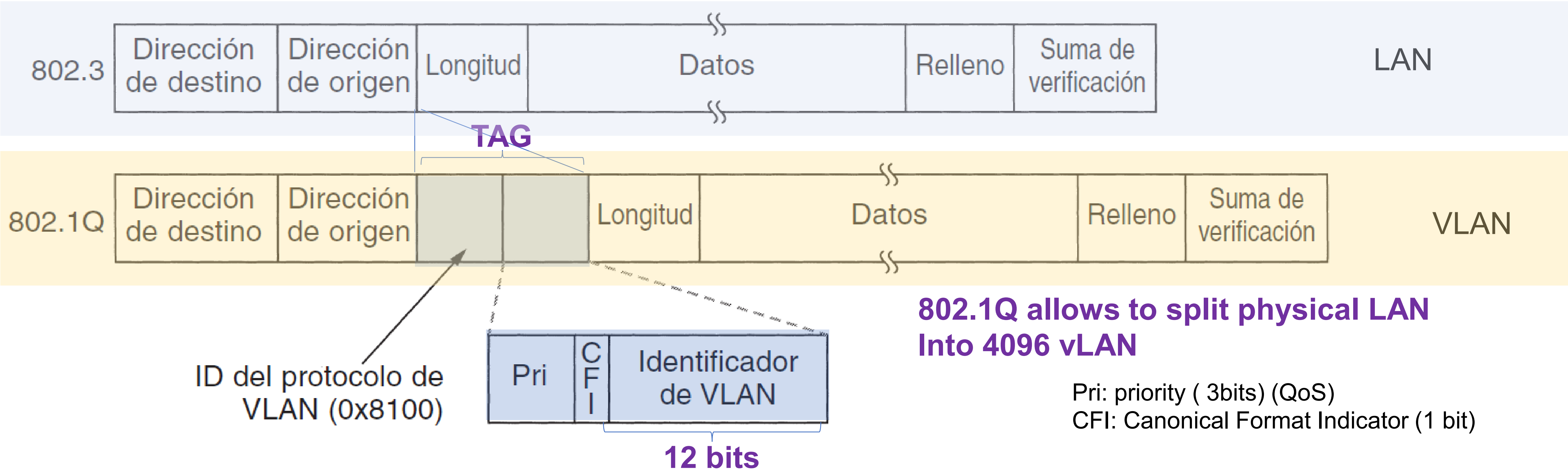
The way to separate broadcast domains, is to create vLANs.

In order to identify each vLAN within the only one LAN, is adding a TAG

The TAG has two main parts:
Protocol ID: (0x8100)
VLAN ID: 12 bits /4096 vlans



TAGGED VLAN 802.1Q



802.1Q allows to split physical LAN Into 4096 vLAN

Pri: priority (3bits) (QoS)
CFI: Canonical Format Indicator (1 bit)

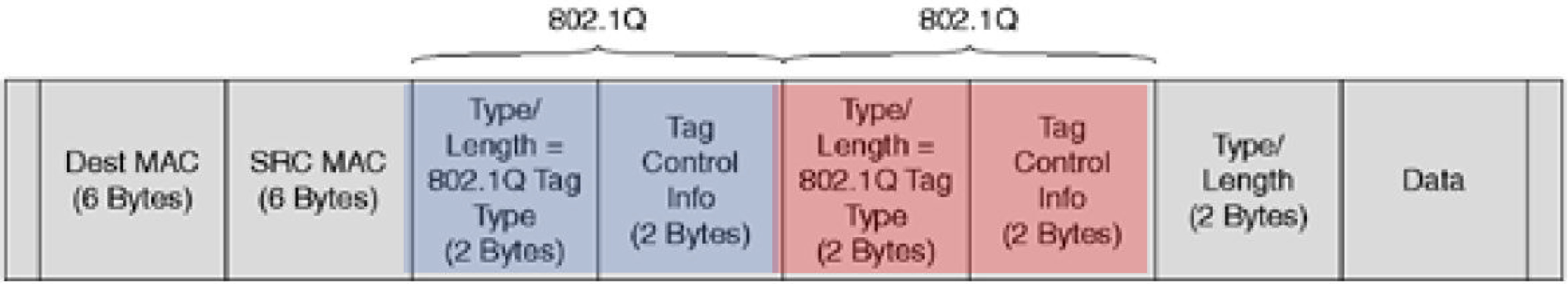
Let's now take a look at the 802.1Q frame format. The only change is the addition of a pair of two-byte fields. The first is the VLAN protocol ID. It always has the value 0x8100. Since this number is greater than 1500, all Ethernet cards interpret it as a type rather than a length. The second two-byte field (**Tag Control Information)** contains three subfields. The main one is the **VLAN Identifier,** which occupies the low order 12 bits. This is the crux of the question: which VLAN does the frame belong to? The 3-bit Priority field has absolutely nothing to do with VLANs, but with changing the Ethernet header. This field makes it possible to distinguish strict real-time traffic from soft real-time traffic and non-delay sensitive traffic, in order to offer a better quality of service over Ethernet. This is necessary for the transport of voice over Ethernet (although, to be fair, IP has had a similar field for a quarter of a century and nobody uses it).

VALN ID > 1500 --- > type

VLAN ID: 0x8100 : 1000 0001 0000 0000 : 16.384 + 1024 > 1500

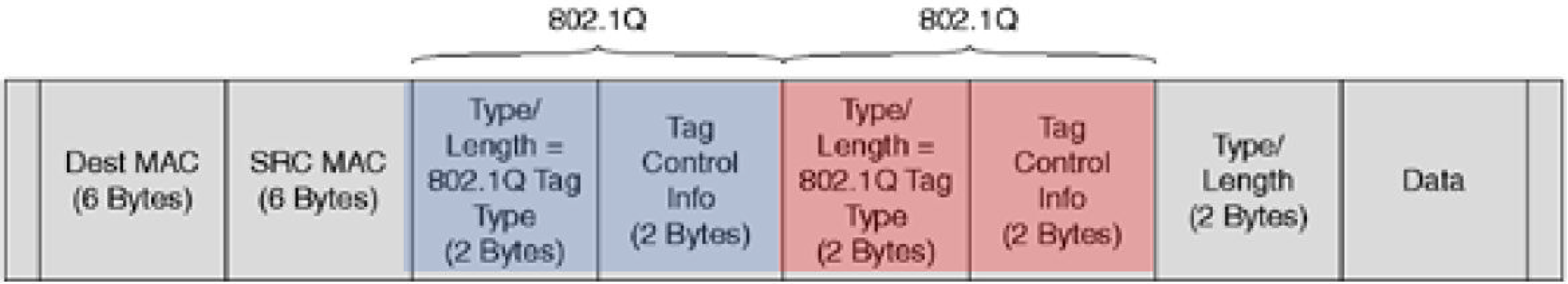
VLAN STACKS

Q in Q (Cisco)


802.1Q
802.1Q
Dest MAC (6 Bytes)
SRC MAC (6 Bytes)
Type/ Length = 802.1Q Tag Type (2 Bytes)
Tag Control Info (2 Bytes)
Type/ Length = 802.1Q Tag Type (2 Bytes)
Tag Control Info (2 Bytes)
Type/ Length (2 Bytes)
Data

8100 8100.

Standard 802.1ad


802.1Q
802.1Q
Dest MAC (6 Bytes)
SRC MAC (6 Bytes)
Type/ Length = 802.1Q Tag Type (2 Bytes)
Tag Control Info (2 Bytes)
Type/ Length = 802.1Q Tag Type (2 Bytes)
Tag Control Info (2 Bytes)
Type/ Length (2 Bytes)
Data

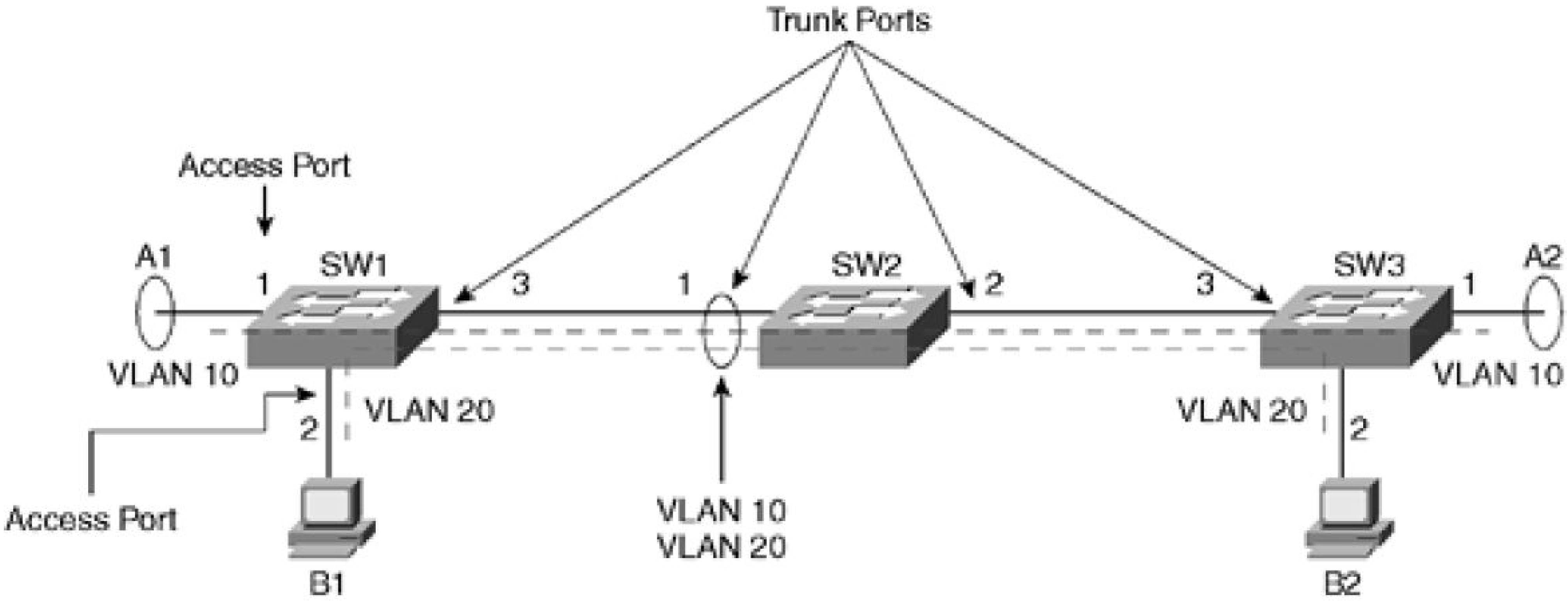
88A8 8100.

VLAN Stack (QinQ) introduces “Two” VLAN tags to be inserted into a single frame, an essential capability for implementing Metro Ethernet network topologies.

This allows customers to run their own VLANs inside service provider's provided VLAN. This way **the service provider can just configure one VLAN for the customer and customer can then treat that VLAN as if it were a trunk**.

In SDN, VXLAN will be introduced later

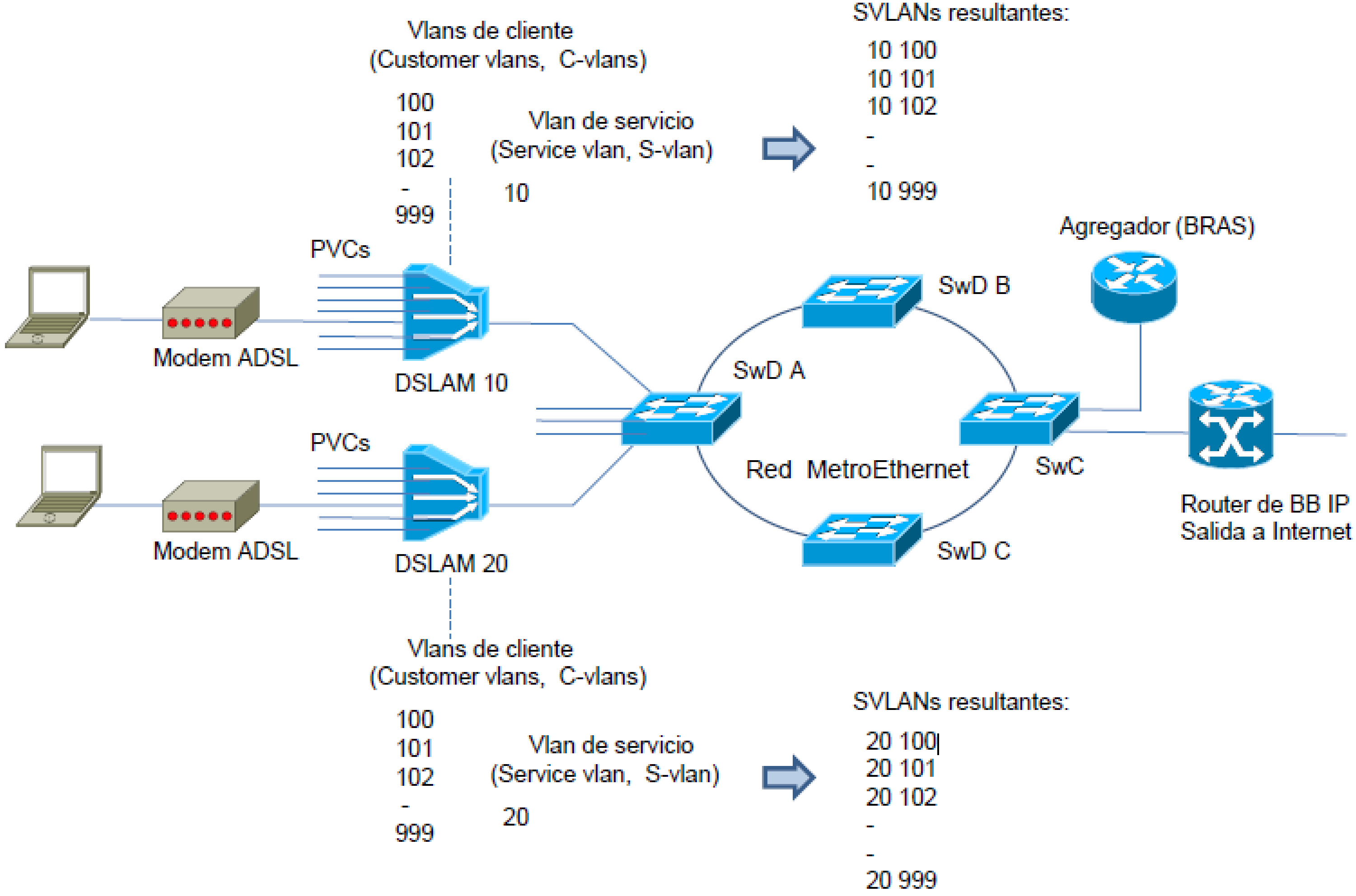
USE OF VLANS



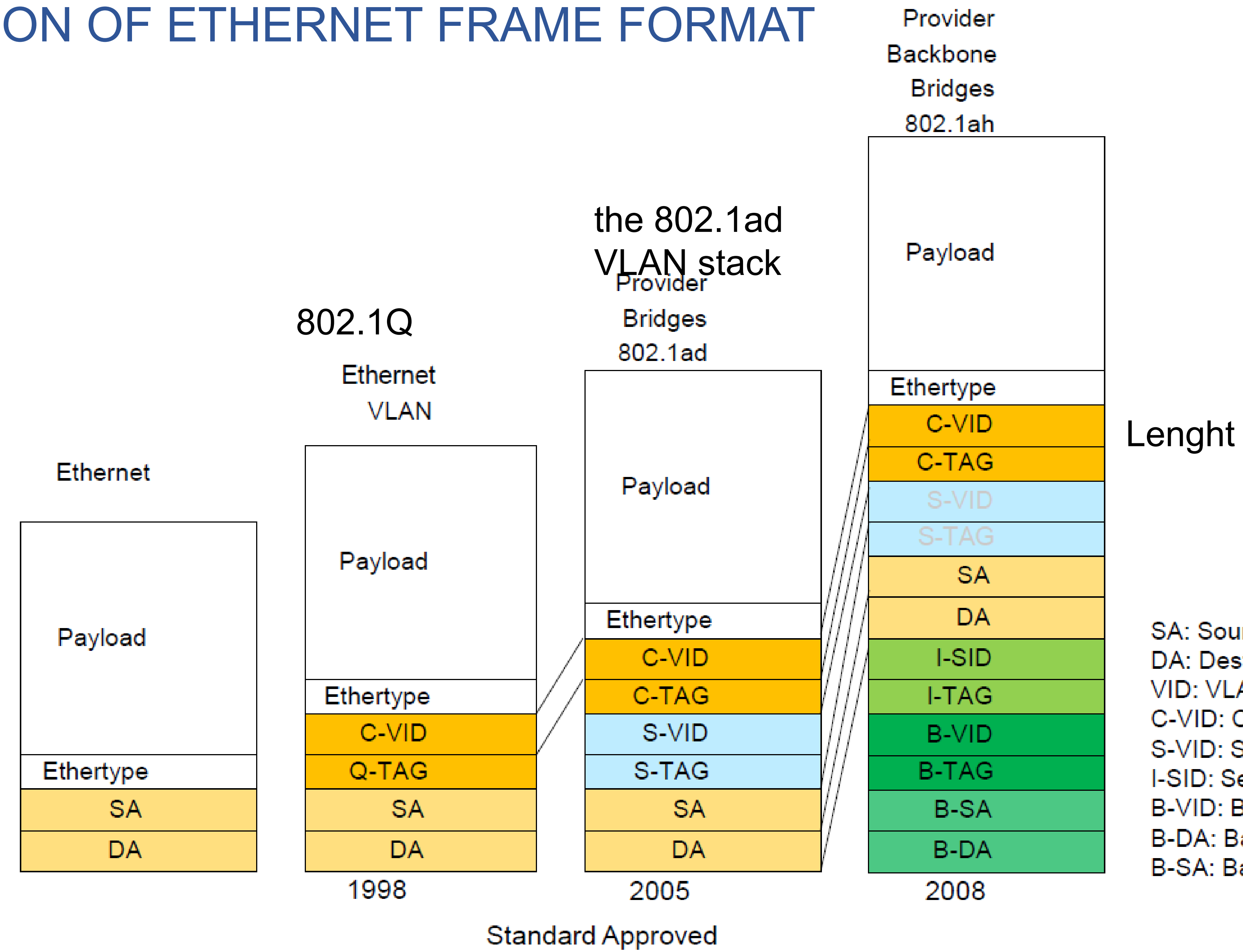
Use of vlans: type of ports in a switch

Access ports
Trunk ports
Type dot.1Q ports (ports with capacity for adding de second tag)

USE OF VLANS


Vlans de cliente
(Customer vlans, C-vlans)
100
101
102
-
999
Vlan de servicio
(Service vlan, S-vlan)
10
SVLANs resultantes:
10 100
10 101
10 102
-
-
10 999
Agregador (BRAS)
PVCs
SwD B
Modem ADSL
DSLAM 10
SwD A
PVCs
Red MetroEthernet
SwC
Router de BB IP
Salida a Internet
Modem ADSL
DSLAM 20
SwD C
Vlans de cliente
(Customer vlans, C-vlans)
100
101
102
-
999
Vlan de servicio
(Service vlan, S-vlan)
20
SVLANs resultantes:
20 100
20 101
20 102
-
-
20 999

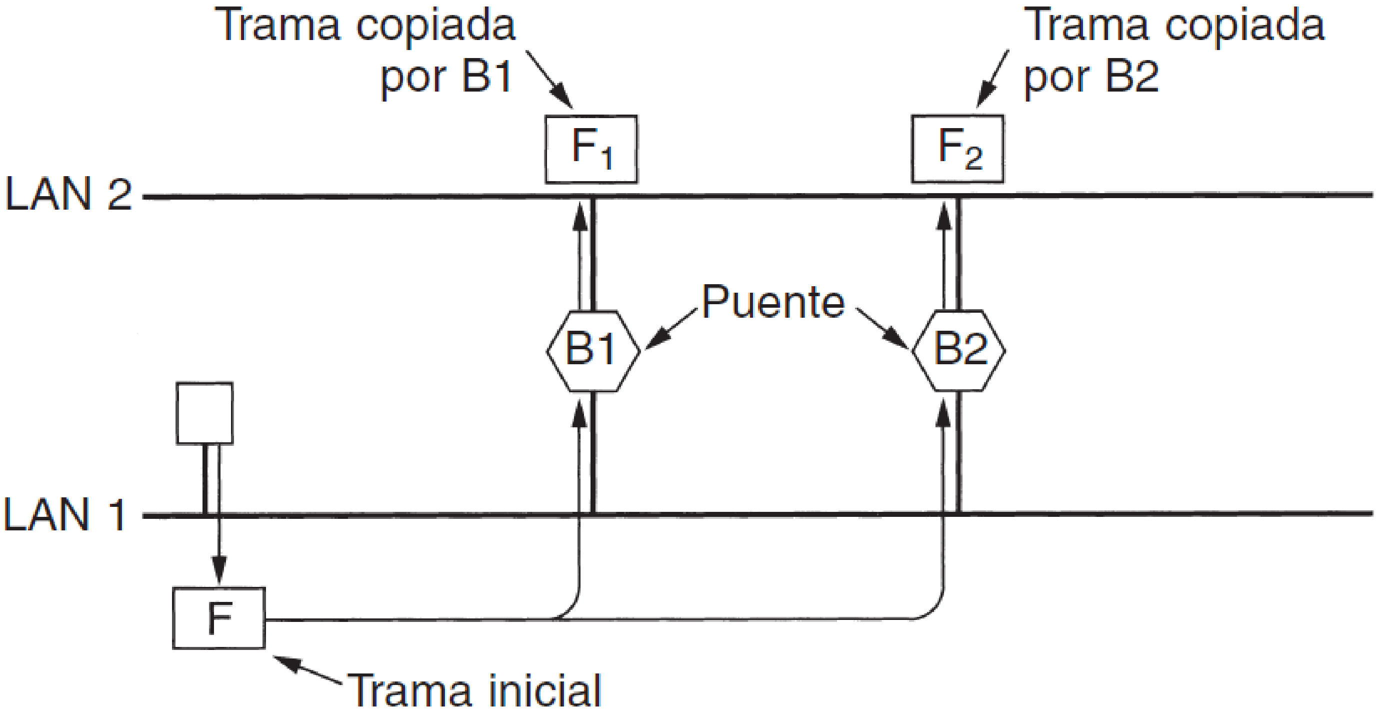
EVOLUTION OF ETHERNET FRAME FORMAT



SA: Source MAC Address
DA: Destination MAC Address
VID: VLAN ID
C-VID: Customer VLAN ID
S-VID: Service VID
I-SID: Service ID
B-VID: Backbone VID
B-DA: Backbone DA
B-SA: Backbone SA

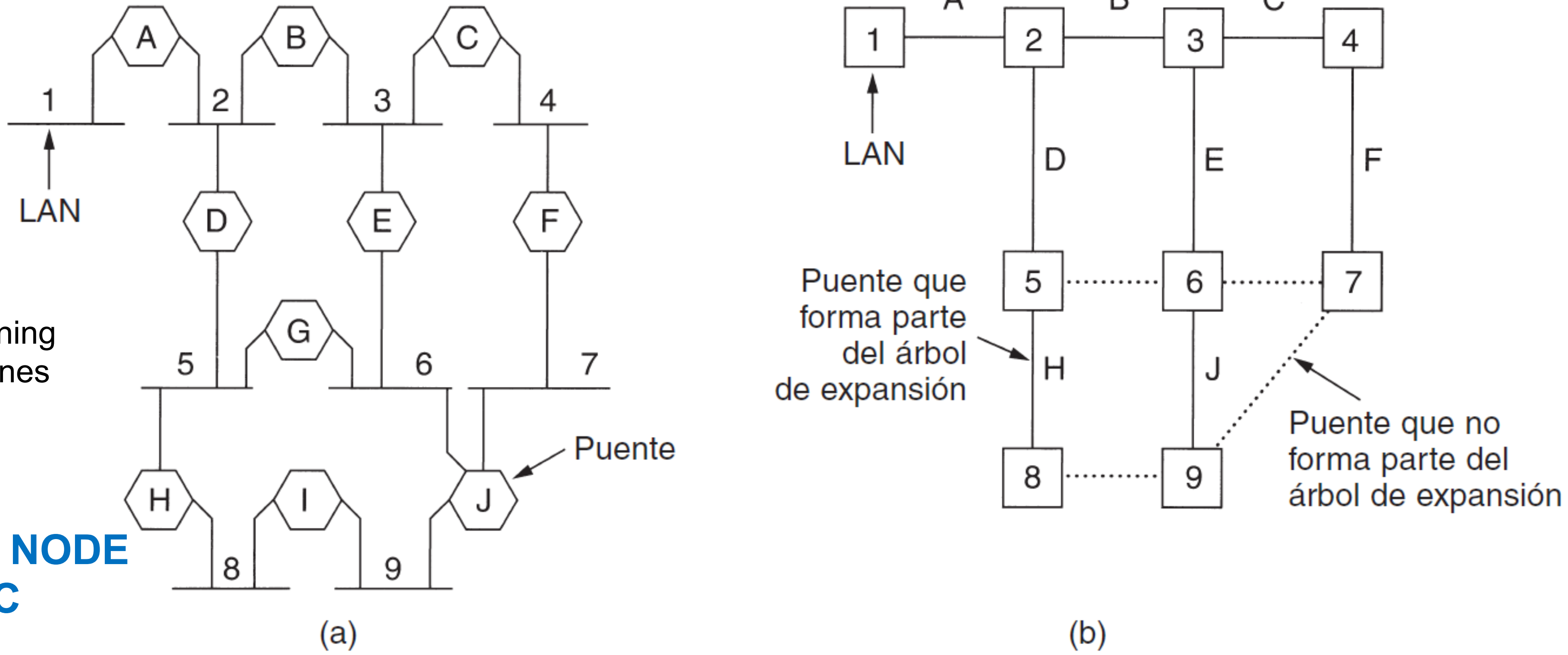
SPANNING TREE PROTOCOL

To increase reliability, some sites use two or more bridges in parallel between pairs of LANs, as shown in the following figure. However, this arrangement also generates some additional problems because it produces loops in the topology. We have a simple example of these problems when observing how a frame, F, with an unknown destination, is handled in the following figure. Each bridge, following the normal rules for handling unknown destinations, resorts to flooding, which in this example is just copying the frame to LAN 2. Shortly after, bridge 1 detects at F2, a frame with an unknown destination, and copies it to LAN 1, which generates F3 (not shown). Similarly, bridge 2 copies F1 to LAN 1 and outputs F4 (also not shown). Bridge 1 now forwards to F4 and bridge 2 copies to F3. This cycle is repeated again and again.



SPANNING TREE PROTOCOL

The solution to this problem is for the bridges to communicate with each other and cover the existing topology with a spanning tree that reaches all LANs. In reality, some potential connections between LANs are ignored in the effort to construct a fictitious loop-free topology. In the following figure part (a) we see nine LANs interconnected by ten bridges. This configuration can be abstracted into a graph with the LANs as nodes. An arc connects two LANs that are linked by a bridge. The graph can be reduced to a spanning tree by removing the arcs shown as dashed lines in figure part (b).



A
B
C
1
2
3
4
LAN
D
E
F
G
5
6
7
Puente
H
I
J
8
9
(a)
A
B
C
1
2
3
4
LAN
D
E
F
5
6
7
Puente que forma parte del árbol de expansión
H
J
8
9
Puente que no forma parte del árbol de expansión
(b)

(a) Interconnected LANs. (b) Spanning tree spanning LANs. The dashed lines are not part of the spanning tree.

LAN segments become a NODE
BRIDGES become an ARC

SPANNING TREE PROTOCOL

Spanning Tree Protocol, fundamentals

With this spanning tree there is exactly one route from each LAN to the other LANs. Once the bridges agree on the spanning tree, all forwarding between LANs is done through the spanning tree. Since there is only one route from each source to each destination, it is impossible for loops to occur.

Spanning tree technique

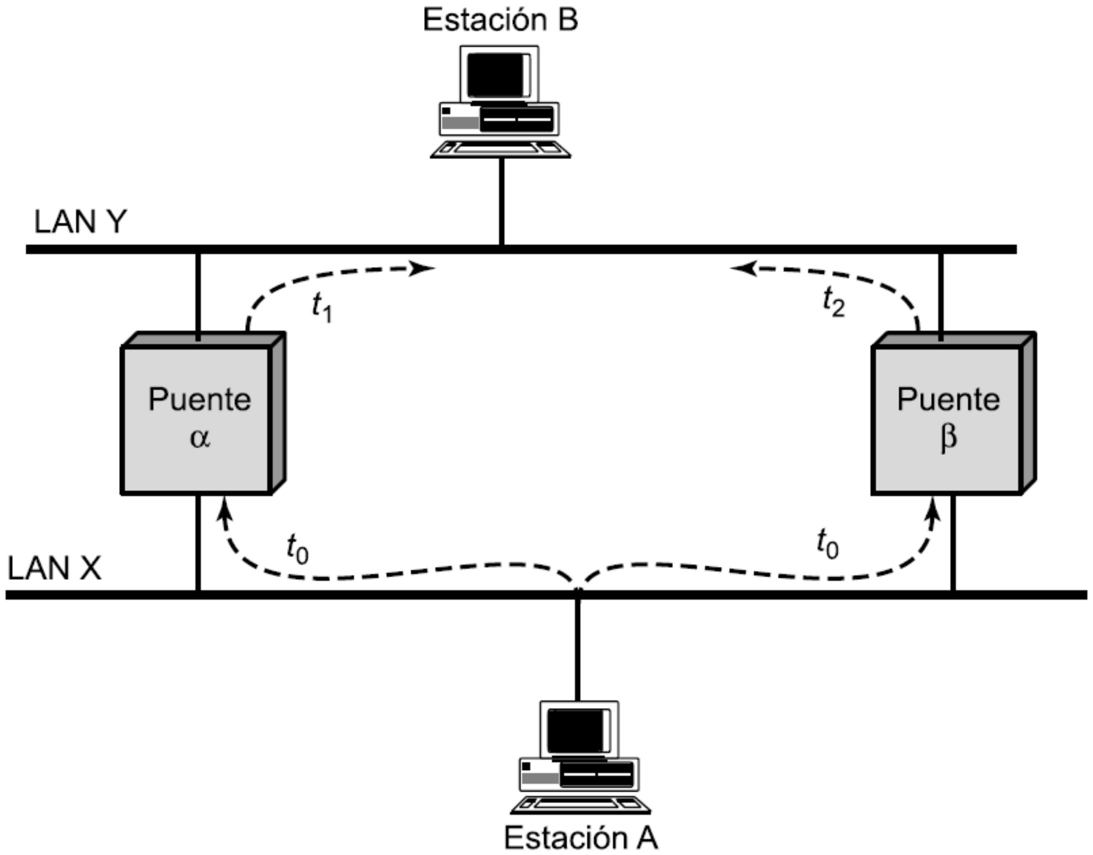
The spanning tree method is a mechanism in which bridges automatically develop a routing table and update it in response to changes in the topology. The algorithm consists of three mechanisms: frame retransmission, address learning, and loop avoidance mechanism. The first two mechanisms were already analyzed previously. We will now analyze the algorithm to avoid loops. With this spanning tree there is exactly one route from each LAN to the other LANs. Once the bridges agree on the spanning tree, all forwarding between LANs is done through the spanning tree. Since there is only one route from each source to each destination, it is impossible for loops to occur.

The result of this algorithm is that a unique route is established from each LAN to the root of the tree and therefore, to all other LANs. Although the tree covers all LANs, they are not necessarily all bridges in the tree present (to avoid loops). Even after it has been established the spanning tree, the algorithm continues to operate to automatically detect topology changes and update the tree. The distributed algorithm used to construct the spanning tree was invented by Radia Perlman and is described in detail in (Perlman, 2000). It was standardized in IEEE 802.1D. The purpose of STP is to keep the network topology free of closed loop.

SPANNING TREE PROTOCOL

Spanning tree algorithm

The address learning mechanism described above is effective if the topology of the networking is a tree; that is, if there are no alternative routes on the network. The existence of alternative routes implies the appearance of closed loops. To analyze the problem created by the existence of a closed loop, consider the following figure. Station A transmits a frame destined for station B at time t0.



Both bridges capture this frame and update their databases to indicate that station A is in the direction of LAN X, and retransmit the frame over LAN Y. Suppose that bridge α retransmits it at time t1 and the β bridge a little later, at t2.Thus, B will receive two copies of the frame. In addition, each bridge will receive the transmissions of the others through LAN Y. Note that each transmission is a MAC frame with the source address of A and the destination address of B, so each bridge will update its database to indicate that station A is located in the direction of LAN Y. No bridge is now capable of relaying a frame addressed to station A.

SPANNING TREE PROTOCOL

Spanning tree algorithm

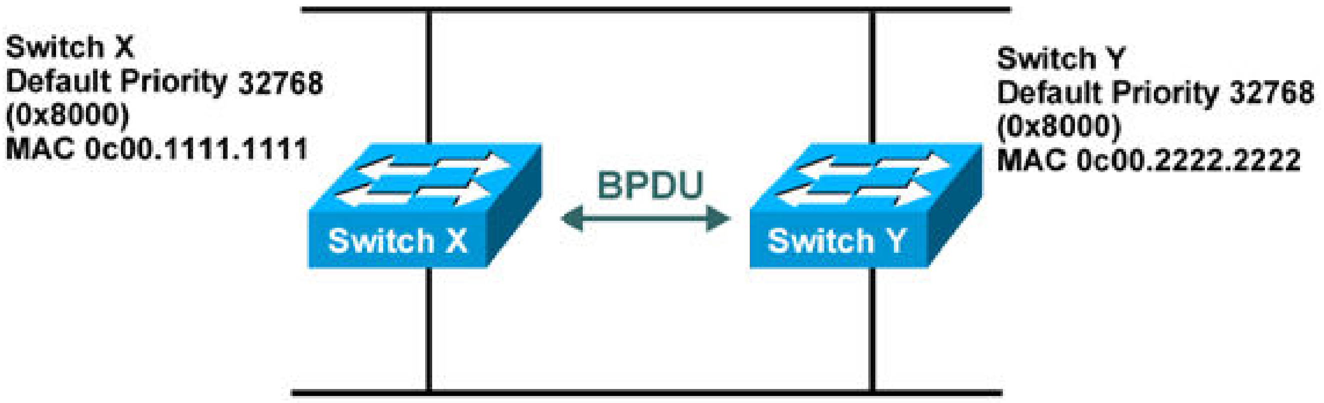
For any connected graph, composed of nodes and terminals connecting each pair of nodes, there exists a spanning tree of terminals that maintains the connectivity of the graph but does not contain closed loops. In terms of interconnection, each LAN corresponds to a node of the graph and each bridge to an edge. It is desirable to develop a simple algorithm through which the interconnection bridges can exchange sufficient information (without user intervention) to obtain the spanning tree. The algorithm must be dynamic; That is, bridges must be able to notice a change in topology and automatically obtain a new spanning tree.The spanning tree algorithm developed by IEEE 802.1, as its name suggests, can develop such a spanning tree.

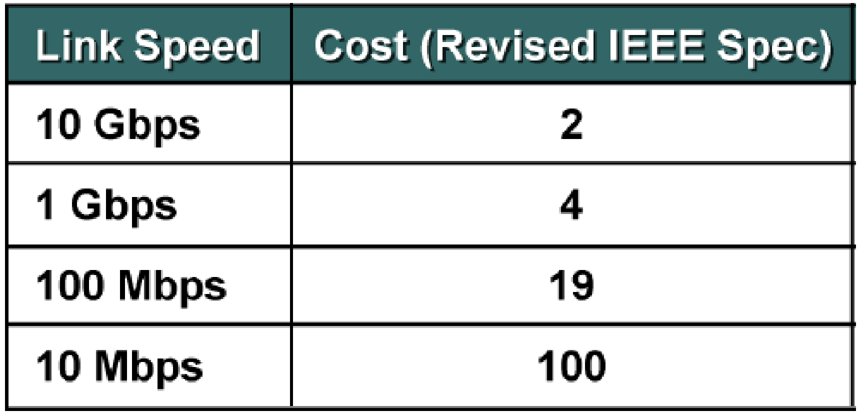
All that is required is that each of the bridges be assigned a unique identifier and costs that are associated with each of the bridge ports. Apart from any special considerations, all costs could be equal, which would produce a tree with fewer hops. The algorithm involves exchanging a small number of messages between all bridges to obtain the minimum cost spanning tree. When a topology change occurs, the bridges will automatically recalculate the spanning tree.

SPANNING TREE PROTOCOL

Operation Spanning-Tree:

There are three steps that STP performs to converge to a closed-loop free topology.

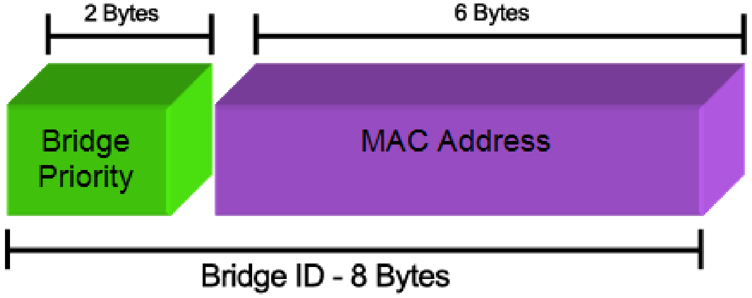




Link Speed	Cost (Revised IEEE Spec)
10 Gbps	2
1 Gbps	4
100 Mbps	19
10 Mbps	100

1. Election of root bridge: STP has a process to elect a root bridge. Only a bridge can act as root bridge in a network (broadcast domain). On the root bridge, all ports are in designated ports mode. The designated ports are normally in forwarding status. When a port is in the forwarding state, a port can send and receive traffic. Switches and bridges exchange messages with other switches and bridges at regular intervals (every two seconds by default). BPDUs (bridge protocol data unit) are exchanged. The exchange is done using multicast. One of the information included in the BPDU is the Bridge ID (BID)

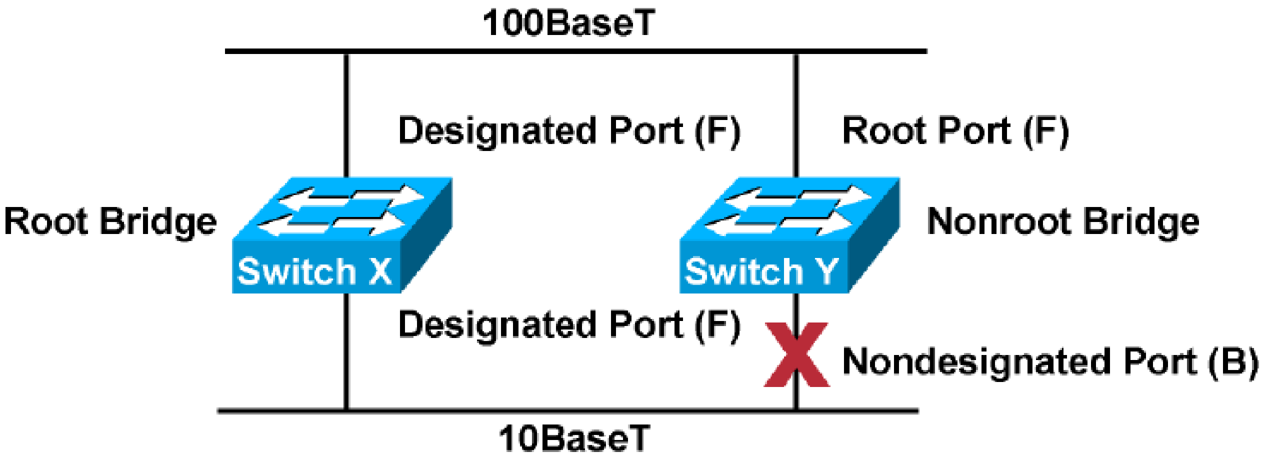
STP uses a single value to identify each switch or bridge, BID. Typically, the BID is composed by the priority value (two bytes) and the MAC address of the bridge (six bytes). The default priority, for IEEE 802.1d, is 32,768 (1000 0000 0000 0000 in binary or 0x8000 in hexa decimal), which is the average value.



SPANNING TREE PROTOCOL

The root bridge is the one with the lowest BID. According to the values that appear in the figure switch X will be chosen as the root bridge.

Switch X
Default Priority 32768
(0x8000)
MAC: 1111.1111.1111

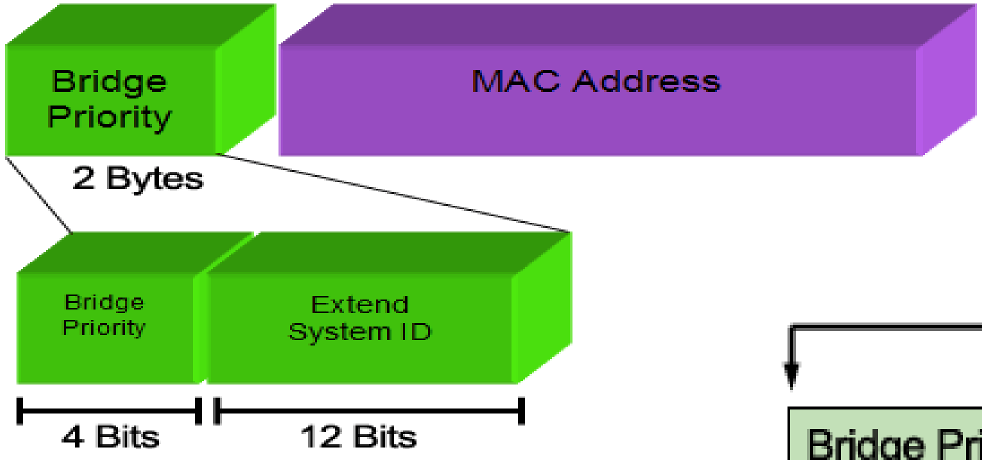


Switch Y
Default Priority 32768
(0x8000)
MAC: 2222.2222.2222

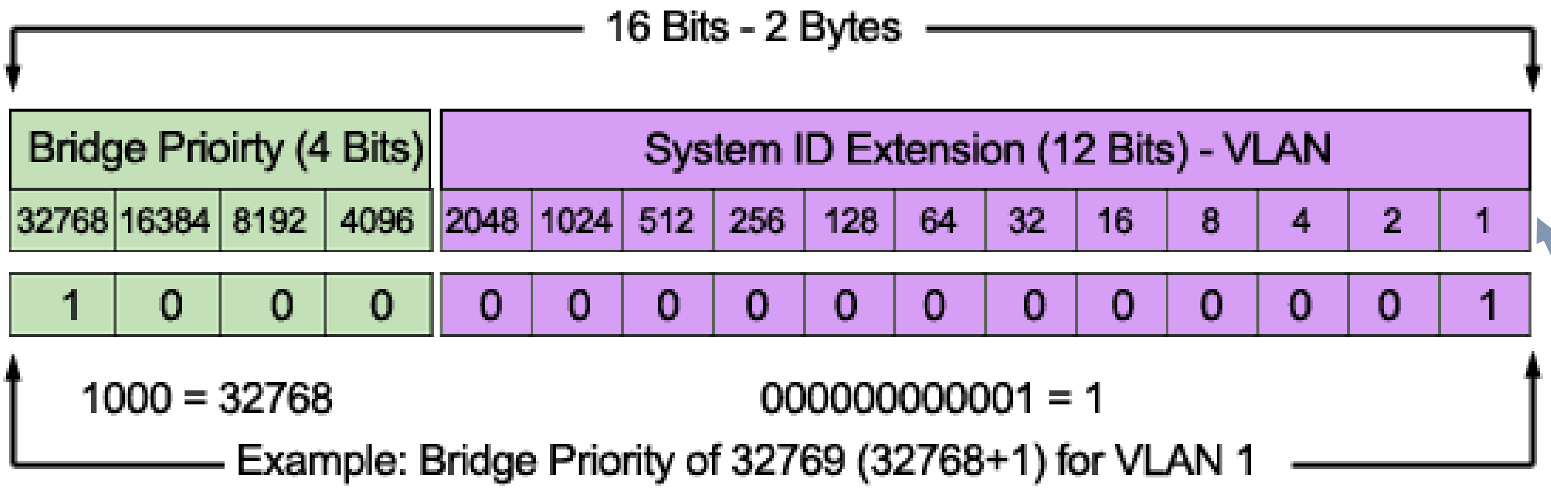
The bridge id above assumes that there is a single STP instance for the entire network. This is also called CST (Common Spanning Tree)

SPANNING TREE PROTOCOL

As networks began to grow and become more complex, VLANs were introduced. This allowed the creation of multiple logical and physical networks. It was necessary to run multiple instances of STP to accommodate each network - VLAN. These multiple instances are called Multiple Spanning Tree (MST), Per-VLAN Spanning Tree (PVST) and Per-VLAN Spanning Tree Plus (PVST+). To accommodate the additional VLAN information, the extended system ID field was introduced, taking 12 bits of the original bridge priority:



The bridge priority value and the extended system ID extension together constitute a 16-bit (2-byte) value. The priority of the bridge that makes up the left majority of the bits is a value from 0 to 61440.The extended system ID is a value from 1 to 4095 that corresponds to the respective VLAN participating in STP. The bridge priority increases in blocks of 4096 to allow the System ID Extension to enter each increment.



16 Bits - 2 Bytes

Bridge Prioirty (4 Bits)				System ID Extension (12 Bits) - VLAN											
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

1000 = 32768

000000000001 = 1

Example: Bridge Priority of 32769 (32768+1) for VLAN 1

The amount of priority levels is reduced to 16

The extended ID is new concept

SPANNING TREE PROTOCOL

The election process uses several STP messages sent between bridges (switches) that help each bridge decide who is the root bridge. These messages are called Hello BPDU where BPDU stands for Bridge Protocol Data Unit. It is important to understand the information these BPDUs carry as it will help to understand the election process itself. Each BPDU contains several fields. The following table defines each field:

Field	Description
Root Bridge ID o Root BID	BID of the bridge that the issuer of this BPDU believes to be the root bridge
Emmiter Brigde ID	BID of the bridge that sends this Hello BPDU
Root bridge cost	The STP cost between this bridge and the current root
Timer values at the root Bridge	Hello Timer, Max Age Timer, Forward Delay Timer

Now, the selection process itself is very simple. The bridge with the lowest BID becomes the root bridge. Since the BID starts with the Bridge Priority field, basically the bridge with the lowest Bridge Priority field becomes the root bridge. If there is a tie between two bridges that have the same priority value, the switch with the lower MAC address becomes the root bridge.

The STP root bridge election process begins when each bridge announces itself as the root bridge and constructs the Hello BPDU accordingly. Then, each bridge lists its own BID as the Root BID. The bridge id of the issuer is of course the same as that of the root BID, since it is again its own BID. Within the BPDU, the Cost field is listed with a value of 0, because there is no cost involved. The bridges send the Hello BPDU constructed as above, to the network. They will continue to maintain their status as Root Bridge by default, until they receive a Hello BPDU carrying a lower BID. This Hello BPDU is then converted to a higher BPDU.

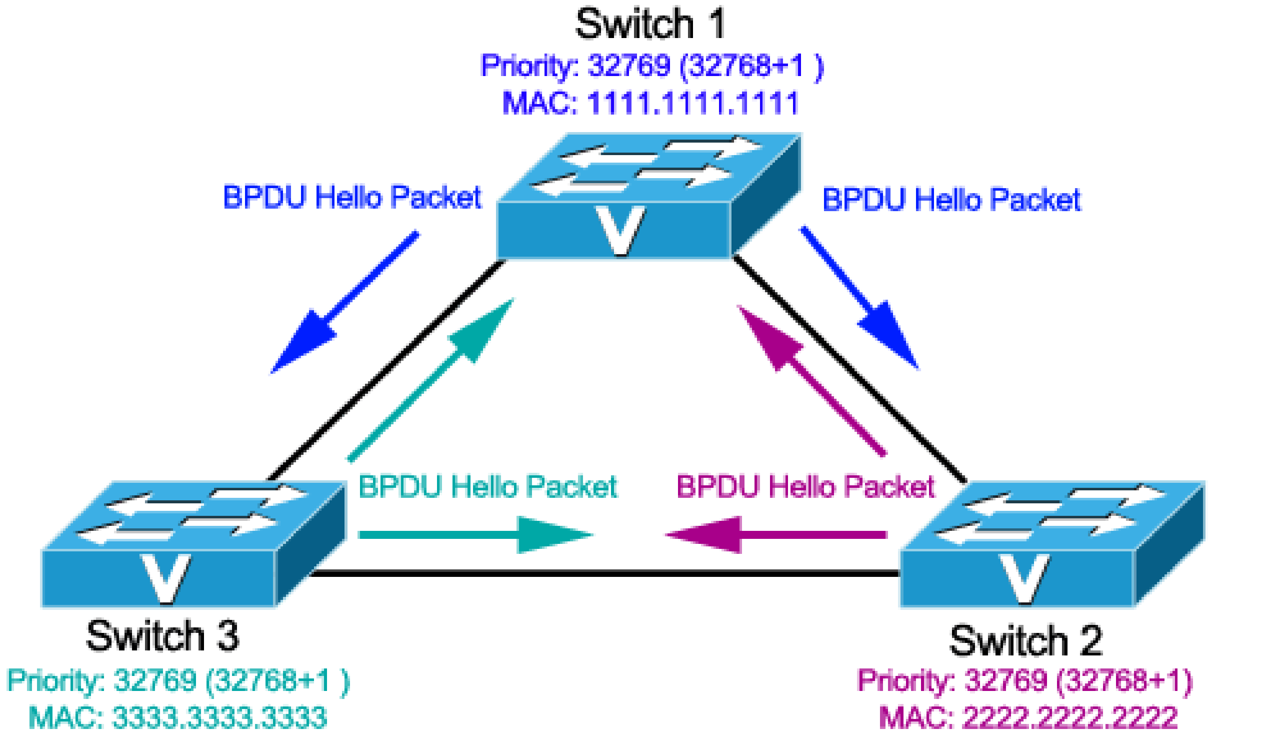
SPANNING TREE PROTOCOL

The bridge receiving this top BPDU makes changes to the Hello BPDU it has been sending. Changes the root BID value to reflect the root BID of the top Hello BPDU. This process continues until each change agrees with which bridge has the lowest BID, and therefore deserves to be the root bridge. Example of choosing the root bridgeLet's look at this process using a combination of three switches within a network. For the sake of simplicity, the MAC address of each switch has been changed to a simple value:

SW 1. It has a priority of 32769 and a MAC address of 1111.1111.1111. So your BID becomes 32769.1111.1111.1111. When SW1 creates its own BPDU, it sets BID and Root BID to 32769.1111.1111.1111.

SW2. It has a priority of 32769 and a MAC address of 2222.2222.2222. So your BID becomes 32769.2222.2222.2222. When SW2 creates its own BPDU, it sets BID and Root BID to 32769.2222.2222.2222.

SW3. It has a priority of 32769 and a MAC address of 3333.3333.3333. So your BID becomes 32769.3333.3333.3333.



Switch 1 BPDU Hello Packet
BID=Sw1 Priority+MAC
Root BID= Sw1 Priority+MAC
Root Path Cost=0

Switch 2 BPDU Hello Packet
BID=Sw2 Priority+MAC
Root BID= Sw2 Priority+MAC
Root Path Cost=0

Switch 3 BPDU Hello Packet
BID=Sw3 Priority+MAC
Root BID= Sw3 Priority+MAC
Root Path Cost=0

SPANNING TREE PROTOCOL

At this point, all switches have mutually received BPDUs and have agreed that SW1 has the lowest BID value and is therefore the legitimate root bridge on the network. Both SW2 and SW3 now agree that SW1 is the Root Bridge and begin organizing their respective links on the root ports and designated ports. STP operation continues as follows:

2. Root port selection on a non-root bridge:
STP chooses a root port on a non-root bridge. The root port is the one with the lowest cost path to the root bridge. The root ports are normally in the forwarding state. The spanning.-tree cost is the cumulative cost calculated over the bandwidth. In the figure, the least cost path to the root bridge is from the Y switch over the 100BaseT Fast Ethernet link.

3. Selection of the designated port in each segment:
In each segment STP establishes a designated port. The designated port is selected on the bridge for that port that has the lowest cost path to the root bridge. The designated ports are normally in the forwarding state. In the figure, there is a designated port for both segments on the root bridge because it is connected to both segments. The 10BaseT Ethernet port on the Y switch is a non-designated port because there is only one designated port per segment. Undesignated ports are normally in locked mode to logically break the topology loop. When a port is in blocking mode, it does not send traffic but can receive BPDUs.

Finally, all root ports and all designated ports are placed in a forwarding state. These are the only ports that are enabled to forward frames. The other ports are placed in blocking state.

SPANNING TREE PROTOCOL: States that ports pass through and default times

With STP, ports go through these four states:

- Blocking
- Listening
- Learning
- Forwarding

When STP is enabled, each bridge on the network goes from the blocking state through the listening and learning states upon initialization. With proper configuration, ports are stabilized in the sending or blocking state. Ports in forward mode provide the lowest cost to the root bridge. During a topology change, a port temporarily transitions between listening and learning states.

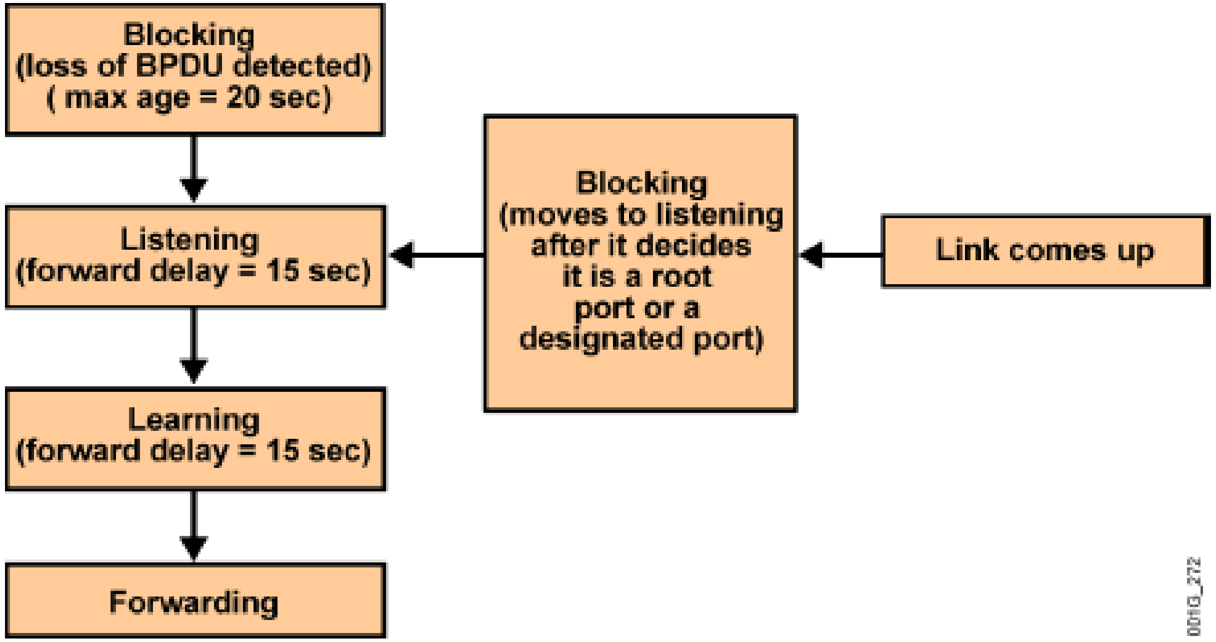
Initially all bridge ports start from the blocking state, from which they listen for BPDUs.
When a bridge starts up, this bridge thinks it is the root bridge and goes into listening state.
An absence of BPDUs for a certain period of time is called max_age, which defaults to 20 seconds. If a port is in blocking mode and does not receive a new BPDU within the max_age, the bridge goes from blocking state to listening state. When a port is in the listen state, can send and receive BPDUs to determine the active topology. During the listen state, the bridge follows these steps:

- Select the root bridge.
- Select the root port on non-root bridges.
- Select the designated ports in each segment.

The time it takes for a port to go from the listen state to the learn state or from the learn state to the forward state is called forward delay. The shipping delay has a value of 15 seconds.

SPANNING TREE PROTOCOL

The learning state reduces data flooding when you start sending. If a port is a port designated or root at the end of learning state, the port enters the sending state. In the sending state, a port can send and receive data. Ports that are not designated or root port go into blocking state. Normally transitions from the learning state to the sending state are in 30 to 50 seconds.



Blocking (loss of BPDU detected) (max age = 20 sec)

Listening (forward delay = 15 sec)

Learning (forward delay = 15 sec)

Forwarding

Blocking (moves to listening after it decides it is a root port or a designated port)

Link comes up

The spanning-tree timers can be modified to adjust the timing, but these timers should not be left at the default values. The default values are set to give the network enough time to gather all the correct information about the network topology.

-STP:802.1d: Convergence time of around 50 sec. Single instance for all vlans

-PVSTP: Cisco Proprietary One STP instance per vlan

-RSTP: 802.1w reduces convergence time when physical topology changes occur or the configuration parameters are changed. RSTP defines additional roles for ports, such as toggle and backup, port status is defined as discard, learn or send.

-MSTP: 802.1s RSTP based
One instance for multiple vlans. Supports load balancing

MST: Multiple Spanning Tree
PVST: Per-VLAN Spanning Tree
PVST+: Per-VLAN Spanning Tree Plus

MSTP: 802.1s

MSTP provides an extension of RSTP.

Allows you to create multiple instances (MSTI) to balance traffic from all available physical links.

Each MSTP instance, that is, each MSTI creates an STP topology that we can map to one or more vlans. This way we can redound and balance layer 2 traffic.

In MSTP, a port can belong to multiple vlans and can be dynamically blocked in one instance and allowed in another. This reduces network resource consumption and keeps CPUs at an optimal level. It also provides the RSTP reconvergence time.

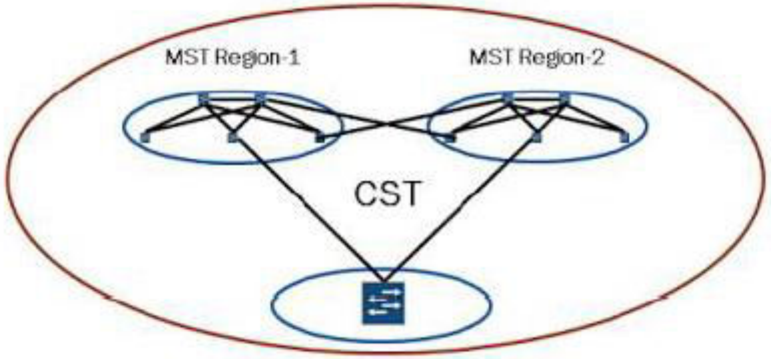
MSTP allows switches to be grouped into an "MST Region", where they share the region group name, revision level, and the mappings of the instances to the vlans. Each region supports up to 64 MSTIs.

MSTP reduces the number of BPDUs on the LAN by including spanning-tree information for all MSTIs in a single BPDU.

MSTP chooses a regional root bridge for each MSTI, which is calculated based on priority.

MSTP is compatible with STP and RSTP thanks to CST, which allows you to interconnect multiple MST regions or connect an MST with a switch running only STP.

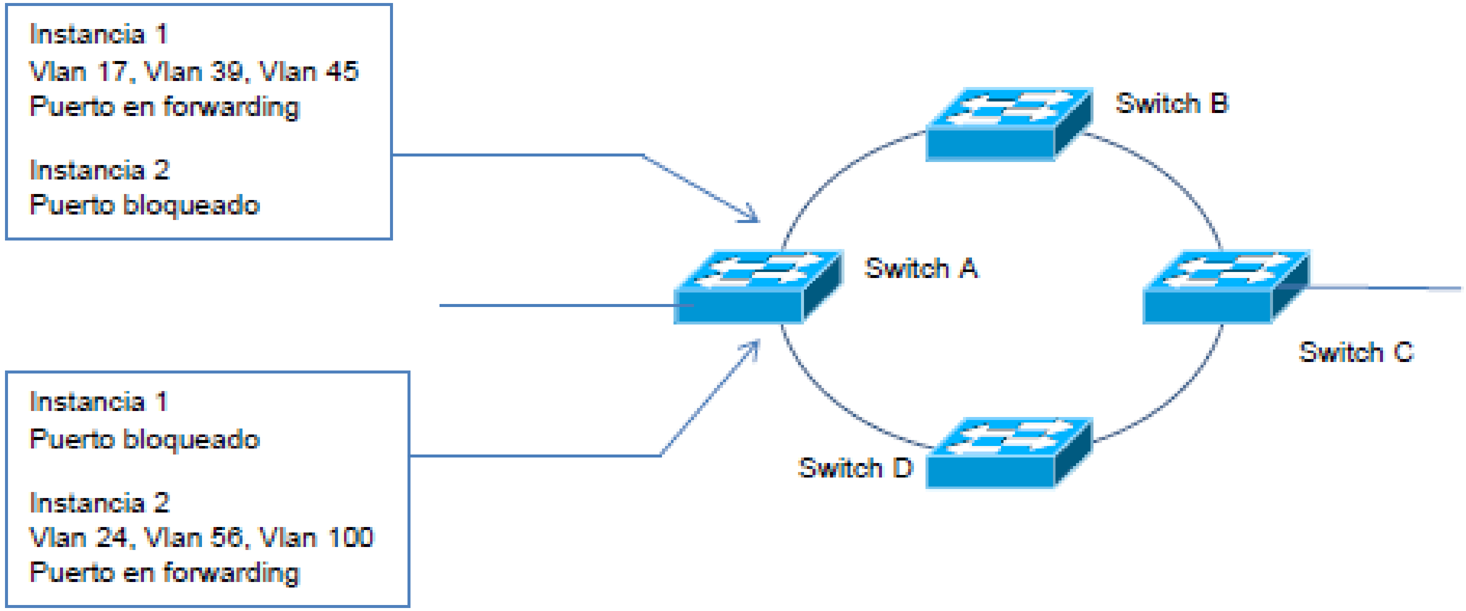
CST: Common Spanning Tree



Switch A configuration:

```
spanning-tree mode mst                          activa MST
spanning-tree mst configuration
name <nombre_de_la_region>
revision <numero>
instance <numero> vlan <vlans_definidas_en_la_instancia>
```

```
spanning-tree mst configuration
name ARBOL
revision 1
instance 1 vlan 17, 39, 45
instance 2 vlan 24, 56, 100
!
```



Instancia 1
Vlan 17, Vlan 39, Vlan 45
Puerto en forwarding

Instancia 2
Puerto bloqueado

Instancia 1
Puerto bloqueado

Instancia 2
Vlan 24, Vlan 56, Vlan 100
Puerto en forwarding

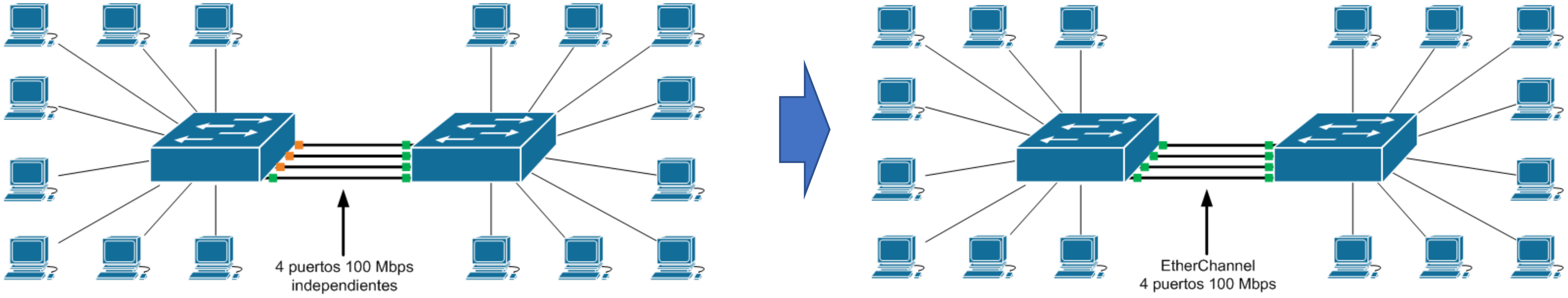
```
spanning-tree mst 1 priority 28672       backup root instancia 1
spanning-tree mst 2 priority 24576       root para la instancia 2
```

LINK AGGREGATION

Link Aggregation or Etherchannel:

Link aggregation allows the logical grouping of multiple physical Ethernet links. This grouping is treated as a single link and allows the nominal speed of each physical Ethernet port used to be added and thus obtain a high-speed trunk link. A maximum of 8 ports can be grouped together to form an EtherChannel. The ports used must have the same characteristics and configuration. By appearing to have a single port, the Spanning Tree protocol does not block them, which means that instead of having one port enabled and one or more backup ports (in case the one that is active fails), there can be two, four, eight active ports.

Note: EtherChannel is a Cisco technology built according to IEEE 802.3 standards, which can be used on both Layer 2 and Layer 3 ports. EtherChannel and the IEEE 802.3ad standard are very similar, serving the same purpose, although there are some small differences between the two.



LINK AGGREGATION

There are two protocols for port grouping:

- PAgP: “Port Aggregation Protocol”, proprietary to Cisco Systems.
- LACP: “Link Aggregation Control Protocol”, based on standards.

PAgP:

Operating modes:

- On: forces the ports to establish the channel.
- Off: Prevents ports from establishing a channel.
- Auto: wait to receive packets to negotiate the channel.
- Desirable: Sets the port to negotiate channel establishment using PAgP.

LACP:

Operating modes:

- On: forces the ports to establish the channel.
- Off: Prevents the channel from being established.
- Passive: puts the port on standby to receive LACP packets to negotiate the channel.
- Active: Sets the port to send packets to start channel negotiation.

It is necessary that the two devices use the same protocol for them to establish an EtherChannel.While, if two devices use different protocols and both have the “On” mode set, an EtherChannel will also be formed.The “On” mode requires having the “On” mode configured at the other end as well. If there is an interface in “Desirable” mode and the other end in “On” mode, the EtherChannel will not be established.

INTRODUCTION TO SDN

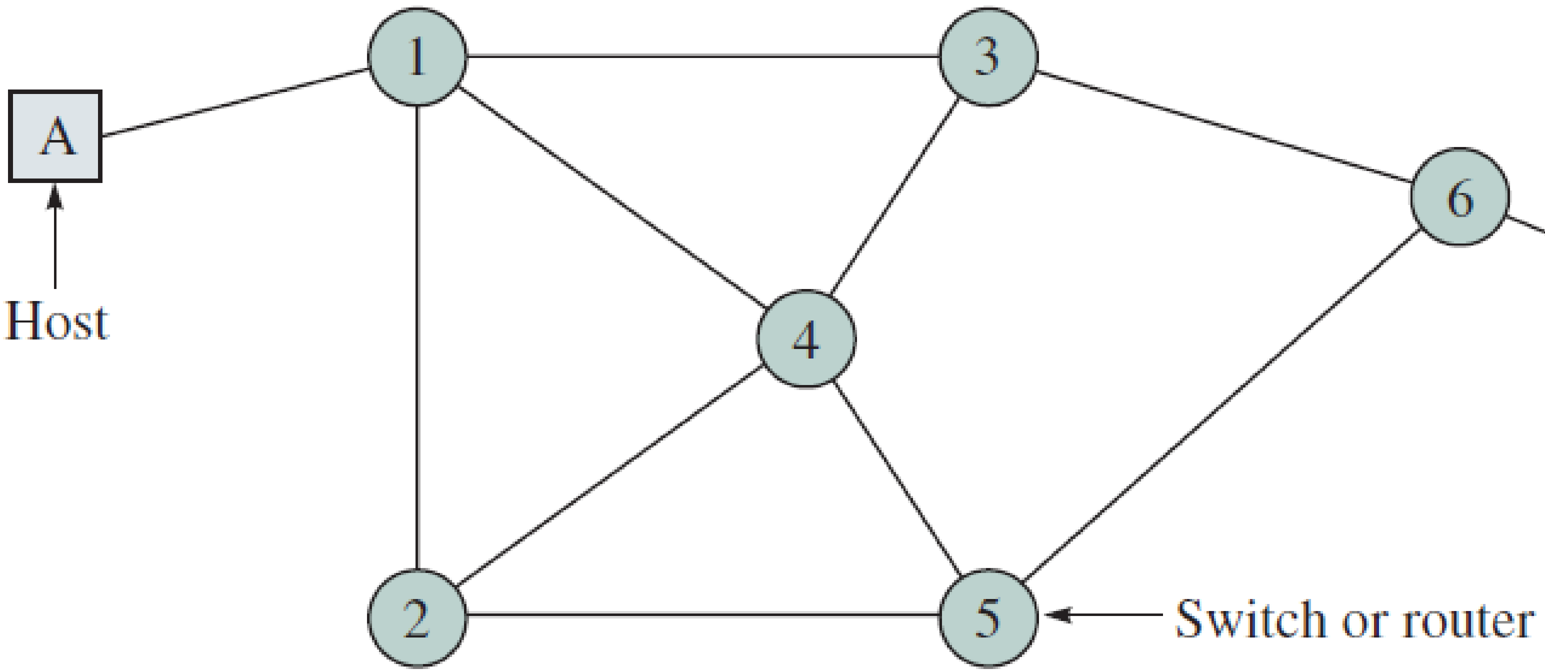
NETWORKING REVIEW PART B

Although NFV and SDN, have being adopted during last years, some “classical/underlay” technologies must be reviewed prior to move forward to the next step

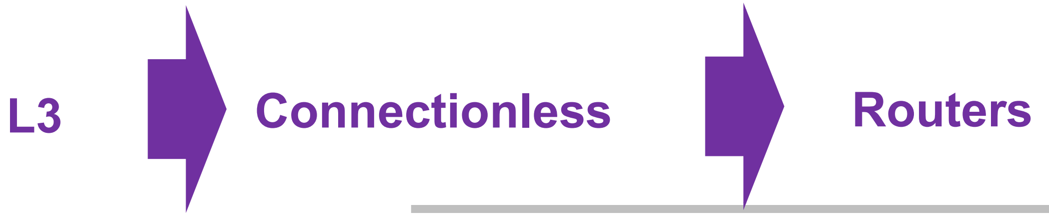
IP v4/v6 (Suite TCP/IP and related y protocols

Introduction to connectionless packet-switched networks.
IP networks, routers
DARPA model (TCP/IP)
IP protocol: packet format, header description.
Complementary protocols: ARP, ICMP, DNS
IP addresses
Addressing with classes
Public and private addressing
Subnet concept.
Fixed length mask, variable length mask
CIDR (Classless Inter-Domain Routing) Classless inter-domain routing
Using NAT (Network Address Translation) and PAT (Port Address Translation)
Comparison between IPv4 and IPv6
Transport level capabilities
Introduction to TCP
Introduction to UDP
Some applications supported in the DARPA model (Telnet, DNS, SNMP, DHCP)
IPSec
VRRP

IP NETWORKS



An IP network is a connectionless packet-switched network. It is made up of routers and links that link them together. As its name indicates, it is based on the IP protocol (layer 3 protocol, routed) and on the use of mechanisms (static routes/routing protocols) that allow routers to "learn" the destination networks, for the purposes of routing the packets to the final host.

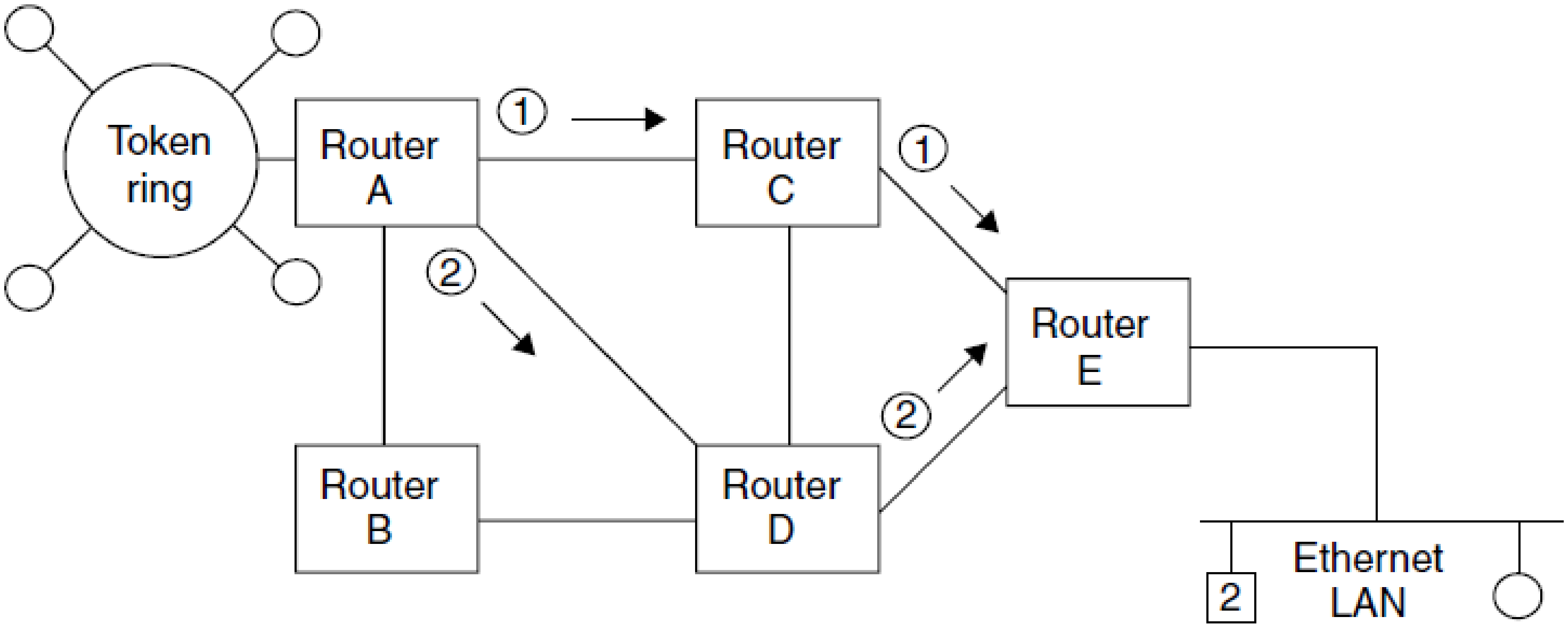


L3

Connectionless

Routers

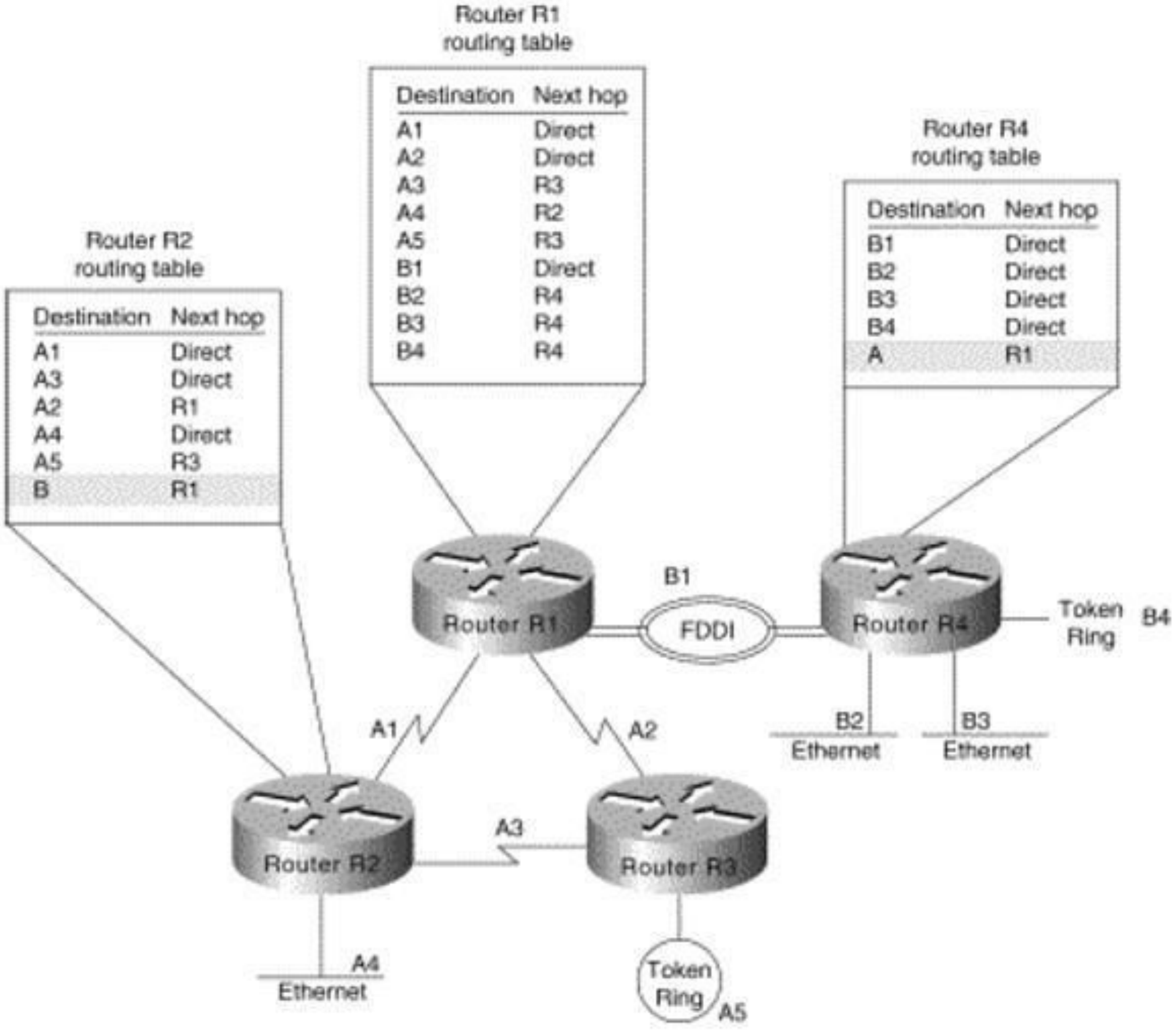
IP NETWORKS, CHARACTERISTICS



An IP network is a network that does not offer any type of guarantee. The packets can reach the final destination through different paths. Packages can get lost, they can get destination in different order as expected

IP NETWORKS, ROUTERS

Routing tables can be statics or there can be routing protocols



Router R1 routing table

Destination	Next hop
A1	Direct
A2	Direct
A3	R3
A4	R2
A5	R3
B1	Direct
B2	R4
B3	R4
B4	R4

Router R2 routing table

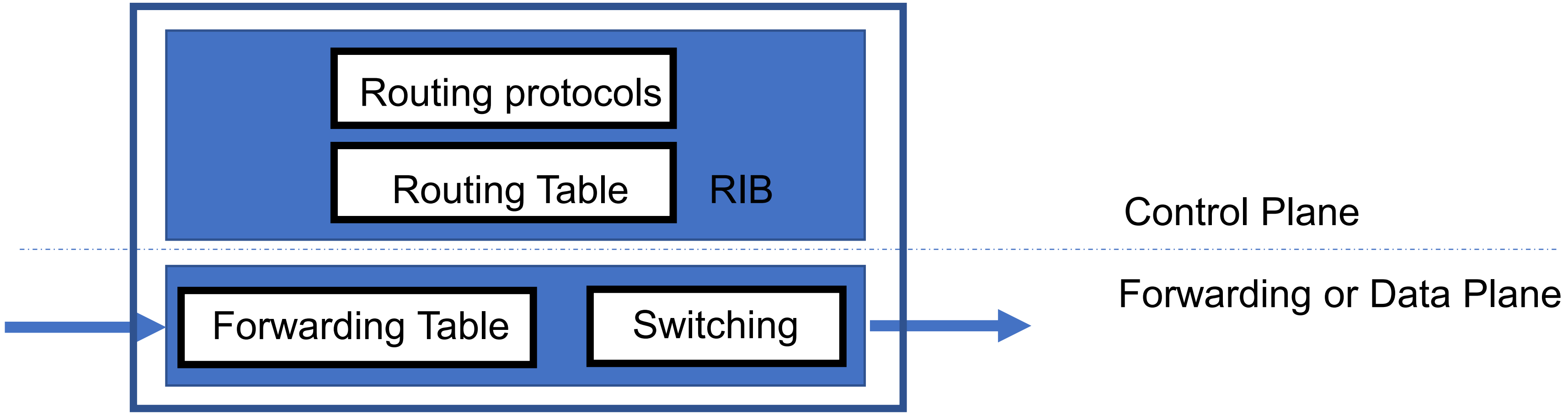
Destination	Next hop
A1	Direct
A3	Direct
A2	R1
A4	Direct
A5	R3
B	R1

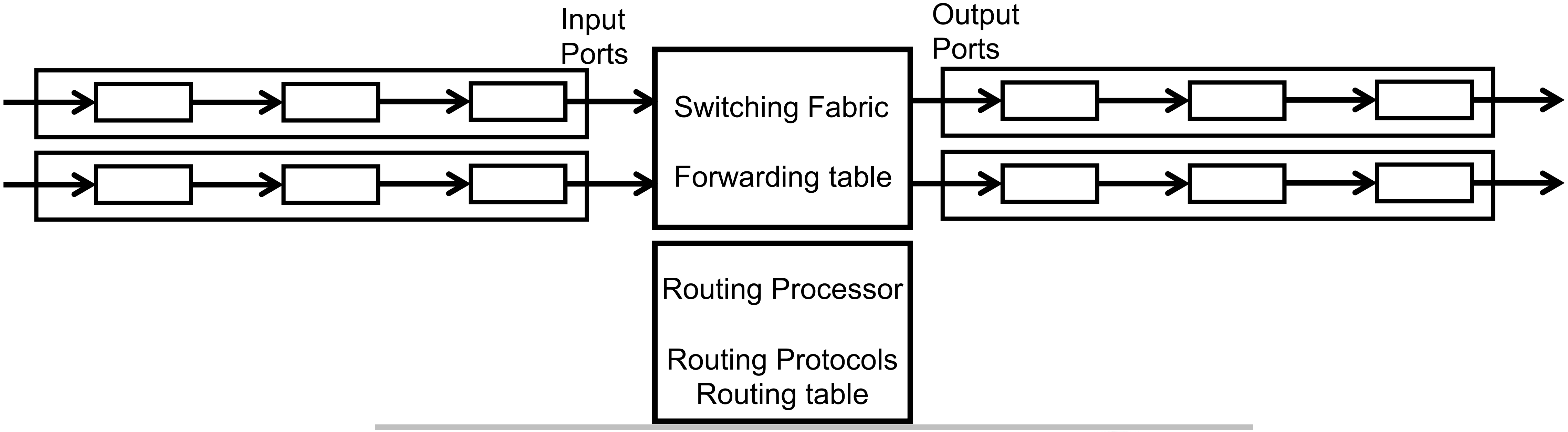
Router R4 routing table

Destination	Next hop
B1	Direct
B2	Direct
B3	Direct
B4	Direct
A	R1

IP NETWORKS, ROUTERS

The router executes two basic functions
-It runs routing protocols (control plane)
-It forward packets from Input Ports to Output Ports (forwarding Plane)

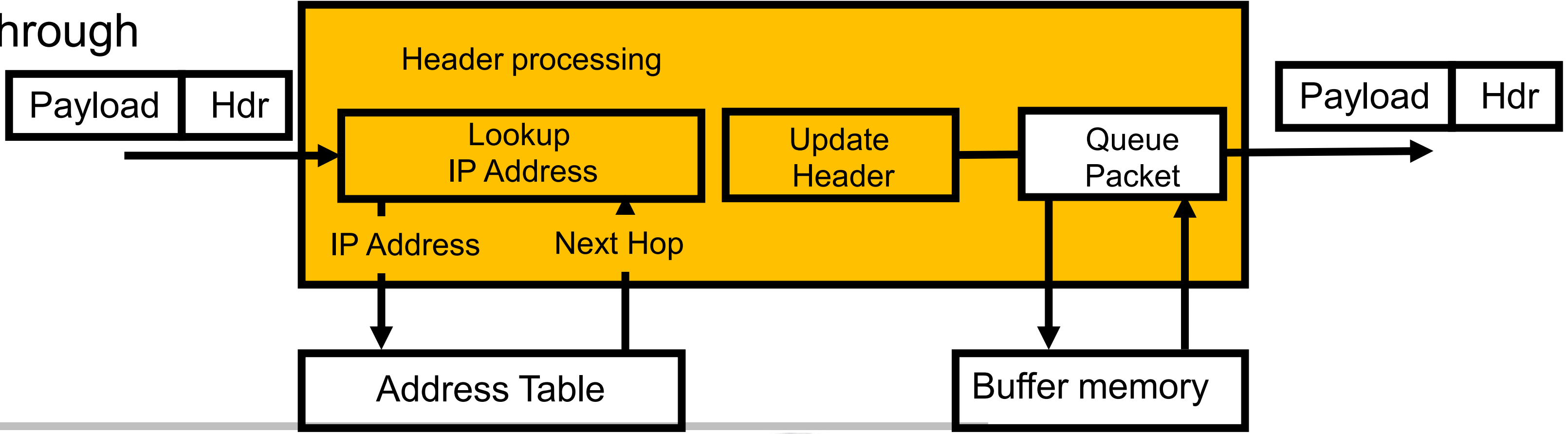




IP NETWORKS, PROCESSING IN A ROUTER

- Accept packets on the incoming interfaces
- **Lookup:** routers explore the routing tables of the different routing protocols (if more than one routing protocol is running) and select the best route to each destination. Routers associate with that destination the data link layer address of the next-hop device and the local exit interface to be used when forwarding packets toward the destination. Note that the next hop device could be another router, or perhaps the destination host. The forwarding information of the next hop device (data link layer address plus the outgoing interface) is located in the router's forwarding table. When a router receives a packet, the router examines the packet header to determine the destination address. The router consults the forwarding table to obtain the output interface and the next hop address to reach the destination.

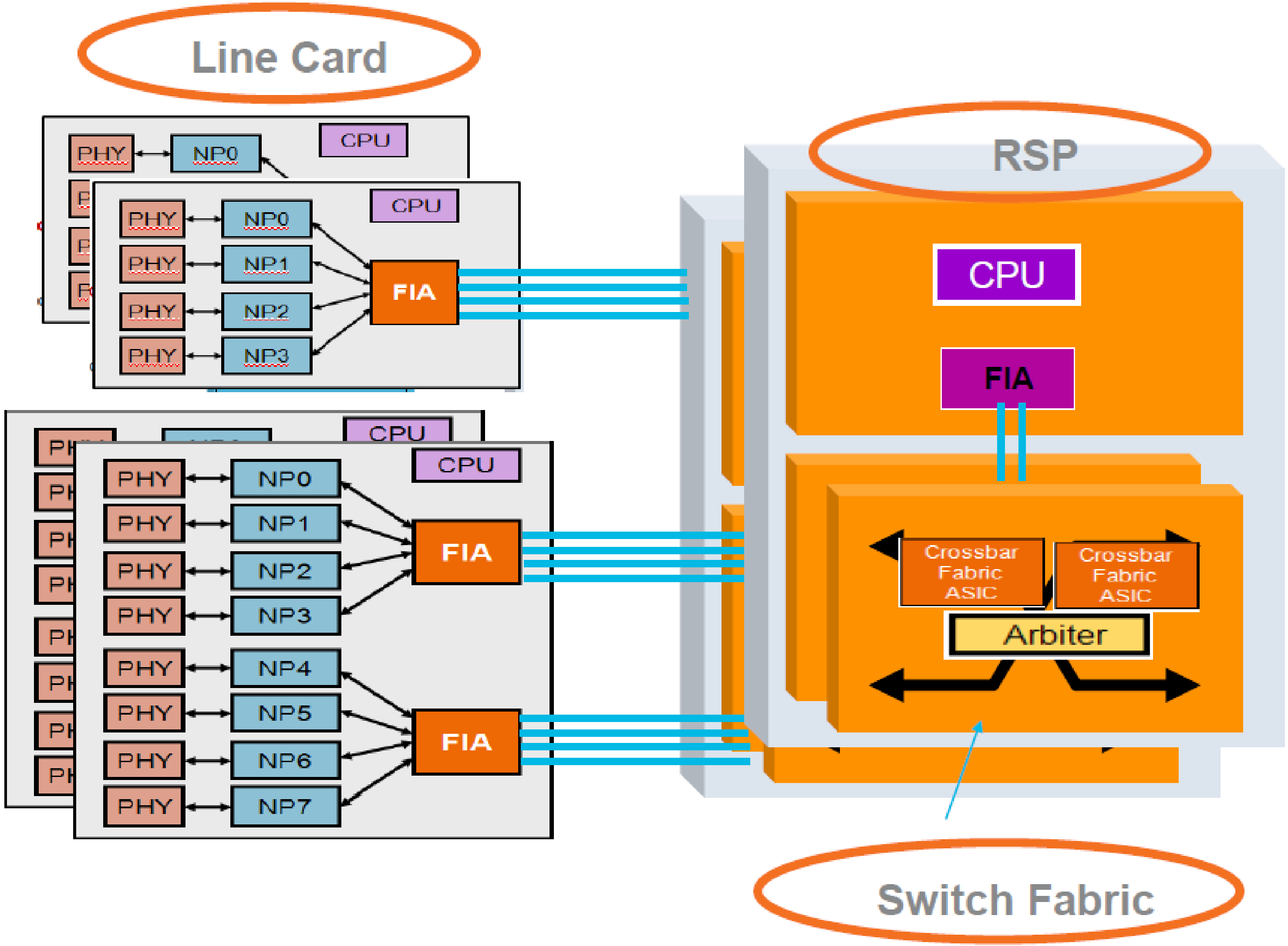
- **Processing:** IP header manipulation
- **Switching:** Send packet to destination port
- **Buffering:** Storage packet in output queue
- **Transmittion:** transmit the packet through the Output interface



CePETel
Sindicato de los Profesionales
de las Telecomunicaciones
SECRETARÍA TÉCNICA
IPEI

IP NETWORKS, ROUTER INTERNAL ARCHITECTURE

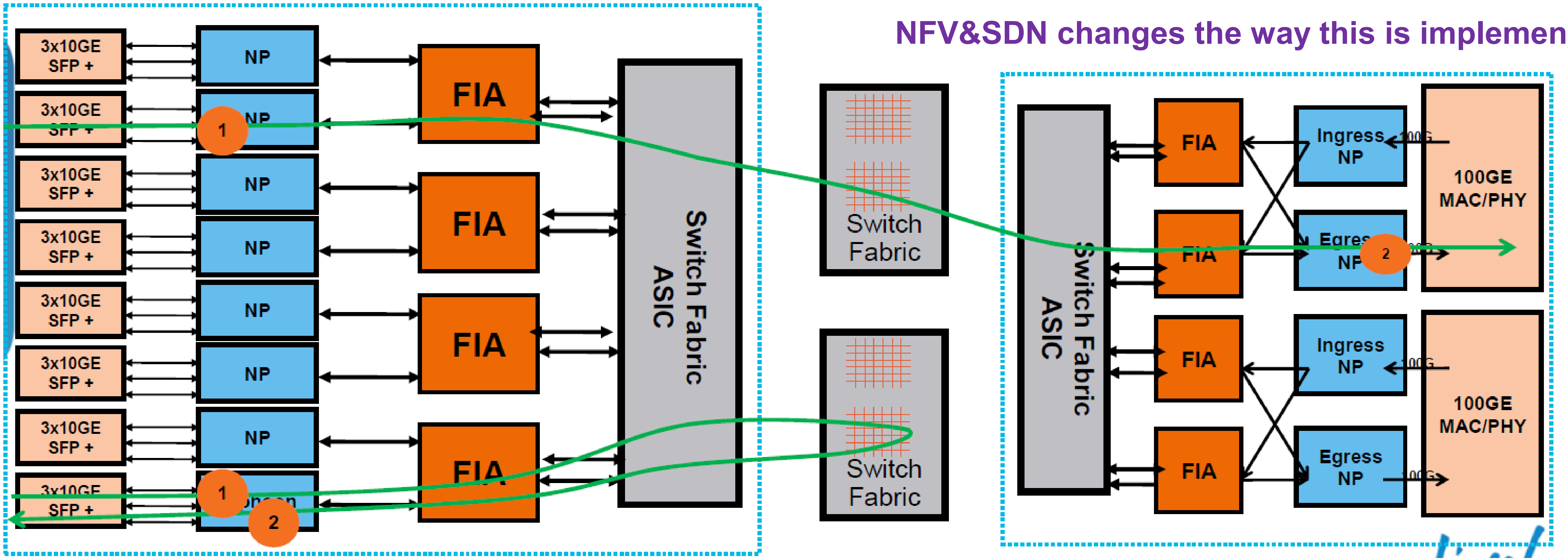
Example of Cisco ASR9000 system. Look at details of dedicated HW



CPU: Distributed Control Plane

IP NETWORKS, ROUTER INTERNAL ARCHITECTURE

Example of Cisco ASR9000 system. Packet flow. Look at details of dedicated HW



NFV&SDN changes the way this is implemented

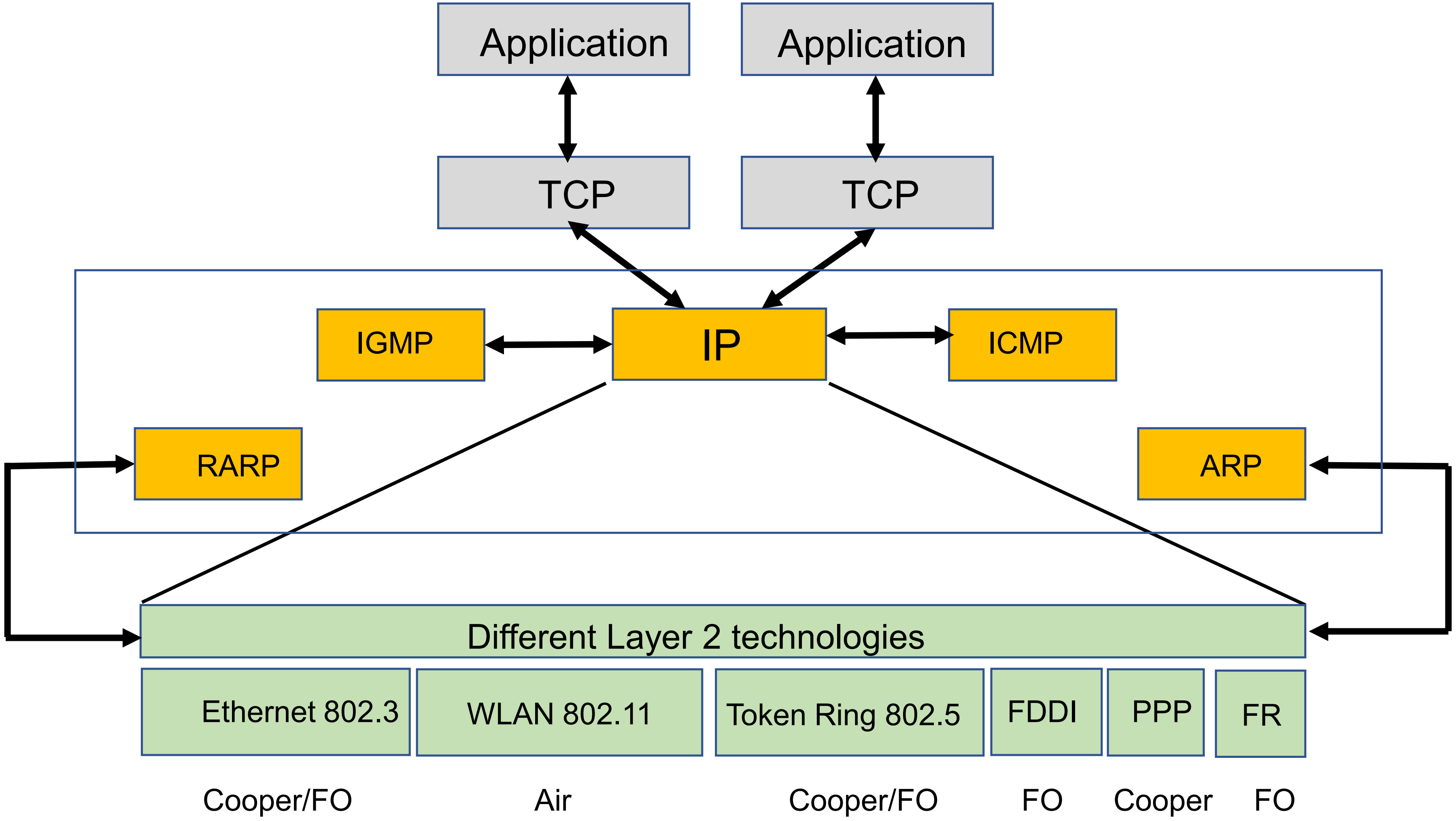
NP: Network Processor

- Forwarding engine and functionalities for the LC
- Shorter connections, communication channels faster
- Higher performance, higher density with lower power consumption

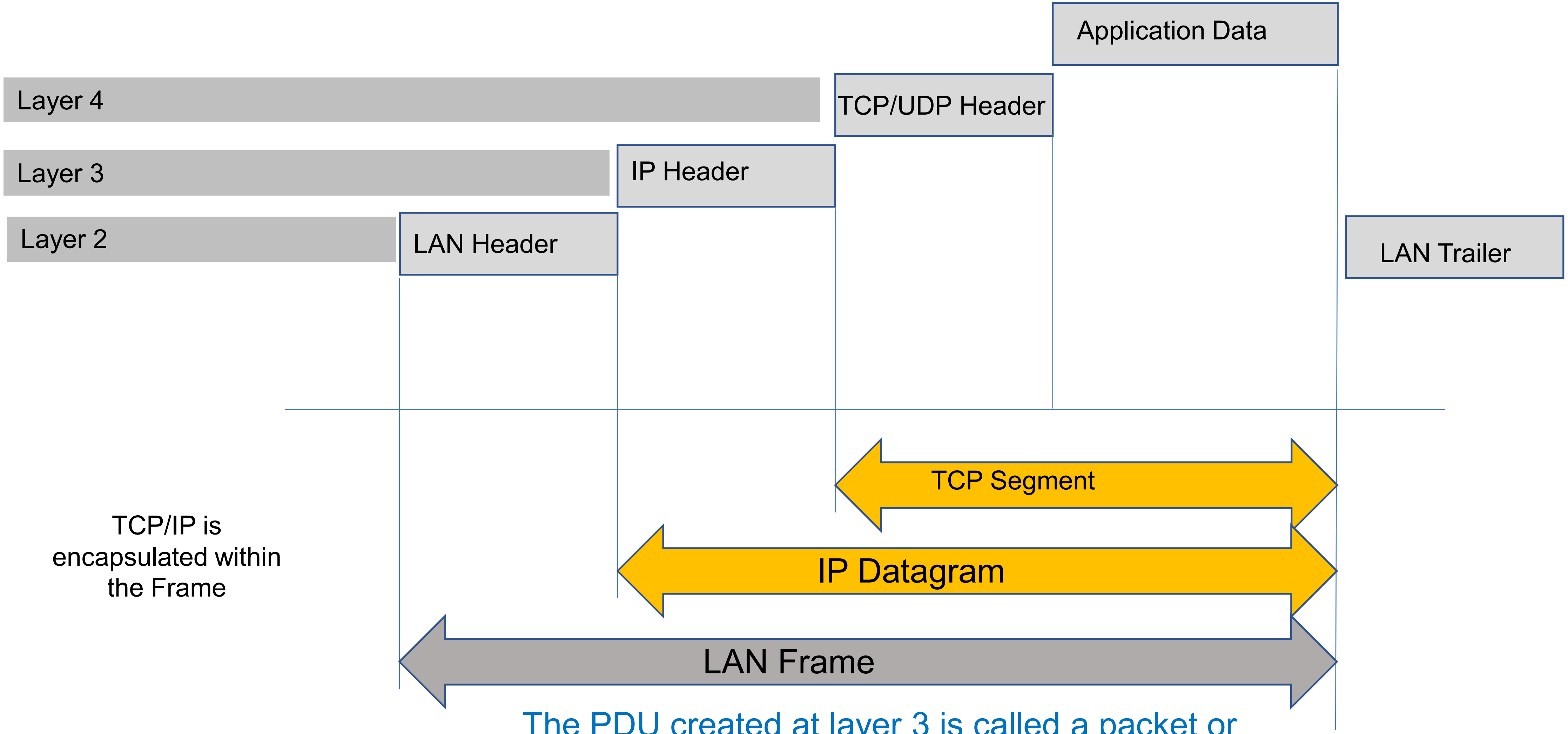
FIA (Fabric Interface ASIC dedicated to work as interface between processor and switch fabric)

- Arbitrations, framing and accounting in HW
- Provides buffering and virtual output queuing for switch fabric
- QoS handling: flexibility regarding the relative priority ofunicast vs. multicast

IP NETWORKS, DARPA MODEL



IP NETWORKS, DARPA MODEL



TCP/IP is
encapsulated within
the Frame

The PDU created at layer 3 is called a packet or datagram. The packets are PDUs of variable length.

CePETel

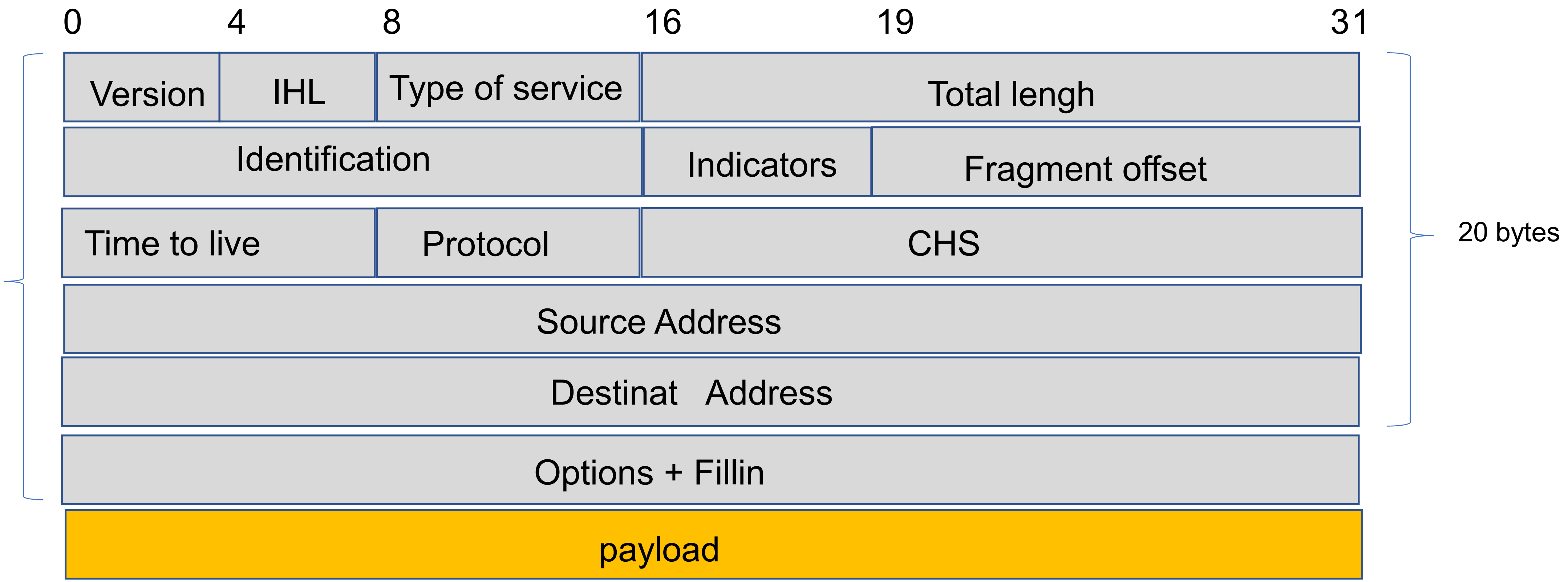
Sindicato de los Profesionales
de las Telecomunicaciones

SECRETARÍA TÉCNICA

IPEI

IP PACKET FORMAT and HEADER DESCRIPTION

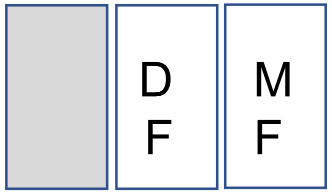
RFC
791



20 to 60
bytes

Minimum: 20 bytes : 5 32 bits words
Maximum: 60 bytes: 15 32 bits words

Indicators (3 bits)



Don’t fragment
More fragment

IPV4 Header is too long, has too many fields
Which is time consuming for routers.
IPV6 solve this issue

IP PACKET FORMAT

Version:
indicates the version number of the protocol, it is to allow its evolution; the value is 4.

IHL, Internet Header Length) (4 bits):
Header length expressed in 32-bit words. The minimum value is five, corresponding to a length of the minimum header of 20 octets. The maximum value of this 4-bit field is 15, which limits the header to 60 bytes and therefore the Options field to 40 bytes.

Service type (8 bits):
Used to specify the treatment of the data unit in its transmission through network components. Specifies the parameters of reliability, priority, delay and performance. The Service Type field is one of the few fields that has changed its meaning (slightly) for years. Its purpose is still to distinguish between different kinds of services. Various combinations of reliability and speed are possible. For digitized voice, the fast delivery is more important accurate than delivery. For file transfer, it is more important to error-free transmission than fast.

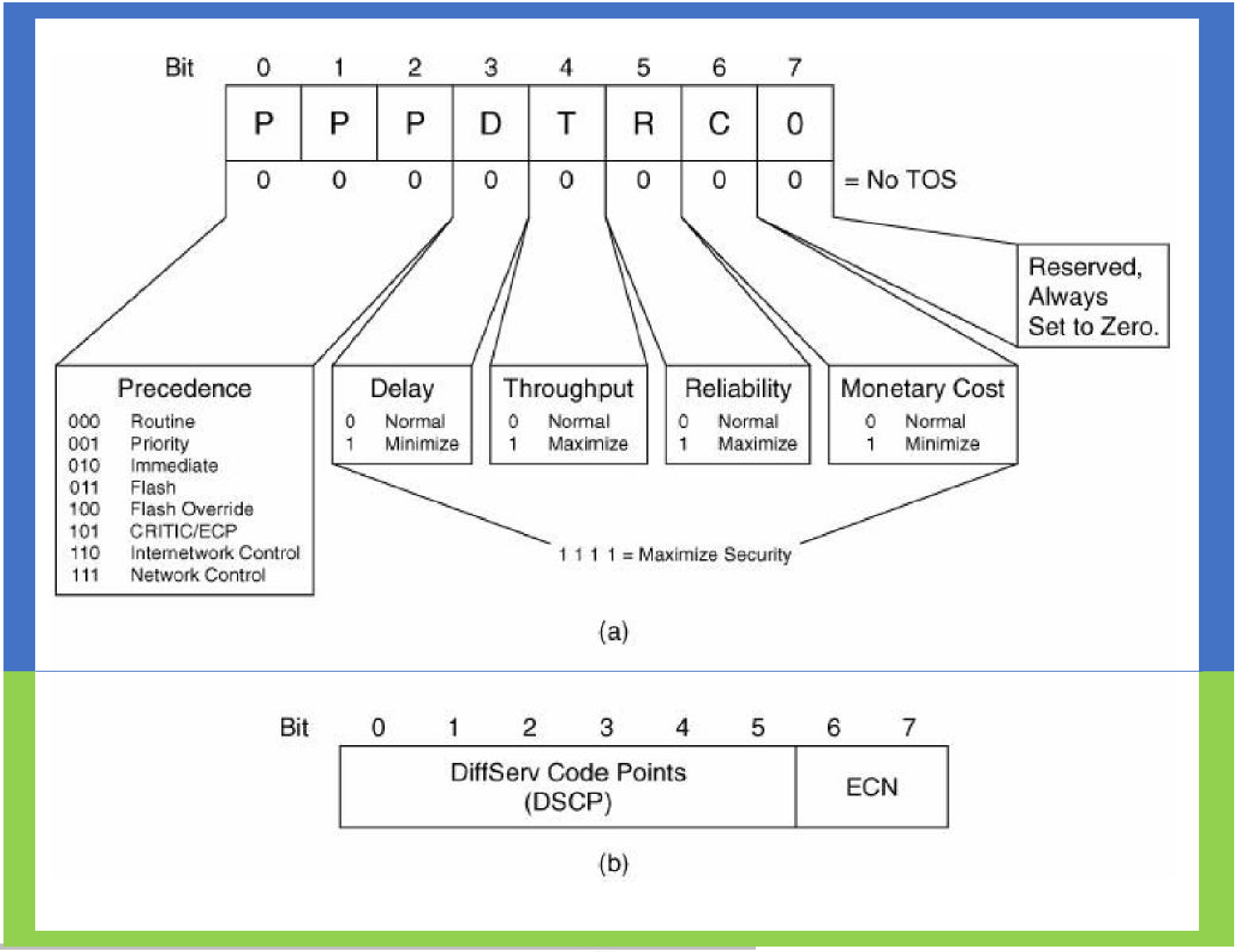
Originally, the 6-bit field contained (from left to right) a Precedence field three bits and three flags, D, T and R. The Precedence field is a priority, from 0 (normal) to 7(network control package). The three flag bits allow the host to specify what it is most important: {delay, throughput, reliability}.
In theory, these fields allow routers to make decisions between, for example, a link high-throughput, high-delay satellite line or a leased line with low throughput and low delay.

In practice, current routers completely ignore the Type of Service field unless tell them otherwise. It interpretation has recently been replaced.The first 6 bits of the field are now called the differentiated services field (DS, Differentiated Services).
The remaining 2 bits are reserved for a notification field Explicit Congestion Control (ECN), currently in the standardization phase. The ECN field provides a explicit congestion signaling. See next figure.

IP PACKET. TOS FIELD

In practice, current routers ignore the Type of Service field entirely, unless they are ask to read it.
Its interpretation has recently been superseded.
The first 6 bits of the field are now called the differentiated services field. (DS, Differentiated Services). See b).
The remaining 2 bits are reserved for a notification field of “ explicit congestion control (ECN)”, currently in the standardization phase. The ECN field provides a explicit signaling of congestion.

DSCP used for QoS purposes



(a)

(b)

IP PACKET FORMAT

Total length (16 bits): Total length of the datagram, in octets.

Identifier (16 bits):
a sequence number that, together with the source and destination address and the protocol user, is used to uniquely identify a datagram. Therefore, the identifier must be unique for the source address of the datagram, the destination address and the user protocol during the time in which that the datagram remains on the network. The Identification field is required for the destination host to determine which datagram a newly arrived fragment belongs to. All the fragments of a datagram contain the same ID value.

Indicators (3 bits):
Following the previous field comes an unused bit and then two 1-bit fields.
DF means Don't Fragment; It is an order to the routers not to fragment the datagram, because the destination is unable to put the pieces back together. For example, when booting a computer, its ROM might prompt you to send a memory image as a single datagram. By marking the datagram with the DF bit, the transmitter knows that it will arrive in one piece, even if it means that the datagram must avoid a small packet network on the best path and take a suboptimal path. This bit can be useful if the destination is known to have no fragment reassembly capability. However, if this bit is set to 1, the datagram will be discarded if the maximum size of a network on the path is exceeded. The “no fragmentation” bit prohibits fragmentation when it is 1. All machines are required to accept chunks of 576 bytes or less.
MF means more fragments. All fragments except the last one have this bit set, which is necessary to know when all the fragments of a datagram have arrived.

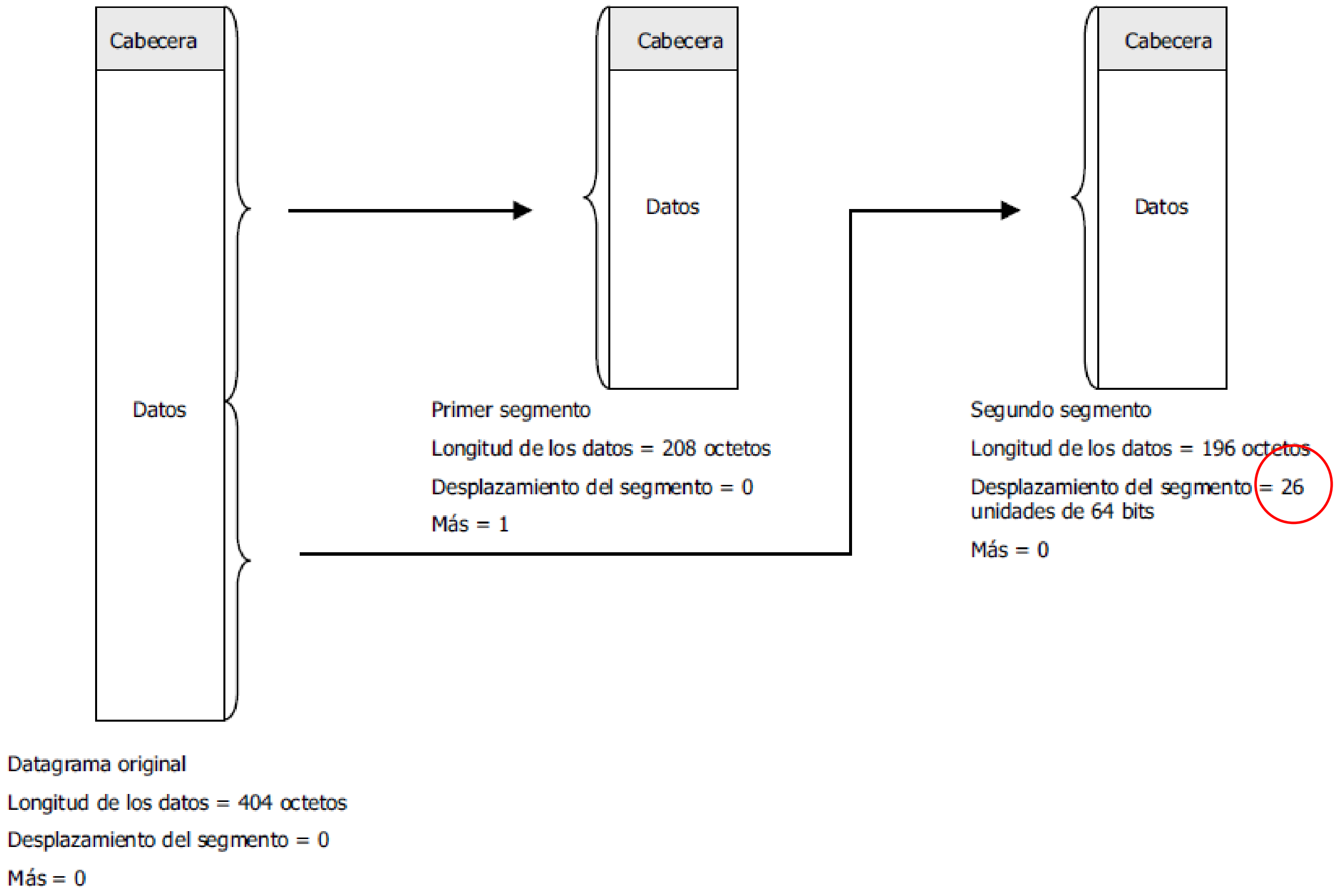
DF: Don´t fragment
MF: More fragment

IP PACKET FORMAT

Fragment Offset (13 bits):

Indicates where the fragment is located within the original datagram, measured in 64-bit units. This implies that all fragments except the last one contain a data field with a length multiple of 64 bits. Since they are provided 13 bits, there can be a maximum of 8192 fragments per datagram, giving a maximum length of 65,536-byte datagram, one more than the Total Length field.

65,536: 8192*64/8



Cabecera

Datos

Datagrama original

Longitud de los datos = 404 octetos

Desplazamiento del segmento = 0

Más = 0

Cabecera

Datos

Primer segmento

Longitud de los datos = 208 octetos

Desplazamiento del segmento = 0

Más = 1

Cabecera

Datos

Segundo segmento

Longitud de los datos = 196 octetos

Desplazamiento del segmento = 26 unidades de 64 bits

Más = 0

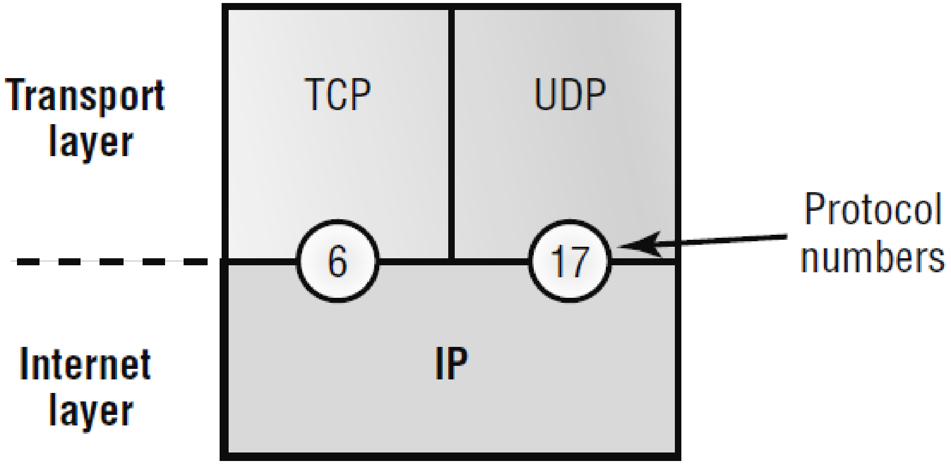
208 bytes = 1664 bits= 26 64-bits units

Protocol (8 bits): the Protocol field indicates the protocol of the upper layers to which it should deliver the package.

IP PACKET. PROTOCOL FIELD

Protocol (8 bits): the Protocol field indicates the protocol of the upper layers to which it should deliver the package.

ICMP:	protocol number 1
IGMP:	protocol number 2
IP in IP:	protocol number 4
RSVP:	protocol number 46
GRE:	protocol number 47
OSPF:	protocol number 89

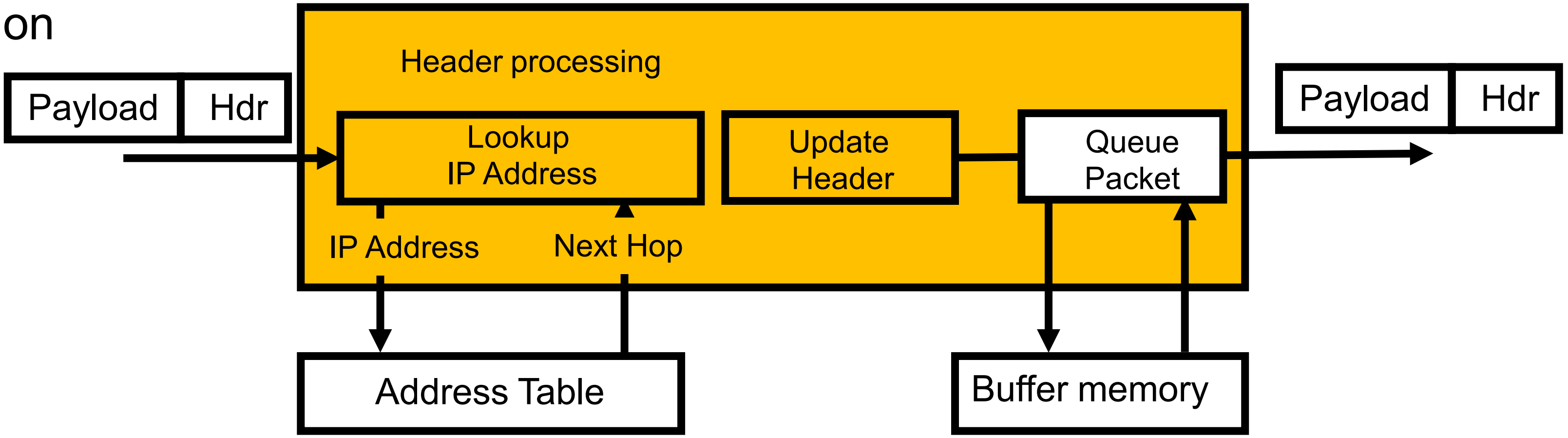


Time to live (8 bits): Specifies how much time, in seconds, a datagram is allowed stay on the network. Each routing device that processes the datagram must decrease this field by at least one unit, so that the lifetime is some how similar to a skip count.

IP PACKET. PROTOCOL FIELD

Header checksum (16 bits):
The header checksum verifies only the header. Such a checksum is useful for detecting errors generated by wrong words in a router. The algorithm adds all the 16-bit halfwords as they arrive, using one's complement arithmetic, and then obtains the one's complement of the result. For the purposes of this algorithm, the header checksum is assumed to be zero when the packet arrives at the destination. This algorithm is more robust than a normal addition. Note that the header checksum must be recalculated on each hop, since at least one of the fields always changes (the Time to Live field), but some tricks can be used to speed up the calculation.

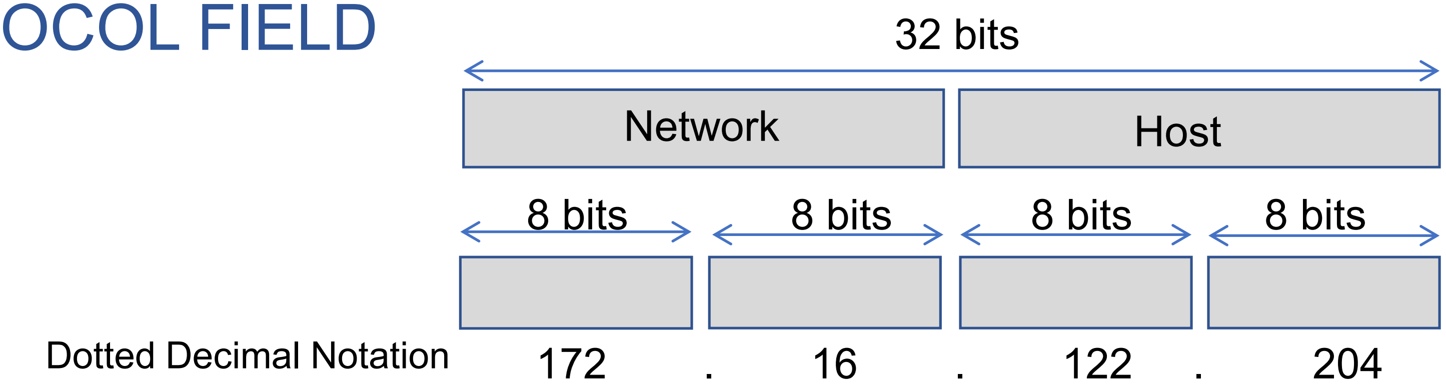
IP header manipulation



Source address (32 bits): encoded to allow variable bit assignment for specify the network and the end system (host) connected to the specified network.

Destination address (32 bits): same as the previous field.

IP PACKET. PROTOCOL FIELD



32 bits

Network

Host

8 bits

8 bits

8 bits

8 bits

Dotted Decimal Notation

172 . 16 . 122 . 204

Options (variable): Contains the options requested by the user submitting the data. Five options were originally defined, as listed in the table below, but have been added others more.

Option	Description
Security	Specifies if datagram is secret or not
Strict routing from source	Indicates the complete route to destination
Free routing from source	Provide a list of routers which must not be avoided
Register route	Ensure that each router insert its own IP address
Time stamp	Ensure that each router insert its own IP address and time stamp

Fillin (variable): used to ensure that the datagram header has a length multiple of four bytes (32 bits).

Data (variable): the data field must have a length multiple of 8 bits. The maximum length ofa datagram (data field plus header) is 65,535 octets.

- The security option indicates how secret the information is.

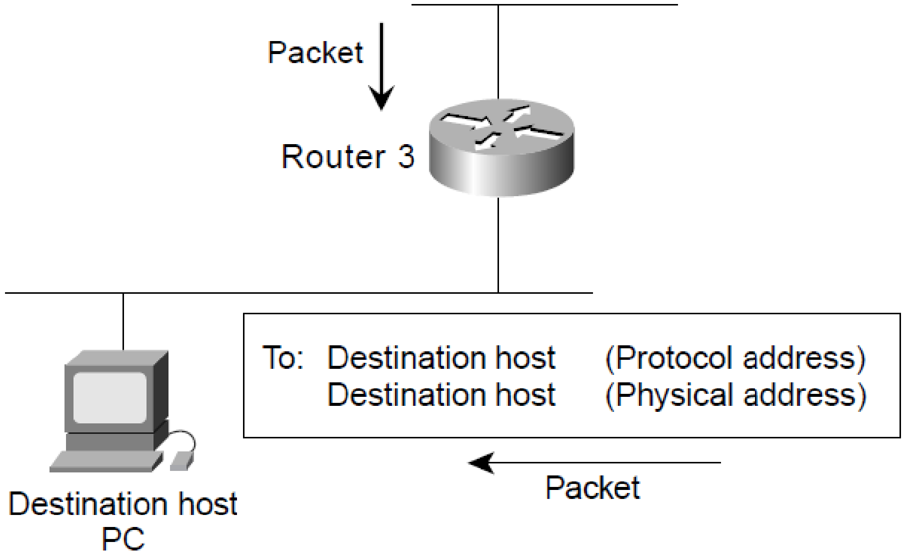
- Strict routing from source option gives the full path from source to destination as a sequence of IP addresses. The datagram is required to follow that exact path.

- The free routing from source option requires the packet to pass through the routers indicated in the list, and in the order specified, but is allowed to pass through other routersin the path.

- The register route option tells routers along the route to add their IP address to the option field. This allows system administrators to look for flaws in the algorithmsrouting

- The timestamp option is like the register route option, except that in addition to registering its 32-bit IP address, each router also registers a 32-bit timestamp. Thisoption is also primarily for searching for faults in routing algorithms.

ARP (Address Resolution Protocol)

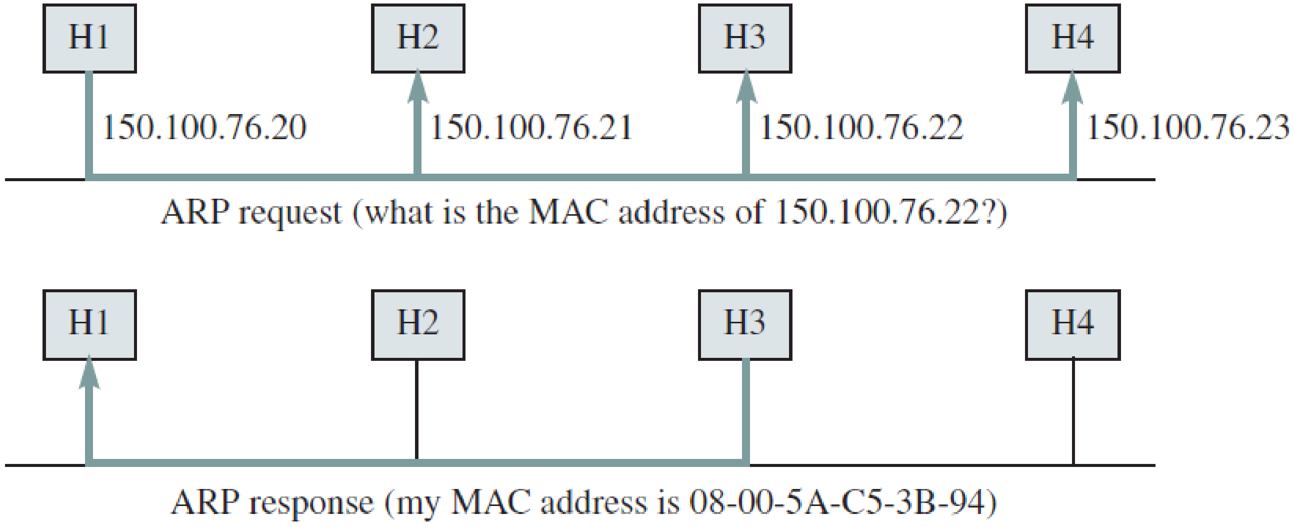

Packet
Router 3
To: Destination host (Protocol address)
Destination host (Physical address)
Packet
Destination host
PC

The cards send and receive frames based on 48-bit Ethernet addresses. They don't know anything about 32-bit IP addresses. The question now is: how are IP addresses linked to data link layer addresses, such as Ethernet?

Router 3 looks up the destination address in its table and can see that the destination is on its own network, but it needs some way to find the destination Ethernet address. One solution is to have a configuration file somewhere on the system that maps IP addresses to Ethernet addresses.

While this solution is certainly possible, for organizations with thousands of machines, keeping all these files up-to-date is error-prone and time-consuming.

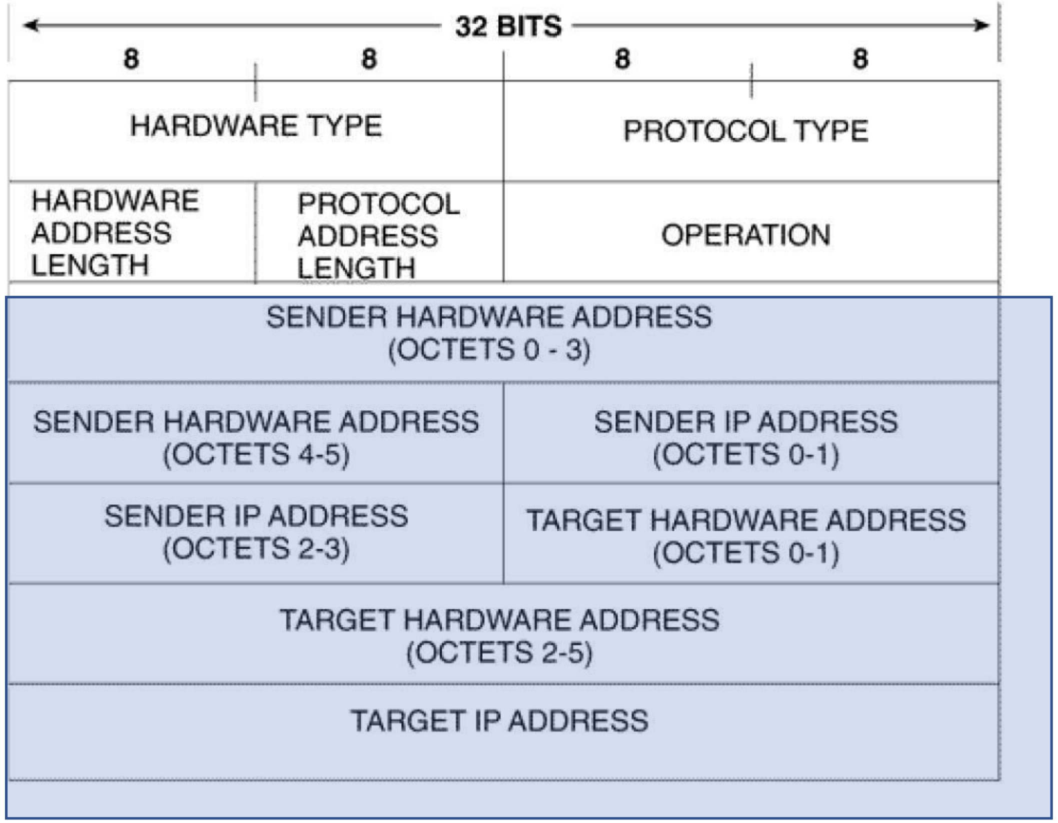
A better solution is for the router to output a broadcast packet to the Ethernet LAN asking: who owns such a destination IP address? The broadcast will reach every machine on the Ethernet LAN, and each will verify its IP address. The host in question will simply reply with its MAC address. In this way, the router learns that this IP address is on the host with the obtained Ethernet address.


H1
H2
H3
H4
150.100.76.20
150.100.76.21
150.100.76.22
150.100.76.23
ARP request (what is the MAC address of 150.100.76.22?)
H1
H2
H3
H4
ARP response (my MAC address is 08-00-5A-C5-3B-94)

ARP (Address Resolution Protocol)

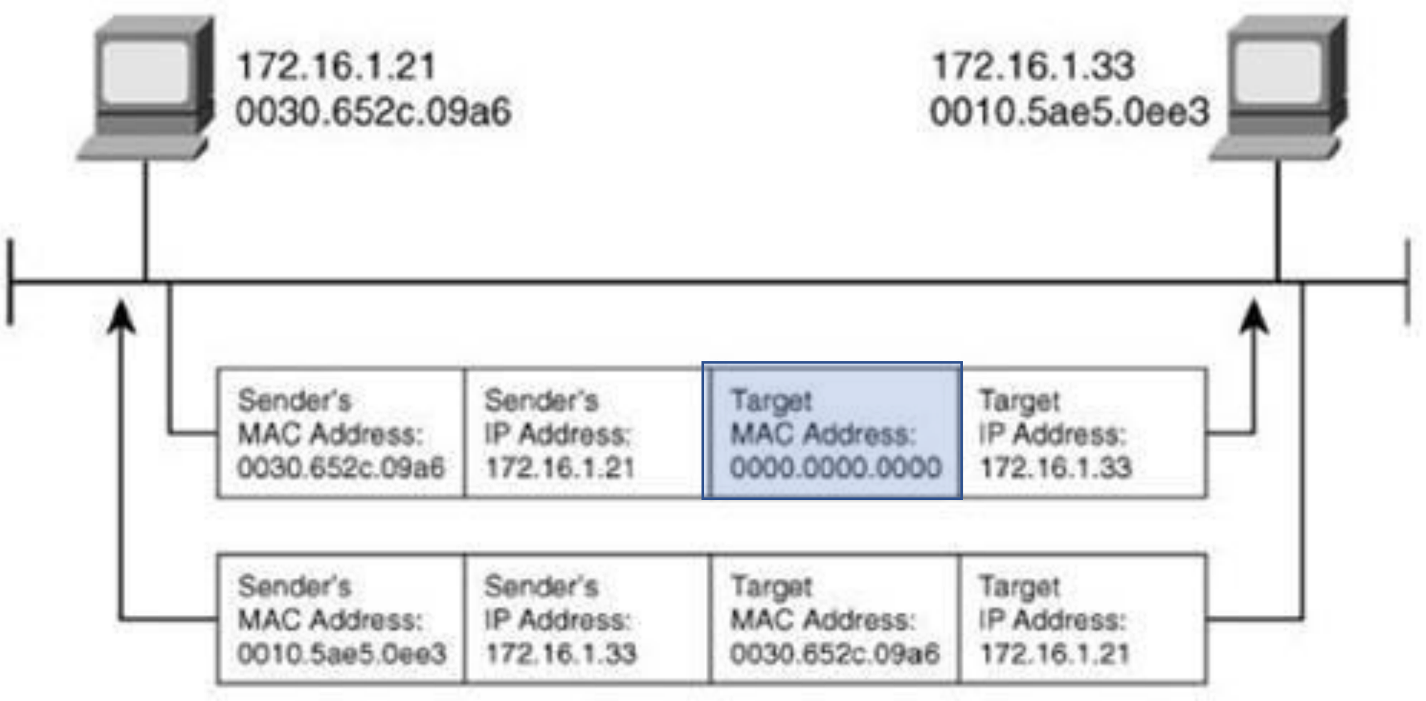
The protocol used to ask this question and get the answer is called: Address Resolution Protocol (ARP). Every machine on the Internet runs it. The definition of ARP is in RFC 826. Several optimizations can be made to make ARP work more efficiently. To start, once a machine has executed ARP, it saves the result in case it needs to contact in a short time again in contact with the same machine. The next time it will find the match in its own cache, thus eliminating the need for a second broadcast.

Hardware address, MAC address: 6 octets, 48 bits
Logical address, IP address: 4 octets, 32 bits



32 BITS

8	8	8	8
HARDWARE TYPE		PROTOCOL TYPE	
HARDWARE ADDRESS LENGTH	PROTOCOL ADDRESS LENGTH	OPERATION	
SENDER HARDWARE ADDRESS (OCTETS 0 - 3)			
SENDER HARDWARE ADDRESS (OCTETS 4-5)		SENDER IP ADDRESS (OCTETS 0-1)	
SENDER IP ADDRESS (OCTETS 2-3)		TARGET HARDWARE ADDRESS (OCTETS 0-1)	
TARGET HARDWARE ADDRESS (OCTETS 2-5)			
TARGET IP ADDRESS			



172.16.1.21
0030.652c.09a6

172.16.1.33
0010.5ae5.0ee3

Sender's MAC Address: 0030.652c.09a6	Sender's IP Address: 172.16.1.21	Target MAC Address: 0000.0000.0000	Target IP Address: 172.16.1.33
Sender's MAC Address: 0010.5ae5.0ee3	Sender's IP Address: 172.16.1.33	Target MAC Address: 0030.652c.09a6	Target IP Address: 172.16.1.21

ARP (Address Resolution Protocol)

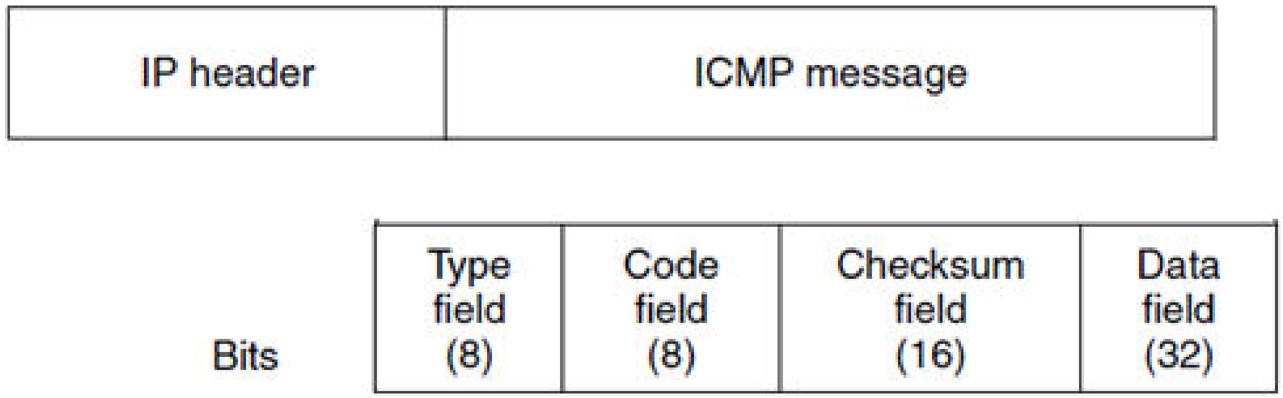
The protocol used to ask “who owns such a destination IP address?” or Which is the MAC address?” and get the answer is called: Address Resolution Protocol (ARP). Every machine on the Internet runs it. The definition of ARP is in RFC 826. Several optimizations can be made to make ARP work more efficiently. To begin with, once a machine has executed ARP, it saves the result in case it has to contact the same machine again in a short time. The next time it will find the match in its own cache, thus eliminating the need for a second broadcast.

RARP

ARP solves the problem of finding which Ethernet address corresponds to a given IP address. Sometimes you have to solve the inverse problem: given an Ethernet address, what is the corresponding IP address? In particular, this problem occurs when a diskless workstation is initialized. Such a machine will normally receive the binary image of its operating system from a remote file server. But how do you learn your IP address? The first solution invented was to use RARP (Reverse Address Resolution Protocol) (its definition is in RFC 903). This protocol allows a newly initialized workstation to broadcast its Ethernet address and say: “My 48-bit Ethernet address is 14.04.05.18.01.25. Does anyone know my IP address?” The RARP server sees this request, looks up the Ethernet address in its configuration files, and returns the corresponding IP address.

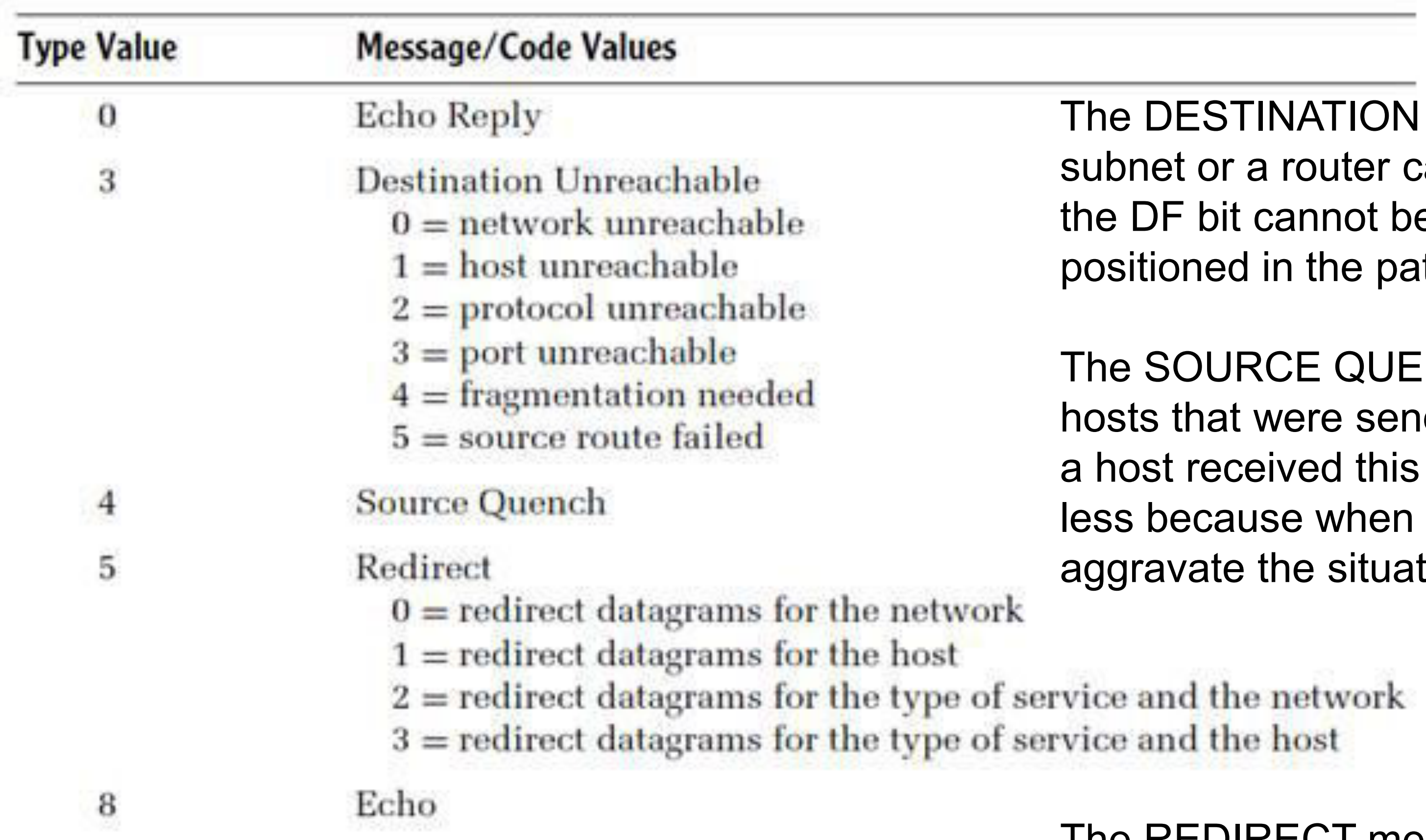
ICMP (Internet Control Message Protocol)

In essence, ICMP provides feedback information about problems in the communication environment. Some situations where it is used are: when a datagram cannot reach its destination, when the router does not have the ability to buffer to forward the datagram, and when the router instructs a station to send the traffic down a shorter route. In most cases, the ICMP message is sent in response to a datagram, either by a router in the path of the datagram or by the intended destination computer. Although ICMP is, for all intents and purposes, on the same level as IP in the TCP/IP protocol suite, it is a user of IP. When an ICMP message is constructed, it is passed to IP, which encapsulates the message with an IP header and then transmits the resulting datagram in the normal way. Since ICMP messages are transmitted in IP datagrams, their delivery is not guaranteed and their use cannot be considered reliable.



Type (8 bits): Specifies the type of ICMP message.
Code (8 bits): Used to specify message parameters that can be encoded in one or a few bits.
Checksum (16 bits): Checksum of the entire ICMP message. The same checksum algorithm is used as in IP.
Parameters (32 bits): Used to specify longer parameters. These fields are usually followed by additional information fields that further specify the content of the message.

ICMP (Internet Control Message Protocol)



Type Value	Message/Code Values
0	Echo Reply
3	Destination Unreachable 0 = network unreachable 1 = host unreachable 2 = protocol unreachable 3 = port unreachable 4 = fragmentation needed 5 = source route failed
4	Source Quench
5	Redirect 0 = redirect datagrams for the network 1 = redirect datagrams for the host 2 = redirect datagrams for the type of service and the network 3 = redirect datagrams for the type of service and the host
8	Echo

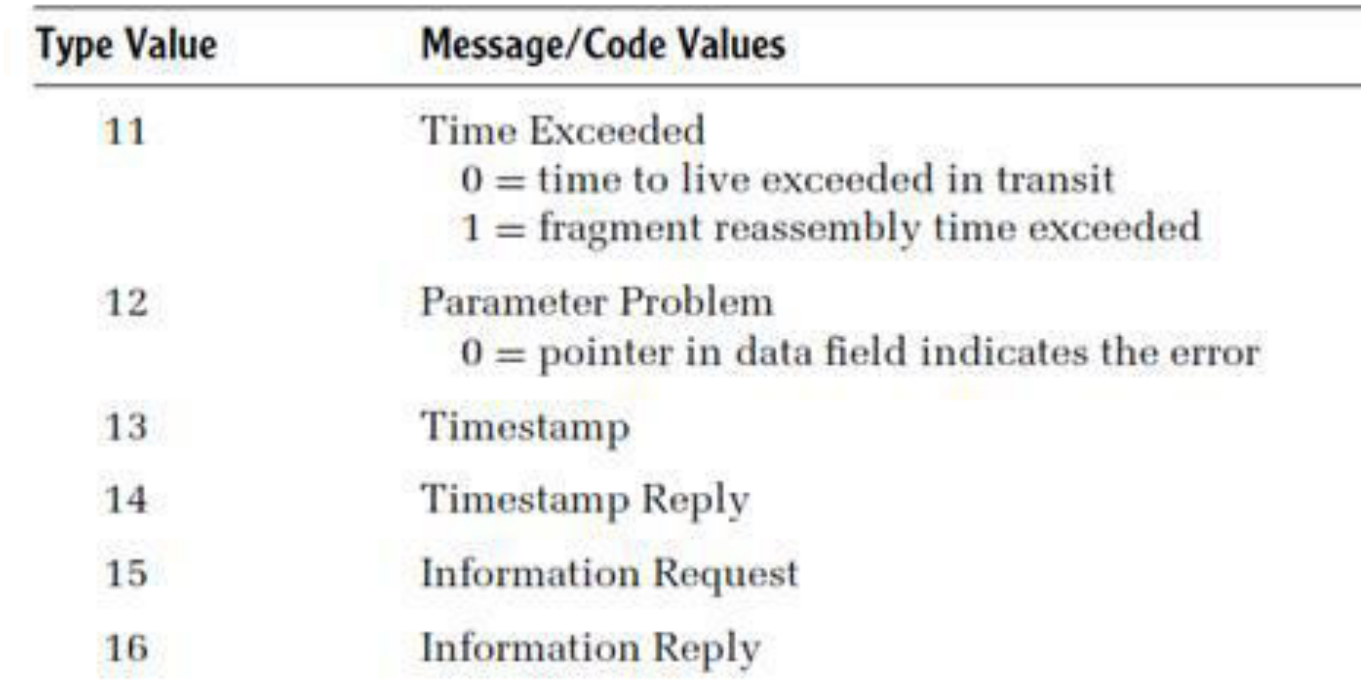
The DESTINATION UNREACHABLE message is used when the subnet or a router cannot locate the destination or when a packet with the DF bit cannot be delivered because a "small packet" network is positioned in the path.

The SOURCE QUENCH message was previously used to throttle hosts that were sending too many packets. It was expected that when a host received this message it would slow down. It is used less and less because when congestion occurs, these packets tend to further aggravate the situation.

The REDIRECT message is used when a router notices that a packet appears to be misrouted. It is used by the router to notify the sending host of the likely failure.

The ECHO and ECHO REPLY messages are used to see if a given destination is reachable and alive. The destination is expected to send back an ECHO REPLY message after receiving the ECHO message (Ping)

ICMP (Internet Control Message Protocol)



Type Value	Message/Code Values
11	Time Exceeded 0 = time to live exceeded in transit 1 = fragment reassembly time exceeded
12	Parameter Problem 0 = pointer in data field indicates the error
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply

The TIME EXCEEDED message is sent when a packet is dropped because its counter has reached zero. This event is a symptom that packets are repeating, there is massive congestion, or the timer values have been set too low.

The PARAMETER PROBLEM message indicates that an illegal value has been discovered in a header field.This problem indicates a bug in the IP software of the sending host or possibly in the software of a transit router.

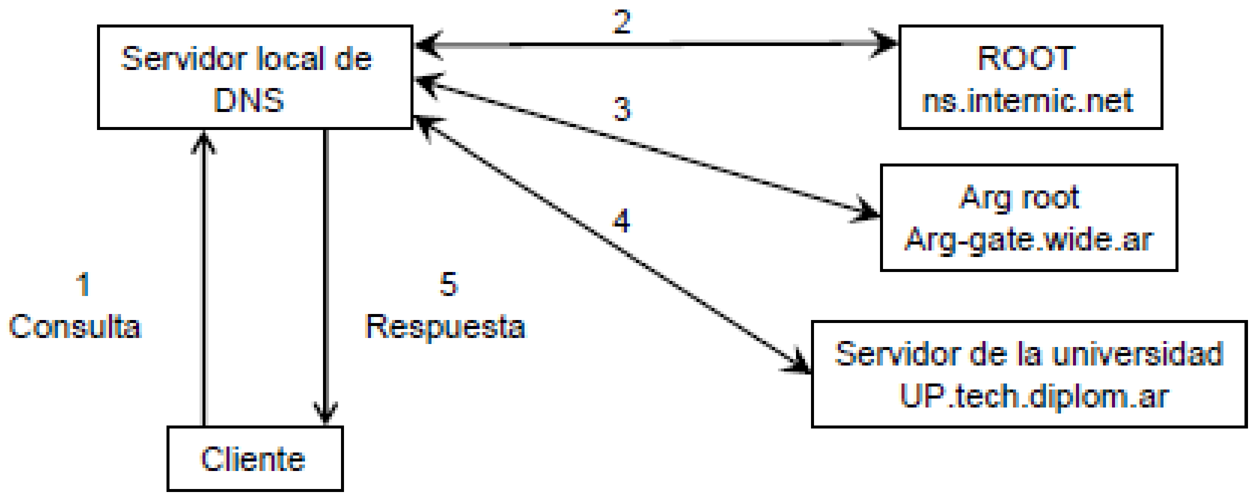
The ECHO and ECHO REPLY messages are used to see if a given destination is reachable and alive. The destination is expected to send back an ECHO REPLY message after receiving the ECHO (Ping) message.

TIMESTAMP REQUEST and TIMESTAMP REPLY messages are similar, except that the time of the arrival of the message and the departure of the reply are recorded in the reply. This feature is used to measure network performance.

COMPLEMEMTARY PROTOCOLS: DNS (DOMAIN NAME SERVER))

Typically, an end user knows the name of a host, but not its address. However, IP needs to know the address of a host in order to communicate with it. Also the end user, or the application that the end user has invoked, needs a mechanism to obtain the address from the name of a given host.The Domain Name System (DNS) was created to provide a better method for keeping track of names and addresses on the Internet. DNS is a distributed database. Internet names and addresses are stored on servers around the world. DNS databases provide automatic name-to-address translation services.

Traducción de nombre a dirección



Translation from address to name

The Domain Name System is versatile and also allows address to name translation. The way to do it using nslookup may seem a bit strange:

- It is determined that the query is of type ptr.
- The address is written in reverse, followed by .in – addr.arpa

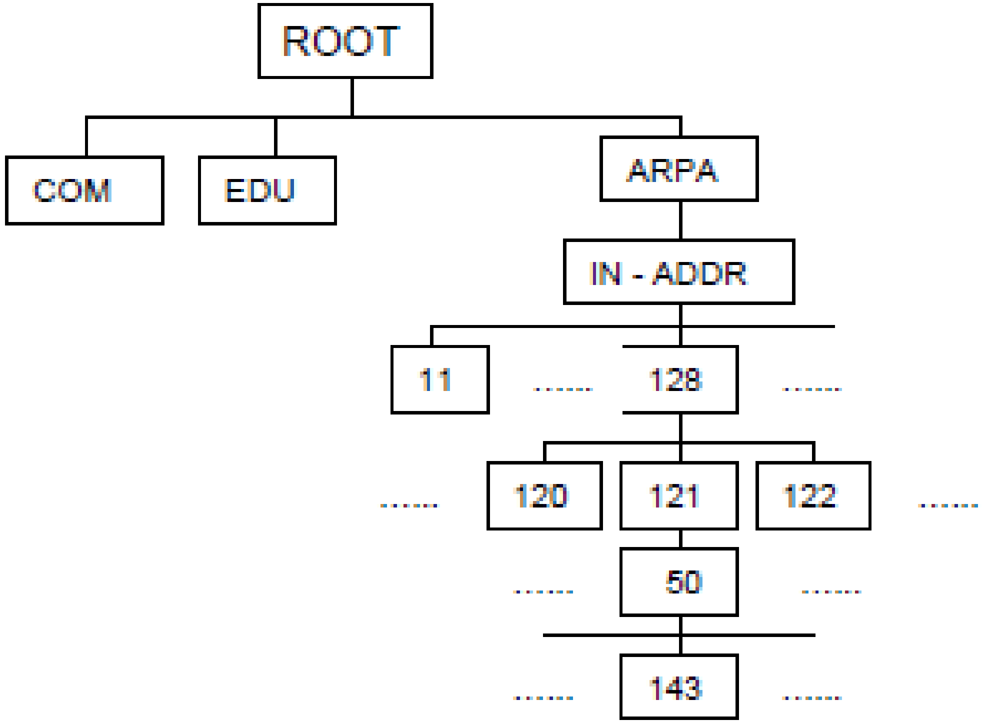
```
>Set type=ptr
```

- 143.50.121.128. in – addr.arpa.

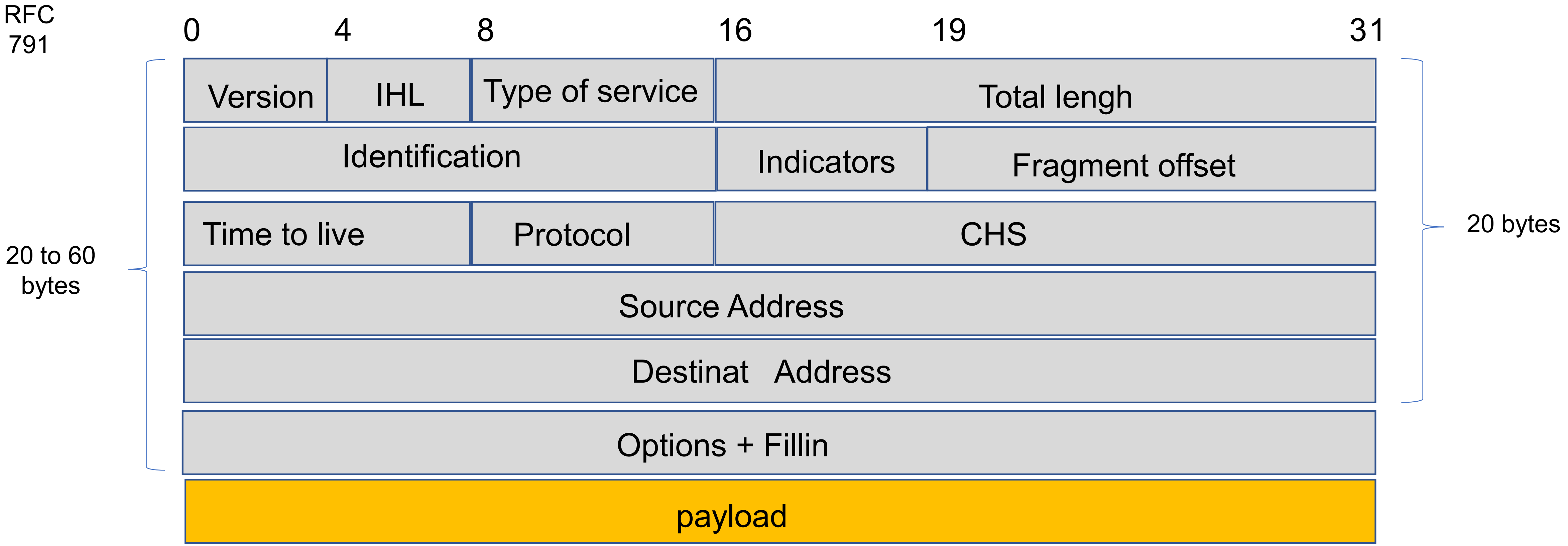
- Server: r2d2.jvnc.net
- Address:128.121.50.2

143.50.121.128. in – addr.arpa host name = abc.jvnc.net

This is done in a table that is independent of the correspondences between names and addresses.



IP ADDRESS



RFC 791

0 4 8 16 19 31

Version | IHL | Type of service | Total lengh

Identification | Indicators | Fragment offset

Time to live | Protocol | CHS

Source Address

Destinat Address

Options + Fillin

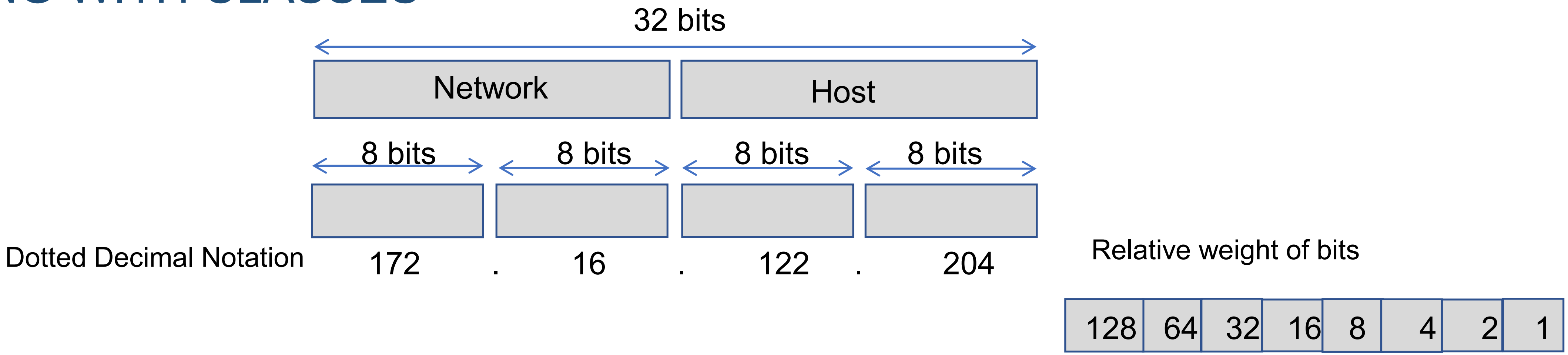
payload

20 to 60 bytes

20 bytes

Every host and router on the Internet has an IP address, which encodes its network number and host number. The combination is unique: no two machines have the same IP address. All IP addresses are 32 bits long and are used in the Source Address and Destination Address fields of IP packets. It is important to mention that an IP address does not really refer to a host. It actually refers to a network interface, so if a host is on two networks, it must have two IP addresses. However, in practice, most hosts are located on a network and therefore have an IP address. IP address format:The 32-bit address field consists of two parts: a network or link number (which identifies the network portion of the address) and a host number (which identifies a host on the network segment). See figure below.

ADDRESSING WITH CLASSES

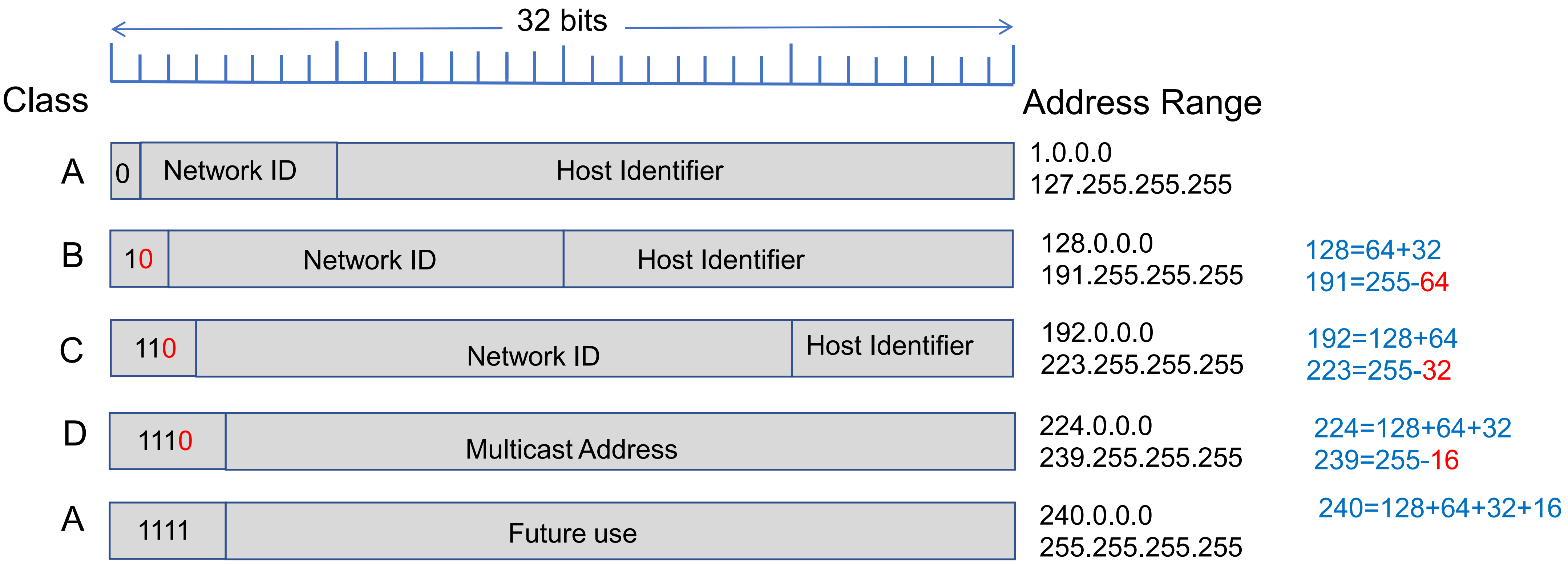


128	64	32	16	8	4	2	1

Addressing with classes (Classful)

For several decades, IP addresses were divided into five categories, which are listed in the following figure. This assignment has been called classful addressing. It is no longer used, but it is still common to find references in the literature. The address is encoded to allow a variable assignment of bits to specify the network and the host. This encoding scheme provides flexibility in assigning addresses to hosts and allows a mix of network sizes in a set of networks.

ADDRESSING WITH CLASSES



32 bits

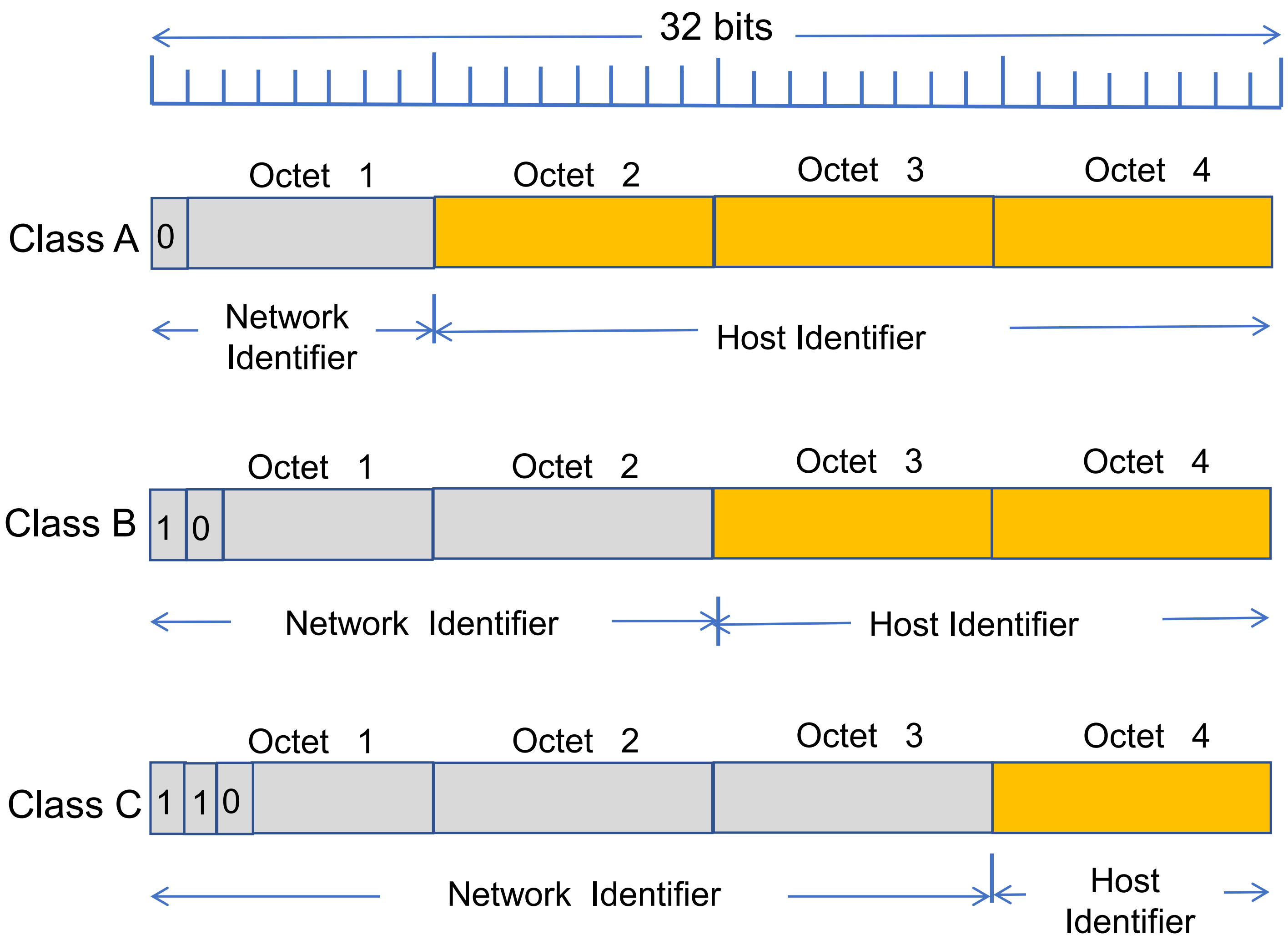
Class				Address Range	
A	0	Network ID	Host Identifier	1.0.0.0 127.255.255.255	
B	10	Network ID	Host Identifier	128.0.0.0 191.255.255.255	128=64+32 191=255-64
C	110	Network ID	Host Identifier	192.0.0.0 223.255.255.255	192=128+64 223=255-32
D	1110	Multicast Address		224.0.0.0 239.255.255.255	224=128+64+32 239=255-16
A	1111	Future use		240.0.0.0 255.255.255.255	240=128+64+32+16

10.0.0.0/8 is class A

10.0.0.0 to 10.255.255.255

10.1.1.1 belongs to that network

ADDRESSING WITH CLASSES



32 bits

Class A

Octet 1 | Octet 2 | Octet 3 | Octet 4

0

Network Identifier

Host Identifier

Class B

Octet 1 | Octet 2 | Octet 3 | Octet 4

1 0

Network Identifier

Host Identifier

Class C

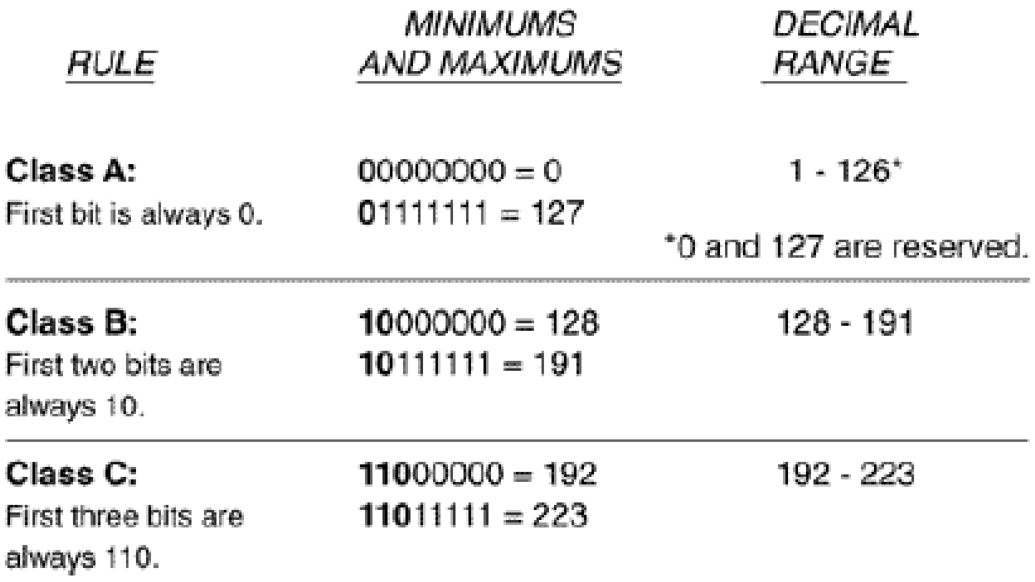
Octet 1 | Octet 2 | Octet 3 | Octet 4

1 1 0

Network Identifier

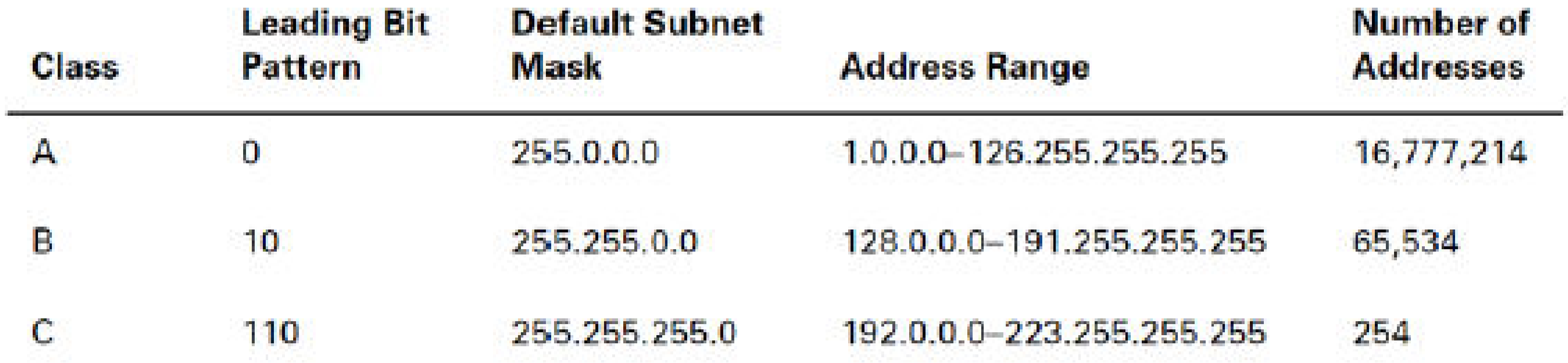
Host Identifier

First Octet rule



RULE	*MINIMUMS AND MAXIMUMS*	*DECIMAL RANGE*
Class A: First bit is always 0.	00000000 = 0 01111111 = 127	1 - 126* *0 and 127 are reserved.
Class B: First two bits are always 10.	10000000 = 128 10111111 = 191	128 - 191
Class C: First three bits are always 110.	11000000 = 192 11011111 = 223	192 - 223

Full range of addresses per class and number of addresses per network:



Class	Leading Bit Pattern	Default Subnet Mask	Address Range	Number of Addresses
A	0	255.0.0.0	1.0.0.0–126.255.255.255	16,777,214
B	10	255.255.0.0	128.0.0.0–191.255.255.255	65,534
C	110	255.255.255.0	192.0.0.0–223.255.255.255	254

Natural mask for class A, B and C network addresses

Class	Mask	decimal notation
A	11111111 00000000 00000000 00000000	255.0.0.0
B	11111111 11111111 00000000 00000000	255.255.0.0
C	11111111 11111111 11111111 00000000	255.255.255.0

The following figure shows how, for a given IP address, the network mask is used to determine the network address. The mask has a 1 in each bit position corresponding to a network bit of the address and a 0 in each bit position corresponding to a host bit. Because 172.21.35.17 is a class B address, the mask must have the first two octets all set to 1 and the last two octets, the host part, set all to 0. As the table shows, this mask can be represented in dotted decimal notation as: 255.255.0.0.

ADDRESSING WITH CLASSES

Truth Table for Boolean AND:

0 AND 0 = 0
0 AND 1 = 0
1 AND 0 = 0
1 AND 1 = 0

	0	1
0	0	0
1	1	1

IP Address IP: 1010 1100 0001 0101 0010 0011 0001 0001 = 172.21.35.17
Mask: 1111 1111 1111 1111 0000 0000 0000 0000 = 255.255.0.0

Ntework Address: 1010 1100 0001 0101 0000 0000 0000 0000 = 172.21.0.0

Private and Public IP addresses

RFC 1918 has defined an IP address space for private use. Companies can use them internally whenever they want. The only rule is that no packet containing these addresses can appear on the Internet itself. The three reserved ranges are:

- Class A network: 1.0.0.0 to 10.255.255.255 aprox 10,4 millones
- Class B network: 172.16.0.0 to 172.31.255.255 aprox 0,9 millones
- Class C network: 192.168.0.0 to 192.168.255.255 aprox 65 mil

SUBNET

As we have seen, all hosts on a network must have the same network number. This property of IP addressing can cause problems as networks grow. For example, consider a university that started with a class B network used by the Dept. of Computer Science for your Ethernet computers. A year later, the Dept. of Electrical Engineering wanted to connect to the Internet, so he purchased a repeater to extend the Ethernet to his building. As time went on, many other departments acquired computers and the limit of four repeaters per Ethernet was quickly reached. A different organization was required.

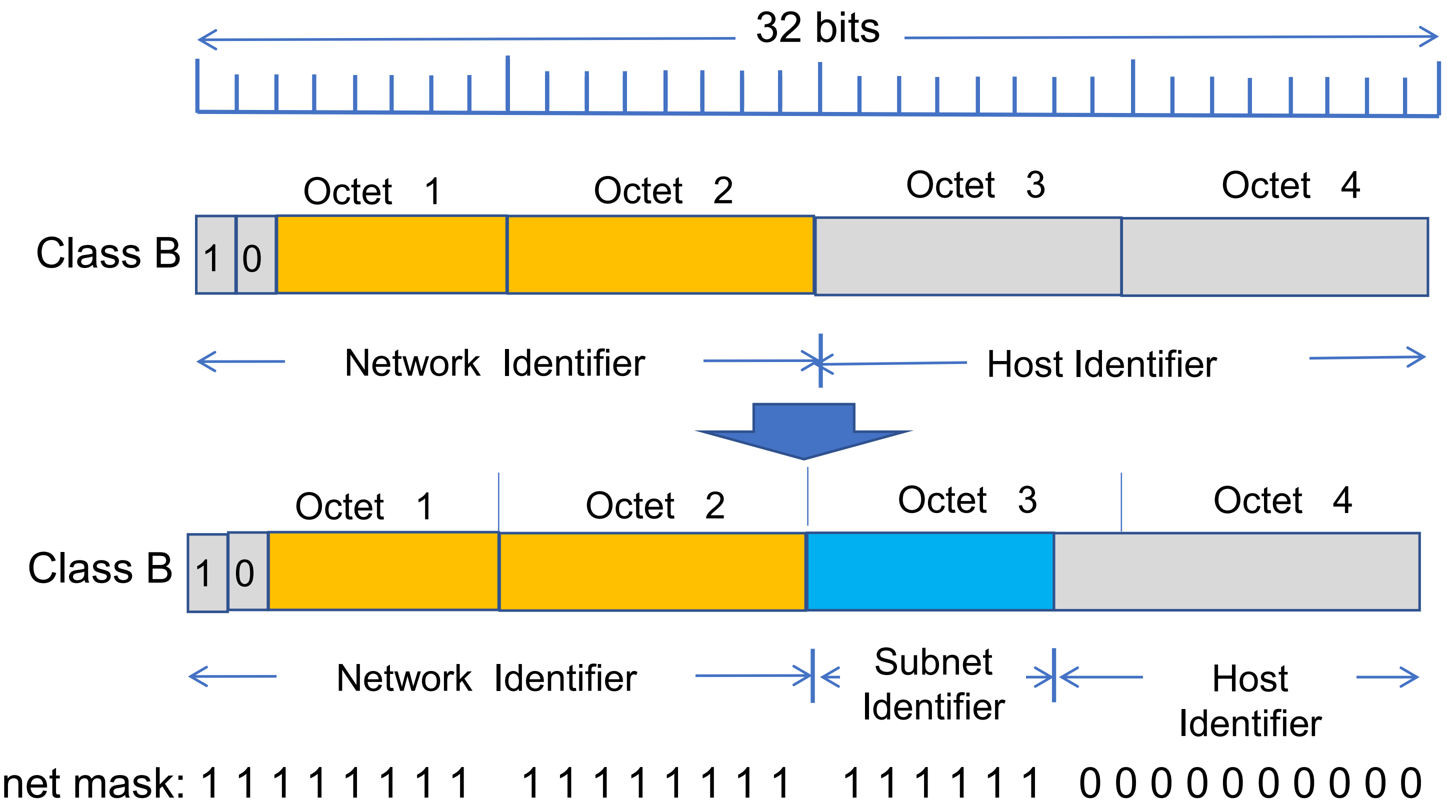
Obtaining a second network address would be difficult because network addresses are scarce and the university already has enough addresses for approximately 60,000 hosts. The problem is the rule that a single class A, B, or C address refers to a network, not a collection of LANs. As more and more organizations found themselves in this situation, a small change was made to the addressing system to handle such a situation.

The solution to this problem is to allow a network to be split into multiple parts for internal use, but still act as a single network to the outside world. A subnet is a subset of a class A, B, or C network. As explained above, IP addresses consist of a network portion and a host portion, representing a static two-level hierarchical addressing model (networks &hosts).

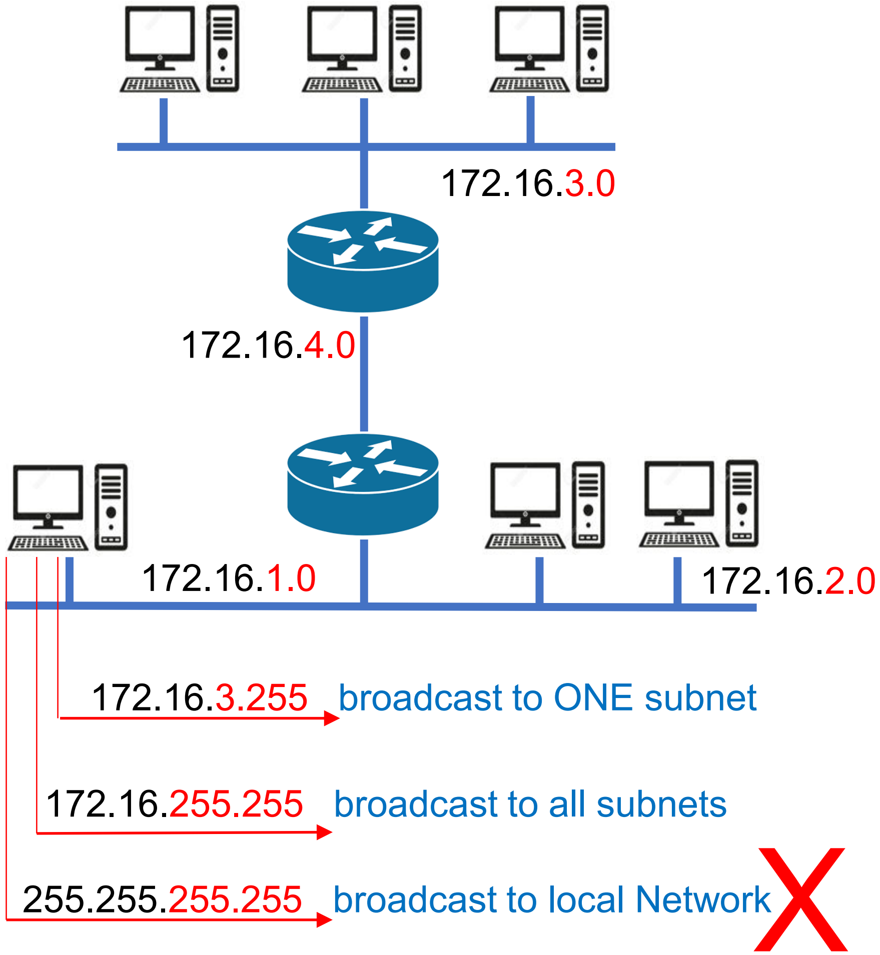
Subnetting IP introduces a third level of hierarchy with the concept of a network (or subnet) mask. The subnet mask determines which portion of our IP address belongs to the subnet portion and which portion belongs to the host portion.

ADDRESSING WITH CLASSES

A class B network divided into 64 subnets



SUBNET &BROADCAST


172.16.3.0
172.16.4.0
172.16.1.0
172.16.2.0
172.16.3.255 broadcast to ONE subnet
172.16.255.255 broadcast to all subnets
255.255.255.255 broadcast to local Network
X

Broadcast

Flooded broadcasts (255.255.255.255) are not spread. They are considered local broadcast (to the entire local network).

The messages of Broadcast must reach all hosts on the network. The broadcast address consists of all ones in the IP address.

Broadcasts directed to a specific network are allowed and are broadcasted by the router.
These targeted broadcasts have all the bits of the address part corresponding to the host in 1.

SUBNET &BROADCAST

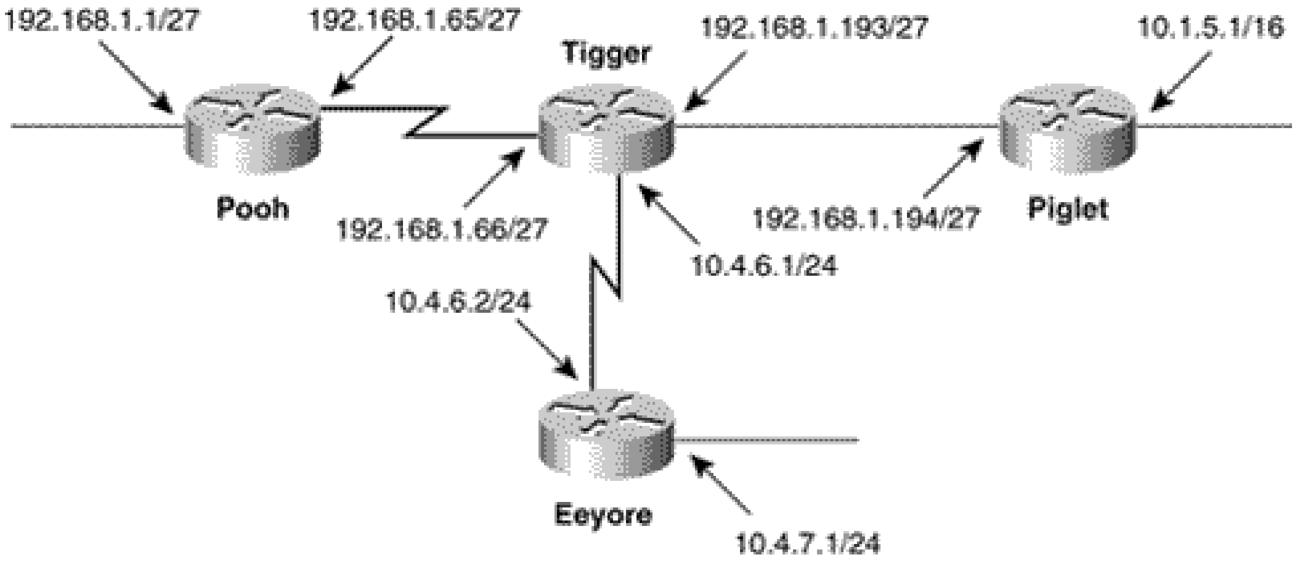
You can also broadcast messages to all hosts on a subnet and all subnets on a network. To broadcast a message to all hosts on an individual subnet, the host portion of the address must have all bits set to 1. The following example will broadcast messages to all hosts on network 172.16, subnet 3:
All hosts on the specific subnet: 172.16.3.255.

Additionally, messages can be broadcast to all hosts on all subnets within an individual network. To broadcast a message to all hosts in all subnets of an individual network, the host and subnet parts of the address must have all bits set to 1.The following example will broadcast a message to all hosts on all subnets on the 172.16 network:

All hosts on all subnets on a specific network: 172.16.255.255

Fixed length mask (FLSM), variable length mask (VLSM)

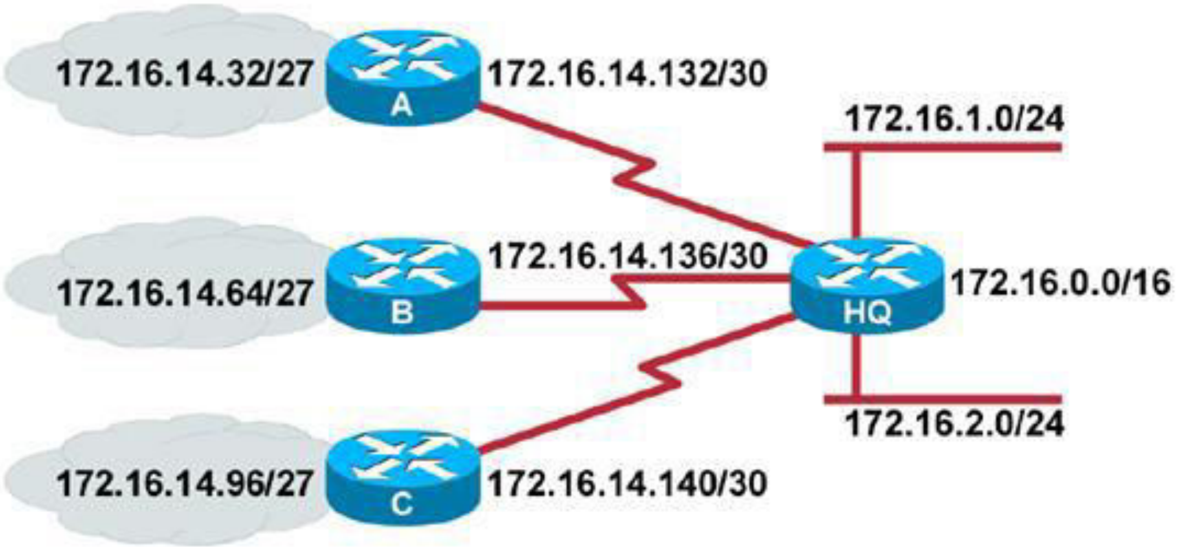
Classful routing protocols, such as RIPv1 and IGRP, cannot easily determine routing in variably non-contiguous subnetted networks, such as the one shown in the figure:



VARIABLE LENGTH SUBNET MASK

It was developed to allow multiple levels of IP subnet addressing within a single network. This strategy can be used only when it is supported by the routing protocol in use, such as OSPF, EIGRP. VLSM is the key in large networks. Knowledge of VLSM capabilities is important when planning a large network.

What is Variable Length Subnet Mask?



The 172.16.14.0/24 subnet is in turn divided into other subnets:

- Subnets with /27 mask
- From a free /27 subnet they are made/30 subnets.

VLSMs provide the ability to include more than one subnet mask within a network and the ability to create subnets from an already subnetted network. VLSM offers the following benefits:

- More efficient use of IP addressing: without the use of VLSMs, companies must implement a single subnet mask within some of the network number classes A, B, or C.
- Ability to use route summarization (summarization): VLSM allows a routing plan more hierarchical by allowing route summary within routing tables.

VARIABLE LENGTH SUBNET MASK

C class potential subnets

Last Octet	Binary representation	Number of subnets	Number of hosts
128	1000 0000	2	128
192	1100 0000	4	64
224	1110 0000	8	32
240	1111 0000	16	16
248	1111 1000	32	8
252	1111 1100	64	4

[*] Note that the Number of Hosts field includes the network number and directed broadcast addresses.

CIDR (Classless Inter-Domain Routing)
CIDR is a mechanism developed to help alleviate the problem of IP address exhaustion and routing table growth. The idea behind CIDR is that blocks of multiple class C addresses can be combined, or aggregated, in order to create a larger pool of IP addresses (i.e., with more hosts allowed). Blocks of class C network numbers are assigned to each network service provider. Organizations that use the network service provider for Internet connectivity are assigned subsets of the service provider's address space, as appropriate.

These class C addresses can be summarized in routing tables, which implies that there are fewer route publications.
CIDR is described in more detail in RFCs 1518 and 1519.

VARIABLE LENGTH SUBNET MASK

The basic concept of CIDR, which is described in RFC 1519, is to allocate the remaining IP addresses in blocks of variable size, regardless of classes. In CIDR, an IP network is represented by a prefix, which is the IP address of a network, followed by a slash and finally an indication of the number of leftmost contiguous bits corresponding to its network mask that is associated with the network address.
For example, consider the network 198.32.0.0 with the prefix /16, written as 198.32.0.0/16. The /16 indicates that there are 16 mask bits to count from the left end. This stands for IP network 198.32.0.0 with a netmask of 255.255.0.0.

If a site needs, say, 4000 addresses, it is given a block of 4096 addresses (12 bits for host). To comply with the request, 16 blocks of class C addresses must be assigned. Suppose that the following blocks are assigned:

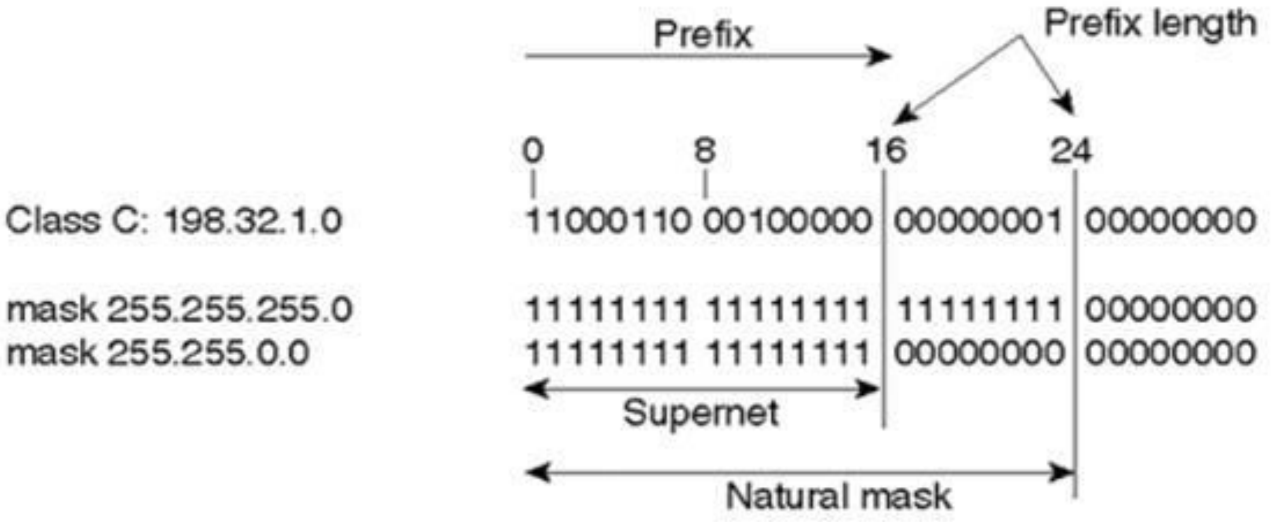
192.168.0.0/24
192.168.1.0/24
192.168.2.0/24
192.168.3.0/24
192.168.4.0/24
-
-
192.168.15.0/24

$16 = 4^2$
16bits -12 bits = 4

The use of CIDR allows, independent of class, to assign the address 192.168.0.0/20 to satisfy the request of 4000 addresses.When the router publishes the available addresses, it publishes the route summary instead of separately publishing the 16 Class C networks. By publishing 192.168.0.0/20, the router indicates that it can reach all destination addresses that have the first 20 bits same as the first 20 bits of address 192.168.0.0(Note: route summary is discussed below)

SUPERNET

A network is referred to as a supernet when the prefix netmask contains fewer bits than the natural mask for that class. A class C network, for example 198.32.1.0, has a natural mask 255.255.255.0, which corresponds to /24 in CIDR notation.The representation 198.32.0.0 255.255.0.0 can also be represented as 198.32.0.0/16, both of which have a shorter mask than the natural mask (16 is shorter than 24); Therefore the network is known as a supergrid.



	0	8	16	24
Class C: 198.32.1.0	11000110	00100000	00000001	00000000
mask 255.255.255.0	11111111	11111111	11111111	00000000
mask 255.255.0.0	11111111	11111111	00000000	00000000

198.32.1.0 255.255.255.0 <==> 198.32.1.0/24

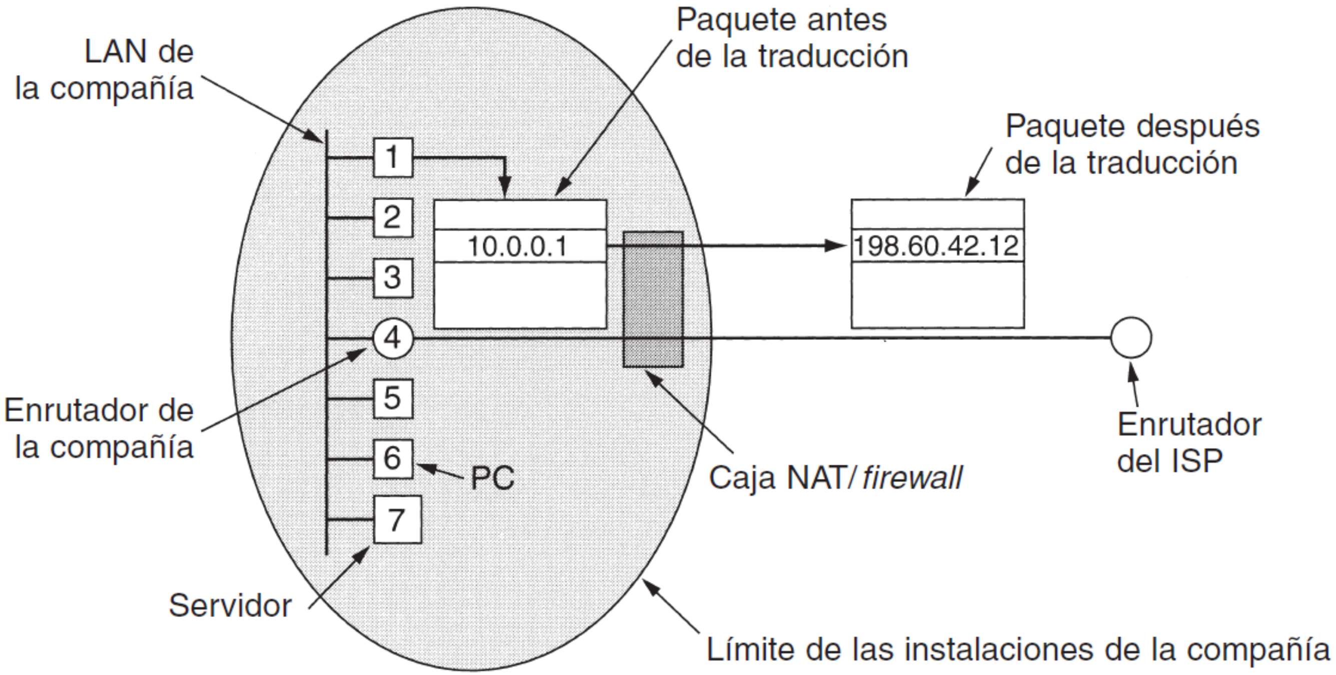
198.32.0.0 255.255.0.0 <==> 198.32.0.0/16

This notation provides a mechanism to easily group all the routes most specific to 198.32.0.0/16 (for example, 198.32.0.0, 198.32.1.0, 198.32.2.0, and so on) into an advertisement referred to as an aggregate. It's easy to get confused by all the terminology, especially since the terms aggregate, CIDR block, and supernet are used interchangeably. Generally, these terms all indicate that a group of contiguous IP networks has been summarized in a route advertisement. More precisely, CIDR is represented by the notation <prefix/length>, supernets have a prefix length shorter than the natural mask, and aggregates represent any summary (summarization) path.

NAT (Network Address Translation) and PAT (Port Address Translation)

The problem of running out of IP addresses is not a theoretical problem that could occur at some point in the distant future. It is happening here and right now. The long-term solution is for the entire Internet to migrate to IPv6, which has 128-bit addresses. This transition is happening slowly, but the entire process will be complete in a matter of years. As a result, some people felt that a quick, short-term fix was needed. This arrangement arose in the form called NAT (Network Address Translation) described in RFC 3022.

The basic idea of NAT is to assign a single public IP address to each company (or at most, a small number) for Internet traffic. Within the company, each computer has a unique private IP address that is used to route internal traffic. When a packet leaves the company and goes to the ISP, an address translation is performed.



NAT table	
Local IPV4 Address (private) internal	Global IPV4 Address (public) internal
10.0.0.1	198.60.42.12
10.0.0.2	198.60.42.12

NAT (Network Address Translation) and PAT (Port Address Translation)

So far we have ignored one small detail:

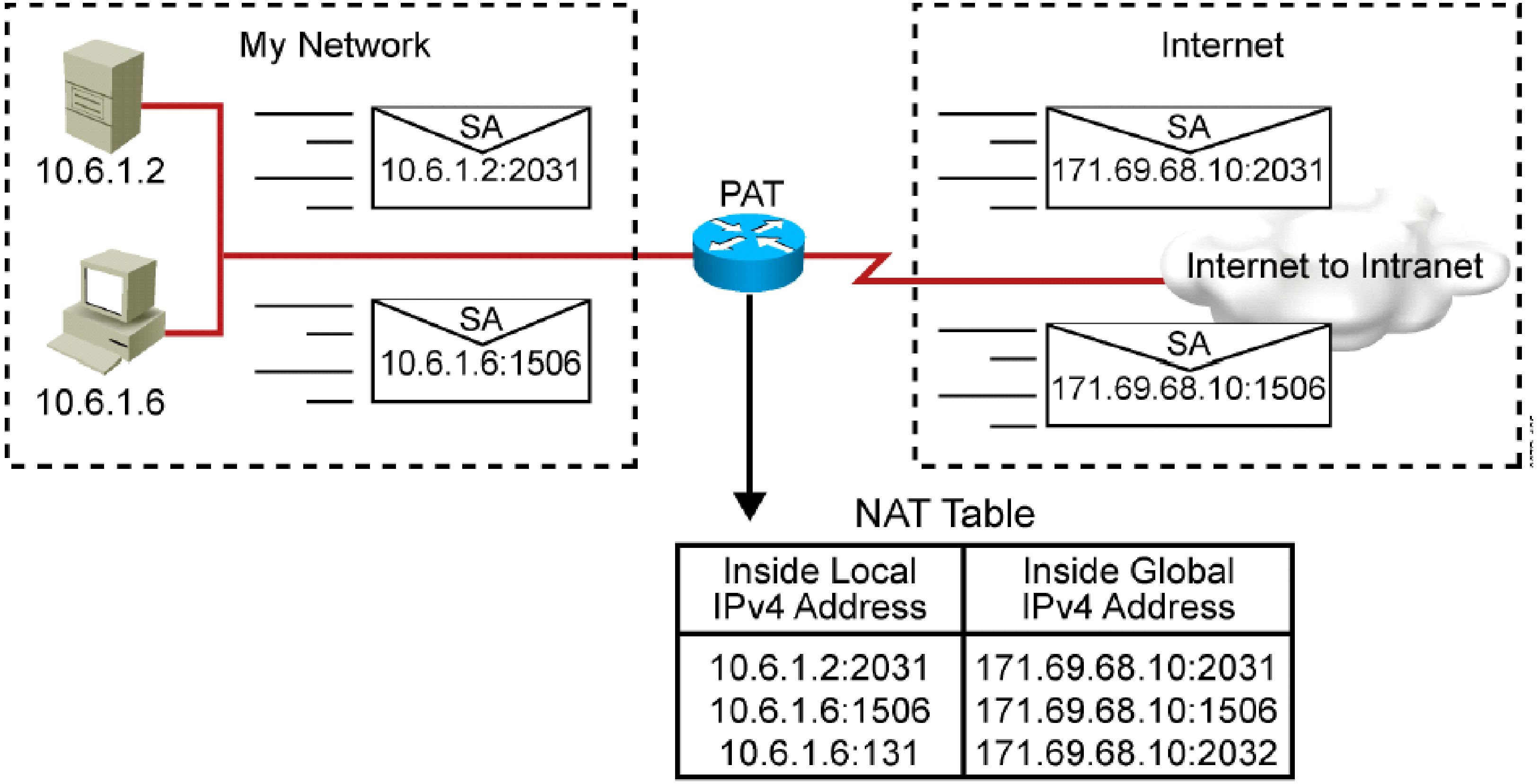
When the response comes back (e.g. from a Web server), it is naturally directed to 198.60.42.12, so how does the NAT box now know which address to replace it with? Here is the problem with NAT. If there were a spare field in the IP header, that field could be used to hold the record of the actual sender, but only 1 bit is left unused. NAT designers observed that most IP packets carry TCP or UDP payloads. When we study TCP and UDP in the next chapter, we will see that they both have headers that contain a source port and a destination port. We will explain TCP ports later, but the same explanation is valid for UDP ports. Ports are 16-bit integers that indicate where the TCP connection begins and ends.

These ports provide the field required to make NAT work. Using the source port field, we can solve our conversion problem.

Whenever an outgoing packet enters the NAT box, the 10.x.y.z source address is replaced with the company's true IP address. Additionally, the TCP source port field is replaced by an index in the translation table of NAT box entry 65,536. This table entry contains the original source port and IP address. Finally, the checksums of the IP and TCP headers are recalculated and inserted into the packet. You need to replace the source port because it could happen that both connections on machines 10.0.0.1 and 10.0.0.2 use port 5000, for example, so the source port is not enough to identify the sending process. When a packet arrives at the NAT box from the ISP, the source port in the TCP header is extracted and used as an index in the NAT box translation table. From the localized entry, the internal IP address and TCP source port are extracted and inserted into the packet. The IP and TCP checksums are then recalculated and inserted into the packet. The packet is then passed to the company's router for normal delivery using the address 10.x.y.z.

NAT (Network Address Translation) and PAT (Port Address Translation)

An example



NAT Table

Inside Local IPv4 Address	Inside Global IPv4 Address
10.6.1.2:2031	171.69.68.10:2031
10.6.1.6:1506	171.69.68.10:1506
10.6.1.6:131	171.69.68.10:2032

Comparison between IPv4 and IPv6

While CIDR and NAT may last a few more years, everyone realizes that the days of IP in its current form (IPv4) are reaching its end.
In addition to these technical problems, there is another issue lurking. Before the decade of 1990, the Internet has been used largely by universities,high-tech industries and the government (especially the Department of Defense of USA). With the explosion of interest in the Internet that began in the middle of the decade1990, in next years it became a much larger group of people, especially people with different needs.

Seeing these problems on the horizon, the IETF began work in 1990 on a new version of IP, one that would never run out of addresses, would solve several other problems, and would be more flexible and efficient as well. Its main goals were:

1. Handle billions of hosts, even with inefficient address space allocation.
2. Reduce the size of routing tables.
3. Simplify the protocol, to allow routers to process packets faster.
4. Provide greater security (authenticity and confidentiality verification) than the current IP.
5. Pay more attention to the type of service, especially with real-time data.
6. Assist multicast by allowing scope specification.
7. Enable a host to be mobile without changing its address.
8. Allow the protocol to evolve.
9. Allow the old and new protocol to coexist for years

Three of the best proposals were published in IEEE Network (Deering, 1993; Francis, 1993,and Katz and Ford, 1993). After much analysis, revision and intrigue, a modified version was selected of the combination of the Deering and Francis proposals, now called SIPP (Protocol Simple Internet Enhanced), and was given the IPv6 designation.

IP V6

IPv6 gets the job done pretty well: it keeps the good features of IP, discards and reduces the bad ones, and adds new ones where they are needed. In general, IPv6 is not compatible with IPv4, but it is **compatible with all other Internet protocols,** including TCP, UDP, ICMP, IGMP, OSPF, BGP, and DNS, sometimes with minor modifications (mainly to handle larger addresses).

In principle, and most importantly, IPv6 has larger addresses than IPv4; are **16 bytes long**, which solves the problem they were meant to solve: providing a quantity Virtually unlimited number of Internet addresses.

The second main improvement of IPv6 is the **simplification of the header**, which contains only 7 fields. (against 13 in IPv4). This change allows routers to process packets more quickly. and therefore improve the real speed of transport. The third major improvement was better support options.

This change was essential with the new header, as fields that were previously required are now optional. Also, the way options are represented is different, making it easier for routers to ignore options not addressed to them.

This feature **improves packet processing time**. A fourth area in which IPv6 represents an important advance is security. Authentication and privacy are key features of the new IP.

Finally, greater attention has been paid to the **quality of service**. Several weak efforts have been made in the past, but with the current growth of multimedia on the Internet, more effort is required

IP V6 Header

The IPv6 header is shown in the figure below.

- **The Version** field is always 6 for IPv6 (and 4 for IPv4). During the transition period from IPv4 to IPv6, which will probably take a decade, routers will be able to examine this field to know the type of package they have.

- **The Traffic Class** field is used to distinguish between packets with different real-time delivery requirements. A field designed for this purpose has been in the IP since the beginning, but routers have implemented it only sporadically. Experiments are now underway to determine how good it can be for use in multimedia delivery.

- **The Flow Label** field is still experimental, but will be used to allow a source and destination to establish a pseudoconnection with particular properties and requirements. For example, a chain of packets from a process on a certain source host directed to a certain process on a certain destination host may have very strict delay requirements and therefore need reserved bandwidth.

When a packet appears with a non-zero Flow Label, all routers can look it up in their internal tables to see what kind of special treatment it requires. In effect, flows are an attempt to have the best of both worlds: the flexibility of a datagram subnet and the guarantees of a virtual circuit subnet. Each flow is designated by the source address, destination address, and flow number, so many flows can be active at the same time between a given pair of IP addresses.

IP V6 Header

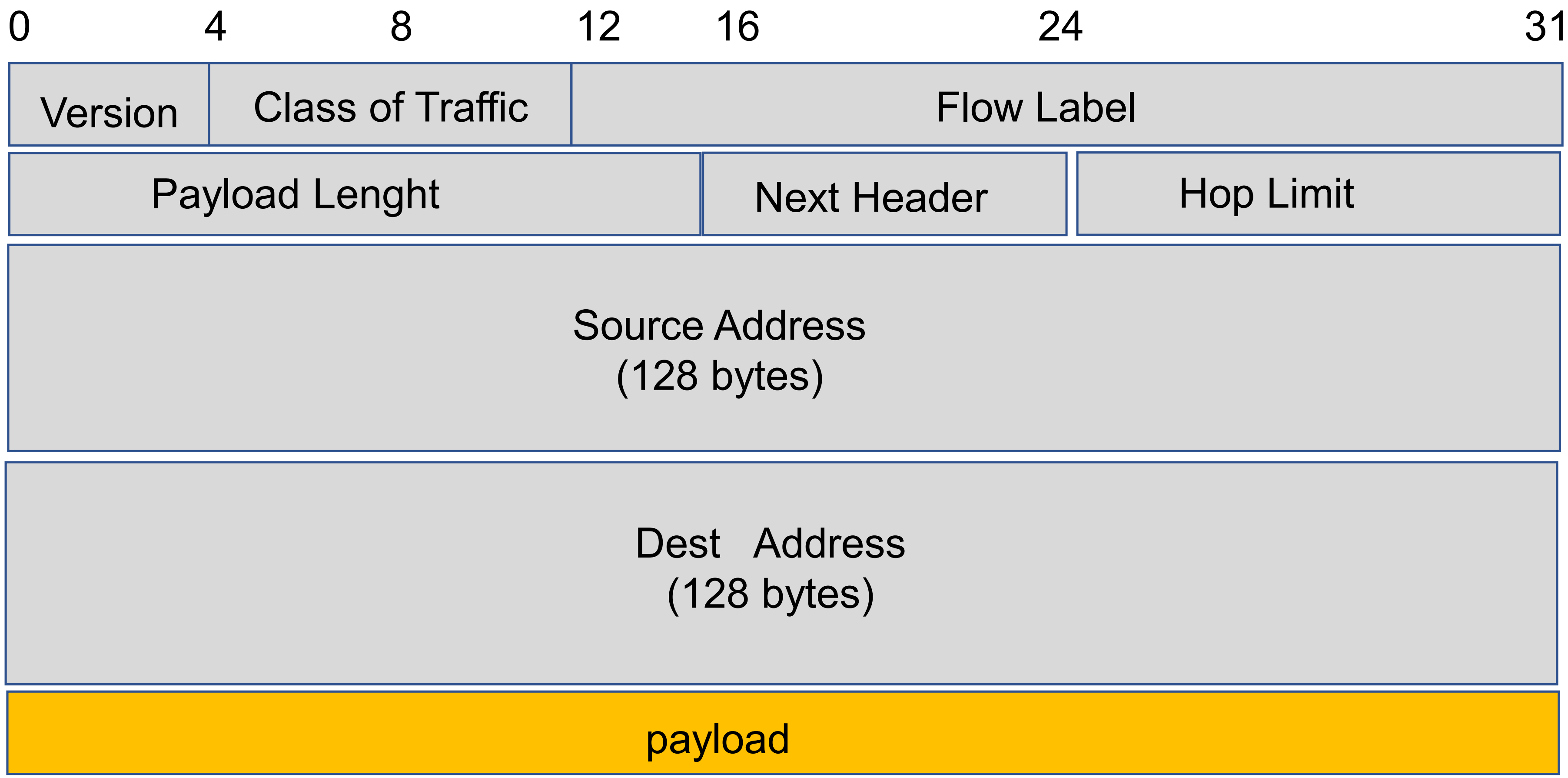
- **The Payload Length** field indicates how many bytes follow the 40-byte header in the figure.

- **The Next Header** field reveals the secret. The reason the header could be simplified is that there may be additional (optional) extension headers. This field indicates which of the six (currently) extension headers, if any, follows this one. If this header is the last IP header, the Next Header field indicates the handler of transport protocol (e.g. TCP, UDP) to which the packet will be delivered.

- **The Hop Limit** field is used to prevent packets from living forever. In practice it is equal to the IPv4 Time to Live field, that is, a field that is decreased with each hop.

- **Source Address and Destination Address** fields. Deering's original proposal, the SIPP, used addresses of 8, but during the review period many people felt that with addresses of 8 of 8 bytes, in a few decades IPv6 would run out of addresses, and that with 16 byte addresses they would never run out.

IP V6 Header



0 4 8 12 16 24 31

Version | Class of Traffic | Flow Label

Payload Lenght | Next Header | Hop Limit

Source Address
(128 bytes)

Dest Address
(128 bytes)

payload

IPV4 Header Vs IPV6 Header

It is instructive to compare the IPv4 header with the IPv6 header (see figure below) to see what has been left out of IPv6. The IHL field is gone because the IPv6 header has a fixed length.

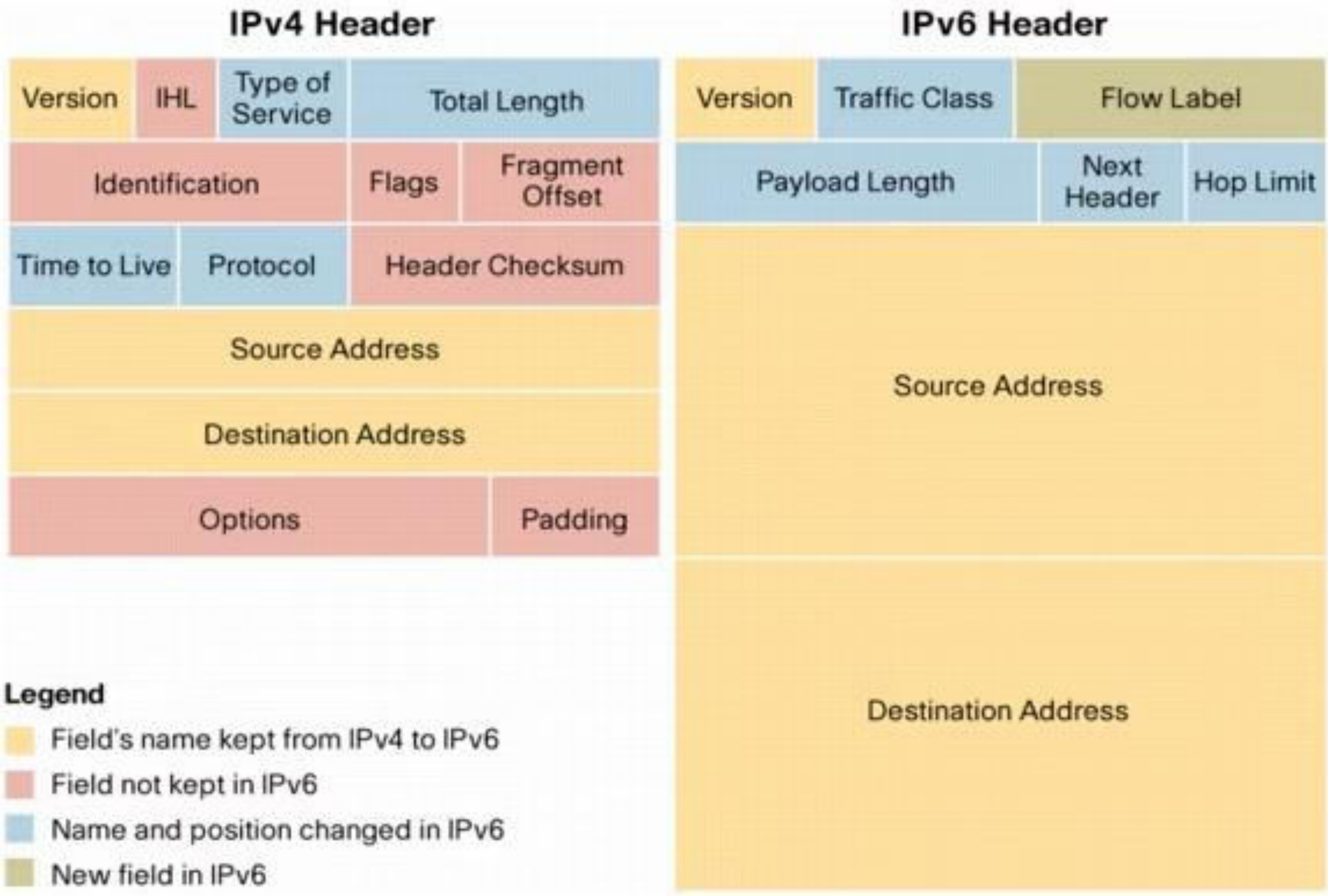
The Protocol field was removed because the Next Header field indicates what follows the last IP header (for example, a UDP or TCP segment).

All fields related to fragmentation have been removed since IPv6 has a different approach to fragmentation.
To start, all IPv6-compliant hosts must dynamically determine the datagram size.
Also, the minimum was increased from 576 to 1280 to allow 1024 bytes of data and multiple headers. This rule makes fragmentation less likely to occur. Additionally, when a host sends an IPv6 packet that is too large, instead of fragmenting it, the router that is unable to forward it returns an error message. This message instructs the host to split all future packets to that destination.

It is much more efficient to make the host send packets of the correct size from the beginning than to have routers to fragment them on the fly.

Finally, the Checksum field disappears, because its calculation greatly reduces performance. With today's reliable networks, in addition to the fact that the data link layer and transport layers typically have their own checksums, the benefit of another checksum was not worth the performance cost it generated. Removing these features has left a compact and solid network layer protocol. Therefore, the goal of IPv6 (a fast and flexible protocol with plenty of address space) has been met with this design.

IPV4 Header Vs IPV6 Header



IPv4 Header

Version | IHL | Type of Service | Total Length

Identification | Flags | Fragment Offset

Time to Live | Protocol | Header Checksum

Source Address

Destination Address

Options | Padding

IPv6 Header

Version | Traffic Class | Flow Label

Payload Length | Next Header | Hop Limit

Source Address

Destination Address

Legend

- 🟨 Field's name kept from IPv4 to IPv6
- 🟥 Field not kept in IPv6
- 🟦 Name and position changed in IPv6
- 🟫 New field in IPv6

Source: Cisco

IPV6 Extention Headers

However, some of the missing IPv4 fields are sometimes necessary, which is why IPv6 introduced the concept of **an (optional) extension header**. These headers can be used to provide extra information, but encoded in an efficient way. There are currently six types of extension headers defined, which are listed in the figure below. All are optional, but if there is more than one, they should appear just after the fixed header, and preferably in the order listed.

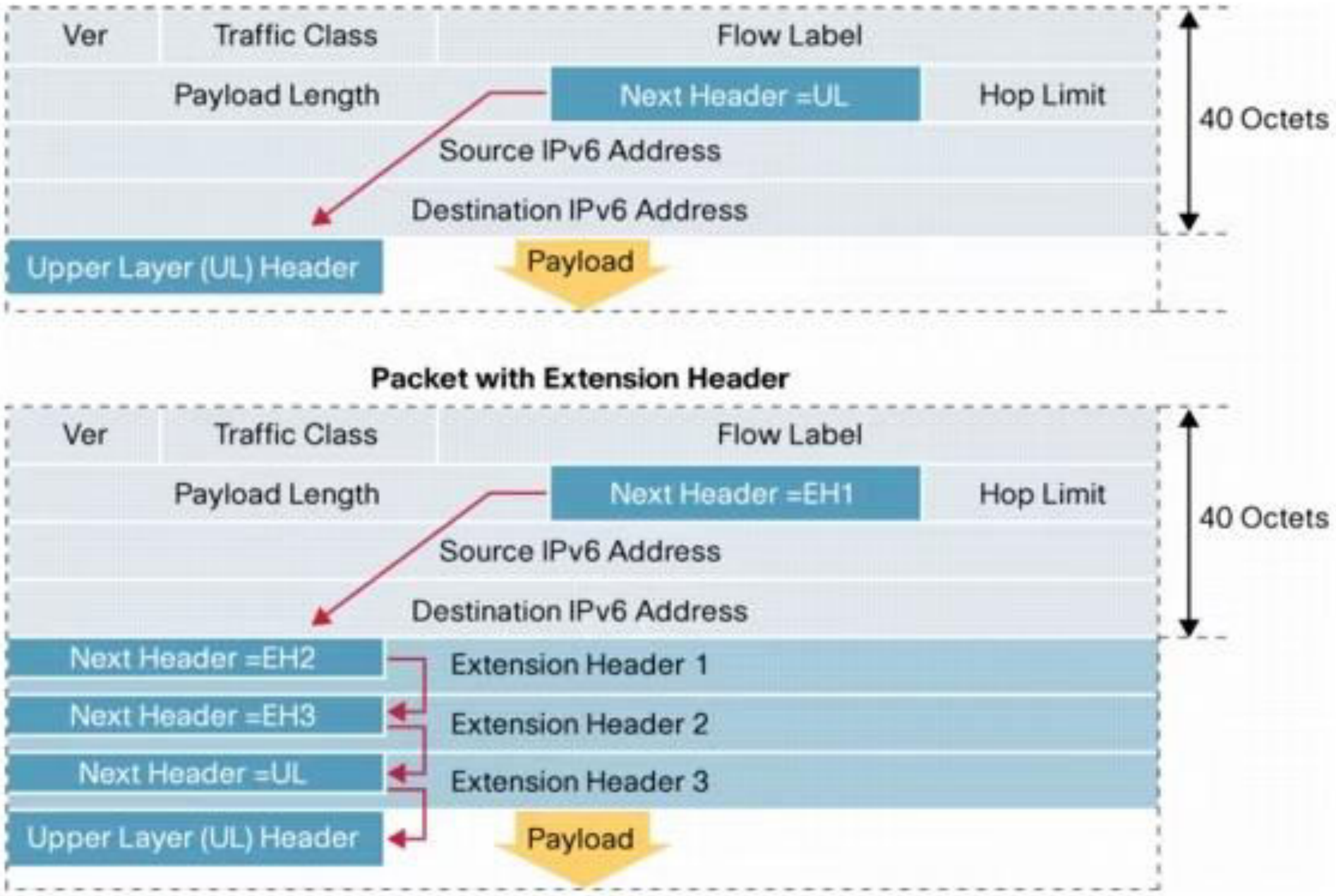
Some of the headers have a fixed format; others contain a variable number of variable-length fields. In these, each element is encoded as a tuple (type, length, value). The Type is a 1-byte field that indicates the option in question. The Type values have been chosen so that the first two bits tell routers that do not know how to process the option, what to do.

The Length is also a 1-byte field, and indicates the length of the value (0 to 255 bytes). The Value is any required information, up to 255 bytes.

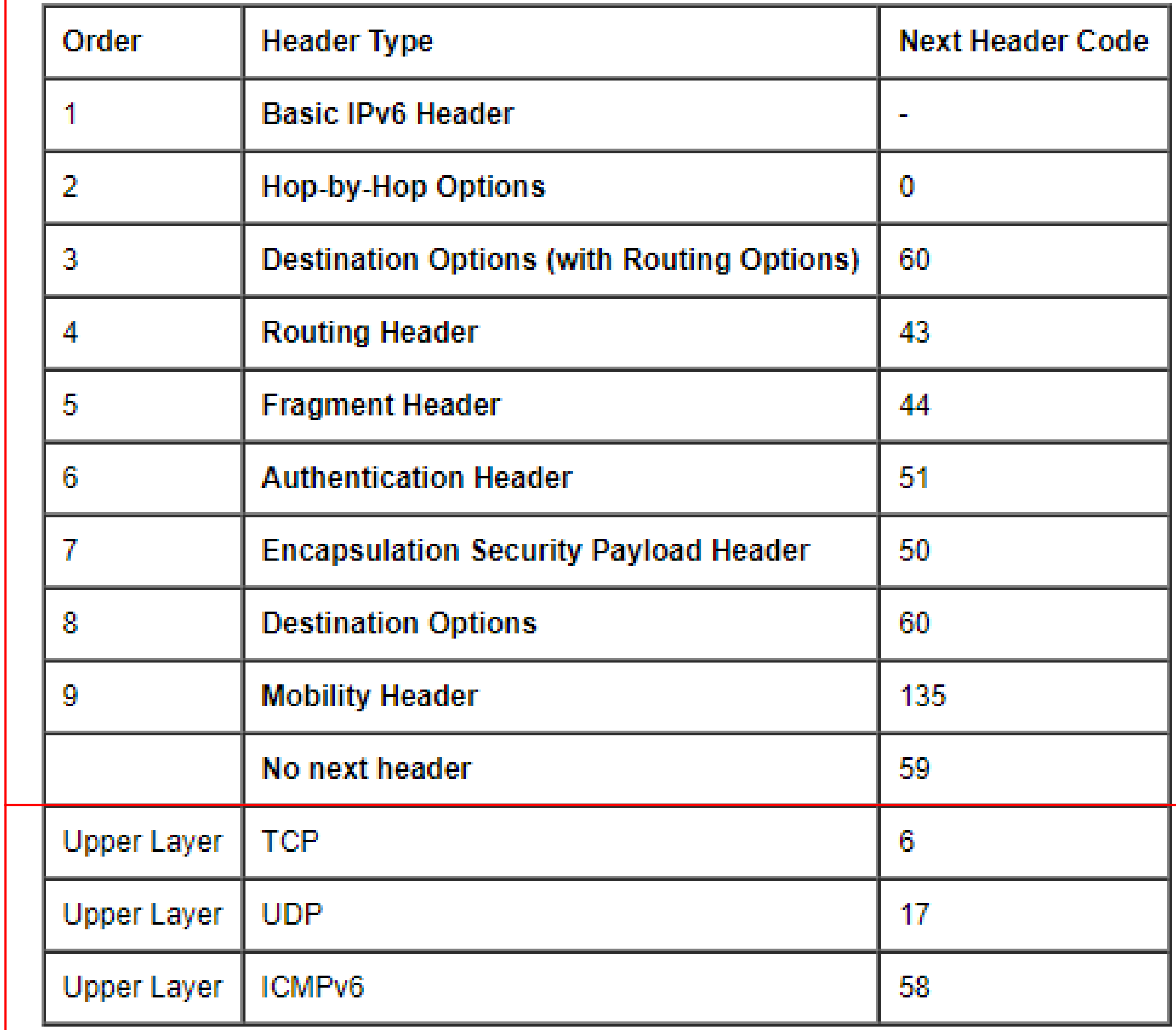
IPv6 extension headers:

Extention Header	Description
Hop-by-hop Options	Diverse Information for routers
Destination Options	Additional information for destination
Routing header	Total or partial route to take
Fragment Header	Datagram fragments management
Authentication Header	Emitting identity verification
Encapsulation Security Payload Header	Information related to encrypted contents

IPV6 Extention Headers



IPV6 Extention Headers



Order	Header Type	Next Header Code
1	Basic IPv6 Header	-
2	Hop-by-Hop Options	0
3	Destination Options (with Routing Options)	60
4	Routing Header	43
5	Fragment Header	44
6	Authentication Header	51
7	Encapsulation Security Payload Header	50
8	Destination Options	60
9	Mobility Header	135
	No next header	59
Upper Layer	TCP	6
Upper Layer	UDP	17
Upper Layer	ICMPv6	58

extention Headers included

No extention Headers

IP V6

The **“Flow Label”** field is still experimental, but will be used to allow a source and a target establish a pseudo-connection with particular properties and requirements. For example, one chain of packets from a process on a certain source host addressed to a certain process on a certain source host destination may have very stringent delay requirements and thus require bandwidth reserved.

SLICING!!!

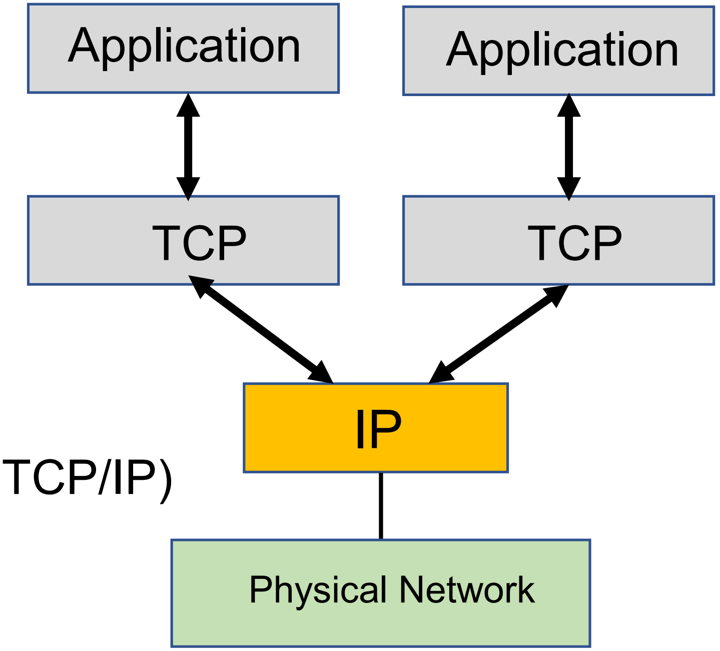
The **“Next Header”** field reveals the secret. The reason why it could be simplified header is that there may be additional (optional) extension headers. This field indicates which of the (currently) six extension headers, if any, follows this one. if this header is the last IP header, the Next Header field indicates the handler transport protocol (for example, TCP, UDP) to which the packet will be delivered.

SIMPLIFY

Transport Level Capabilities

Transport level capabilities
In a protocol architecture, the transport protocol sits above the network or interconnection layer, which provides network-related services, and just below the application layers and other higher layer protocols (see figure below).

The transport protocol provides services to Transport Service (TS) users, such as FTP, SMTP, and TELNET. The local transport entity communicates with some other remote transport entity using the services of some lower layer, such as the Internet Protocol (IP). The general service provided by a transport protocol is the end-to-end transport of data in a manner that isolates the TS user from the details of the underlying communications systems.



DARPA Model (TCP/IP)

The transport layer is not just another layer. It is, together with IP, the center of the entire protocol hierarchy. The task of this layer is to provide reliable and economical transport of data from the source machine to the destination machine, regardless of the physical network or networks in use. Without the transport layer, the entire concept of layered protocols would make little sense.

Transport Level Capabilities

Basic questions:

- The transport protocol provides an end-to-end data transfer service that isolates the upper layers from the details of the intermediate network or networks. A transport protocol can be connection-oriented, as in the case of TCP, or connectionless, as in the case of UDP.

- If the underlying network or interconnection service is unreliable, as is the case with IP, a reliable connection-oriented transport protocol turns out to be very complex. The basic cause of this complexity lies in the need to deal with the variable and relatively high delays experienced between end systems. These delays complicate flow control and error control.

- TCP employs a credit-based flow control technique that is somewhat different from the sliding window flow control found in HDLC. Basically, TCP separates the confirmations from the management of the sliding window size.

- Although TCP's credit-based mechanism was designed for end-to-end flow control, it is also used to assist in congestion control over network interconnections. When a TCP entity detects the presence of congestion on the Internet, it reduces the flow of data it sends over the Internet until it detects relief in congestion.

Introduction to TCP

TCP (Transmission Control Protocol) was specifically designed to provide a reliable end-to-end stream of bytes over an unreliable internetwork. An Internet differs from a single network because various parts could have different topologies, bandwidths, delays, packet sizes, and other parameters. TCP has a design that dynamically adapts to the properties of the Internet and overcomes many types of failures.

TCP was formally defined in RFC 793. As time went on, several errors and inconsistencies were detected, and requirements in some areas changed. These specifications and some bug fixes are detailed in RFC 1122. Some extensions are given in RFC 1323. Every machine that supports TCP has a TCP transport entity, whether it is a library procedure, a user process, or part of the kernel. In all cases, it handles TCP flows and interacts with the IP layer. A TCP entity accepts streams of user data from local processes, splits them into chunks not exceeding 64 KB (in practice, typically 1460 bytes of data that fit into a single Ethernet frame with IP and TCP headers), and sends each fragment as a separate IP datagram. When datagrams containing TCP data arrive at a machine, they are passed to the TCP entity, which reconstructs the original byte streams.

For simplicity, we will sometimes use “TCP” to refer to the TCP transport entity (a piece of software) or the TCP protocol (a set of rules). The context will make it clear what we mean. For example, in the phrase “The user provides the data to TCP”, it is clear that we are referring to the TCP transport entity. The IP layer provides no guarantee that datagrams will be delivered properly, so it is up to TCP to terminate timers and retransmit datagrams as necessary. Arriving datagrams may arrive in the wrong order; It is also up to TCP to reassemble them into messages in the appropriate sequence. In short, TCP should provide the reliability that most users want and that IP does not.

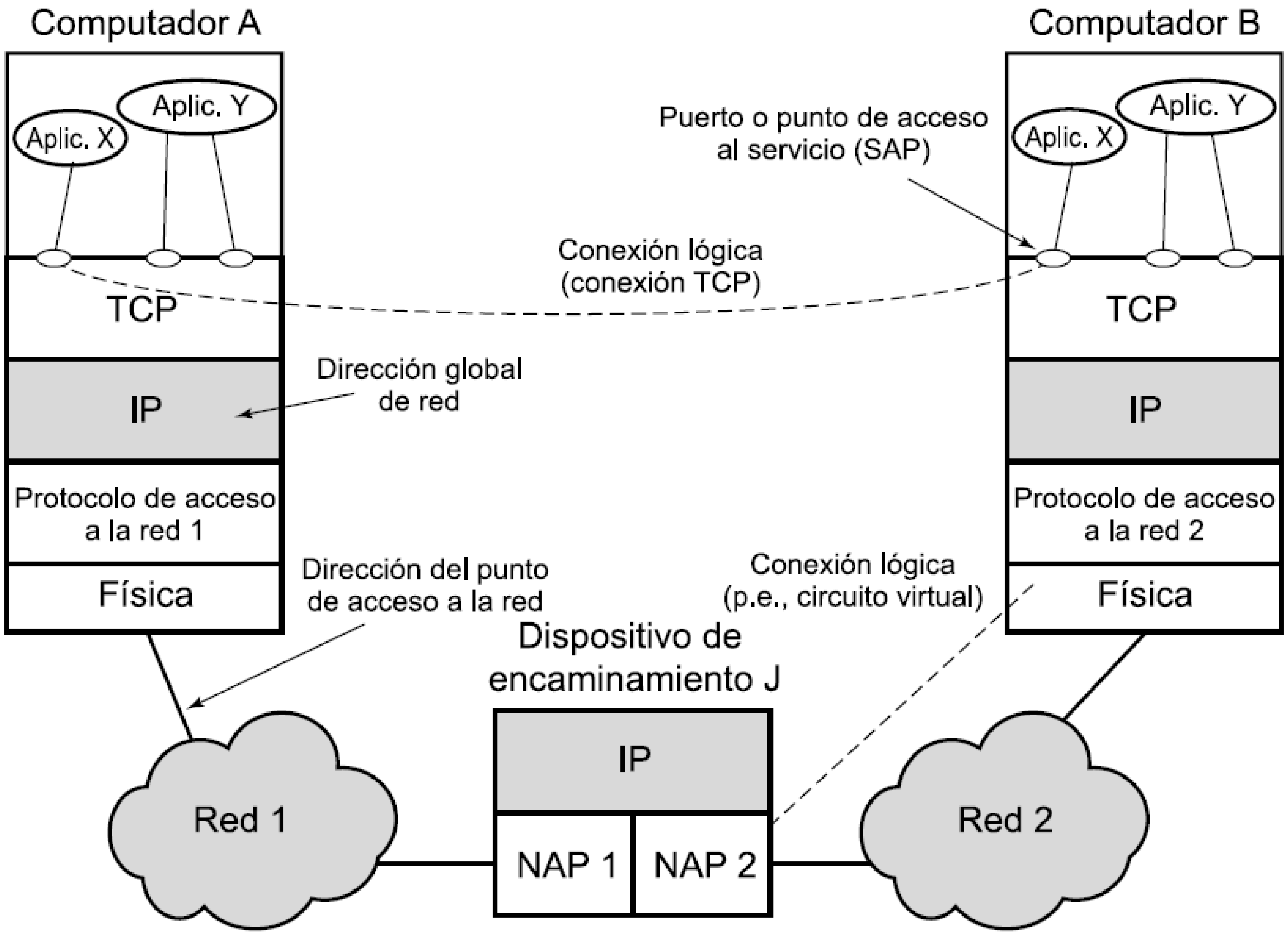
The Transmission Control Protocol (TCP) service model

Transmission Control Protocol (TCP) provides applications with a reliable, connection-oriented service. In other words, TCP provides the appearance of a point-to-point connection. Point-to-point connections have two characteristics:
-They have only one path to the destination. A packet that comes in at one end of the connection cannot be lost, because the only place to go is the other end.
- Packages arrive in the same order in which they are sent.

TCP uses three fundamental mechanisms to perform a connection-oriented service on top of a layer that only provides a connectionless service:

- Packets are tagged with sequence numbers so that the receiving TCP service can put out-of-sequence packets into the correct sequence before delivering them to the destination application.
- TCP uses a system of acknowledgments, checksums, and timers to provide reliability. A receiver can notify a sender when it recognizes that a packet in a sequence has not arrived or has errors, or a sender can assume that a packet has not arrived if the receiver does not send an acknowledgment within a certain period of time after transmission. In both cases, the sender will resend the package in question.
- TCP uses a mechanism called windowing to regulate the flow of packets; windowing decreases the chances of packets being dropped because the buffers at the receiver are full.

TCP service is obtained by having both the server and the client create endpoints, called sockets. Each socket has a number (address), which consists of the host's IP address, and a 16-bit number, which is local to that host, called a port. A port is the TCP name for a TSAP. To obtain TCP service, a connection must be explicitly established between a socket on the sending machine and one on the receiving machine.

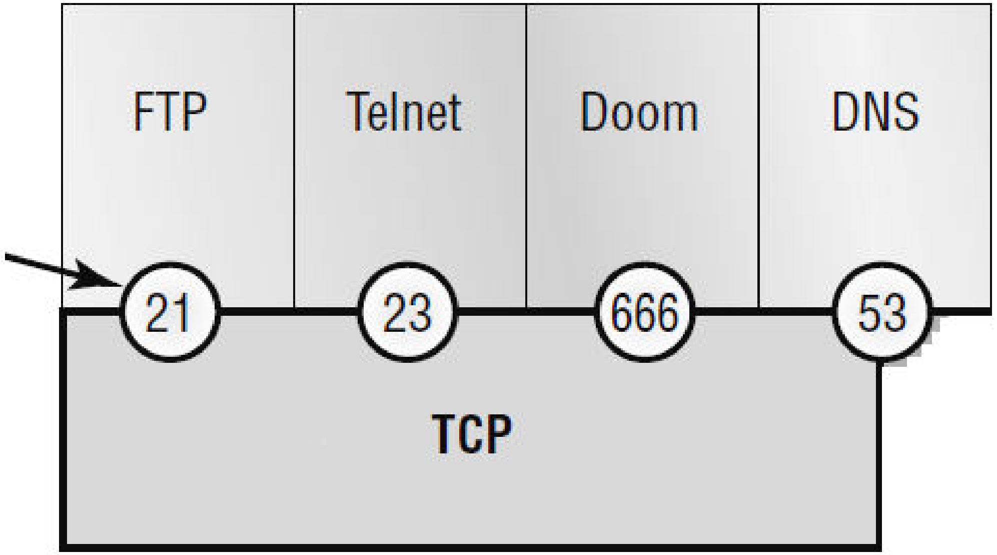

Computador A
Computador B
Aplic. X
Aplic. Y
Aplic. X
Aplic. Y
Puerto o punto de acceso al servicio (SAP)
Conexión lógica (conexión TCP)
TCP
TCP
IP
IP
Dirección global de red
Protocolo de acceso a la red 1
Protocolo de acceso a la red 2
Física
Física
Dirección del punto de acceso a la red
Conexión lógica (p.e., circuito virtual)
Dispositivo de encaminamiento J
IP
NAP 1
NAP 2
Red 1
Red 2

Segments in transit can also be delayed for so long that the sender's timer expires and the segments are retransmitted. Retransmissions may include different byte ranges than the original transmission, requiring careful management to keep track of which bytes have been successfully received at any given time. However, this is feasible since each byte in the stream has its own unique offset. The TCP must be prepared to handle and resolve these problems in an efficient manner. A considerable amount of effort has been invested in optimizing the performance of TCP flows, even in the face of network problems. Below we will study several of the algorithms used by many TCP implementations.

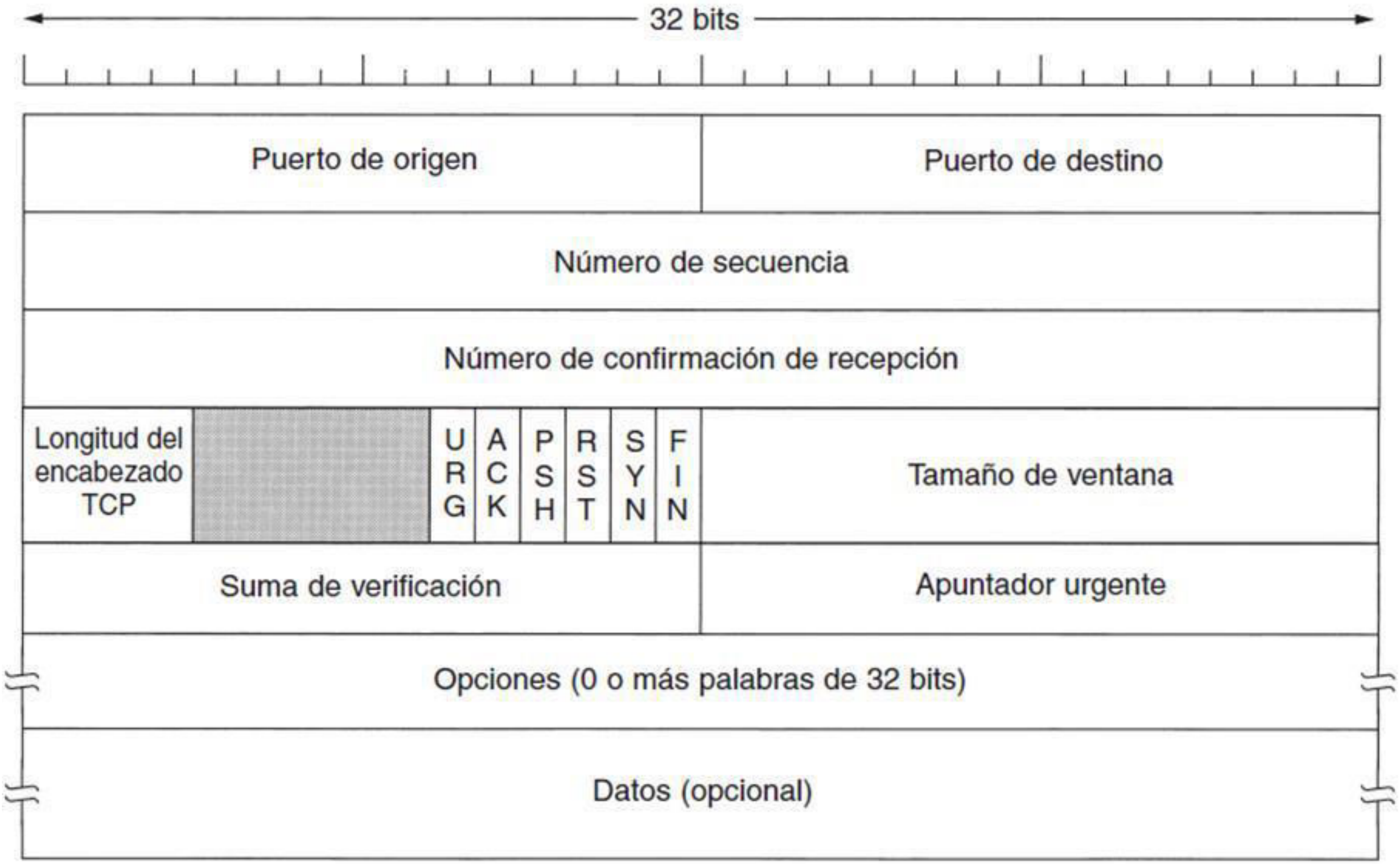
TCP Segment Header Description

The following figure shows the header of a TCP segment. Each segment starts with a 20-byte fixed format header. The sticky header can be followed by header options. After options, if any, they can continue up to 65,535 − 20 − 20 = 65,495 bytes of data, where the first 20 refer to the IP header and the second to the TCP header. Segments without data are legal and are commonly used for receipt confirmations and control messages. The TCP header is analyzed field by field below.

- The Source Port and Destination Port fields identify the endpoints of the connection. A port's address plus its host's IP address form a single 48-bit endpoint. The source and destination endpoints together identify the connection. A socket uniquely identifies each application on an internetwork.



48-bit endpoint: IP address (32 bits)
+ TCP port (16 bits)



32 bits

Puerto de origen

Puerto de destino

Número de secuencia

Número de confirmación de recepción

Longitud del encabezado TCP

URG

ACK

PSH

RST

SYN

FIN

Tamaño de ventana

Suma de verificación

Apuntador urgente

Opciones (0 o más palabras de 32 bits)

Datos (opcional)

- **Sequence Number:** is a 32-bit number that identifies where the encapsulated data fits, from the sender, within a data stream. For example, if the sequence number of a segment is 1343 and the segment contains 512 octets of data, the next segment must have a sequence number of 1343 + 512 + 1 = 1856.
- **Confirmation Number:** is a 32-bit field that identifies the next sequence number that the source expects to receive from the destination. If a host receives a confirmation number that does not match the next sequence number it intends to send (or has sent), it knows not only that the packets were lost, but also which packets were lost.

Note that the Confirmation Number specifies the next expected byte, not the last correctly received byte. Both are 32 bits long because each byte of data is numbered in a TCP stream.

- **TCP Header Length:** indicates the number of 32-bit words contained in the TCP header. This information is necessary because the Options field is variable length, so is the header. Technically, this field actually indicates the beginning of the data in the segment, measured in 32-bit words, but that number is simply the length of the header in words, so the effect is the same.

- Next comes a 6-bit field that is not used. That this field has survived intact for over a decade is testament to how well thought out TCP is. Protocols inferiors would have needed it to correct errors in the original design.
- Now there are six 1-bit indicators.
- **URG** bit is set to 1 if the urgent pointer is in use. The urgent pointer is used to indicate an offset in bytes from the current sequence number in which urgent data is located. This facility replaces interrupt messages. As mentioned before, this facility is a rudimentary mechanism to allow the sender to send a signal to the receiver without implicating TCP in the reason for the interruption.
- **ACK** Bit is set to 1 to indicate that the Acknowledgment Number is valid. If the ACK is 0, the segment does not contain an acknowledgment, so the “32-bit Word Acknowledgment Number field” is ignored.
- **PSH** bit indicates data that must be transmitted immediately. The receiver is hereby carefully requested to deliver the data to the application upon arrival and not buffer it until receipt of a full buffer (which it might otherwise do for efficiency reasons).
- **RST** bit is used to reset a connection that has been confused due to a host crash or other reason; It is also used to reject an invalid segment or an attempt to open a connection. Typically, if you receive a segment with the RST bit on, you have a problem on your hands.

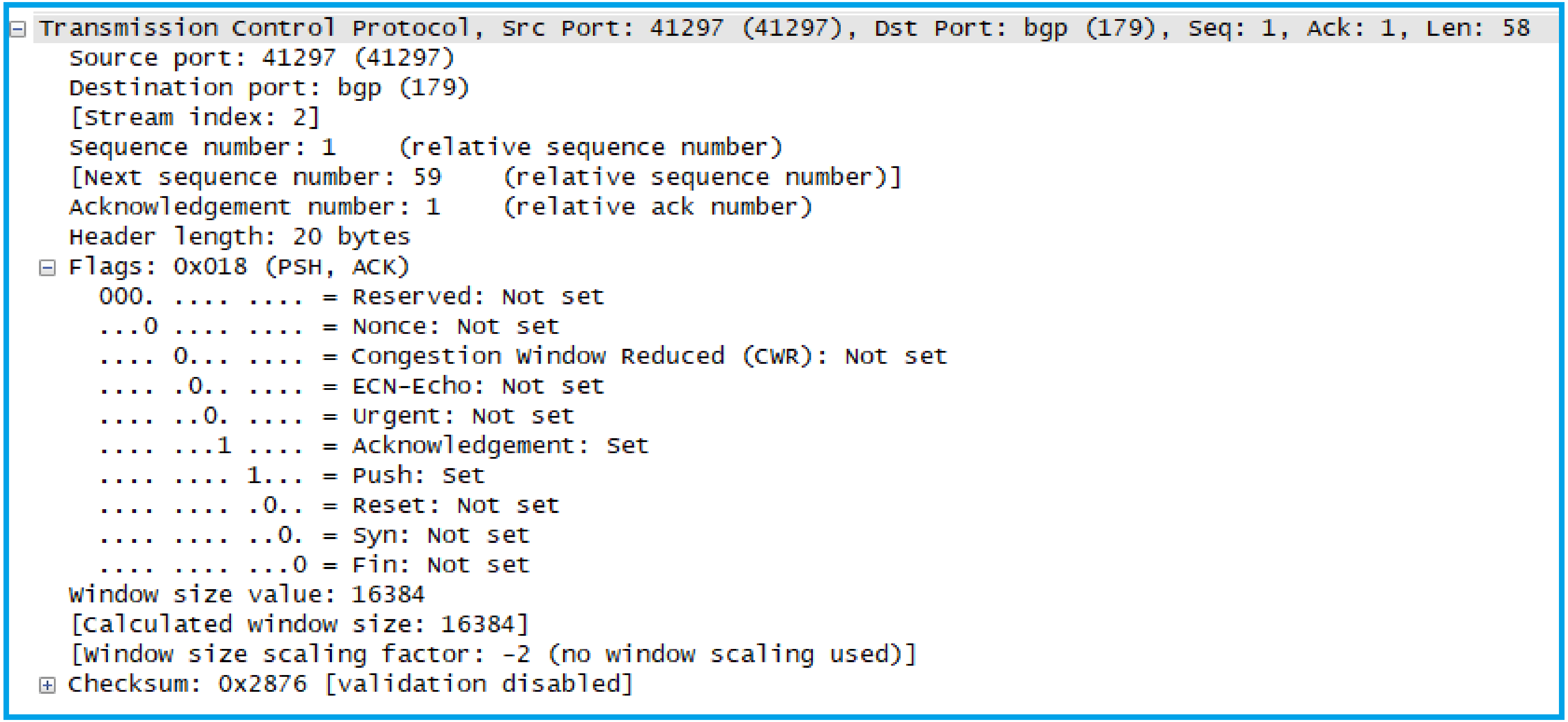
SYN bit is used to establish connections. The connection request has SYN = 1 and ACK = 0 to indicate that the built-in acknowledgment field is not in use. The connection response does carry an acknowledgment, so it has SYN = 1 and ACK = 1. Essentially, the SYN bit is used to denote CONNECTION REQUEST and CONNECTION ACCEPTED, and the ACK bit serves to distinguish between the two possibilities.

FIN bit is used to release a connection; specifies that the sender has no more data to transmit. However, after closing a connection, a process can continue to receive data indefinitely. Both SYN and FIN segments have sequence numbers and are therefore guaranteed to be processed in the correct order.

- **Flow control** in TCP is handled using a variable-sized sliding window. The Window Size field specifies the number of octets, starting with the octet indicated by the acknowledgment number, that the sender of the segment will accept from its peer at the other end of the connection before the peer must stop transmitting and wait for a Acknowledgment of receipt. A Window Size field of 0 is valid, and indicates that bytes up to and including Receipt Acknowledgment Number − 1 have been received, but that the receiver currently needs a break and would like to not receive any more data at this time, thank you.

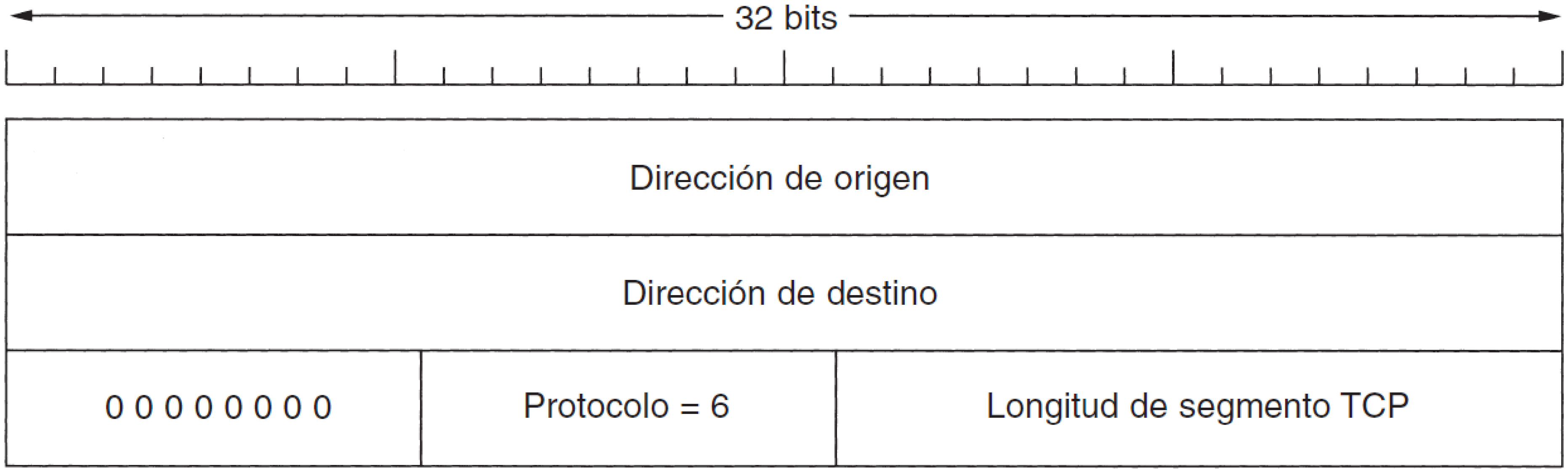
- **Permission to send** can be granted later by sending a segment with the same Receipt Confirmation Number and a non-zero Window Size field. In link layer protocols, acknowledgments of received frames and permissions to send new frames were linked. This was a consequence of a fixed window size for each protocol. In TCP, receipt confirmations and permissions to send additional data are completely independent. In effect, a receiver can say: “I have received bytes up to k, but for now I do not want more.” This independence (in effect, a resizable window) gives additional flexibility.

- A **Checksum** is also provided to add reliability. It is a checksum of the header, data, and conceptual pseudo-header shown in the figure below, which allows for error detection.



```
⊟ Transmission Control Protocol, Src Port: 41297 (41297), Dst Port: bgp (179), Seq: 1, Ack: 1, Len: 58
    Source port: 41297 (41297)
    Destination port: bgp (179)
    [Stream index: 2]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 59    (relative sequence number)]
    Acknowledgement number: 1    (relative ack number)
    Header length: 20 bytes
  ⊟ Flags: 0x018 (PSH, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgement: Set
      .... .... 1... = Push: Set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
    Window size value: 16384
    [Calculated window size: 16384]
    [Window size scaling factor: -2 (no window scaling used)]
  ⊞ Checksum: 0x2876 [validation disabled]
```

Performing this calculation sets the TCP Checksum field to zero, and pads the data field with an additional zero byte if the length is an odd number. The checksum algorithm simply adds all 16-bit words in 1's complement and then obtains the 1's complement of the sum.As a result, when the receiver performs the calculation with the entire segment, including the Checksum field, the result must be 0.The pseudo-header contains the 32-bit IP addresses of the source and destination machines, the TCP protocol number (6), and the byte count of the TCP segment (including the header). Including the pseudo-header in the TCP checksum calculation helps detect misdelivered packets, but doing so violates the protocol hierarchy since the IP addresses it contains belong to the IP layer, not the TCP layer. UDP uses the same pseudo header for its checksum.

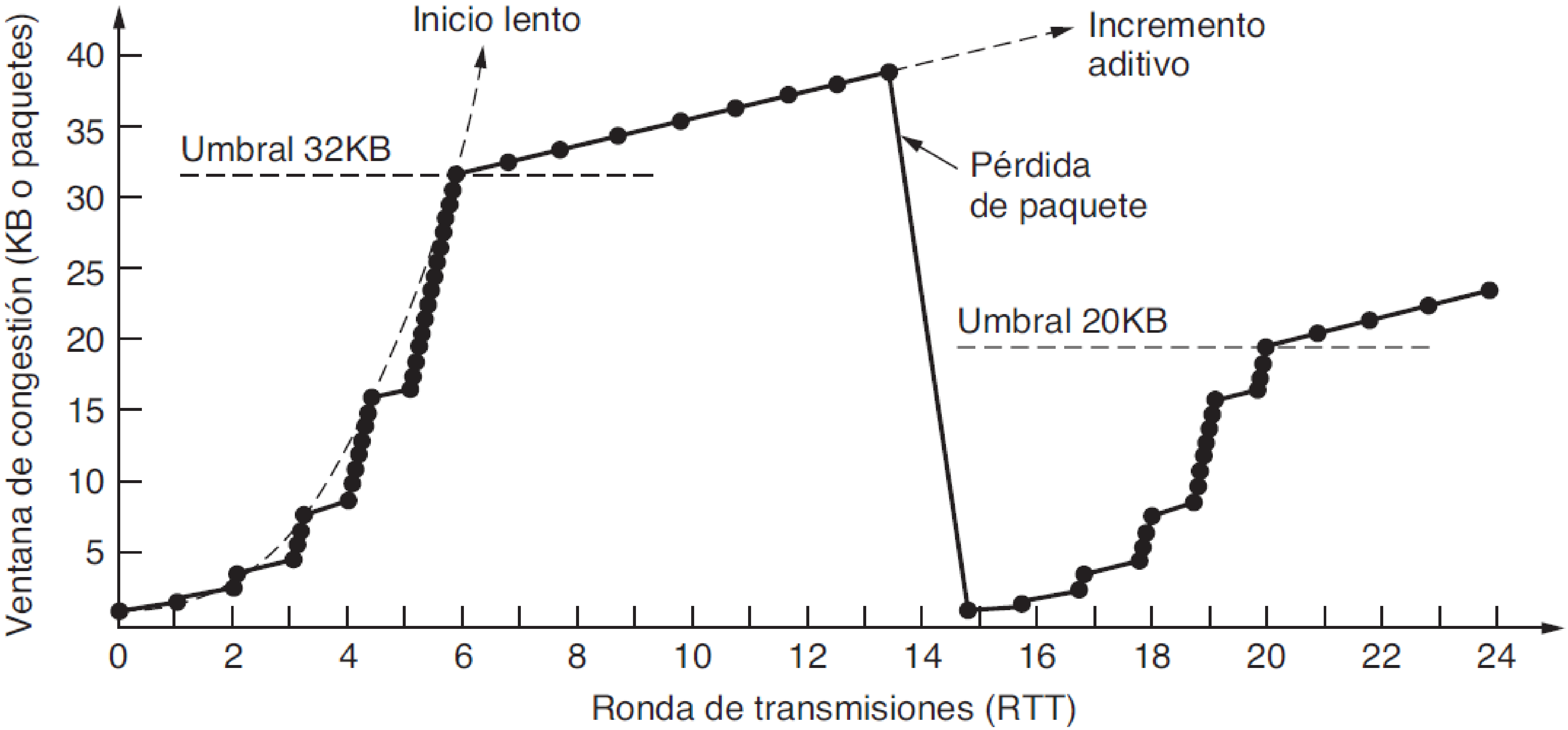


The TCP checksum is mandatory. The TCP checksum is applied over the entire segment. The TCP size is calculated by adding the TCP header size plus the data size.On an incoming segment the checksum is calculated and compared to the checksum field of the TCP header; if the values do not match, the segment is discarded.

- The Options field offers a way to add extra features not covered by the headernormal. The most important option is the one that informs the receiver of the largest segment that the transmitter is willing to accept (Maximum Segment Size). Using large segments is more efficient than using small segments, since the 20-byte header can then be amortized over more data, but small hosts may not be able to handle very large segments.During connection establishment, each side can announce its maximum and see its peer's. If a host does not use this option, it has a default payload of 536 bytes. All hosts on the Internet are required to accept TCP segments of 536 + 20 = 556 bytes. The maximum segment size in both directions does not need to be the same.

- On lines with high bandwidth, high delay, or both, the 64 KB window is often a problem. On a T3 line (44.736 Mbps) only 12 msec is required to send a complete 64 KB window. If the round-trip propagation delay is 50 msec (typical of a transcontinental fiber), the sender will be idle 3/4 of the time waiting for acknowledgments. In a satellite connection the situation is even worse. A larger window size will allow the sender to continue sending data, but since the Window Size field is 16 bits, it is impossible to express such a size. A window scaling option was proposed in RFC 1323 that allows the sender and receiver to negotiate a window scaling factor. This number gives the possibility for both sides to shift the Window Size field up to 14 bits to the left, therefore allowing windows of up to 230 bytes. Most current TCP implementations handle this option.

- Another option proposed in RFC 1106 and now in widespread use is the use of selective replay instead of the rollback protocol n. If the receiver receives a bad segment and then a large number of good segments, the normal TCP protocol timer will expire at some point and all unacknowledged segments will be retransmitted, including those that were received correctly. RFC 1106 introduced NAKs, to allow the receiver to request a specific segment (or segments). After receiving it, it can send an acknowledgment of receipt of all the data it has buffered, thus reducing the amount of data retransmitted.

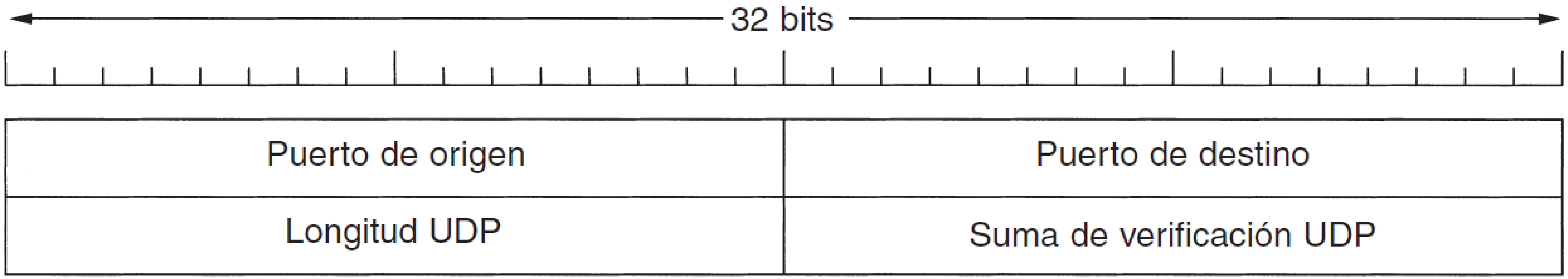

Inicio lento
Incremento aditivo
Umbral 32KB
Pérdida de paquete
Umbral 20KB
Ventana de congestión (KB o paquetes)
40
35
30
25
20
15
10
5
0
2
4
6
8
10
12
14
16
18
20
22
24
Ronda de transmisiones (RTT)

Introduction to UDP

The Internet protocol suite supports a connectionless transport protocol,UDP (User Datagram Protocol). This protocol provides a way for applications to send encapsulated IP datagrams without having to establish a connection. UDP is described in RFC 768.

UDP transmits segments consisting of an 8-byte header followed by the payload.The following figure shows such a header. The two ports serve to identify the endpoints within the source and destination machines. When a UDP packet arrives, its payload isdelivered to the process that is bound to the destination port. This binding occurs when the BIND primitive or something similar is used, as we saw for TCP (the binding process is the same for UDP).

In fact, the main value of having UDP instead of just using pure IP is the addition of source and destination ports. Without the port fields, the transport layer would not know what to do with the packet. With them, deliver the segments correctly.



Puerto de origen	Puerto de destino
Longitud UDP	Suma de verificación UDP

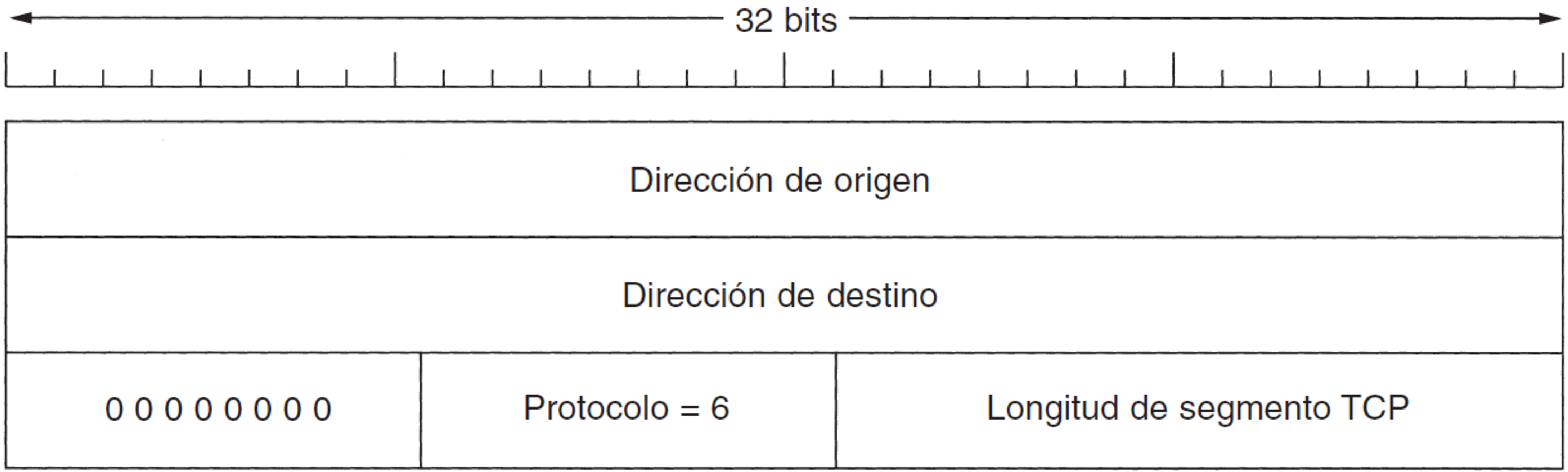
Header Description

The source port is mainly needed when a response must be sent to the source. By copying the Source Port field of the arriving segment to the Destination Port field of the outgoing segment, the process sending the response can specify which process on the sending machine is to obtain it.
The UDP Length field includes the 8-byte header and the data.
The UDP Checksum field is optional and is calculated using a pseudo header. Disabling it does not make sense unless the quality of data service does not matter (for example, in digitized voice). It's probably worth explicitly mentioning some of the things that UDP doesn't do. Does not perform flow control, error control, or retransmission when a segment is received wrong. All of the above corresponds to user processes. What it does do is provide an interface to the IP protocol with the added feature of demultiplexing multiple processes using ports.
This is all it does. For applications that need to have precise control over packet flow, error control, or timing, UDP is ideal. One area where UDP is especially useful is in client-server situations. Often, the client sends a short request to the server and waits for a short response. If the request or response is lost, the client can simply terminate and try again.

Not only is the code simple, but very few messages are needed (one in each direction) compared to a protocol that requires initial configuration. One application that uses UDP in this way is DNS (the Domain Name System). In short, a program that needs to look up the IP address of some host, for example, www.cs.berkeley.edu, can send a UDP packet containing the name of that host to the DNS server. The server responds with a UDP packet containing the host's IP address. No upfront configuration or post-release is required. Only two messages travel through the network.

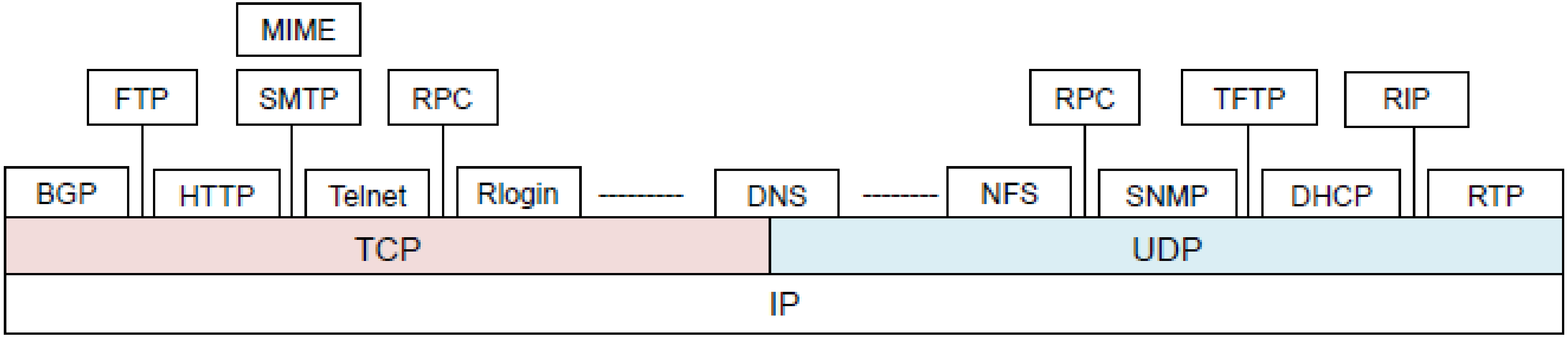
Checksum calculation

The purpose of the UDP checksum is to validate the content of a UDP message. The UDP checksum is calculated on the combination of a specially constructed pseudo-header with some IP information, the UDP header, and the message data. The pseudo header format that adds the checksum is shown below



The source address, destination address and protocol fields are taken from the IP header. The use of UDP checksum in a particular communication is optional. If not used, the field is 0. If a checksum has been calculated and its value becomes 0, it is represented with a field of ones

Some applications supported in the DARPA model

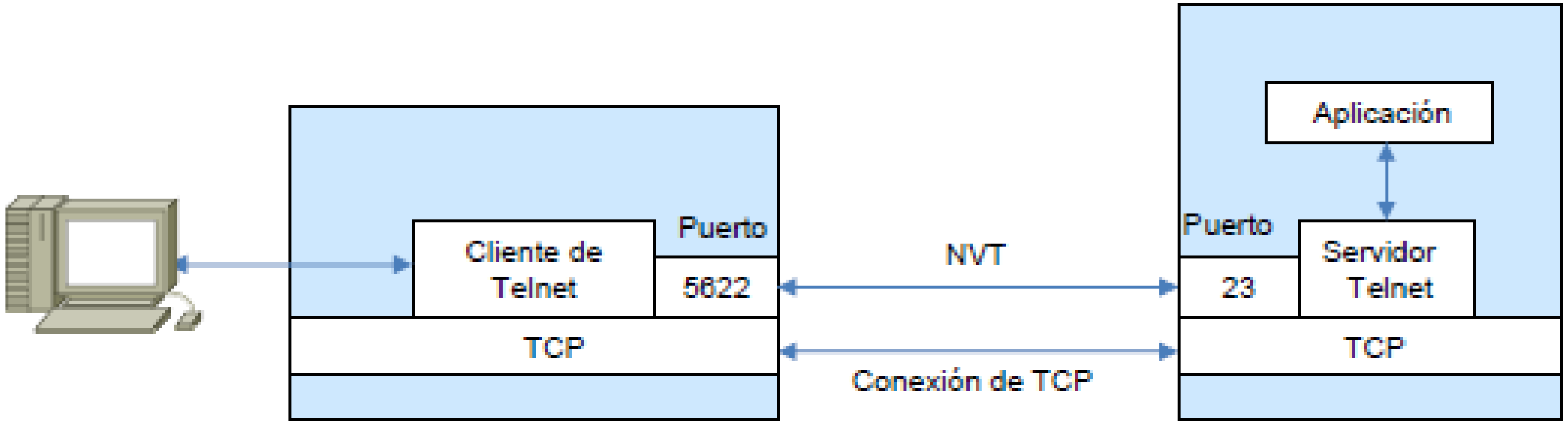


BGP: Border Gate Protocol
NFS: Network File System
FTP: File Transfer Protocol
SNMP: Simple Network Management Protocol
HTTP: Hypertext Transfer Protocol
TFTP: Trivial File Transfer Protocol
SMTP: Simple Mail Transfer Protocol
DHCP: Dynamic Host Configuration Protocol

MIME: Multipurpose Internet Email Extension
RIP: Routing Information Protocol
Telnet: Telecommunication Network
RTP: Real Time Protocol
RPC: Remote Procedure Call
Rlogin: Remote Access
DNS: Domain Name System

Telnet

The TCP/IP protocol suite includes a simple remote terminal protocol called TELNET that allows a user to log into a computer across an internet.



As the figure shows, when a user invokes TELNET, an application program on the user's machine becomes the client. The client establishes a TCP connection to the server over which they will communicate. Once the connection has been established, the client accepts keystrokes from the user's keyboard and sends them to the server, while it concurrently accepts characters that the server sends back and displays them on the user's screen. The server must accept a TCP connection from the client, and then relay data between the TCP connection and the local operating system.

Telnet

TELNET offers three basic services.

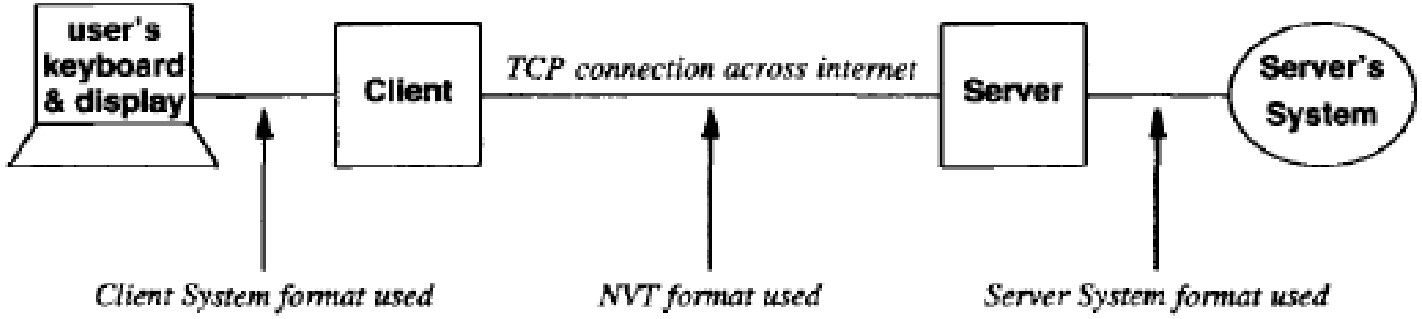
- First, it defines a network virtual terminal that provides a standard interface to remote systems.

Client programs do not have to understand the details of all possible remote systems; they are built to use the standard interface.

- Second, TELNET includes a mechanism that allows the client and server to negotiate options, and it

provides a set of standard options (e.g., one of the options controls whether data passed across the connection uses the standard 7-bit ASCII character set or an 8-bit character set).

- Finally, TELNET treats both ends of the connection symmetrically.

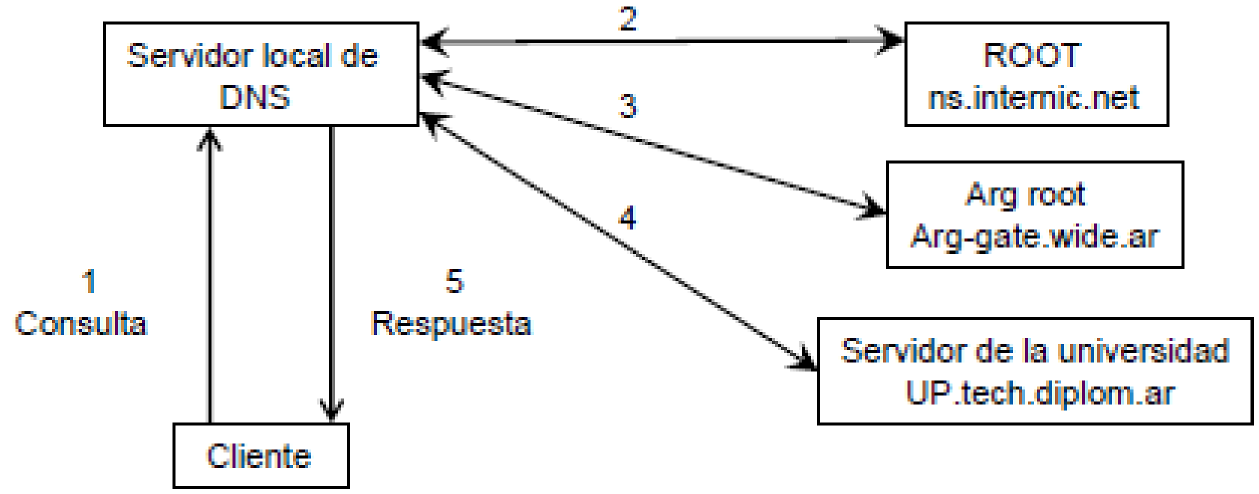

user's keyboard & display
Client
TCP connection across internet
Server
Server's System
Client System format used
NVT format used
Server System format used

To accommodate heterogeneity, TELNET defines how data and command sequences are sent across the Internet. The definition is known as the ***network virtual terminal*** *(NVT).* As figure illustrates, the client software translates keystrokes and command sequences from the user's terminal into NVT format and sends them to the server. Server software translates incoming data and commands from NVT format into the format the remote system requires. For data returning, the remote server translates from the remote machine's format to NVT, and the local client translates from NVT to the local machine's format.

DNS (DOMAIN NAME SERVER))

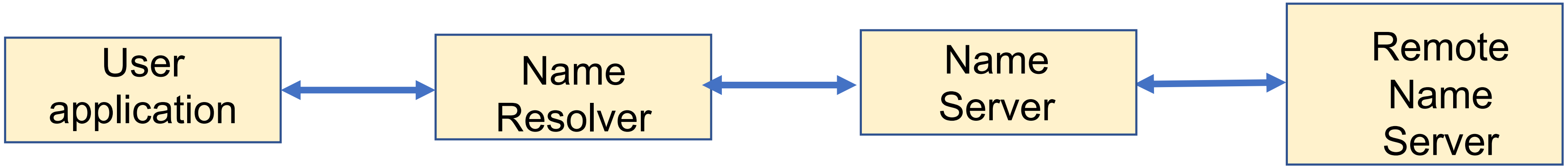
Typically, an end user knows the name of a host, but not its address. However, IP needs to know the address of a host in order to communicate with it. Also the end user, or the application that the end user has invoked, needs a mechanism to obtain the address from the name of a given host.The Domain Name System (DNS) was created to provide a better method for keeping track of names and addresses on the Internet. DNS is a distributed database. Internet names and addresses are stored on servers around the world. DNS databases provide automatic name-to-address translation services.

Traducción de nombre a dirección



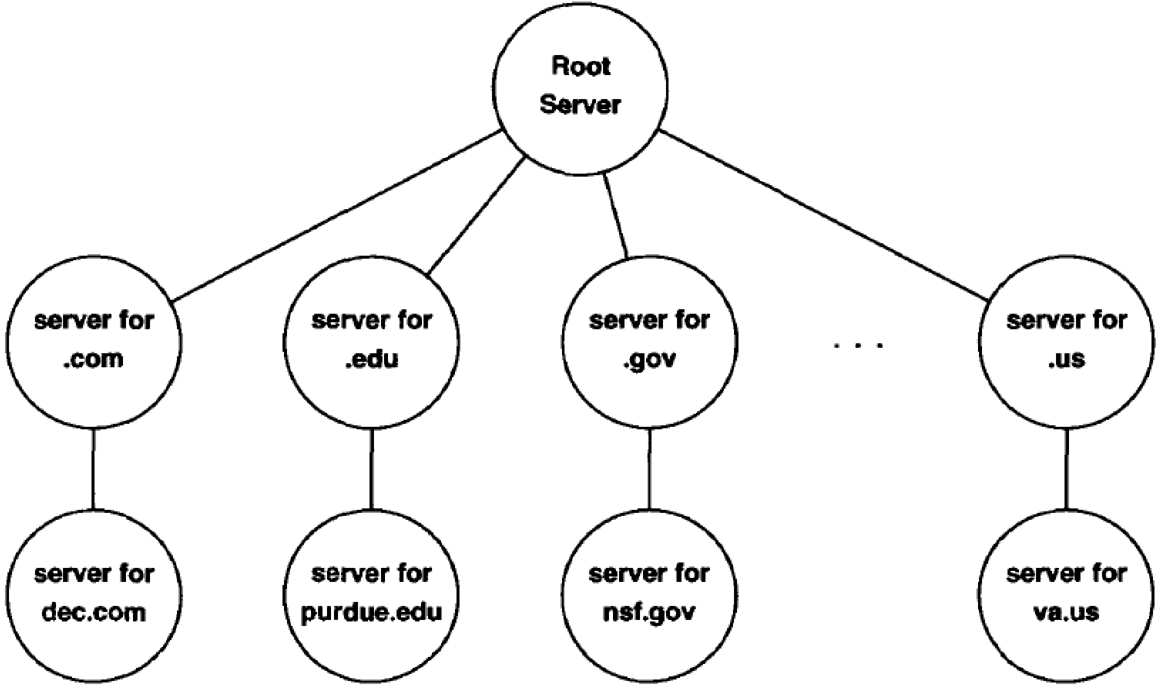
DNS, Mapping Domain Names To Addresses

In addition to the rules for name syntax and delegation of authority, the domain name scheme includes an efficient, reliable, general purpose, distributed system for mapping names to addresses. The system is distributed in the technical sense, meaning that a set of servers operating at multiple sites cooperatively solve the mapping problem.
It is efficient in the sense that most names can be mapped locally; only a few require internet traffic.
It is general purpose because it is not restricted to machine names.
Finally, it is reliable in that no single machine failure will prevent the system from operating correctly.
The domain mechanism for mapping names to addresses consists of independent, cooperative systems called ***name servers.*** A name server is a server that supplies name-to-address translation, mapping from domain names to IP addresses. Often, server software executes on a dedicated processor, and the machine itself is called the name server. The client software, called a ***name resolver,*** uses one or more name servers when translating a name.



DNS, Mapping Domain Names To Addresses

The easiest way to understand how domain servers work is to imagine them arranged in a tree structure that corresponds to the naming hierarchy, as figure illustrates.
The root of the tree is a server that recognizes the top-level domains and knows which server resolves each domain. Given a name to resolve, the root can choose the correct server for that name. At the next level, a set of name servers each provide answers for one top-level domain (e.g., *edu).* A server at this level knows which servers can resolve each of the subdomains under its domain. At the third level of the tree, name servers provide answers for subdomains (e.g., *purdue under edu*). The conceptual tree continues with one server at each level for which a subdomain has been defined



The tree of servers has few levels because a single physical server can contain all of the information for large parts of the naming hierarchy.

A root server contains information about the root and top-level domains, and each organization uses a single server for its names.

DNS, Domain Name Resolution

When a domain name server receives a query, it checks to see if the name lies in the subdomain for which it is an authority. If so, it translates the name to an address according to its database, and appends an answer to the query before sending it back to the client. If the name server cannot resolve the name completely, it checks to see what type of interaction the client specified.

If the client requested complete translation ***(recursive resolution, in domain name terminology),*** the server contacts a domain name server that can resolve the name and returns the answer to the client.

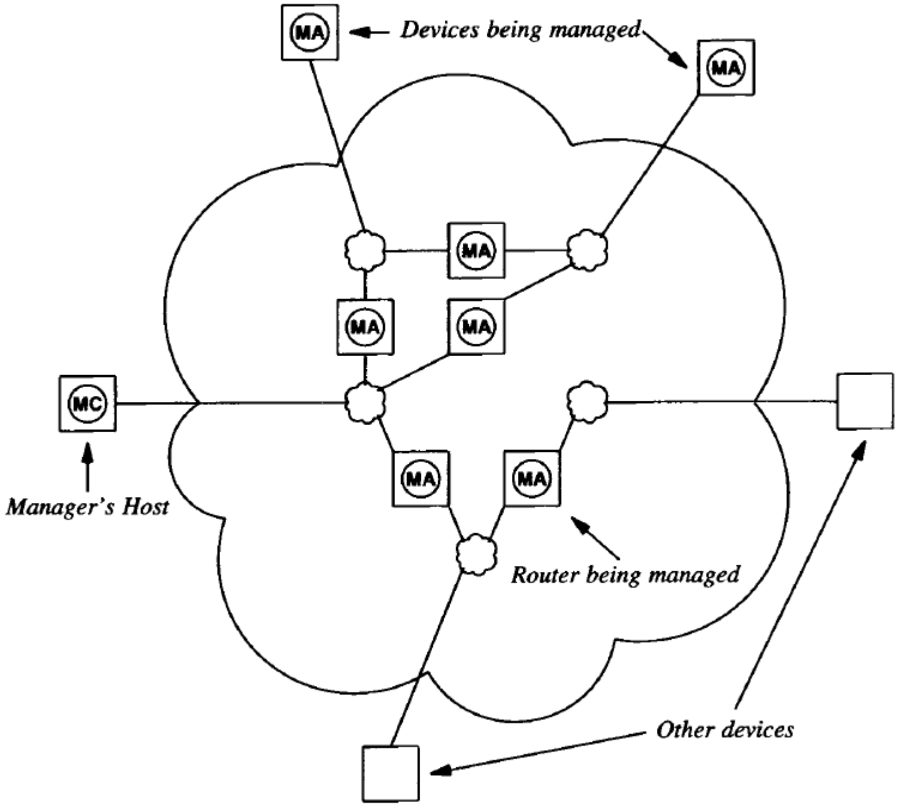
If the client requested non-recursive resolution ***(iterative resolution),*** the name server cannot supply an answer. It generates a reply that specifies the name server the client should contact next to resolve the name.

Domain name servers use a well-known protocol port for all communication, so clients know how to communicate with a server once they know the IP address of the machine in which the server executes.

SNMP

The Simple Network Management Protocol (SNMP) was originally developed as a mechanism for managing TCP/IP and Ethernet networks.

In addition to protocols that provide network level services and application programs that use those services, an internet needs software that allows managers to debug problems, control routing, and find computers that violate protocol standards.



In a TCP/IP internet, a manager needs to examine and control routers and other network devices.

Because such devices attach to arbitrary networks, protocols for internet management operate at the Application level and communicate using TCP/IP transport-level protocols.

As the figure shows, client software usually runs on the manager's workstation.

Each participating router or host runs a server program. Technically, the server software is called a management agent or merely an agent. A manager invokes client software on the local host computer and specifies an agent with which it communicates.

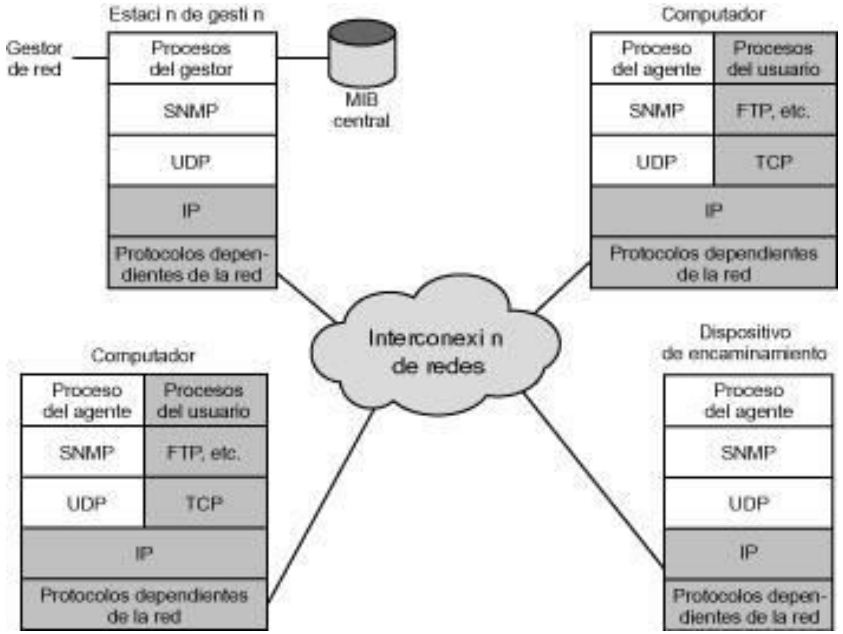
After the client contacts the agent, it sends queries to obtain information or it sends commands to change conditions in the router.

A manager invokes management client (MC) software that can contact management agent (MA) software that runs on devices throughout the internet.

SNMP, Architecture

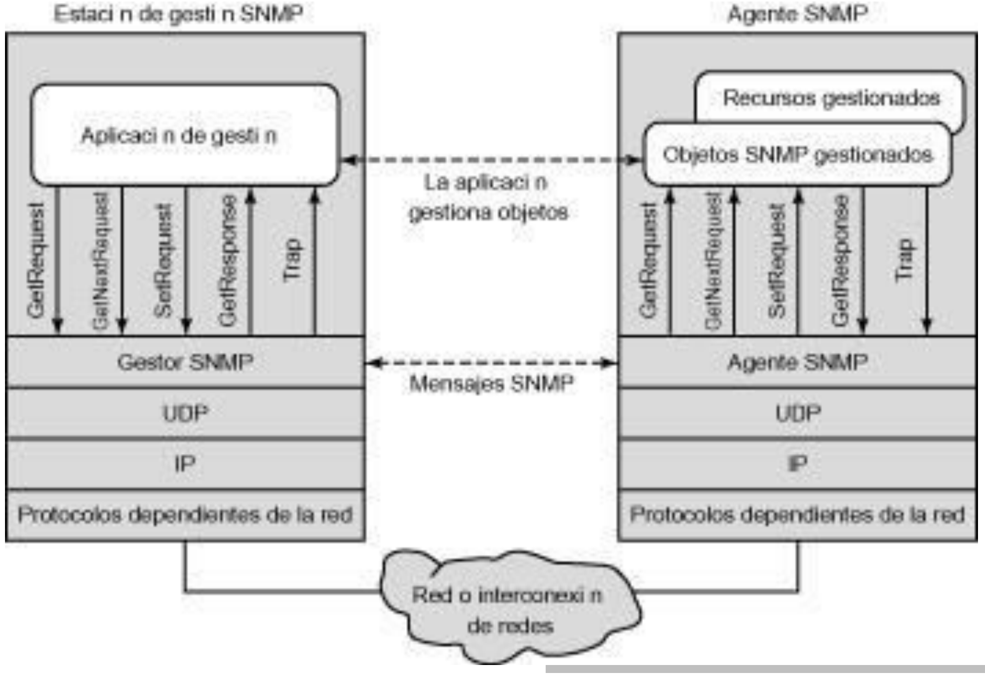
SNMP is based upon three components: management software, agent software, and management information bases (MIB), the latter representing databases for managed devices.

- Management software operates on a network management station (NMS) and is responsible for querying agents using SNMP commands.
- Agent software represents one or more program modules that operate within a managed device, such as a workstation, bridge, router, or gateway.
Each managed agent stores data and provides stored information to the manager upon the latter’s request.
-The MIB represents a database that provides a standard representation of collected data.
This database is structured as a tree and includes groups of objects that can be managed.



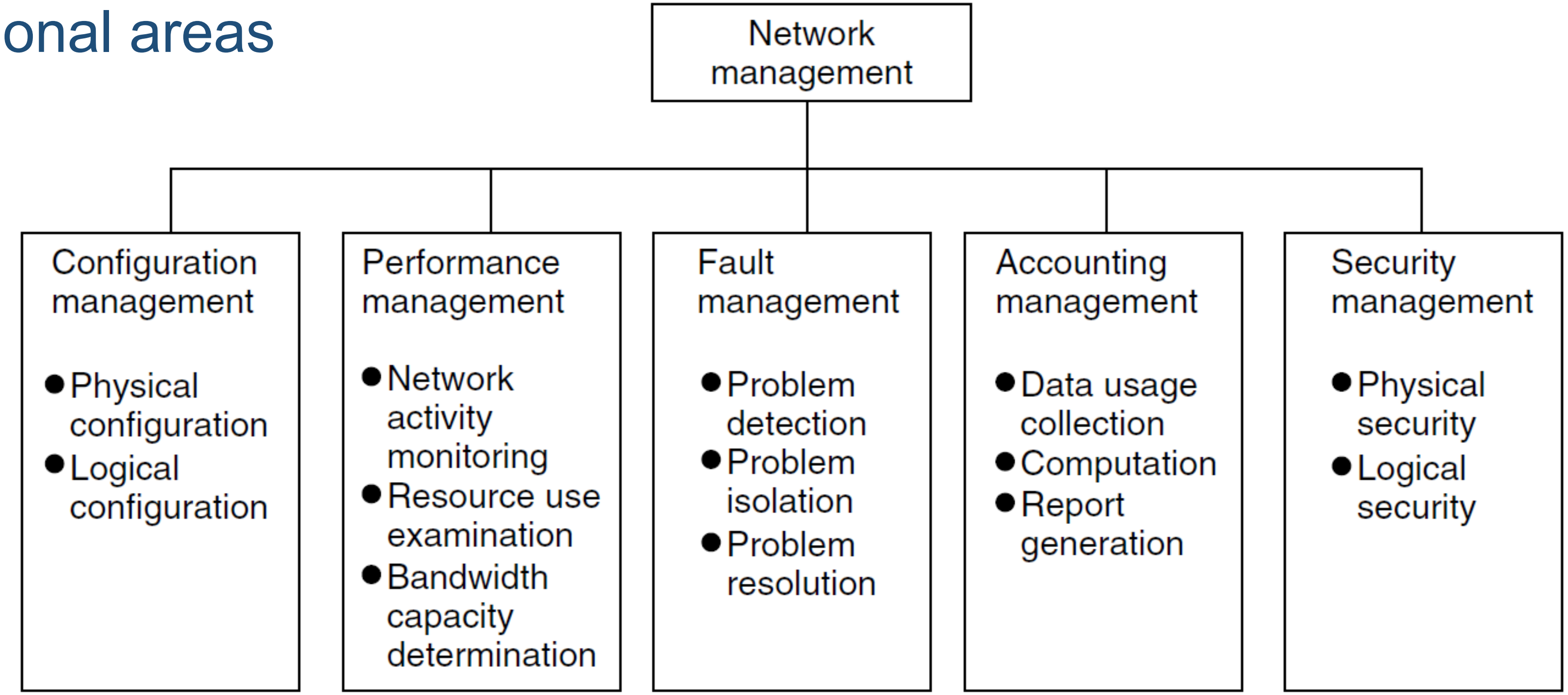
SNMP, versions

SNMP has a core set of five commands referred to as protocol data units (PDUs) – SNMP Version 1.
Those PDUs include GetRequest, GetNextRequest, SetRequest, GetResponse, and Trap.
The Network Management Station (NMS) issues a GetRequest to retrieve a single value from an agent's MIB, while a GetNextRequest is used to *walk* through the agent's MIB table.
When an agent responds to either request, it does so with a GetResponse.
The SetRequest provides a manager with the ability to alter an agent's MIB.
Since SNMP is a polling protocol, a mechanism was required to alert managers to a situation that requires their attention.
Otherwise, a long polling interval could result in the occurrence of a serious problem that might go undetected for a relatively long period of time on a large network.
The mechanism used to alert a manager is a Trap command, issued by an agent to a manager



Under SNMPVersion 2, two additional PDUs were added: GetBulkRequest and InformRequest.
The GetBulkRequest command supports the retrieval of multiple rows of data from an agent's MIB with one request. The InformRequest PDU enables one manager to transmit unsolicited information to another manager, permitting the support of distributed network management.
The introduction of SNMP Version 3 added authentication as well as encryption, resulting in a network management message received by an agent to be recognized if it was altered, as well as to be verified that it was issued by the appropriate manager.

SNMP, Functional areas



Network management

Configuration management
- Physical configuration
- Logical configuration

Performance management
- Network activity monitoring
- Resource use examination
- Bandwidth capacity determination

Fault management
- Problem detection
- Problem isolation
- Problem resolution

Accounting management
- Data usage collection
- Computation
- Report generation

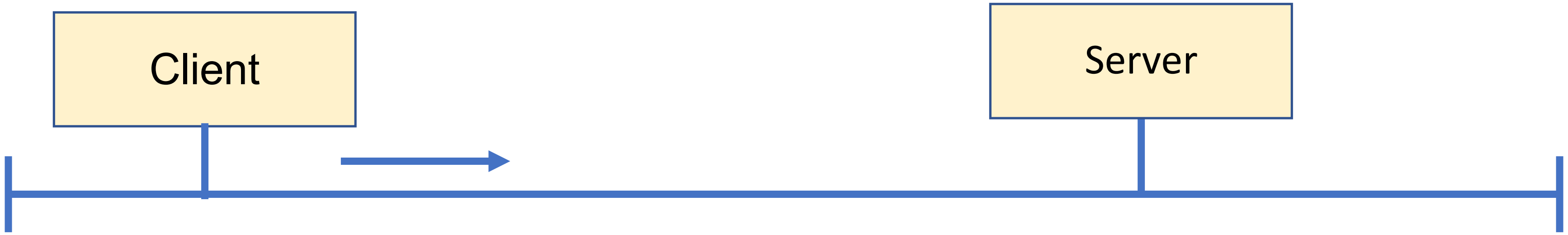
Security management
- Physical security
- Logical security

- The process of configuration management covers both the hardware and software settings required to provide an efficient and effective data transportation highway.
- Performance management involves those activities required to ensure that the network operates in an orderly manner without unreasonable service delays.
- Networks have their less desirable moments in which components fail, software is configured incorrectly, and other problems occur. *Fault management* is the set of functions required to detect, isolate, and correct network problems.
- Accounting management is a set of activities that enables you to determine network usage, generate usage reports, and assign costs to individuals or groups of users by organization or by department.
- As discussed in our overview of configuration management, security management involves primarily the assignment of network access passwords and access permissions to applications and file storage areas on the network.

DHCP (Dynamic Host Configuration Protocol) - BOOTP (Protocolo de arranque)

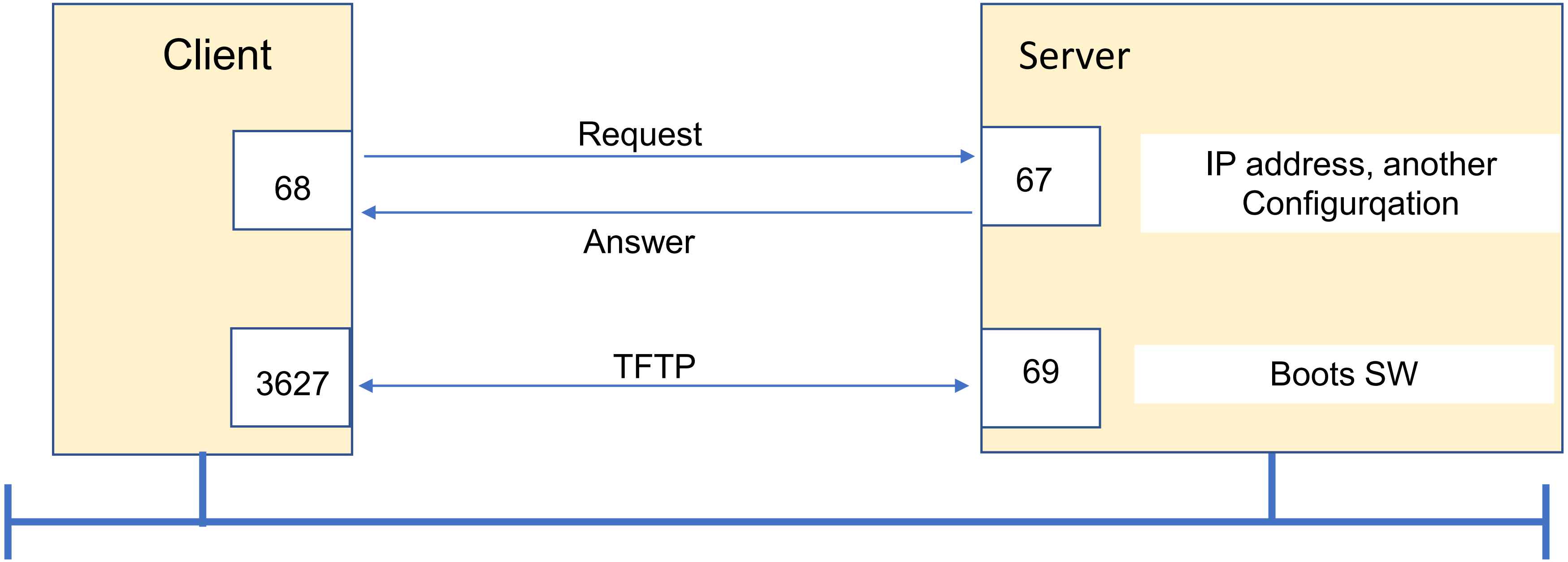
To overcome some of the drawbacks of RARP, researchers developed the BOOTstrap Protocol (BOOTP).
Later, the Dynamic Host Configuration Protocol (DHCP) was developed as a successor to BOOTP.
Because the two protocols are closely related, most of the description applies to both.
We will describe BOOTP first, and then see how DHCP extends the functionality to provide dynamic address assignment.
Because it uses UDP and IP, BOOTP can be implemented with an application program.
Like RARP, BOOTP operates in the client-server paradigm and requires only a single packet exchange.

To understand how a computer can send BOOTP in an IP datagram before the computer learns its IP address, recall that there are several special-case IP addresses.



Client broadcasts BOOTP, using 0.0.0.0 as its source address

BOOTP, Boots protocol



DHCP (Dynamic Host Configuration Protocol)

To handle automated address assignment, the IETF has designed a new protocol.

Known as the ***Dynamic Host Configuration Protocol (DHCP),*** the new protocol extends BOOTP in two ways.

First, DHCP allows a computer to acquire all the configuration information it needs in a single message. For example, in addition to an IP address, a DHCP message can contain a subnet mask.

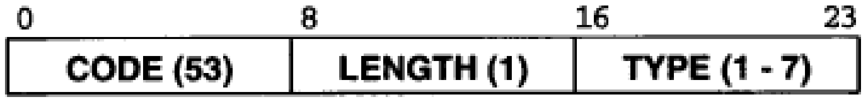
Second, DHCP allows a computer to obtain an IP address quickly and dynamically.

To use DHCP's dynamic address allocation mechanism, a manager must configure a DHCP server by supplying a set of IP addresses

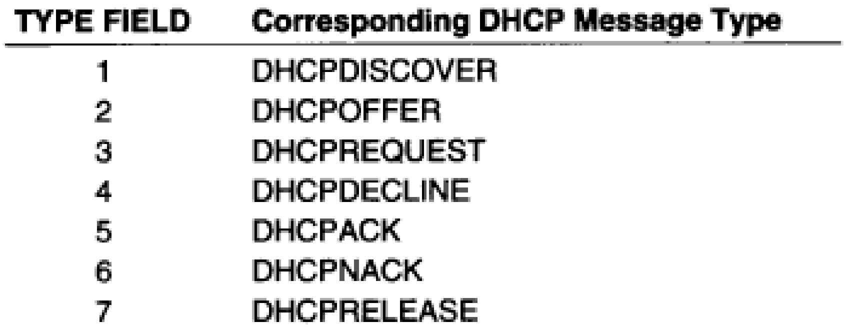
DHCP (Dynamic Host Configuration Protocol)

DHCP allows three types of address assignment; a manager chooses how DHCP will respond for each network or for each host.
Like BOOTP, DHCP allows **manual configuration** in which a manager can configure a specific address for a specific computer.
DHCP also permits **automatic configuration** in which a manager allows a DHCP server to assign a permanent address when a computer first attaches to the network.
Finally, DHCP permits completely **dynamic configuration** in which a server "loans" an address to a computer for a limited time.
To use DHCP, a host becomes a client by broadcasting a message to all servers on the local network.
The host then collects offers from servers, selects one of the offers, and verifies acceptance with the server.

Surprisingly, DHCP does not add new fixed fields to the BOOTP message format, nor does it change the meaning of most fields. To encode information such as the lease duration, DHCP uses options.
In particular, figure illustrates the DHCP message type option used to specify which DHCP message is being sent



0	8	16	23
CODE (53)	LENGTH (1)	TYPE (1 - 7)	



TYPE FIELD	Corresponding DHCP Message Type
1	DHCPDISCOVER
2	DHCPOFFER
3	DHCPREQUEST
4	DHCPDECLINE
5	DHCPACK
6	DHCPNACK
7	DHCPRELEASE

IPSec

IETF knew for years that there was a lack of security on the Internet. The most serious types of attacks included IP spoofing, in which an intruder creates packets with fake IP addresses and exploits applications that use IP address-based authentication. Also included were various forms of eavesdropping and packet capture (packet sniffing), in which attackers read transmitted information, including system login information and database content.Adding it was not easy as a controversy arose over where to place it.Most security experts believed that to be truly secure, encryption and integrity checks had to be carried out end-to-end (i.e., at the application layer). In this way, the source process encrypts and/or protects the integrity of the data and sends it to the destination process where it is decrypted and/or verified. Therefore, any alteration made in between these two processes, or in any operating system, can be detected.

The problem with this approach was that it required changing all applications to be security aware. From this perspective, the next best approach was to place encryption at the transport layer or a new layer between the application and transport layers, thereby preserving the end-to-end approach but not requiring changing the Applications.The opposite perspective was that users do not understand security and are not able to use it correctly, and that no one wants to modify existing programs in any way, so the network layer must authenticate and/or encrypt packets without users are involved. After years of fierce discussion, this view gained enough support for a network layer security standard to be defined. The argument was partly that having network layer encryption did not prevent security-aware users from applying it correctly and that it helped non-security-aware users to some extent. The result of this discussion was a design called IPsec (IP Security), which is described in RFCs 2401, 2402 and 2406, among others .Not all users want encryption (as this is computationally expensive).

IPSec

Instead of making it optional, it was decided to require encryption all the time but allow the use of a null algorithm. This is described and valued for its simplicity, ease of implementation and great speed in RFC 2410.

The complete IPsec design is a framework for multiple services, algorithms, and granularities.
- The reason for multiple services is that not all people want to pay the price to have all the services all the time, so the services are available a la carte. The main services are confidentiality, data integrity and protection against replay attacks (an intruder repeats a conversation). All of these are based on symmetric cryptography because high performance is crucial.
- The reason for having multiple algorithms is that an algorithm that is thought to be secure now may be violated in the future. By making the IPsec algorithm independent, the structure can survive even ifsome particular algorithm is subsequently violated.
- The reason for having multiple granularities is to make it possible to protect a single TCP connection, all traffic between a pair of hosts, or all traffic between a pair of secure routers, among other possibilities.

A slightly surprising aspect of IPsec is that although it is at the IP layer, it is connection-oriented. Nowadays, that is not so surprising because to be safe, it is necessary to establish and use a key for some period - in essence, a type of connection. A “connection” in the context of IPsec is known as a SA (security association).

IPSec applications
IPSec provides the ability to secure communications over a LAN, over a private or public WAN, and over the Internet. Some examples of its use include the following:- Secure connectivity between branches over the Internet: a company can build a virtual private network over the Internet or through a public WAN. This allows a business to support firmly on the Internet and reduce your need for a private network, saving costs and additional network management.

IPSec

- Secure remote access over the Internet: An end user whose system is equipped with IP security protocols can make a local call to an Internet Service Provider (ISP) and gain secure access to the network. a company. This reduces the cost of expenses for mobile employees and remote workers.

- Establishing intranet and extranet connectivity with partners: IPSec can be used to secure communication with other organizations, ensuring authentication and privacy and providing a key exchange mechanism.

- Improved security in e-commerce: although some web and e-commerce applications have built-in security protocols, the use of IPSec improves that security.

The main feature of IPSec that allows it to support these various applications is in that it can encrypt and/or authenticate all traffic at the IP level. Thus, all distributed applications, including remote connection, client/server applications, email, file transfer, web access, etc., can be made secure.

IPSec: Core Services
IPSec provides three main services: an authentication-only function known as Authentication Header (AH), a combined authentication/encryption function called Encapsulating Security Payload (ESP), and a key exchange.
For virtual private networks, authentication and encryption are generally desired, since it is important to both (1) ensure that unauthorized users do not enter the virtual private network and (2) ensure that observers on the Internet cannot read the messages sent by the virtual private network. Since both features are desirable, most implementations use ESP instead of AH. The key exchange feature allows for manual key exchange as well as an automatic scheme. The IPSec specification is quite complex and comprises numerous documents.

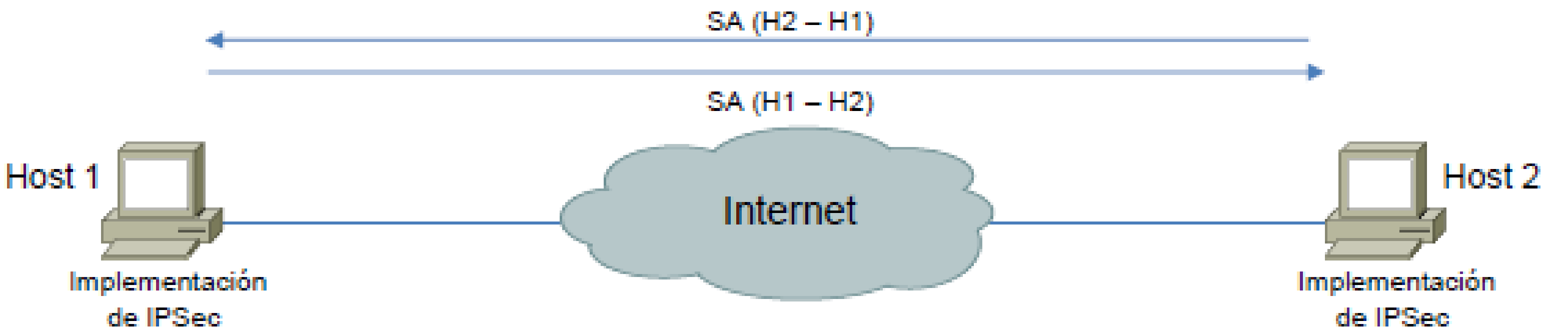
IPSec

The most important, published in November 1998, are RFCs 2401, 2402, 2406, and 2408. This section provides an overview of some of the most important elements of IPSec.

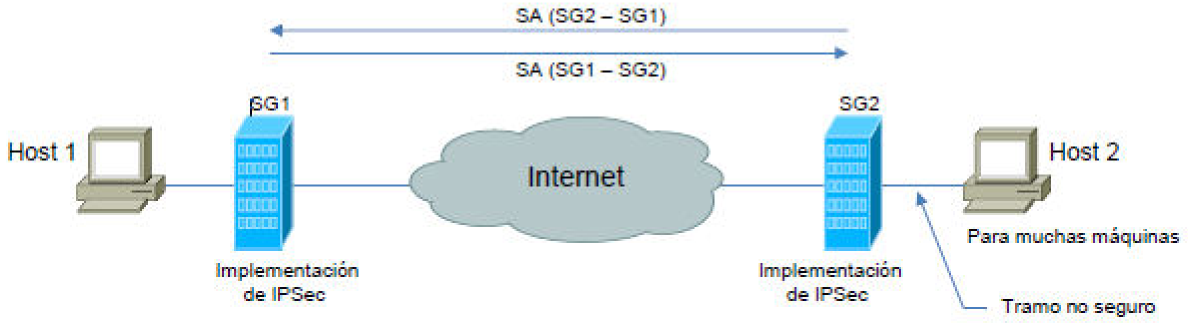
Security associations:
A key concept that appears in both authentication and privacy mechanisms in IP, is the security association (SA). An SA is a simplex connection between two endpoints and has a security identifier associated with it. If secure traffic is needed in both directions, two security associations are required. Security identifiers are carried in packets that travel over these secure connections and are used to look up keys and other relevant information when a secure packet arrives.
Security services are provided to an SA to use either AH or ESP, but not both.

A security association is uniquely identified by three parameters:

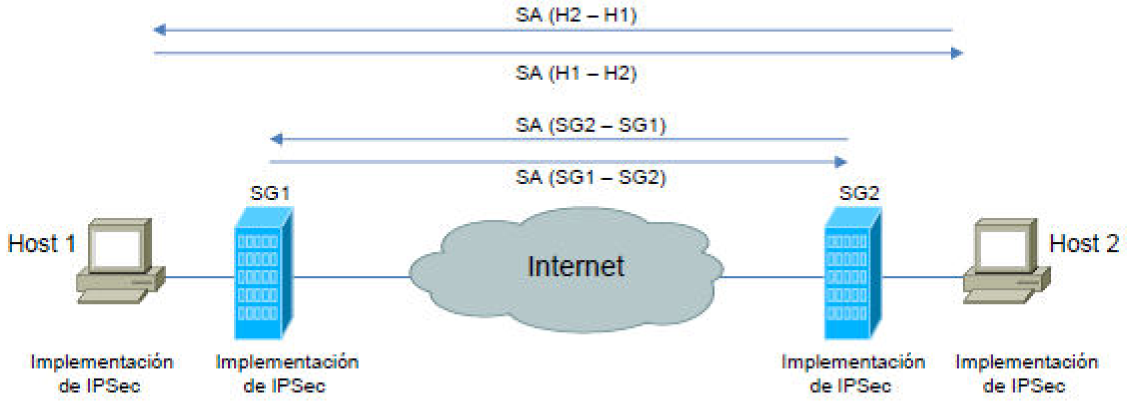
- Security Parameters Index (SPI): A string of bits assigned to this SA and with local meaning only. The SPI is carried in the AH and ESP headers to allow the receiving system to select the SA under which a received packet will be processed.
- Destination IP address: currently only unicast addresses are allowed. This is the address of the destination endpoint of the SA, which can be an end user or a network system, such as a firewall or a routing device.
- Security protocol identifier: distinguishes between an AH or ESP security association. Below are some examples of SA



IPSec



SG (Security Gateway): Gateway de seguridad



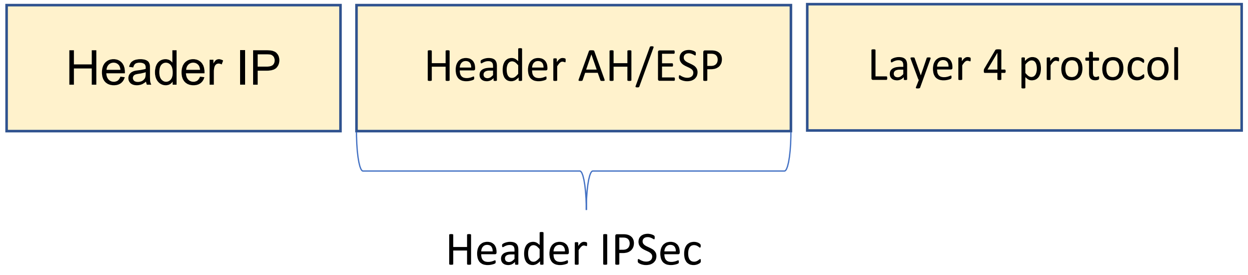
IPSec

An IPSec implementation includes a security association database that defines the parameters associated with each SA.

Technically, IPsec has two main parts.
The first describes two new headers that can be added to packets to carry security identifier, integrity check data, among other information.
The other part, ISAKMP (Internet Security Association and Key Management Protocol), is all about establishing keys. Its main protocol is IKE (Internet Key Exchange)

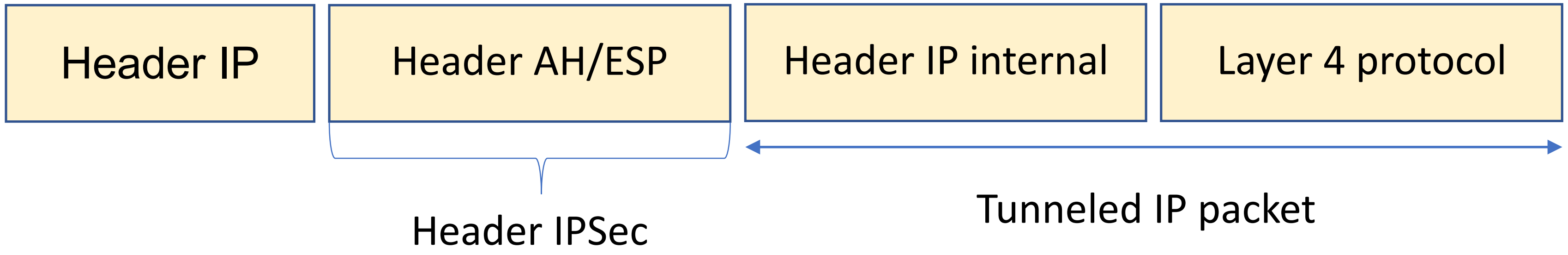
The key management mechanism used to distribute keys is coupled to the authentication and privacy mechanisms only through the security parameter index. Therefore, authentication and privacy have been specified independently of any specific key management mechanism.

IPsec can be used in either of two modes.
In **transport mode**, the IPsec header is inserted right after the IP header. The Protocol field of the IP header is changed to indicate that an IPsec header follows the normal IP header (for example, before the TCP header). The IPsec header contains security information, primarily the SA identifier, a new sequence number, and perhaps a payload field integrity check.



IPSec

In **tunnel mode**, the entire IP packet, header and so on, is encapsulated in the body of a new IP packet with a completely new IP header. Tunnel mode is useful when a tunnel terminates at a location other than the final destination. In some cases, the end of the tunnel is a security gateway machine, for example, a company firewall. In this mode, the firewall encapsulates and decapsulates packets as they pass through the firewall. By terminating the tunnel on this secure machine, machines on the company's LAN do not have to be IPsec aware. Only the firewall has to know about it.



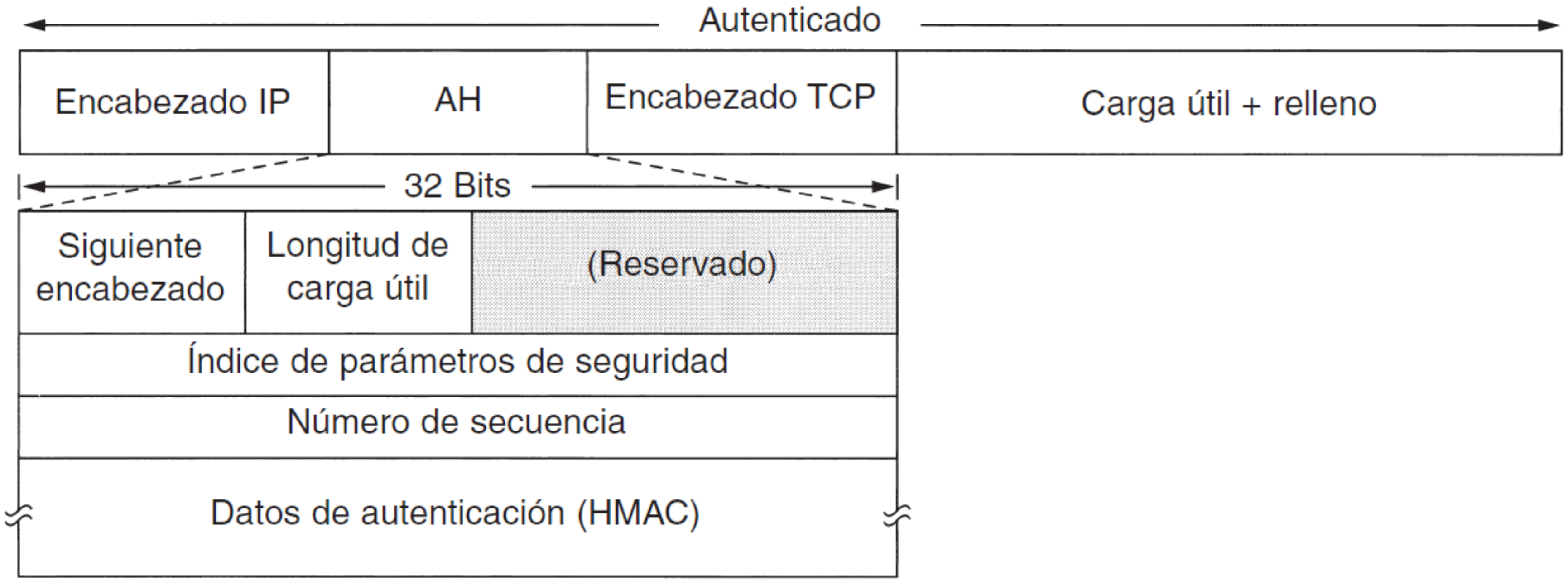
Tunnel mode is also useful when you aggregate a set of TCP connections and handle them as a single encrypted stream because this prevents an eavesdropper from seeing who is sending how many packets to whom. Sometimes just knowing how much traffic is passing through and where it's going is valuable information.

The study of packet flow patterns, even if they are encrypted, is known as traffic analysis. Tunnel mode provides a way to thwart it to some extent. The disadvantage of tunnel mode is that it adds an extra IP header, thereby increasing the packet size substantially. In contrast, the mode of transportation does not affect package size as much.

The first new header is AH (authentication header). It provides integrity verification and anti-replay security, but not confidentiality (i.e., no data encryption).The use of AH in transport mode is illustrated in the figure below. In IPv4 it is placed between the IP header (including any options) and the TCP. In IPv6 it is just another extension headerand is treated as such. In fact, the format is close to that of a standard IPv6 extension header.

IPSec

The payload may need to be padded to some particular length for the authentication algorithm, as shown.



Let's examine the AH heading. The Next Header field is used to store the previous value that the IP Protocol field had before it was replaced with “51” to indicate that it followed an AH header. In many cases, the code for TCP (6) will go here. The Payload Length is the number of 32-bit words in the AH header minus 2. The Security Parameter Index is the connection indicator. It is inserted by the sender to indicate a particular record in the receiver's database. This record contains the shared key used in this connection and other information about that connection. The Sequence Number field is used to number all packets sent in an SA. Each packet gets a unique number, even retransmissions. In other words, the retransmission of a packet gets a different number here than the original (although its sequence numberTCP be the same). The purpose of this field is to detect replay attacks.

IPSec

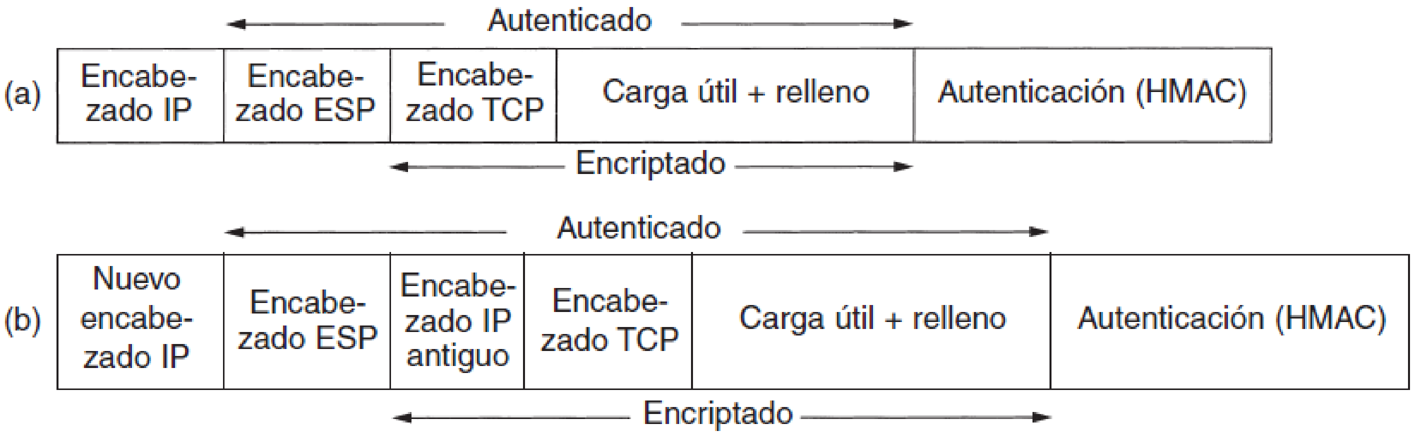
Maybe these sequence numbers don't fit. If all 232 are used up, a new SA must be established to continue communication.

Finally, let's look at the Authentication Data field, which is variable length and contains the digital signature of the payload. When the SA is established, the two sides negotiate which signing algorithm to use. Public key cryptography is generally not used here because packets must be processed extremely fast and all known public key algorithms are very slow. Since IPsec is based on symmetric key cryptography and the sender and receiver negotiate a shared key before establishing an SA, the shared key is used in the signature calculation. A simple way is to calculate the hash over the packet plus the shared key. Of course, this is not transmitted. A scheme like this is known as HMAC (Hash-based Message Authentication Code). It is faster to perform the calculation than first running SHA-1 (Secure Hash Algorithm 1) and then RSA (Rivest, Shamir, Adleman) on the result.

The AH header does not allow data encryption, so it is mainly useful when integrity verification is necessary but confidentiality is not. A worthwhile feature of AH is that the integrity check covers some of the fields in the IP header, primarily those that do not change as the packet moves from router to router. The Time to Live field changes on each hop, for example, so it cannot be included in the integrity check. However, the source IP address is included in the verification, making it impossible for an intruder to spoof the origin of a packet.

The alternative IPsec header is ESP (Security Encapsulation Payload). Its use for transport mode and for tunnel mode is shown in the following figure.The ESP header consists of two 32-bit words. These are the Security Parameter Index and Sequence Number fields that we saw in AH. A third word that usually comes after them (but is not technically part of the header) is the Initialization Vector used for data encryption, unless null encryption is used, in which case it is ignored.

IPSec



ESP also includes HMAC integrity checks, just like AH, but instead of being included in the header, they come after the payload, as shown in the figure above. Placing the HMAC last has an advantage in a hardware implementation. The HMAC can be calculated as the bits are transmitted over the network interface and added at the end. This is why Ethernet and other LANs have their CRCs at the termination of the frame, rather than in a header. With AH, the packet has to be buffered and the signature has to be calculated before the packet is sent, potentially reducing the number of packets/sec that can be sent.

Since ESP can do the same thing as AH and more, and because it is more efficient to boot, the question arises: Why bother having AH? The answer is mainly historical.Initially, AH handled only integrity and ESP handled only confidentiality.

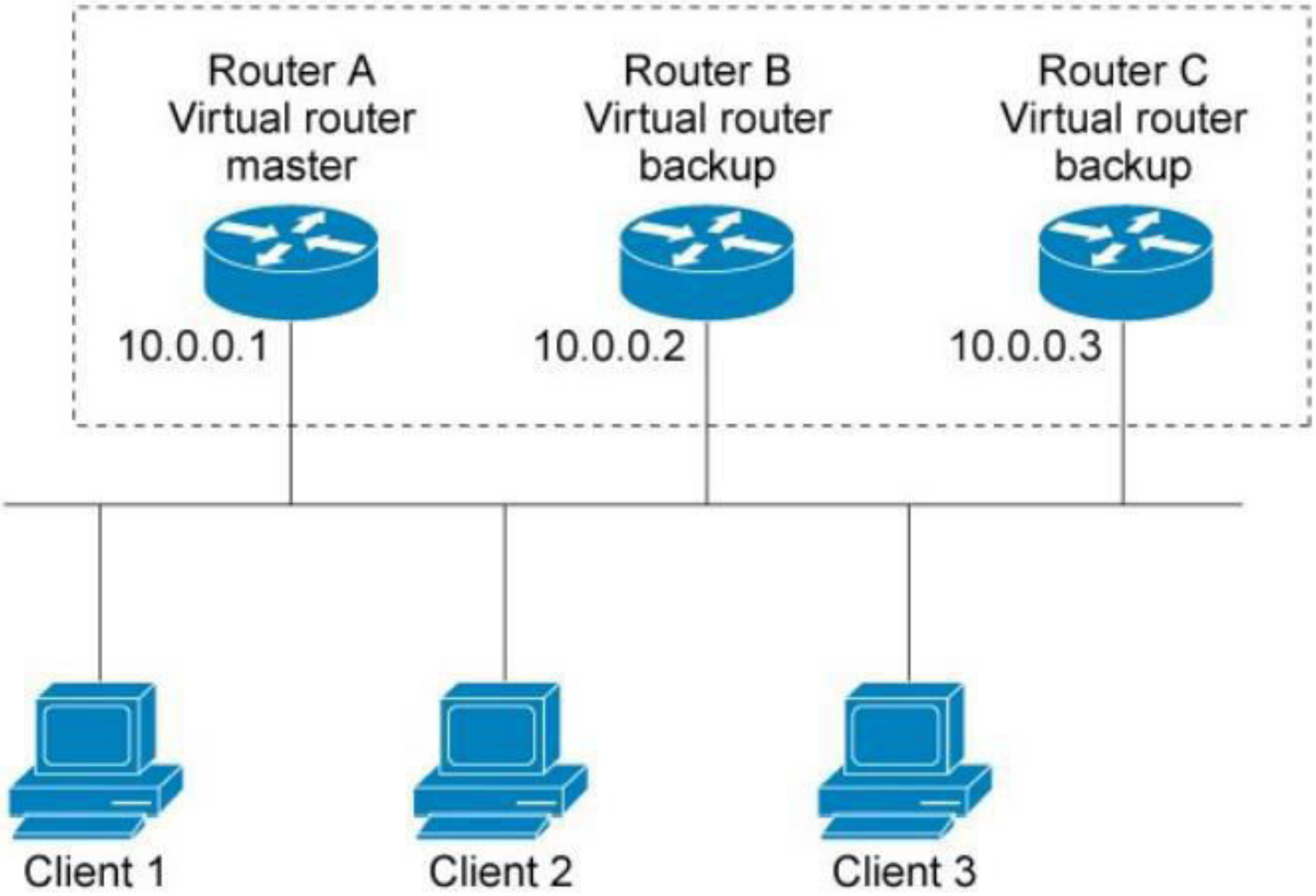
Integrity was later added to ESP, but the people who designed AH didn't want to let it die after all the work they had done. However, your only argument is that AH checks part of the IP header, which ESP doesn't do, but it's a weak argument. Another weak argument is that a product that supports AH and not ESP might have fewer problems obtaining an export license because it cannot perform encryption. AH is likely to be displaced in the future.

VRRP

VRRP (Virtual Router Redundancy Protocol)
VRRP provides redundancy for IP networks, ensuring that user traffic immediately and transparently recovers from failures at the first-hop (DG) router. VRRP allows multiple routers on a LAN to share a virtual MAC and IP address that is configured as the default gateway on the hosts. From the group of routers configured in a VRRP group, there is one router chosen as the active router and another as the backup router. The active router assumes the role of forwarding packets sent to the virtual IP address. If the active router fails, the standby router takes over as the new active router.

The VRRP protocol is a protocol that provides automatic assignment of available routers to participating hosts. This increases the availability and reliability of routing via automatic selections of default gateways on an IP subnet.The protocol does this by creating virtual routers, which are an abstract representation of multiple routers, such as master and backup routers, acting as a group. The default gateway of a participating host (on the LAN) is assigned to a virtual router instead of a physical router.

The following figure illustrates a basic VRRP topology. In this example, routers A, B, and C are running VRRP and together form a virtual router. The address of this virtual router is 10.0.0.1 (the same address as the physical interface of Router A). Because the virtual router uses the IP address of Router A's physical interface, Router A is the master VRRP router, while routers B and C serve as backup VRRP routers. Clients 1, 2 and 3 are configured with the default gateway IP address 10.0.0.1. As the master router, Router A forwards packets sent to its IP address. If the master virtual router fails, the router configured with the highest priority becomes the master virtual router and provides uninterrupted service for hosts on the LAN. When Router A recovers, it becomes the virtual master router again.



Virtual router group
IP address: 10.0.0.1

INTRODUCTION TO SDN

INTRODUCTION TO VIRTUALIZATION

VIRTUALIZATION IS NOT NEW AT ALL

- Over the last years revolution in the way IT works from an infrastructure perspective, took place.
- Purchasing and deploying a new server for each new application, is not a good idea anymore
- Instead, IT has become much more adept at sharing existent underutilized resources, what is called *virtualization*.

- Back in the days of mainframe computers, IT would segment processing capacity to provide smaller logical CPUs.
- Similarly, networks have used virtualization for years to create segregated logical networks that share the same wire (for example, virtual local-area networks [VLANs]).
- Even in desktop computing, IT has logically partitioned large hard disks to create several smaller independent drives.

So, technique of virtualization has existed in processing, storage, and networking for several decades. What is new or different is the concept and development of the virtual machine (VM).

From VMWare site:

> “A VM is a tightly isolated software container that runs its own operating system and applications as if it were a physical computer.”

But….. Why or how these things have transformed servers, storage, and networking?

WHAT IS VIRTUALIZATION?

Today’s x86 computer hardware was designed to run a single operating system and a single application, leaving most machines vastly **underutilized**. Virtualization lets you run **multiple virtual machines** on a single physical machine, with each virtual machine **sharing the resources** of that one physical computer across multiple environments. Different virtual machines can run different operating systems and multiple applications on the same physical computer.

Source: Vmware

SERVER PROLIFERATION BEFORE VIRTUALIZATION

Each application often requires its own server

Over the past 30 years, we have become a data-driven society that relies on computers to help run both daily lives and businesses
Enterprises need to run applications for nearly every facet of their business, and all these applications have to run on servers.

In fact, *each* application often requires its *own* server to run on even when the server is not used to its fullest capacity by the application.

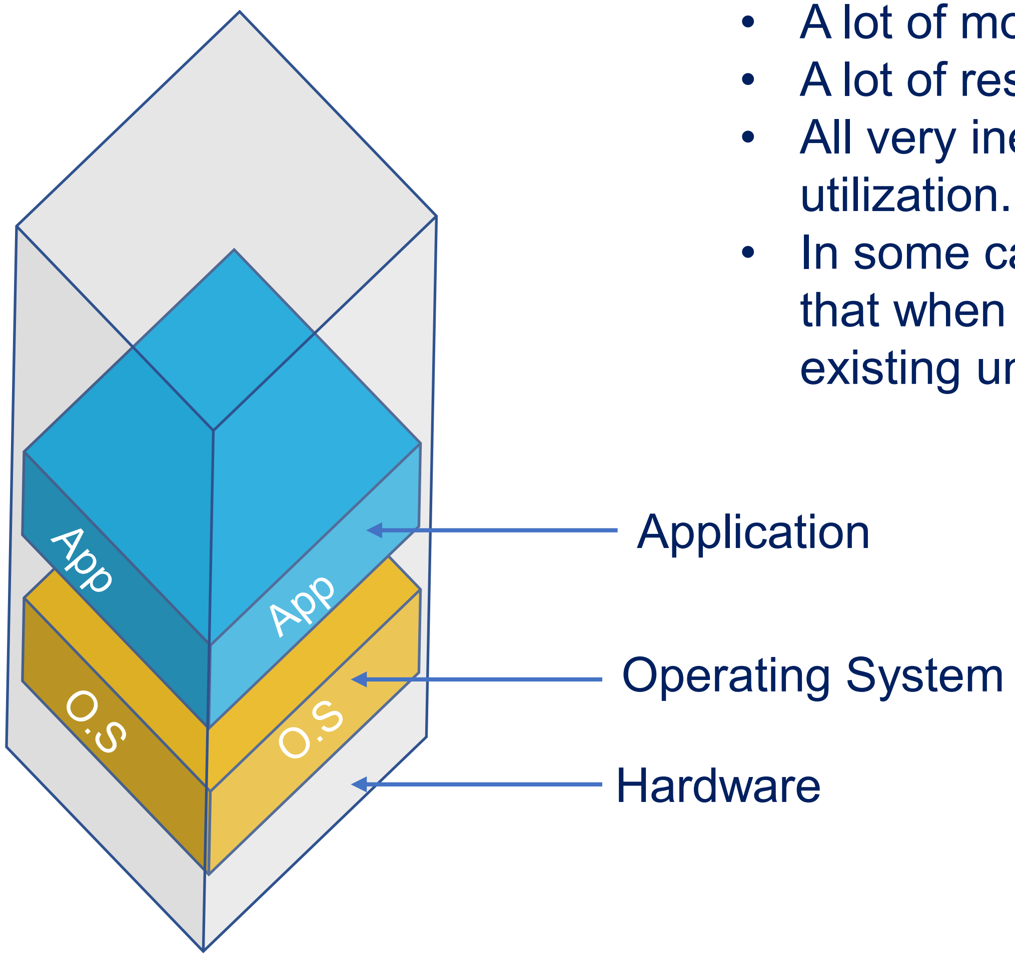
In fact, this has led to a phenomenon known as *server proliferation* It refers to the ever-increasing need to buy more and more single-use servers to account for increasing data and application usage.

Data proliferation is compounded by the need for businesses to always have their computing resources available, effectively doubling or tripling the number of servers that a company needs,

SERVER PROLIFERATION BEFORE VIRTUALIZATION

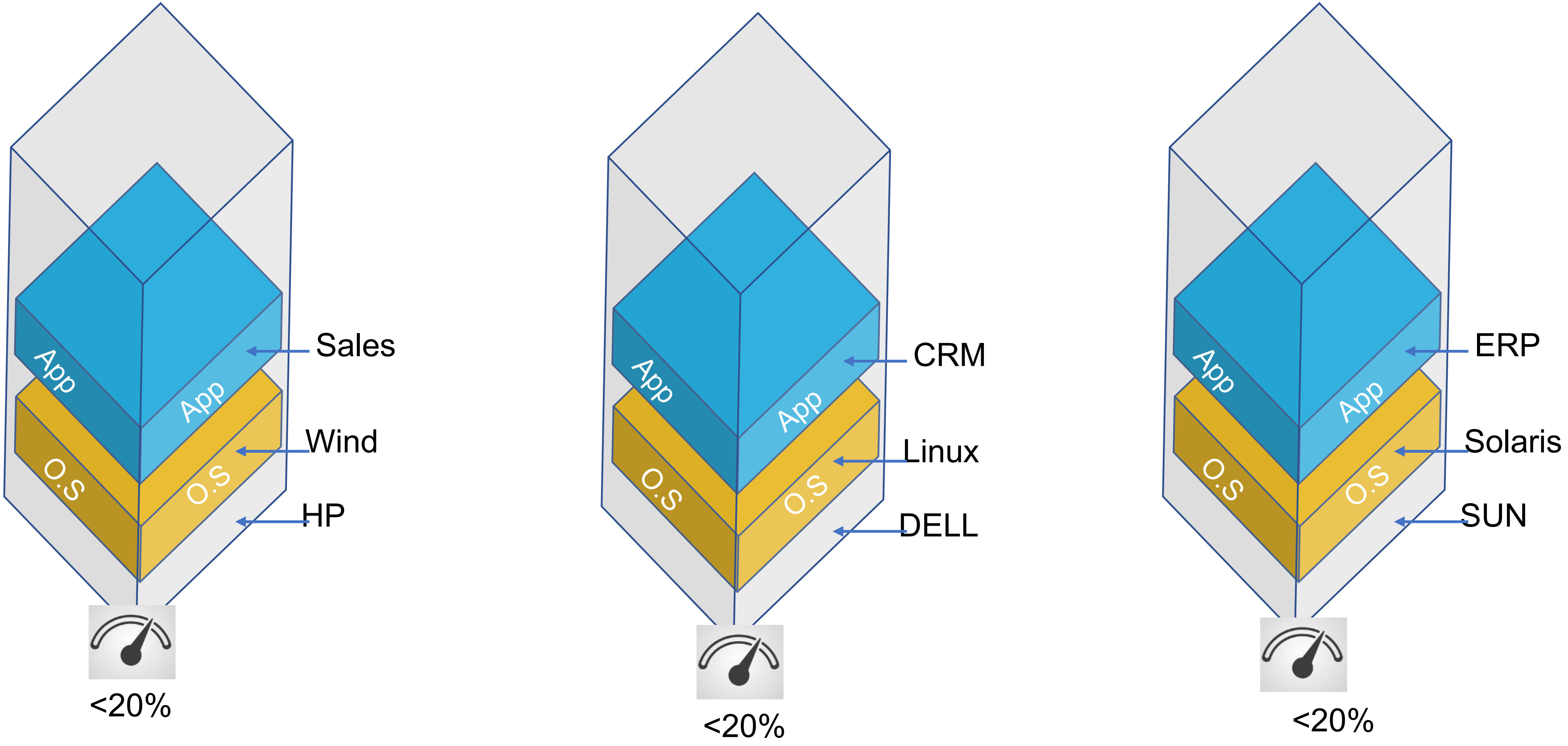
Server proliferation

- A lot of money.
- A lot of resources (power, cooling, and personnel to manage it all).
- All very inefficient. servers are only running at about 5% to 10% utilization.
- In some cases servers have a 1:1 OS to application ratio. This means that when a new application is needed, we can't just load it on an existing underutilized server; we have to buy a new one.



INEFFICIENCY

SERVER PROLIFERATION BEFORE VIRTUALIZATION



Sales
Wind
HP
<20%

CRM
Linux
DELL
<20%

ERP
Solaris
SUN
<20%

SUN was adquired by Oracle

Solaris,
Parte de Unix System V
ORACLE
Solaris

SERVER PROLIFERATION BEFORE VIRTUALIZATION

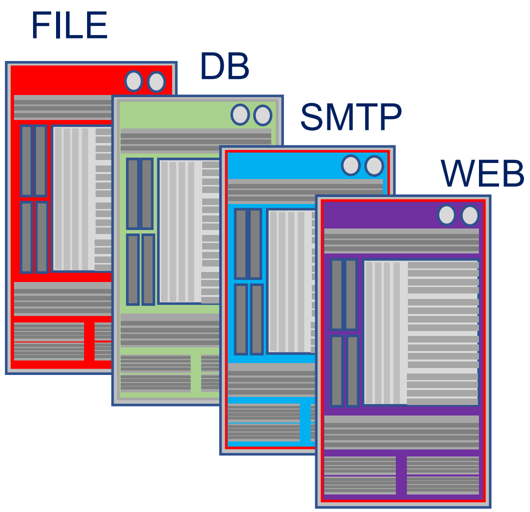
Many servers, each running at about 10% to 15%.
Servers backed up in case a power outage or some other disaster occurs. So more servers
All servers about 90% of their computing resources
Servers have to be powered on all the time
Even idle servers generate a lot of heat; so, we need to keep them cool, and that takes a lot more power.
Servers are often more expensive to cool than they are to keep on.
There's also a matter of real estate. Server farm can grow really quickly, space-planning issue on top of everything else.

INEFFICIENCY

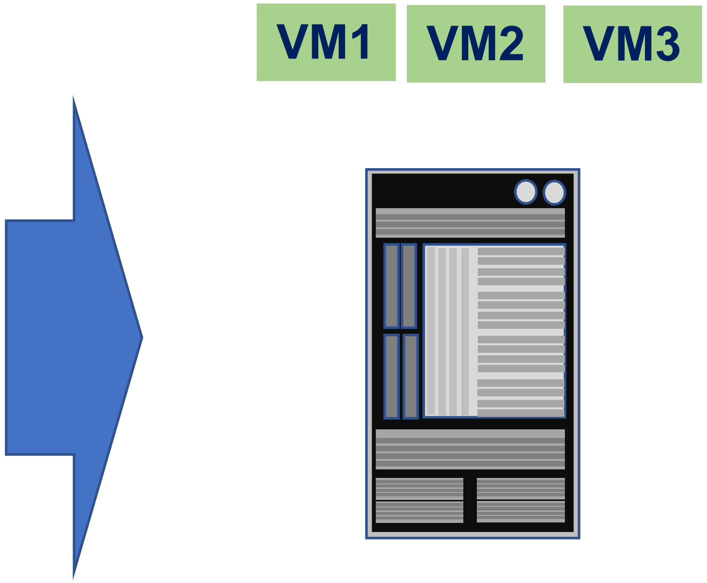
Need for a new application
IT team has to go through the process of specification, procurement, get servers installed, making sure they are replicated, loading the application, so on

A lot of work and a lot of expense, and even servers are mostly underutilized and many are just running on idle.
This problem is exactly what VMs fix.

INTRODUCTION TO CLOUDING



- New service: new server
- Physical Space
- Energy
- Internal infra (switches, cable,..)
- Administration
- Failure

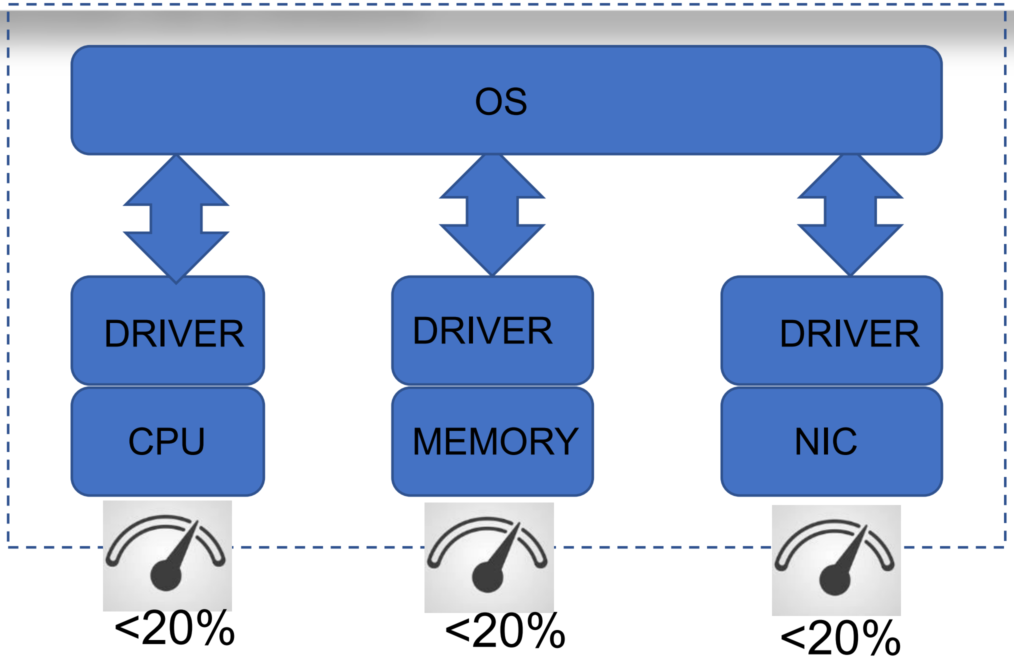


Common HW supply all resources, reduce costs

HOW SERVERS WORK

A server, like most computers, is a collection of specialized hardware resources that are accessed by a software OS through a series of specialized drivers. These resources can be many things, but commonly consist of the following:

CPU processor: Does the computing part
RAM (or memory): Stores and stacks short-term instructions and information
Storage: Keeps long-term data
Network interface card (NIC): (Pronounced “nick”) Allows the machine to connect to other computers or devices via a network



Once the OS and drivers are loaded, the hardware is locked in

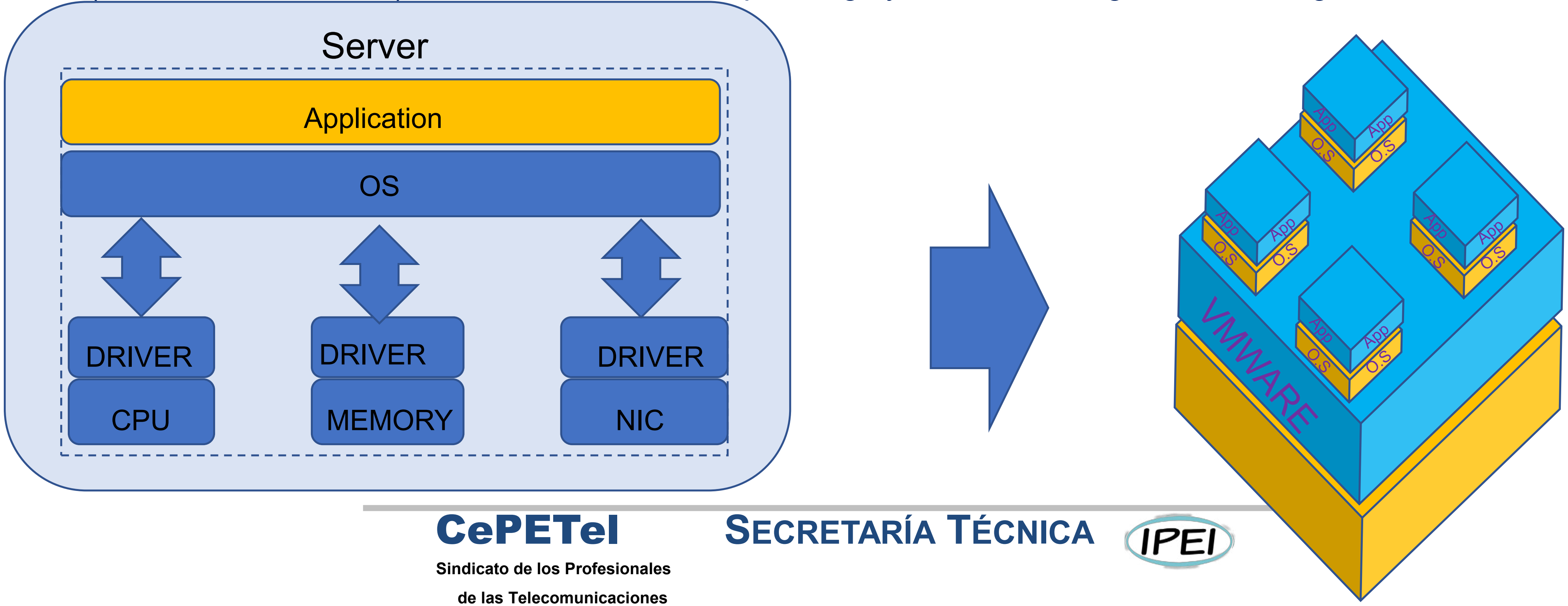
Back in time, utilization rates were higher mainly because of the limited capabilities of computers, and because there were few applications

HOW SERVERS WORK

The application is loaded onto the OS, thus becoming locked in to the OS, drivers, and hardware.
The Job of server is to run the application. That’s all !!

Now this might be confusing to people who’s only experience with applications is what they load onto their phone or tablet. After all, those things hold tons of apps, and they’re really little; so why can’t these big servers do more than one thing?

The applications running on servers are not simply or light applications. They are much more complicated and do things such as run email systems for large enterprises, run a **Packet Core** or **IMS** or **RAN** (in Telco networks).
The operating systems are dedicated to the application and are the link to the drivers, which enable access to the hardware resources (CPU, RAM, and disk). Also, it turns out that operating systems are not good at sharing.



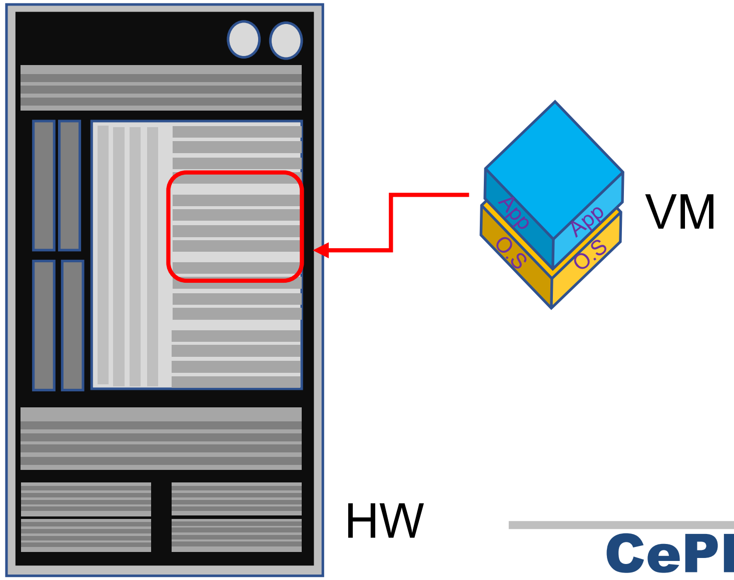
HOW TO FACE THE UNDERUTILISED SERVER HW

A VM is like a software version of a server that runs within the hardware server (physical or main server). VM is an exact copy of the physical server. Many VMs can run on the same physical server, and they can all be customized to the application needs (using a descriptor). A physical server can now mix together Windows and Linux applications on the same piece of hardware.

In addition, because VMs are software, not only can you run them on any server but you can also suspend or freeze a VM and then you can move it to a different server any time you want. And it's all quick and easy.

It is also possible to make a copy of a VM or even a whole bunch of copies and spin them up on multiple servers.

The Virtual Machine and the hypervisor!!!

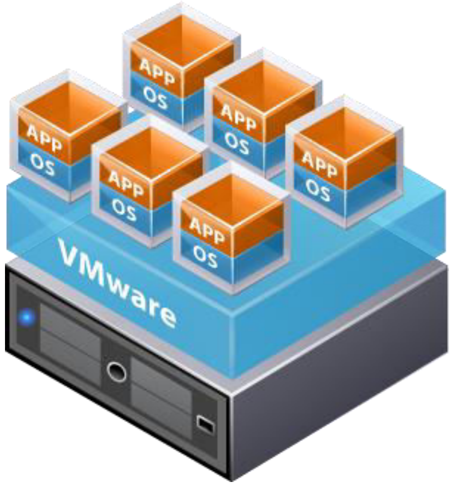


In the early days it was not easy to move an OS from one HW to another, the OS was so tightly coupled to the server because it could only access the server's resources through Drivers, and those drivers were written for specific operating systems: The servers were able to run only a single application.

Now we need to access those resources, and we still need to do it through the drivers, so how does that work with a VM?

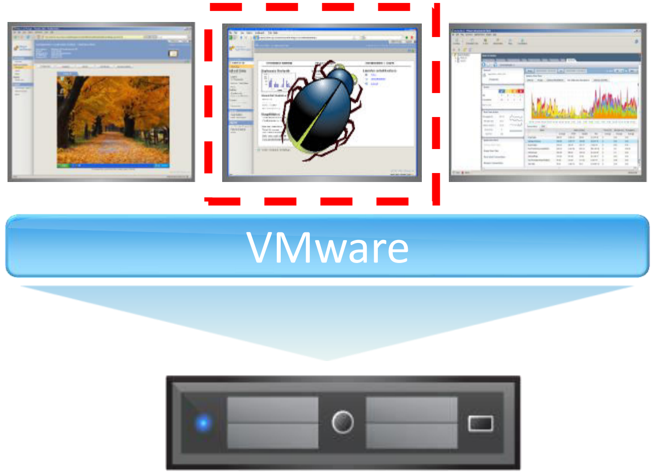
The hypervisor!!!

KEY PROPERTIES OF VIRTUAL MACHINES



Partitioning

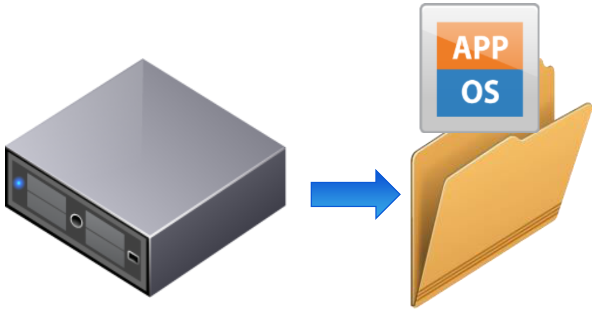
- Run multiple operating systems on one physical machine
- Divide system resources between virtual machines



Isolation

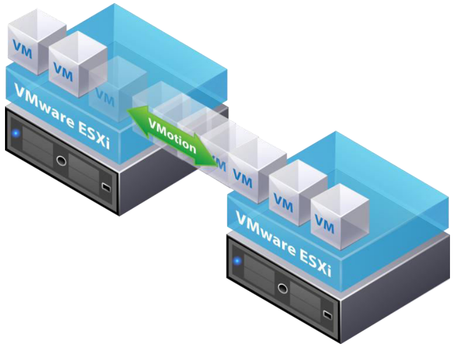
- Fault and security isolation at the hardware level
- Advanced resource controls preserve performance

KEY PROPERTIES OF VIRTUAL MACHINES



Encapsulation

- Entire state of the virtual machine can be saved to files
- Move and copy virtual machines as easily as moving and copying files

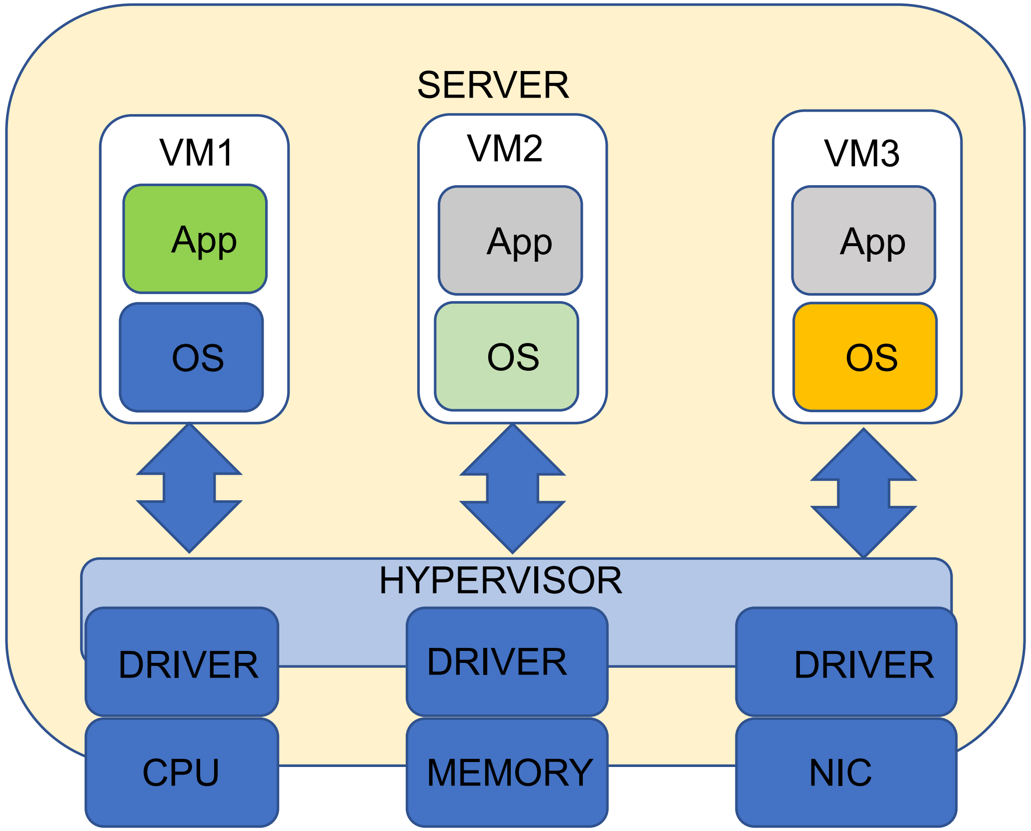


Hardware Independence

- Provision or migrate any virtual machine to any similar or different physical server

THE HYPERVISOR

A hypervisor is piece of software that sits between the OS and the server's specialized hardware resources. When in place, it's the hypervisor, rather than the OS, that manages the connections between the drivers and the server's resources.



But…..a new layer has been added in. So, how can this lead to greater efficiencies?" That's a good question.

The answer is that a hypervisor can present a *virtual connection* to the server's resources for a VM's OS. What's significant, though, is that it can do it for a lot of VMs all running on the same server, even if those VMs all have different operating systems and unrelated applications.

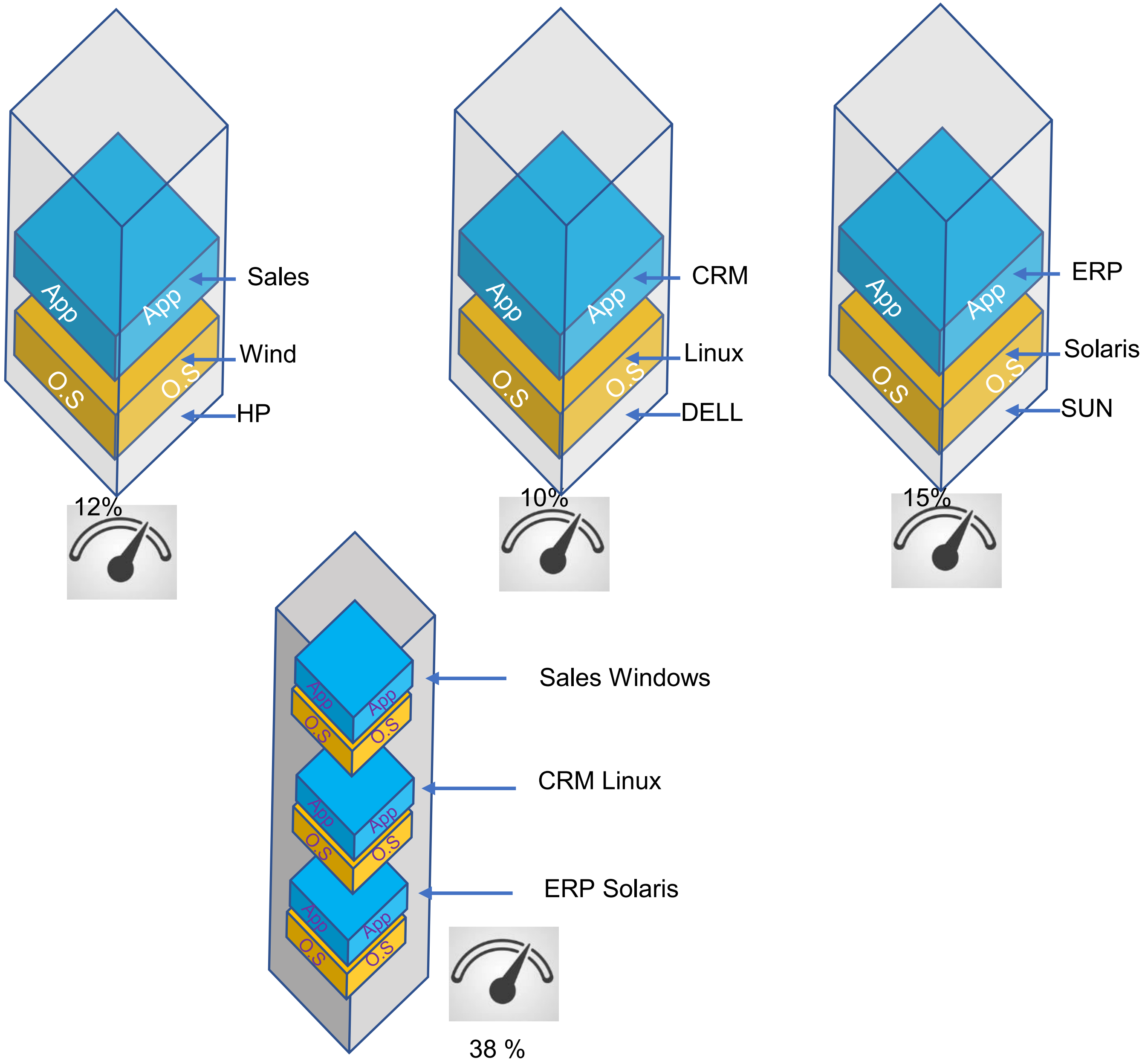
EFFICIENCY, FLEXIBILITY, HA

Efficiency: It has changed the IT, storage, server and networking industries in profound ways.
Even being cautious to keep the CPU below 80%, we can still achieve anywhere from a 6x to 15x reduction in the number of physical servers.

There are other efficiencies as well. With a 10x reduction in the number of servers, companies enjoy a massive reduction in energy costs for server power and cooling.

Flexibility: It's relatively easy to migrate the servers within an entire data center to another data center in a matter of hours with little or no disruption in service.

Virtualization also results in a much better **disaster recovery capability** and nets a huge decrease in the time it takes to provision new applications.



VM: A BIG DEAL TO UNDERSTAND SDN

An elastic network

VM is a big deal because it means a transformation in how enterprises provision, deploy, operate, manage, and back up computing resources.

Virtualization tools means eliminate a great deal of inefficiency.

But there were still a lot of inefficiencies in the system. For example, what if there is a lot of volatility in the amount of computing resources you need (scale up scale down)? This is a common issue for ecommerce vendors during the holidays, when there is a big spike in computing activity.

There's also the matter of designing, tuning, or modifying the network to accommodate **changing** needs in usage, availability, security, or other factors. It turns out that for all the benefits of virtualization with respect to server operating systems and applications, there are still massive gains to be had by virtualizing the entire network infrastructure.

The concept of *virtualizing the network* leads us to the thrust of this course: software-defined networking (SDN). This concept is every bit as profound as server virtualization. Our goal is to understand and discuss SDN, but first we have to lay the foundation by going a little deeper into virtualization, which will lead up to "the cloud," and from there to SDN.

BENEFITS OF VMs

The x86 server architecture (the way modern servers are designed and built) supports only one operating system at a time.

Each operating system is typically dedicated to a single application.

It would be great if we could run lots of applications on one standard server.

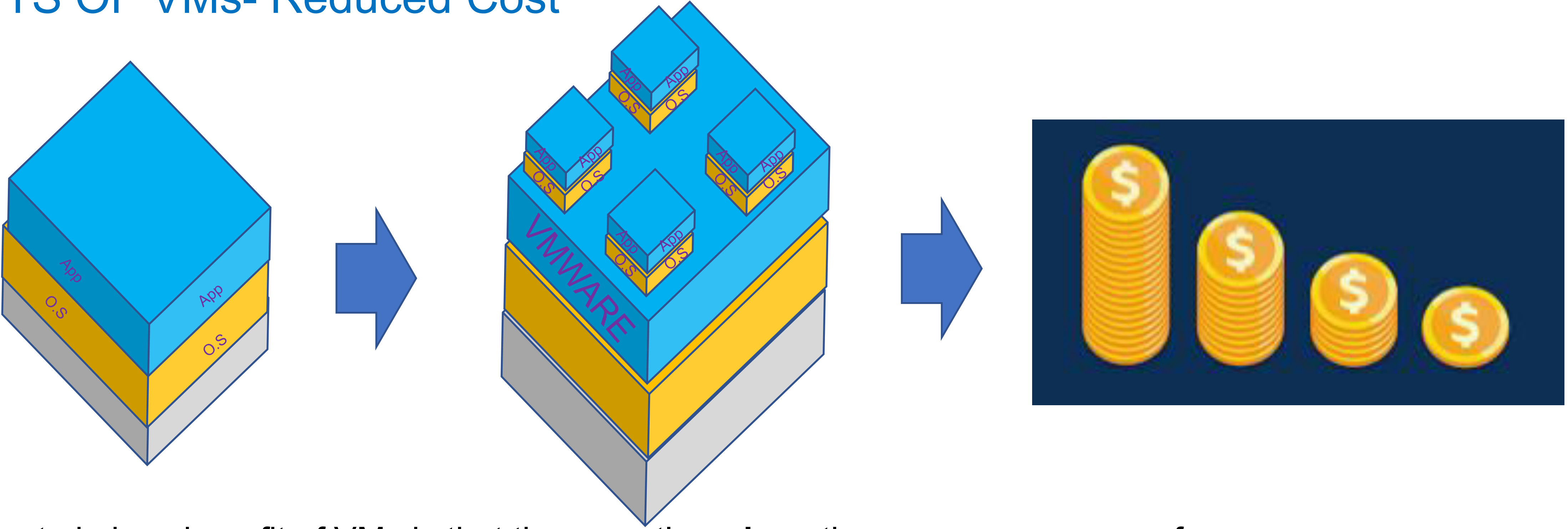
Despite the inefficiencies, there were a few benefits of the one-server/application model. Not only does this model minimize resource competition and system maintenance on any given server, but it also isolates application failures

If there is more than one application running on the same server, the process of rebooting the operating system (OS) takes all the applications on that server out of service.

The good news is that with a VM running on a server you can reboot the VM (the app and the OS) without impacting the other VMs on the server.

Further, if the server needs to be reset, it's a simple matter to suspend the VMs, move them to other servers, and then reactivate them or to even ***live migrate* them using tools like *vMotion***.

BENEFITS OF VMs- Reduced Cost



The most obvious benefit of VMs is that they greatly **reduce** the runaway expenses from server proliferation. These costs are compounded by the extremely inefficient usage and the costs associated with 24/7 power and energy consumption, which contribute to the rising temperature of the data centers and consequently the high costs of cooling the environment.

-Several operating systems and applications to run on one physical server.

-This in itself solves the problems of server inefficiency and addresses the wasted power and cooling costs.

-VM technology solves the IT best practice dilemma of one application per server/OS

-Each VM is isolated from any other VM hosted on the same server

-Each VM uses only the amount of server resources it requires.

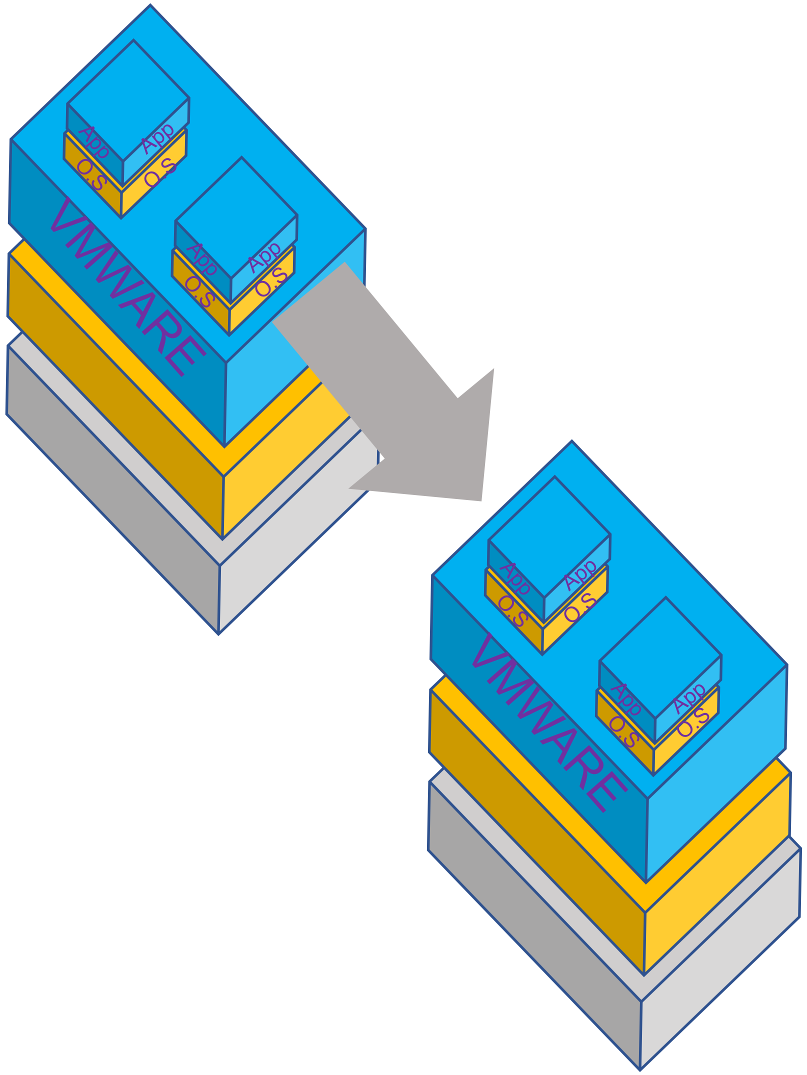
BENEFITS OF VMs- Less Space

-In the large data centers, the cost of the actual hardware is only a fraction of the total cost-
-Other costs include real estate (land), buildings, power, racks, and cables.
-The cost of a data center is roughly double (Redundancy) depending on the cost of land and power.
-One of the biggest considerations with spacing is the rate of growth.
-There is also the cost of power, which is a major financial consideration. All of these servers generate an enormous amount of heat, and the servers need to be kept cool.
-Another consideration in the cost of data centers (especially single-use server data centers) is cabling.

With VMs, all of these costs are significantly reduced by factors of 3x to 10x in a virtuous cycle of cascading benefit. Fewer servers, less space, lower power costs, fewer cables...

BENEFITS OF VMs- Reduced Cost

Cost savings through server consolidation may be the most obvious benefit to server virtualization, but there are others.
-Application availability, fault tolerance, and flexibility.
-With VMs, no need for clustered or fault-tolerant hardware with complex failover protocols and configurations
-If a server hosting several VMs were to fail, all the VMs would continue to run or restart on another server without any downtime or data loss.
-The centralized controller manages this by monitoring each VM's state using a heartbeat function
-Server virtualization using VMs has an important role in disaster recovery and business
continuity. IT can move a VM from one server to another as simply as if copying a file.

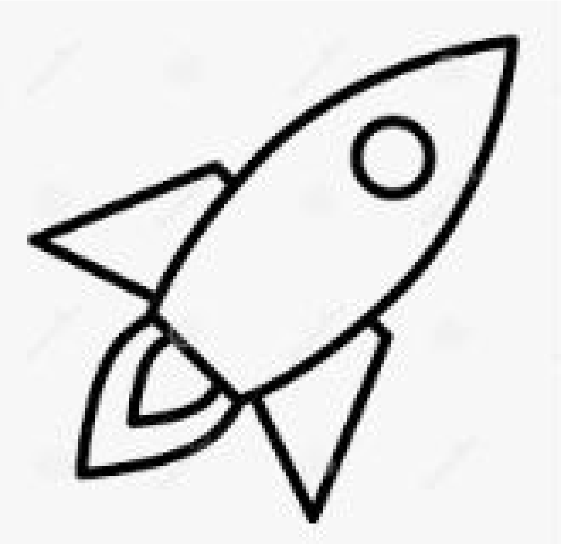


BENEFITS OF VMs- Faster Application Spin-Up and Provisioning

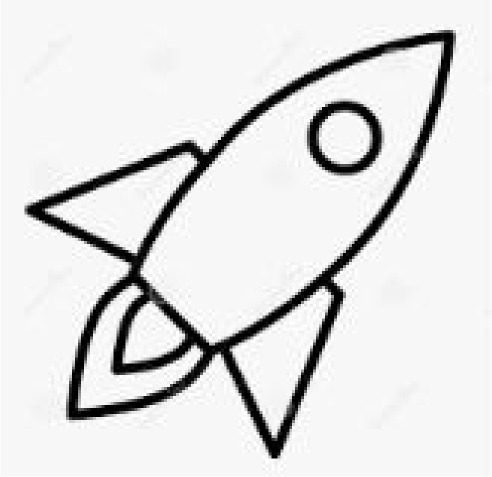
VM technology brings a lot of benefits to systems and operations (SysOps) and development operations (DevOps)

In the past…..building, testing, developing, and publishing servers for application developers was a tedious, difficult, and expensive task because of the different test environments required for every development project.

Now ……. VMs can be installed and started when required, and then shut down and stored between projects, without using up any precious resources or conflicting with other test and development environments.



BENEFITS OF VMs- Faster Application Spin-Up and Provisioning



VMs come with tools that enable granular server management by allocating, regulating, and enabling the fine-tuning of the computer resources that each VM may use.

VMs naturally promote standardization. The VMs only see what CPU is on the host. Therefore, the host server hardware can vary, as it often will, but the VM will always be a standard image across all the servers regardless of the underlying hardware.
VM administration tools also facilitate the creation and management of VM instances: a VM exists as a single file. Furthermore, a VM file can be created, stored, or proliferated across the organization with adjustments for each user or application.
Another administrative advantage is that VMs are copied (backed up) directly to network storage, most often a storage-area network (SAN), using periodic snapshots.

BENEFITS OF VMs- Easier access for development

Engineers and developers can ease and affordability gain access to and use applications via VMs.

In the past….. Getting access to an application meant getting our own dedicated server.
This added to the server proliferation, and not only did this raise costs for their respective departments, but these requests also had to be served by what was already an overworked IT group.
Time lag of ordering and receiving the hardware and cables, getting the application installed, waiting for a maintenance window to bring it on line...

Now ….. with VMs, in matter of minutes, an engineer or developer can download and install a virtual version of an application and install it on an inexpensive server (or in the cloud), but more on that later. Once installed, the application can be easily replicated so that each team or each developer can have his or her own version to work on.

HYPERVISORES - Vmware, KVM, others

A hypervisor is:

- An operating system for operating systems
- A Virtual Machine (VM) Monitor

An operating system for operating systems

VMs are described as self-contained operating systems that run applications. They are “abstracted” from the hardware.
But, This abstraction, however, is problematic, because it’s the operating system that provides the application with access to the server’s resources.

The hypervisor is a piece of software that allows the computer’s hardware devices to share their resource between VMs running as guests and sitting a top the physical hardware. In one type of hypervisor, the software sits directly on top of the hardware—no server OS is loaded—and the hypervisor interacts directly with the guest VMs.
From this perspective, it’s easy to see why we can consider the hypervisor as an operating system for operating systems. For example, the hypervisor has replaced the server’s own OS and as such has taken responsibility for interacting between the hardware devices and the VM’s internal OS. It’s doing for the operating system portion of a VM just what the server’s own operating system would do for applications on a dedicated machine.

HYPERVISORS - Vmware, KVM, others

A hypervisor is:

- An operating system for operating systems
- A Virtual Machine (VM) Monitor

A Virtual Machine (VM) Monitor

-Multiple VMs can run on a single server.

-The hypervisor must also provide a monitoring function for each VM to manage the access requests and information flows from the VMs to the computing resources and vice versa.

-The VM functionality that a hypervisor plays is critical. Not only must the hypervisor successfully allow multi-VM access to the hardware via a process called *multiplexing*, but it must do this in a way that is transparent to the VMs.

-Incidentally, multiplexing used to be a core function solely of operating systems—the fact that hypervisors do it too supports the idea that a hypervisor is an operating system for operating systems.

-The VM function of hypervisors is what makes transparency of operation possible. Transparency of operation means that we can not only do all the things we used to do, and use all the programs you already use, but that we can do these things without any noticeable difference in how the applications and programs work

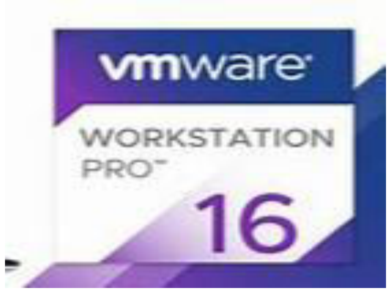
TYPE OF HYPERVISORS

Hypervisors type 1.

Operating systems specially designed for virtualization. They are called a *bare-metal* hypervisor, it runs directly on the server hardware without any native operating system. The hypervisor is the operating system of the server providing direct access to the hardware resources for the guest VMs riding on top of it. Most of the servers in production today are using Type 1 hypervisors.

Hypervisors type 2.

Usually installed over other Operating systems. They are called *hosted hypervisors*, run on top of a native OS. In this use case, the hypervisor is a shim that sits between the OS of the VM above and the OS of the server or computer below.

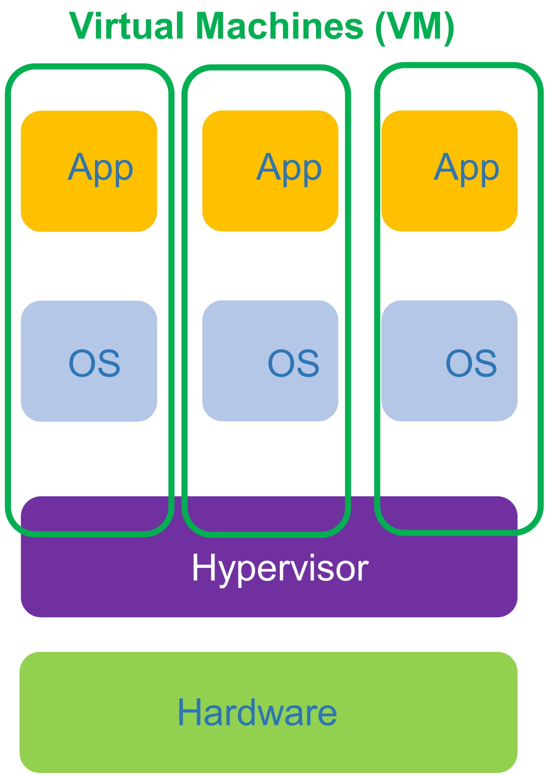


TYPE OF HYPERVISORS

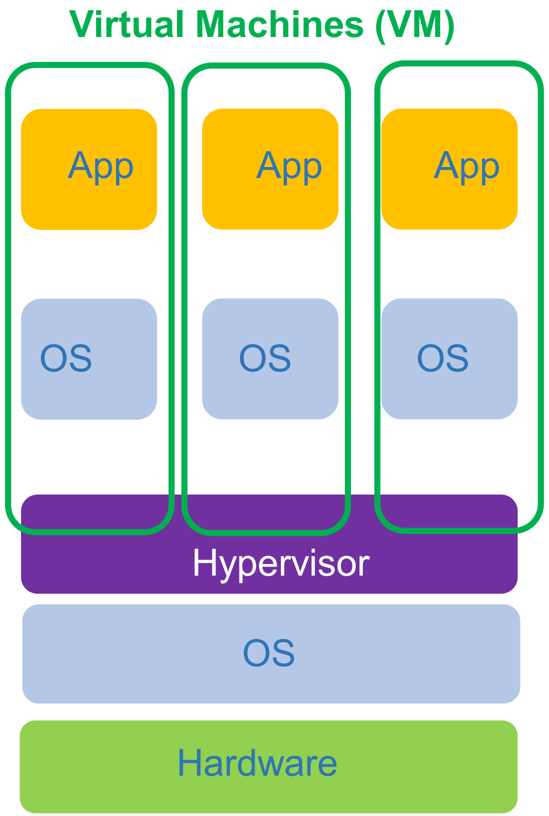
The hypervisor is the bridge between the operating systems and the server hardware. It carries the input/output (I/O) commands and interrupt commands from the virtualized OS to the hardware.

The other main function of the hypervisor is that it also manages both the metering (usage) of the various resources as well as the network access.

Hypervisors type 1.



Hypervisors type 2.



HYPERVISORS VENDORS

As data centers proliferated, the battleground for market share was around the operating systems. Sun (today Oracle), Microsoft, and various Linux providers (such as Red Hat) fought hard to get their operating systems purchased and installed on as many servers as they could

Today, a similar fight for market share is being waged, this one on hypervisors and Container Engines

VitualBox	Oracle
Virtuozzo	Virtuozzo
Apple Hypervisor	Apple
Parallels	Parallels Desktop for Mac
Qemu	Quemu
Vmware	Vmware ESXi
Vmware	vSphere Hypervisor
Vmware	vMware WorkStation Player
Joyent	Triton Smart
Red Hat	Red Hat Virtualization
Microsoft	Microsoft Hyper-V

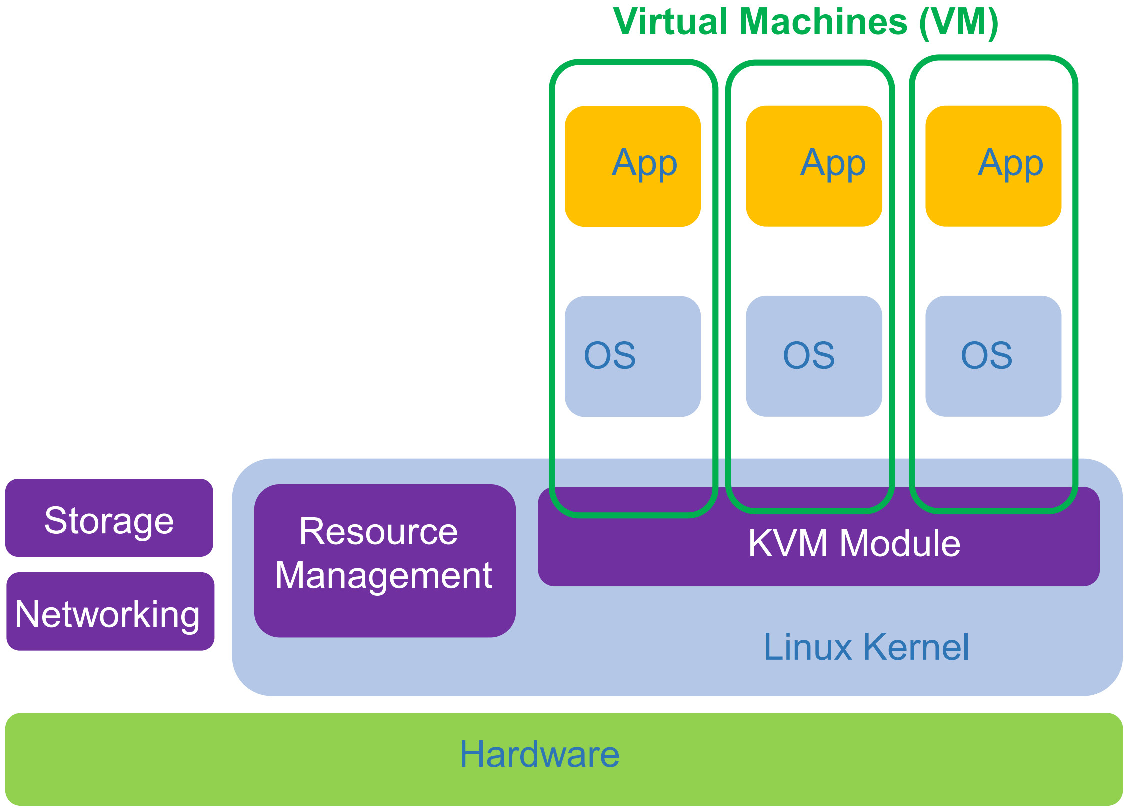
Oracle	Oracle VM
KVM	KVM Red Hat
Xvisor	Xvisor
L Guest	L Guest
Green Hills	µ-visor
Proxmox	Proxmox VE
oV	oVirt
Vmware	Vmware Fusion
Canonical	LXD
Codeweavers	Cross Over

https://sourceforge.net/software/hypervisors/

HYPERVISORS VENDORS- KVM

KVM (acquired by Red Hat) is an acronym that stands for Kernel-based Virtual Machine. A kernel is the part of the OS that interfaces directly with the hardware. The kernel as the main part of the OS. If we strip away all the code that makes Windows look and act like Windows and Mac OS look and act like Mac OS, what's left is the base code that accesses the CPU, memory, and disk. That's the kernel.

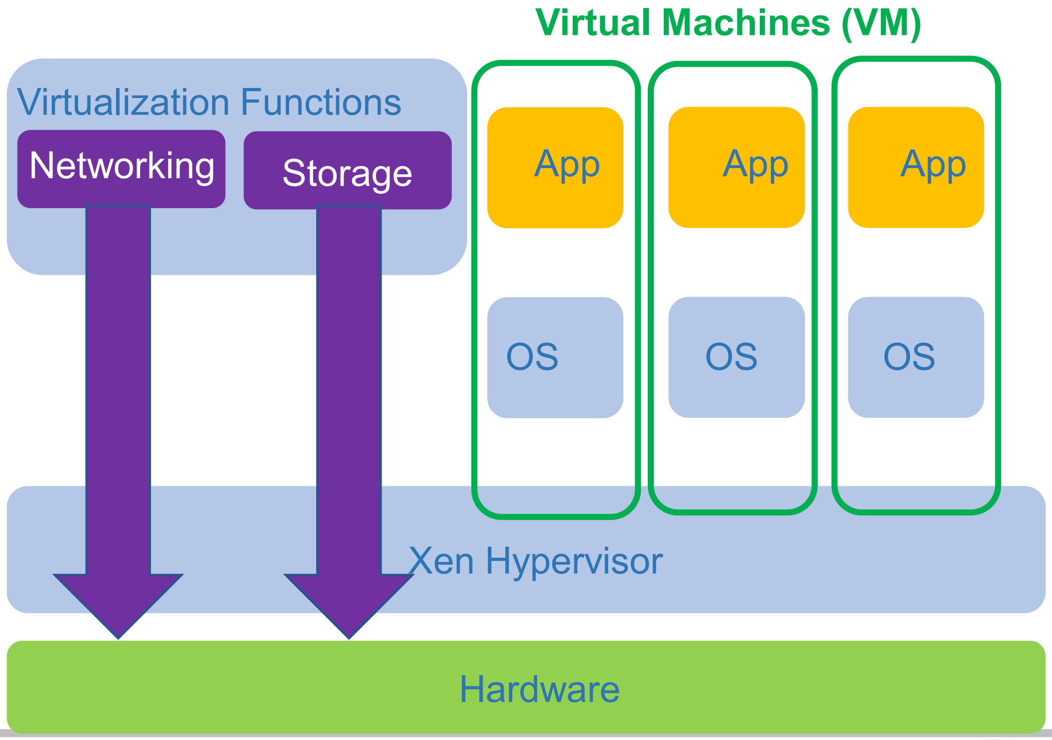
KVM is a part of Linux and is a *Type 2*, or *hosted hypervisor*. Type 2 hypervisors are a little easier to install and operate. However, because there is an added layer, the performance is not always as good as hypervisors that run on bare metal.



HYPERVISORS VENDORS- XEN

Xen is a Type 1 hypervisor based on open source code. Xen has been acquired by Citrix. Typically, the open source community (a large group of independent coders contributing to the code development) will develop base code for an open source package, and then companies will form based on a business model where they stabilize, test, package, and support their version of the open source software. Red Hat is the most notable company to have done this. (They did it with the Linux code base mentioned in the previous section back in the 1990s.)

Xen uses *para-virtualization*, which means that the guest operating systems are made aware of the fact that they are not running on their own dedicated hardware. This requires some modification of the guest OS, but this disadvantage is made up for with increased performance as the OS modification essentially “tunes” the guest to operate more efficiently in a virtualized environment.

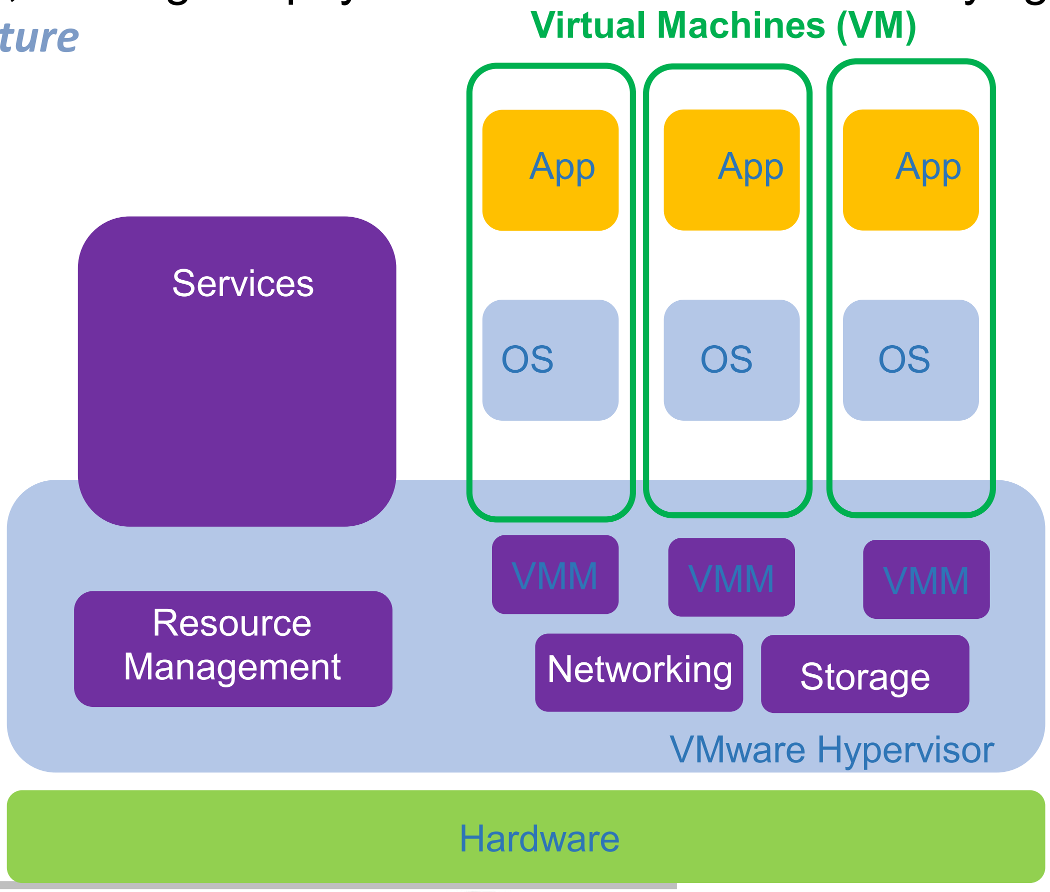


HYPERVISORS VENDORS- Vmware ESXi

VMware leads the hypervisor market in both revenue and total share.
VMware offers both a Type 1 hypervisor called **ESXi** (their main server-based hypervisor) and a popular Type 2 (hosted) hypervisor called **VMware Fusion**, which runs on desktops and laptops.
vSphere 5.1 (ESXi) is a bare-metal hypervisor, meaning that it installs directly on top of the physical server and partitions it into multiple VMs that can run simultaneously, sharing the physical resources of the underlying server.

vSphere is a Free Version of VMware's Leading ESXi Architecture

The ESXi version of the VMware hypervisor is based on the original ESX hypervisor. VMware claims that this version is less than 5% of the size of the original ESX.



HYPERVISORS VENDORS- Microsoft Hyper-V

Microsoft, the largest OS provider, is also a major player in the hypervisor market.
Their virtualization platform, now called Hyper-V (originally it was called Windows Server Virtualization), was originally released in 2008, and now has versions for several of the server platform operating systems.

- The first consideration in choosing a hypervisor is whether we want a hosted or bare-metal hypervisor.
- Hosted hypervisors are easier to install, but they do tend to run at less than full capacity due to the extra layer.
- They do have greater flexibility, though, and if we already have servers running a specific operating system, the transition to virtualization is easier.
- Native hypervisors typically offer better performance for the VMs running on them because they are directly connected to the hardware.
- This improved latency response is an important factor, especially in a service model where there may be service level agreements (SLAs) that specify server and application performance.

MANAGING VIRTUAL RESOURCES

One of the top benefits of VM is the improved administration and management of the virtual machines. Virtualization provides the tools and opportunities to create, configure, and manage virtual machines (VMs) in a way administrators could never have previously imagined. Examples of the virtual resources that can be managed through common administrative tasks include the following:

- Creating VMs, from scratch or from templates
- Starting, suspending, and migrating VMs
- Using snapshots to back up and restore VMs
- Importing or exporting VMs
- Converting VMs from foreign hypervisors

VM,

- Make easy to spin up on-demand instances of a virtual server.
- Enhances the administrator’s time to restore service after a failure.

However, what does it take, in terms of knowledge, effort, and time to create these VM instances?

MANAGING VIRTUAL RESOURCES

All the knowledge that an administrator requires for most vendors' 'wizard' driven applications is to know some key parameters in order to create the VM. These key parameters are as follows:

- The physical server's hostname.
- A VM name.
- The defined memory allocation assigned to the VM.
- The number of cores and CPU sockets being assigned to the VM.
- A default display type, and remote access protocol.
- The type and version of OS.

Workload: Starting, running, pausing, and stopping the VM.

WORKLOAD

A workload is not just a program or an application. More than just the application, the workload also includes resources needed to run it, including operating system burden, compute cycles, memory, storage, and network connectivity. In the VM environment, the workload can even include the load on the **components apart from the server** itself, such as network or storage components that are not permanently attached to a server.

One common description of a workload uses the concept of a *container. A* workload is often viewed as being the **complete set of resources needed** for the task, which is independent of other elements.

In most cases, the required resources are drawn from different resource pools, and these pools can be (and often are) in different locations. In the virtualization model, these workloads are able to be spun up and torn down very quickly, and in some cases require that another workload be spun up to complete a subtask

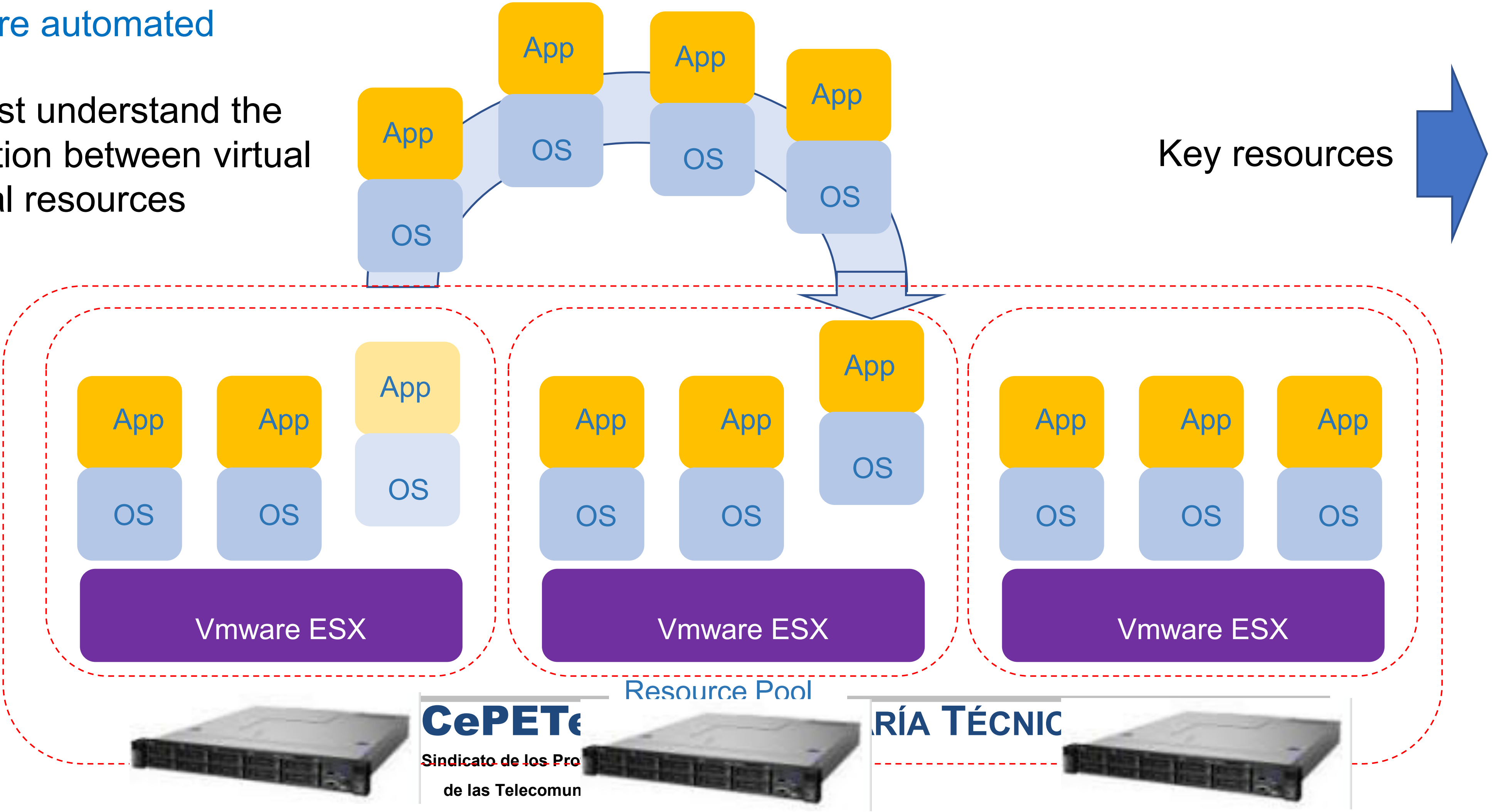
Furthermore, all this could be happening to thousands of workloads at any given time. Virtualization is what allows us to break up all the various resources into pools so that we can create very efficient workloads (such that the containers all have the resources they need, and no more), but it’s what we commonly know as cloud networking that allows us to stitch these workloads together quickly and reliably.

Managing Virtual Resources in the Hypervisor, CPU

Hypervisor manages the VM’s virtual resources, but we must ensure it is managing the resources properly. Hypervisor “thinks” it has infinite resources available. It is therefore essential that we remember to manage both **physical** and **virtual** resources.

VMs get moved from **physical** host to **physical** host even while being utilized. Managing resources under these constantly changing conditions is more than even the most diligent and talented tech team can manage and is therefore automated

We must understand the correlation between virtual and real resources



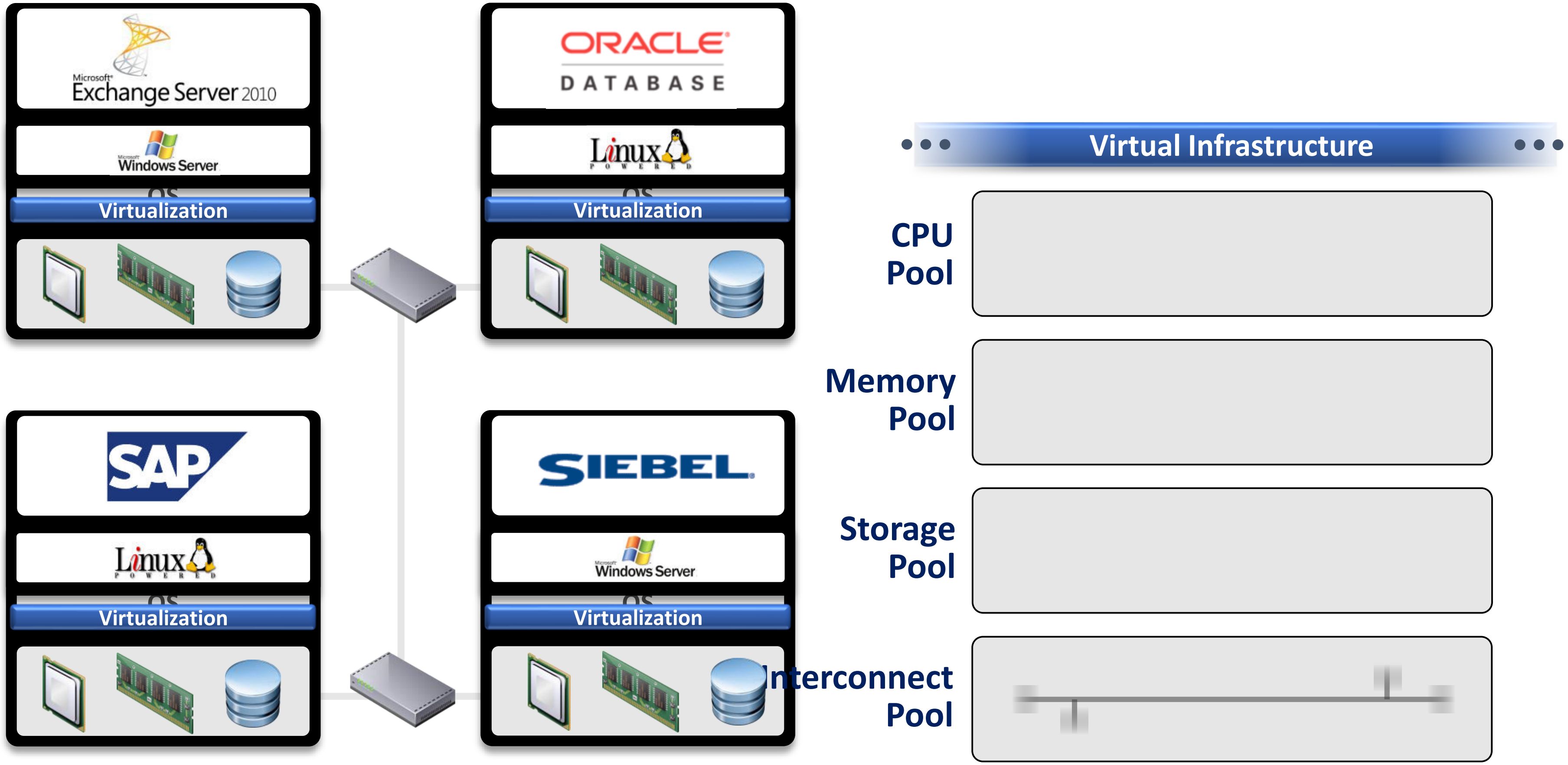
Key resources

- CPU
- Memory
- Storage
- Network
- P…?

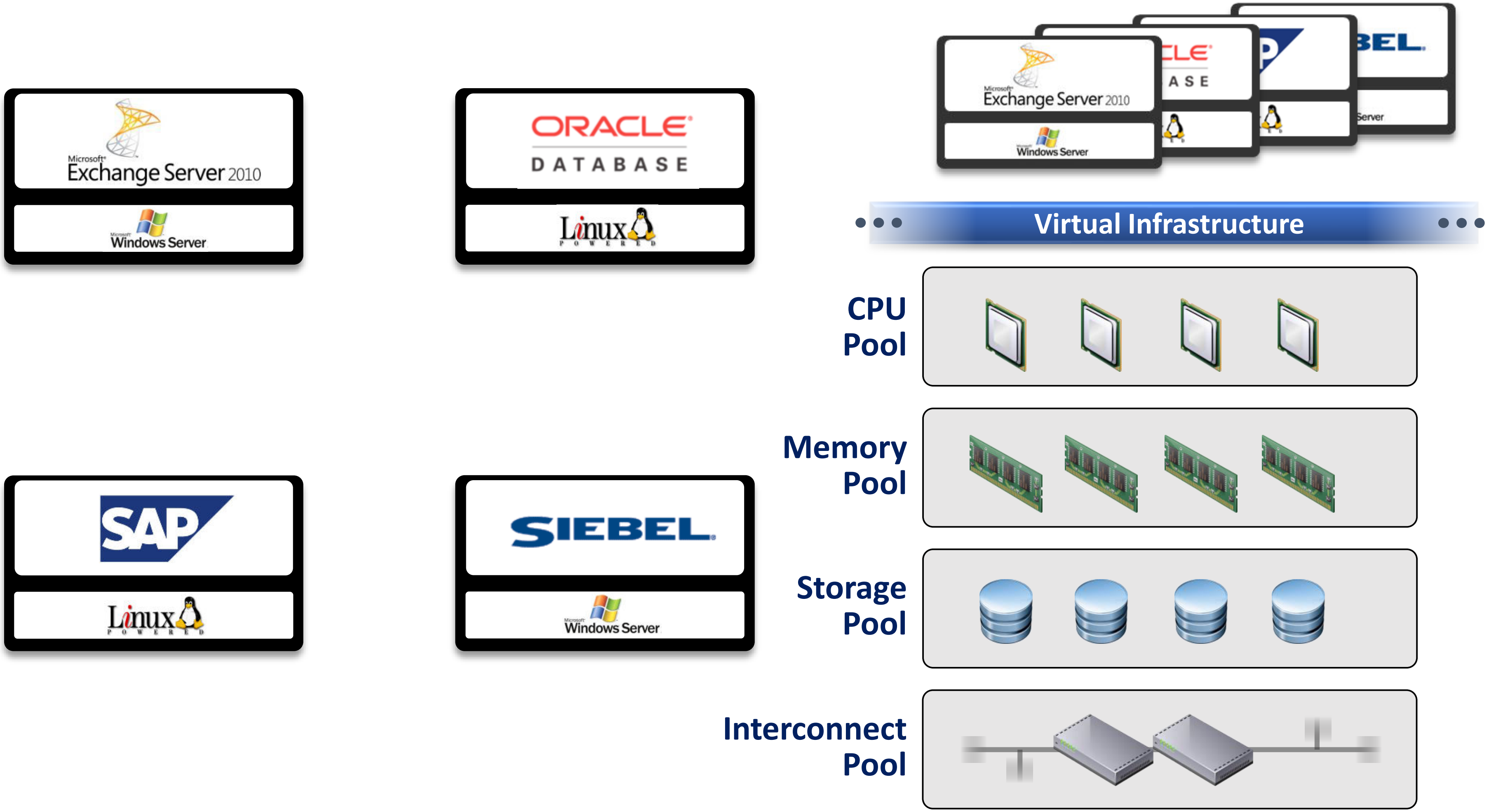
ABSTRACTION & POOLING =REDUCED COMPLEXITY

Traditional View

Virtual Infrastructure



VIRTUAL RESOURCE PROVIDERS AND CONSUMERS



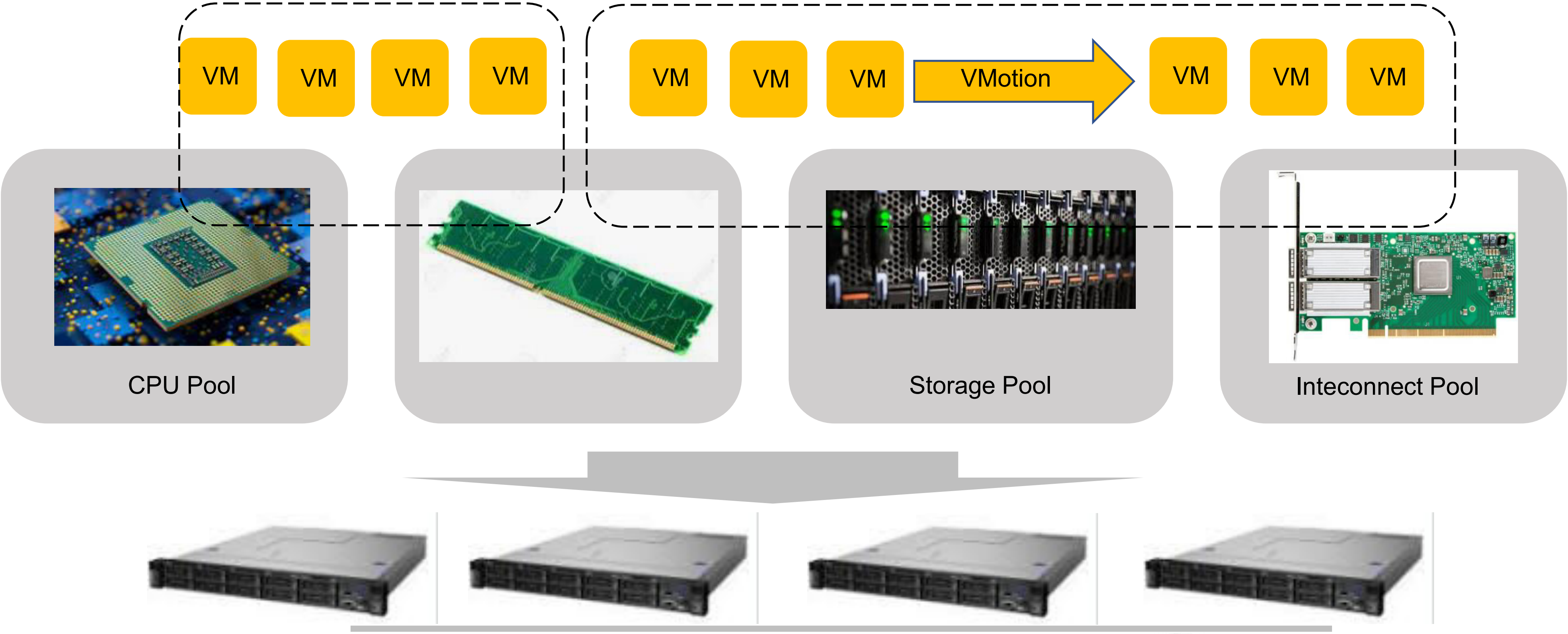
VIRTUAL RESOURCE PROVIDERS AND CONSUMERS

In traditional data centers, all required resources were supplied by a **single host.**

In virtualized data centers, there are **resource clusters**, which are groups of physical resources.

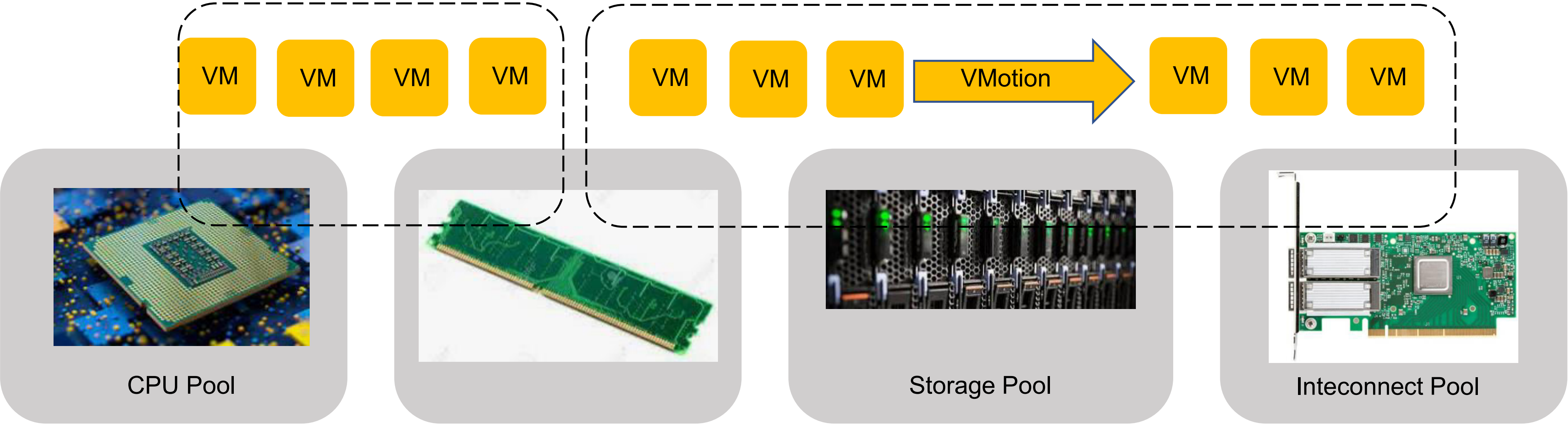
Resource clusters can include the following:

- Storage
- Memory
- Data stores (storage)
- Hosts (physical servers with VMs loaded on them)



VIRTUAL RESOURCE PROVIDERS AND CONSUMERS

It is from clusters of hosts that large virtual infrastructures typically draw their virtual resources. In most large data centers, these clusters providing resources are distributed via distributed resource scheduler (DRS). In the case of storage, it is the data stores that are grouped into clusters, and then the I/O utilization and capacity can be balanced as required using the storage distributed resource scheduler (SDRS).



Clusters are the sources for the virtual resource providers
VM are the consumers.
So many consumers and fixed resources (at least in the near term), the resource limits for VMs must be set correctly such that the total virtual consumption does not exceed the physical resources available.

HOW TO MANAGE VIRTUAL RESOURCES?

So How Do You Manage Virtual Resources?

Performance isolation must be maintained (applications must run with a high level of performance)

Applications must have predictable performance even if virtual resources are a shared resource.

Ensure active VMs do not monopolize system resources

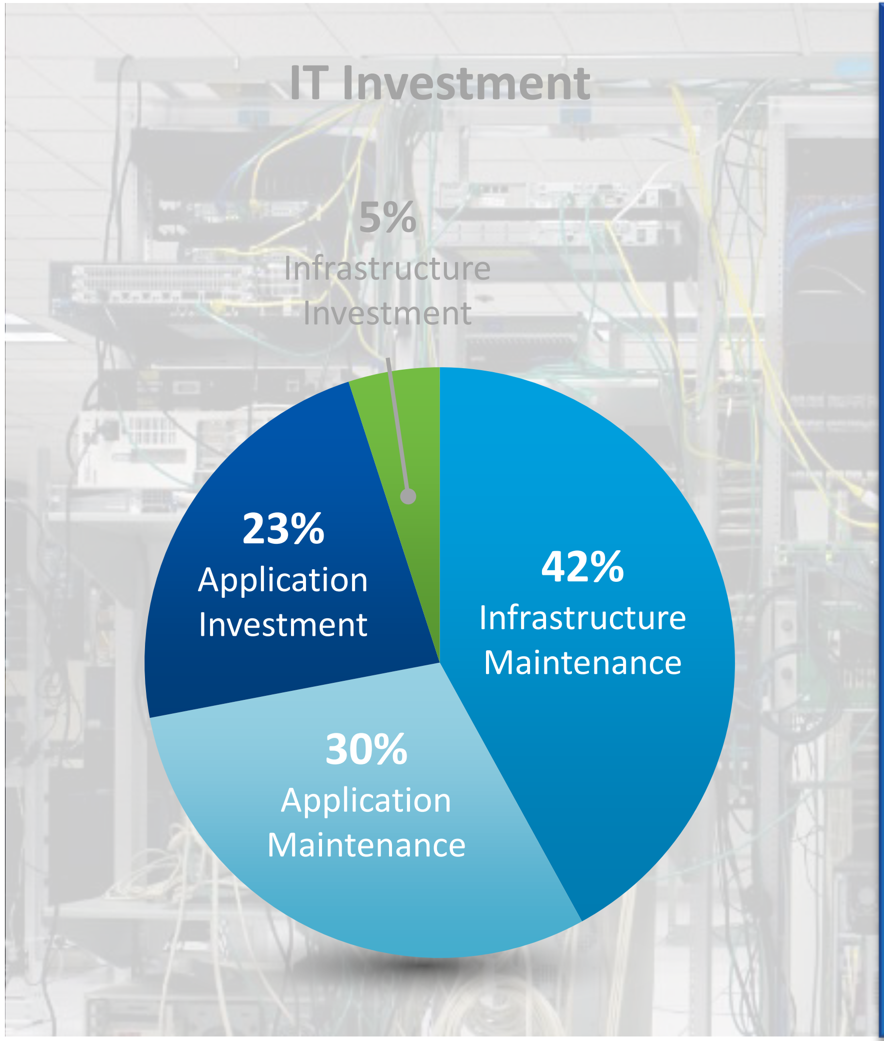
Ensure that VM density is neither too high nor too low on a physical server.

In addition, the resources of individual VMs must also be managed; this is especially true if created via a standard template. It's also important to ensure VMs that require resources have not been set too low and that those idle VMs are not set too high, which only creates waste.

Virtual resources can be managed through both administrative controls and through automated hypervisor virtual management.

Both have to be configured as preset VM resource allocations, which if not predicted carefully can create resource starvation in busy or active VMs and resource waste in less-intensive or idle VMs. However, by utilizing good management and oversight, it is possible to enable dynamic stretching and shrinking of resource pools, which allows applications to demand and consume resources on an as-needed basis without impacting performance.

IT is Traditionally Forced to Focus on Non-Value-Add Activity



Overwhelming complexity

+

Brittle infrastructure

=

< 30% of IT budgets goes to innovation and competitive advantage

Business Agility Depends on IT Agility

Source: VMware Fortune 100 Customers

Before Virtualization: The State of at IT Infrastructure

Server Sprawl

- 36M physical x86 servers by 2011— a ten-fold increase over 15 years
- $140 bn in excess server capacity - a 3-year supply

Power & Cooling

- $1 for every $1 spent on servers
- $29 bn in power and cooling industry wide

Space Crunch Costs

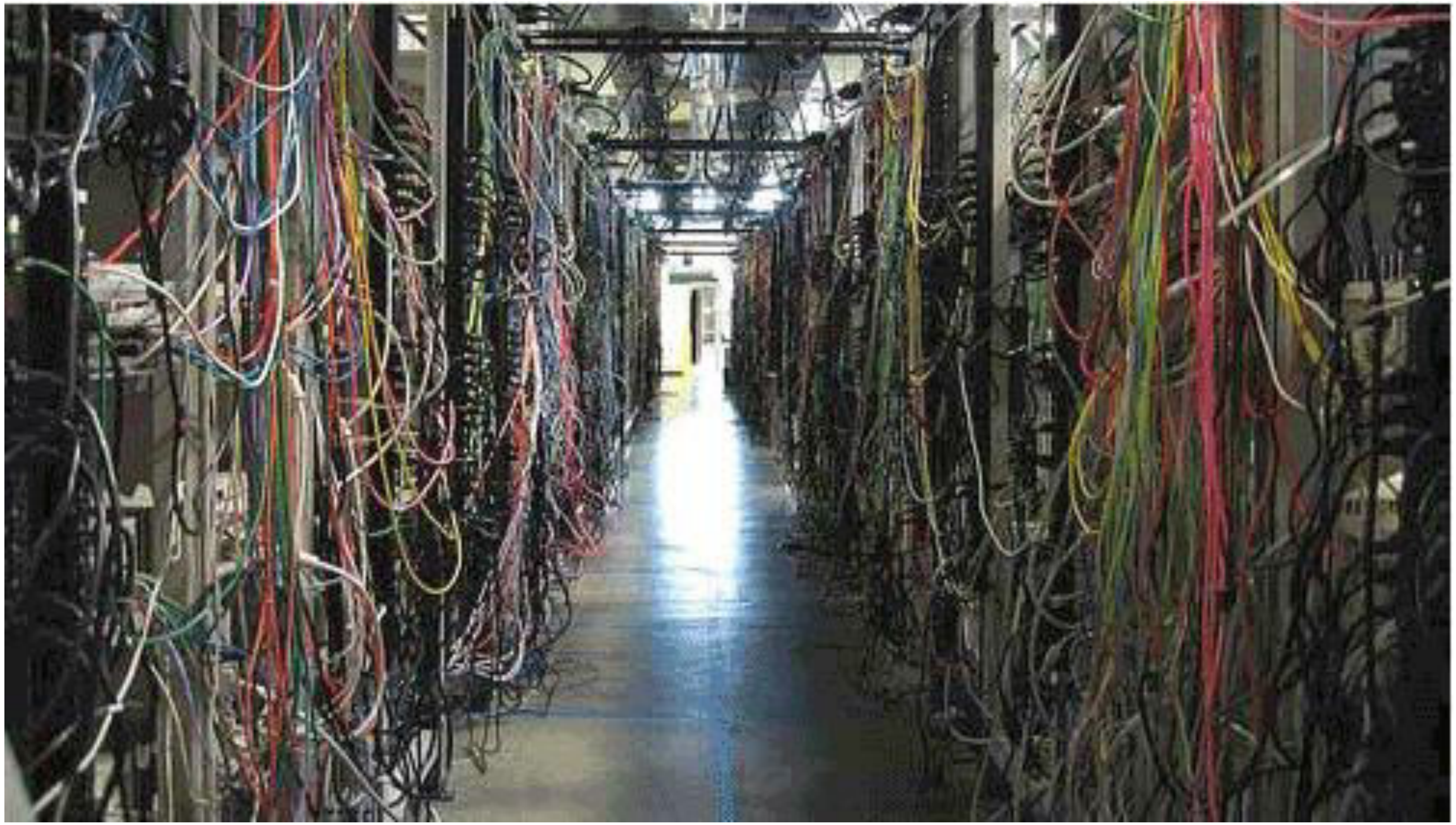
- $1,000 / sq ft
- $2,400 / server
- $40,000 / rack

Operating Costs

- $8 in maintenance for every $1 spent on new infrastructure
- 20-30 : 1 server-to-admin ratio

Source: VMware Fortune 100 Customers

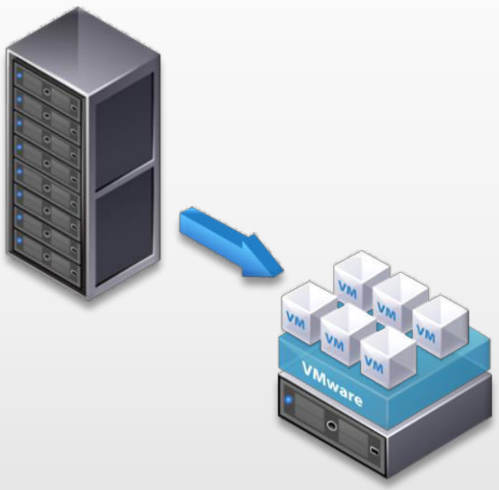
Obviously something isn't working very well…



CORE BENEFITS OF VIRTUALIZATION

1. Reduce the Complexity

to simplify operations and maintenance



2. Dramatically Lower Costs

to redirect investment into value-add opportunities

3. Enable Flexible, Agile IT Service Delivery

to meet and anticipate the needs of the business

Source: VMware Fortune 100 Customers

HOW DO I GET THOSE BENEFITS?

Consolidation - One-time event that moves existing applications onto a fewer number of servers

Containment - An ongoing effort to virtualize new applications and manage growth of existing ones

Availability – Introducing virtualization to increase application availability and data recoverability

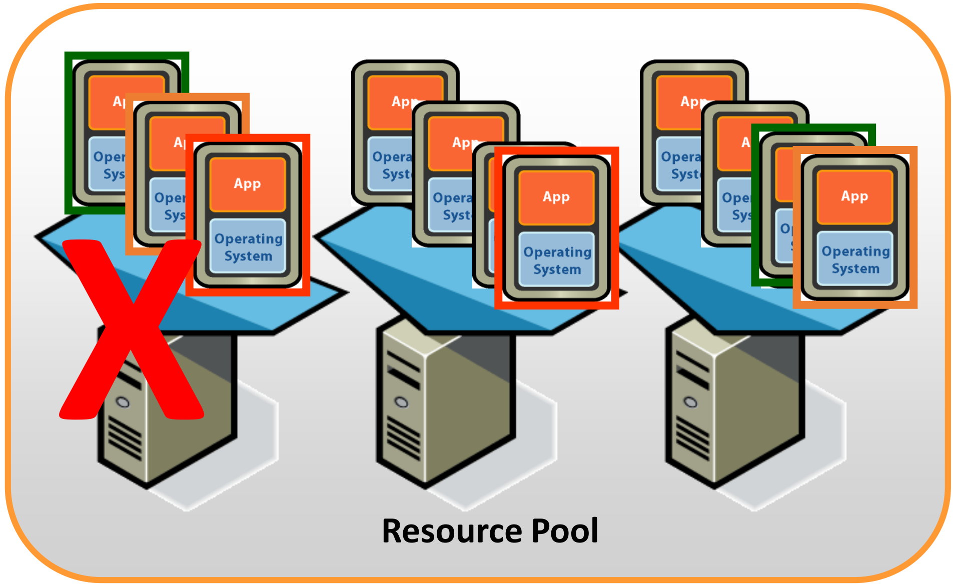
…there are many more benefits of virtualization

Source: VMware Fortune 100 Customers

HIGH AVAILABILITY

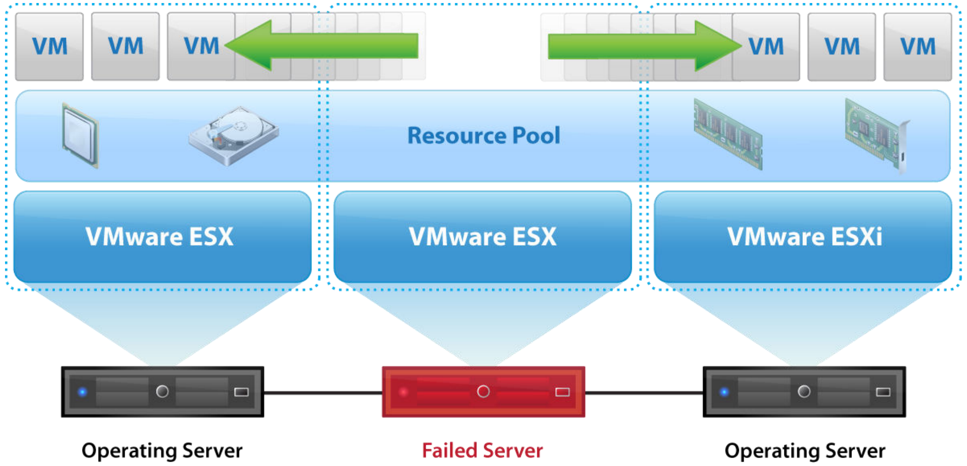
Description:

Enables the high availability of virtual machines by restarting them on a different vSphere host in the event of a failure



Benefits:

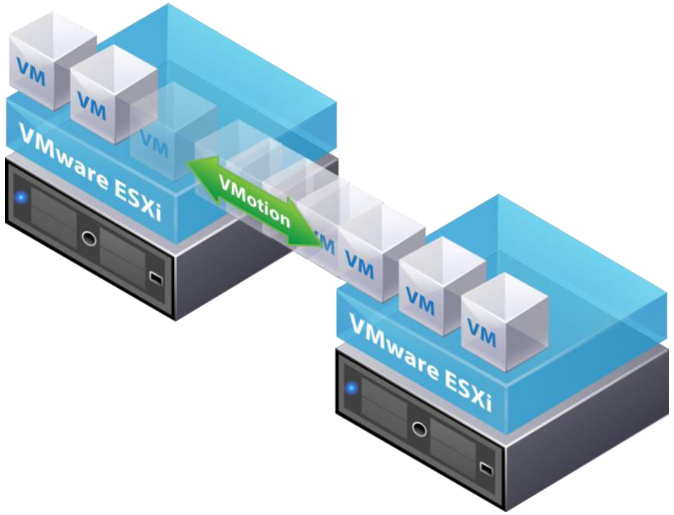
- Minimizes downtime and IT service disruption
- Reduce cost and complexity compared to traditional clustering



LIVE MACHINE MIGRATION WITH ZERO DOWNTIME

Description:

Enables the live migration of virtual machines from one host to another with continuous service availability.



Benefits:

- Revolutionary technology that is the basis for automated virtual machine movement
- Meets service level and performance goals

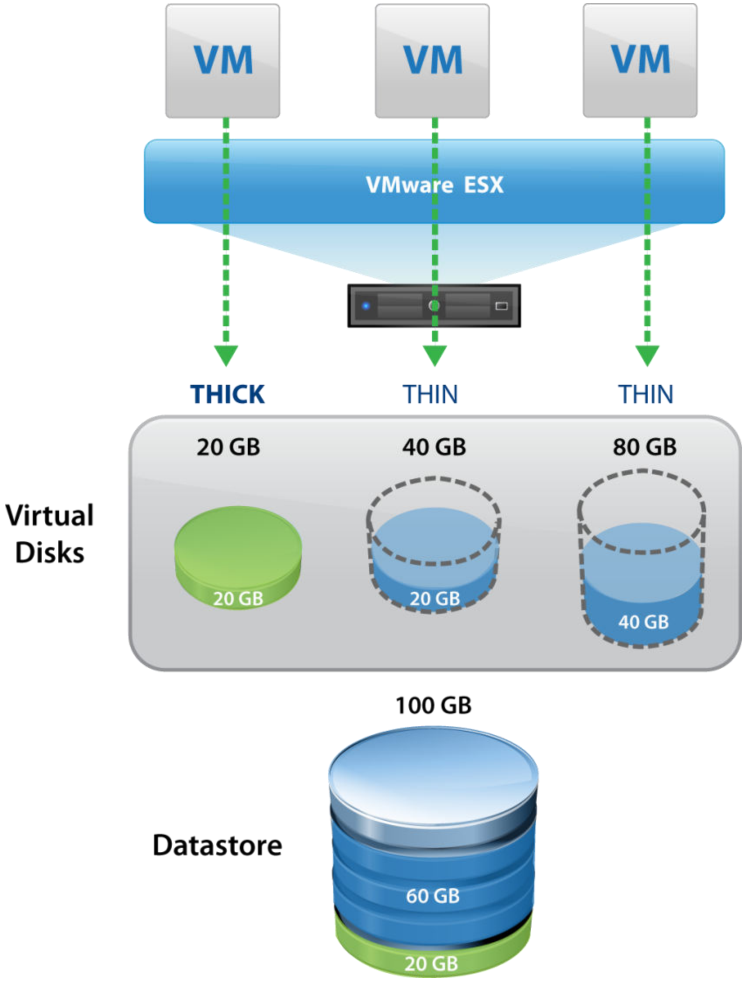
BETTER STORAGE UTILIZATION AND EFFICIENCY

Description:

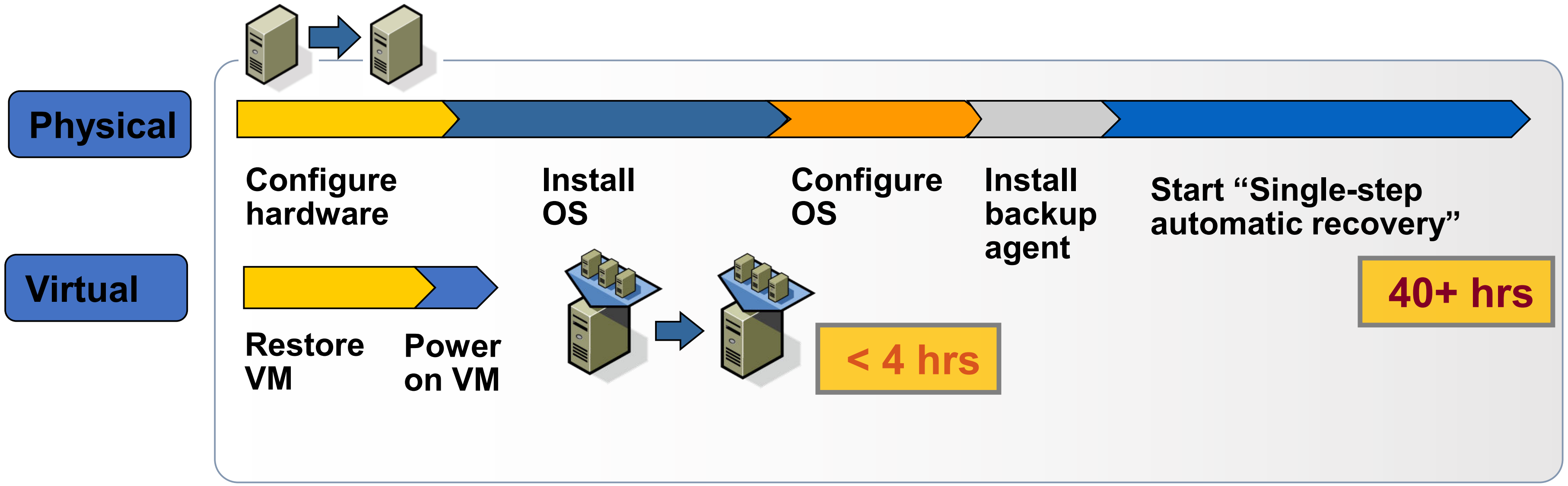
Provisioning storage only based on what is needed now and grow into requested size over time

Benefits

- Eliminate over-allocating storage
- Reduce CapEx purchases
- More granular controls over storage resources
- Easy to convert from virtual disks that were previously thick (Storage vMotion)



SAVE TIME DURING DISASTER RECOVERY



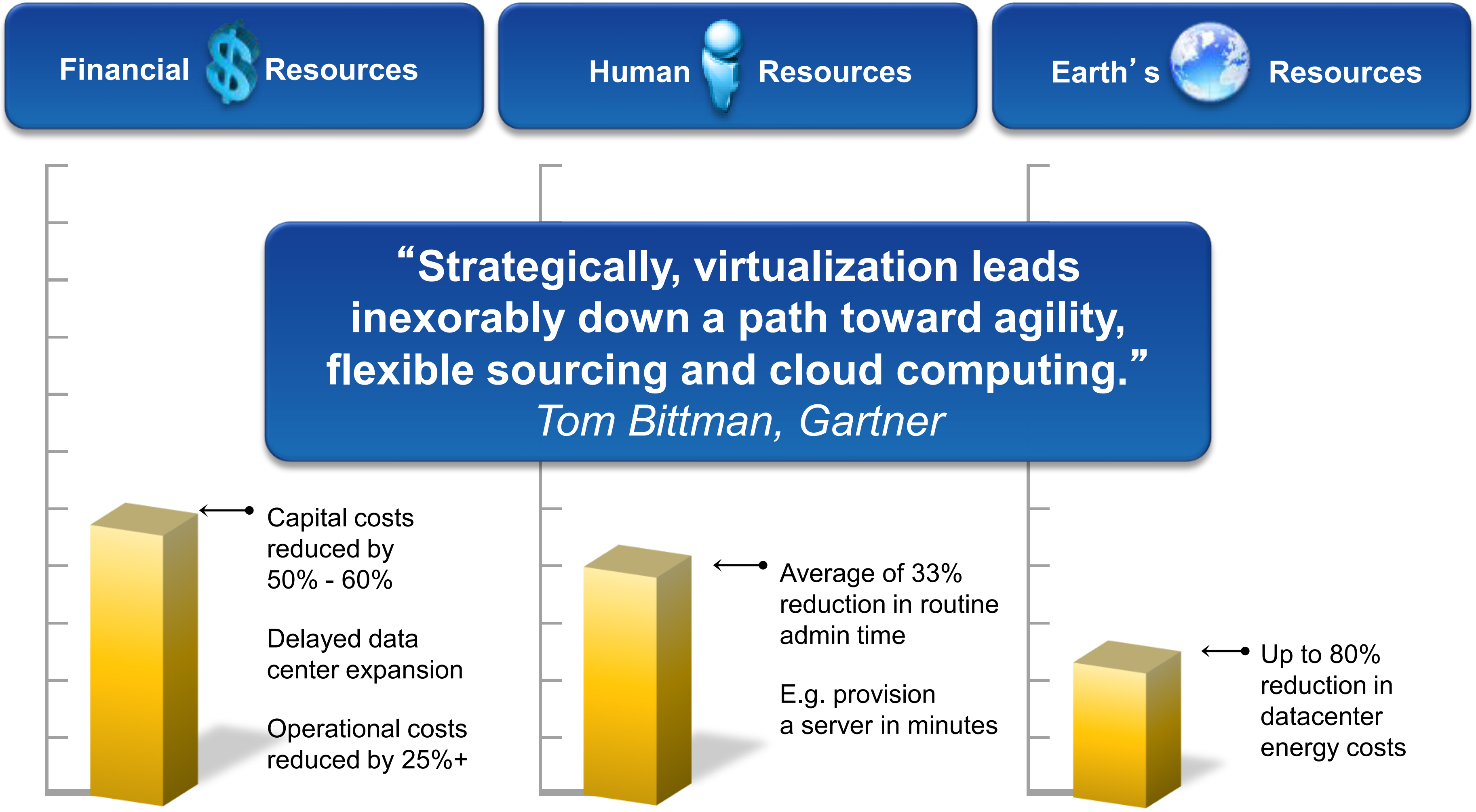
Eliminate recovery steps

- No operating system re-install or bare-metal recovery
- No time spent reconfiguring hardware

Standardize recovery process

- Consistent process independent of operating system and hardware

THE RESULTS ARE TRANSFORMATIONAL



Financial $ Resources

Human Resources

Earth's Resources

> **"Strategically, virtualization leads inexorably down a path toward agility, flexible sourcing and cloud computing."**
> *Tom Bittman, Gartner*

Capital costs reduced by 50% - 60%

Delayed data center expansion

Operational costs reduced by 25%+

Average of 33% reduction in routine admin time

E.g. provision a server in minutes

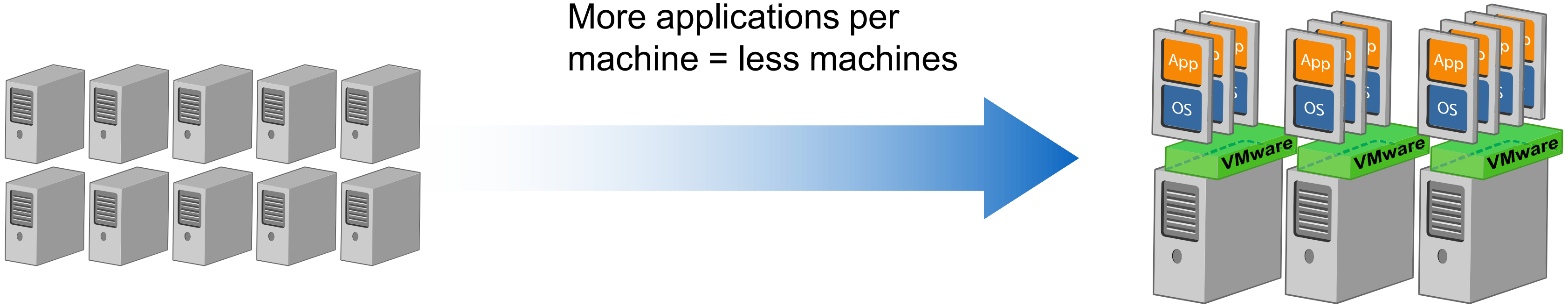
Up to 80% reduction in datacenter energy costs

Source: Gartner; 29 July 2010 *Q&A: Six Misconceptions About Server Virtualization*, Tom Bittman.

The CapEx Story: Make better use of existing infrastructure

Before Virtualization

After Virtualization



More applications per machine = less machines

Servers	10
Utilization	8%
Annual cost per server	$4,000
Total Cost	**$40,000**

Servers	3
Utilization	80%
Annual cost per server	$4,000
Total Cost	**$12,000**

$28,000 in cost avoidance

Source: IT Business Edge, “The Business Value of Server Virtualization” – cost for average a 2 x CPU server in three-year amortized hardware purchase, and annual support and maintenance contract costs 9/07

INTRODUCTION TO SDN

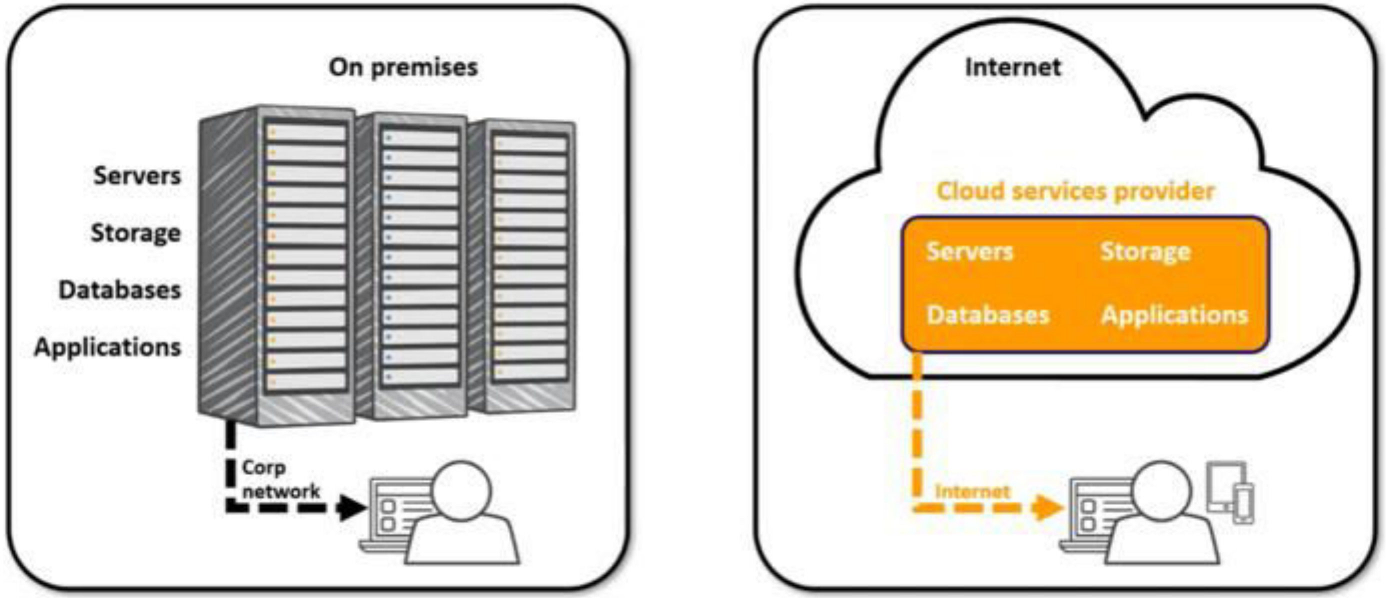
AWS CLOUD SUMMARY

MODULES

1- Introduction to the AWS Cloud
2- Getting started with the cloud
3- Building in the cloud
4- Secure your cloud applications
5- Support your cloud applications
6- Architecture

WHAT IS THE AWS CLOUD?

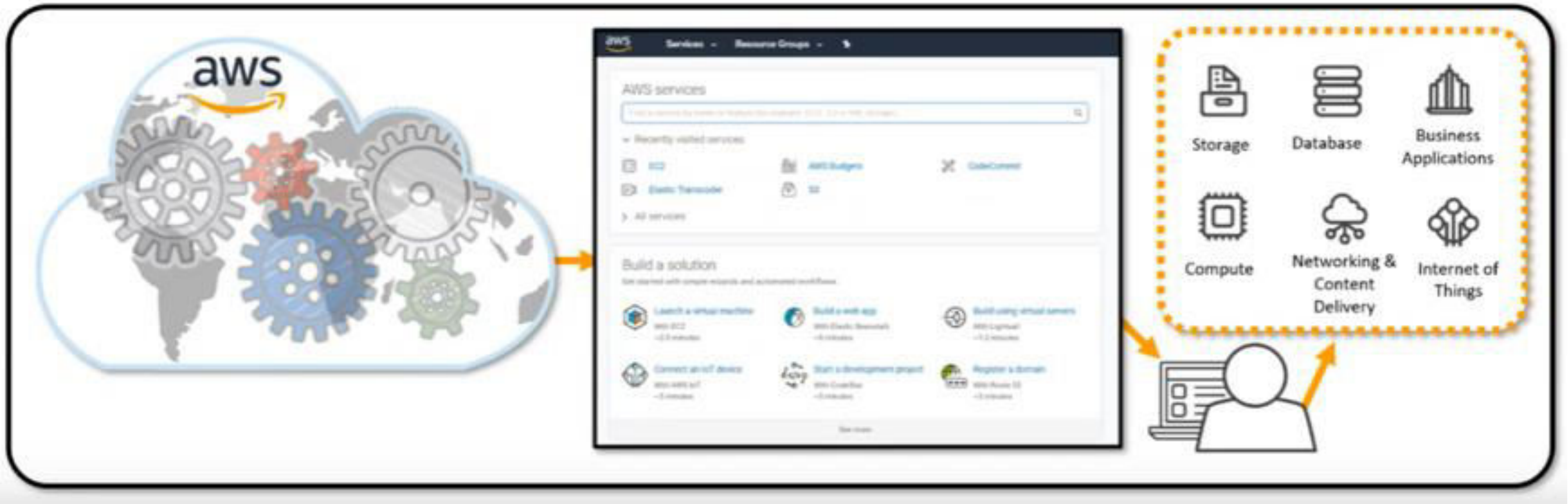
WHAT IS THE CLOUD?



Fuente: AWS

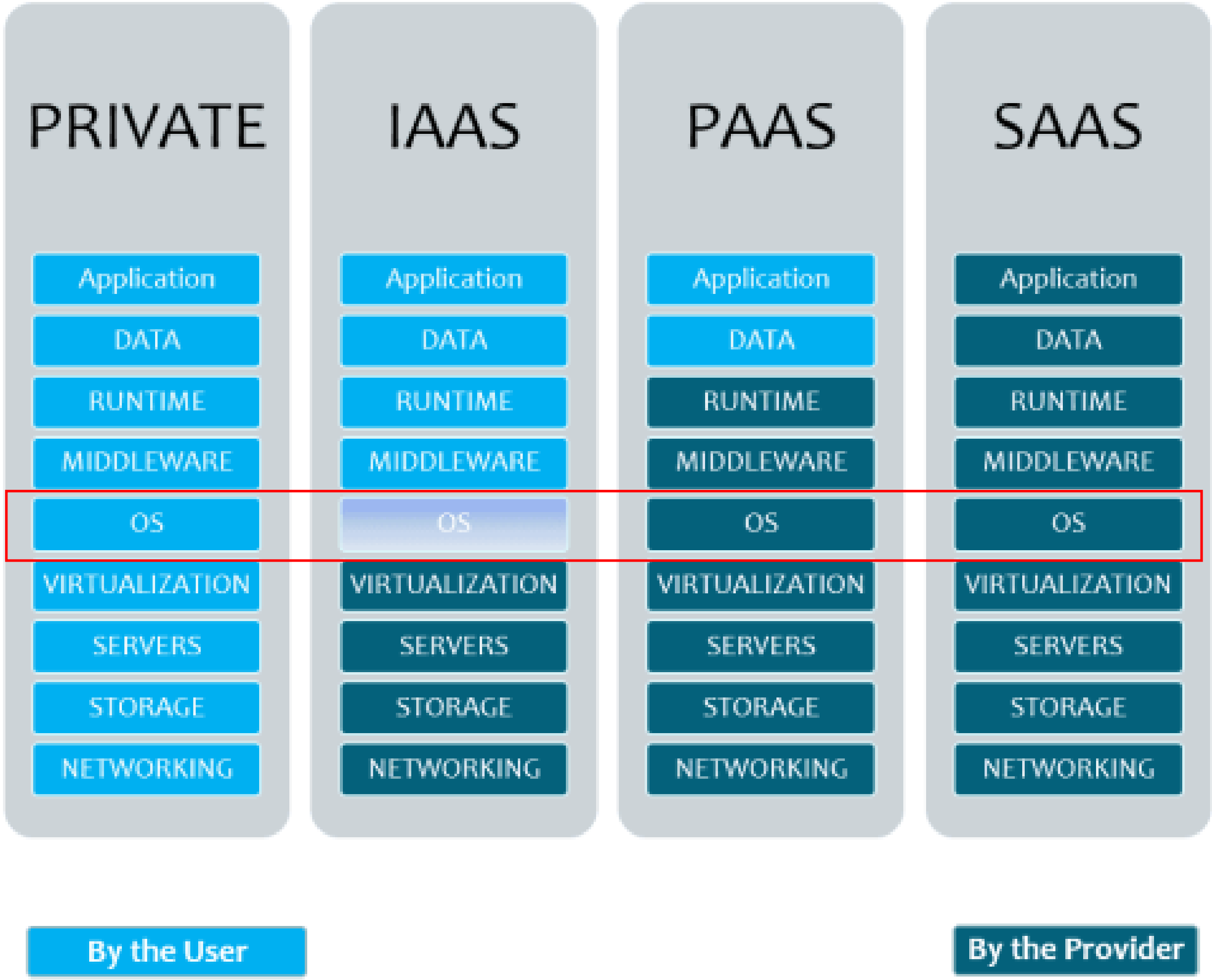
HOW DOES IT WORK?

- **AWS owns and mantains the network-connected hardware**
- **The Customers provide and use what they need**



VIRTUALIZATION- DIFFERENT TYPE OR MODELS OF CLOUD

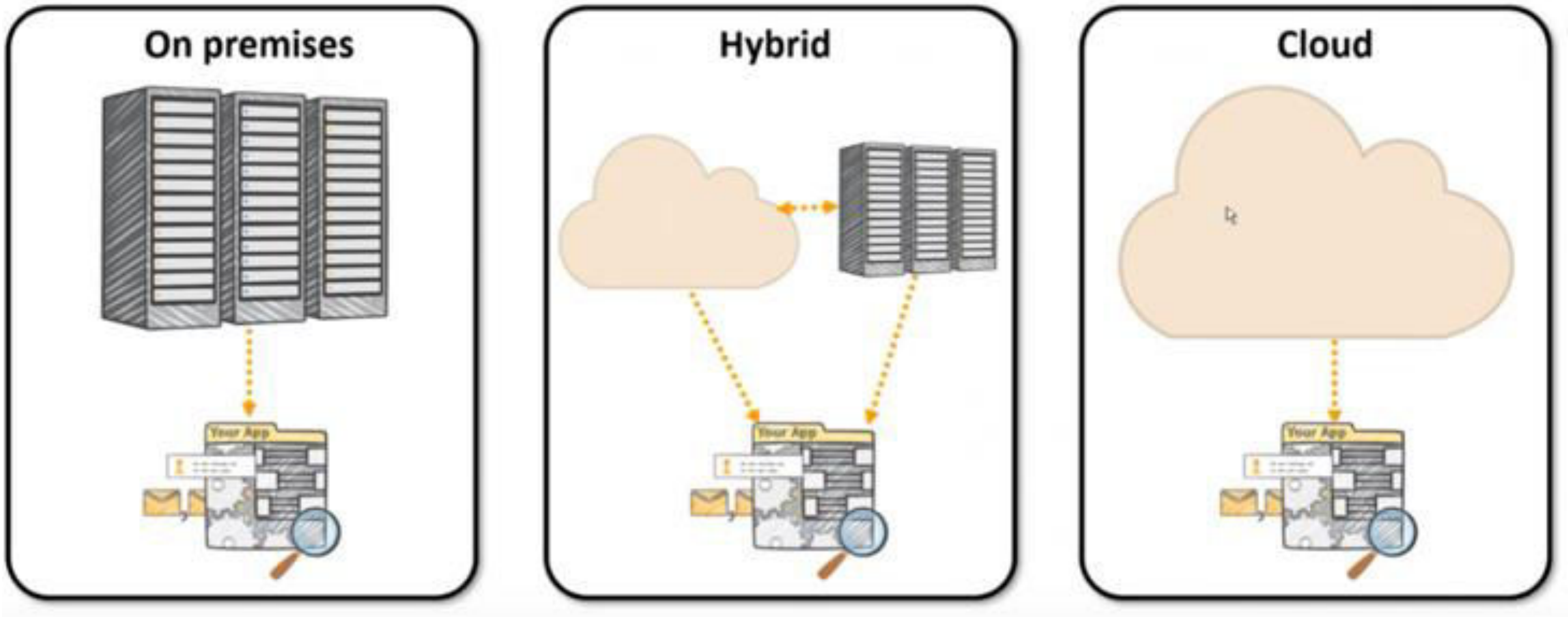
Different type of Cloud in comparison with the classical model



PRIVATE	IAAS	PAAS	SAAS
Application	Application	Application	Application
DATA	DATA	DATA	DATA
RUNTIME	RUNTIME	RUNTIME	RUNTIME
MIDDLEWARE	MIDDLEWARE	MIDDLEWARE	MIDDLEWARE
OS	OS	OS	OS
VIRTUALIZATION	VIRTUALIZATION	VIRTUALIZATION	VIRTUALIZATION
SERVERS	SERVERS	SERVERS	SERVERS
STORAGE	STORAGE	STORAGE	STORAGE
NETWORKING	NETWORKING	NETWORKING	NETWORKING

Openstack can offer one infrastructure for cloud public and private

CLOUD DEPLOYMENT MODELS



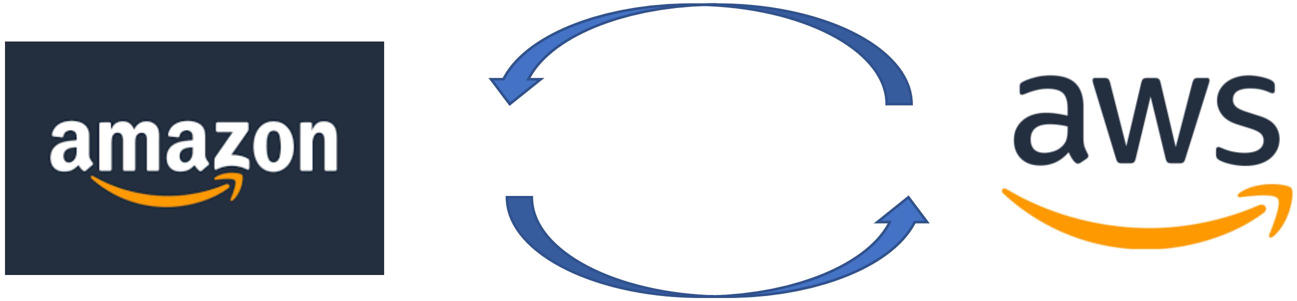
Fuente: AWS

WHAT ARE THE BENEFITS OF AWS CLOUD?

BENEFITS- SAVE COST

Data center investment based upon forecast

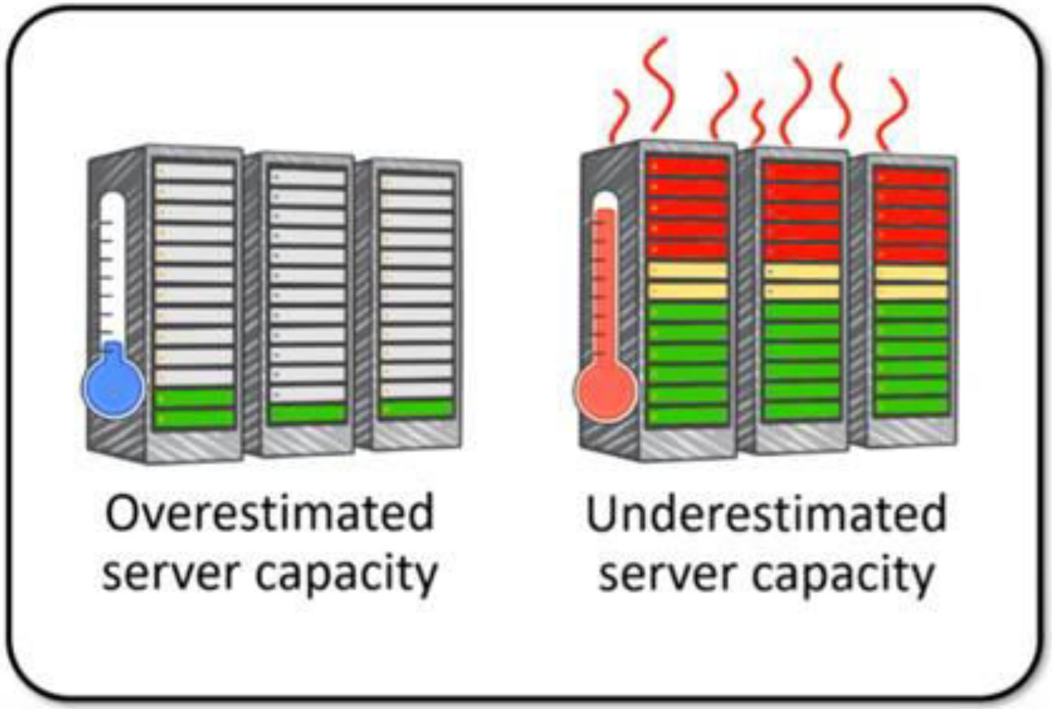
Pay only for the amount you consume



At the time, Amazon’s business was growing at the rate of a “hockey stick” graph—doubling every six to nine months. As a result, growth had to stay ahead of demand for its computing services, which served its retail ordering, stock, and warehouse management systems, as well as internal IT systems. As a result, Amazon’s IT department was forced to order large quantities of storage, network, and computing resources in advance, but faced the dilemma of having that equipment sit idle until the demand caught up with those resources. Amazon Web Services (AWS) was invented as a way to commercialize this unused resource pool so that it would be utilized at a rate closer to 100%. When internal resources needed more resources, AWS would simply push off retail users, and when it was not, retail compute users could use up the unused resources. Some call this elastic computing services, but Thomas D. Nadeau & Ken Gray call it *hyper virtualization*.

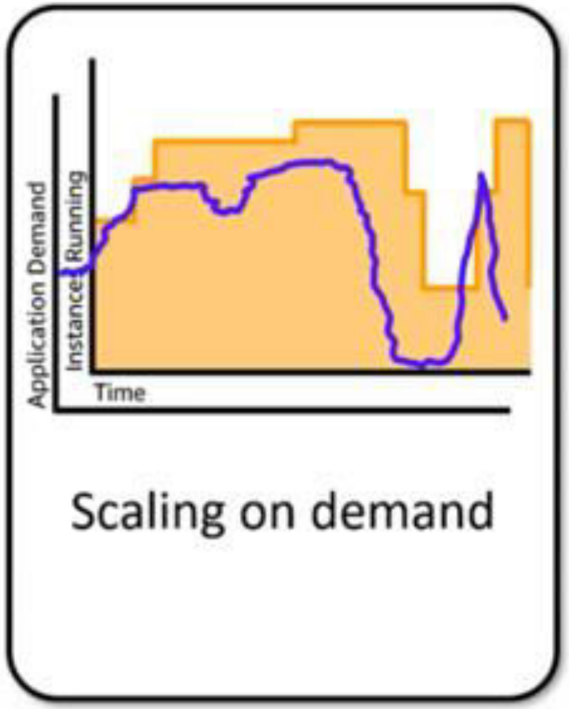
* SDN Software Defined Networks By Thomas D. Nadeau & Ken Gray

BENEFITS- STOP GUESSING CAPACITY



Overestimated server capacity

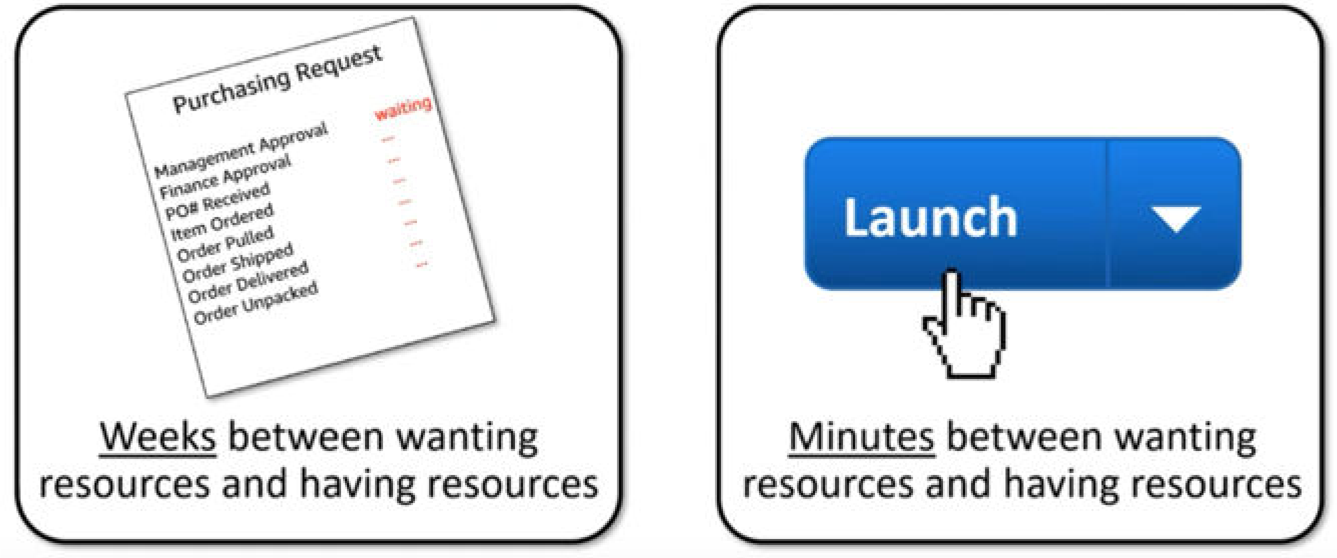
Underestimated server capacity



Scaling on demand

Fuente: AWS

BENEFITS- INCREASE SPEED AND AGILITY

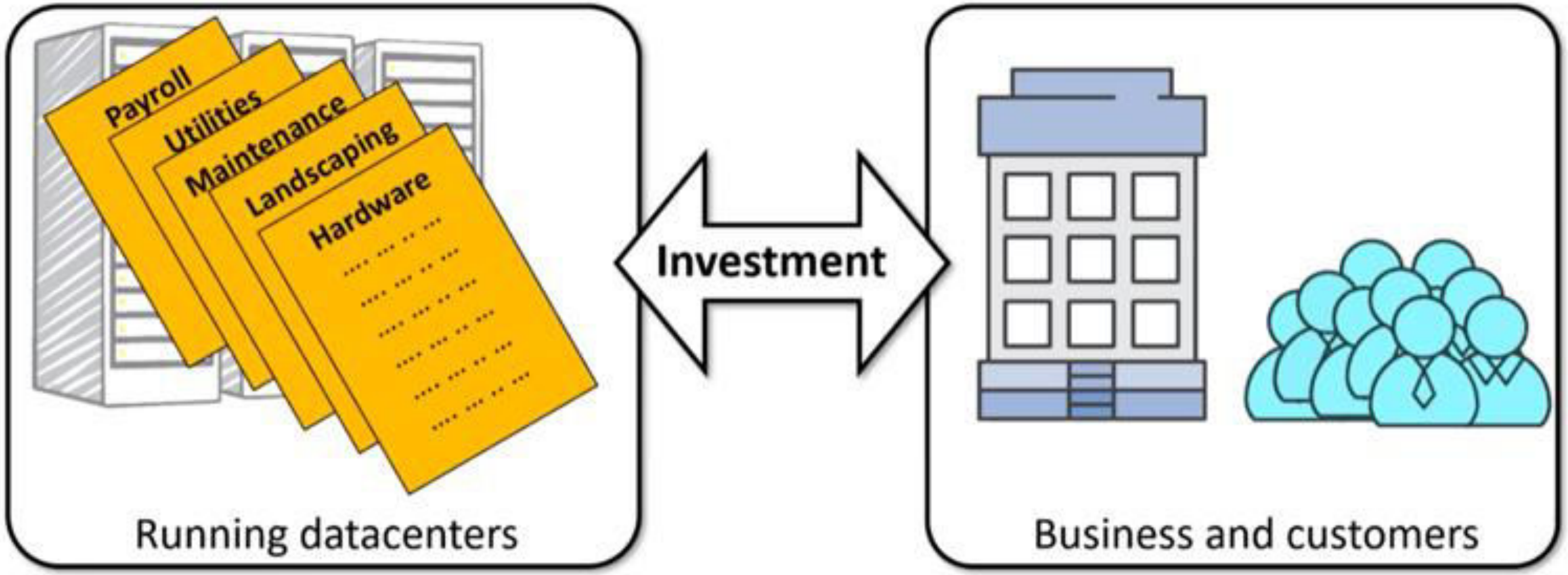


Weeks between wanting resources and having resources

Minutes between wanting resources and having resources

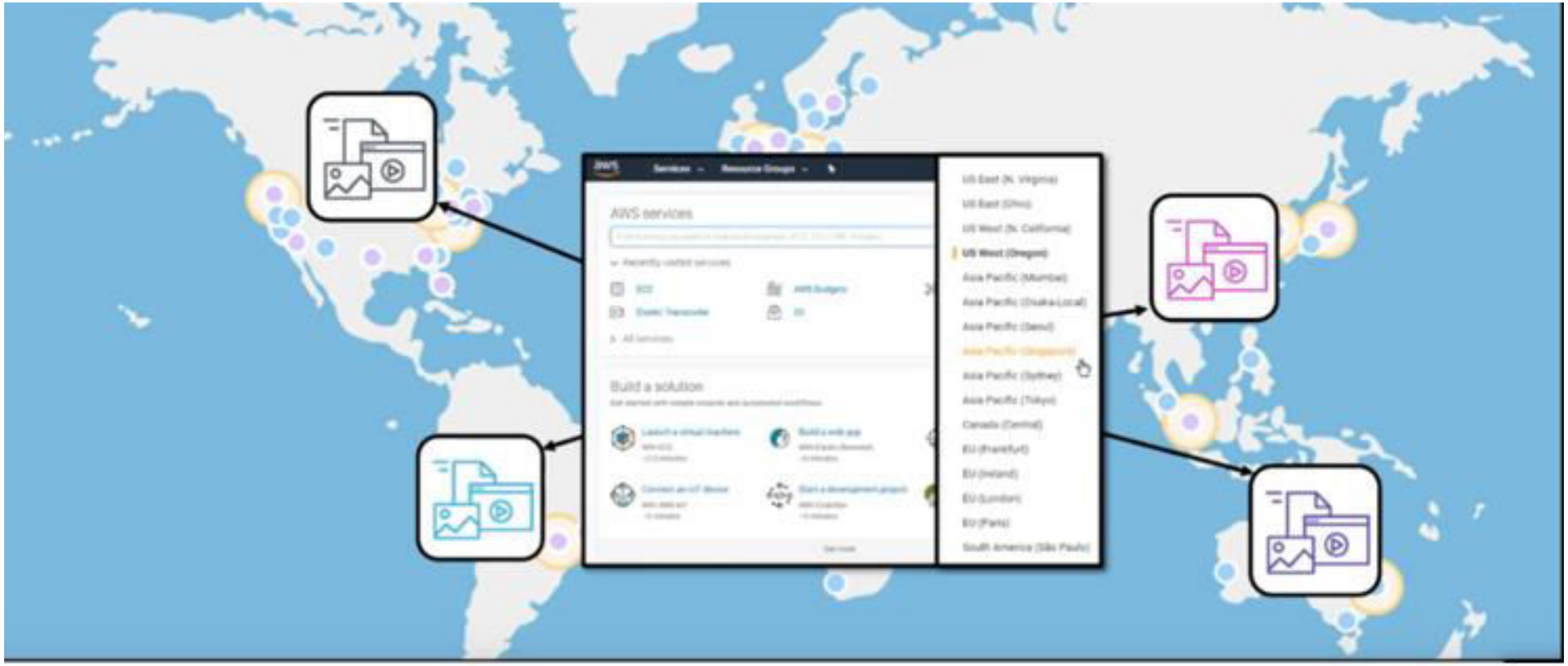
Fuente: AWS

BENEFITS- STOP SPENDING MONEY ON RUNNIG AND MANTAINING DATACENTERS



Fuente: AWS

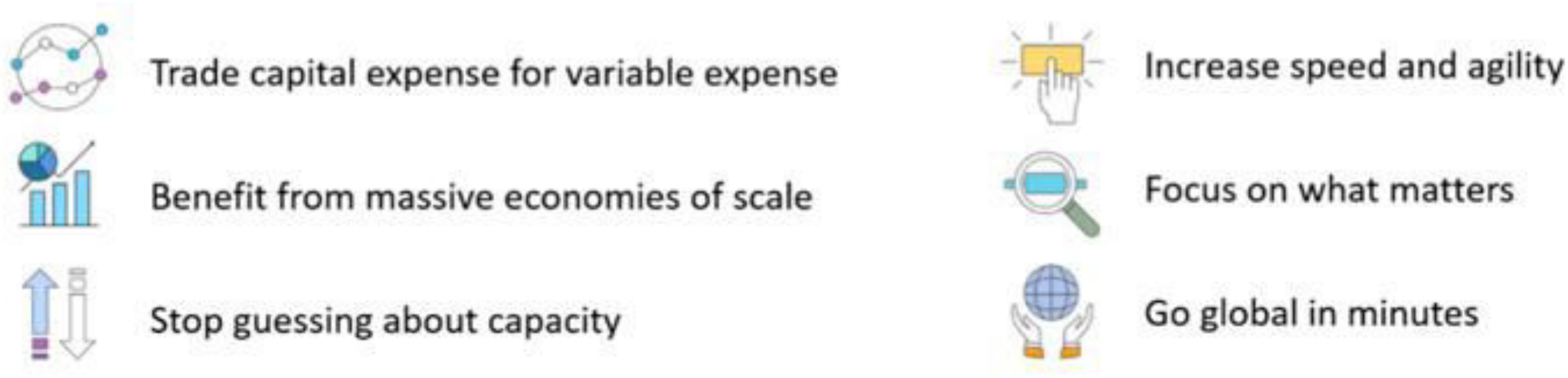
BENEFITS- GO GLOBAL IN MINUTES



However……..

Think about services which can be critical to location……….

BENEFITS- SIX ADAVNTAGES OF CLOUD COMPUTING



Trade capital expense for variable expense

Benefit from massive economies of scale

Stop guessing about capacity

Increase speed and agility

Focus on what matters

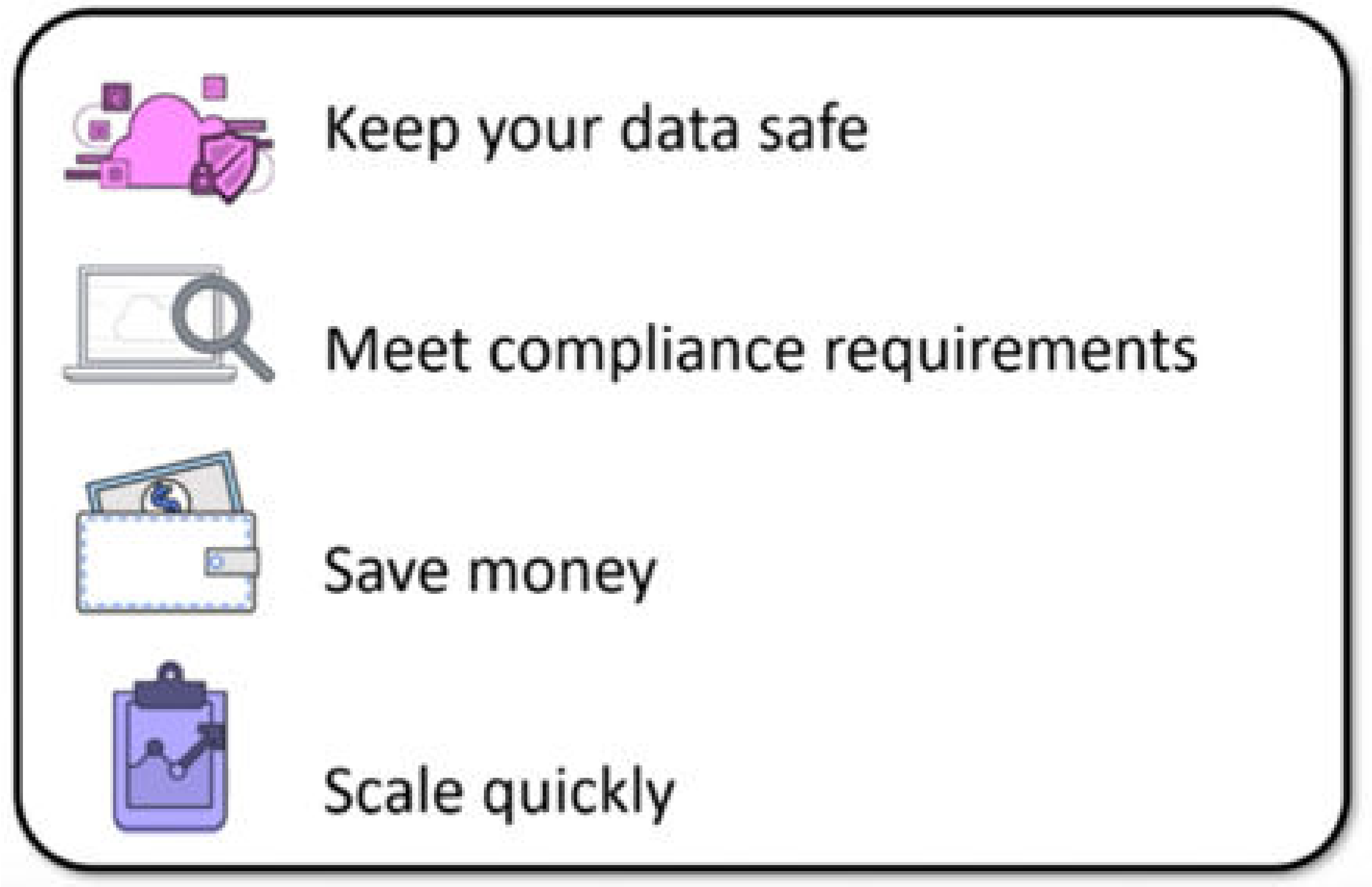
Go global in minutes

https://www.youtube.com/watch?v=yMJ75k9X5_8
The Six Main Benefits of Cloud Computing with Amazon Web Services video 7 minutes

https://docs.aws.amazon.com/whitepapers/latest/aws-overview/six-advantages-of-cloud-computing.html 104 slides
The Six Main Benefits of Cloud Computing with Amazon Web Services wp

Fuente: AWS

AWS SECURITY



Keep your data safe

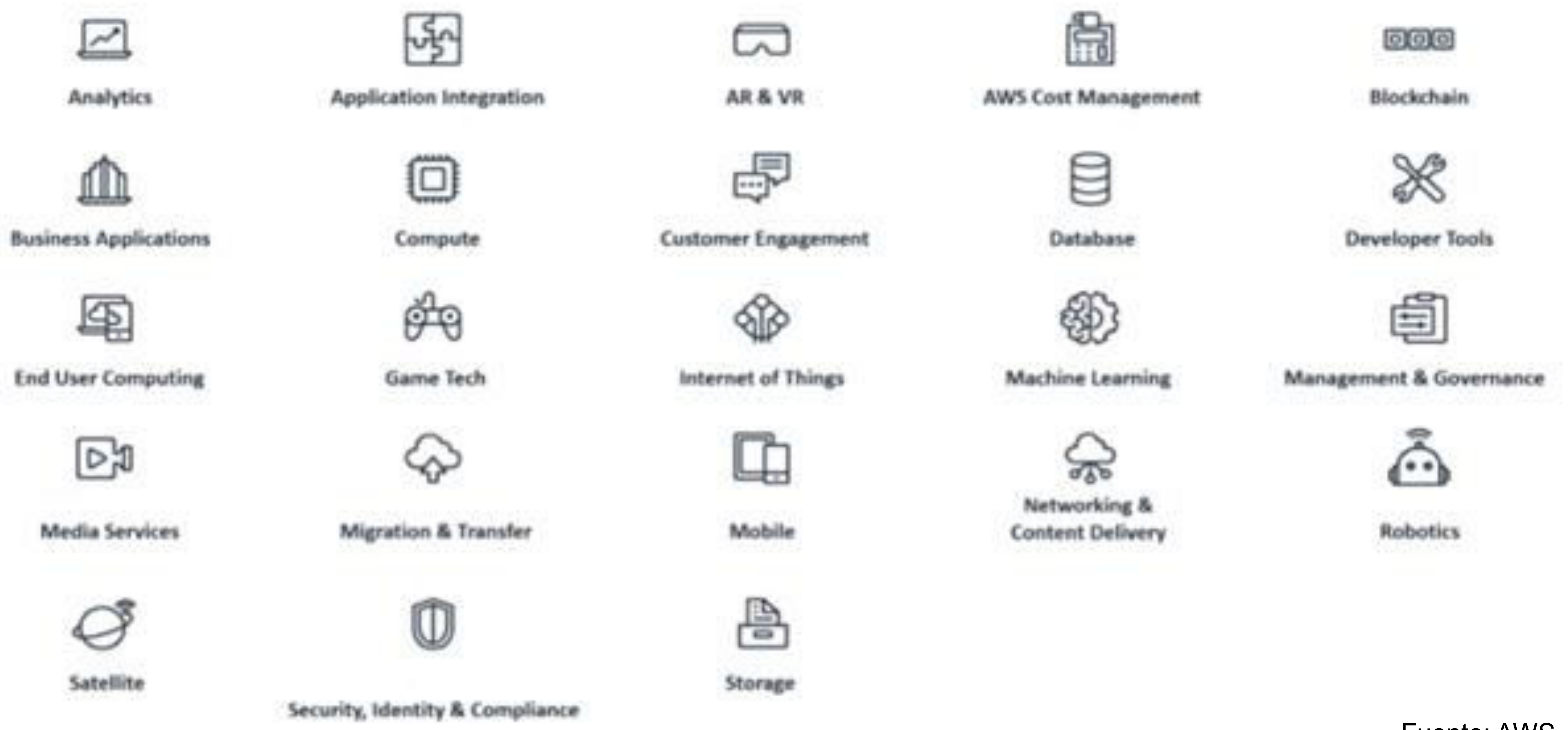
Meet compliance requirements

Save money

Scale quickly

Fuente: AWS

AWS SERVICE CATEGORIES



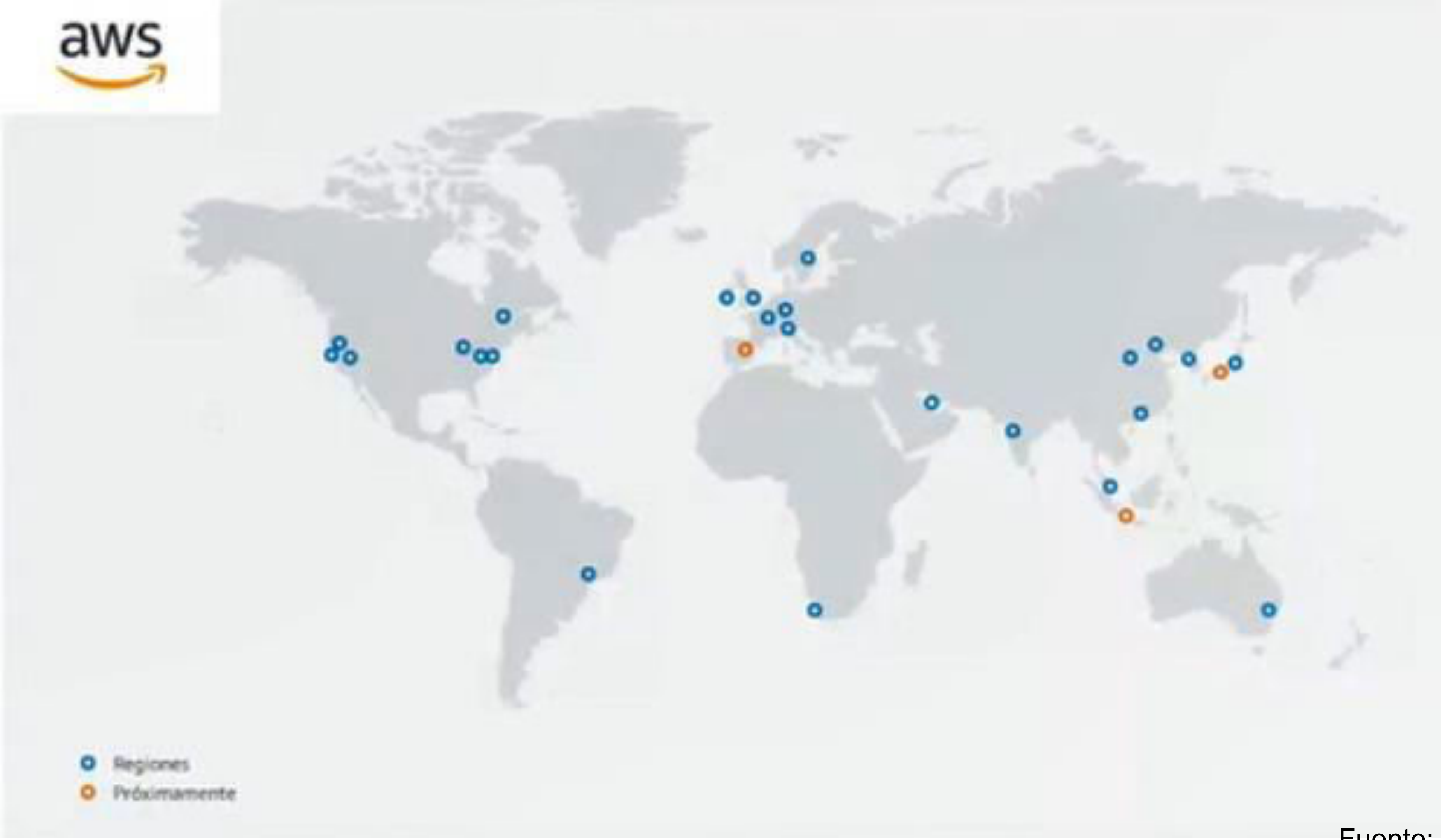
- Analytics
- Application Integration
- AR & VR
- AWS Cost Management
- Blockchain
- Business Applications
- Compute
- Customer Engagement
- Database
- Developer Tools
- End User Computing
- Game Tech
- Internet of Things
- Machine Learning
- Management & Governance
- Media Services
- Migration & Transfer
- Mobile
- Networking & Content Delivery
- Robotics
- Satellite
- Security, Identity & Compliance
- Storage

Fuente: AWS

AWS GLOBAL INFRAESTRUCTURE

REGIONS

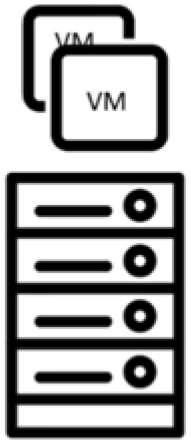
Regions containing several avalibiliy zones



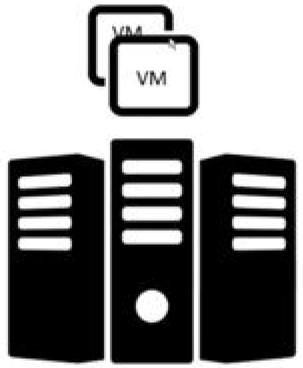
Fuente: AWS

AWS GLOBAL INFRAESTRUCTURE

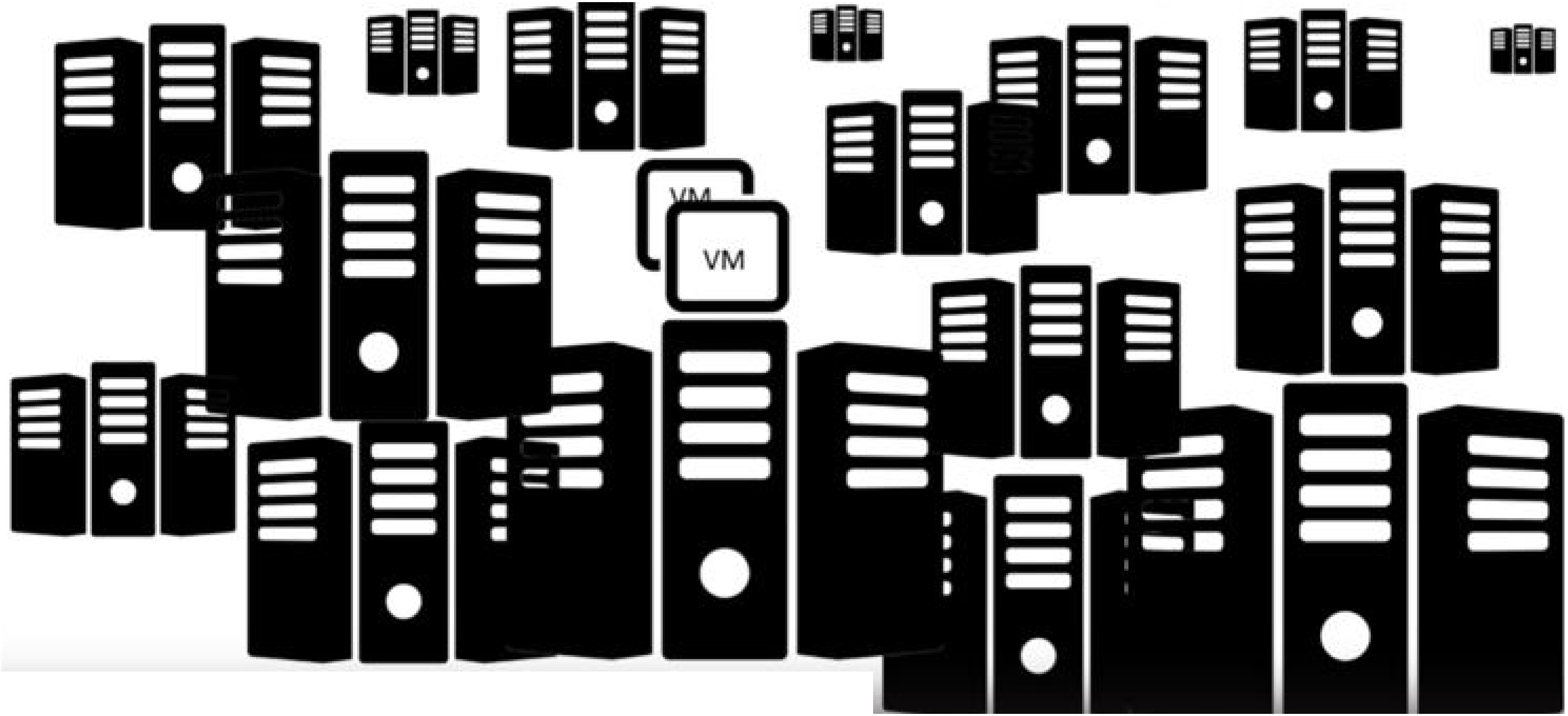
AWS GLOBAL INFRAESTRUCTURE



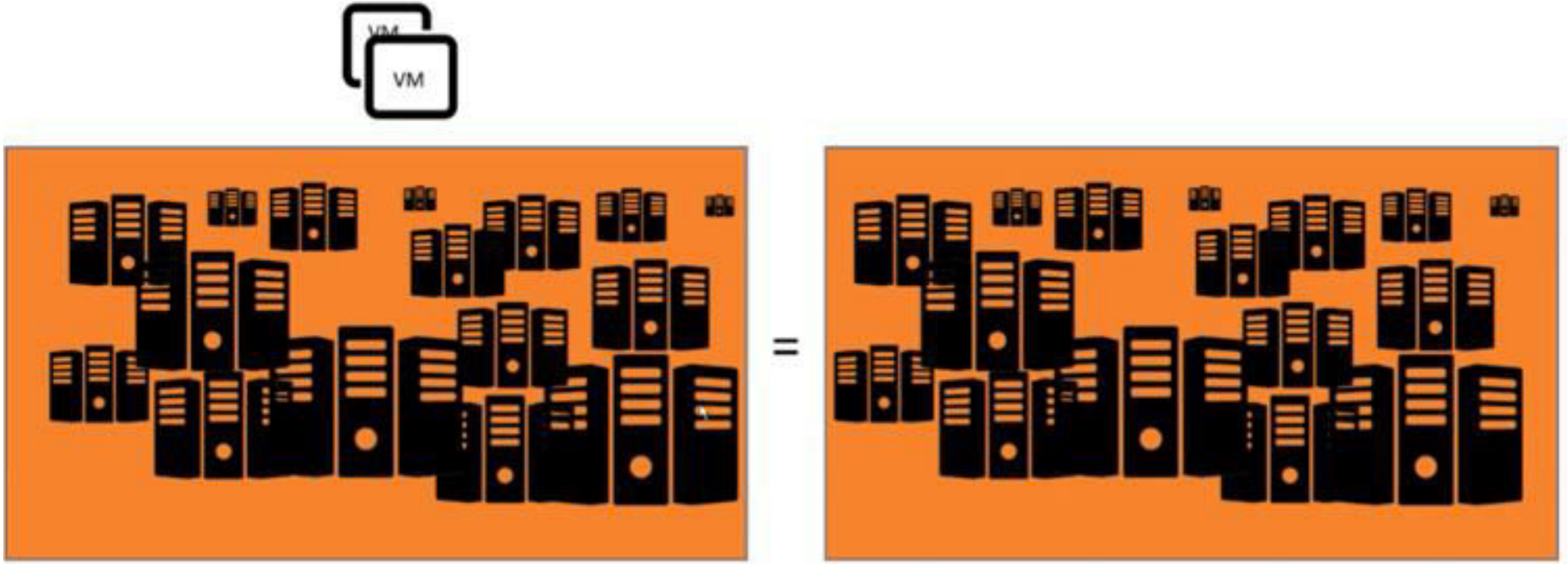
AWS GLOBAL INFRAESTRUCTURE



AWS GLOBAL INFRAESTRUCTURE

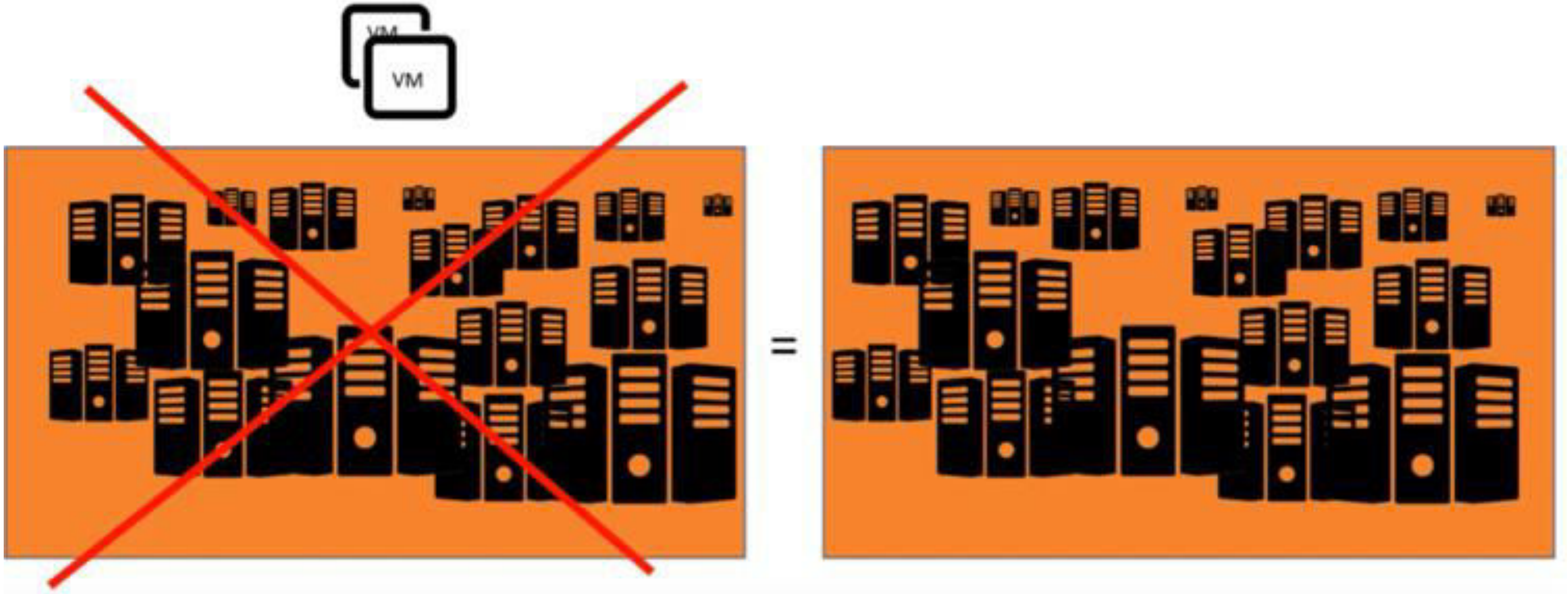


AWS GLOBAL INFRAESTRUCTURE



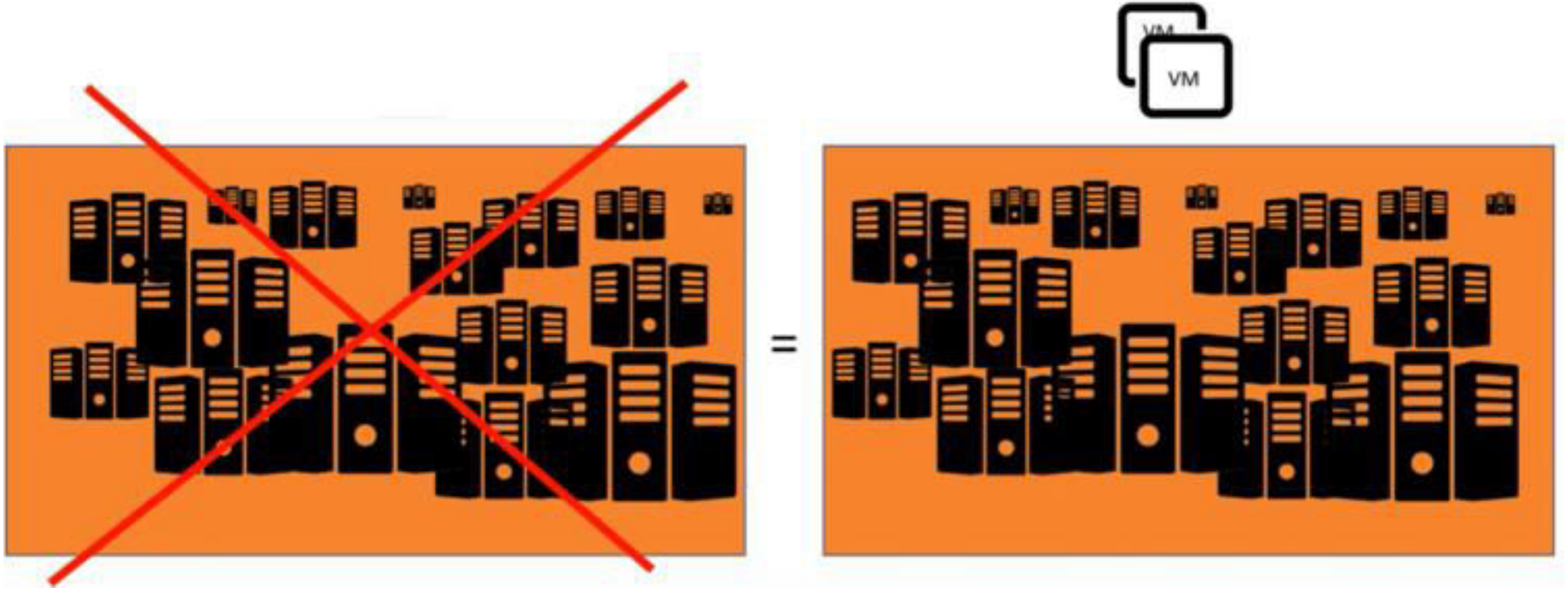
DATA CENTER clusters

AWS GLOBAL INFRAESTRUCTURE



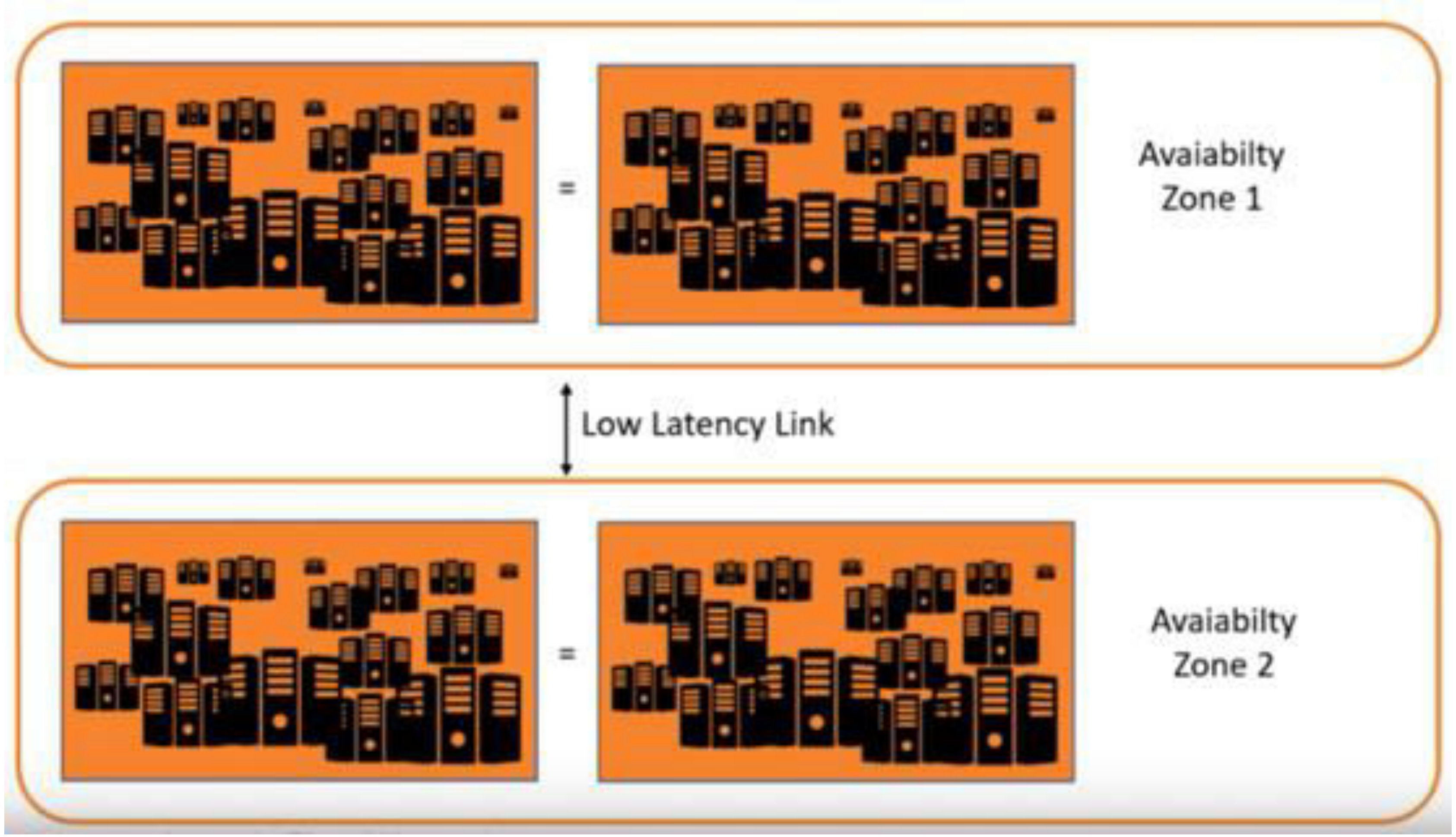
DATA CENTER clusters

AWS GLOBAL INFRAESTRUCTURE



DATA CENTER clusters

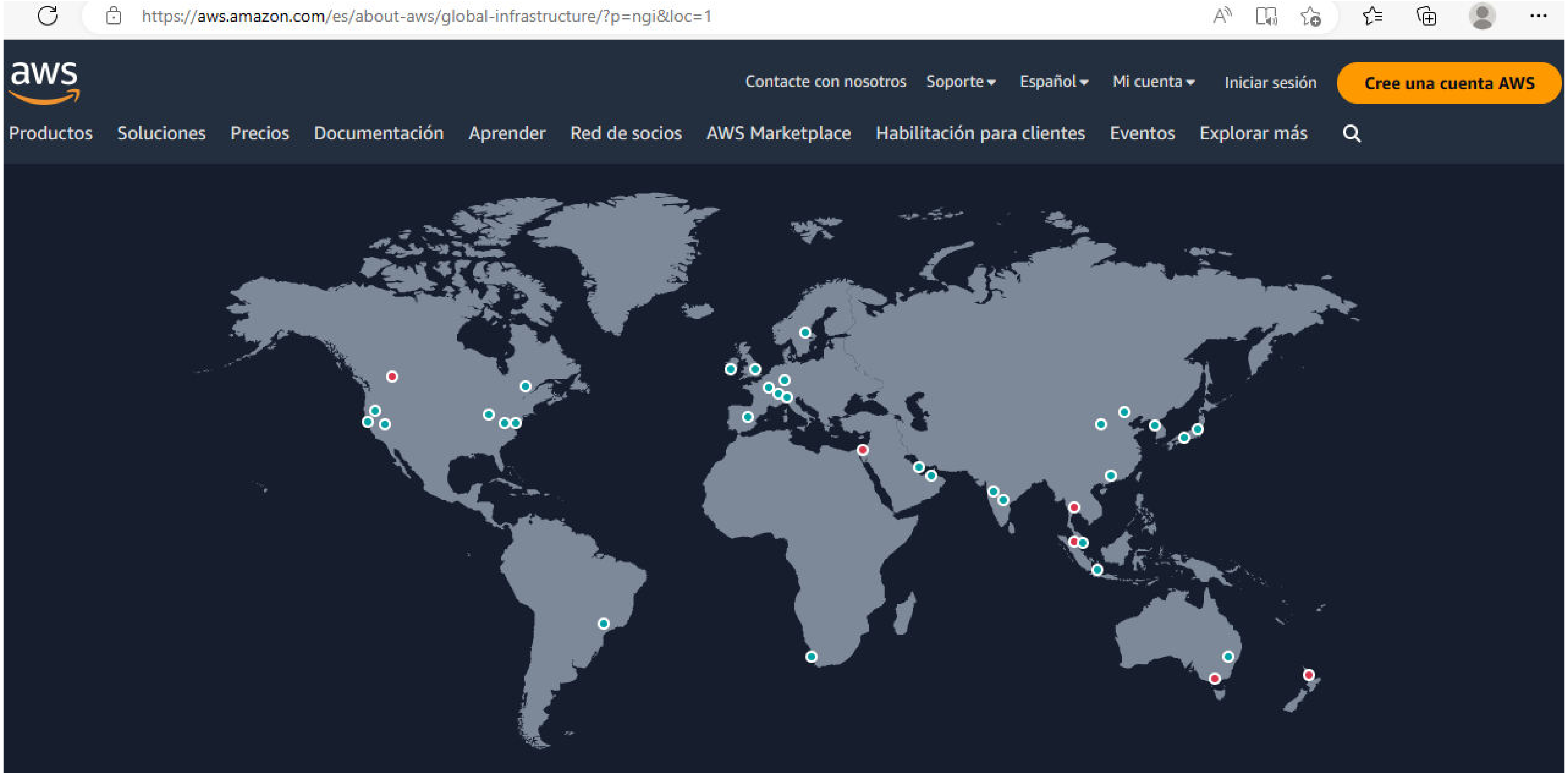
AWS GLOBAL INFRAESTRUCTURE



Again……..

Think about services which can be critical to location……….

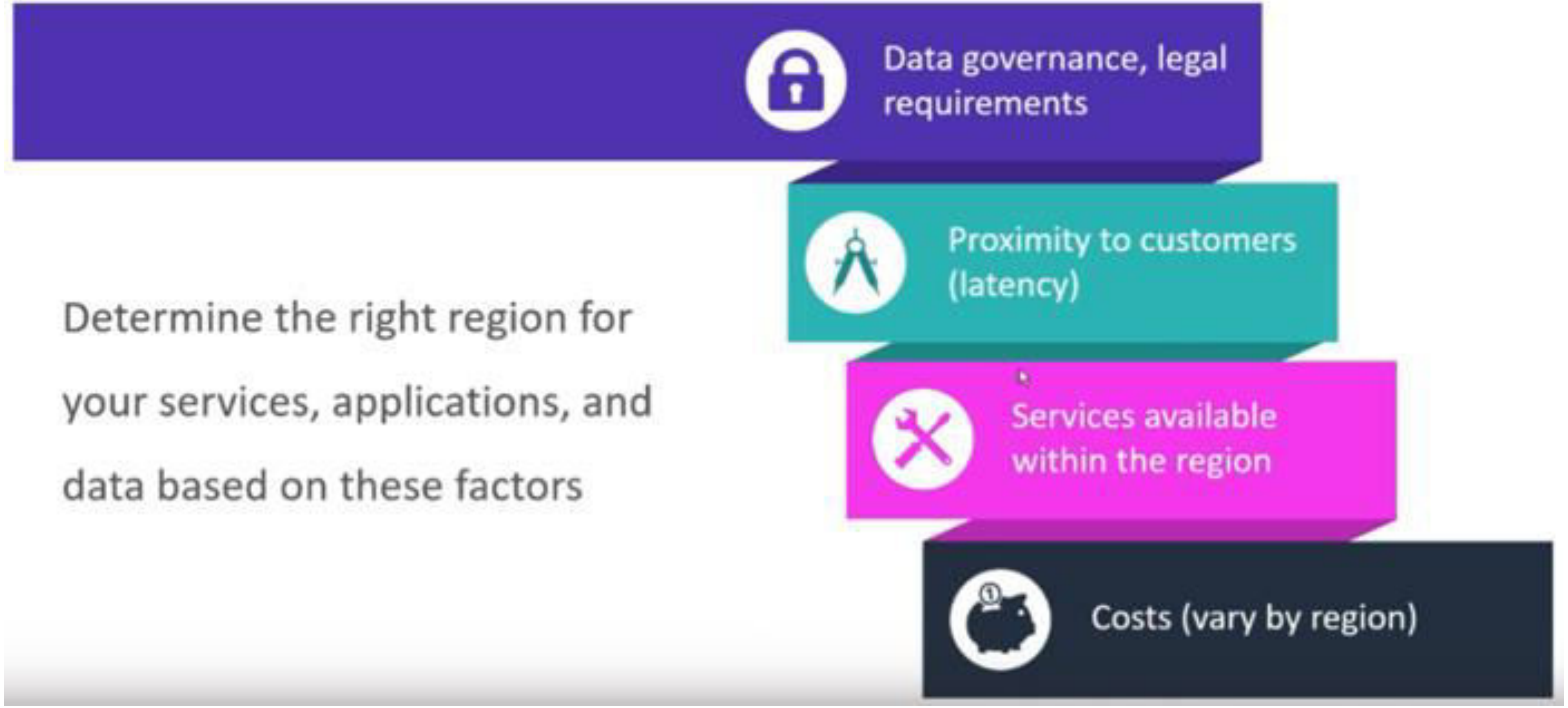
AWS GLOBAL INFRAESTRUCTURE



[Regiones y zonas de disponibilidad de la infraestructura global (amazon.com)](https://aws.amazon.com/es/about-aws/global-infrastructure/regions_az/)

https://aws.amazon.com/es/about-aws/global-infrastructure/regions_az/

SELECTING A REGION



Determine the right region for your services, applications, and data based on these factors

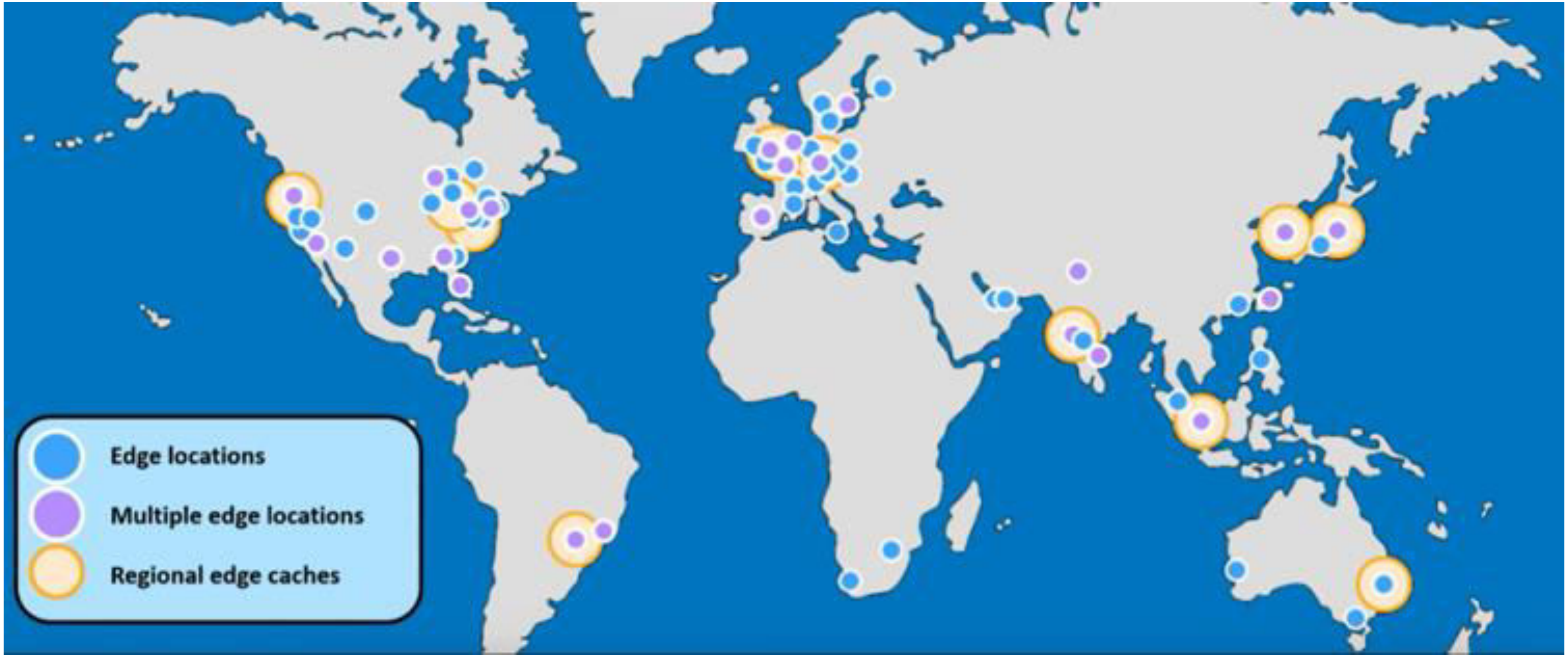
Data governance, legal requirements

Proximity to customers (latency)

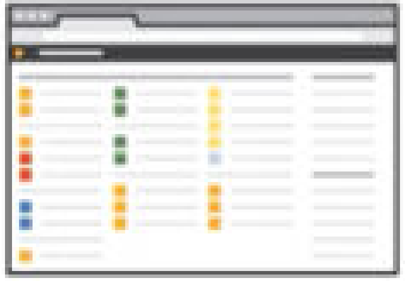
Services available within the region

Costs (vary by region)

EDGE LOCATIONS- REACHING DISTANT CUSTONERS



THREE WAYS TO INTERACT WITH AWS



AWS Management Console

Easy-to-use graphical interface

Command Line Interface (AWS CLI)

Access to services by discrete command

Software development Kits (SDK)

Access services in your code

THREE WAYS TO INTERACT WITH AWS

AWS Management Console

AWS Management Console

AWS CLI

Interfaz de línea de comandos de código abierto

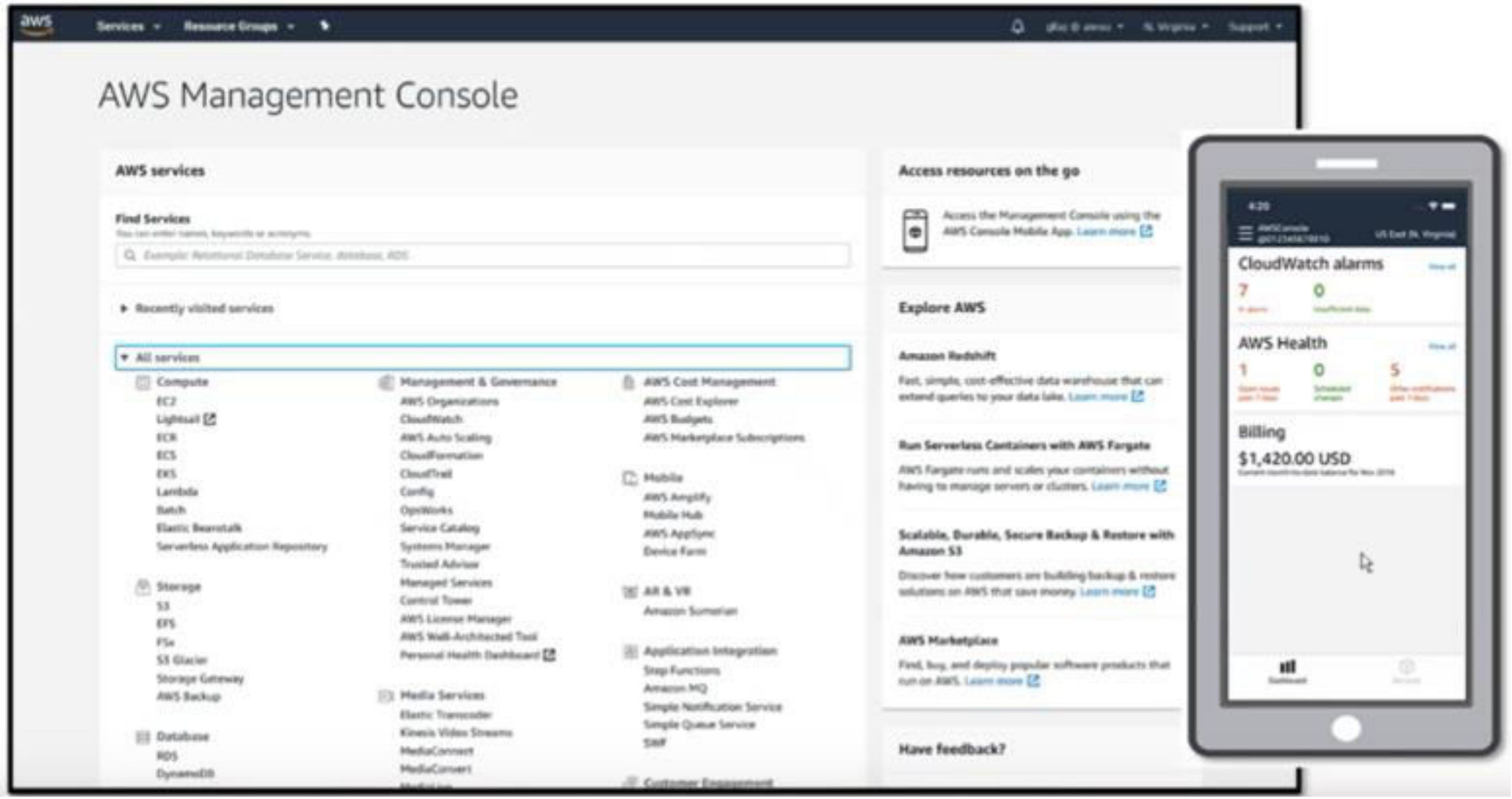
Entornos:

- Linux
- MacOS
- Windows

SDKs

- C++
- Go
- Java
- JavaScript
- .NET
- Node.js
- PHP
- Python
- Ruby

AWS MANAGEMENT CONSOLE



AWS Management Console

AWS services

Find Services

Example: Relational Database Service, database, RDS

▸ Recently visited services

▾ All services

Compute
EC2
Lightsail
ECR
ECS
EKS
Lambda
Batch
Elastic Beanstalk
Serverless Application Repository

Storage
S3
EFS
FSx
S3 Glacier
Storage Gateway
AWS Backup

Database
RDS
DynamoDB

Management & Governance
AWS Organizations
CloudWatch
AWS Auto Scaling
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Systems Manager
Trusted Advisor
Managed Services
Control Tower
AWS License Manager
AWS Well-Architected Tool
Personal Health Dashboard

Media Services
Elastic Transcoder
Kinesis Video Streams
MediaConnect
MediaConvert

AWS Cost Management
AWS Cost Explorer
AWS Budgets
AWS Marketplace Subscriptions

Mobile
AWS Amplify
Mobile Hub
AWS AppSync
Device Farm

AR & VR
Amazon Sumerian

Application Integration
Step Functions
Amazon MQ
Simple Notification Service
Simple Queue Service
SWF

Customer Engagement

Access resources on the go

Access the Management Console using the AWS Console Mobile App. Learn more

Explore AWS

Amazon Redshift

Fast, simple, cost-effective data warehouse that can extend queries to your data lake. Learn more

Run Serverless Containers with AWS Fargate

AWS Fargate runs and scales your containers without having to manage servers or clusters. Learn more

Scalable, Durable, Secure Backup & Restore with Amazon S3

Discover how customers are building backup & restore solutions on AWS that save money. Learn more

AWS Marketplace

Find, buy, and deploy popular software products that run on AWS. Learn more

Have feedback?

4:20

US East (N. Virginia)

CloudWatch alarms

7

0

AWS Health

1

0

5

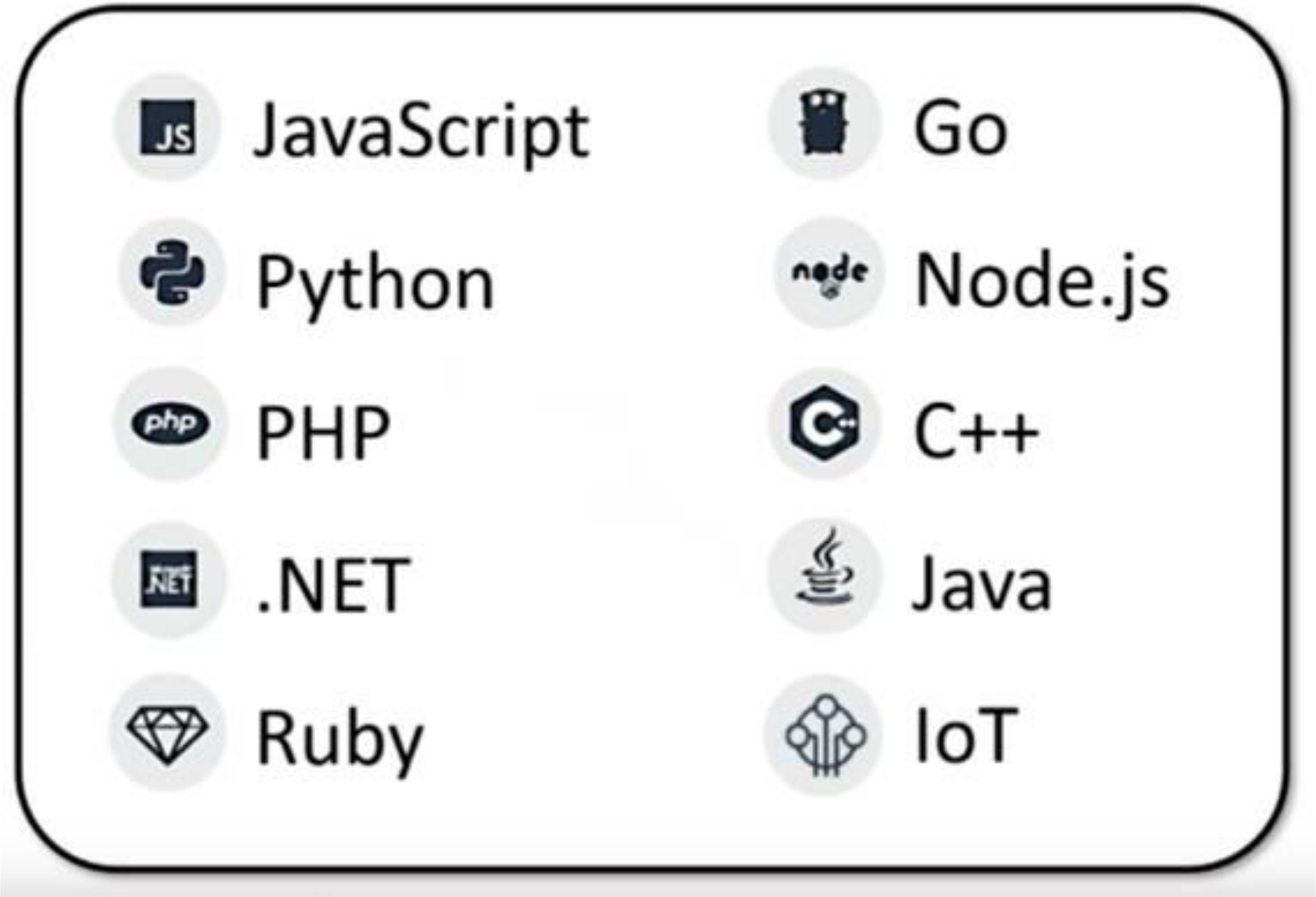
Billing

$1,420.00 USD

AWS CLI

- Open source tool for interacting with AWS services
- Environments
  - Linux
  - MacOS
  - Windows

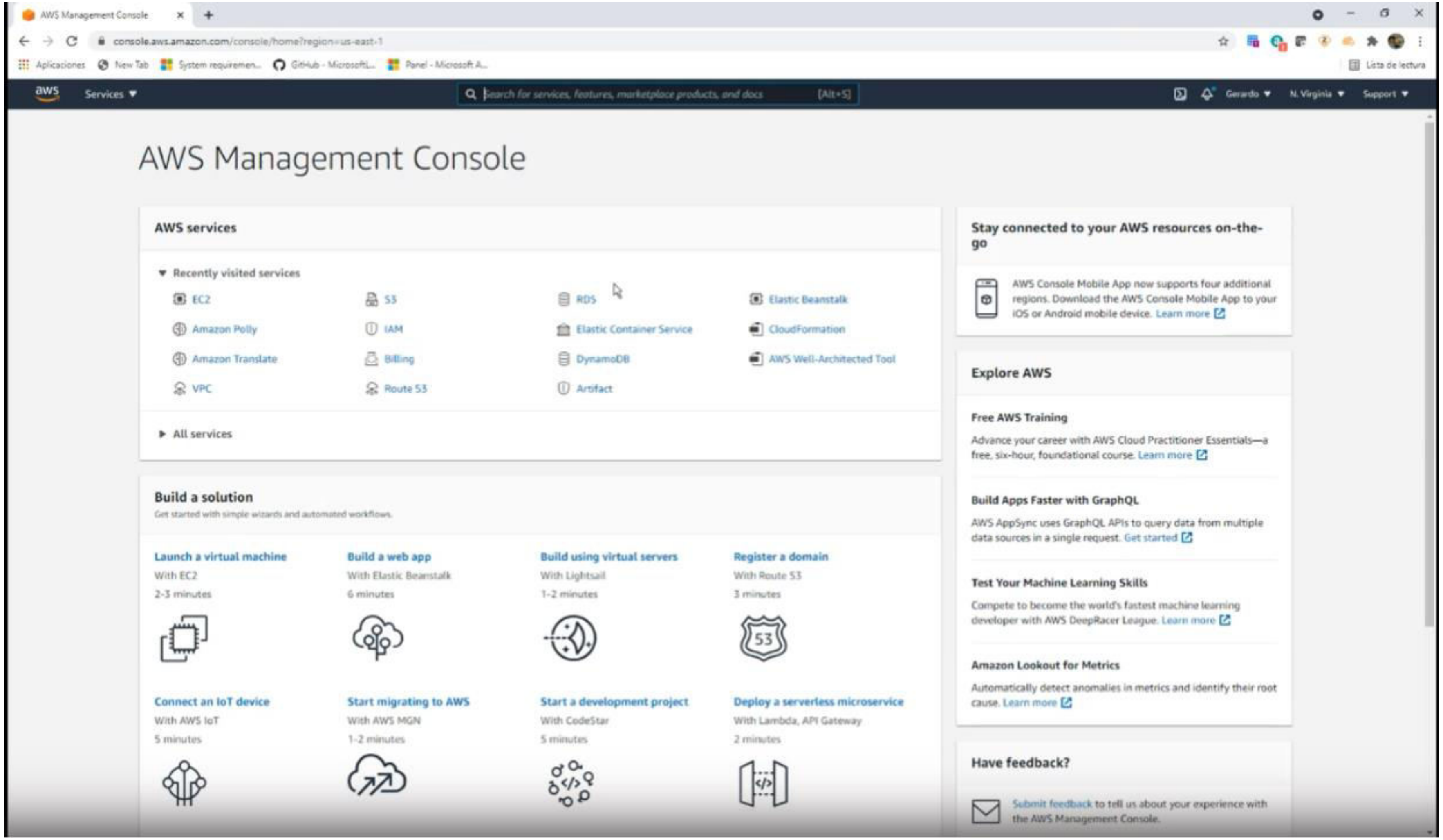
AWS SDK



- JavaScript
- Python
- PHP
- .NET
- Ruby
- Go
- Node.js
- C++
- Java
- IoT

DEMO: CLI , CONSOLE

FREE TIER



AWS Management Console

AWS services

▼ Recently visited services

EC2	S3	RDS	Elastic Beanstalk
Amazon Polly	IAM	Elastic Container Service	CloudFormation
Amazon Translate	Billing	DynamoDB	AWS Well-Architected Tool
VPC	Route 53	Artifact	

▶ All services

Build a solution

Get started with simple wizards and automated workflows.

Launch a virtual machine With EC2 2-3 minutes	Build a web app With Elastic Beanstalk 6 minutes	Build using virtual servers With Lightsail 1-2 minutes	Register a domain With Route 53 3 minutes
Connect an IoT device With AWS IoT 5 minutes	Start migrating to AWS With AWS MGN 1-2 minutes	Start a development project With CodeStar 5 minutes	Deploy a serverless microservice With Lambda, API Gateway 2 minutes

Stay connected to your AWS resources on-the-go

AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more

Explore AWS

Free AWS Training

Advance your career with AWS Cloud Practitioner Essentials—a free, six-hour, foundational course. Learn more

Build Apps Faster with GraphQL

AWS AppSync uses GraphQL APIs to query data from multiple data sources in a single request. Get started

Test Your Machine Learning Skills

Compete to become the world's fastest machine learning developer with AWS DeepRacer League. Learn more

Amazon Lookout for Metrics

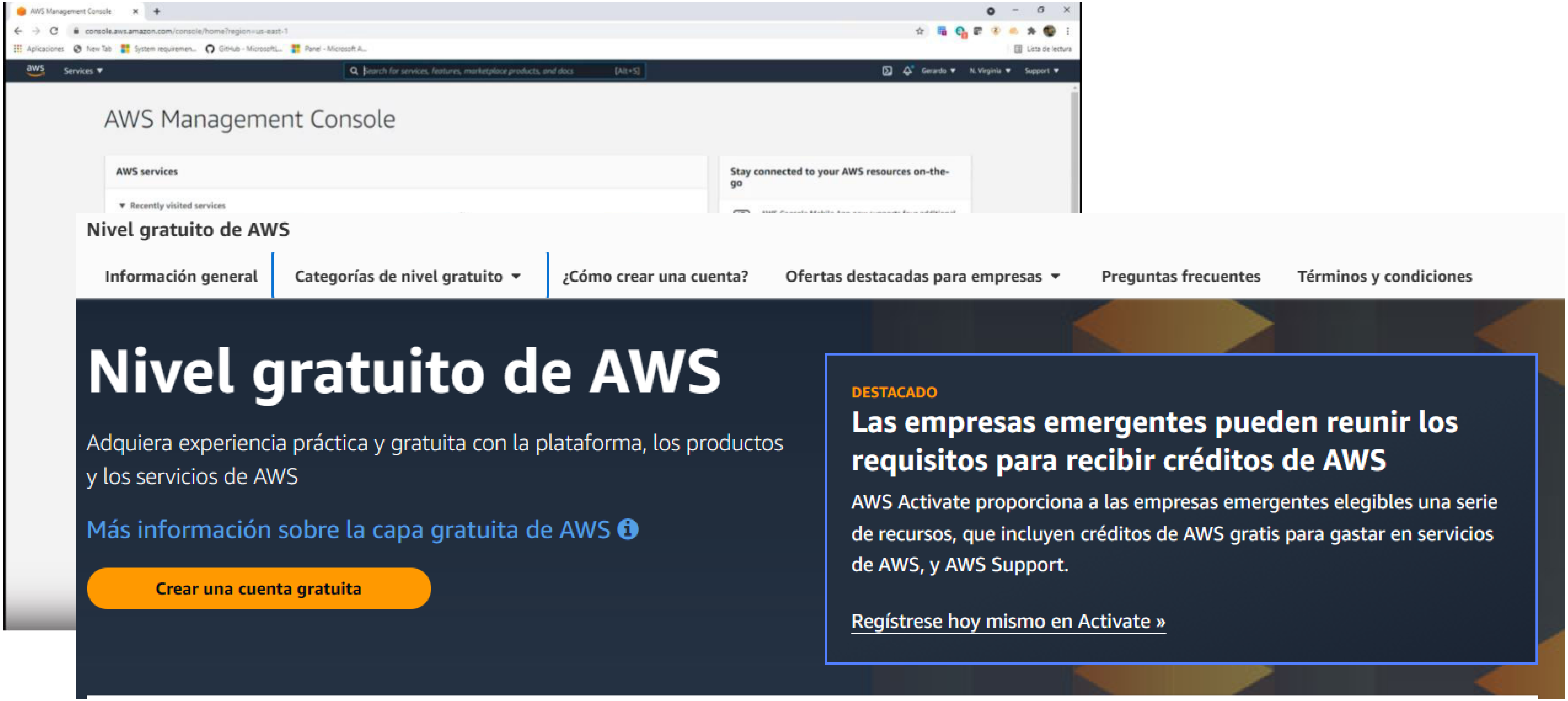
Automatically detect anomalies in metrics and identify their root cause. Learn more

Have feedback?

Submit feedback to tell us about your experience with the AWS Management Console.

https://aws.amazon.com/es/free/

FREE TIER



AWS Management Console

AWS services

▼ Recently visited services

Stay connected to your AWS resources on-the-go

Nivel gratuito de AWS

Información general | Categorías de nivel gratuito ▾ | ¿Cómo crear una cuenta? | Ofertas destacadas para empresas ▾ | Preguntas frecuentes | Términos y condiciones

Nivel gratuito de AWS

Adquiera experiencia práctica y gratuita con la plataforma, los productos y los servicios de AWS

Más información sobre la capa gratuita de AWS

Crear una cuenta gratuita

DESTACADO

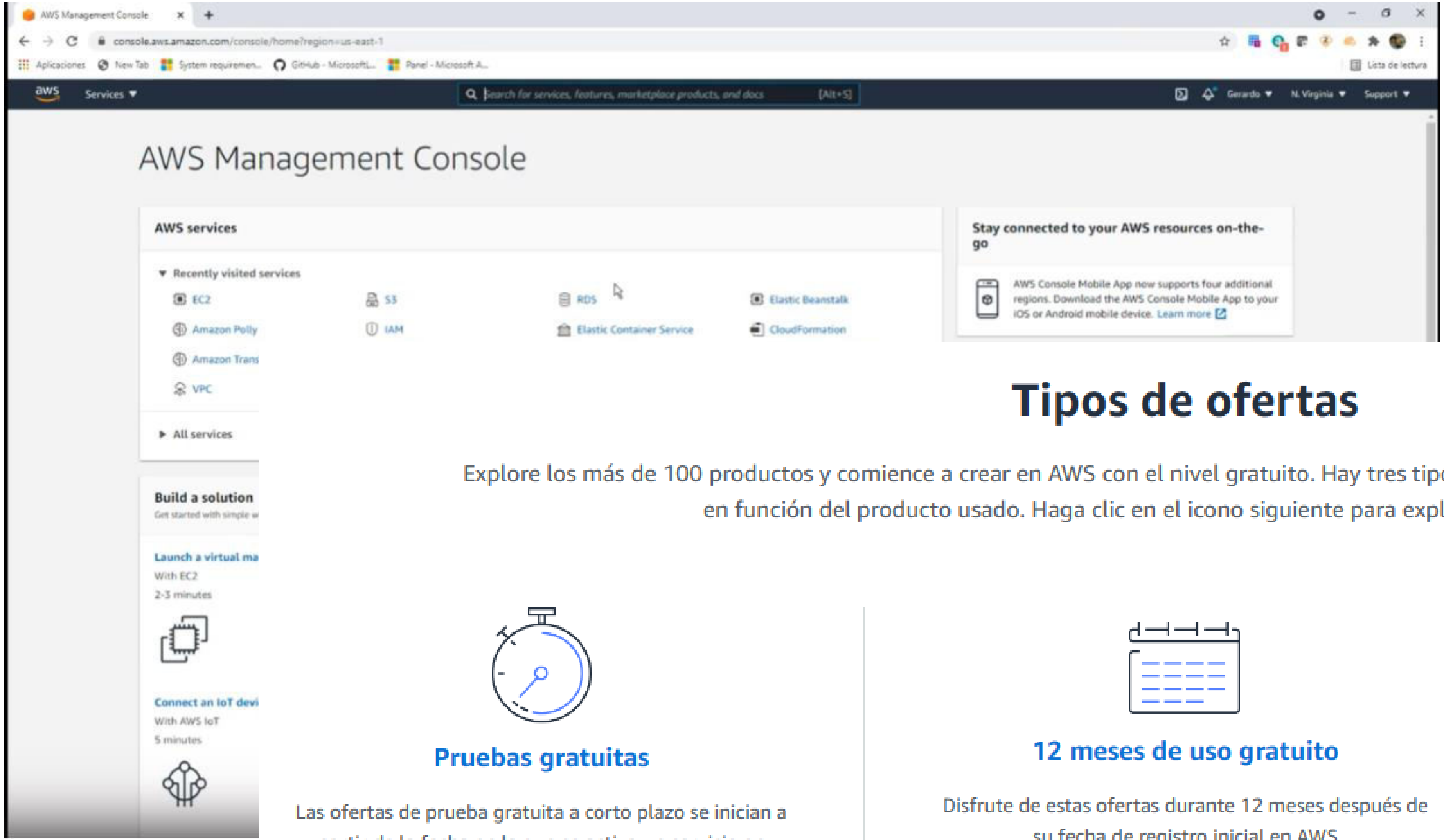
Las empresas emergentes pueden reunir los requisitos para recibir créditos de AWS

AWS Activate proporciona a las empresas emergentes elegibles una serie de recursos, que incluyen créditos de AWS gratis para gastar en servicios de AWS, y AWS Support.

Regístrese hoy mismo en Activate »

https://aws.amazon.com/es/free/

FREE TIER



AWS Management Console

AWS services

▼ Recently visited services

- EC2
- S3
- RDS
- Elastic Beanstalk
- Amazon Polly
- IAM
- Elastic Container Service
- CloudFormation
- Amazon Trans
- VPC

▶ All services

Build a solution

Get started with simple w

Launch a virtual ma
With EC2
2-3 minutes

Connect an IoT devi
With AWS IoT
5 minutes

Stay connected to your AWS resources on-the-go

AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more

Tipos de ofertas

Explore los más de 100 productos y comience a crear en AWS con el nivel gratuito. Hay tres tipos diferentes de ofertas gratuitas disponibles en función del producto usado. Haga clic en el icono siguiente para explorar nuestras ofertas.

Pruebas gratuitas

Las ofertas de prueba gratuita a corto plazo se inician a partir de la fecha en la que se activa un servicio en particular

12 meses de uso gratuito

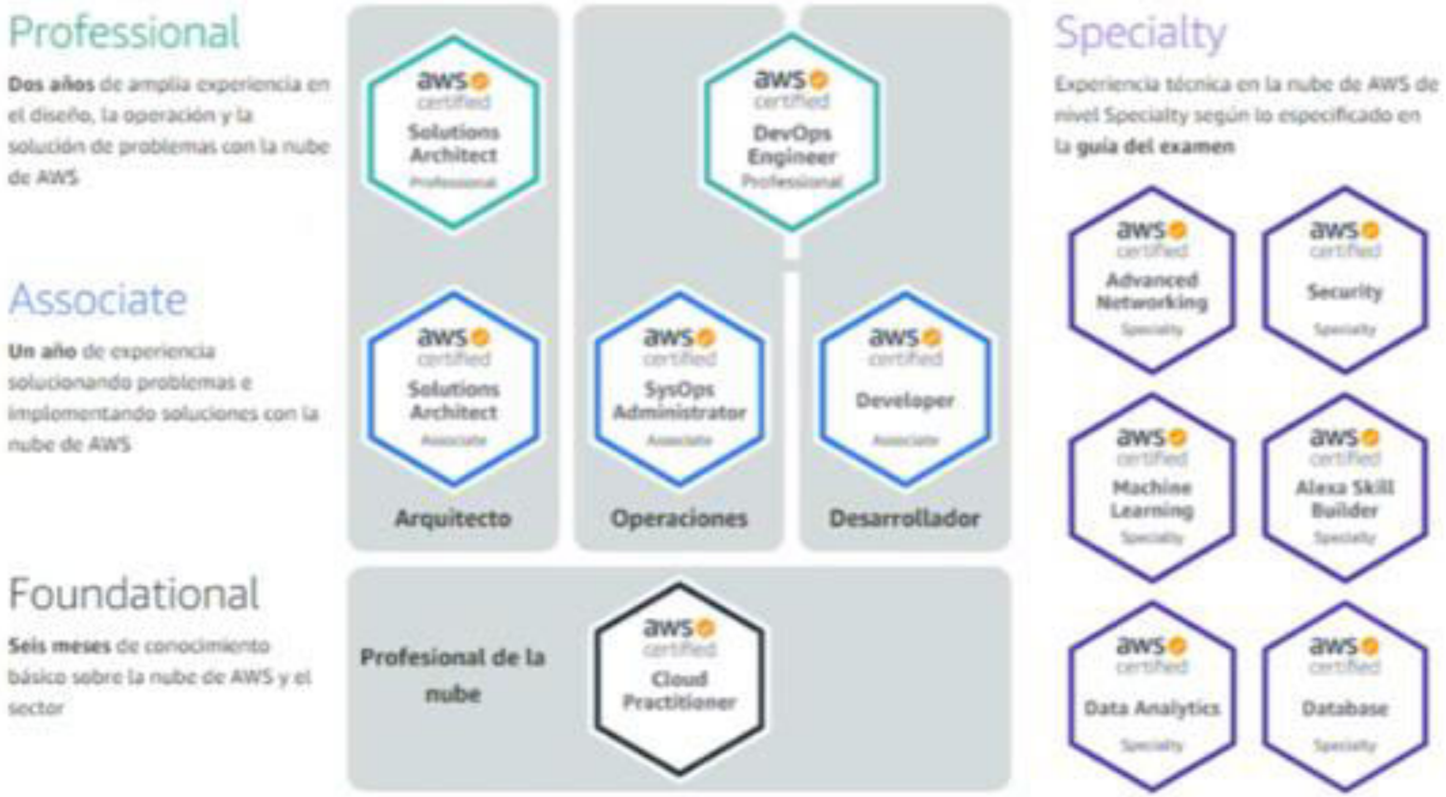
Disfrute de estas ofertas durante 12 meses después de su fecha de registro inicial en AWS

Gratis para siempre

Estas ofertas del nivel gratuito no caducan y están disponibles para todos los clientes de AWS

https://aws.amazon.com/es/free/

AWS PRACTITIONER



Professional

Dos años de amplia experiencia en el diseño, la operación y la solución de problemas con la nube de AWS

Associate

Un año de experiencia solucionando problemas e implementando soluciones con la nube de AWS

Foundational

Seis meses de conocimiento básico sobre la nube de AWS y el sector

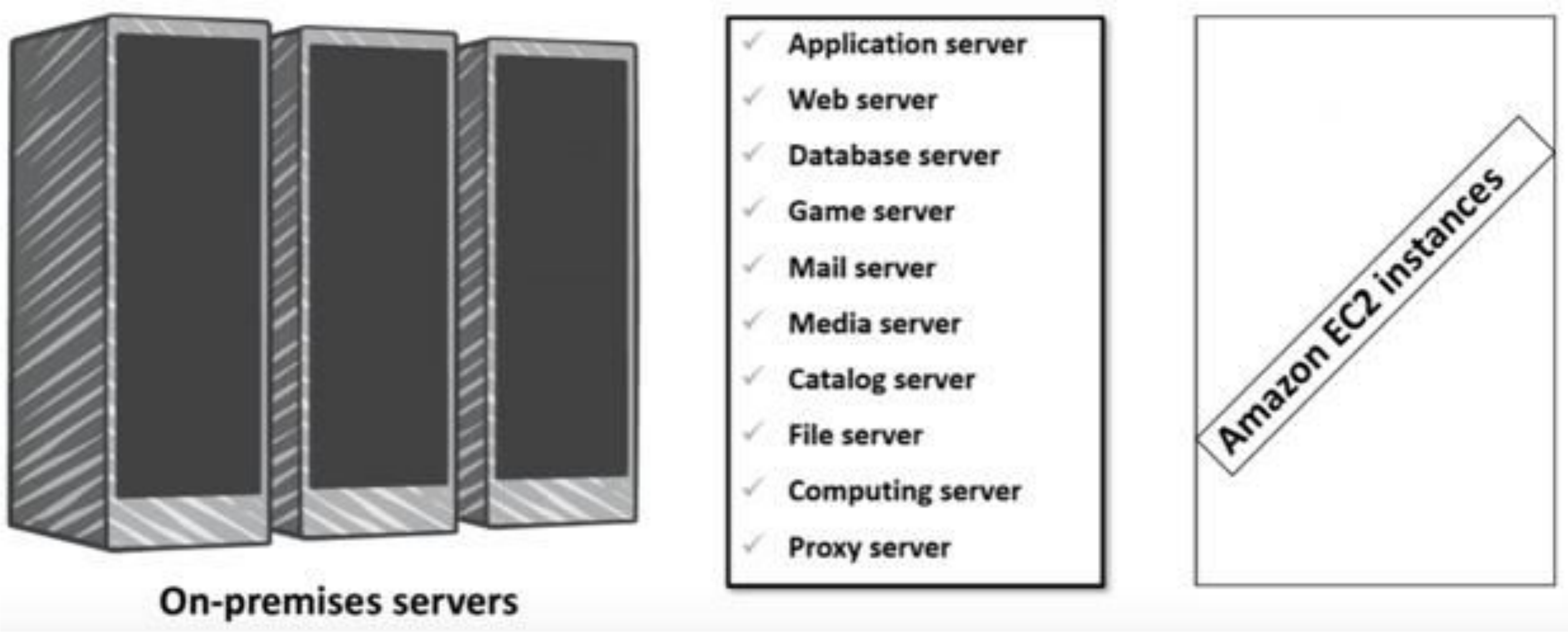
Specialty

Experiencia técnica en la nube de AWS de nivel Specialty según lo especificado en **la guía del examen**

aws certified Solutions Architect Professional
aws certified DevOps Engineer Professional
aws certified Solutions Architect Associate
aws certified SysOps Administrator Associate
aws certified Developer Associate
Arquitecto
Operaciones
Desarrollador
Profesional de la nube
aws certified Cloud Practitioner
aws certified Advanced Networking Specialty
aws certified Security Specialty
aws certified Machine Learning Specialty
aws certified Alexa Skill Builder Specialty
aws certified Data Analytics Specialty
aws certified Database Specialty

Source: AWS

WHAT IS AMAZON E2C?

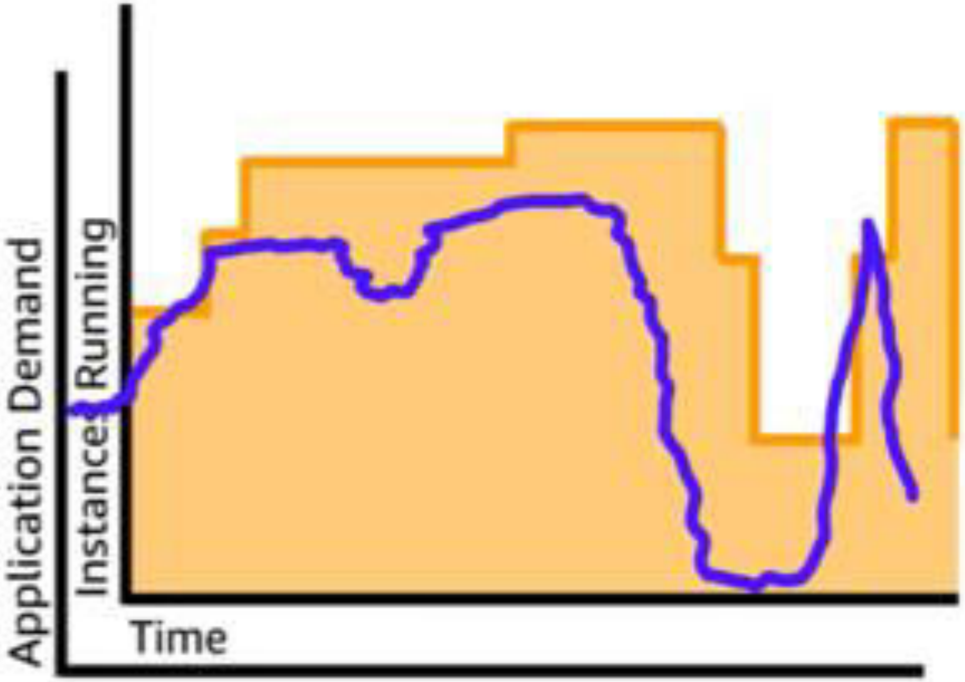


Source: AWS

E2C is compute memory and networking infrastructure in the AWS cloud

BENEFITS OF AMAZON E2C

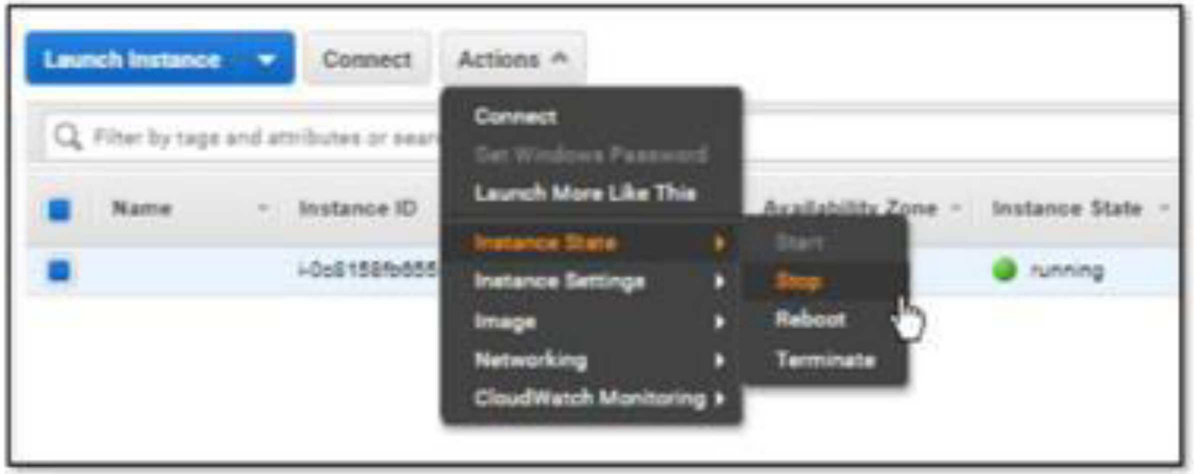
- Elasticity



Source: AWS

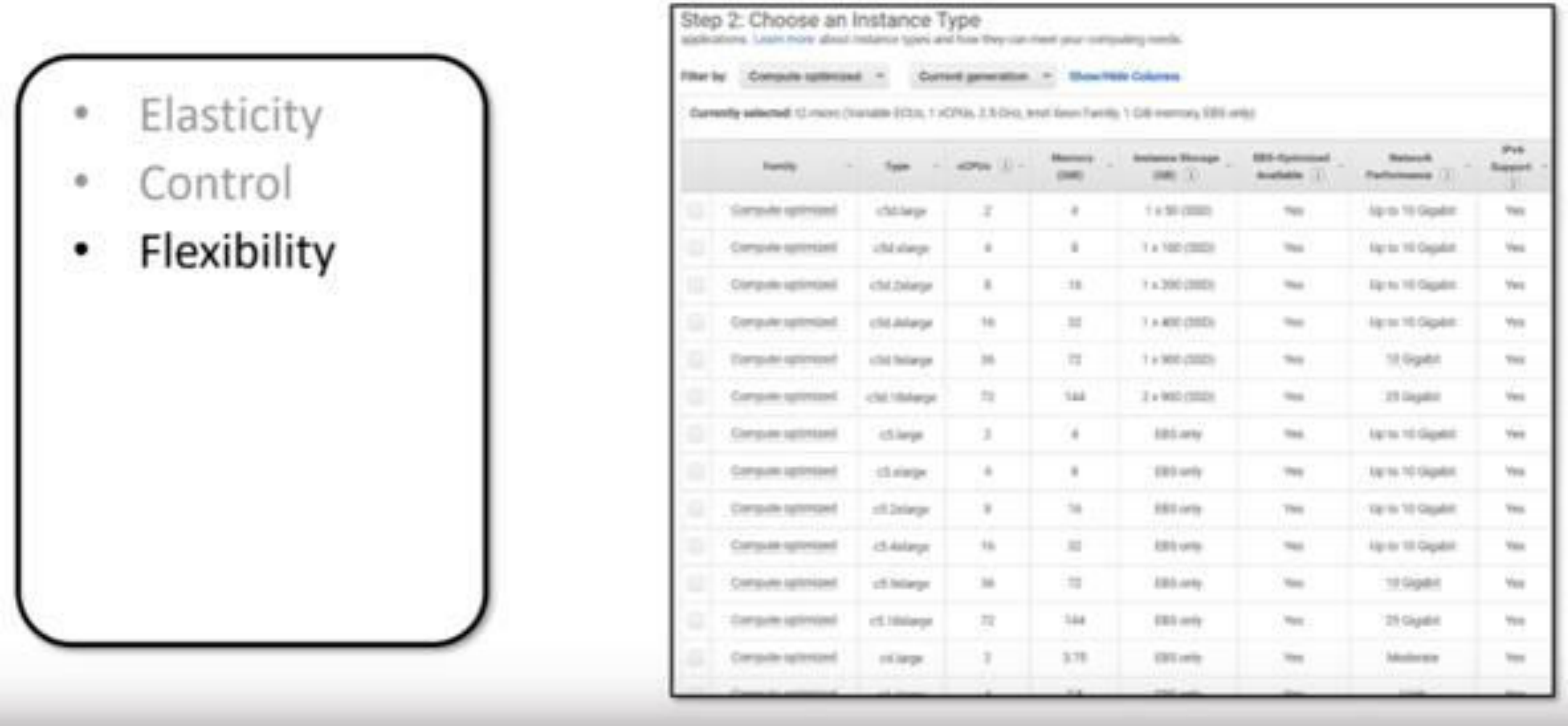
BENEFITS OF AMAZON E2C

- Elasticity
- Control



Source: AWS

BENEFITS OF AMAZON E2C



- Elasticity
- Control
- Flexibility

Step 2: Choose an Instance Type

applications. Learn more about instance types and how they can meet your computing needs.

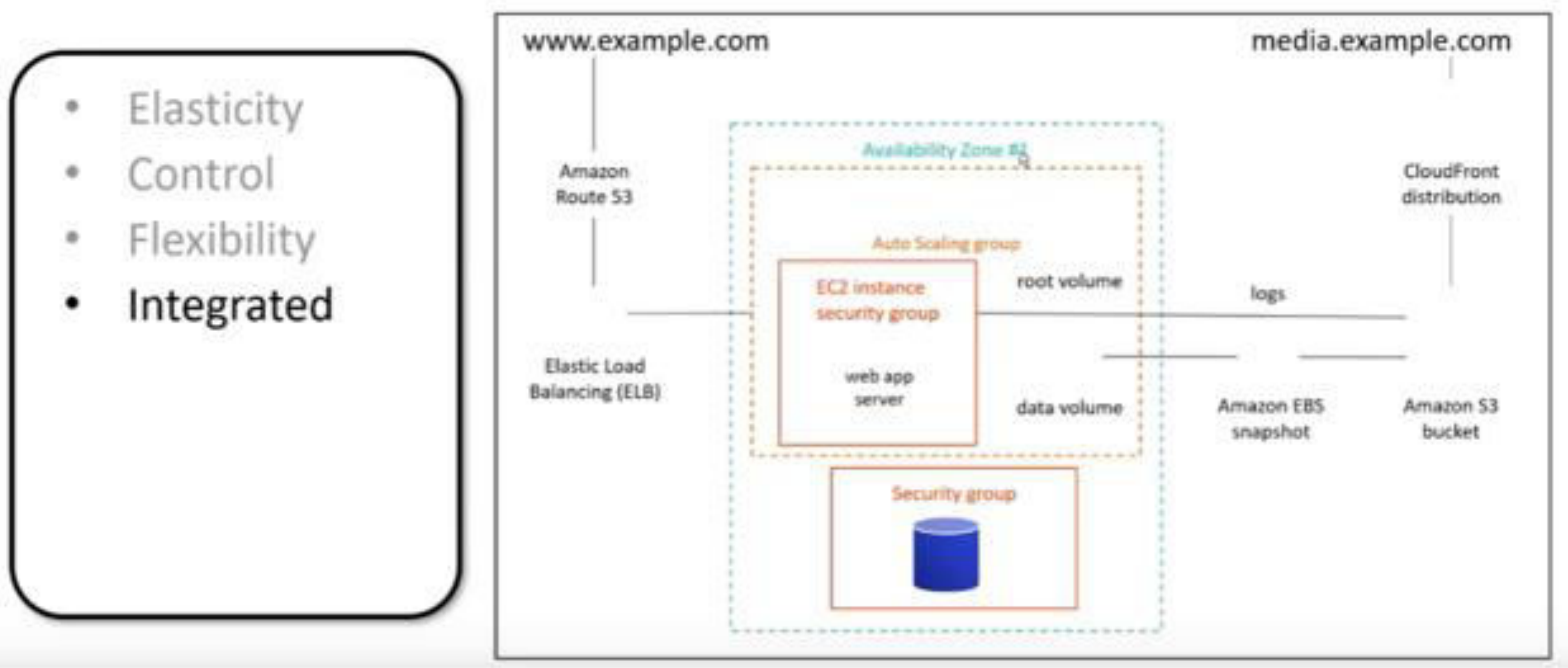
Filter by: Compute optimized ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
□	Compute optimized	c5d.large	2	4	1 x 50 (SSD)	Yes	Up to 10 Gigabit	Yes
□	Compute optimized	c5d.xlarge	4	8	1 x 100 (SSD)	Yes	Up to 10 Gigabit	Yes
□	Compute optimized	c5d.2xlarge	8	16	1 x 200 (SSD)	Yes	Up to 10 Gigabit	Yes
□	Compute optimized	c5d.4xlarge	16	32	1 x 400 (SSD)	Yes	Up to 10 Gigabit	Yes
□	Compute optimized	c5d.9xlarge	36	72	1 x 900 (SSD)	Yes	10 Gigabit	Yes
□	Compute optimized	c5d.18xlarge	72	144	2 x 900 (SSD)	Yes	25 Gigabit	Yes
□	Compute optimized	c5.large	2	4	EBS only	Yes	Up to 10 Gigabit	Yes
□	Compute optimized	c5.xlarge	4	8	EBS only	Yes	Up to 10 Gigabit	Yes
□	Compute optimized	c5.2xlarge	8	16	EBS only	Yes	Up to 10 Gigabit	Yes
□	Compute optimized	c5.4xlarge	16	32	EBS only	Yes	Up to 10 Gigabit	Yes
□	Compute optimized	c5.9xlarge	36	72	EBS only	Yes	10 Gigabit	Yes
□	Compute optimized	c5.18xlarge	72	144	EBS only	Yes	25 Gigabit	Yes
□	Compute optimized	c4.large	2	3.75	EBS only	Yes	Moderate	Yes

Source: AWS

BENEFITS OF AMAZON E2C



- Elasticity
- Control
- Flexibility
- Integrated

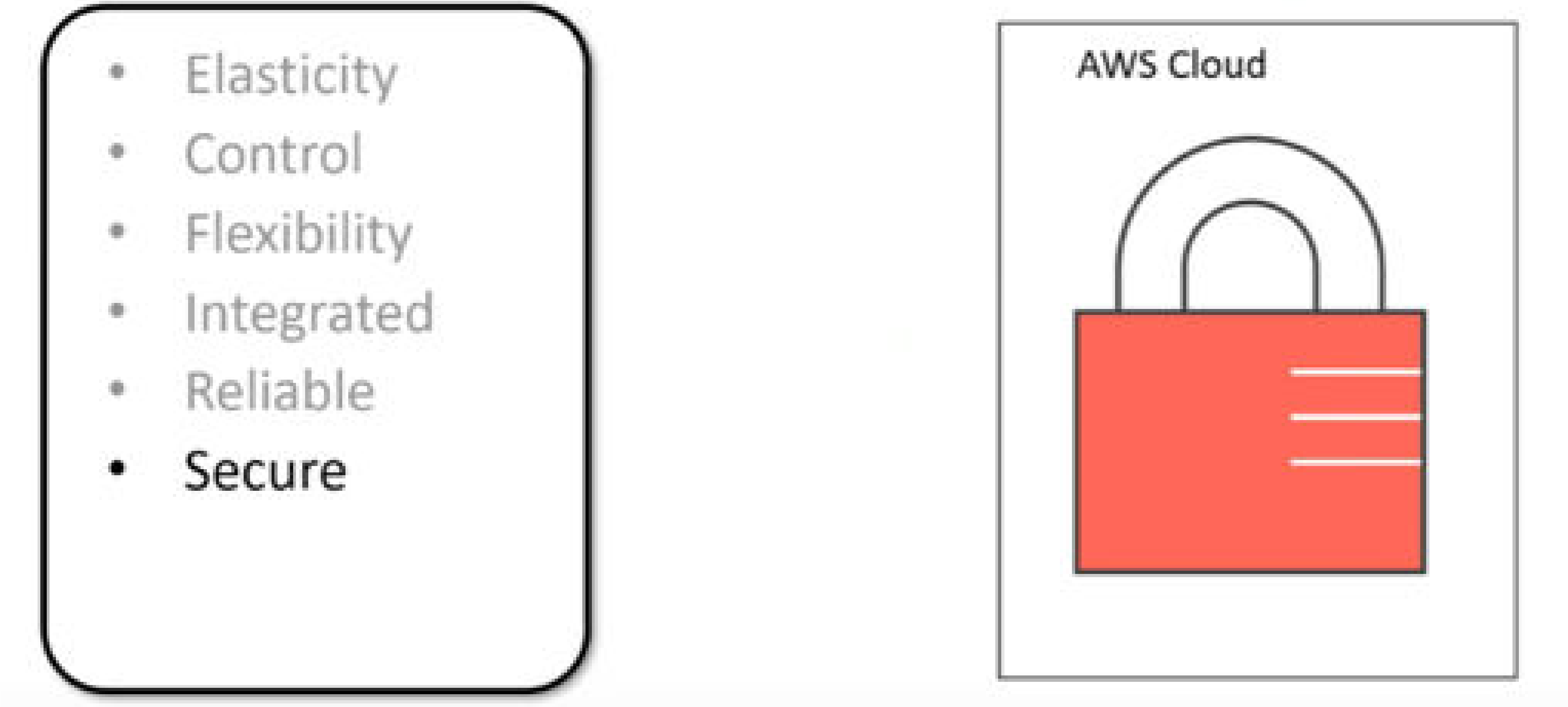
Source: AWS

BENEFITS OF AMAZON E2C

- Elasticity
- Control
- Flexibility
- Integrated
- Reliable

Source: AWS

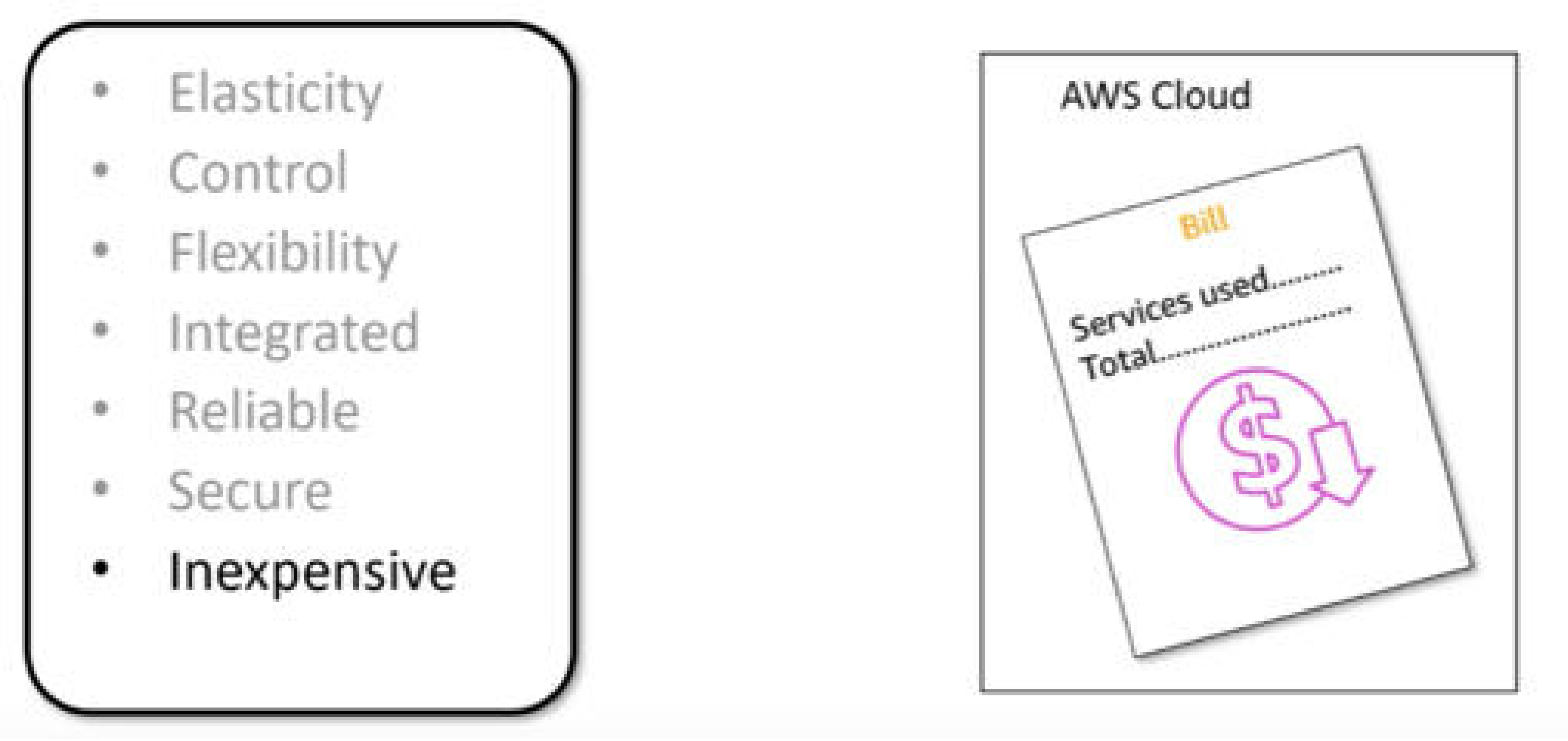
BENEFITS OF AMAZON E2C



- Elasticity
- Control
- Flexibility
- Integrated
- Reliable
- Secure

Source: AWS

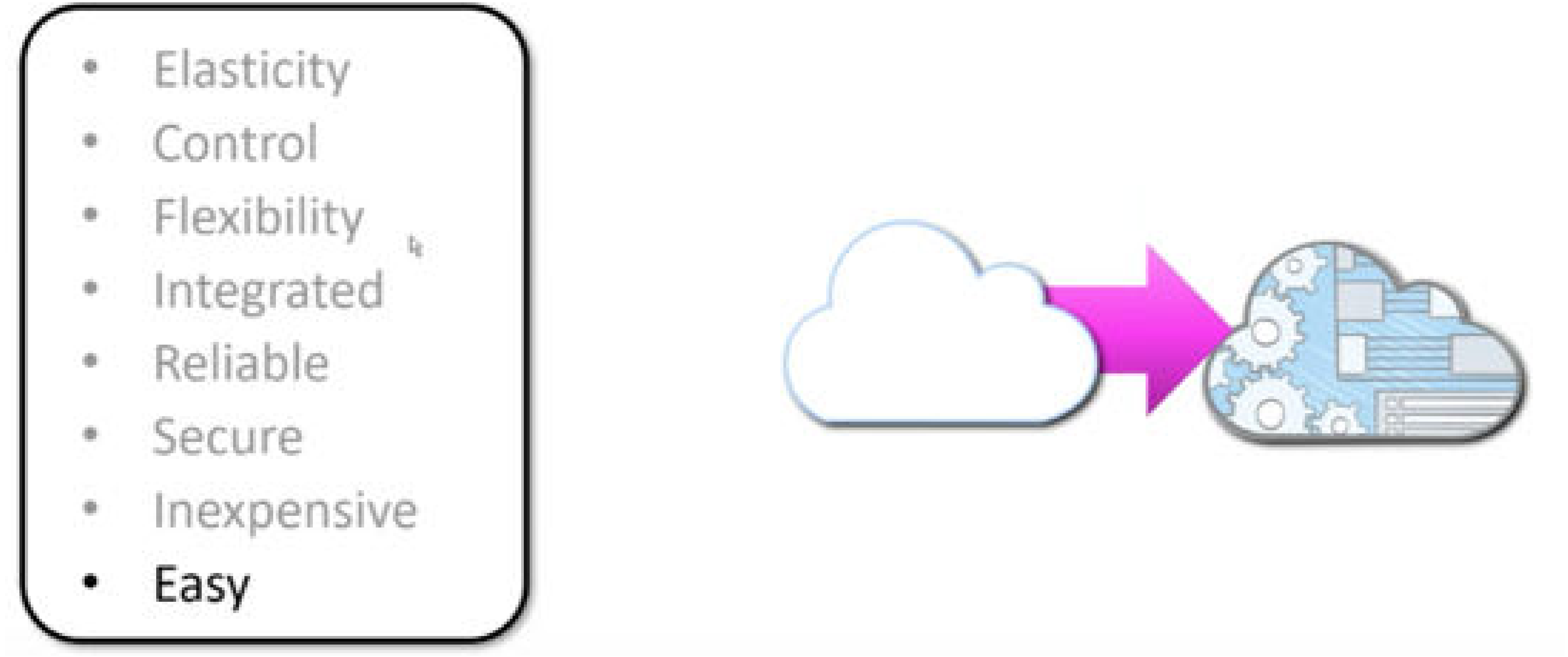
BENEFITS OF AMAZON E2C



- Elasticity
- Control
- Flexibility
- Integrated
- Reliable
- Secure
- Inexpensive

Source: AWS

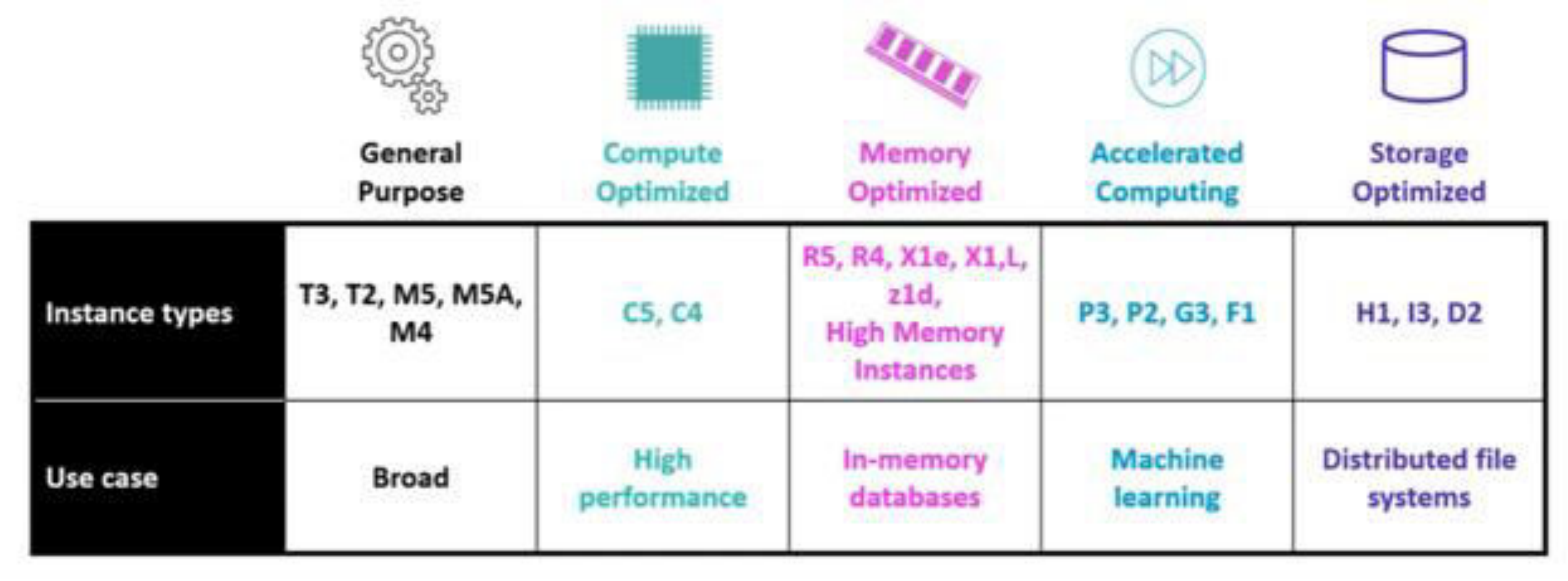
BENEFITS OF AMAZON E2C



- Elasticity
- Control
- Flexibility
- Integrated
- Reliable
- Secure
- Inexpensive
- Easy

Source: AWS

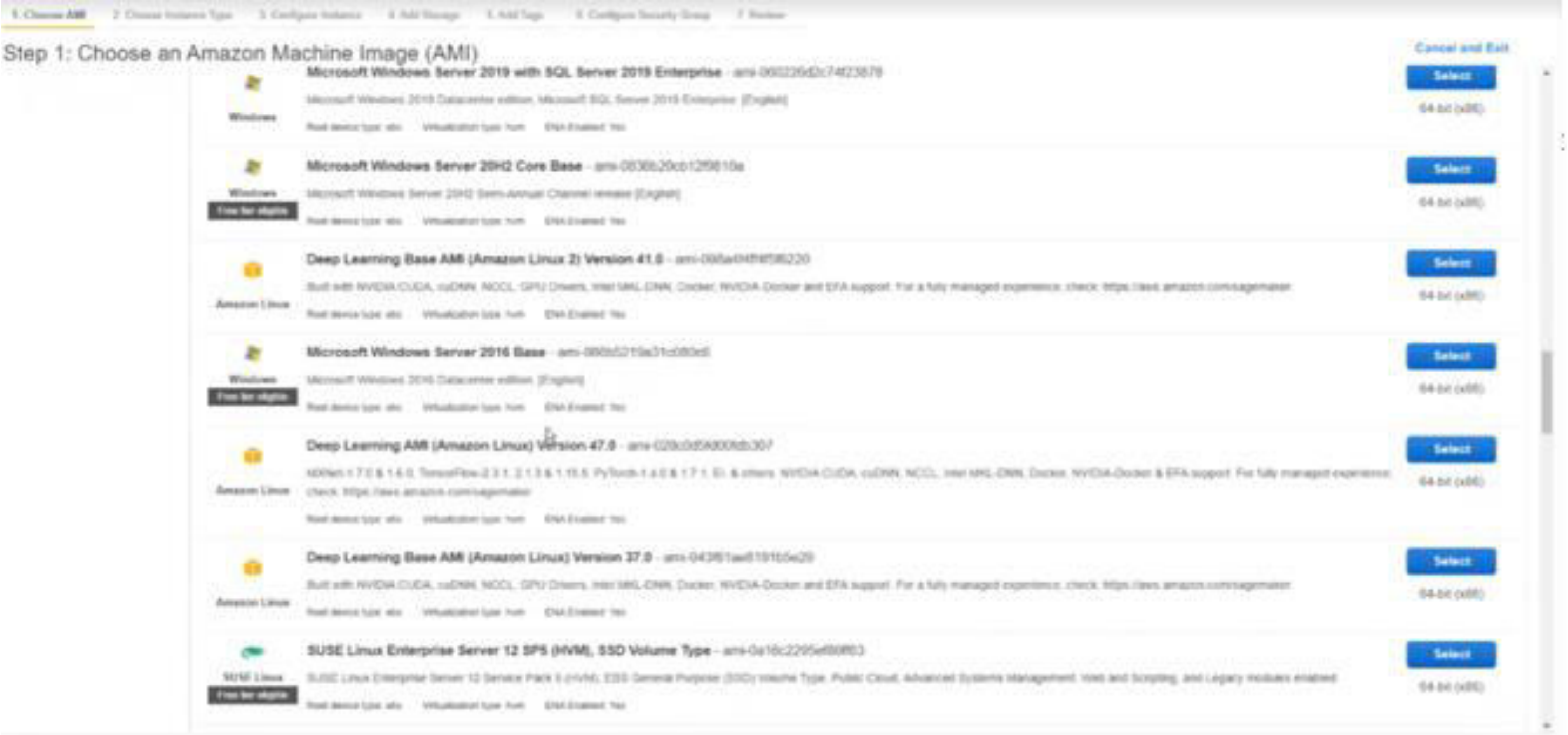
WHAT IS YOUR CASE?



	General Purpose	Compute Optimized	Memory Optimized	Accelerated Computing	Storage Optimized
Instance types	T3, T2, M5, M5A, M4	C5, C4	R5, R4, X1e, X1,L, z1d, High Memory Instances	P3, P2, G3, F1	H1, I3, D2
Use case	Broad	High performance	In-memory databases	Machine learning	Distributed file systems

Source: AWS

AMAZON MACHINE IMAGE



Step 1: Choose an Amazon Machine Image (AMI)

Microsoft Windows Server 2019 with SQL Server 2019 Enterprise - ami-060226d2c74f23878
Microsoft Windows 2019 Datacenter edition, Microsoft SQL Server 2019 Enterprise [English]
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)

Microsoft Windows Server 20H2 Core Base - ami-0836b20cb12f9810a
Microsoft Windows Server 20H2 Semi-Annual Channel release [English]
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)

Deep Learning Base AMI (Amazon Linux 2) Version 41.0 - ami-008a4f4ff5f86220
Built with NVIDIA CUDA, cuDNN, NCCL, GPU Drivers, Intel MKL-DNN, Docker, NVIDIA-Docker and EFA support. For a fully managed experience, check: https://aws.amazon.com/sagemaker
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)

Microsoft Windows Server 2016 Base - ami-0806b5219a31c080e8
Microsoft Windows 2016 Datacenter edition. [English]
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)

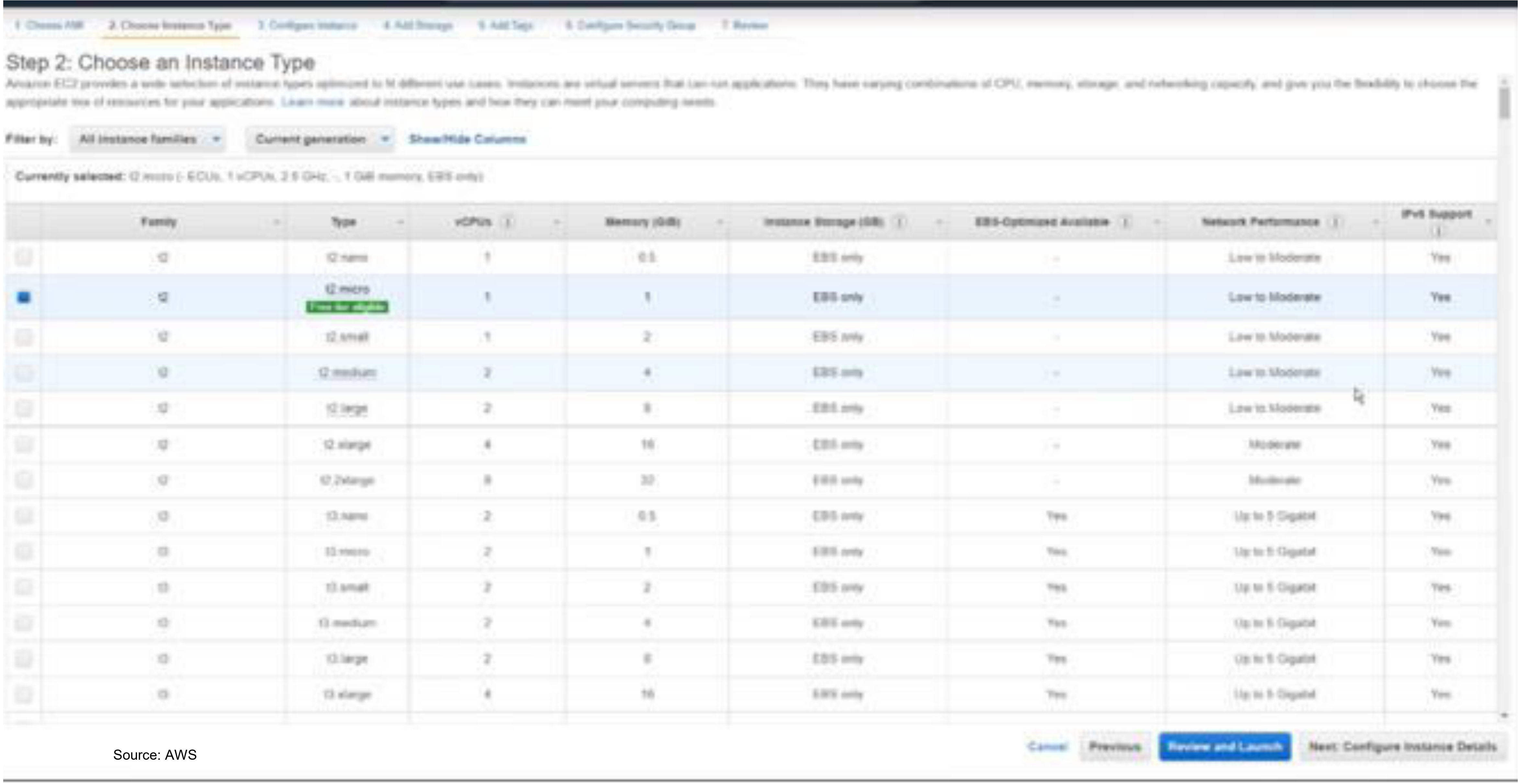
Deep Learning AMI (Amazon Linux) Version 47.0 - ami-020c0d5A000fdb307
MXNet-1.7.0 & 1.6.0, TensorFlow-2.3.1, 2.1.3 & 1.15.5, PyTorch-1.4.0 & 1.7.1, EI, & others. NVIDIA CUDA, cuDNN, NCCL, Intel MKL-DNN, Docker, NVIDIA-Docker & EFA support. For fully managed experience, check: https://aws.amazon.com/sagemaker
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)

Deep Learning Base AMI (Amazon Linux) Version 37.0 - ami-043f61ae8191b5e29
Built with NVIDIA CUDA, cuDNN, NCCL, GPU Drivers, Intel MKL-DNN, Docker, NVIDIA-Docker and EFA support. For a fully managed experience, check: https://aws.amazon.com/sagemaker
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)

SUSE Linux Enterprise Server 12 SP5 (HVM), SSD Volume Type - ami-0a16c2295e80f63
SUSE Linux Enterprise Server 12 Service Pack 5 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)

Source: AWS

AMAZON INSTANCE TYPE



1. Choose AMI · 2. Choose Instance Type · 3. Configure Instance · 4. Add Storage · 5. Add Tags · 6. Configure Security Group · 7. Review

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

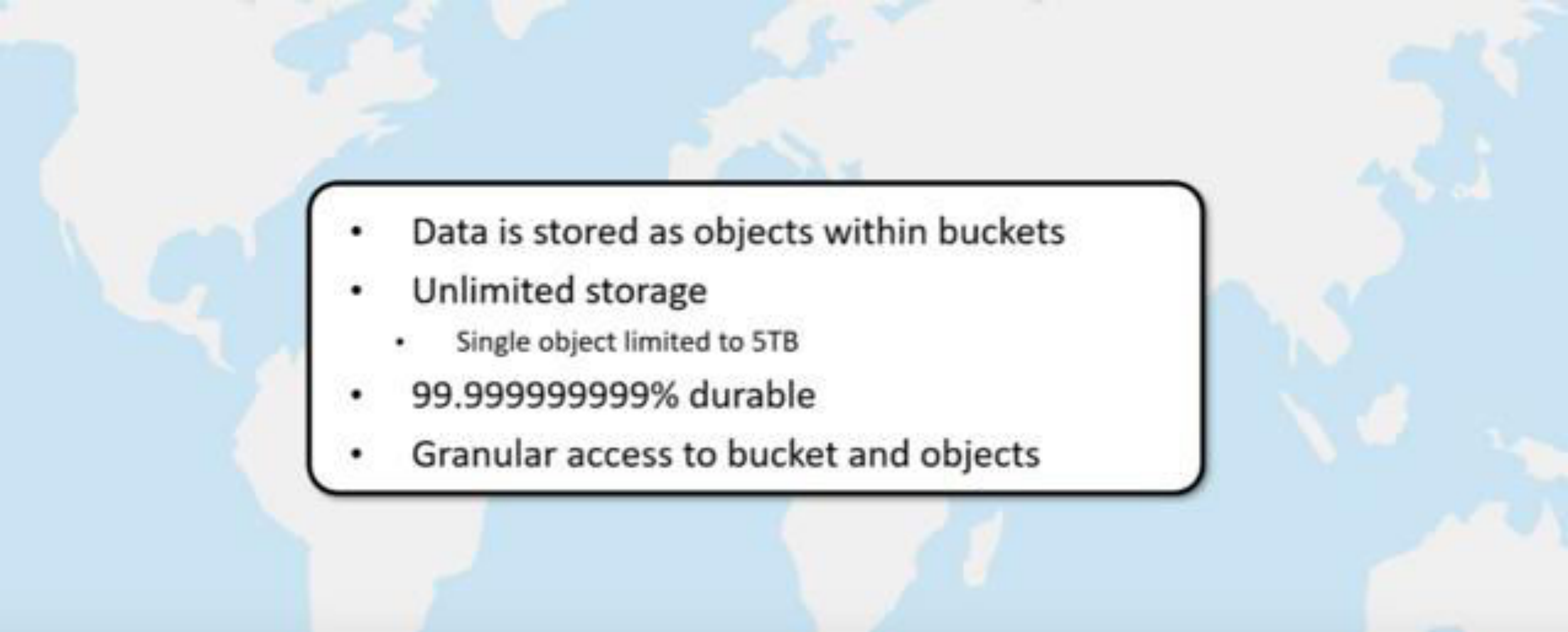
Filter by: All instance families ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
□	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
☑	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
□	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
□	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
□	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
□	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
□	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
□	t3	t3.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
□	t3	t3.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
□	t3	t3.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
□	t3	t3.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes
□	t3	t3.xlarge	4	16	EBS only	Yes	Up to 5 Gigabit	Yes

Source: AWS

WHAT IS AMAZON S3?



- Data is stored as objects within buckets
- Unlimited storage
  - Single object limited to 5TB
- 99.999999999% durable
- Granular access to bucket and objects

Source: AWS

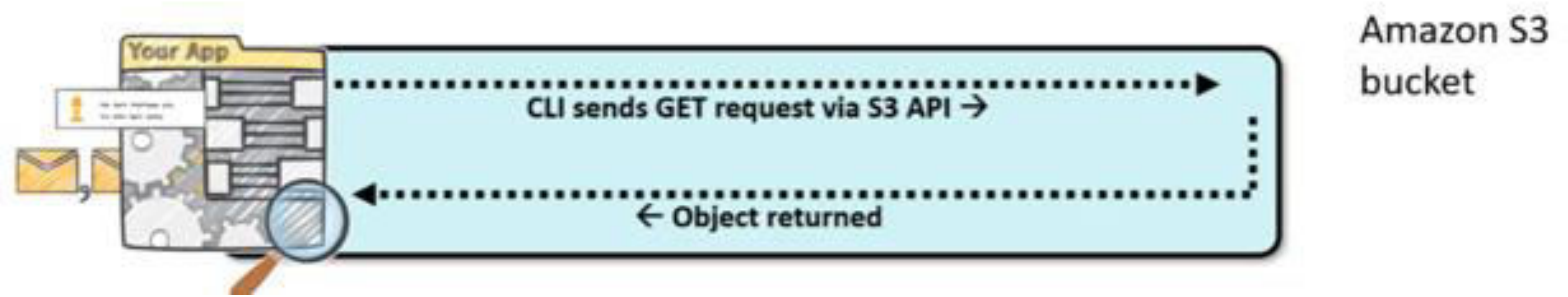
AMAZON S3 (SIMPLE STORAGE SERVICE)

- El servicio S3 es útil para guardar datos que no se utilizan mucho (por ej backups).
- Se guardan como objeto en buckets (repositorio).
- Cada objeto no puede superar los 5TB.

Source: AWS

AMAZON S3 (CORE FUNCTIONALITY)

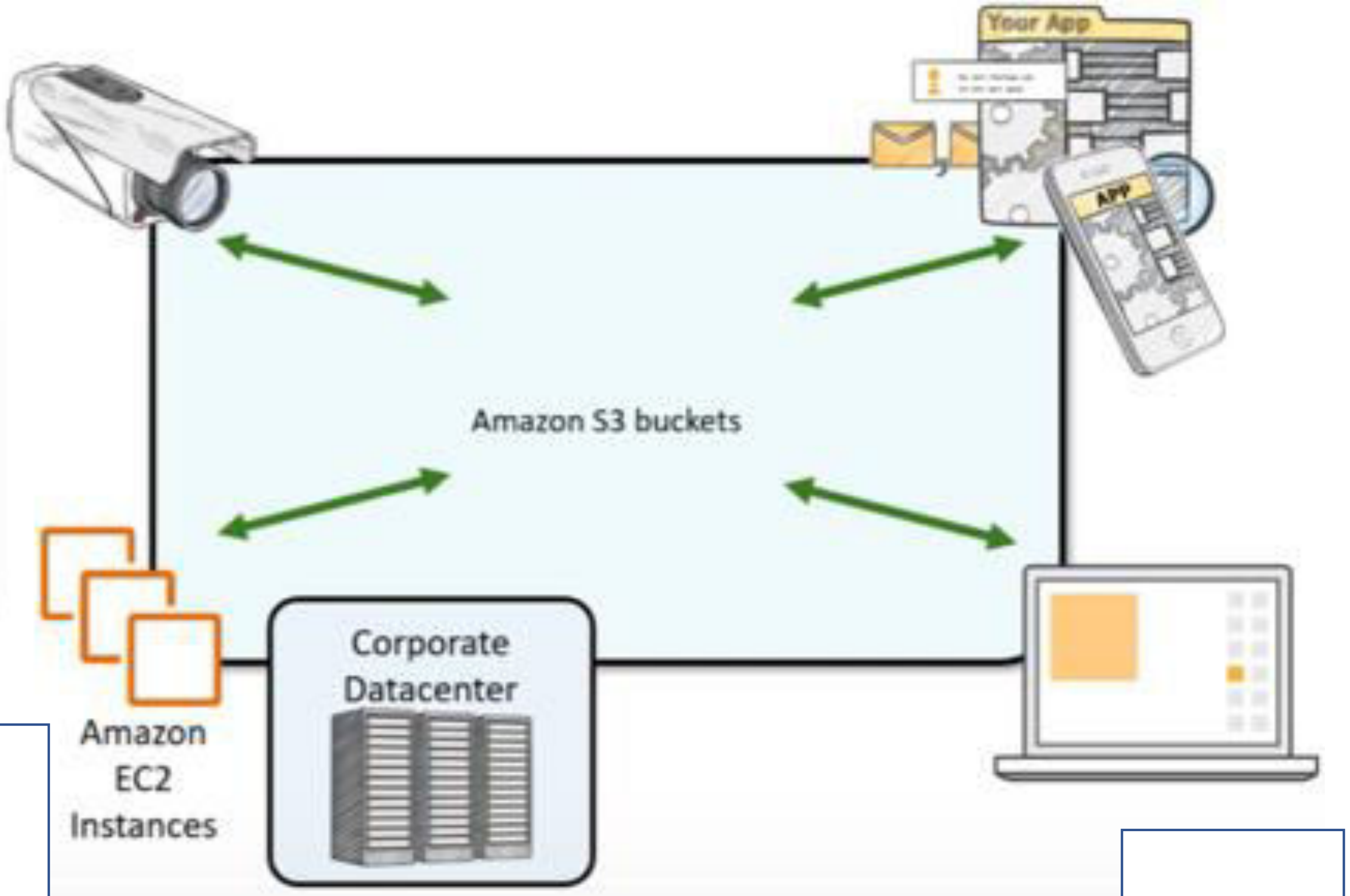
- Fast, durable, highly available key-based access to objects
- Object storage built to store and retrieve data
- Not a file system



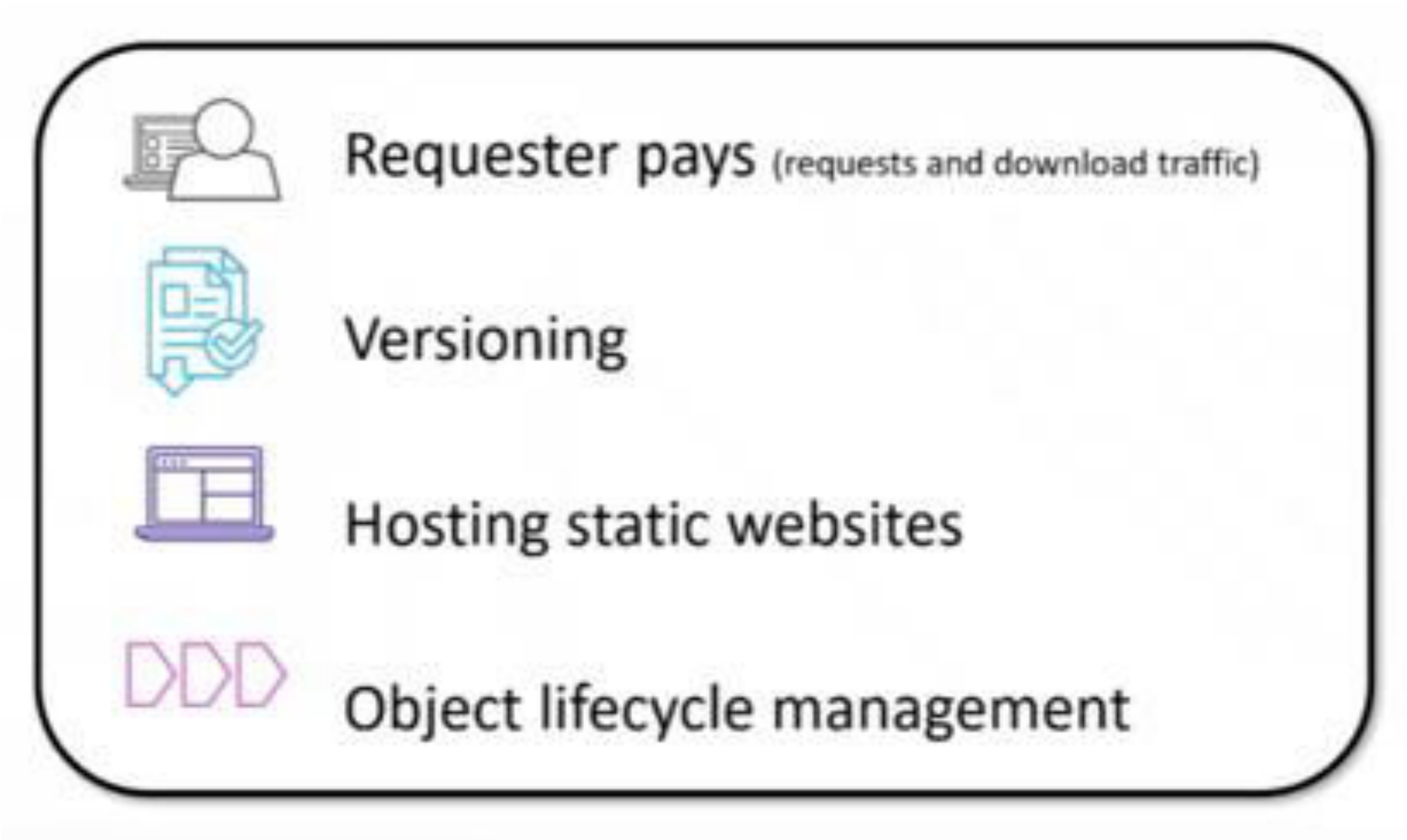
Source: AWS

AMAZON S3 (COMMON SCENARIOS)

- Backup and storage
- Application hosting
- Media hosting
- Software delivery

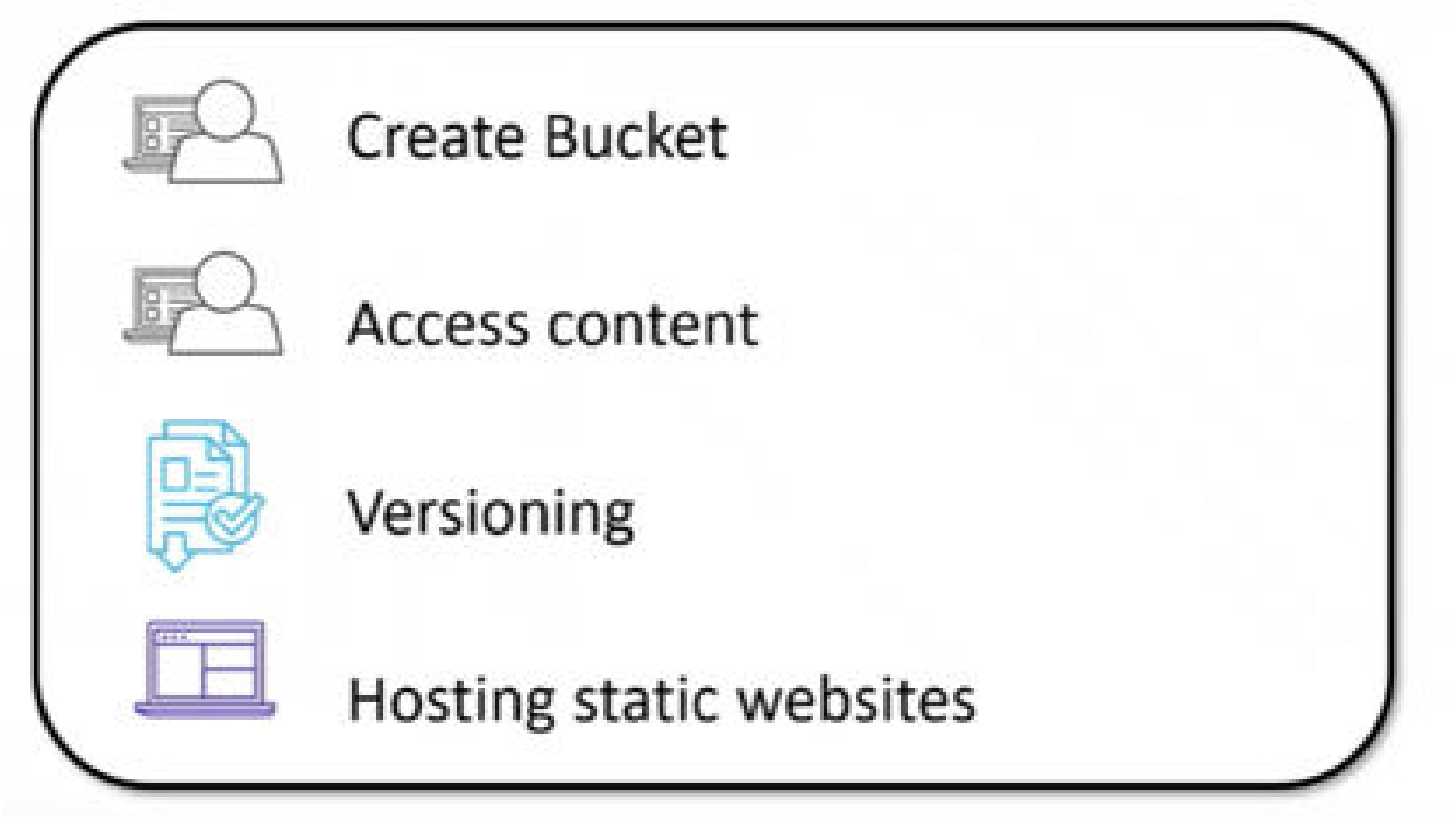


NOT JUSTA A STORAGE BUCKET



- Requester pays (requests and download traffic)
- Versioning
- Hosting static websites
- Object lifecycle management

S3 DEMO



Create Bucket

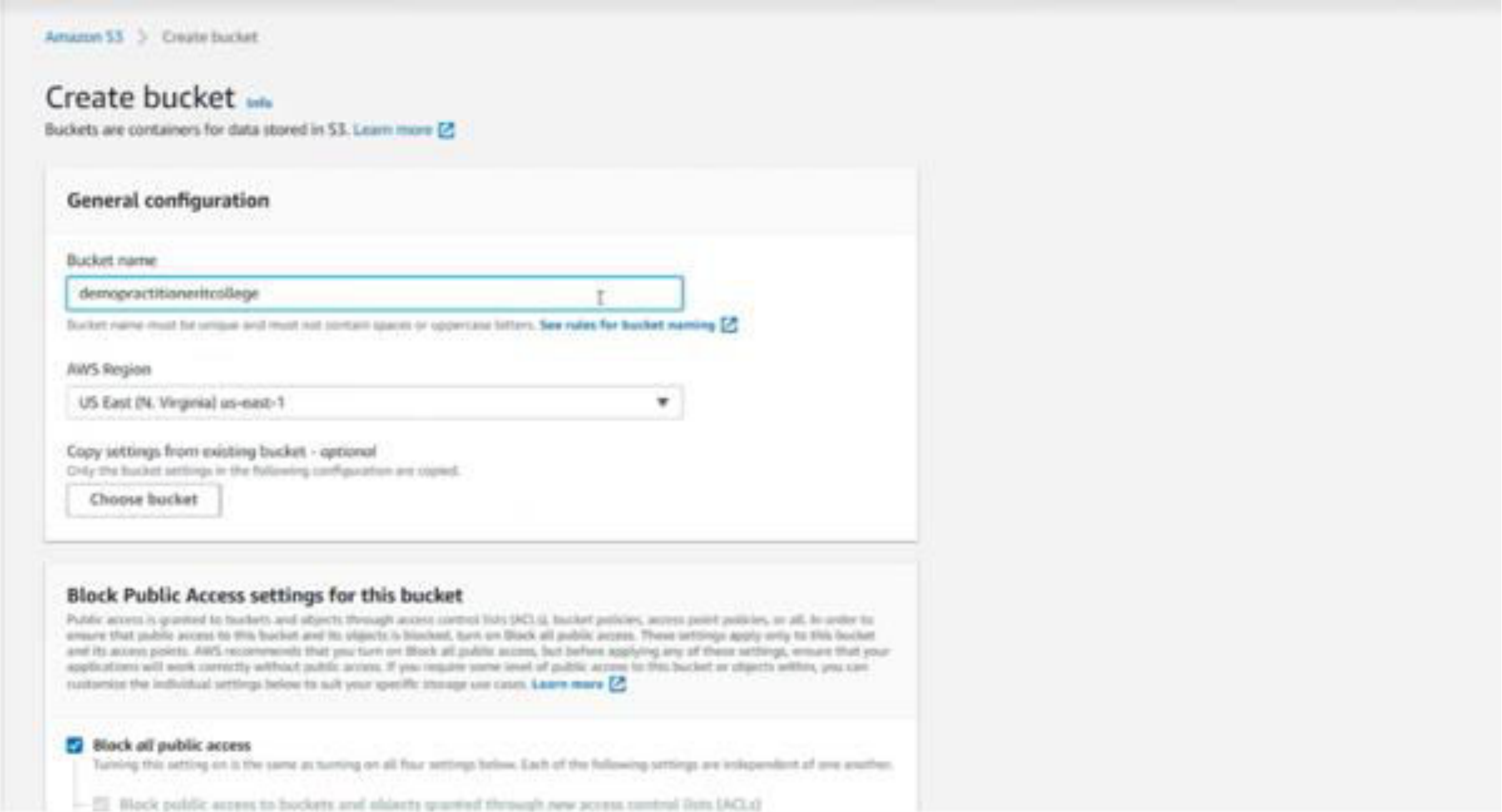
Access content

Versioning

Hosting static websites

Source: AWS

CREATE BUCKET



Amazon S3 > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name

demopractitioneritcollege

Bucket name must be unique and must not contain spaces or uppercase letters. See rules for bucket naming

AWS Region

US East (N. Virginia) us-east-1 ▼

Copy settings from existing bucket - optional

Only the bucket settings in the following configuration are copied.

Choose bucket

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

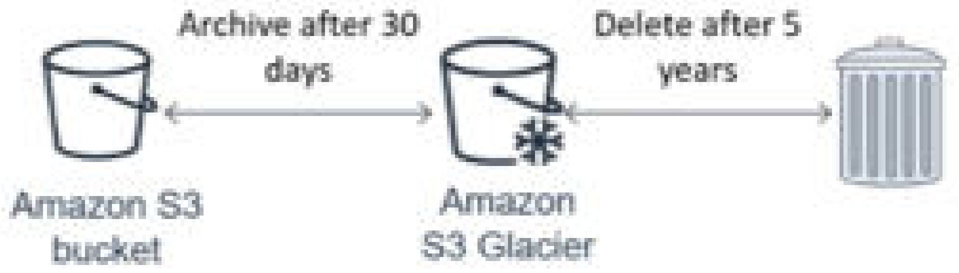
☑ **Block all public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

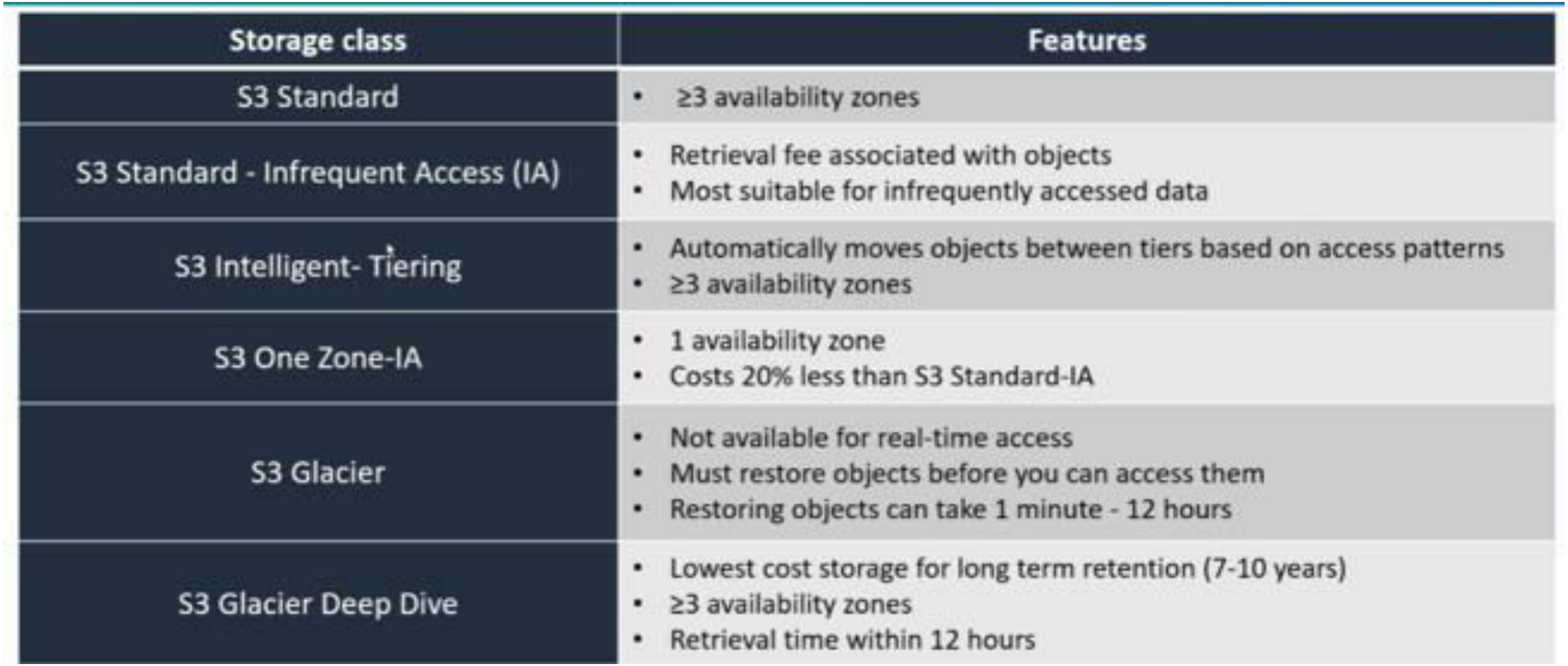
☐ Block public access to buckets and objects granted through new access control lists (ACLs)

WHAT IS AMAZON S3 GLACIER?

- Low-cost data archiving and long-term backup
- 3- to 5-hour or within 12 hours*
- Can configure lifecycle archiving of Amazon S3 content to Amazon Glacier

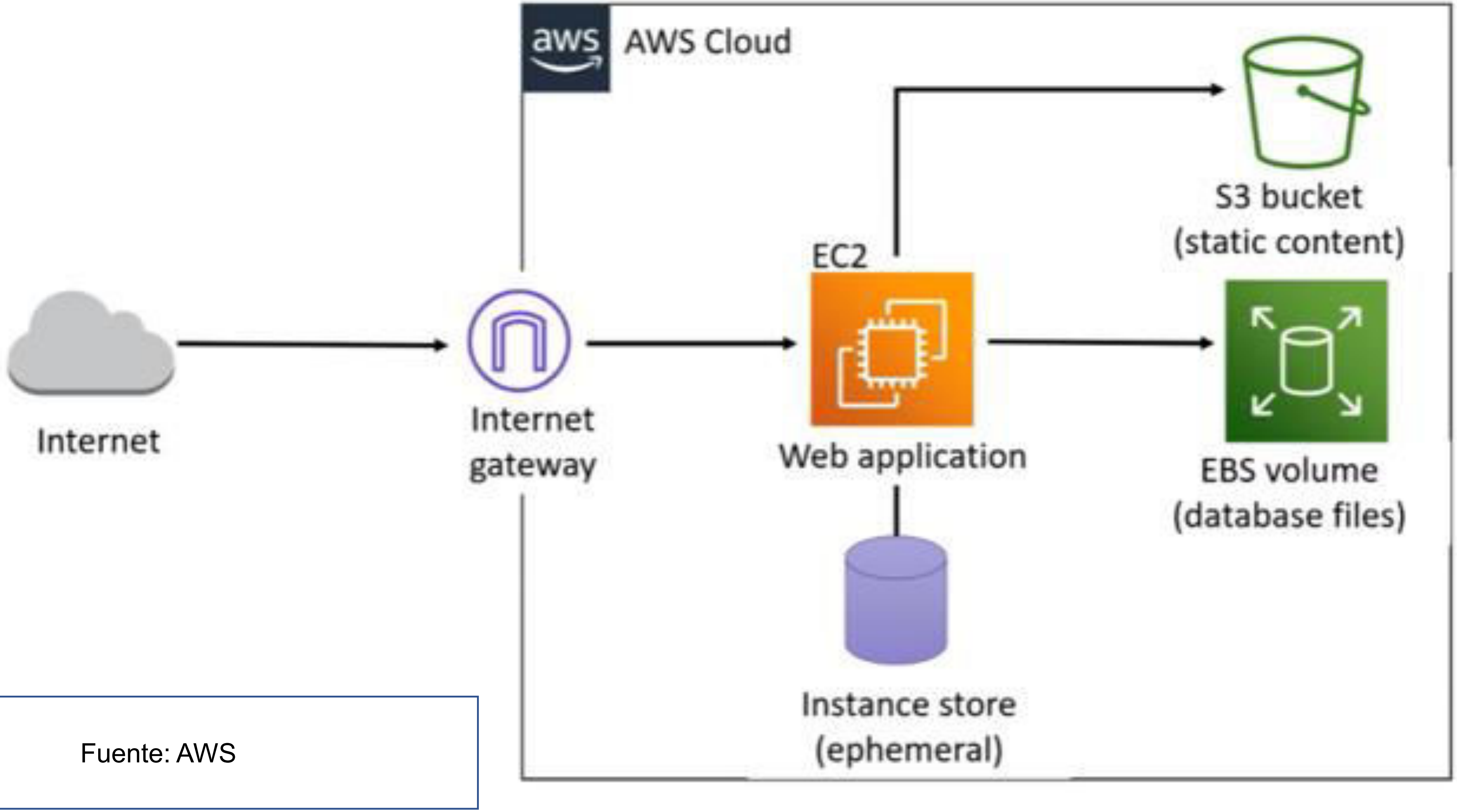


AMAZON S3 STORAGE CLASSES



Storage class	Features
S3 Standard	- ≥3 availability zones
S3 Standard - Infrequent Access (IA)	- Retrieval fee associated with objects - Most suitable for infrequently accessed data
S3 Intelligent- Tiering	- Automatically moves objects between tiers based on access patterns - ≥3 availability zones
S3 One Zone-IA	- 1 availability zone - Costs 20% less than S3 Standard-IA
S3 Glacier	- Not available for real-time access - Must restore objects before you can access them - Restoring objects can take 1 minute - 12 hours
S3 Glacier Deep Dive	- Lowest cost storage for long term retention (7-10 years) - ≥3 availability zones - Retrieval time within 12 hours

ARCHITECTURE EXAMPLE



Fuente: AWS

INTRODUCTION TO SDN

DATA CENTER FACILITIES

CePETel

Sindicato de los Profesionales
de las Telecomunicaciones

SECRETARÍA TÉCNICA

IPEI

FACILITIES

DataCenters are controlled environments where:

Critical processing resources are stored
Under centralized management

This allows the Company:

Operate your business requirements
And thus generate the following benefits:

1) Business continuity
2) Security
3) Consolidation of servers and applications
4) Storage Consolidation

Data Center design is essencial to ensure the infrastructure needed to deploy then NFVs
Compute, Storage and Networking are more than “just resources”

BUSINESS DRIVERS

Agilidad

Habilidad de moverse rápidamente.

Resistencia

Preparada para recuperarse rápidamente de una falla de equipo o de un desastre natural.

Modularidad y escalabilidad

Ampliación de la infraestructura rápida y fácil.

Confiabilidad y disponibilidad

Confiabilidad: Capacidad del equipo para realizar una función determinada
Disponibilidad: Capacidad de un elemento para estar en un estado como para poder realizar una función requerida.

Sostenibilidad

Aplicando las mejores prácticas de diseño ecológico, construcción y operaciones de DataCenters para reducir los impactos ambientales.

TCO

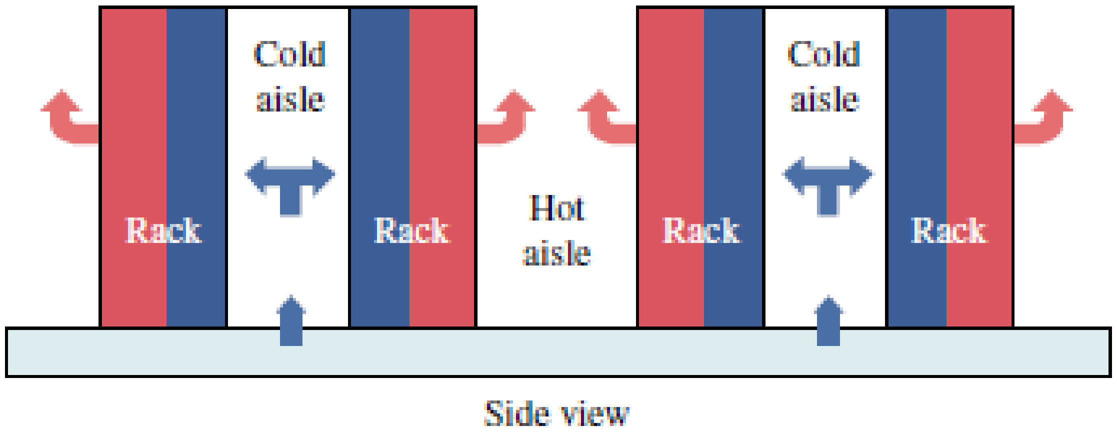
Total Cost ownership: Implica el Costo total del ciclo de vida del CAPEX (terreno, construcción, diseño ecológico y armado del DataCenter)
Y del OPEX (por ejemplo costos de energía)

DATA CENTERS COOLING SYSTEM

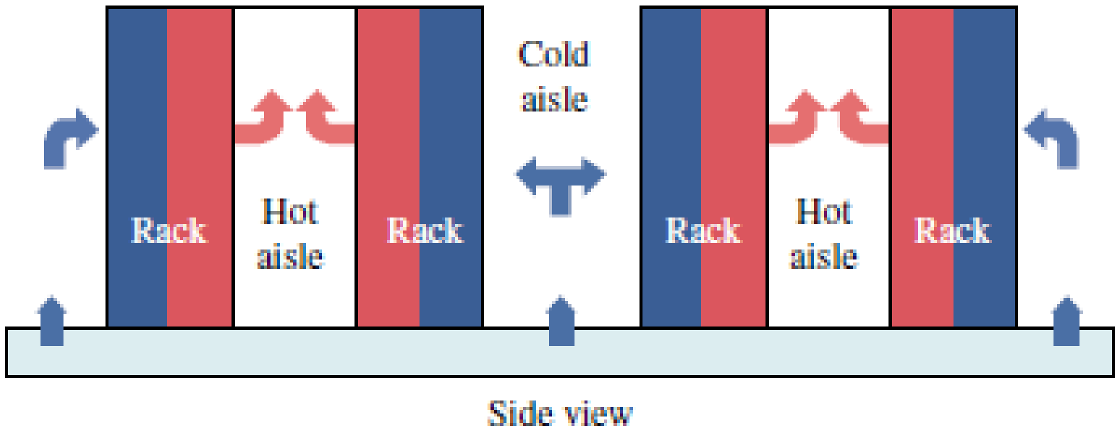
Hot and cold halls consist on the technique of accommodating the equipment racks of a data center and optimizing its cooling capacities. Among the benefits we find:

-Increase the capacity and efficiency of the cooling system, affecting the return air temperature
-Provide more predictable and reliable inlet air temperatures to IT equipment.
-Improve redundancy in the Row Based Cooling System by extending the sphere of influence for cooling units

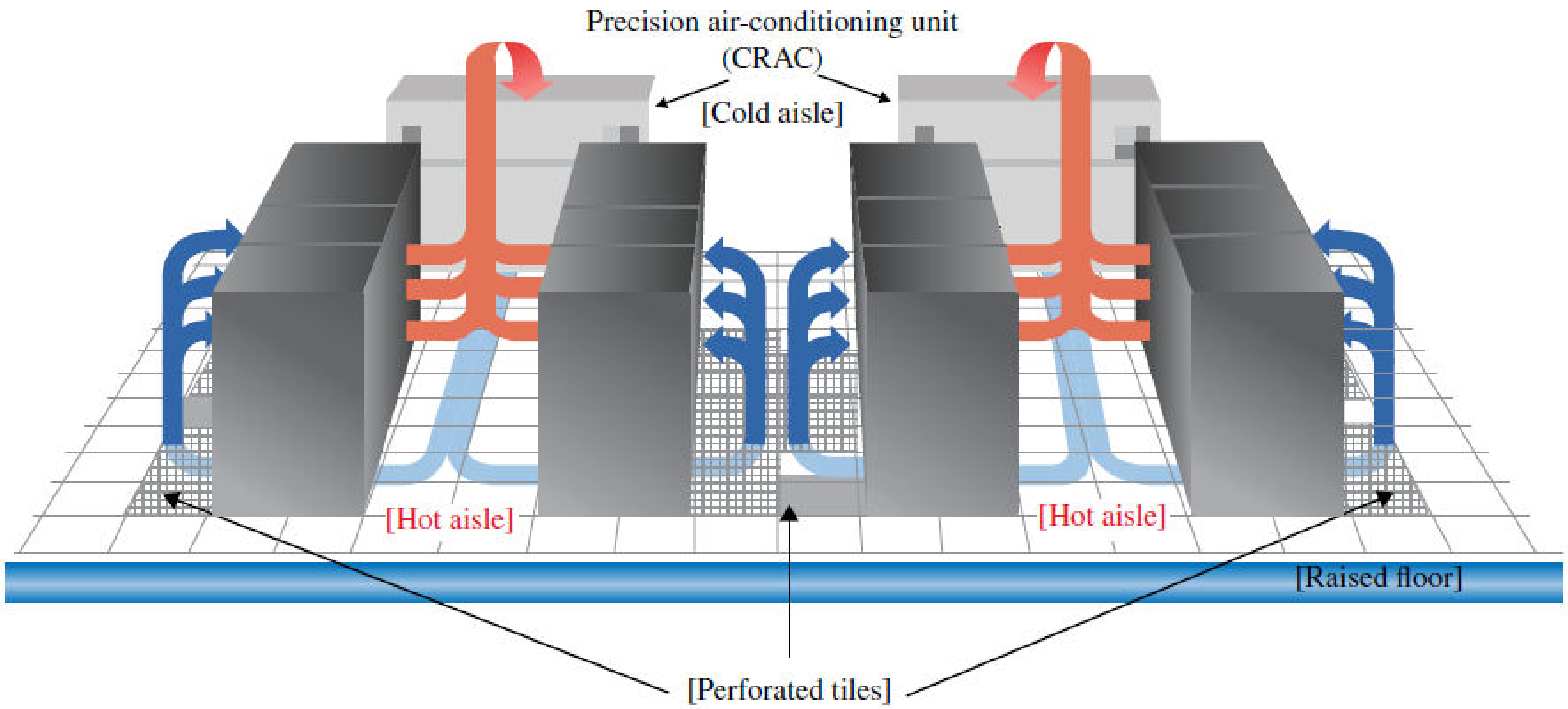
Cold aisle containment (CAC)



Hot aisle containment (HAC)



DATA CENTERS COOLING SYSTEM



CRAC: computer room air conditioning

DATA CENTERS COOLING SYSTEM- cold corridors and hot corridors

Basically the system injects cold air through the fronts of the rack.

The cold air passes through the equipment that expels it with a higher temperature.
The warm air weighs less, then it rises towards the ceiling.

There it is extracted and carried to the cooling room to return through the cold air ducts.

This system ensures that the teams always get cold air. This increases the efficiency of the refrigeration system.

In addition, there are methods to force and create watertight sectors where both corridors are much better isolated.

PUE: POWER USAGE EFECTIVENESS

PUE: Power Usage Effectiveness

It is a metric used to determine the energy efficiency of the data center

It represents the ratio of:

-power that is being consumed from the energy provider, and

-the amount actually consumed by the processing equipment stored in the datacenter

$$\text{PUE}= \frac{\text{power that is being consumed}}{\text{amount actually consumed by the processing equipment}}$$

PUE should be arround 1,6

DATA CENTERS TIERING

Datacenter standards evaluate the quality and reliability of the data center to accommodate equipment. The Uptime Institute uses a sort of Tiers-based ranking to determine trustworthiness. These 4 Tiers can be defined as follows:

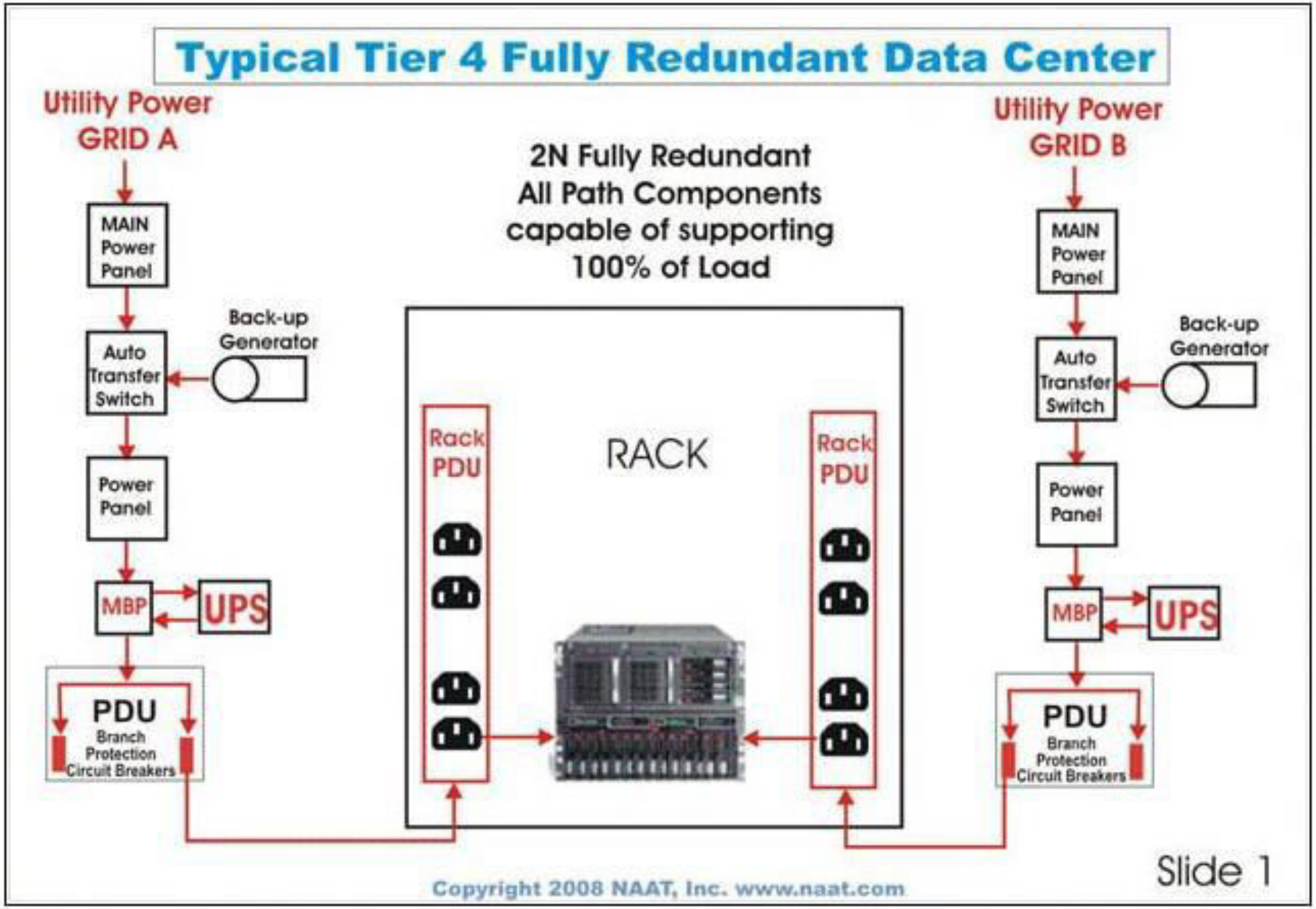
Tier I
99.671%Uptime, 28,8 hs downtime per year allowed
No redundancy

Tier II
99.749%Uptime, 22 hs downtime per year allowed
Partial redundancy in Energy feeding and cooling system

Tier III
99.982%Uptime, 1,6 hs downtime per year allowed
N+1 fault tolerant. Until 72 hs protection against energy supply interruption

Tier IV
99.995%Uptime, 1,6 hs downtime per year allowed
2N+1 in redundant infrastructure
Until 96 hs protection against energy supply interruption
26.3 minutes downtime per year allowed

DATA CENTERS TIERING



MBP: Maintenance bypass panel
UPS: Uninterruptible Power Supply

PDU: Power Distribution Unit

DATA CENTERS TIERING

N+1: Parallel redundancy

Configuration in which 2 UPSs support the load of the systems at the same time.

Each one being able to bear the entire load completely.

It is one of the most common configurations.

Requires UPSs to be synchronized, and usually they are from the same manufacturer.

Design has single points of failure.
Both in the supply of the UPSs.
As in the distribution towards the systems.

It is not fault tolerant.

This can be qualified according to the implementation in tier 2, 3 and 4 according to TIA-942

DATA CENTERS TIERING

2(N+1): Double Redundancy in parallel

Two parallel redundancy configurations , simultaneously powering critical equipment.

It requires at least Quadrupling the electrical power necessary to feed the computer systems, since each of the 4 minimum UPSs required, it has to be able to protect the entire load completely.

Requires 2 generators capable of independently supporting the entire load of the installation.

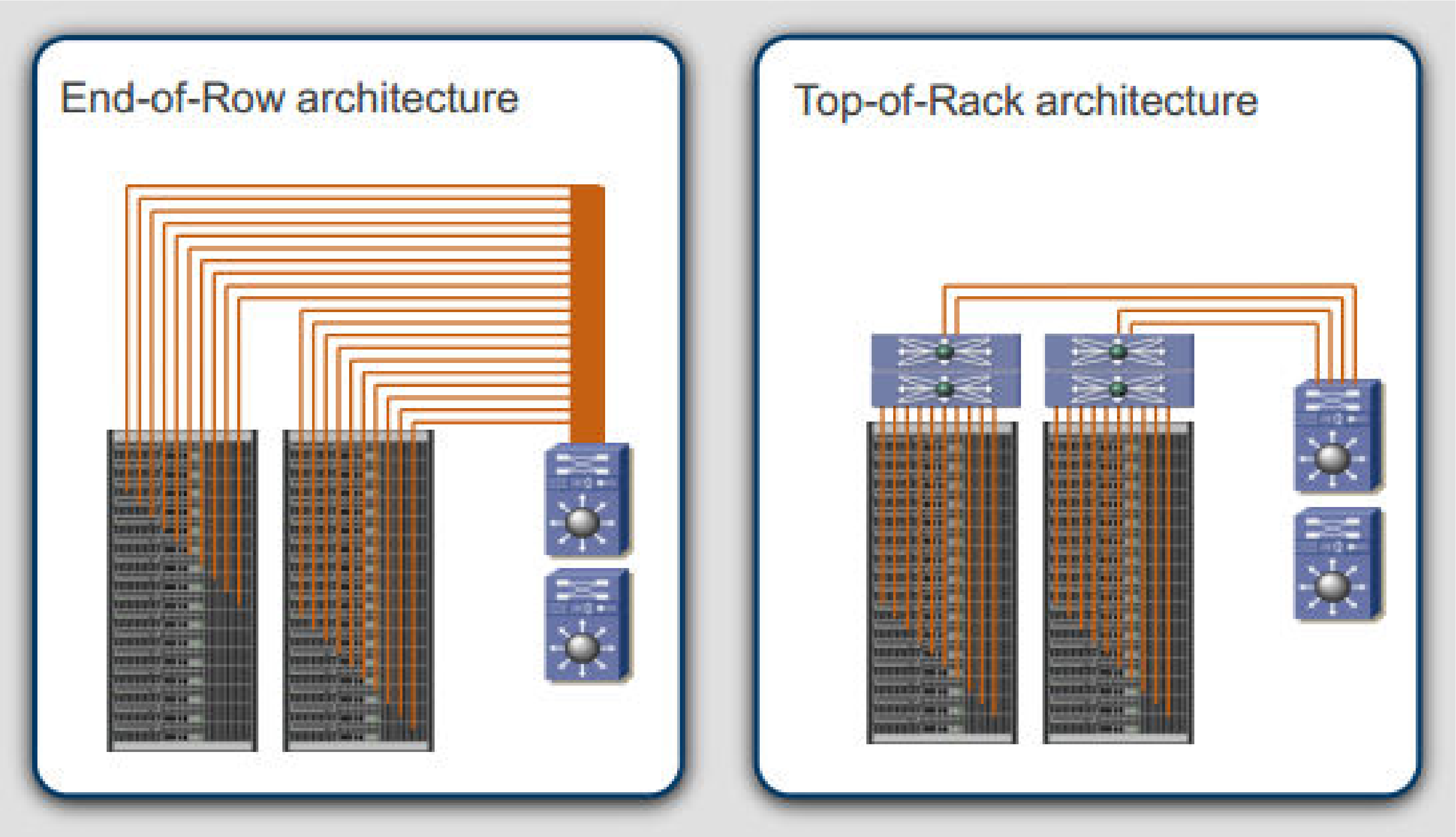
The entire system is fault tolerant.

Can be maintained without exposing systems to service interruptions.

Corresponds to TIER4 in the TIA-942, as long as 2 different power supplies are used.

ANSI/TIA-942-B es el estándar de referencia para certificar Centro de Datos.

EOR Vs TOR ARCHITECTURE



TOP OF RACK VS END OF ROW

- TOR (top of rack) and EOR (end of row) are two different network design concepts in the data center.
- You can implement one, the other, or a combination of both.

- In the TOR design, all the physical connectivity of the servers in a rack ends in a switch that is commonly installed in the highest part of the rack and then this is connected to an aggregation layer.

- In the two-tier EOR, the connectivity of the servers in a rack ends at a switch located in one position of the row of racks.

- In a combination, the TOR model could be used and end up in a switch like EOR design

Advantages and limitations of the TOR model

Simpler wiring
Less wires Copper
Cables inside the rack
Modular implementation

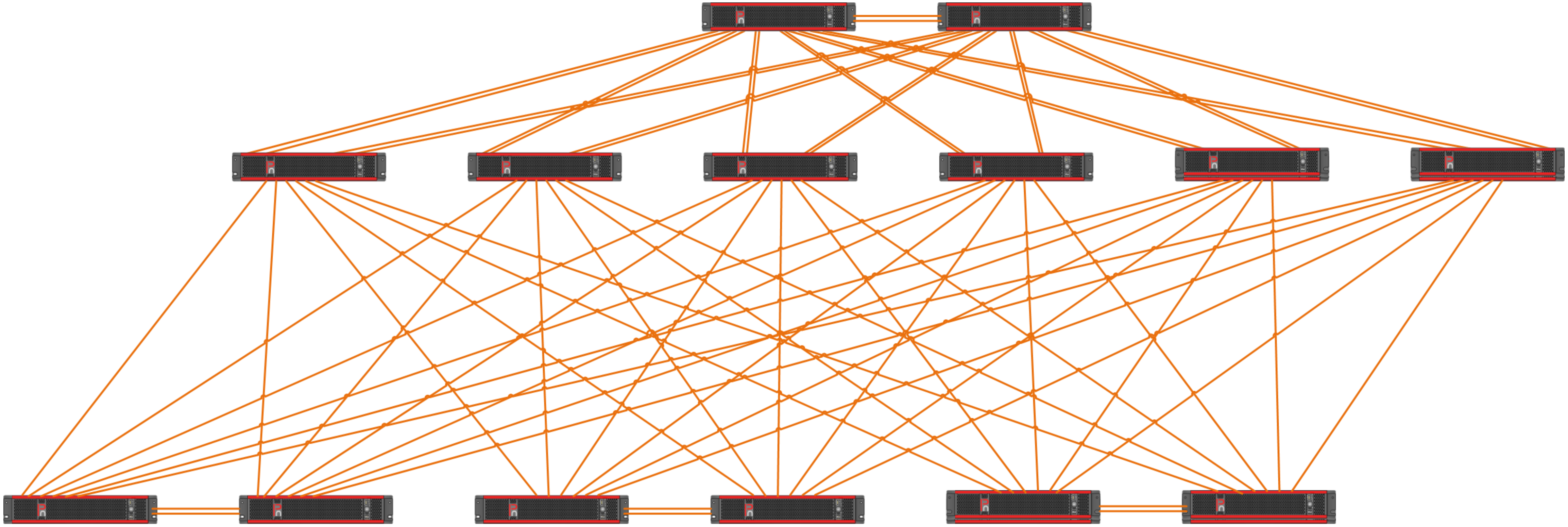
Difficult scalability
There may be many unused ports

Advantages and limitations of the EOR model

Big switches
Expansion is easier
High availability 1+1. N+1 or N+N
Separate server location
Number of switches/ports is less
Latency is reduced (fewer switches)
High traffic servers can connect to the same board

Longer cables cable cost
The upgrade (ie 1GE to 10 GE) of cables is more complex

LEAF AND SPINE IN DATA FABRIC



SERVERS

SERVERS

Servers are processing equipments which unlike a personal computer can include more processors, more memory, redundancy, etc

-CPU
-Memory
-NUMA

They can be classified as:
-Rackable
-Blade (they include their own cabinet, energy feed and connectivity)

SERVERS- CPU

A CPU or processor is an electronic circuit that responds to or processes instructions.

The 4 functions of the processor are:

- Capture
- Decode
- Execute
- To rewrite

There are technologies like Intel Hyperthreading that make it possible for a processor to act as 2 logical processors in which it can run 2 applications at the same time.

There are processors with various architectures, processors with several cores on the same chip, or even have a chipset to control peripherals.

One concept to keep in mind is the term “core”. They are like "CPUs" inside a tablet called a processor. Generally a physical CPU has one or more cores.

Currently there are processors that have from 4 to 28 cores in a physical chip.

SERVERS –MEMORY

They are a special element in a computer or a server, since they store information.

In general, there can be memories of the RAM or Cache type.

RAMs are fast memories in terms of read and write time in which the processor can read and store data.

Cache memories are generally very fast and are included within the CPU chip, so their access is much faster and with less latency.

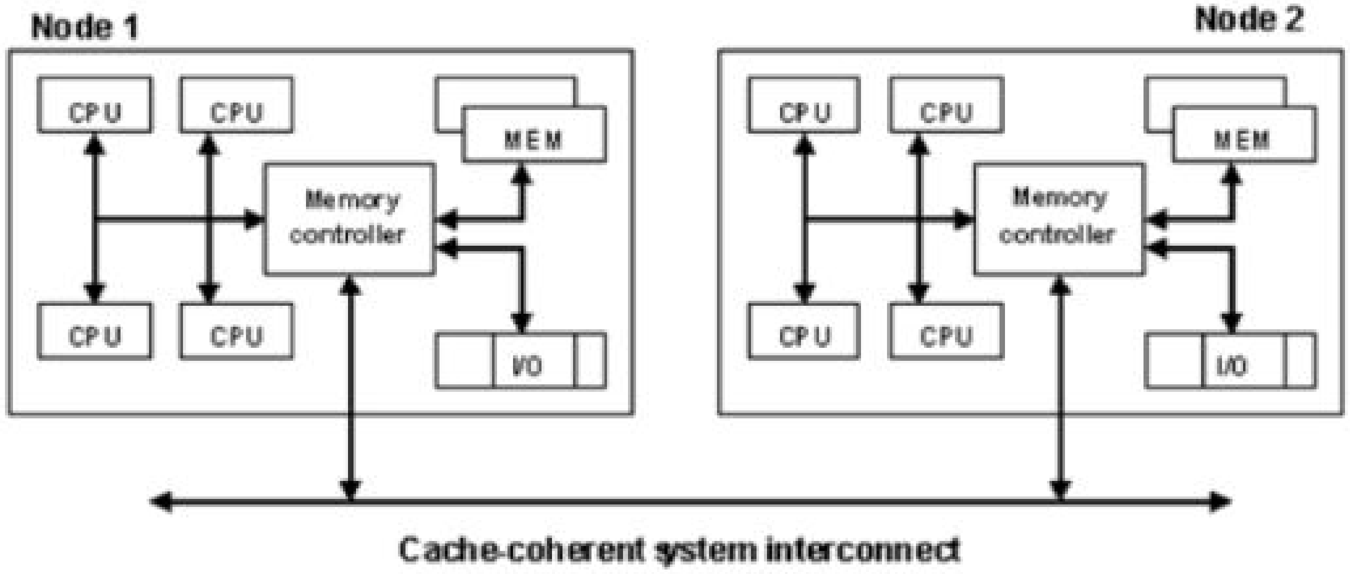
Cache memories typically come in 4; 8; 16 GBytes or more.

SERVERS – NUMA

Non Uniform Memory Access “NUMA” is a method of configuring in a cluster of microprocessors in a multiprocessing system local access to shared memory, improving performance, and making the system easier and more expandable.

Each cluster is called a NUMA Node.

In servers usually with several processors, each one of these has memory resources and I/O devices.



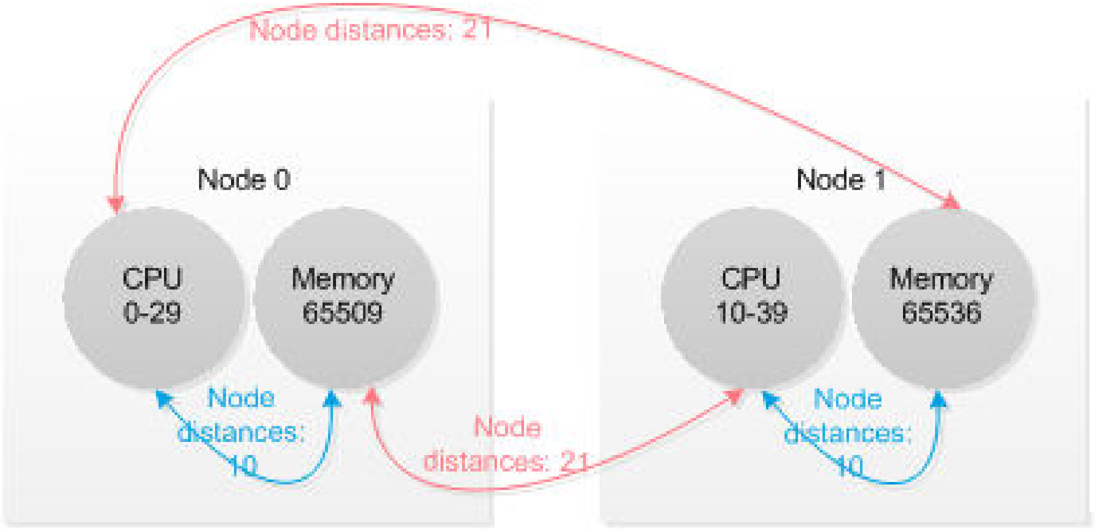
SERVERS – NUMA

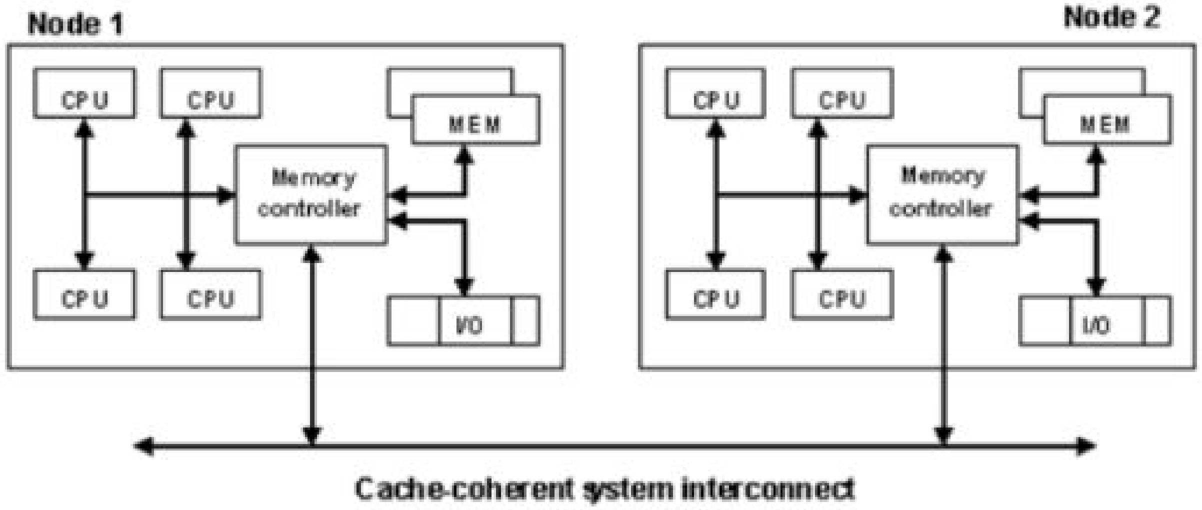
NUMA (non uniform access) is a method which consist on how memory access is configured in a cluster with many proccesors. Each cluster is called NUMA Node

In a server, NUMA nodes are interconnected by interconnect modules to exchange data.

If the CPU tries to access a remote memory (which is not in its “NUMA”), it will have to wait a while.

So NUMA performance is not linear as CPUs are added .A CPU's access to memory within its NUMA is much faster than memory access from a remote NUMA.





PERIPHERALS

Servers have different type of peripherals as:

NIC (Network Interface Card): allows interconnection to external word. 1Gb, 10Gb, 25 Gb, 40 Gb. Include interruption and DMA, tx/rx queues, logical partition, TCP offload

HBA (Host Bus Adapter): to connect server to storage devices.

- CNA (Converged Network Adapter) for:
  - FCoE (FC over Ethernet)
  - FC (Fiber channel)
  - iSCSI (SCSI) over IP

RAID Controller: handle hard disk in the server. They can handle RAID levels

STORAGE

STORAGE

- interface protocols
- Disks and storage
- RAID
- High End Storage
- Medium Range Storage

STORAGE: Interface protocols

Storage interface protocols allow communication between the node or host and the storage.

Interface protocols are implemented using interfaces or controllers, both at origin and destination.

Interface protocols:

SATA
NEARLINE SAS
SCSI SAS
FIBER CHANNEL
IP
FCoE

- SATA: Serial Advanced Technology Attachment: es una interfaz de transferencia de datos en serie entre la placa base y algunos dispositivos de almacenamiento
- SAS_ Serial Attached SCSI: es una interfaz de transferencia sepi de transferencia de datos en serie, sucesor del Small Computer System Interface (SCSI) paralelo, aunque sigue utilizando comandos SCSI para interaccionar con los dispositivos SAS.

The formal distinction between online, nearline, and offline storage is:

- Online storage is immediately available for input/output (I/O).
- Nearline storage is not immediately available, but can be made online quickly without human intervention.
- Offline storage is not immediately available, and requires some human intervention to become online.

STORAGE

Interface protocols: allows to interconnect between the host and the storage

- **SATA:** it is similar to old IDE/ATA. High performance and low cost. 6GBps (rev 3)
- **NERALINE SAS:** it is a disk SATA optimization
- **SCSI-SAS:** parallel data transmition. Expensive, not used in personal equipments. SAS, a serial 6 Gbps protocol is a derivation of it. SAS is compatible with disk SATA.
- **FIBRE CHANNEL:** serial protocol cable/fiber, widely used. 16 Gbps
- **IP:** widely used to interconnect hosts. Also used for storage. Low cost. iSCSI (SCSI over IP) is widely used.
- **FCoE:** As FC is widely used, FCoE (over ethernet) was developed. Its configuration, however is not as simple as iSCI.

STORAGE: Interface protocols. SATA & NEARLINE SAS

SATA interface protocols:

It is the serial version of the old IDE/ATA protocol.
High performance.
Low cost.
Revision 3 of this SATA protocol allows reaching 6 Gbit/s.

NEARLINE SAS interface protocols:

It is a firmware optimization on SATA drives.

STORAGE: Interface protocols. SCSI & SAS

SCSI interface protocols:

SCSI emerged as one of the preferred protocols for the server environment, it uses much improved parallel data transmission (compared to the old ATA):

The performance
Scalability
Compatibility

But it has the disadvantage:

The cost, which limited its popularity on PCs

However:

Over the years this protocol was improved
An alternative called SAS was born

SAS interface protocols:

SAS is a serial transfer protocol up to 6Gbit/s
SAS controllers are compatible with SATA drives (since they share the same cable format and connection)

STORAGE: Interface protocols. Fiber Channel & IP

Fiber Channel interface protocols:

Fiber Channel is a widely used protocol for communications with high-speed storage devices.

It is a serial transmission protocol

Operates with copper or optical cabling

The latest version of this protocol is 16Gbit/s

IP interface protocols:

The IP protocol is generally used for data transfer between hosts.

But with the improvements in the link layers, they made this protocol an alternative for accessing storage devices.

Advantages:

Cost

Maturity

Companies can amortize the cost with existing infrastructure

There are two protocols:

iSCSI is currently the most widely used due to its easy implementation

FCIP iSCSI

iSCSI:SCSI over IP

STORAGE: Interface protocols. Fiber Channel & IP

FCoE interface protocols:

Fiber Channel is a protocol widely used in data center networks.

An attempt was made to unify Fiber Channel with Ethernet, which is a very mature protocol.

FCoE is a modification of Ethernet in which adapters called CNAs are used.

CNA adapters connect to switches that support that protocol.

Disadvantage:

Although it is proposed as the successor to FC, its configuration is not easy

STORAGE: DISKS

Mechanical Discs:

They are data storage devices. They may be:

- Mechanics
- or of memories

Data can be accessed through interface protocols

Characteristics:

- Capacity
- Interface protocol
- Rotation speed (only on mechanical discs)
    - 5400 (SATA)
    - 7200 (SATA)
    - 10K (SAS/FC)
    - 15K (SAS/FC)
    - It influences the so-called seek time: Time it takes for the R/W head to position itself across the plate with a radial movement.
    - Rotational time: It is the time that when the R/W head is positioned on the track it takes to read the selected sector.
- Format

DISKS AND STORAGE

SSD drives:
They are data storage devices in flash memories
They are much faster.
There are no seek and rotational times.
There are several SSD technologies
Some SSD technologies are based on the Enterprise or Consumer Market
The difference is the duration time, since they have a more limited life time.

DISKS AND STORAGE

Storage:

Storage is the component of the datacenter where the data is stored:

- They use disks to carry out this task
- Difference between storage and disks:
- The storage provides protection, more speed and other characteristics that the disk does not have.

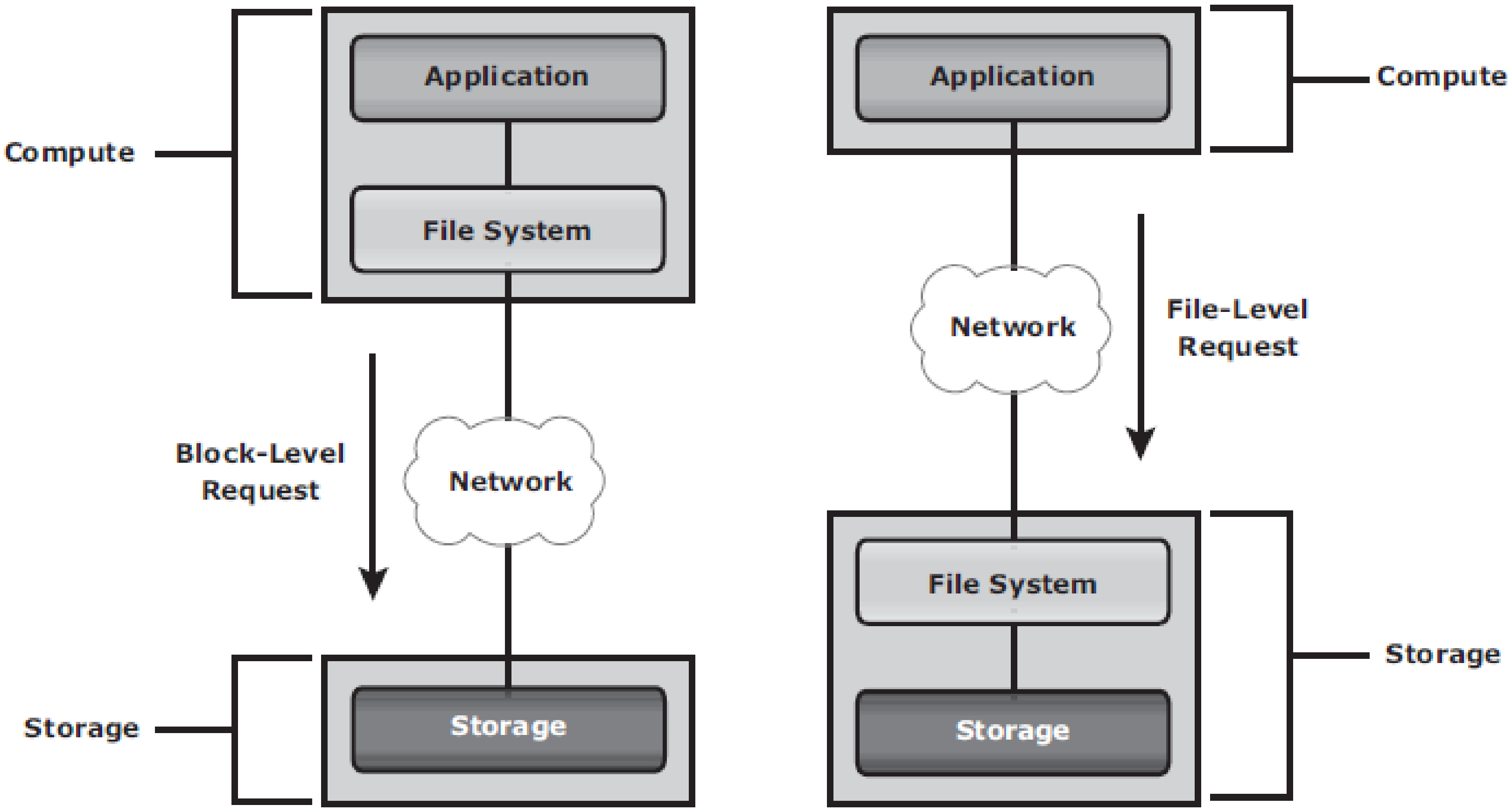
Storage can:

- Concatenate multiple disks
- Use disks in parallel
- Add a memory cache
- Can provide host connection redundancy

Storage can be accessed in two ways:

- Blocks
- Files

Different ways to Access to Storage



(a) Block-Level Access

(b) File-Level Access

Different ways to Access to Storage

Ways to access storage:
The way to access the storage (Block or File) defines the type of access network.

Block level Access:
It is called SAN; With FC or FCoE Access protocol
File level Access:
It is called NAS; With NFS or CIFS Access protocol

There is a particular case for iSCSI:
It has access at the Blocks level but it is taken as NAS

DISKS AND STORAGE: RAID

RAID is a technology that takes advantage of multiple drives as part of the array that provides data protection against drive failure.

RAID: Redundant Array of Independent Disks.

RAID implementations generally improve storage system performance by serving I/O from multiple disks simultaneously.

Modern arrays with flash drives also benefit in terms of protection and performance by using RAID.

RAID can be implemented by hardware or by software.

Hardware RAID implementations provide the most performance.

RAID: redundant array of independent disks

RAID

Raid is a technology that uses multiple disks as a part of bigger set and offers Data protection against fails in Disk Units.

RAID, which can be implemented based on HW or SW, also improves the performance of storage system.

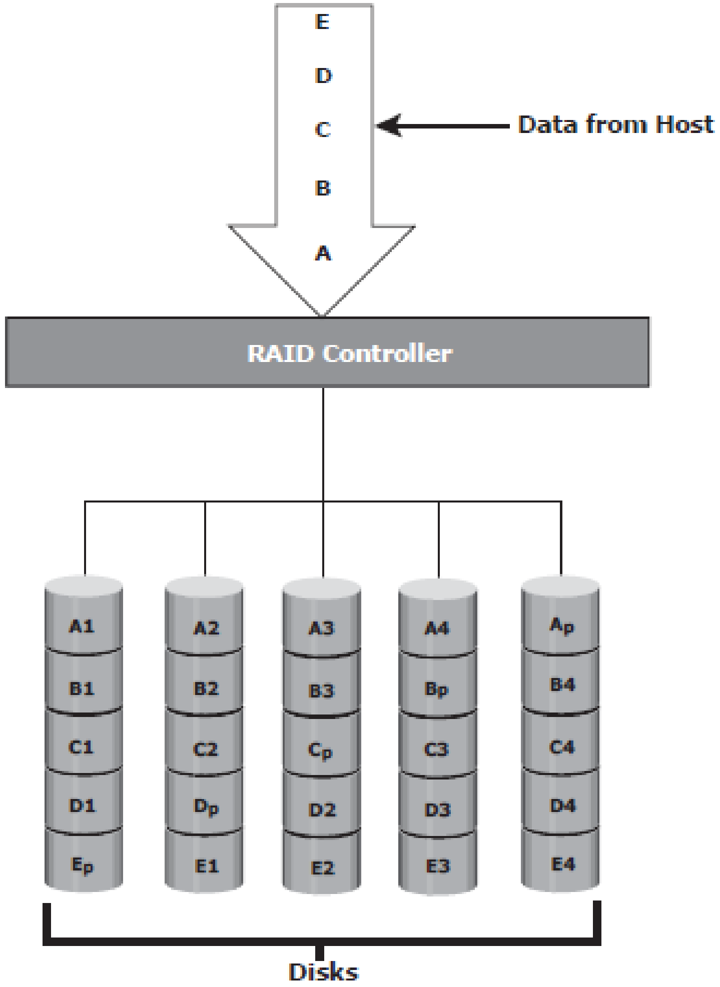
RAID 0. Concatenates Data using all disks. High capacity, but not protection.

RAID 1. Mirrored Data. Data must be written in all disks firstly. High availability. High cost.

NESTED RAID: RAID 1 + 0 , or RAID 0 +1. This allow performance, capacity and security.

RAID 5: used mainly for random reading. Uses three disks. For each Data, it generates a parity (XOR) which is storage in diff disks.

RAID 6: It generates two parity (XOR). Uses 4 disks.



DISK AND STORAGE

DISK

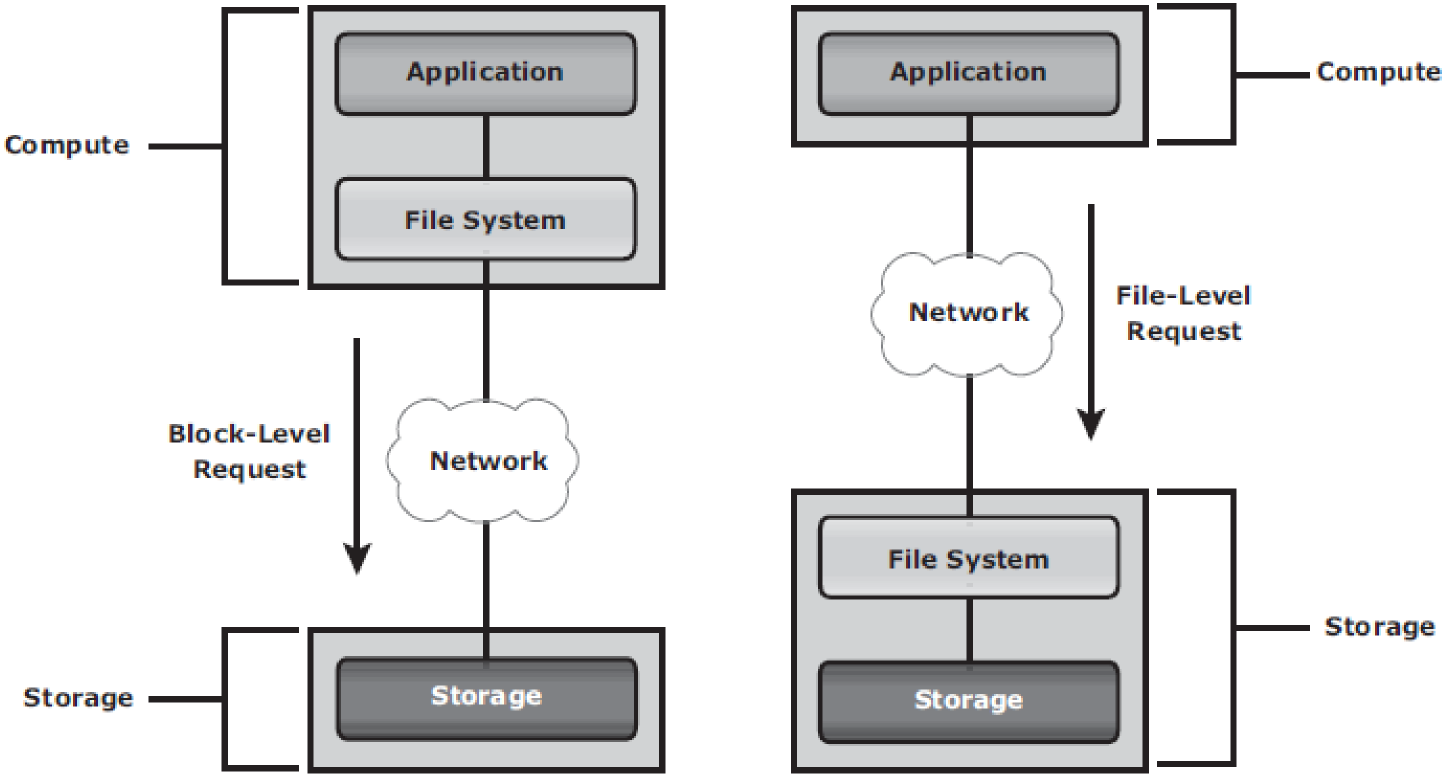
Mechanic or flash memory type. Capacity, protocol interface, speed (only in mechanics), format
In mechanic type disks, rotation speed impacts “seek time” (time required by R/W head to position itself on the track through a radial movement.
SSD type (flash memory) are fasters

STORAGE

It is the Component in Compute Centre where Data are stored.

They use disk, but include other additional components as security, speed improvement, etc.

In general, storage can concatenate several disks, access in parallel, add memory cache, include redundancy. They can be accessed as a block or as files



(a) Block-Level Access (b) File-Level Access

DISKS AND STORAGE: RAID

RAID 0

Use data concatenation techniques.

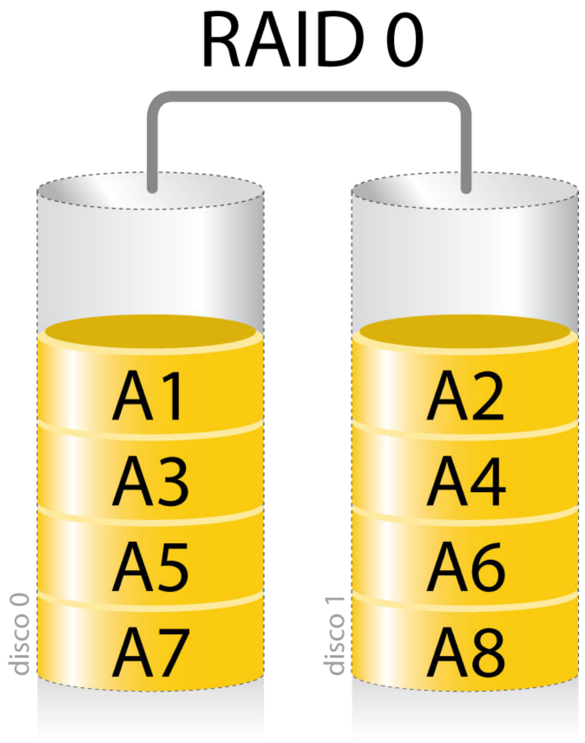
Concatenated data using all disks within a RAID set.

The total RAID capacity is the sum of the capacity of all the disks.

It has a high performance (And this increases with more disks because it can read and write different disks at the same time.

This type of RAID does not provide protection.

The failure of a RAID disk can corrupt the same.



DISKS AND STORAGE: RAID

RAID 1

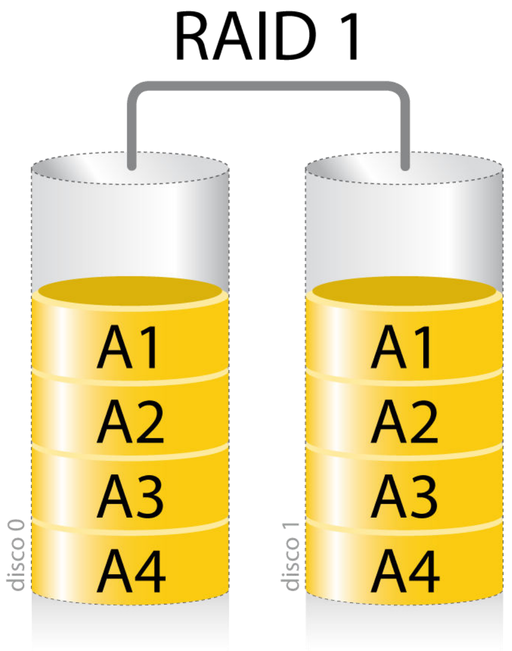
This type of RAID is based on data mirroring

Data that needs to be written to the RAID must first be written to the disks that make it up.

The capacity of the RAID will be equal to that of the disk with the smallest capacity.

That is why identical discs are generally used.

This RAID is ideal for applications that need high availability regardless of cost



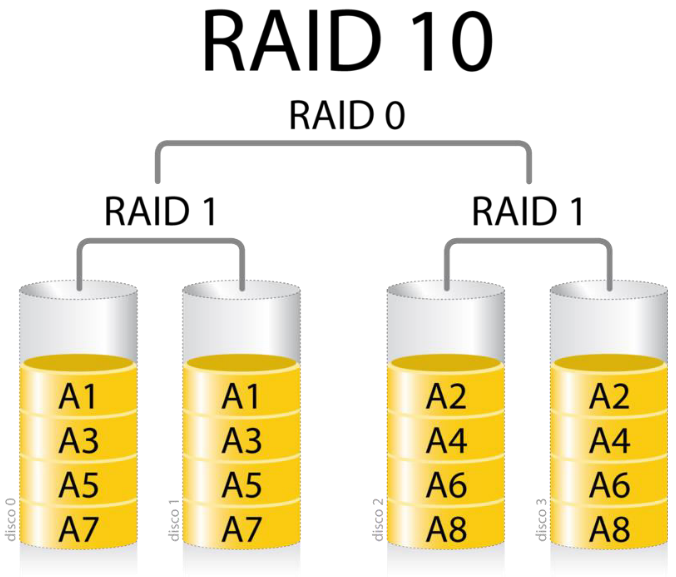
DISKS AND STORAGE: RAID

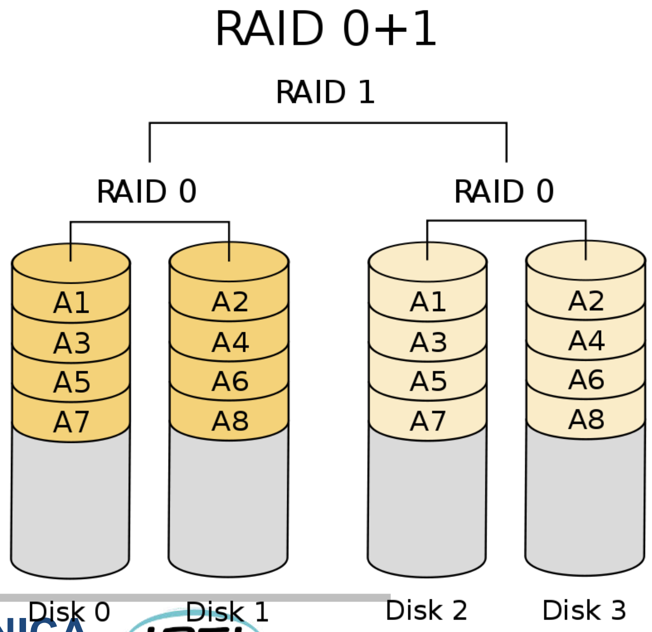
NESTED RAID

Sometimes redundancy and performance of a RAID and large amounts of storage are required.

One way to achieve that goal is to apply nested RAID.

A RAID1+0 (also called RAID 10) or a RAID0+1 is used





DISKS AND STORAGE: RAID

Under normal conditions both types of RAID10 and RAID 0+1 offer the same type of benefits.

The differences are noticeable when performing the recovery of a failed disk:

In RAID 10 only the mirror is recovered

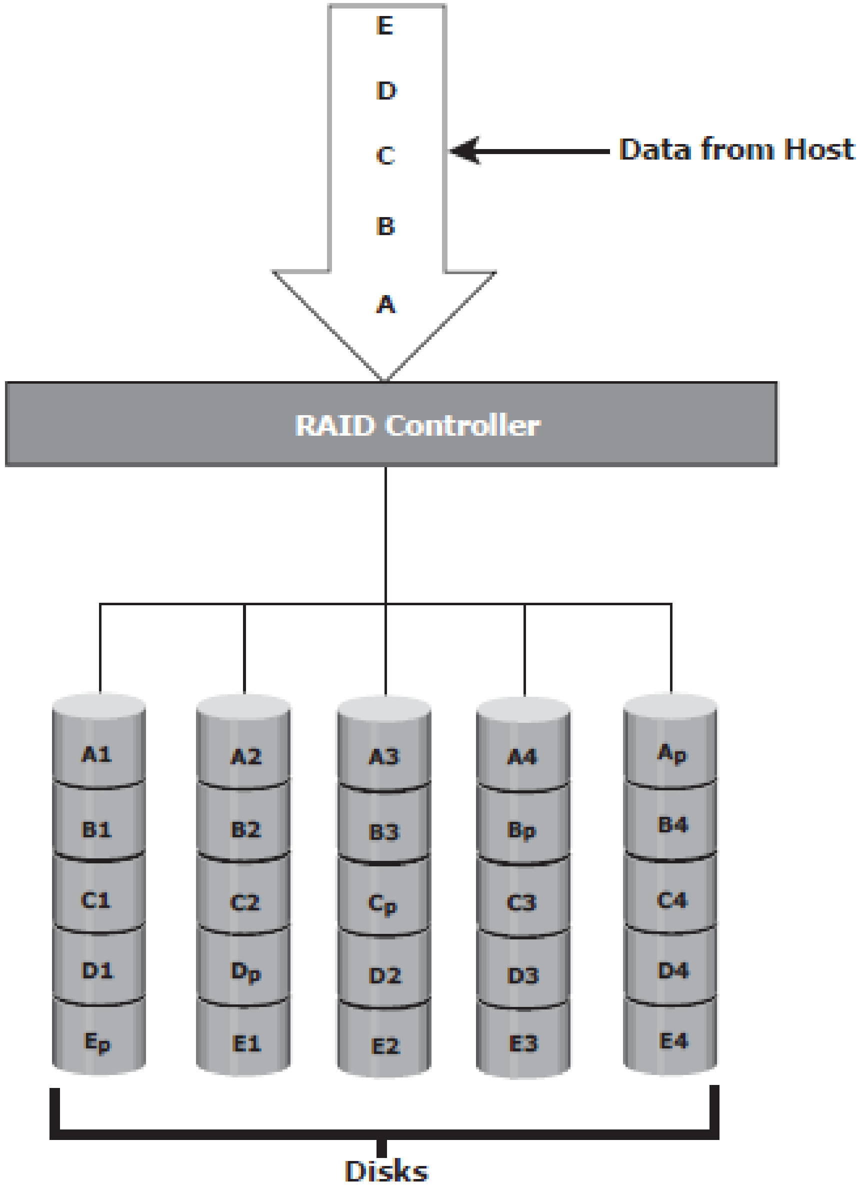
In RAID 0+1 the entire concatenation is recovered, this generates an unnecessary increase in the number of surviving disks and makes the RAID more vulnerable to the failure of a second disk.

DISKS AND STORAGE: RAID

RAID 5

This type of RAID is widely used in applications of intensive random reading in general.

It requires a minimum of 3 disks.
What it generates from each incoming data is the parity (XOR function) and it rotates the disks.
There are RAID not widely used that implement a disk for parity. (That's RAID 3 if you stripe by bytes or RAID 4 if divided by blocks)



DISKS AND STORAGE: RAID

RAID 6

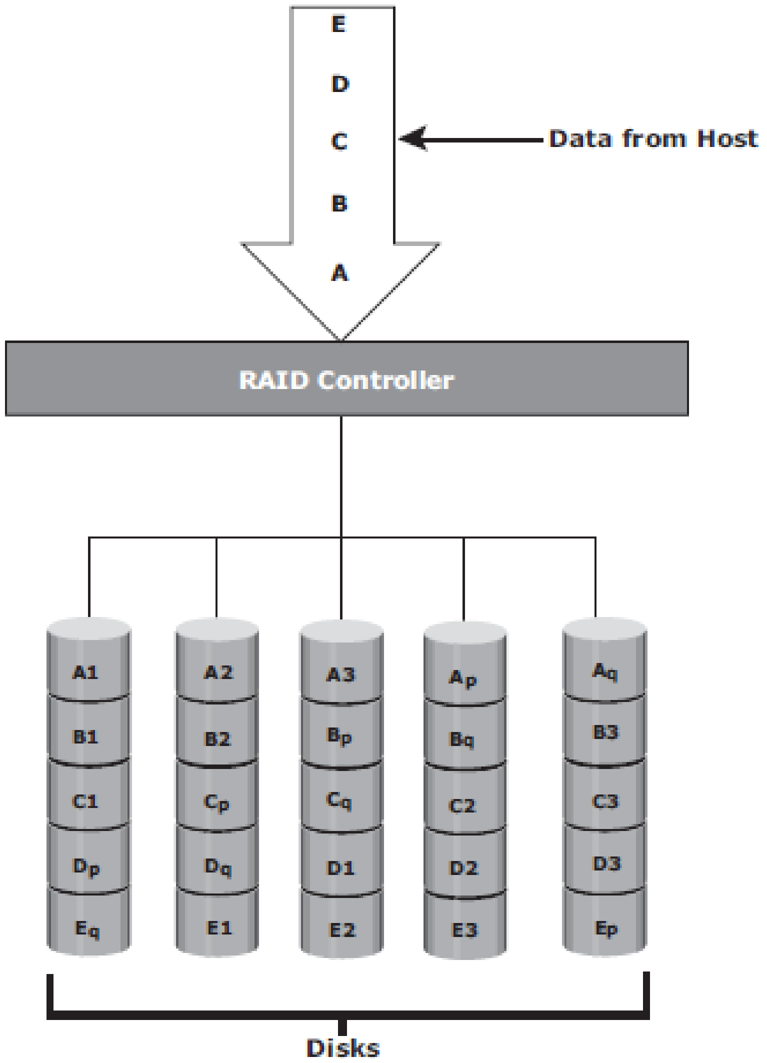
Because the disks are getting bigger and bigger in capacity.

In the event of a RAID5 disk failure:

It would take a long time to retrieve it.
And generally this process increases the disk activity.
Exposing the RAID to a broken second disc.

To mitigate this problem:

RAID6 was created, which generates two parities
and it requires a minimum of 4 discs.

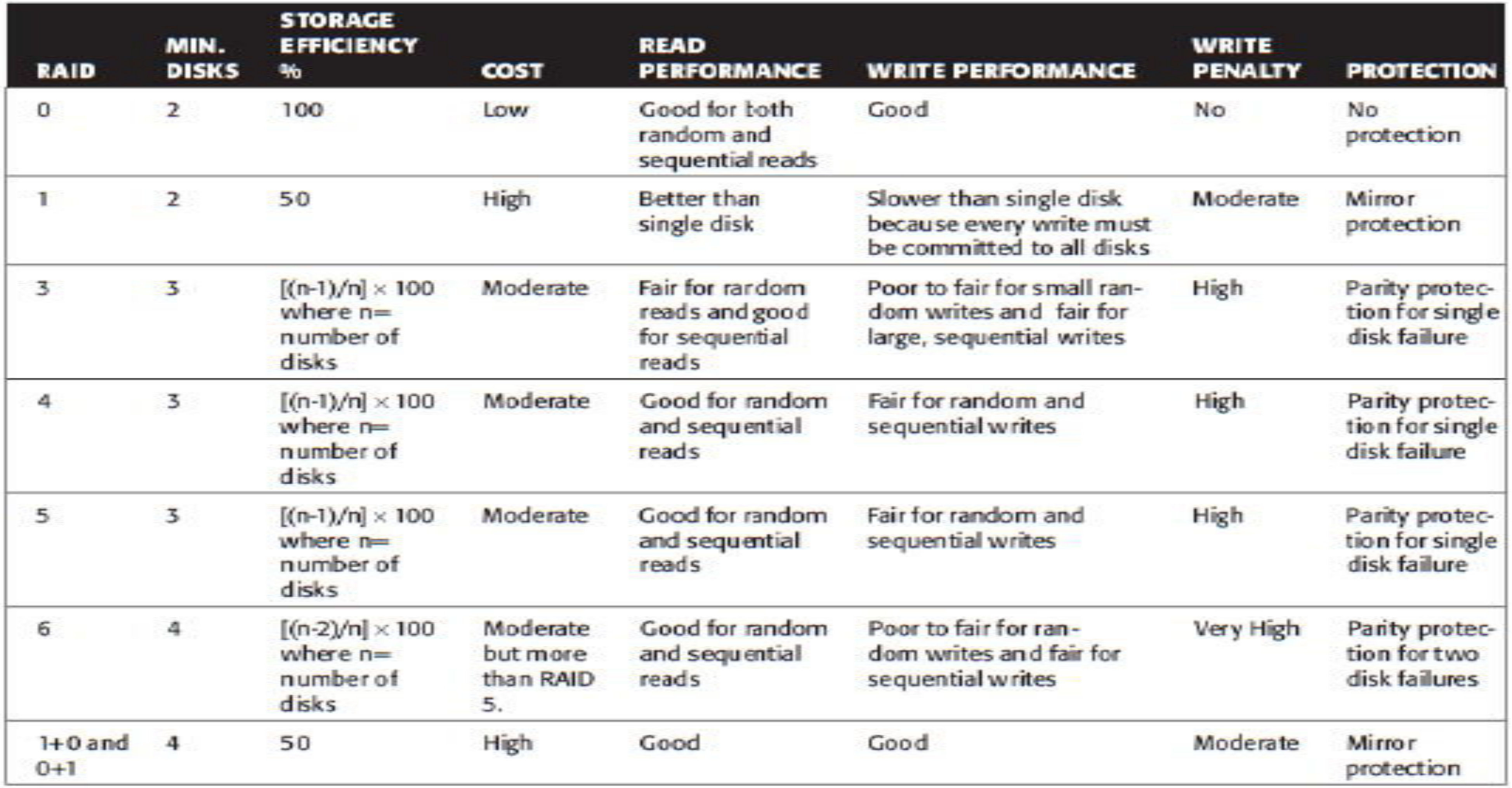


DISKS AND STORAGE: RAID

Hot Spare

It is a spare part which can be used transitorily any time in replacement of damaged disk. When a disk is damaged, all data start to be uploaded to a hot spare

Comparison between RAIDs



RAID	MIN. DISKS	STORAGE EFFICIENCY %	COST	READ PERFORMANCE	WRITE PERFORMANCE	WRITE PENALTY	PROTECTION
0	2	100	Low	Good for both random and sequential reads	Good	No	No protection
1	2	50	High	Better than single disk	Slower than single disk because every write must be committed to all disks	Moderate	Mirror protection
3	3	[(n-1)/n] × 100 where n= number of disks	Moderate	Fair for random reads and good for sequential reads	Poor to fair for small random writes and fair for large, sequential writes	High	Parity protection for single disk failure
4	3	[(n-1)/n] × 100 where n= number of disks	Moderate	Good for random and sequential reads	Fair for random and sequential writes	High	Parity protection for single disk failure
5	3	[(n-1)/n] × 100 where n= number of disks	Moderate	Good for random and sequential reads	Fair for random and sequential writes	High	Parity protection for single disk failure
6	4	[(n-2)/n] × 100 where n= number of disks	Moderate but more than RAID 5.	Good for random and sequential reads	Poor to fair for random writes and fair for sequential writes	Very High	Parity protection for two disk failures
1+0 and 0+1	4	50	High	Good	Good	Moderate	Mirror protection

DISKS AND STORAGE: RAID

HOT Spare

A disk listed as HOT Spare is a disk that is not used

Temporarily replaces or not a broken disk.

As soon as a RAID disk breaks, data begins to be rebuilt on a HOT Spare disk.

This minimizes the time between failure and replacement of the failed drive.

Before changing the disk, it could remain in HOT Spare again or in the other hand the new disk fulfills this function.

HIGH END STORAGE

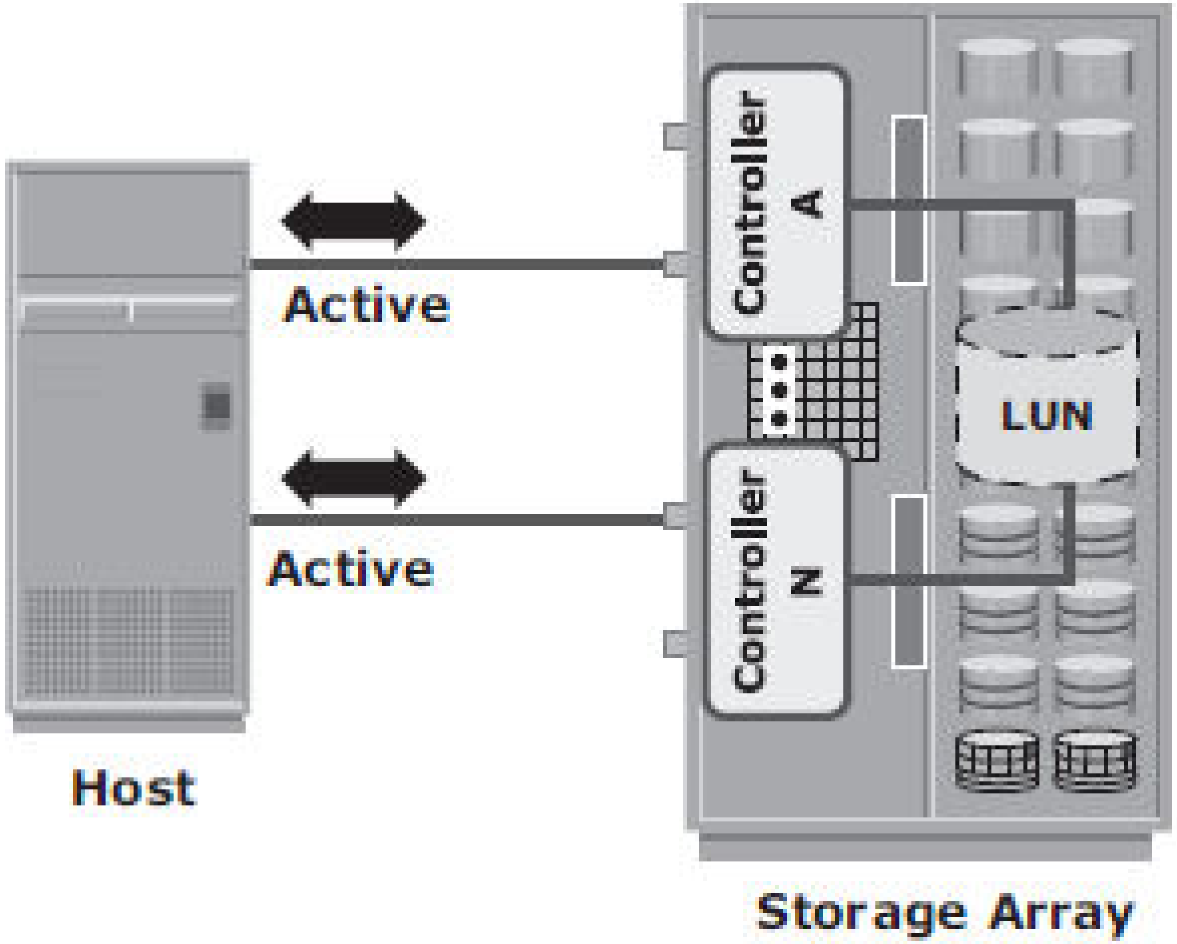
A Highend Storage is a system which:

- Works in active-active mode with N controllers.
- Has great Cache.
- Has disk access matrix.

As it works in active-active mode, the host can send Data to any of controllers available.

In general, these are High Availability equipements which allows some part of them can be swapped or changed without outage.

LUN is Logical Unit Number



MIDRANGE STORAGE

A Midrange Storage is a system which:

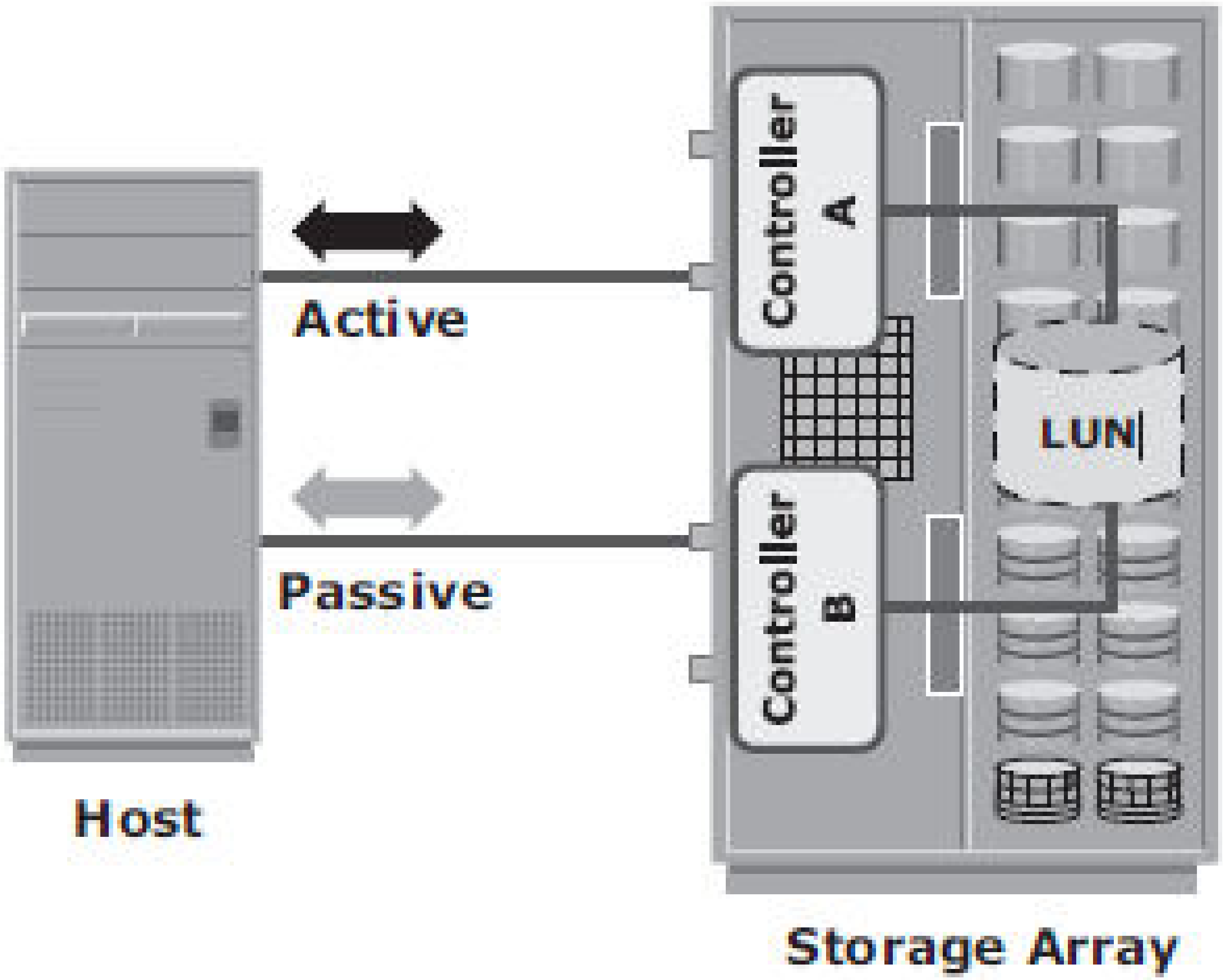
Works in active-pasive mode.

With 2 controllers (one active, one passive).
Are cheaper than Highend Storage.

.

The host must read and write over active controller.

They don’t have internal disk access matrix.



BACKUP

BACKUP

A backup is an additional copy of the production data

Created and retained solely for the purpose of recovery in the event of a lost or data corruption.

Backup window:

It is the period during which the backup source is available to perform the backup.

Sometimes making a backup requires:

That operations at the source are suspended so that the data is consistent

Or maybe the backup doesn't affect performance.

BACKUP GRANULARITY

Full backup:

It is the complete support of production volumes

It is done by copying the data to the backup storage devices

Provides fast data recovery.

But:

- Requires a lot of storage
- Takes a long time to perform

Incremental backup:

It is the backup of the data that changed from:

- the last full backup
- or the last incremental (the last one that has been performed recently)

It is much faster (Since it only copies the data that changed) than a full backup

But: Takes longer to recover

Differential or cumulative backup:

This type of backup always backs up the data since the last full backup

It takes more time than an incremental backup but the restore is much faster

BACKUP GRANULARITY

Synthetic Full backup:

It is a full backup

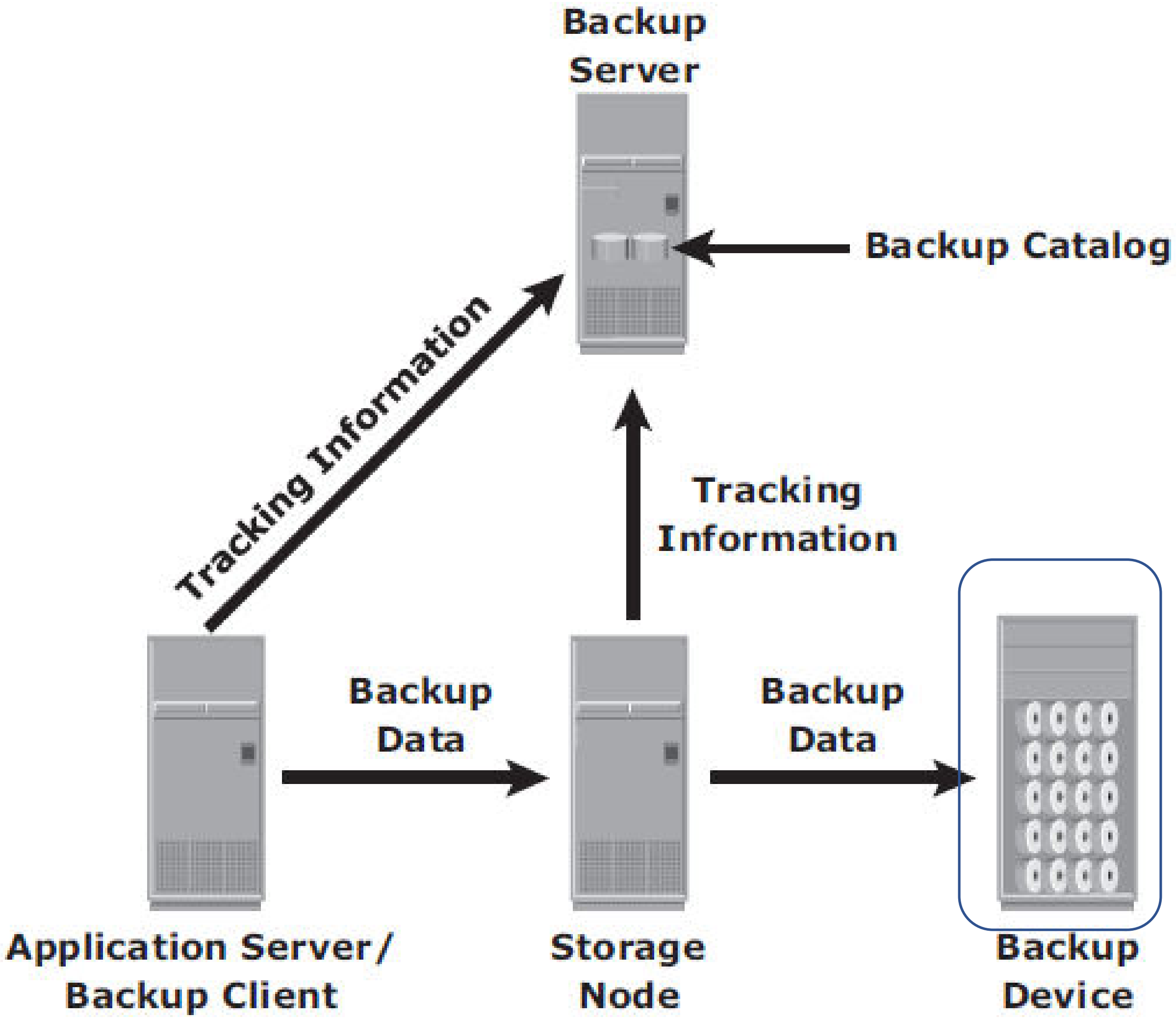
But created using the latest full backup and all the latest incrementals

This type of backup, when generated on the server, does not require resources from the source.

BACKUP ARCHITECTURE

A backup system generally uses a client-server architecture with one backup server and multiple clients. The server manages the backup operations and maintains the catalog, which contains the information about the backups and the metadata. The backup configuration contains information about when to run backups, what client data to back up, and the backup metadata contains information about the data backed up.

The role of the backup client is to send the data to be backed up to the storage node or media server. Also, it sends information to the metadata backup server. The storage node or media server is responsible for writing the data to be backed up on the backup device, which can be a disk, tape or vtl. This also sends information to the backup server. In general, the backup server is integrated in the same equipment as the media server or storage node.



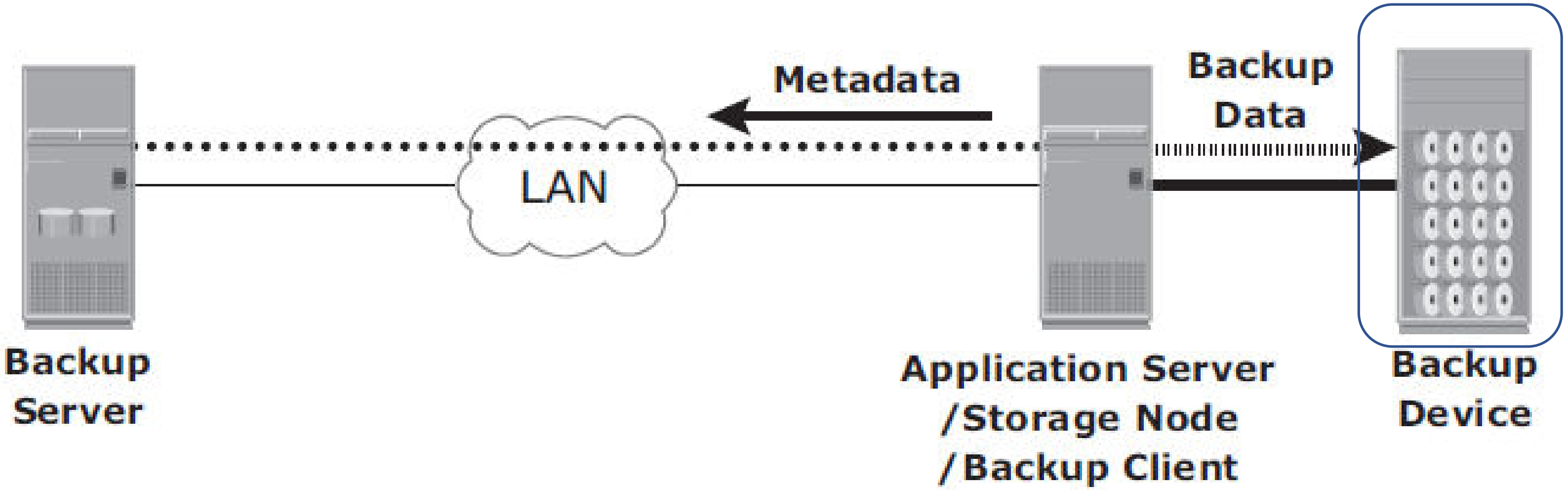
BACKUP TOPOLOGIES

Direct attached backup:

The backup client assumes the role of media server or storage node

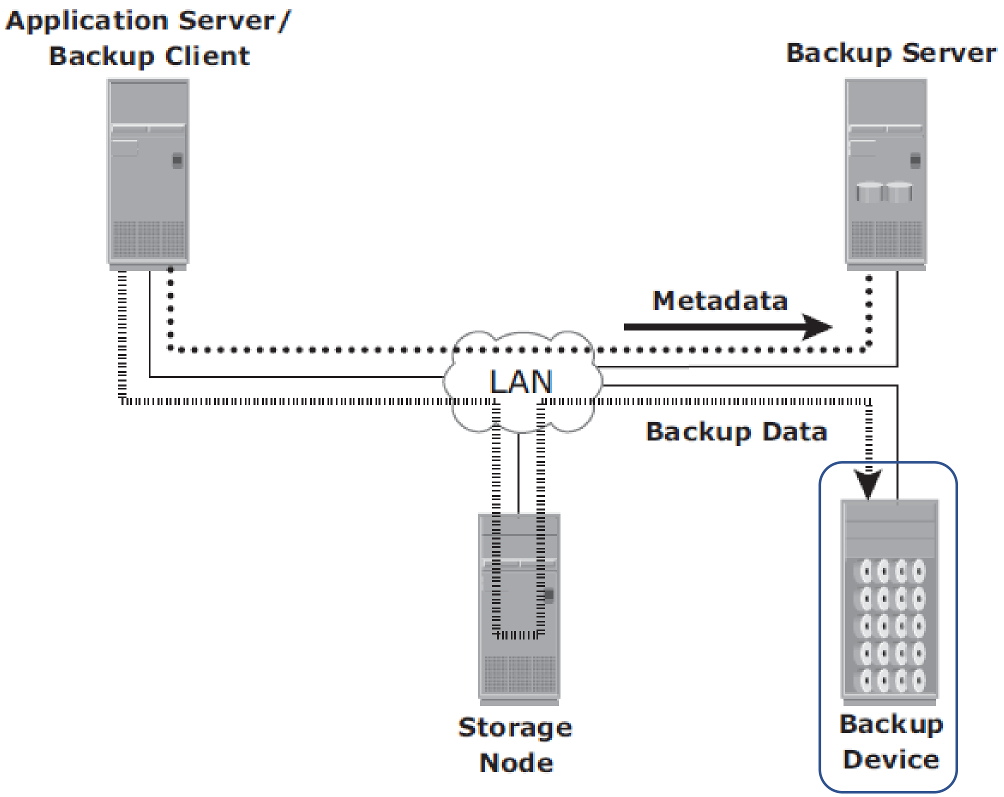
The client sends the data directly to the backup device

This method avoids sending the data over the LAN



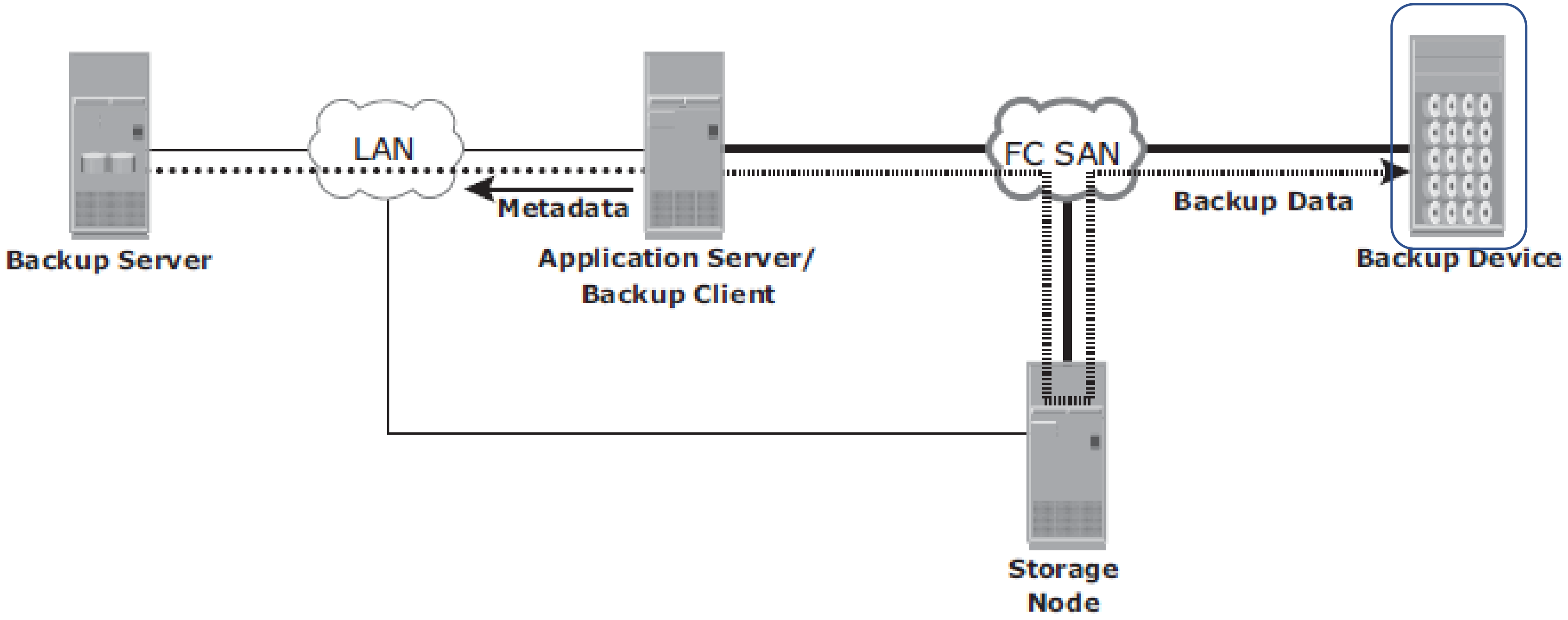
BACKUP TOPOLOGIES

LAN based backup:
The roles are differentiated and are connected through a LAN, where the data to be backed up has to be transferred between the client and the storage or media server.



LAN-based backup

BACKUP TOPOLOGIES



SAN-based backup

SAN-based backup:

Also called LAN-free backup.

It is one of the most appropriate solutions

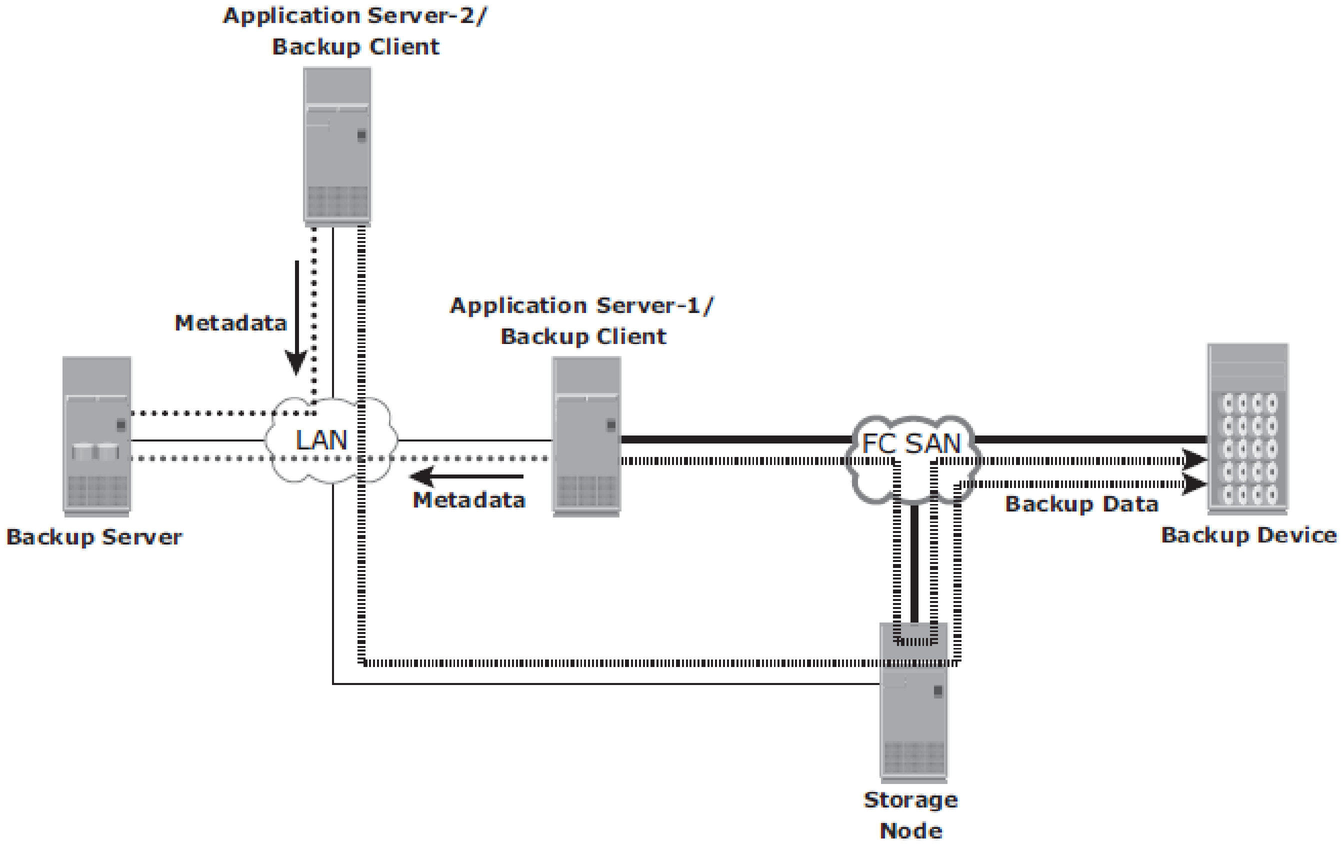
All backup traffic goes through the SAN and is taken directly by the storage node from the disks and sent to the backup device.

BACKUP TOPOLOGIES

Mixed topology:

It is a mix topology of the previous ones.
It can be implemented for a few reasons like:

Costs
Server location
performance considerations



BACKUP TOPOLOGIES

Imaged-based backup:

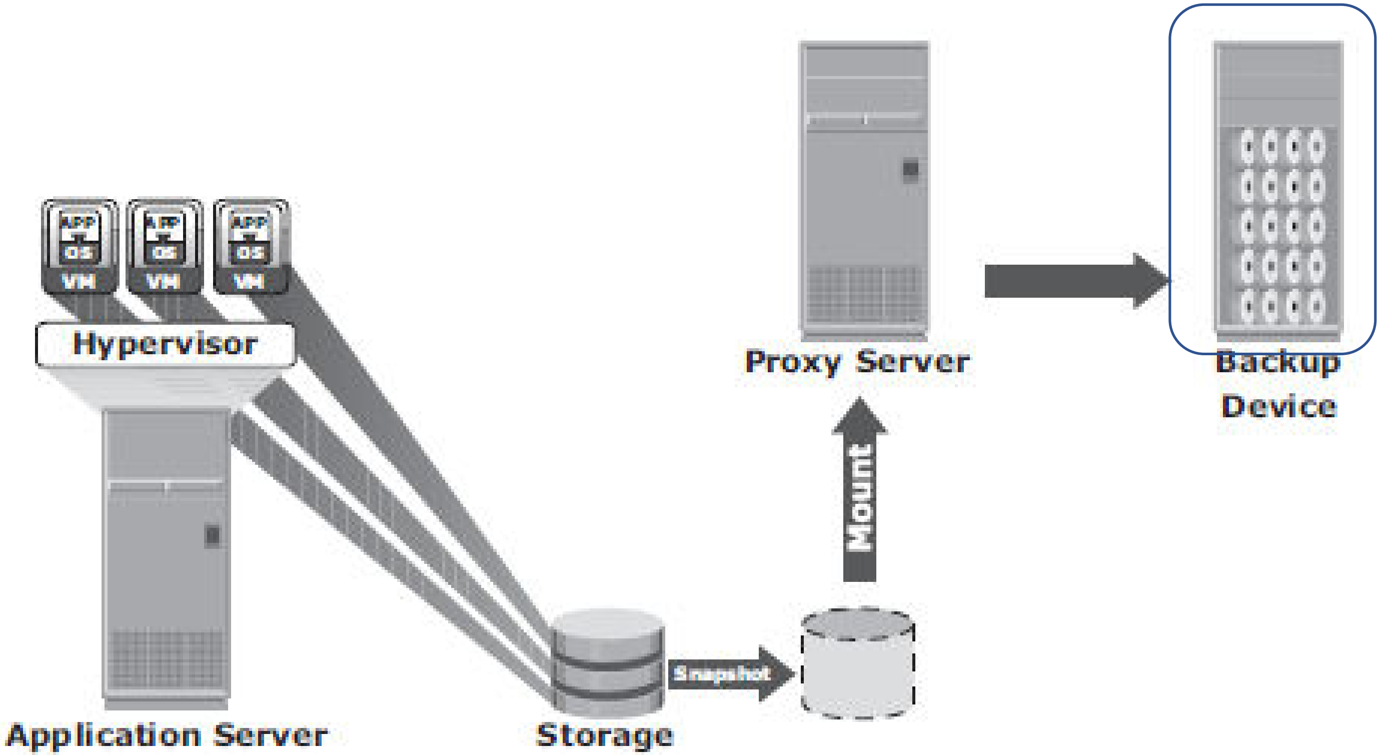
This method works at hypervisor level.

It is a method used in virtualization environments

Create a copy of the system Guest (VM) operating status and VM configuration via snapshot

The backup is stored as an image that is mounted on a proxy server that acts as a backup client.

The backup software backs up that image.



This reduces the load on the hypervisor and the VM

Sometimes the Restore could imply importing a VM or even being able to restore data from the same VM on which it is working.

This requires the installation of the backup agent on the VM

BACKUP

De-duplication:
It is a process to identify and eliminate redundant data.

When duplicate data is detected in a backup, it is removed.

Only the copy already backed up is referenced.

De-duplication allows:

Reduce storage usage for backup.
Shorten the working window.
Reduce network usage.

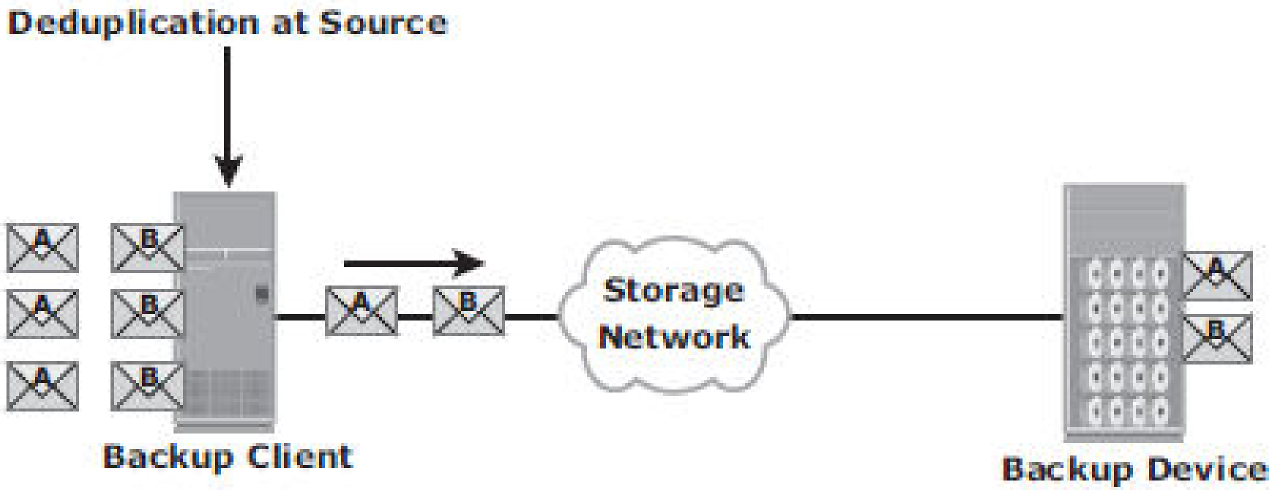
There are two methods (each has its benefits):

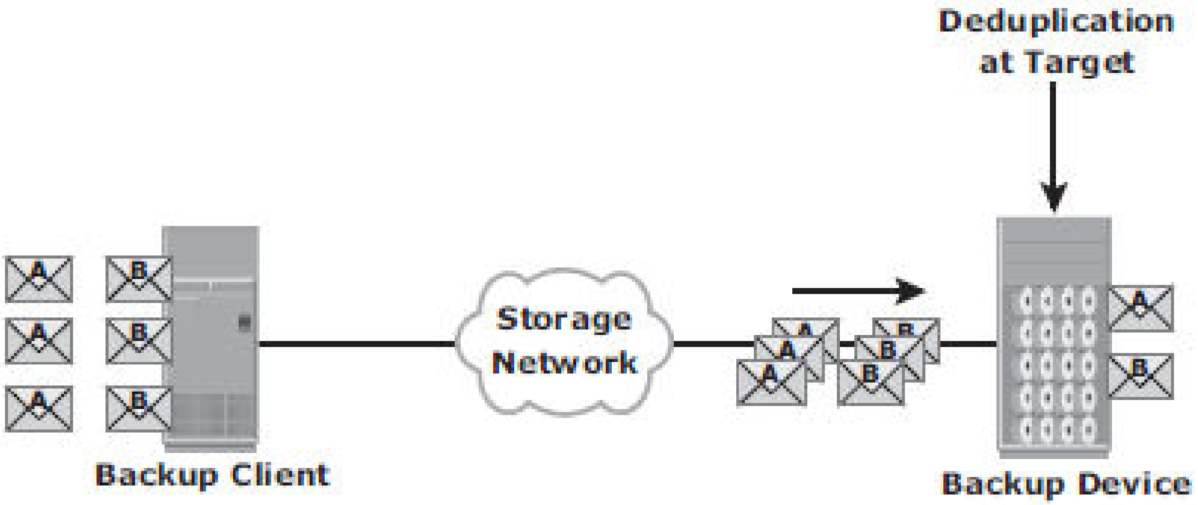
Do it at the source.
Do it at the destination.

There are also de-duplication processes in line or with subsequent off line processes.

BACKUP

De-duplication:





NVF CONCEPTS

NFV CONCEPTS

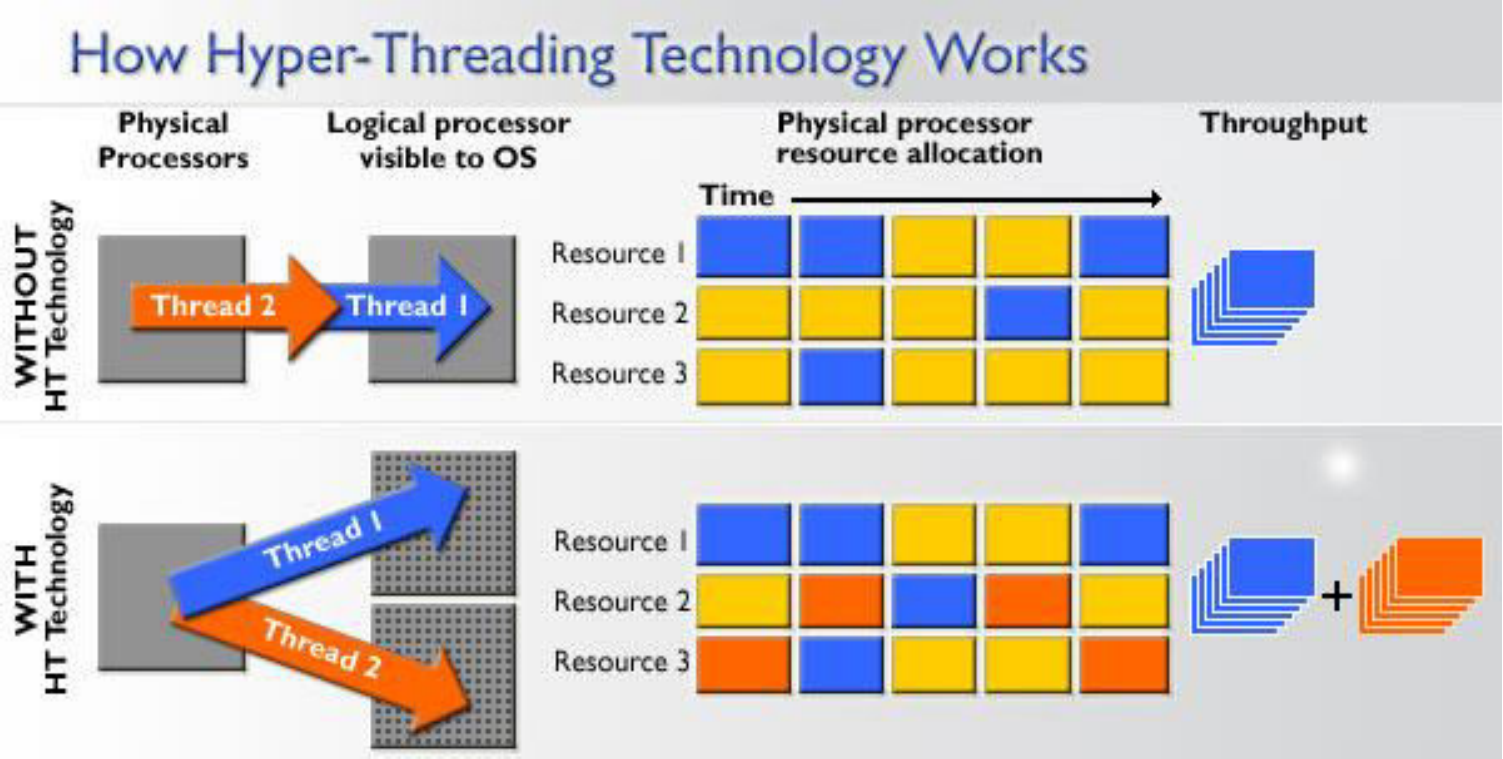
Hyperthreading:

It is an Intel technology that allows a single physical core act as 2 separate logical cores for the operating system and applications.

This technology greatly improves the performance of certain applications.

It depends a lot on the type of application, it can be twice the performance.

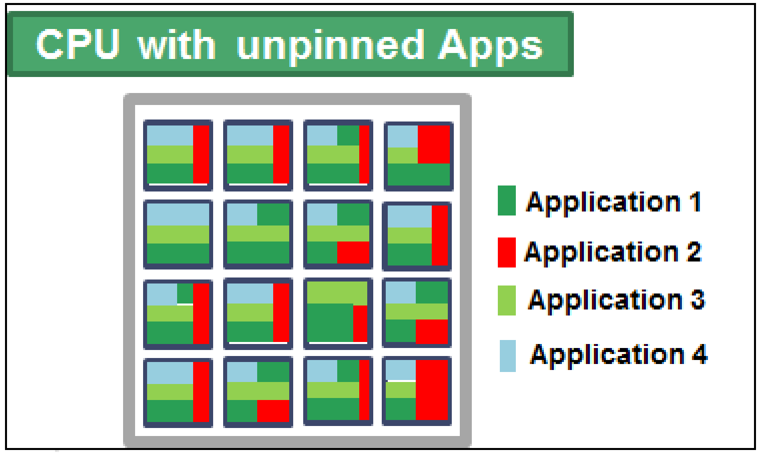
This type of configuration is enabled from the hardware BIOS.

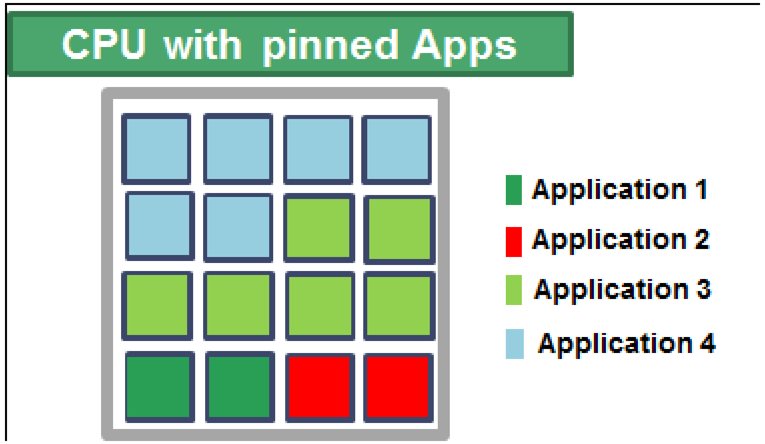


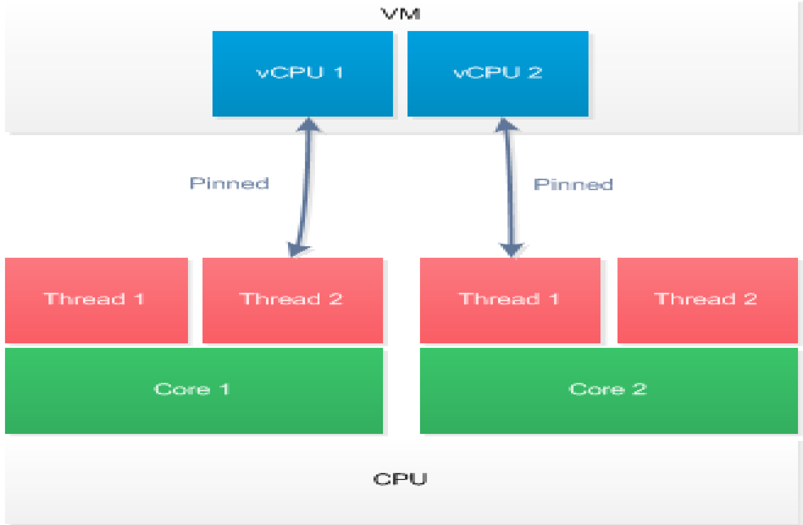
NFV CONCEPTS

CPU Pinning:

It is the ability that exists for a VM to run on a specific physical core. Create a mapping between a vCPU and a physical CPU. This greatly improves the performance of the VM.







NFV CONCEPTS

Flavors:

Flavors define various parameters of a VM (Virtual Machine)

For example:

- vCPUs
- Memory
- Affinity or Anti-affinity with another VM
- Storage
- cpu pinning
- NUMA

NFV CONCEPTS

Open vSwitch “OVS”:

- It is open source software.
- Designed to be used as a virtual switch in virtualized server environments.
- It is in charge of forwarding traffic between different VMs on the same physical host (physical server).
- And also forward the traffic between the VMs and the physical network (The NICs of the server).
- The Open vSwitch is one of the most popular implementations of OpenFlow.
- Open vSwitch supports numerous Linux-based technologies.

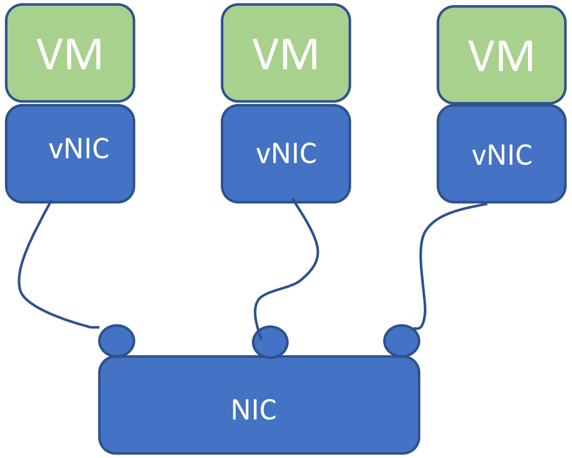
- Open vSwitch provides 2 external management protocols designed for remote management from SDN controllers:
  - OpenFlow: Allows you to consult and modify the flow tables, causing the behavior of the OVS to be dynamically reprogrammed with this protocol, so that the packets can be sent from one port to another port of the OVS.
  - OVSDB (Open vSwitch DataBase Management Protocol): Protocol to manage and modify the OVS configuration.

INTRODUCTION TO SDN

SRIOV, OVS, DPDK, NETCONF

SR IOV DEFINITION

Single Root I/O Virtualization (SR-IOV) is a virtualization technology that allows a physical network card (NIC) to be shared among multiple virtual machines (VMs) or containers with performance very close to that of a dedicated physical NIC. Here is a detailed description of SR-IOV:



SR IOV DEFINITION

GENERAL DESCRIPTION

SR-IOV is a specification that enables a single root port *peripheral component interconnect express* (PCIe) physical device to appear as several different physical devices to the hypervisor or guest operating system. SR-IOV uses physical functions (PF) and virtual functions (VF) to manage global functions of SR-IOV devices.

PFs are complete PCIe functions that can configure and manage SR-IOV functionality. With PFs, PCIe devices can be configured or controlled, and the PF has full ability to put data on and off the device.

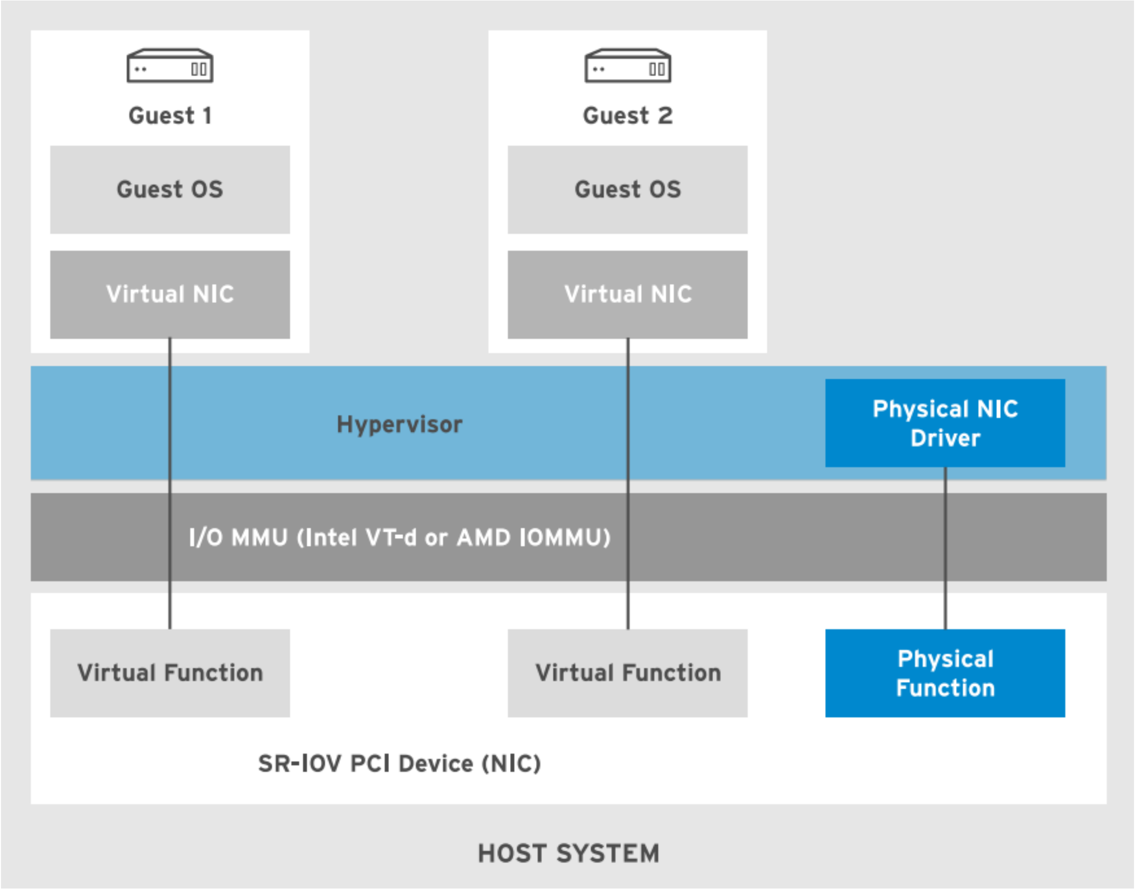
VFs are lightweight PCIe features that support data flow, but have a restricted set of configuration resources. The number of virtual functions that are provided to the hypervisor or guest operating system depends on the device.

SR-IOV-enabled PCIe devices require appropriate hardware and BIOS support, as well as SR-IOV support in the hypervisor instance or guest operating system driver. See SR-IOV Support.

SR IOV DEFINITION

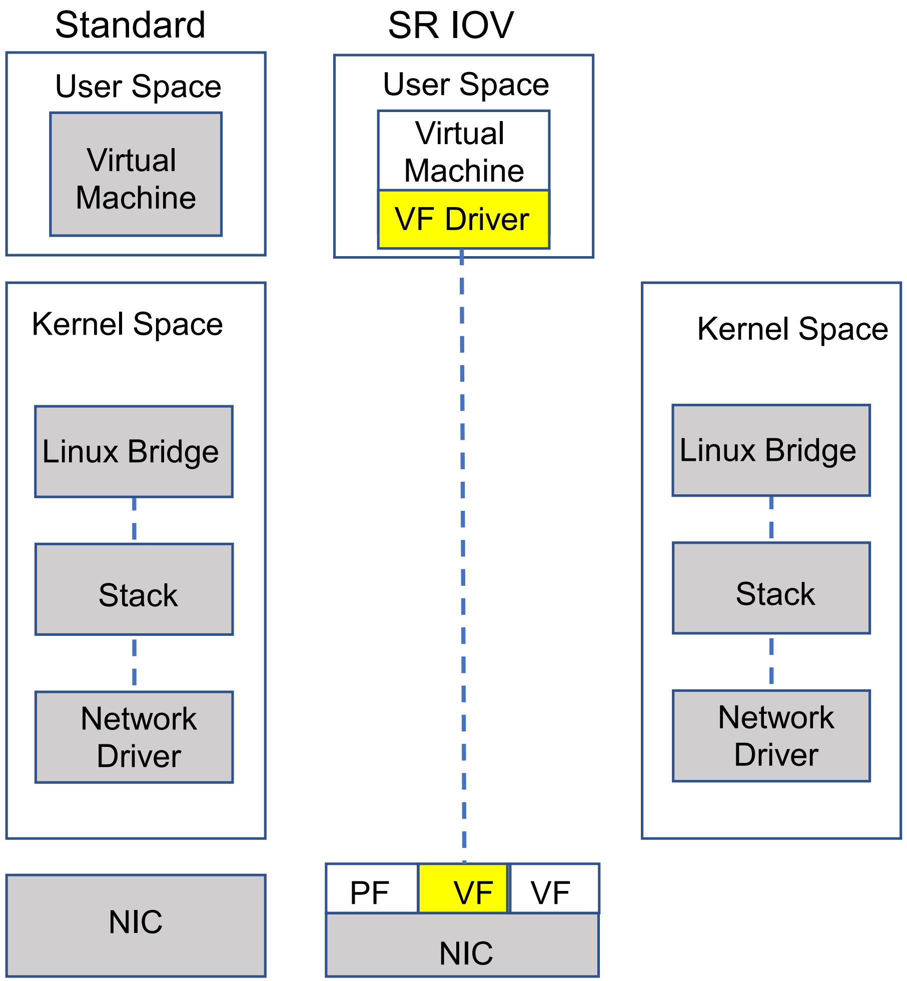
GENERAL DESCRIPTION

WHAT IS SR-IOV? Single-Root I/O Virtualization (SR-IOV) is a feature available in some PCIe network interface cards (NIC), that allows the physical NIC to be presented as several virtual NICs. This means that a single physical NIC can be shared between multiple virtual machines, removing the usual 1:1 mapping limitation when passing PCI devices through from the hypervisor to the virtual guests. In SR-IOV terms, a physical NIC is called a Physical Function (PF), and a virtual NIC is called a Virtual Function (VF).



SR IOV DEFINITION

GENERAL DESCRIPTION



SR IOV DEFINITION

GENERAL DESCRIPTION

PCI Passthrough Prior to SR-IOV: it was possible to provide a guest with direct access to a PCI device using PCI passthrough. This provided better performance but meant that the device was dedicated to the instance, and could not be utilized if the instance was idle. With SR-IOV, this limitation was overcome, allowing direct access to a NIC that is shared between instances. You can still dedicate the whole NIC to a specific instance, by passing through the Physical Function instead of a Virtual Function. You can also use PCI passthrough to provide direct access to other types of PCI devices, or if you do not want to enable SR-IOV on your NIC.

SR-IOV USE CASES: Described simply, virtual NICs are processes emulating a physical NIC, so they consume CPU and memory resources on the hypervisor. Using SR-IOV removes the need for these resources (it bypasses the OVS: usually VM connects to OVS and then to the physical port), offloading packet processing to the NIC hardware, and freeing up hypervisor resources for other tasks. That means that the VM connector, through SRIOV, connects to the logical part of the physical port.

SR IOV DEFINITION

SR-IOV provides direct access to the NIC, avoiding the layer of virtualization in the hypervisor, as well as the virtual switch. This increases performance compared to instances with virtual NICs. High performance networking is often required in an NFV implementation, where the VNF is replacing dedicated, optimized network hardware.

Supported Hardware Red Hat has tested several original Intel network cards with SR-IOV support, they included the following chip sets:

- 82598/82599
- X520/X540/X550
- X710/XL710/X722

Source: Red Hat

SR IOV DEFINITION

GENERAL DESCRIPTION

Red Hat has also tested 10 Gb SR-IOV cards from Mellanox and Qlogic.

SR-IOV CONFIGURATION: To configure SR-IOV during overcloud deployment, add composable roles through rolesdata.yaml to the Heat templates on the Undercloud node. You also need to add a flavor with appropriate settings, and configure kernel arguments, in network-environment.yaml. The neutron-sriov.yaml file contains the SR-IOV agent, drivers, parameters, and devices needed to configure SR-IOV. If your OpenStack environment does not use Heat for deployment, you can configure SR-IOV manually. Manual configuration is documented but is not a simple process. Using Heat templates is the method recommended by Red Hat.

REFERENCES Further information is available in the chapter on Configure SR-IOV Support For Virtual Networking in the Red Hat OpenStack Platform 7 Network Functions Virtualization Configuration Guide at https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/ Further information about hardware compatibility is available in the Red Hat Ecosystem site at https://access.redhat.com/ecosystem

Heat / Neutron, see Open Stack for more details

Source: Red Hat

SR IOV DESCRIPTION

1. **Physical Resource Assignment:**
With SR-IOV, a physical NIC is divided into multiple virtual resources called "VFs" (Virtual Functions). Each VF maps to a VM or container as if it were a dedicated physical NIC.

2. High Performance:
SR-IOV provides performance close to that of a dedicated physical NIC, as the VFs have direct access to physical resources without going through an additional virtualization layer. This minimizes latency and maximizes network performance.

3. Virtualization Driver:
The hypervisor or host operating system must support SR-IOV and have drivers to support it. Additionally, the physical NIC must support SR-IOV and have SR-IOV-enabled drivers.

4. Isolation:
Despite sharing the same physical NIC, VMs or containers using SR-IOV are isolated from each other. This means that a VM or container cannot directly access the network resources of another VM or container.

5. Directly to Hardware:
When a VM or container sends or receives data through a VF, the data is transmitted directly from the VF to the physical hardware of the NIC, avoiding processing overhead on the hypervisor.

SR IOV DESCRIPTION

6. Hardware Requirements:
SR-IOV requires specific physical NICs that support the technology. Also, the hardware must be on a system that also supports SR-IOV. Not all servers and NICs support SR-IOV.

7. Centralized Management:
SR-IOV management is typically done centrally through hypervisor management or system management tools. This makes it easy to configure and assign VFs to VMs or containers.

8. Scalability:
SR-IOV is especially useful in environments where scalability and high network performance are required, such as data centers and server virtualization environments.

9. Applications:
SR-IOV is used in a variety of applications that require high network performance, such as application servers, network gateways, firewalls, and high-performance storage systems.

10. Resource Economy:
SR-IOV can help optimize the use of hardware resources by allowing multiple VMs to share one physical NIC without compromising performance.

SR IOV DESCRIPTION

In short, SR-IOV is an I/O virtualization technology that allows a physical NIC to be shared among multiple VMs or containers with performance very close to that of a dedicated physical NIC. It provides high performance, isolation, and scalability, making it suitable for environments that require high-performance, resource-efficient networking. However, its implementation requires specific hardware and drivers that support SR-IOV.

SR IOV PROS AND CONS

PROS

1. **High Performance:** One of the biggest advantages of SR-IOV is its ability to offer high performance in virtualized environments. Allows virtual machines to directly share the physical NIC, reducing virtualization overhead and providing direct hardware access.

2. **Low Latency:** Since the virtual machines directly access the physical NIC, the latency is low. This is critical in applications that require real-time responses, such as financial or telecommunications applications.

3. **Resource Isolation:** SR-IOV provides an additional level of resource isolation, which means that each virtual machine can be guaranteed specific NIC resources and bandwidth, thus improving quality of service.

4. **Power Efficiency:** SR-IOV can help reduce power consumption by allowing multiple virtual machines to share a single physical NIC instead of using multiple separate physical NICs.

SR IOV PROS AND CONS

CONS

1. **Requires Specific Hardware:** Not all network adapters support SR-IOV. You will need specific hardware that is compatible with this technology.

2. **Configuration Complexity:** Configuring and managing SR-IOV can be complicated, especially in environments with multiple virtual machines and network adapters. It requires a solid knowledge of technology and its configuration.

3. **Mobility Limitations:** Virtual machines using SR-IOV may have mobility limitations, since they are directly tied to a physical NIC. Moving a virtual machine across physical hosts can be tricky.

4. **Security:** Since virtual machines have direct access to the physical NIC, it is important to consider security. A failure in one virtual machine could affect all virtual machines that share the same physical NIC.

In short, SR-IOV is a powerful technology for improving throughput and latency in virtualized environments, but it also comes with some complexity and specific hardware requirements. It is important to carefully assess whether the benefits outweigh the limitations and potential challenges.

SR IOV IN VMWARE

Usar SR-IOV en vSphere

En vSphere, una máquina virtual puede utilizar una función virtual de SR-IOV para las redes. La máquina virtual y el adaptador físico intercambian datos directamente sin utilizar el VMkernel como instancia intermediaria. La omisión del VMkernel en las redes reduce la latencia y mejora la eficiencia de la CPU.

En vSphere, si bien un conmutador virtual (estándar o distribuido) no controla el tráfico de red de una máquina virtual habilitada para SR-IOV que está conectada al conmutador, es posible controlar las funciones virtuales asignadas mediante las directivas de configuración de conmutadores en el nivel de puerto o grupo de puertos.

- Compatibilidad con SR-IOV

vSphere admite SR-IOV exclusivamente en entornos con una configuración específica. Algunas características de vSphere no funcionan cuando se habilita SR-IOV.

- Interacción y arquitectura del componente SR-IOV

La compatibilidad de SR-IOV con vSphere depende de la interacción entre las funciones virtuales (VF) y la función física (PF) del puerto de NIC para lograr un mejor rendimiento, y de la interacción entre el controlador de la PF y el conmutador del host para el control de tráfico.

Fuente Vmware

SR IOV IN VMWARE

- Interacción entre la función virtual y vSphere

Las funciones virtuales (VF) son funciones de PCIe ligeras que contienen todos los recursos necesarios para el intercambio de datos, pero tienen un conjunto reducido de recursos de configuración. La interacción entre vSphere y las VF es limitada.

- DirectPath I/O frente a SR-IOV

SR-IOV ofrece beneficios de rendimiento y compensaciones similares a los de DirectPath I/O. DirectPath I/O y SR-IOV tienen una funcionalidad similar, pero se usan para lograr diferentes objetivos.

- Configurar una máquina virtual para utilizar SR-IOV

Para utilizar las capacidades de SR-IOV, se deben habilitar las funciones virtuales de SR-IOV en el host y conectar una máquina virtual a las funciones.

- Opciones de redes para el tráfico relacionado con una máquina virtual con SR-IOV habilitado

En vSphere, puede configurar ciertas funciones de redes en un adaptador de máquina virtual que tenga una función virtual (virtual function, VF) asociada. Use las opciones de configuración del conmutador, el grupo de puertos o el puerto en función del tipo de conmutador virtual (estándar o distribuido) que controla el tráfico.

Fuente Vmware

OVS

OVS DESCRIPTION

Open vSwitch (OVS) is an open source virtual switch that is widely used in network virtualization and cloud environments. Here is a detailed description of Open vSwitch

1. **Virtual Switch:**

OVS is a virtual network switch that operates at the software level, which means that it is not physical hardware, but rather a software entity that emulates an Ethernet switch.

2. Versatility:

OVS is highly versatile and can be used in a variety of environments, from virtual servers to high performance physical switches. It supports multiple hypervisors, including KVM, Xen, and VMware.

3. Layers of the OSI Model:

OVS operates on multiple layers of the OSI model. It can perform switching and routing functions at layer 2 (data link layer) and layer 3 (network layer). This makes it suitable for a wide range of network applications.

4. **Command Line Interface:**

OVS can be managed through a command line interface (CLI) and through management controllers such as OVN (Open Virtual Network) and OpenStack Neutron.

OVS DESCRIPTION

5. Control Protocols:
OVS supports several control protocols, including OpenFlow, OVSDB (Open vSwitch Database Management Protocol), and Layer 2 Management Protocol (L2MP).

6. OpenFlow:
OpenFlow is a communication protocol that allows network controllers to manage and control OpenFlow-enabled switches and routers, including OVS. This is essential for implementing software-defined networks (SDN).

7. Flow:
In OVS, the filtering and routing rules are managed in the form of flows. Flows are rules that determine how network traffic should be processed. This allows fine-grained control over how traffic is handled on the virtual network.

8. Network Segmentation:
OVS is capable of performing network segmentation through packet labeling. This is useful in cloud and virtualization environments where it is required to isolate traffic between virtual machines.

OVS DESCRIPTION

9. Bridging:
OVS supports the functionality of virtual bridges, which allows you to create virtual network segments and connect virtual machines or containers to them.

10. Cloud Deployments:
OVS is widely used in cloud network deployments, including deployments of OpenStack, Kubernetes, and other cloud orchestration and container management platforms.

11. Active Ecosystem:
OVS has an active development community and supports a variety of projects related to network virtualization and SDN.

12. Controller Support:
OVS supports several management controllers that allow advanced virtual network management and configuration.

OVS DESCRIPTION

In summary, Open vSwitch is a highly versatile and widely used virtual network switch that operates at the software level. It is essential in network virtualization and SDN environments, offering a high degree of flexibility and control over network traffic in a variety of network and cloud environments.

OVS DESCRIPTION

The journey of a packet from the NIC (network interface card) to a virtual machine using OVS-DPDK (Open vSwitch with Data Plane Development Kit) involves several steps. Here is a simplified description of how this process works:

OVS DESCRIPTION

1. Receiving the Packet at the NIC: When a network packet arrives at the physical server NIC, the NIC receives it and temporarily stores it in a buffer.

2. Processing on the NIC: The NIC performs some initial operations, such as checking the checksum, and then passes the packet to the OVS-DPDK userspace.

3. OVS-DPDK in User Space: OVS-DPDK runs in the user space of the physical server operating system. Here, OVS-DPDK takes the packet from the NIC and processes it.

4. Flow Rules: In OVS-DPDK, flow rules are applied to determine how the packet should be handled. These rules can include assigning packets to specific virtual machine ports or performing actions such as routing.

DPDK: Data Plane Development Kit

OVS DESCRIPTION

5. Mapping to a Virtual Machine: If the packet is destined for a specific virtual machine, OVS-DPDK maps it to the corresponding virtual port. This may involve tagging the packet with network information specific to the virtual machine.

6. Delivery to the Virtual Machine: Once the packet is assigned to the virtual machine, OVS-DPDK delivers it to the operating system of the virtual machine in question.

7. Processing in the Virtual Machine: Inside the virtual machine, the VM's operating system takes the packet and processes it based on the VM's network configurations.

8. Final Delivery to the Recipient Process or Application: Finally, the package is delivered to the process or application within the virtual machine that is the final recipient. This could be a web server, an email application, or any other software that uses the network.

OVS DESCRIPTION

Importantly, OVS-DPDK is a critical part of this process, providing an efficient way to handle network traffic in virtualization environments. Using DPDK enables higher network performance and lower latency by accelerating packet processing operations in user space instead of relying exclusively on the physical server operating system kernel

OVS DPDK

DPDK

The Data Plane Development Kit (DPDK) is a set of open source software libraries and drivers designed to speed up packet processing in network and communications applications. DPDK was originally developed by Intel, but is now an open source project supported by the Linux Foundation. Its primary goal is to take full advantage of the performance of network devices, such as network cards (NICs) and hardware accelerators, for applications that require high performance and low latency, such as network gateways, routers, firewalls, and security systems. .Here is a detailed description of the key components and features of DPDK:

DPDK DESCRIPTION

1 Network Packet I/O:
DPDK provides a high-performance abstraction for capturing and forwarding network packets. This is achieved through direct access to network cards (NICs) through optimized drivers and high-performance I/O libraries.

2 Memory Optimization:
DPDK uses memory management techniques to minimize memory management overhead and reduce packet access latency.

3 Programming Model:
DPDK programs are developed using a C programming model, which allows a high degree of control and optimization of the packet processing logic.

4 Control APIs:
DPDK offers a variety of APIs for controlling and configuring network devices, such as NICs and I/O controllers. This allows detailed manipulation and control of the devices.

5 Processing Optimization:
DPDK provides highly optimized packet processing routines, such as routing and filtering functions, which speed up packet processing across multiple CPU cores

DPDK DESCRIPTION

6 Cross-platform:
Although it originated on Intel's x86 architecture, DPDK has been adapted to support various architectures, including ARM and PowerPC.

7 NIC Support:
DPDK integrates with a wide range of high-performance NICs from multiple vendors. In addition, it offers a unified driver architecture that allows applications to interact with different NICs in a consistent manner.

8 Virtualization Support:
DPDK offers support for virtualized environments, which means that it can be used on virtual machines (VMs) to speed up network performance in virtualized environments.

9 Use in Applications:
DPDK is used in a variety of applications that require high network performance, such as network gateways, firewalls, load balancers, network monitoring systems, and more.

10 Open Source Project:
DPDK is an open source project with an active community of developers and a focus on continuous innovation and performance improvement.

DPDK DESCRIPTION

In short, DPDK is a key technology to speed up network packet processing in high-performance applications. It allows applications to interact directly with network devices and optimizes packet handling for exceptional performance. It is widely used in the network industry and has become an essential component in high-performance network applications.

OVS DPDK

OVS-DPDK (Open vSwitch with Data Plane Development Kit) is an extension to Open vSwitch that uses the Data Plane Development Kit (DPDK) to accelerate network data plane performance.

PROS

1. **High Performance:** One of the main advantages of OVS-DPDK is its ability to offer high performance in virtualized network environments. DPDK enables significant performance acceleration by taking advantage of underlying hardware capabilities, such as high-performance network cards.

2. **Low Latency:** OVS-DPDK can provide low latency, which is critical in applications that require real-time responses, such as telecommunication networks and financial systems.

3. **Greater Scalability:** OVS-DPDK can handle large volumes of network traffic and is scalable for environments that require high packet processing capacity.

4. **Support for NFV:** OVS-DPDK is a popular choice in Network Functions Virtualization (NFV) environments due to its performance and scalability.

OVS DPDK

CONS

1. **Specific Hardware Required:** To take full advantage of OVS-DPDK, you need specific hardware that is compatible with DPDK. Not all network adapters are supported.

2. **Higher Resource Consumption:** OVS-DPDK uses more system resources, including CPU, compared to traditional OVS. This can increase operating costs and hardware requirements.

3. **Increased Complexity:** Configuring and managing OVS-DPDK can be more complex than traditional OVS due to additional configurations and hardware dependencies.

4. **Potential Compatibility Issues:** Compatibility issues may arise with certain versions of the DPDK and NIC drivers. Compatibility management can be challenging.

5. **Security:** Since OVS-DPDK has direct access to the hardware NIC, it is important to ensure adequate security, as a configuration failure could have a significant impact on the network.

In summary, OVS-DPDK is a solid choice for environments that require high performance and low latency in virtualized networks, especially in use cases like NFV. However, it also comes with specific hardware requirements and increased configuration and management complexity. Before implementing OVS-DPDK, it is important to carefully assess your hardware compatibility and performance needs.

OVS DPDK USE CASE SBC

OVS DPDK: USE CASE. SBC

When evaluating resources when deploying a Session Border Controller (SBC) using OVS-DPDK (Open vSwitch with Data Plane Development Kit), it is important to consider several key factors to ensure optimal performance and efficient resource allocation. Here are some general guidelines

1 HW REQUIREMENTS:

- **CPU:** Determines the packet processing capacity and workload that the SBC can handle. The more CPUs and cores available, the better.
- **RAM memory:** Make sure you have enough RAM memory for SBC operations and the use of OVS-DPDK. The amount of RAM required depends on the workload.

2 PACKET SIZE:

Consider the average size of the packets to be processed. Larger packages may require more resources than smaller packages due to processing overhead.

3 BANDWIDTH:

Evaluate the bandwidth capacity required for your workload. Make sure the NICs and hardware are capable of handling the expected amount of traffic.

OVS DPDK: USE CASE. SBC

4. OVS-DPDK CONFIGURATION:

- **Number of Virtual Cores (VCPU):** Defines how many virtual cores to assign to OVS-DPDK. This depends on the number of physical CPUs available and the workload.
- **Memory Allocation:** Configures the amount of memory that will be allocated to OVS-DPDK. Make sure you have enough memory available for your needs.
- **DPDK NIC Configuration:** Configures the NICs to use DPDK and assigns specific cores to the NICs for optimal performance.

5. RESOURCE MONITORING:

- Use resource monitoring tools like top, htop, or DPDK-specific tools to assess CPU, memory, and bandwidth utilization.

6. LOAD TESTS:

- Carry out load tests to simulate real traffic situations and evaluate the performance of the SBC. This can help you determine if your current resource settings are adequate.

7. ADJUSTMENT AND OPTIMIZATION:

- Performs adjustments and optimizations in the OVS-DPDK configuration based on the results of load tests and resource monitoring.

OVS DPDK: USE CASE. SBC

8. Scalability:
Consider the ability to scale out by adding more SBC instances and spreading the load evenly.

9. Security:
Make sure that the resources are configured in a secure way and that the SBC meets the necessary security requirements.

10. Technical Support:
It is always useful to have technical support from hardware and software vendors in case of unexpected problems or challenges.

In summary, the evaluation and allocation of resources when implementing an SBC with OVS-DPDK is a critical process to ensure optimal performance and the ability to handle the expected workload. Consider the specific needs of your workload and perform extensive testing to adjust and optimize resource settings as needed.

OVS DPDK: SBC SCALE EXAMPLE WITH OVS DPDK

Determining the resources required to support 30,000 sessions on a Session Border Controller (SBC) with OVS-DPDK and SR-IOV can be challenging, as it depends on several factors, including workload, expected traffic, and hardware capacity. . Here's a general estimate, but keep in mind that the exact settings may vary based on your specific needs:

1. CPU:
A multi-core CPU configuration is recommended for OVS-DPDK, as this approach relies on allocating cores to specific tasks. For 30,000 sessions, you may need at least 16 vCores or more, depending on the complexity of the sessions and any additional functions the SBC needs to perform (such as media transcode).

2. RAM:
The amount of RAM required will depend on the workload and additional applications running on the SBC. For 30,000 sessions, at least 32 GB or more of RAM would be recommended.

3. DPDK NIC:
You will need DPDK-compliant network cards (NICs) and enough CPU cores specifically allocated for packet processing. This may vary depending on the speed of the sessions and the expected data transfer rate.

4. Storage:
Make sure you have enough storage for the operating system and any logs or data generated by the SBC. This may vary based on record retention policies (how long must data be storaged).

5. Optimization:
Performs adjustments and optimizations in the OVS-DPDK configuration to ensure optimal performance and session load balancing.

OVS DPDK: SBC SCALE EXAMPLE WITH SR IOV

1. CPU:
SR-IOV can be less CPU intensive compared to OVS-DPDK since virtual NICs directly allocate hardware resources. However, you will still need multiple vCores to handle 30,000 sessions. May require fewer cores than OVS-DPDK.

2. RAM:
The amount of RAM required will remain similar to that of OVS-DPDK, at least 32 GB or more.

3. SR-IOV NICs:
You must have SR-IOV capable NICs and enough CPU cores to support sessions allocated via SR-IOV. The number of NICs will depend on the capacity of each one and the data transfer rate.

4. Storage:
Similar to OVS-DPDK, you will need storage for the operating system and other data.

It is important to note that these are general estimates and the exact configuration may vary based on your application needs and hardware architecture. Load testing in a lab environment is recommended to determine the precise resources required for your specific workload. Also, consider scalability to handle a larger number of sessions if needed in the future.

NETCONF

WHAT IS NETCONF

- NETCONF, is Network Configuration Protocol

- Initial proposal RFC 4741 year 2006
  RFC 6241 year 2011 replacement
  RFC 7803 2016
  RFC 8526 year 2019

- Netconf provides mechanisms for:
  - Install
  - Manipiulate
  - Erase

  Network Device Configuration

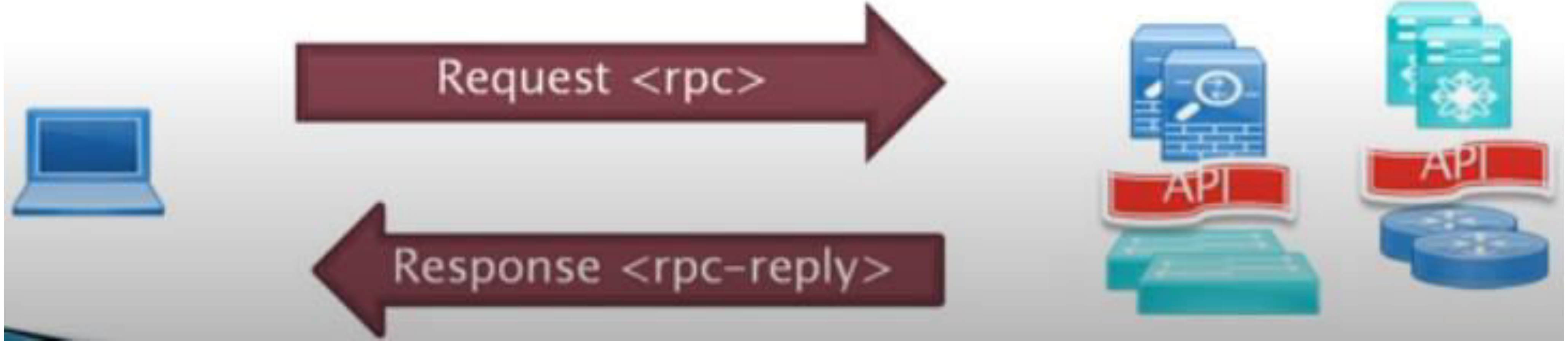
WHAT IS NETCONF

- NETCONF uses information coded using XML/JSON with YANG both:
    - For Configuration Data
    - For Protocol messages
- NETCONF operation are achieved through RPCs (Remote Procedure Calls)

- NETCONF session is a logical connection between:
    - A configuration application, and
    - One or more Network Device

NETCONF DESCRIPTION

A Netwok configuration application modifies or request information through RPCs requests and device administrated will answer with RPCs replay messages



RPC: Remote Procedure Call

NETCONF TERMINOLOGY

CLIENT:

- Invoke operations on a Server
- It can subscribe to receive notification messages from Server when a particular event ocurrs.
- The client can be one script (ie Python), or an application (network management)

SERVER:

- It executes operations invoked by the client
- It can send notifications required by client.
- Server, typically is a Network Device

Note: Client rol and Server rol in NETCONF are oposite of those defined for SNMP

NETCONF TERMINOLOGY

RPCs (Remote Procedure Call):

- It is an operation achieved through messages: <rpc> and <rpc-replay>

SESSION:

- It consist of messages interchange between client and server, using a connection-orient and secure session

MESSAGE:

- XML documents “well formed”

Note: Client rol and Server rol in NETCONF are oposite of those defined for SNMP

WHY NETCONF

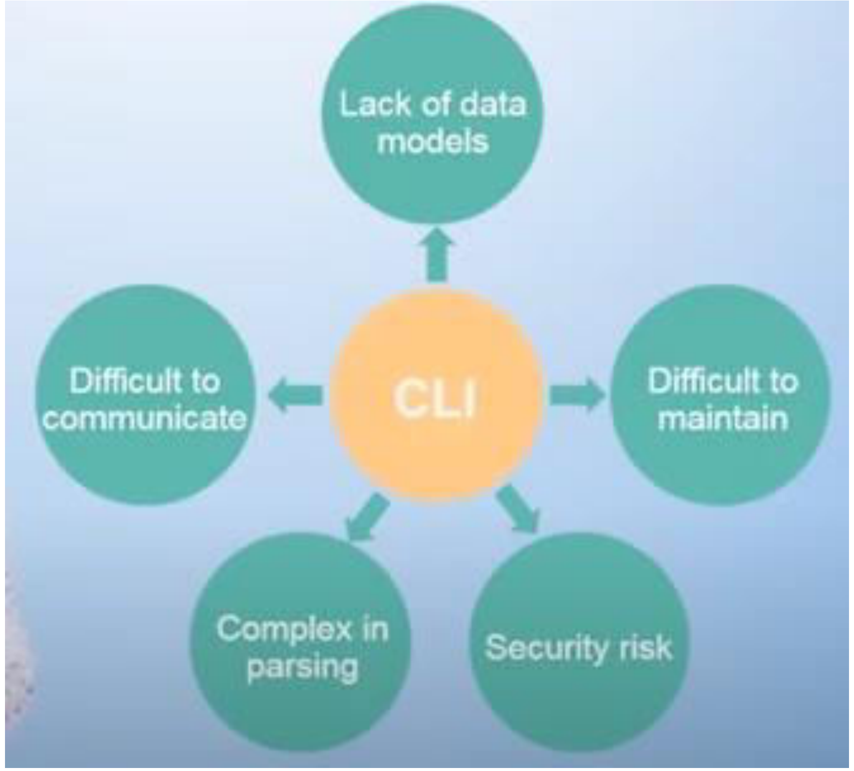
As CLI and SNMP, NETCONF is network device management protocol. It provides a mechanism to configure the devices and quering the network and the state of it.

Similar to SNMP, which uses MIB to model data, Netconf uses YANG to describe the interaction models between the NETCONF client and server

Why do we use NETCONF?
One of the keys network requirement of the cloud era is the Network Automation

Quick, Automatic and demand services provisioning and automatic O&M

CLI and SNMP, don't meet the requirements of the cloud-based networks.



Fuente: Huawei

NETCONF

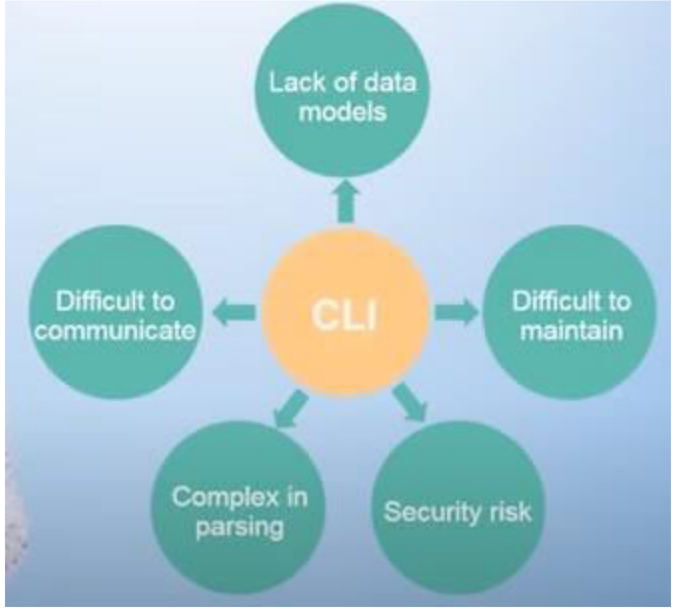
As CLI is base on man-machine interfaces

The configuration is complex

Cost of manual learning configuration and maintenance is high

Devices configuration varies with vendors

Devices interworking is difficult



Fuente: Huawei

NETCONF

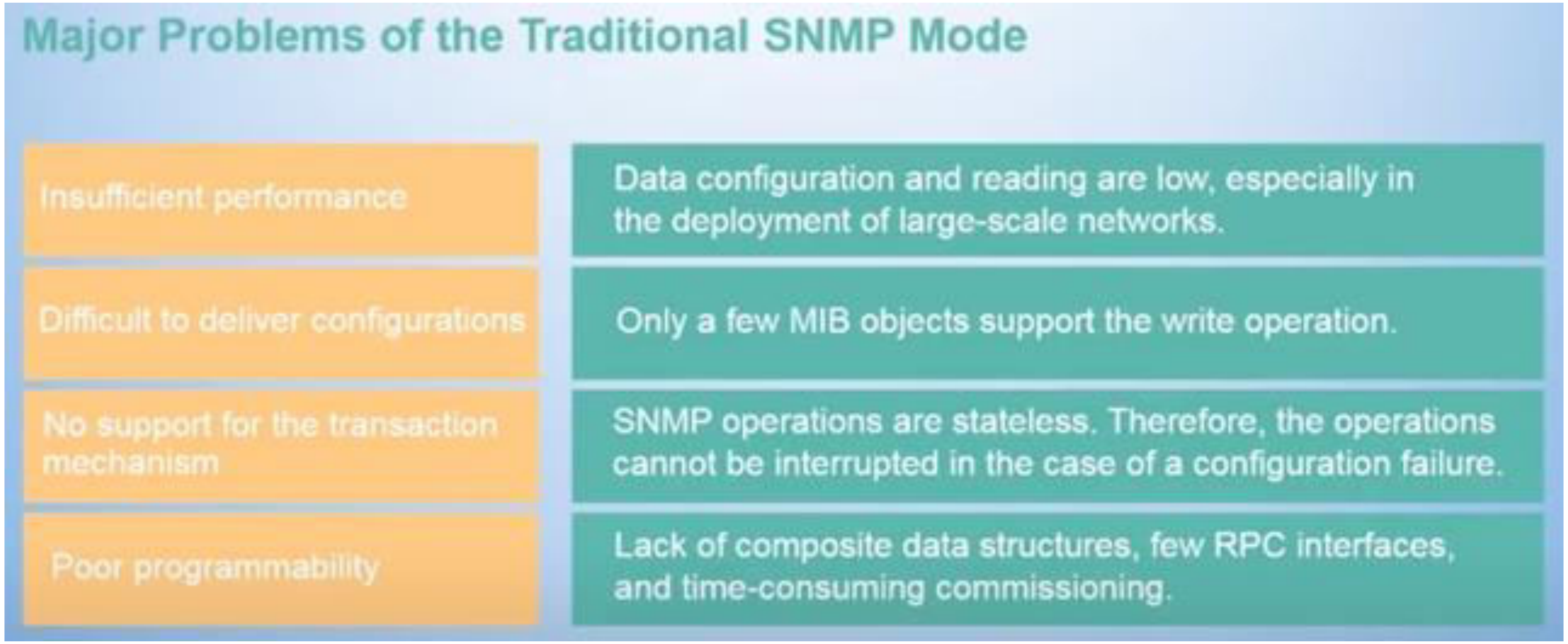
- In NETCONF and YANG scenarios, the corresponding application focuses on the definition differences between device of different vendors.
- Engineers DO NOT need to pay atention to the definition of the YANG models and the differences between YANG models
- The focus is shifted from the device and function differences, to USER requirement.
- The engineers can perform automatic configuration only by operating graphical application

YANG: Yet Another Next Generation is a data modeling language for the definition of data sent over network management protocols such as the NETCONF and RESTCONF. The YANG data modeling language is maintained by the NETMOD working group in the IETF. Initially was published as RFC 6020 in 2010, with an update in 2016 (RFC 7950). The data modeling language can be used to model both configuration data as well as state data of network elements. Furthermore, YANG can be used to define the format of event notifications emitted by network elements and it allows data modelers to define the signature of RPC (remote procedure calls) that can be invoked on network elements via the NETCONF protocol. The language, being protocol independent, can then be converted into any encoding format, e.g. XML or JSON, that the network configuration protocol supports.

Fuente: Huawei

SNMP LIMITATIONS

The SNMP configuration effiency is low and the transaction mechanism is not soported.
There are NOT large number of MIB objects supporting the write operation
Therefore SNMP often used for monitoring



Major Problems of the Traditional SNMP Mode

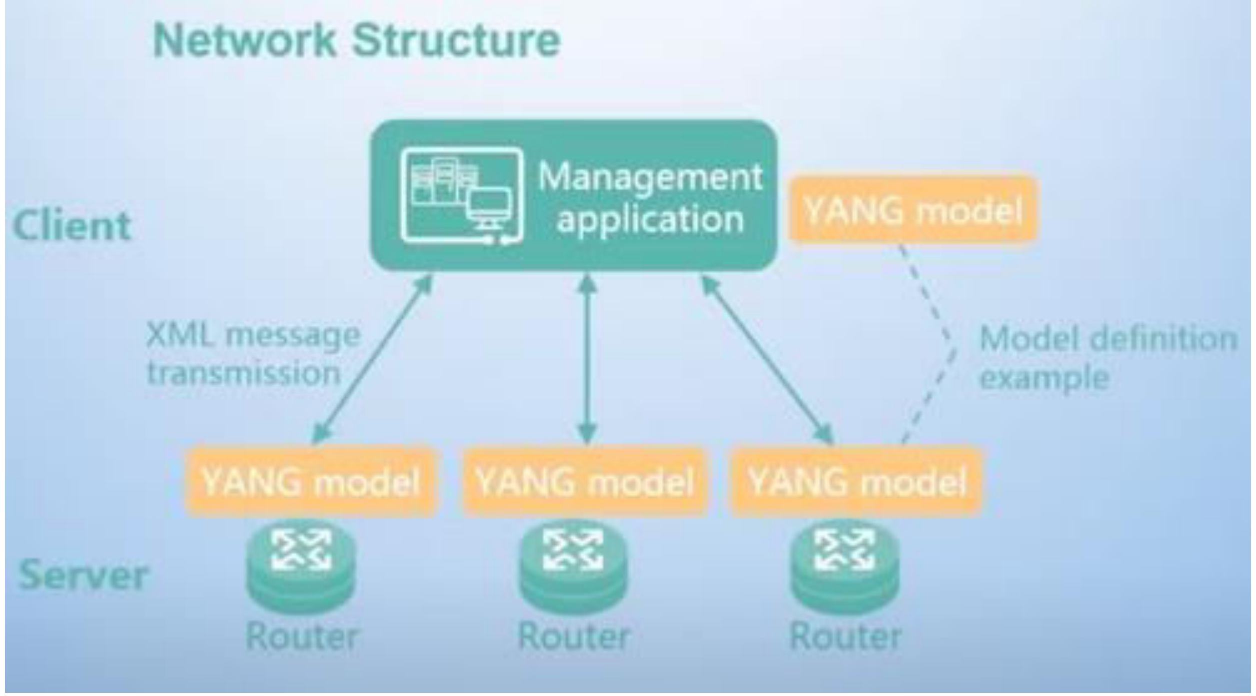
Insufficient performance	Data configuration and reading are low, especially in the deployment of large-scale networks.
Difficult to deliver configurations	Only a few MIB objects support the write operation.
No support for the transaction mechanism	SNMP operations are stateless. Therefore, the operations cannot be interrupted in the case of a configuration failure.
Poor programmability	Lack of composite data structures, few RPC interfaces, and time-consuming commissioning.

Fuente: Huawei

NETCONF- MAIN FEATURES AND IMPLEMETATION PRINCIPLES

The SNMP uses the Client/server comunication mode.
Generally a NMC, controller, or application functions as a client.

A network device such a switch or a router functions as a server. The same YANG model exists on the client and the server. Based on the same YANG model the client generates XML packets that compliant with the NETCONF communication requirements and the server identifies the XML packets and performs the related operations to achieve communication



Fuente: Huawei

NMC: Network management center

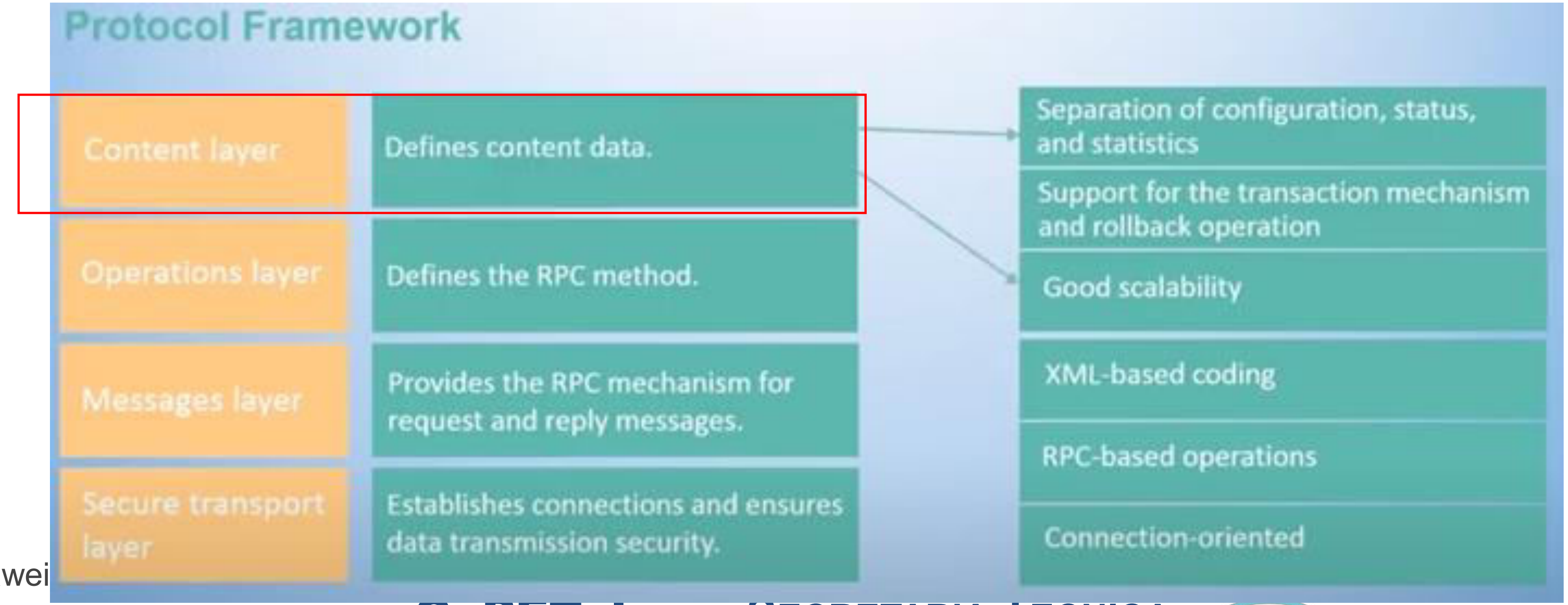
NETCONF- MAIN FEATURES AND IMPLEMETATION PRINCIPLES

NETCONF uses a hierarchical protocol framework including the “Content layer”, the “Operations layer”, the “Message layer”, the “Secure transport layer”.

Layer 4- Content layer defines data models such as YANG models. At this layer, configuration, status, and statistics are isolated. You can query each type of information separately and perform batch operations.
The configuration and query speed are faster than in SNMP.

In RFC 4741 (2006) the Configuration Information format was not specified

The language for data modeling YANG was specified in RFC 6020 in 2010



Protocol Framework

Content layer	Defines content data.
Operations layer	Defines the RPC method.
Messages layer	Provides the RPC mechanism for request and reply messages.
Secure transport layer	Establishes connections and ensures data transmission security.

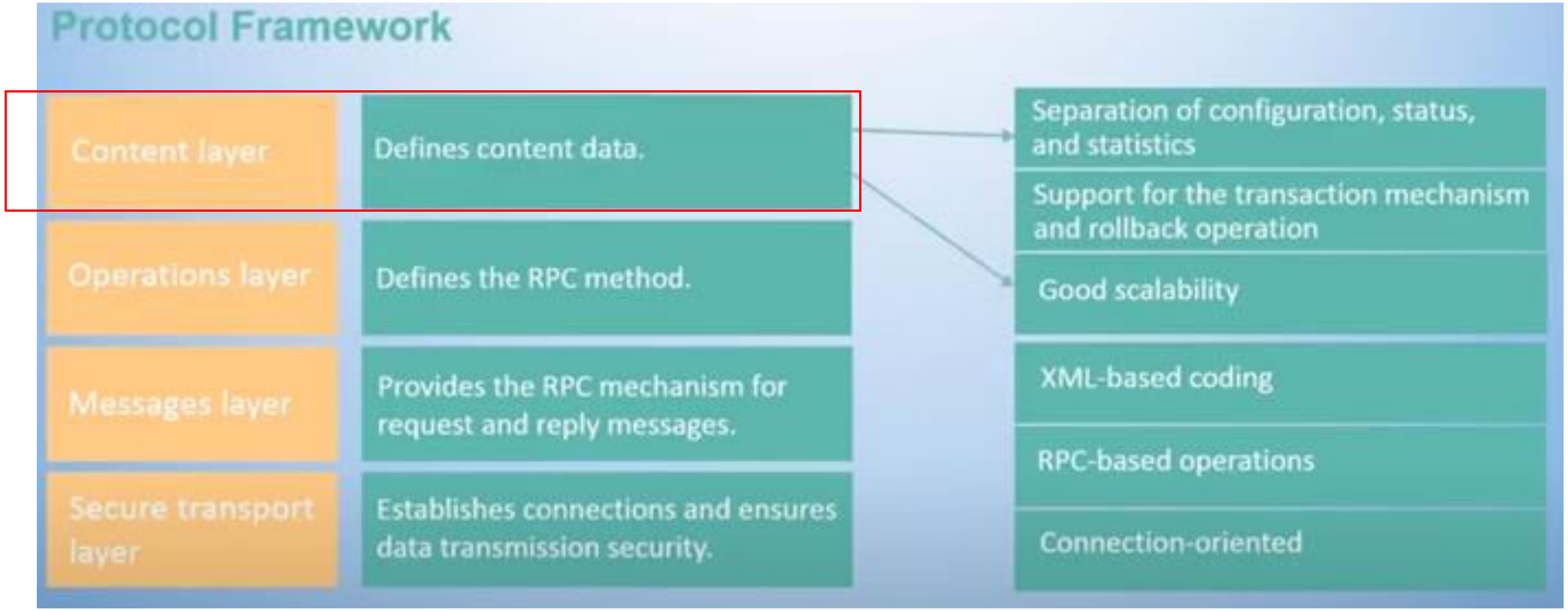
- Separation of configuration, status, and statistics
- Support for the transaction mechanism and rollback operation
- Good scalability
- XML-based coding
- RPC-based operations
- Connection-oriented

Fuente: Huawei

NETCONF- MAIN FEATURES AND IMPLEMETATION PRINCIPLES

Layer 4- Content layer

YANG specifies the data model and the operations performed by layers “Operation” and “Content” of NETCONF. It can be said that YANG covers these two layers



Protocol Framework

Content layer	Defines content data.
Operations layer	Defines the RPC method.
Messages layer	Provides the RPC mechanism for request and reply messages.
Secure transport layer	Establishes connections and ensures data transmission security.

- Separation of configuration, status, and statistics
- Support for the transaction mechanism and rollback operation
- Good scalability
- XML-based coding
- RPC-based operations
- Connection-oriented

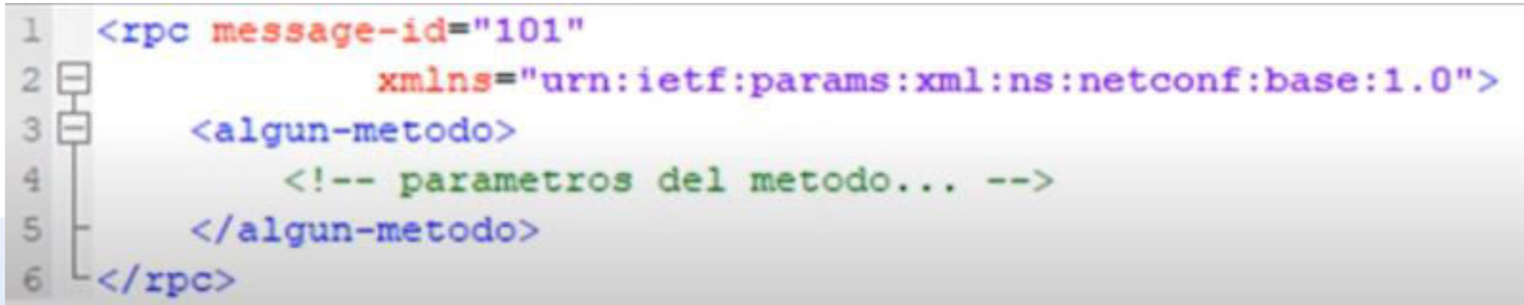
Fuente: Huawei

NETCONF- MAIN FEATURES AND IMPLEMETATION PRINCIPLES

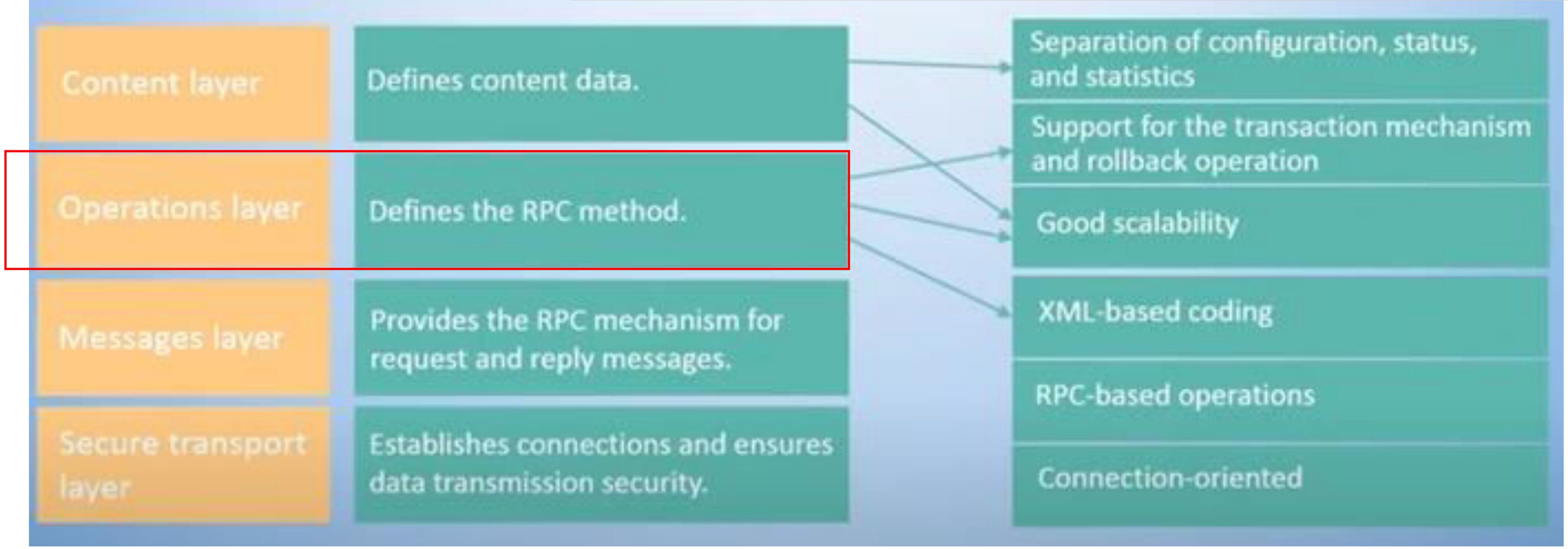
- Layer 3- Operation layer defines a series of basic protocol operations that can be invoked using the RPC method.
- Support transaction and roll back mechanisms.
- Therefore, configuration can be performed in different phases, interrupted or rolled back in the case of failure.
- All NETCONF messages must be “well-formed” XML.
- Must be coded in UTF-8
- All NETCONF are defined in following namespace

  urn:ietf:params:xml:ns:netconf:base:1.0

The <rpc> element is used to contain the request sent by client



```
<rpc message-id="101"
                xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <algun-metodo>
        <!-- parametros del metodo... -->
    </algun-metodo>
</rpc>
```



Protocol Framework

Content layer — Defines content data.

Operations layer — Defines the RPC method.

Messages layer — Provides the RPC mechanism for request and reply messages.

Secure transport layer — Establishes connections and ensures data transmission security.

Separation of configuration, status, and statistics

Support for the transaction mechanism and rollback operation

Good scalability

XML-based coding

RPC-based operations

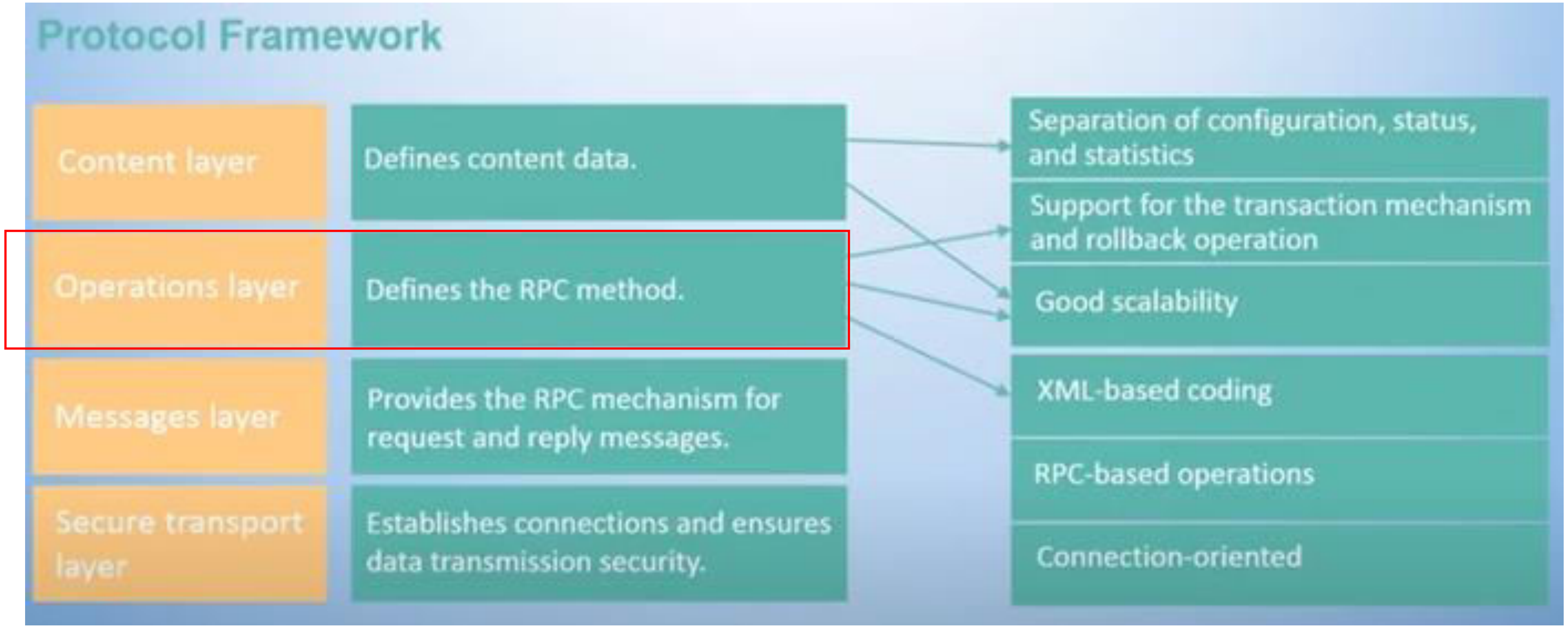
Connection-oriented

Fuente: Huawei

NETCONF- MAIN FEATURES AND IMPLEMETATION PRINCIPLES

- Layer 3- Operation layer defines a series of basic protocol operations that can be invoked using the RPC method.
- Support transaction and roll back mechanisms.
- Therefore, configuration can be performed in different phases, interrupted or rolled back in the case of failure.
- All NETCONF messages must be “well-formed” XML.
- Must be coded in UTF-8
- All NETCONF are defined in following namespace

urn:ietf:params:xml:ns:netconf:base:1.0



Protocol Framework

Content layer — Defines content data.

Operations layer — Defines the RPC method.

Messages layer — Provides the RPC mechanism for request and reply messages.

Secure transport layer — Establishes connections and ensures data transmission security.

Separation of configuration, status, and statistics

Support for the transaction mechanism and rollback operation

Good scalability

XML-based coding

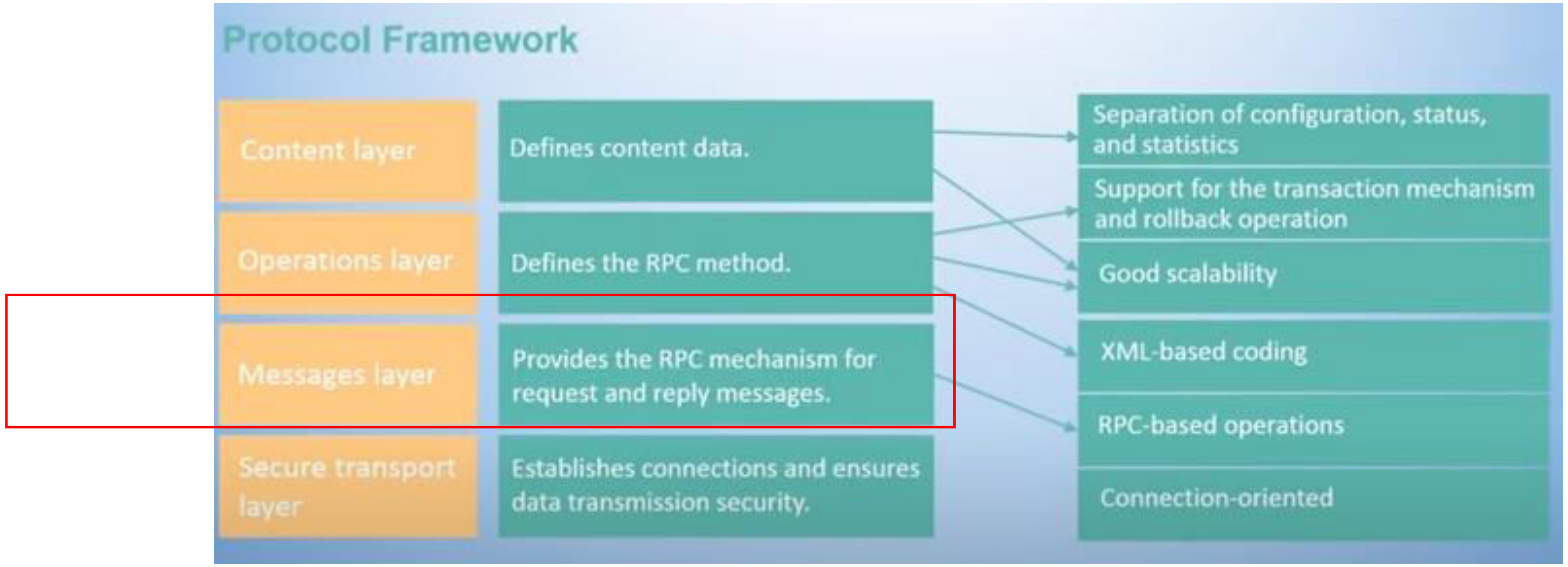
RPC-based operations

Connection-oriented

Fuente: Huawei

Layer 2- Message layer

- Message layer provides a simple and independent transmission mechanism for RPC and notificactions
- Message layer supports framing mechanism, which allows to code RPCs and notifications.
- Entities uses <rpc> and <rpc-replay> to format NETCONF requests and answers



Protocol Framework

Content layer	Defines content data.	Separation of configuration, status, and statistics
		Support for the transaction mechanism and rollback operation
Operations layer	Defines the RPC method.	Good scalability
Messages layer	Provides the RPC mechanism for request and reply messages.	XML-based coding
		RPC-based operations
Secure transport layer	Establishes connections and ensures data transmission security.	Connection-oriented

Fuente: Huawei

NETCONF- MAIN FEATURES AND IMPLEMETATION PRINCIPLES

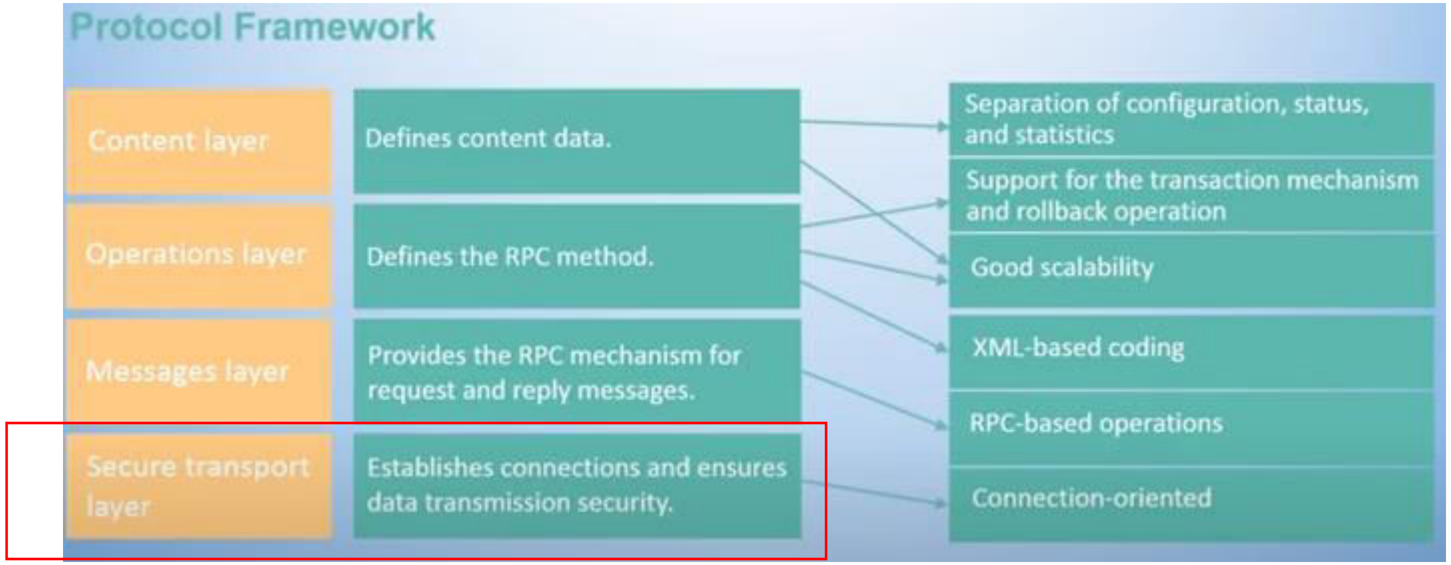
Layer 2- Secure transport layer provides a communicaction channel between the client and the server, supporting authentication, ecription and integrity check.

Client sends some messages or requests RPCs. Server answer with RPC replay

NETCONF can be transported by any transport protocol that meet a serie of requirements (from RFC), they are:

- Connection oriented
- persistent connection between entities
- Provide authentication, data integrity, confidentiality and protection replay

NETCONF implementation must support at least SSH protocol. Also TLS, SOAP/HTTP/TLS



Protocol Framework

Content layer — Defines content data.

Operations layer — Defines the RPC method.

Messages layer — Provides the RPC mechanism for request and reply messages.

Secure transport layer — Establishes connections and ensures data transmission security.

Separation of configuration, status, and statistics

Support for the transaction mechanism and rollback operation

Good scalability

XML-based coding

RPC-based operations

Connection-oriented

Fuente: Huawei

NETCONF- MAIN FEATURES AND IMPLEMETATION PRINCIPLES

DATA STORE:

Configuration Data store consist of a set of configuration data that is required to carry a device from an initial state to a desired state. Configuration Data store is the device Configuration

In the NETCONF base model, only the running Data Store is defined, which must allways be present in all Network devices.

In addition, and depending on vendor policies, a Candidate Data Store and Startup Data Store can also exist

Fuente: Huawei

NETCONF- MAIN FEATURES AND IMPLEMETATION PRINCIPLES

Netconf supports classified data storage and migration involving <running/>, <candidate/> and <startup/> configuration data store.

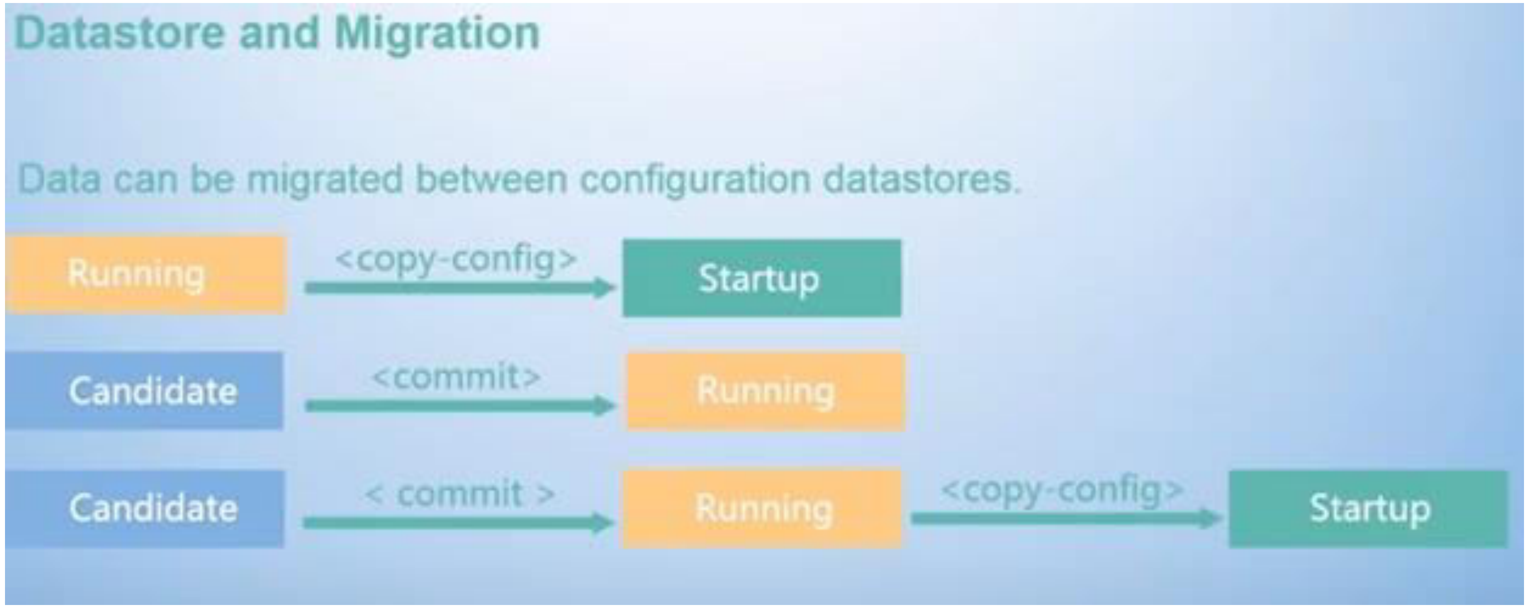
The <running/> configuration data store, stores the complete set of active configurations of a network device.

The <candidate/> configuration data store, stores various configurations data of a network device to be commited to the <running/> configuration data store. Changes in the The <candidate /> configuration data store, DO NOT directly affect the envolved device.

The <startup/> configuration data store, stores configuration data loaded during device startup.

The configuration data can be considered a saved configuration file

The configuration data is isolated between the data store and can often be migrated between the data store through different operations



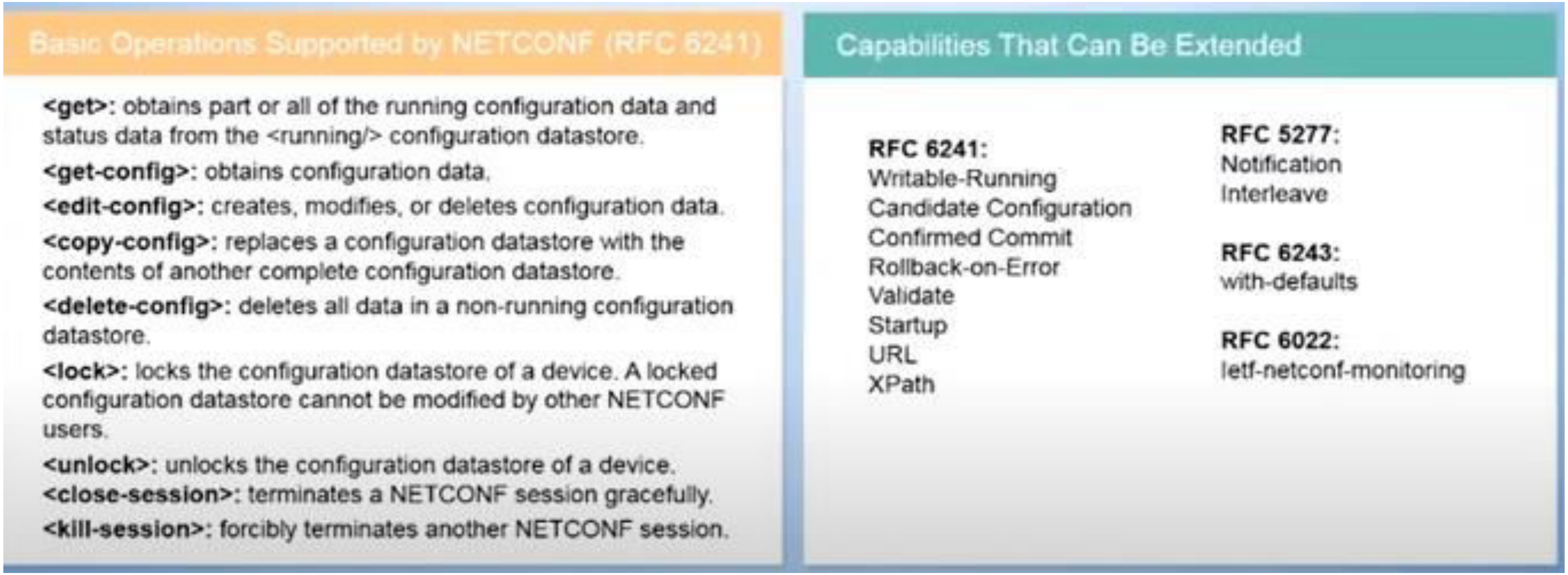
Datastore and Migration

Data can be migrated between configuration datastores.

Fuente: Huawei

NETCONF- BASIC OPERATIONS AND CAPABILITIES

Netconf defines a lot of operation interfaces and support extentions based on capability sets. With these operations interfaces, NETCONF can perform various operations on devices to meet requirements in different use scenarios



Basic Operations Supported by NETCONF (RFC 6241)

<get>: obtains part or all of the running configuration data and status data from the <running/> configuration datastore.

<get-config>: obtains configuration data.

<edit-config>: creates, modifies, or deletes configuration data.

<copy-config>: replaces a configuration datastore with the contents of another complete configuration datastore.

<delete-config>: deletes all data in a non-running configuration datastore.

<lock>: locks the configuration datastore of a device. A locked configuration datastore cannot be modified by other NETCONF users.

<unlock>: unlocks the configuration datastore of a device.

<close-session>: terminates a NETCONF session gracefully.

<kill-session>: forcibly terminates another NETCONF session.

Capabilities That Can Be Extended

RFC 6241:
Writable-Running
Candidate Configuration
Confirmed Commit
Rollback-on-Error
Validate
Startup
URL
XPath

RFC 5277:
Notification
Interleave

RFC 6243:
with-defaults

RFC 6022:
Ietf-netconf-monitoring

Fuente: Huawei

NETCONF- EXAMPLE

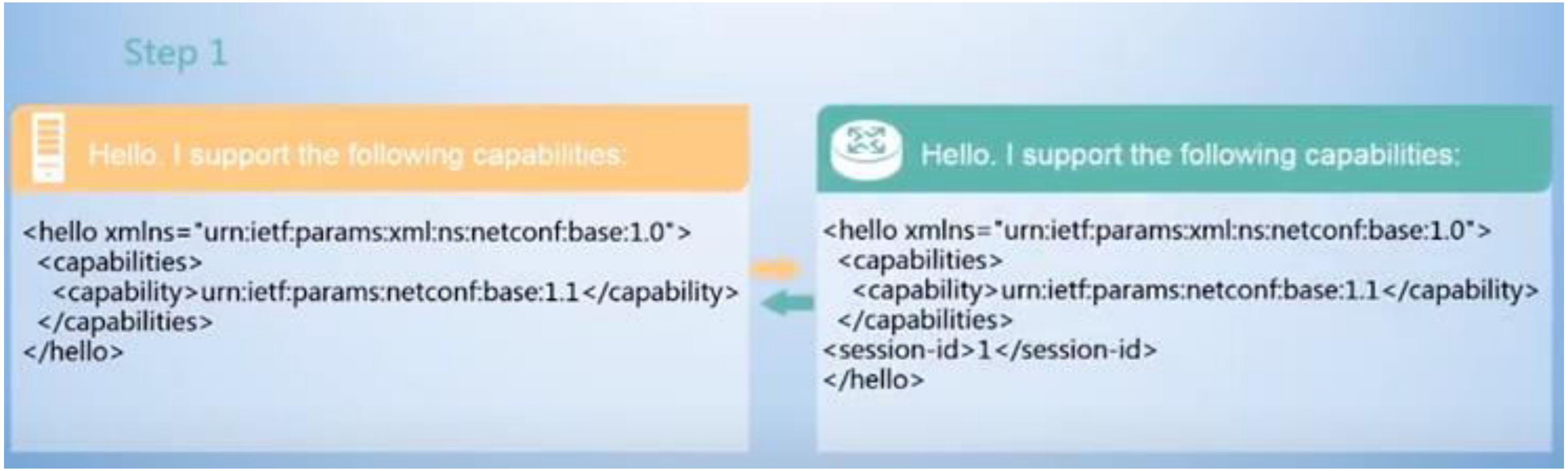
Let’s use a two phase configuration example to see the basic the NETCONF session process.

Suppose that the user wants to configure the IP address for a device interface to the client After the SSH connection, authentication, and the autorization is completed.

Client
Server

STEP 1

The first step is to initiate the NETCONF session stablishment on the client and to advertise the capabilities through “hello” messages.



Step 1

Hello. I support the following capabilities:

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>urn:ietf:params:netconf:base:1.1</capability>
  </capabilities>
</hello>
```

Hello. I support the following capabilities:

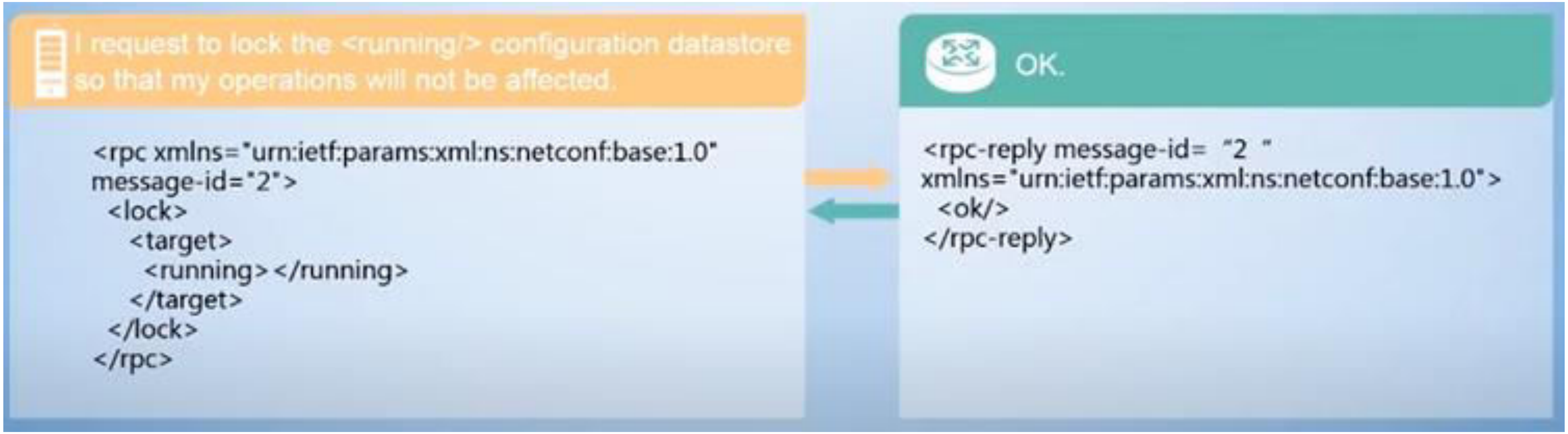
```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>urn:ietf:params:netconf:base:1.1</capability>
  </capabilities>
<session-id>1</session-id>
</hello>
```

Fuente: Huawei

NETCONF- EXAMPLE

STEP 2

After capabilities negotiations successed, the client needs to request to lock running configuration datastore so that its operations will not be affected by other client



I request to lock the <running/> configuration datastore so that my operations will not be affected.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
message-id="2">
  <lock>
    <target>
      <running></running>
    </target>
  </lock>
</rpc>
```

OK.

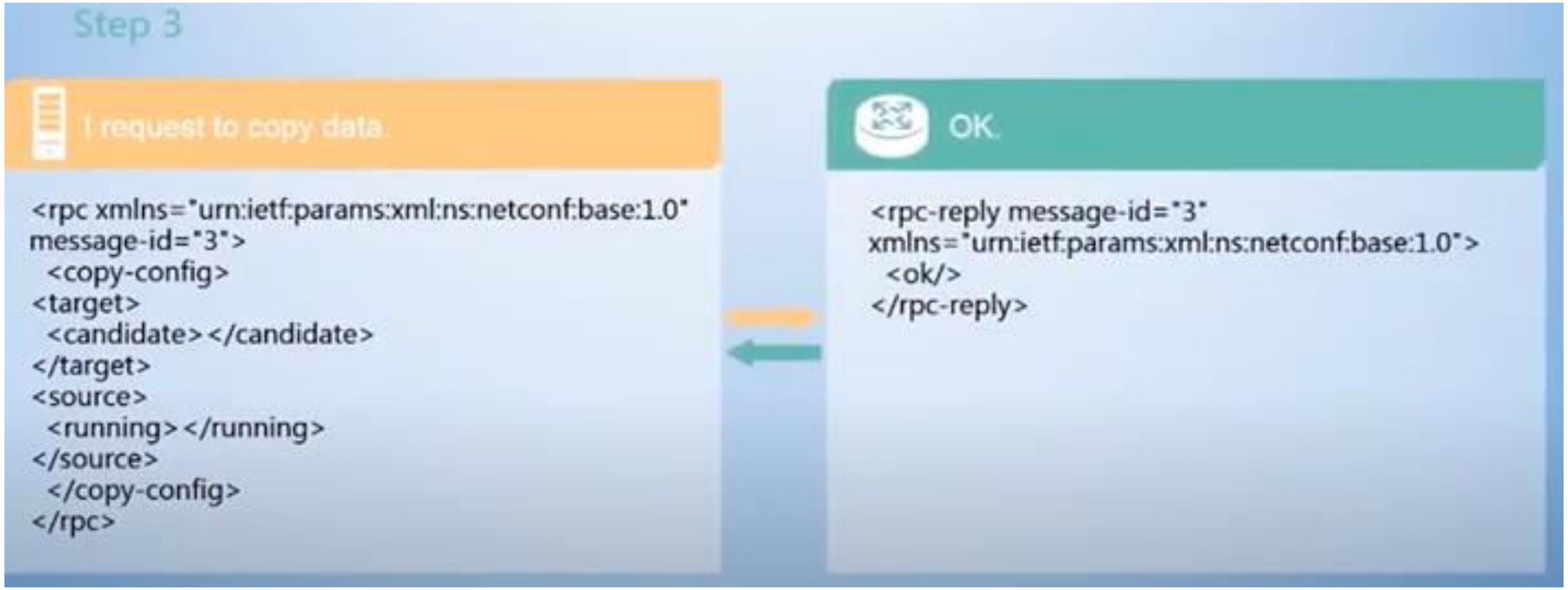
```
<rpc-reply message-id= “2 ”
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

Fuente: Huawei

NETCONF- EXAMPLE

STEP 3

The third step is to copy the data in the running configuration data store to the candidate configuration data store to be sure that the configurations are the latest



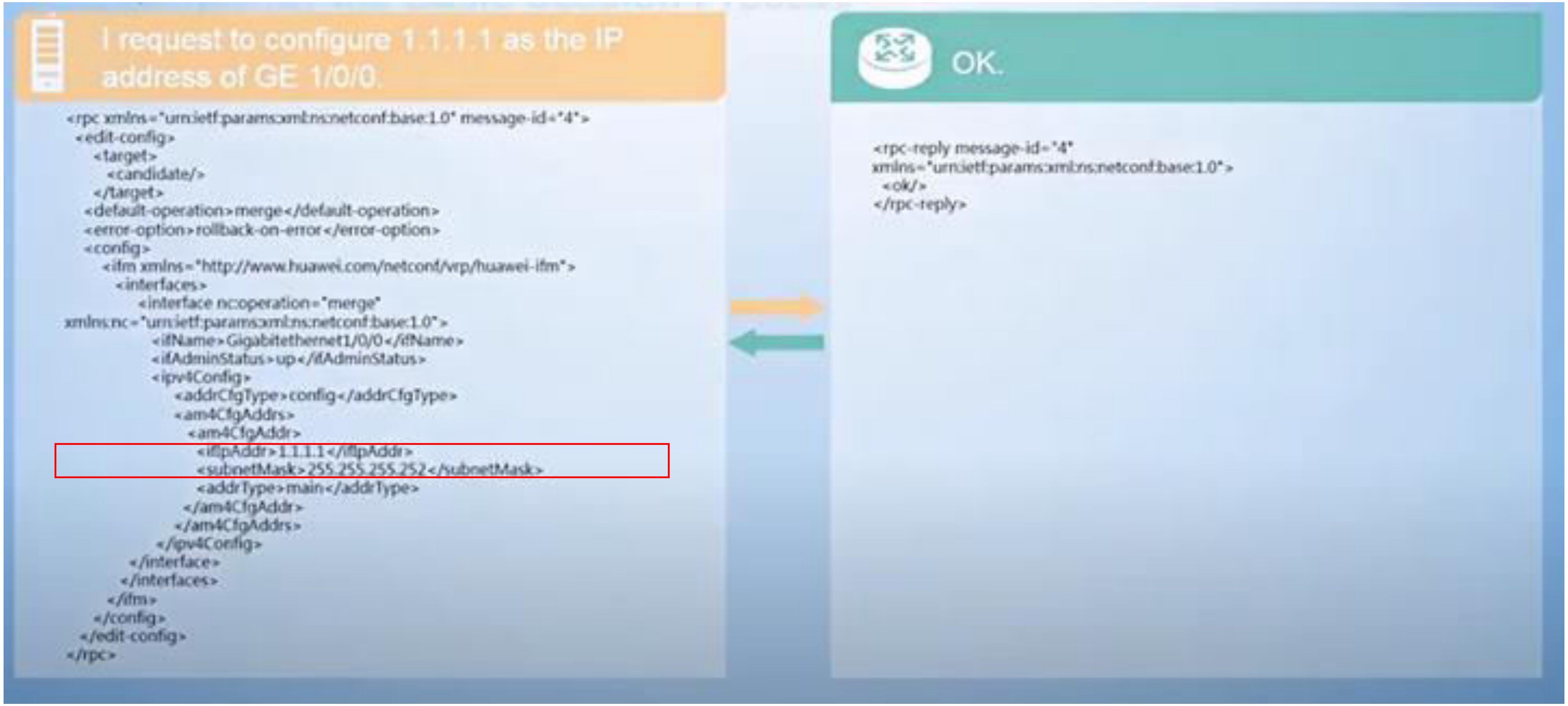
```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
message-id="3">
  <copy-config>
<target>
  <candidate></candidate>
</target>
<source>
  <running></running>
</source>
  </copy-config>
</rpc>
```

```
<rpc-reply message-id="3"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

Fuente: Huawei

STEP 4

The four step is to edit configurations in the candidate configuration data store



I request to configure 1.1.1.1 as the IP address of GE 1/0/0.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="4">
 <edit-config>
   <target>
     <candidate/>
   </target>
  <default-operation>merge</default-operation>
  <error-option>rollback-on-error</error-option>
  <config>
    <ifm xmlns="http://www.huawei.com/netconf/vrp/huawei-ifm">
      <interfaces>
         <interface nc:operation="merge"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
           <ifName>Gigabitethernet1/0/0</ifName>
           <ifAdminStatus>up</ifAdminStatus>
           <ipv4Config>
             <addrCfgType>config</addrCfgType>
             <am4CfgAddrs>
               <am4CfgAddr>
                <ifIpAddr>1.1.1.1</ifIpAddr>
                <subnetMask>255.255.255.252</subnetMask>
                <addrType>main</addrType>
               </am4CfgAddr>
             </am4CfgAddrs>
           </ipv4Config>
         </interface>
       </interfaces>
     </ifm>
   </config>
  </edit-config>
</rpc>
```

OK.

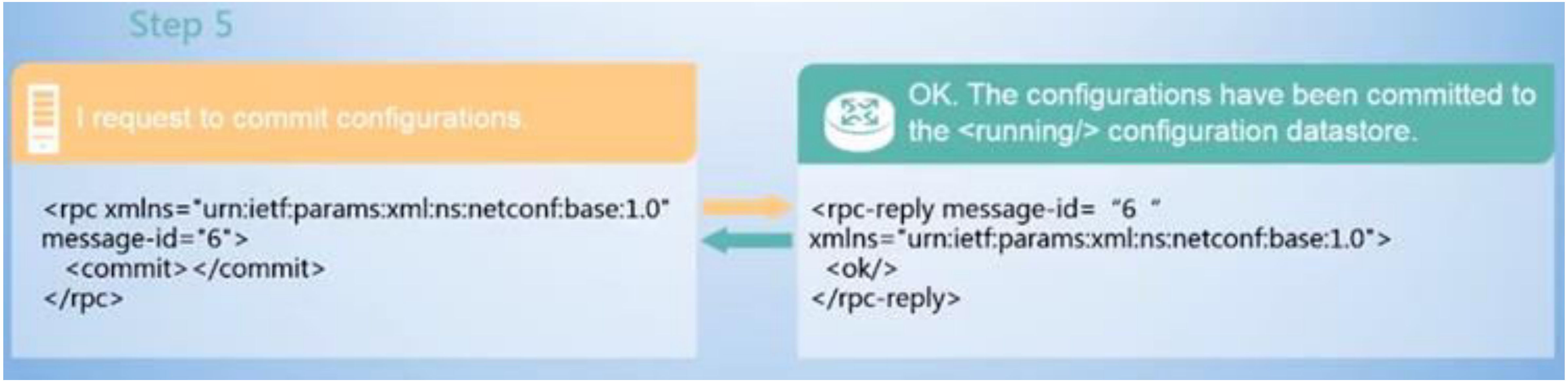
```
<rpc-reply message-id="4"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

Fuente: Huawei

NETCONF- EXAMPLE

STEP 5

The fifth step is to commit the configurations in the candidate configuration data store to the running configuration data store



Step 5

I request to commit configurations.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
message-id="6">
   <commit></commit>
</rpc>
```

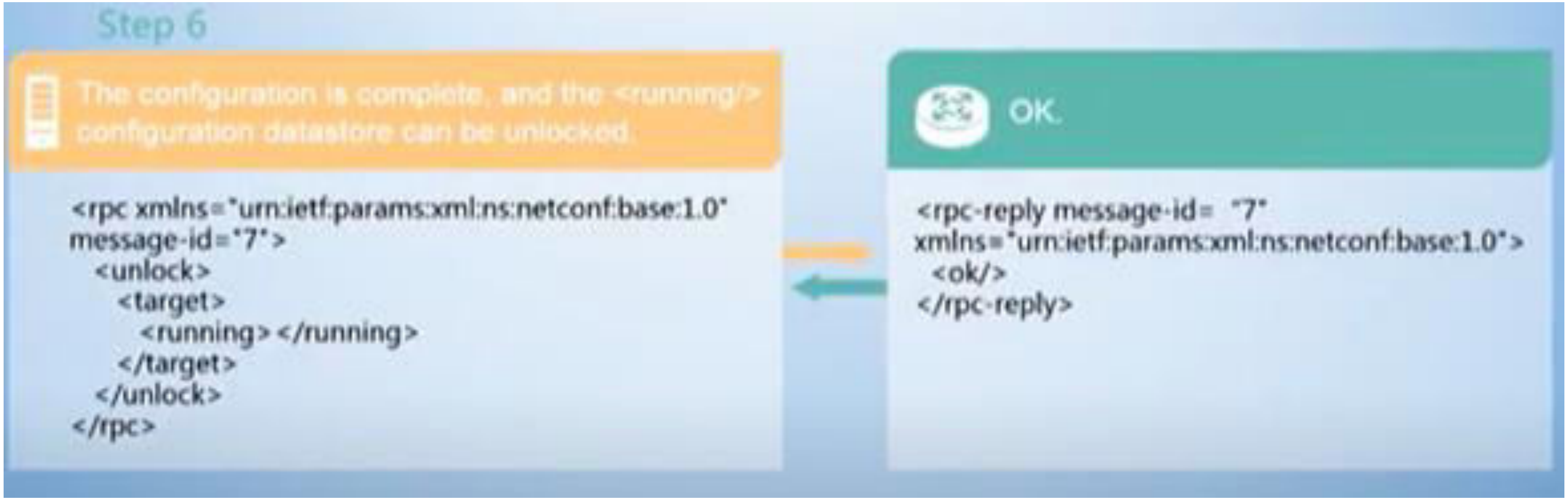
OK. The configurations have been committed to the <running/> configuration datastore.

```
<rpc-reply message-id=  "6  "
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

Fuente: Huawei

STEP 6

The six step is to unlock running configuration data store. Finally terminates the NETCONF sesión and tier down the SSH connection.



Step 6

The configuration is complete, and the <running/> configuration datastore can be unlocked.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
message-id="7">
  <unlock>
    <target>
      <running></running>
    </target>
  </unlock>
</rpc>
```

OK.

```
<rpc-reply message-id= "7"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

Fuente: Huawei