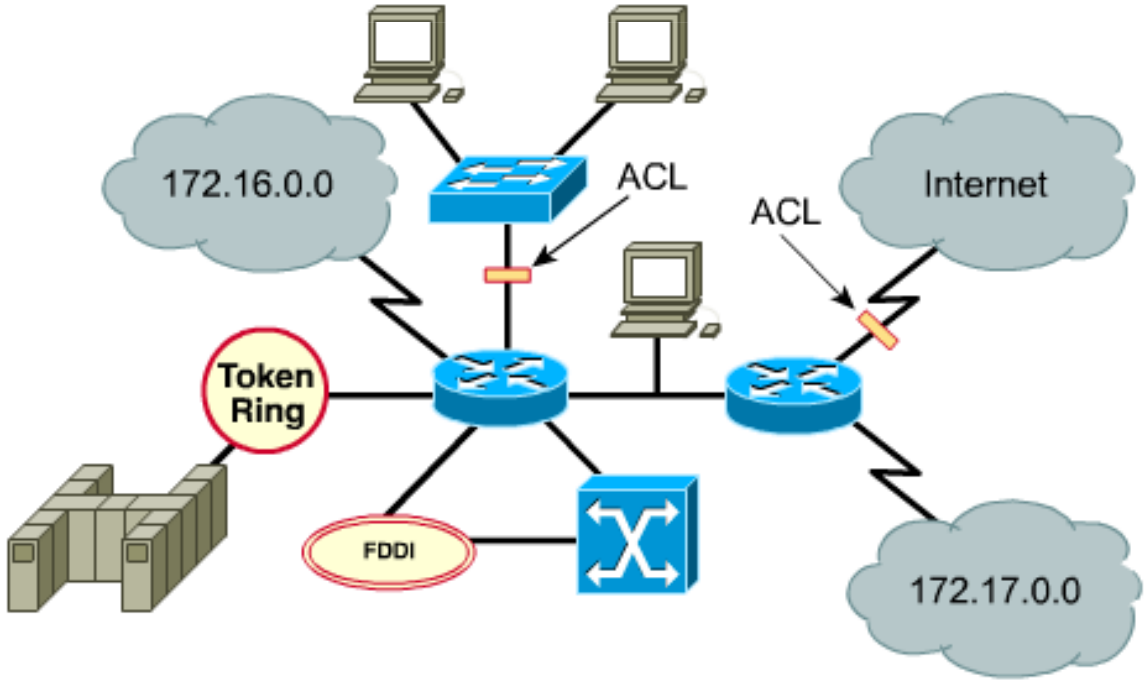
REDDIG II – Computer Networking Training

Access Control Lists (ACLs)

Access Control Lists (ACLs)

What is an Access List?

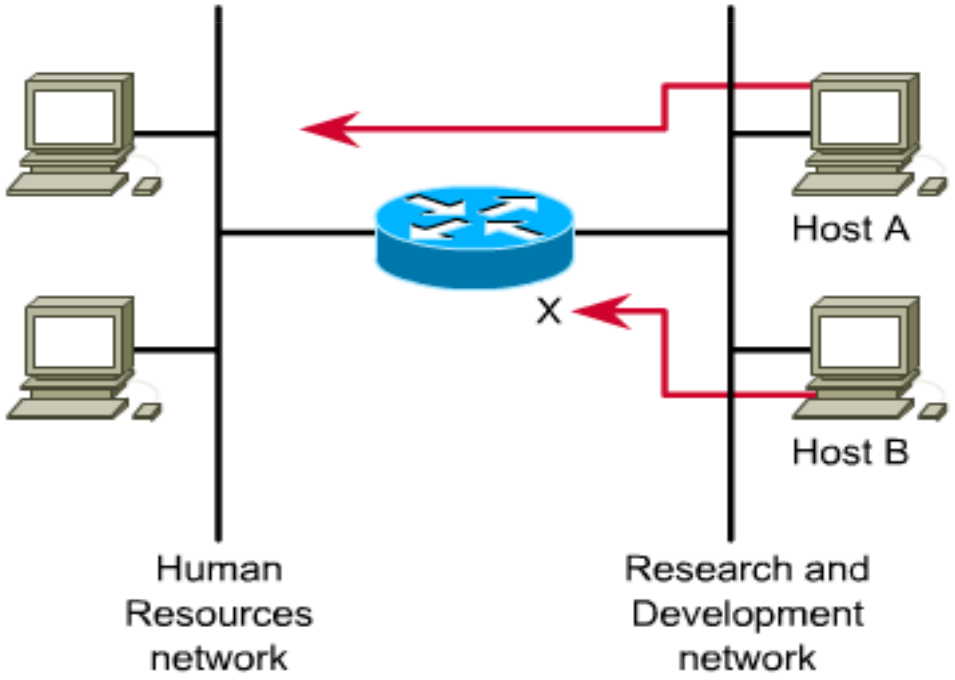
- ACLs are lists of conditions that are applied to traffic traveling across a router's interface.
- These lists tell the router what types of packets to accept or deny.
- Acceptance and denial can be based on specified conditions.
- ACLs enable management of traffic and secure access to and from a network.



Access Control Lists (ACLs)

Reasons to Create ACLs

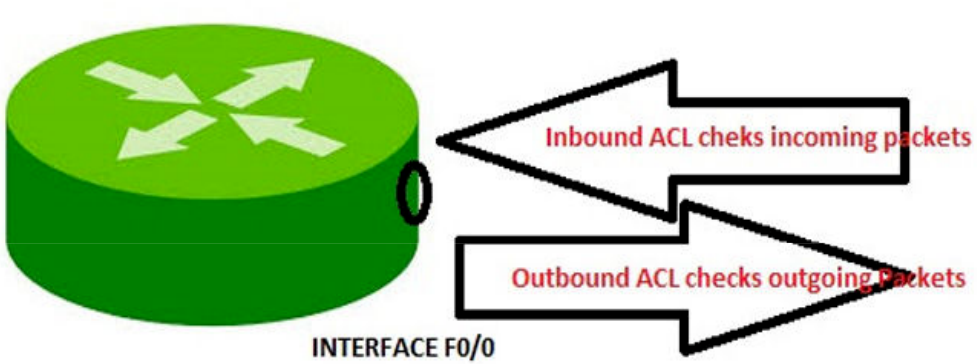
- Limit network traffic and increase network performance
- Provide traffic flow control
- Provide a basic level of security for network access
- Decide which types of traffic are forwarded or blocked at the router interfaces



Access Control Lists (ACLs)

How ACLs Work

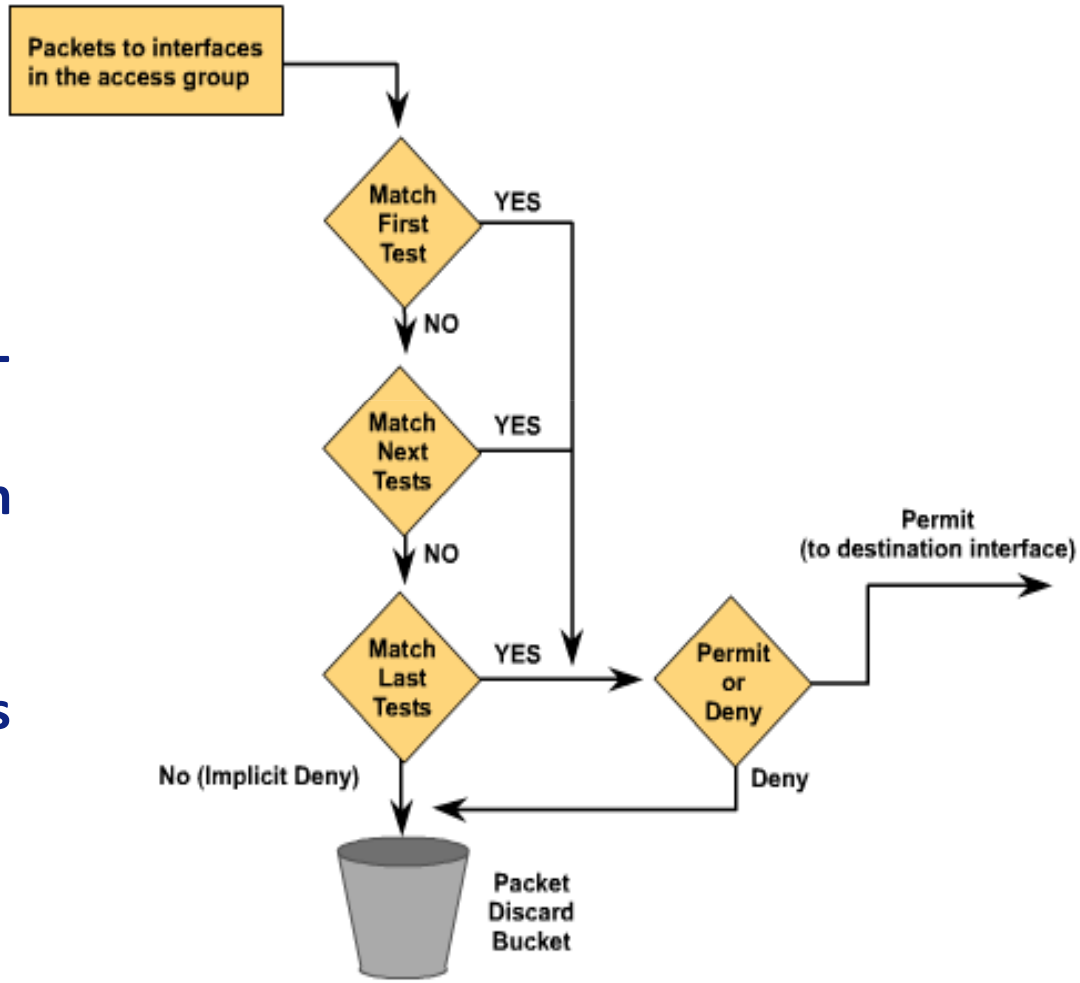
- ACLs must be defined on a:
  - per protocol (IP, IPX, AppleTalk)
  - per direction (in or out)
  - per port (interface) basis.
- ACLs control traffic in one direction at a time on an interface.
- A separate ACL would need to be created for each direction, one for inbound and one for outbound traffic.
- Finally every interface can have multiple protocols and directions defined.



Access Control Lists (ACLs)

How ACLs work

- ACL statements operate in sequential, logical order.
- If a condition match is true, the packet is permitted or denied and the rest of the ACL statements are not checked.
- If all the ACL statements are unmatched, an implicit *"deny any"* statement is placed at the end of the list by *default*. (not visible)
- When first learning how to create ACLs, it is a good idea to add the implicit deny at the end of ACLs to reinforce the dynamic presence of the command line.



Access Control Lists (ACLs)

Types of IP Access Lists

There are two types of IP access lists used:

- **Standard access lists (use numbers 1–99) :**

  The standard IP access lists use only the source IP address in an IP packet to filter the network. This basically permits or denies an entire protocol suite.

- **Extended access lists (use numbers 100–199) :**

  The extended access lists check for source and destination IP address, protocol field in the layer 3 header, and port number at the layer 4 header

  Can filter on:

  - **Source IP address**
  - **Destination IP address**
  - **Protocol (TCP, UDP)**
  - **Port Numbers (Telnet – 23, http – 80, etc.)**
  - ***and other parameters***

Access Control Lists (ACLs)

Wildcard Mask

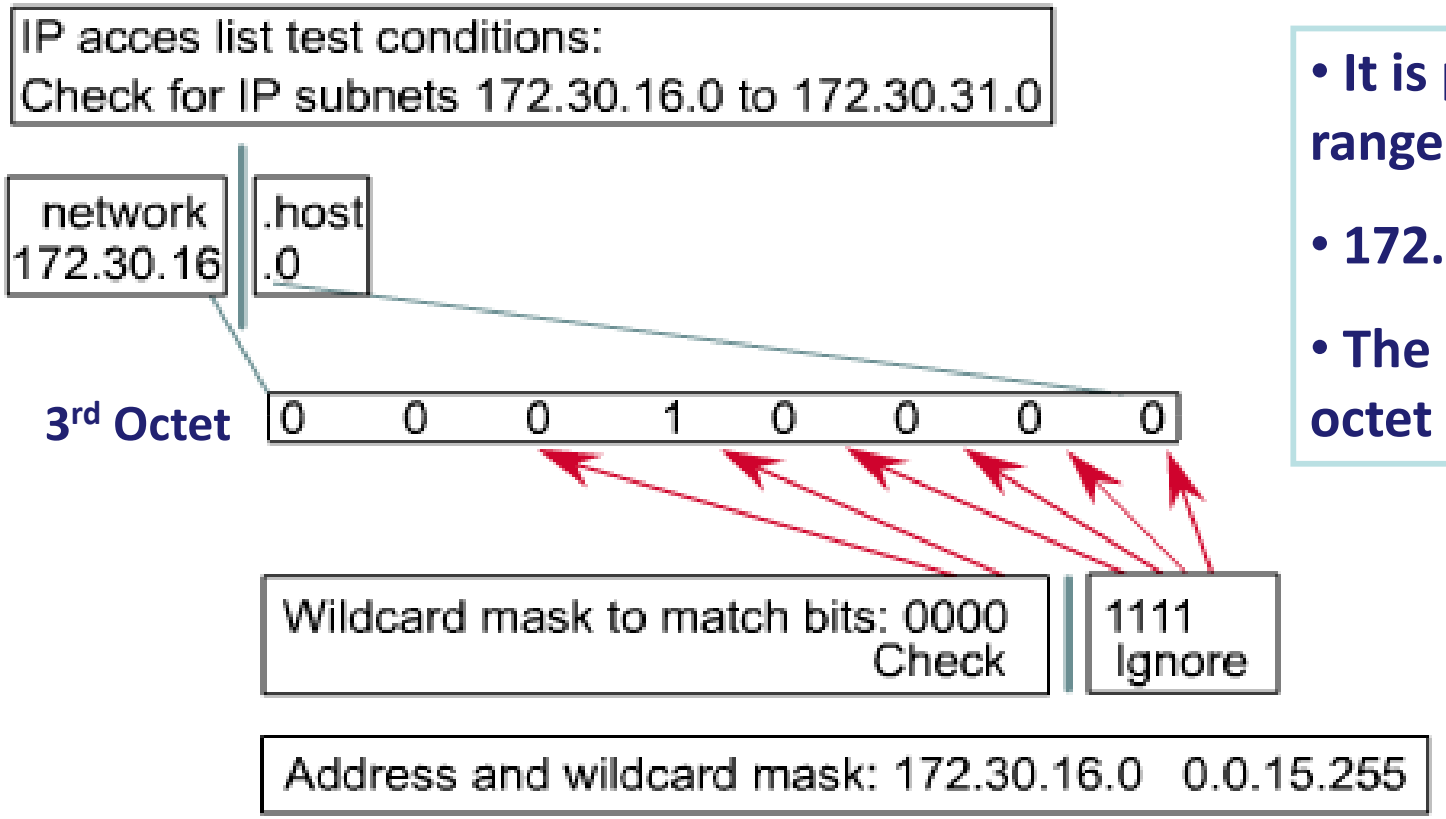
- Wildcard masks are designed to filter individual or groups of IP addresses permitting or denying access to resources based on the address.
- It is often easiest to see wildcard masks as being just the opposite of subnet masks. The subnet mask 255.255.255.0 identifies the network, the wildcard mask 0.0.0.255 identifies the hosts.
- A wildcard is a 32-bit value made up of contiguous 0s indicating a “must match” and then 1s.
- Example:

```
>Router#config t
>Router(config)#access-list 50 deny 192.168.1.0 0.0.0.255
```

- In the above example, the 0s mean the first 3 octets must match exactly. The 255 means any value from 0 to 255 can be here.

Access Control Lists (ACLs)

Wildcard Mask Example



- It is possible to identify a range of subnets.
- 172.30.16.0 0.0.15.255
- The 255 means the last octet can be 0 to 255.

This means any value that can be made using 4 bits (0 to 15), so the range is 16+0 to 16+15.

Access Control Lists (ACLs)

Wildcard Mask - Special Cases

- Keywords Any and Host
  - Any
    - Same as 0.0.0.0 255.255.255.255
    - Example: access-list 12 permit any
  - Host
    - Same as *IP-address* 0.0.0.0
    - Example: 192.168.1.10 0.0.0.0
    - Example: access-list 12 deny host 192.168.1.10
    - Can be omitted in standard IP ACLs

```
Router(config)#access-list 75 deny host 192.168.1.10
Router(config)#access-list 75 deny 192.168.17.123
Router(config)#access-list 75 permit any
```

Access Control Lists (ACLs)

Basic Standart ACL Creation Steps

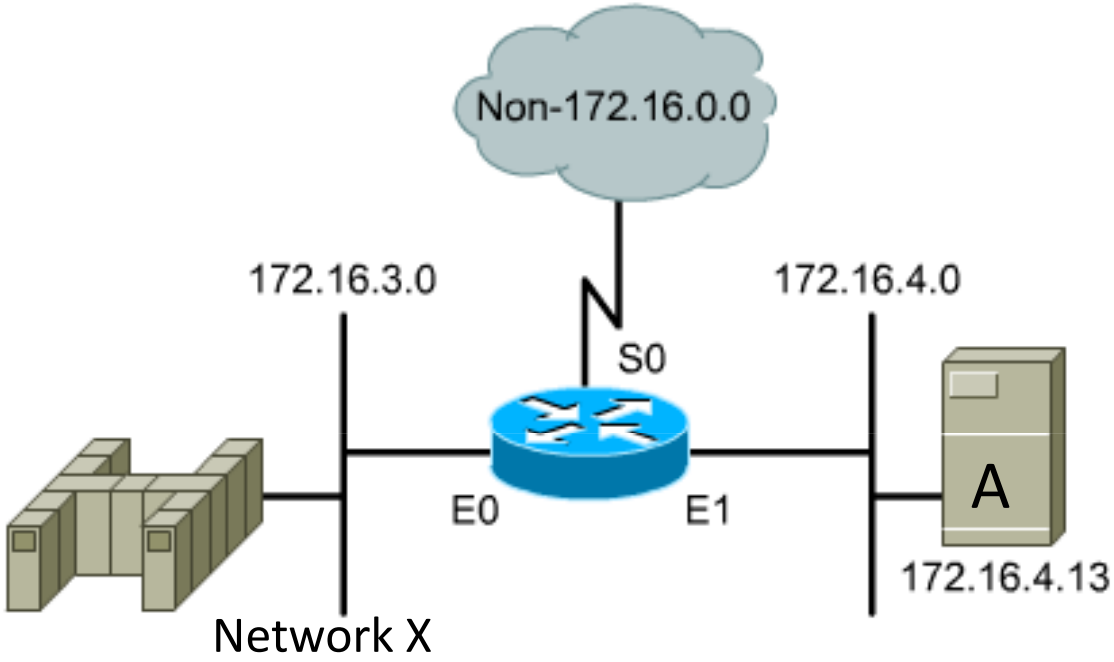
> Router#config t
> Router(config)#access-list 50 permit 192.168.1.10
> Router(config)#access-list 50 deny 192.168.1.0 0.0.0.255
> Router(config)#access-list 50 permit any
> Router(config)#interface Ethernet0
> Router(config-if)#ip address 192.168.5.1 255.255.255.0
> Router(config-if)#ip access-group 50 out << (applying the ACL)

- In the above example, we have a the Standart ACL (number 50). The ACL is applied "outbound" meaning :
  - Only outbound traffic is checked by the ACL.
    - > Out or in is always viewed from the center of the router.
- Keyword ip access-group applies ACL to an interface.
- Each interface can have an in and out ACL for each protocol configured on the interface.
- Keywords permit / deny allow or prevent passage, respectively.

Access Control Lists (ACLs)

Standart ACL Example

- This two-line ACL will deny traffic from a single host A from getting to the X network.
- The second line could and should have been access-list 1 permit any.
- The implicit *deny any* is negated by the previous line, which allowed every thing through



Command Output

```
access-list 1 deny 172.16.4.13  0.0.0.0
access-list 1 permit 0.0.0.0  255.255.255.255



interface ethernet 0
ip access-group 1 out
```

Access Control Lists (ACLs)

Extended ACL Creation Steps

- As with standard lists, the access-list command is used to create each condition of the list – using one condition per line. The syntax for each line in the list is:

```
> access-list access-list-number {permit | deny} {protocol | protocol
  keyword} {source | any} [source-wildcard] [source port] {destination
  | any} [destination-wildcard] [destination port] [options]
```

- Example:
  - Lab-X#config t
  - Lab-X(config)#Access-list 101 deny tcp 192.168.1.0 0.0.0.255 any eq www
  - Lab-X(config)#Access-list 101 deny tcp any eq ftp 192.168.1.25
  - Lab-X(config)#Access-list 101 permit ip any any
  - Lab-X(config)#interface Fastethernet 0/0
  - Lab-X(config-if)#ip access-group 101 out

- The protocol entry defines the protocol to be filtered, such as IP, TCP, UDP, or ICMP for example. Because IP headers transport TCP, UDP, and ICMP, it is important to specify the protocol or you could end up inadvertently filtering more than you want to

Access Control Lists (ACLs)

Extended ACLs – TCP Relational Operators

- The access list TCP protocol option supports both source and destination ports. You can access each by using either the port number or a mnemonic or acronym.
- Keyword relational operators such as those shown in the following code output precede these:

```
>Lab-X(config)#access-list 101 deny tcp any ?
  A.B.C.D  Destination address.
  any      Any destination host.
  eq       Match only packets on a given port number.
  gt       Match only packets with a greater port number.
  host     A single destination host.
  lt       Match only packets with a lower port number.
  neq      Match only packets not on a given port number.
  range    Match only packets in the range of port numbers.
```

Access Control Lists (ACLs)

Extended ACLs – TCP Example

- The first statement could have used the mnemonic “telnet” in place of 23 with exactly the same result. The one advantage to using the mnemonic is that it is more intuitive to anyone having to support the device.

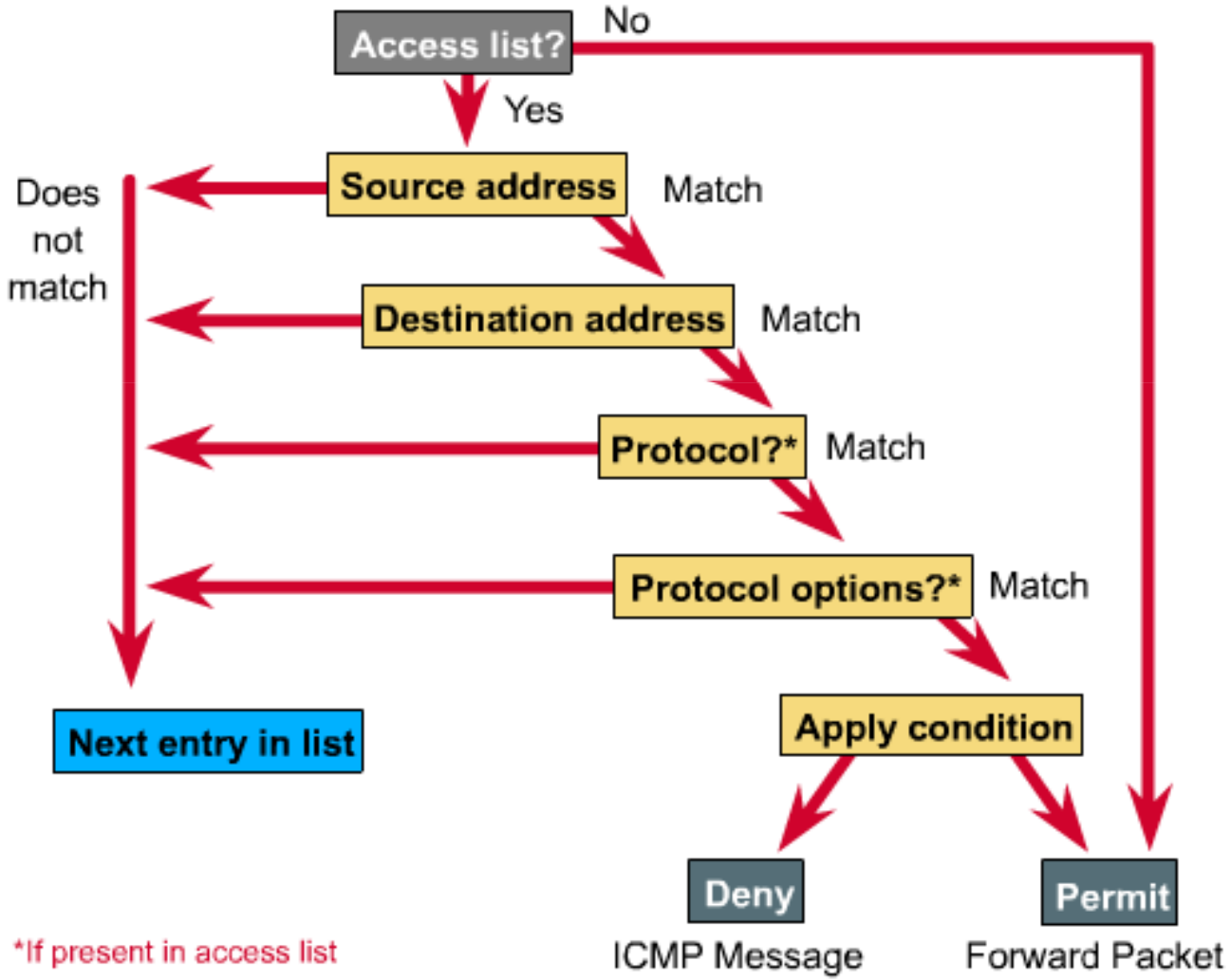
```
> Lab-X#config t
> Lab-X(config)#access-list 101 deny tcp 192.168.5.0 0.0.0.255 any eq 23
> Lab-X(config)#access-list 101 permit ip any any
> Lab-X(config)#interface fastethernet 0/1
> Lab-X(config-if)#ip access-group 101 in
```

If you want to block network 192.168.5.0 from being able to surf the Web while still allowing other services such as FTP, use this code:

```
> Lab-X#config t
> Lab-X(config)#access-list 106 deny tcp 192.168.5.0 0.0.0.255 any eq www
> Lab-X(config)#access-list 106 permit ip any any
> Lab-X(config)#interface ethernet 0
> Lab-X(config-if)#ip access-group 106 in
```

Access Control Lists (ACLs)

Extended Access List Processing



Access Control Lists (ACLs)

Named ACLs – Overview

- The Cisco IOS release supports using named access lists rather than the traditional number designations. This ability to name a list makes them easier to recognize and can make them easier to debug.
- Another advantage is that it is possible to delete individual entries from a specific ACL instead of erasing the entire list.
- A couple things to consider when implementing named ACLs:
    1. Names, like numbers, must be unique on each router.
    2. Named ACLs do not work with IOS releases prior to 11.2.
- The first step is to create the ACL using the following syntax:

```
Router(config)#ip access-list {standard | extended} name
```

- Example:

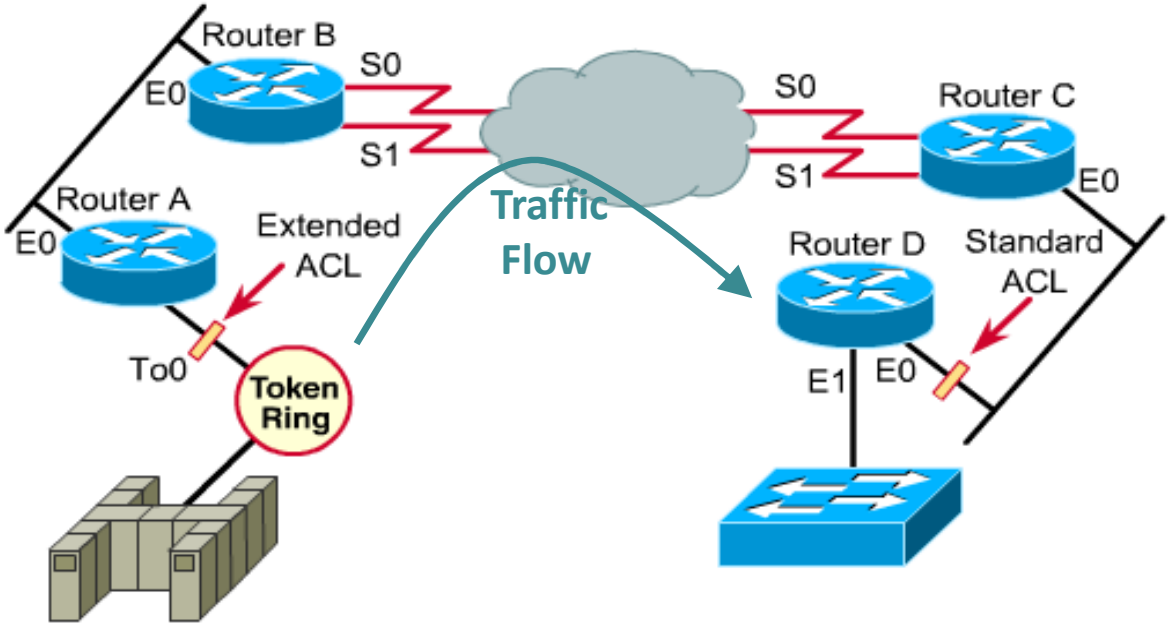
```
Lab-X(config)#ip access-list extended BlockInternet
```

Access Control Lists (ACLs)

ACL Placement

- Standard ACLs do not look at the destination address, therefore, you should place them closest to the destination network that you are filtering packets to.
- For the extended ACLs, you should place them closest to the source of the traffic.

Since they can filter traffic based on different types of criteria, it would be effective to place them on a router closest to the source of the traffic that is being filtered



Access Control Lists (ACLs)

Verifying ACLs

The following *commands can be used to check ACLs:*

- **show ip interface command will tell whether an inbound or outbound access list has been applied to an interface.**
- **show run command to see your access lists and how they are applied**
- **show access-lists command will display all access lists on the router but does not show whether or where they are applied.**
- **show ip access-lists command to see your named ACLs. Example:**

```
Router#show ip access-lists
Extended IP access list MyACL2
    10 permit tcp any any eq www
    20 permit tcp any any eq 443
    30 permit udp any host 192.0.2.1 eq domain
```

Access Control Lists (ACLs)

ACLs used in the REDDIG Network

```
SBCT-CISCO-VSAT-1-A-V15#sh access-lists
Standard IP access list 11
    10 permit 10.0.76.0, wildcard bits 0.0.0.255
Standard IP access list 12
    10 permit 10.0.78.0, wildcard bits 0.0.0.255
Standard IP access list 13
    10 permit 10.0.77.0, wildcard bits 0.0.0.255
Extended IP access list 118
    10 permit udp any eq 57 any eq 57
Extended IP access list 126
    10 permit tcp any any range 1963 1978 (899 matches)
Extended IP access list 127
    10 permit tcp any range 1963 1978 any (288 matches)
Extended IP access list 146
    10 permit udp any range 16384 19000 any range 16384 19000 (6554495 matches)
```

AMHS Services Curitiba

The default listening TCP ports for BSTUN

Ports used for VoIP applications

Network Address Translation (NAT)

Network Address Translation (NAT)

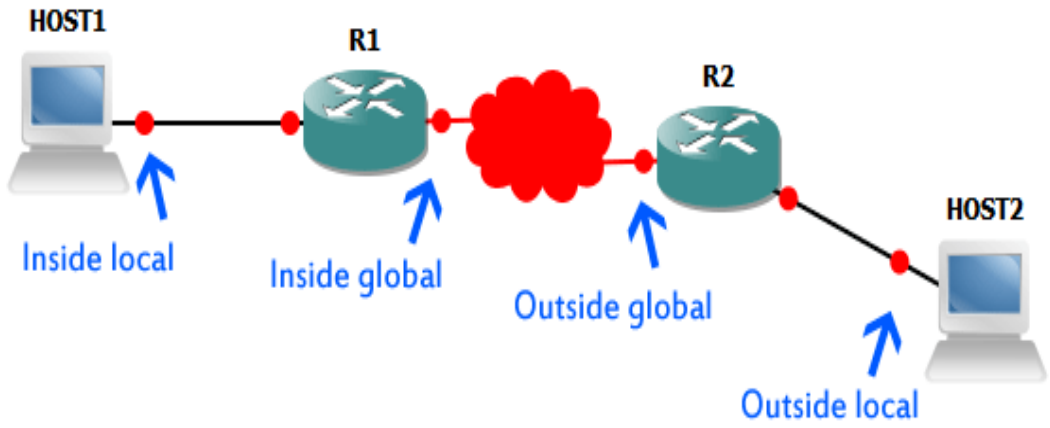
Introducing NAT and PAT

- NAT is designed to conserve IP addresses and enable networks to use private IP addresses on internal networks.
- These private, internal addresses are translated to routable, public addresses.
- NAT, as defined by RFC 1631, is the process of swapping one address for another in the IP packet header.
- In practice, NAT is used to allow hosts that are privately addressed to access the Internet.
- NAT translations can occur dynamically or statically.
- The most powerful feature of NAT routers is their capability to use port address translation (PAT), which allows multiple inside addresses to map to the same global address.
- This is sometimes called a many-to-one NAT.

Network Address Translation (NAT)

NAT Inside and Outside Addresses

- Inside Local – the specific IP address assigned to an inside host behind a NAT-enabled device (usually a private address).
- Inside Global –the address that identifies an inside host to the outside world (usually a public address). Essentially, this is the dynamically or statically-assigned public address assigned to a private host.
- Outside Global – the address assigned to an outside host (usually a public address).
- Outside Local – the address that identifies an outside host to the inside network. Often, this is the same address as the Outside Global. However, it is occasionally necessary to translate an outside (usually public) address to an inside (usually private) address.



NAT Inside and Outside Addresses

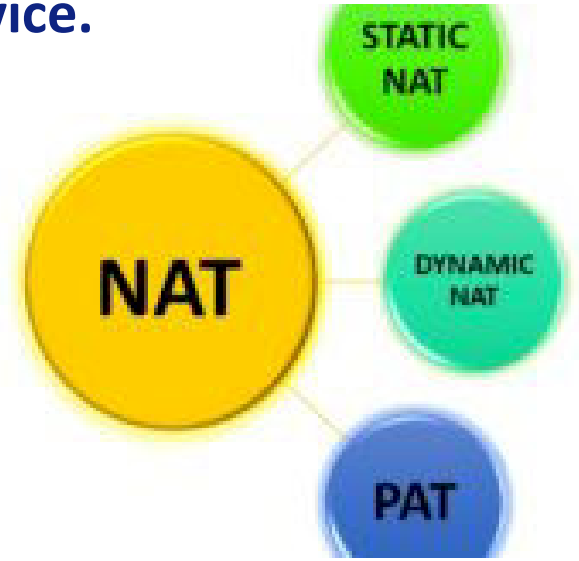
Types of NAT

NAT can be implemented using one of three methods:

1. Static NAT – performs a static one-to-one translation between two addresses, or between a port on one address to a port on another address. Static NAT is most often used to assign a public address to a device behind a NAT-enabled router.

2. Dynamic NAT – utilizes a pool of global addresses to dynamically translate the outbound traffic of clients behind a NAT-enabled device.

3. NAT Overload or Port Address Translation (PAT) – translates the outbound traffic of clients to unique port numbers of a single global address. PAT is necessary when the number of internal clients exceeds the available global addresses.

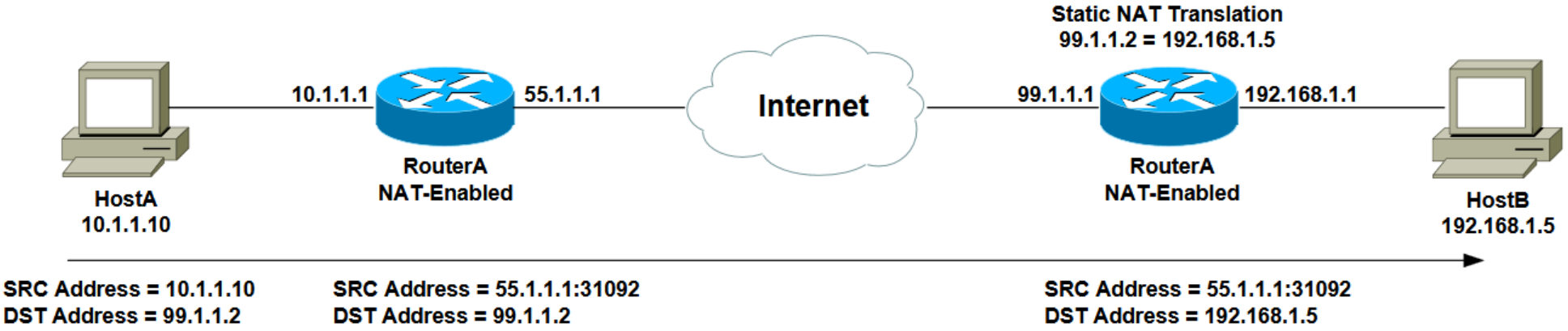


Network Address Translation (NAT)

NAT Operation Example

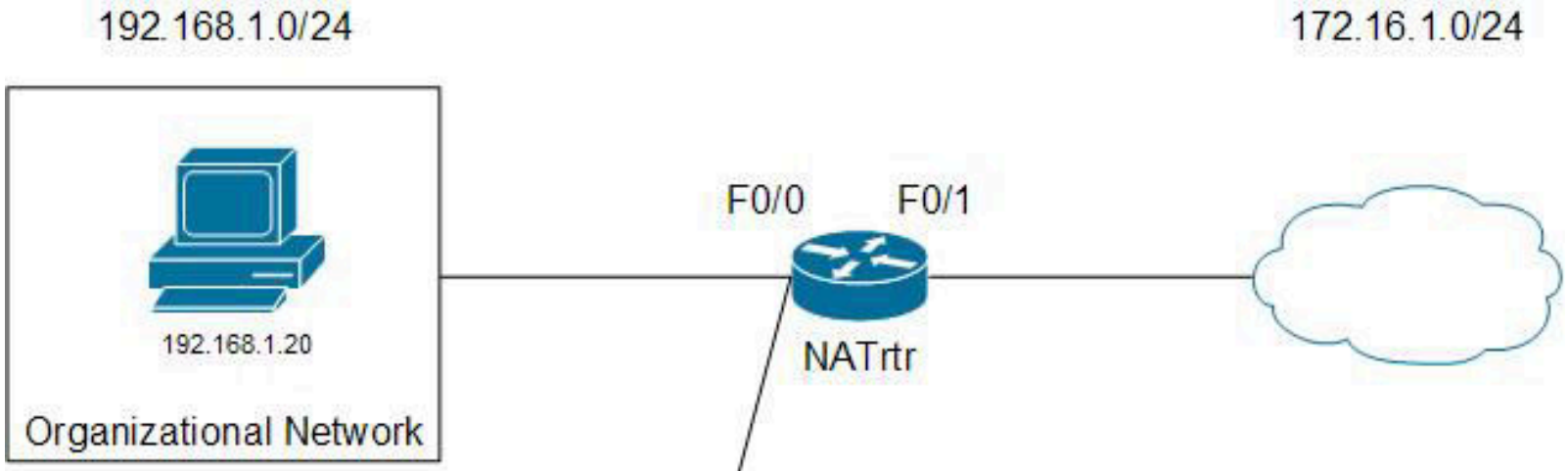
Inside Local Address - 10.1.1.10 **Inside Global Address- 55.1.1.1**
Outside Global Address– 99.1.1.2 **Outside Local Address – 99.1.1.2**

- **The below example demonstrates how the source (SRC) and destination (DST) IP addresses within the Network-Layer header are translated by NAT.**



Network Address Translation (NAT)

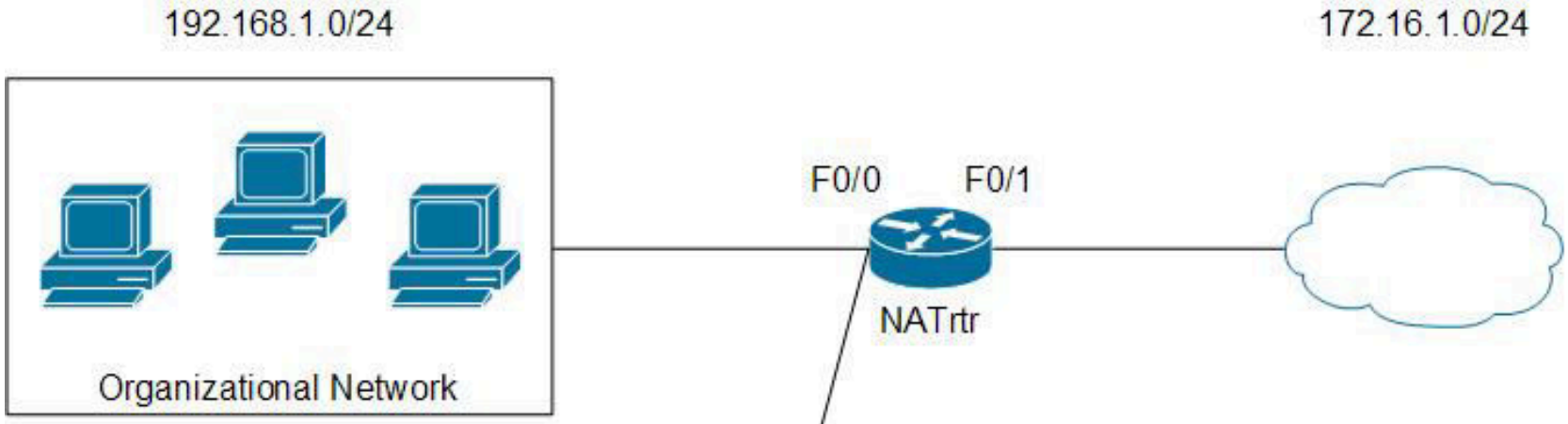
Configuring Static NAT




```
NATrtr#configure terminal
NATrtr(config)#ip nat inside source static 192.168.1.20 172.16.1.5
NATrtr(config)#interface f0/0
NATrtr(config-if)#ip address 192.168.1.1 255.255.255.0
NATrtr(config-if)#ip nat inside
NATrtr(config-if)#interface f0/1
NATrtr(config-if)#ip address 172.16.1.1 255.255.255.0
NATrtr(config-if)#ip nat outside
NATrtr(config-if)#end
```

Network Address Translation (NAT)

Configuring NAT Overload (or PAT)




```
NATrtr#configure terminal
NATrtr(config)#ip nat pool example-pool 172.16.1.10 172.16.1.20 netmask 255.255.255.0
NATrtr(config)#access-list 55 permit 192.168.1.0 0.0.0.255
NATrtr(config)#ip nat inside source list 55 pool example-pool
NATrtr(config)#interface f0/0
NATrtr(config-if)#ip address 192.168.1.1 255.255.255.0
NATrtr(config-if)#ip nat inside
NATrtr(config-if)#interface f0/1
NATrtr(config-if)#ip address 172.16.1.1 255.255.255.0
NATrtr(config-if)#ip nat outside
NATrtr(config-if)#end
```

Network Address Translation (NAT)

Troubleshooting NAT

- To view all current static and dynamic translations:

  Router# show ip nat translations

- To view whether an interface is configure as an inside or outside NAT interface, and to display statistical information regarding active NAT translations:

  Router# show ip nat statistics

- To view NAT translations in real-time:

  Router# debug ip nat

- To clear all dynamic NAT entries from the translation table:

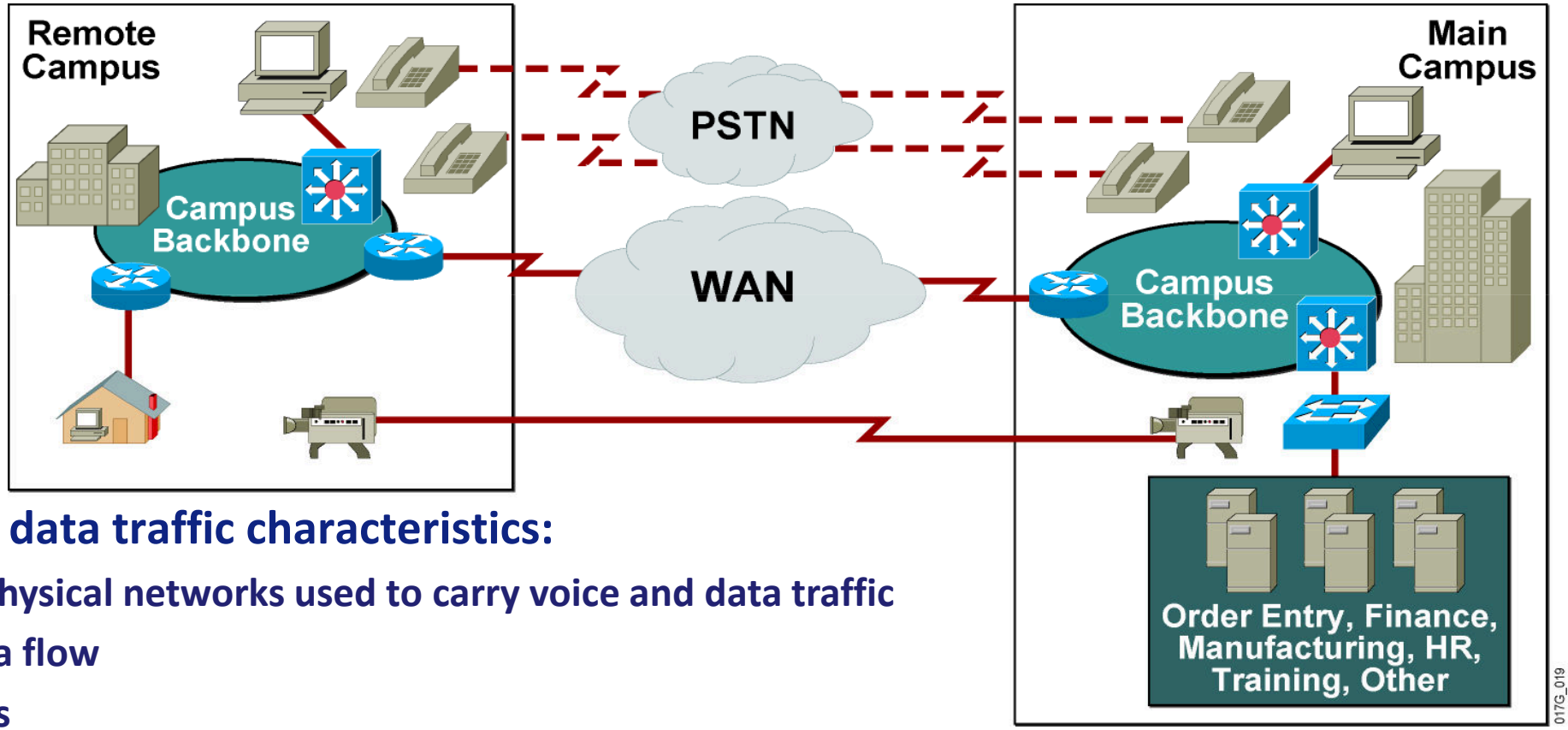
  Router# clear ip nat translation

Quality of Service in IP Networks

Quality of Service in IP Networks

Why QoS is Essential?

- Traditional Nonconverged Network



Traditional data traffic characteristics:

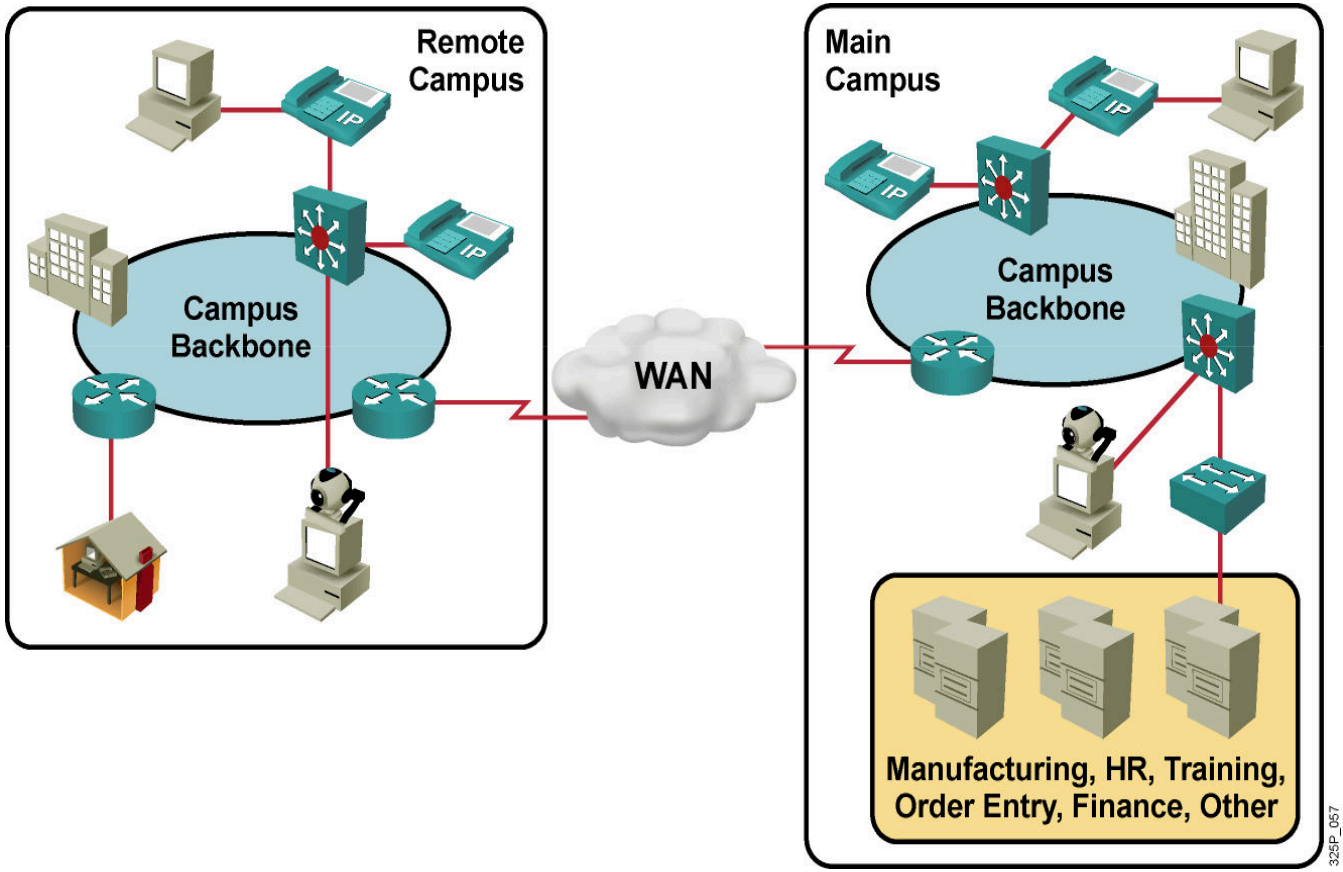
- Separate physical networks used to carry voice and data traffic
- Bursty data flow
- FIFO access
- Not overly time-sensitive; delays OK
- Brief outages are survivable

Quality of Service in IP Networks

Why QoS is Essential?

Converged Network Realities:

- Constant small-packet voice flow competes with bursty data flow.
- Critical traffic must have priority.
- Voice and video are time-sensitive.
- Brief outages are not acceptable.



Quality of Service in IP Networks

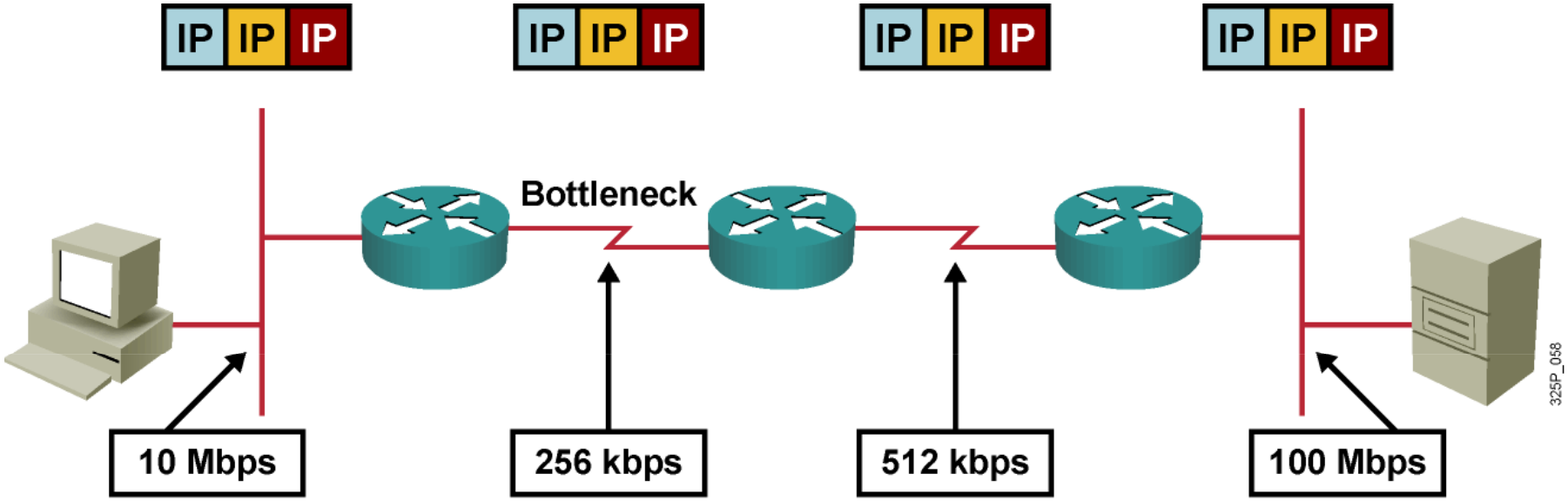
The Network QoS Parameters

There are four main QoS metrics in networking:

1) Loss: This is the failure of a packet, transmitted into the network at its source, to reach its intended destination.
2) Latency: This is the delay that takes place between the transmission of a packet into the network at its source and its arrival at its intended destination.
3) Jitter: This is the variation in latency between consecutive packets in a single flow.
4) Bandwidth: the amount of data that can be carried from one point to another in a given time period (usually a second). Network bandwidth is usually expressed in bits per second (bps).

Quality of Service in IP Networks

Measuring Available Bandwidth

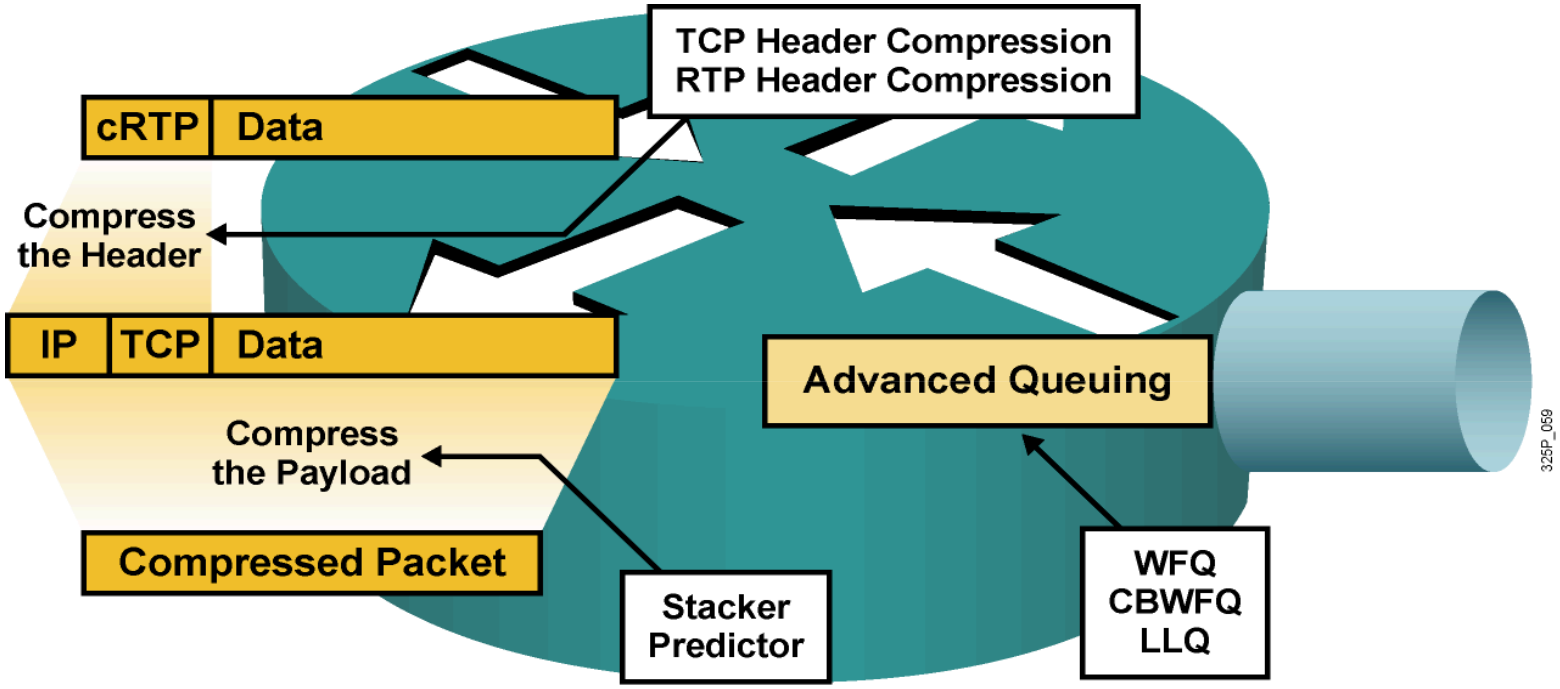


$Bandwidth_{max}$ = min (10 Mbps, 256 kbps, 512 kbps, 100 Mbps) = 256 kbps
$Bandwidth_{avail}$ = $Bandwidth_{max}$ / flows

- The maximum available bandwidth is the bandwidth of the slowest link.
- Multiple flows are competing for the same bandwidth, resulting in much less bandwidth being available to one single application.
- A lack in bandwidth can have performance impacts on network applications.

Quality of Service in IP Networks

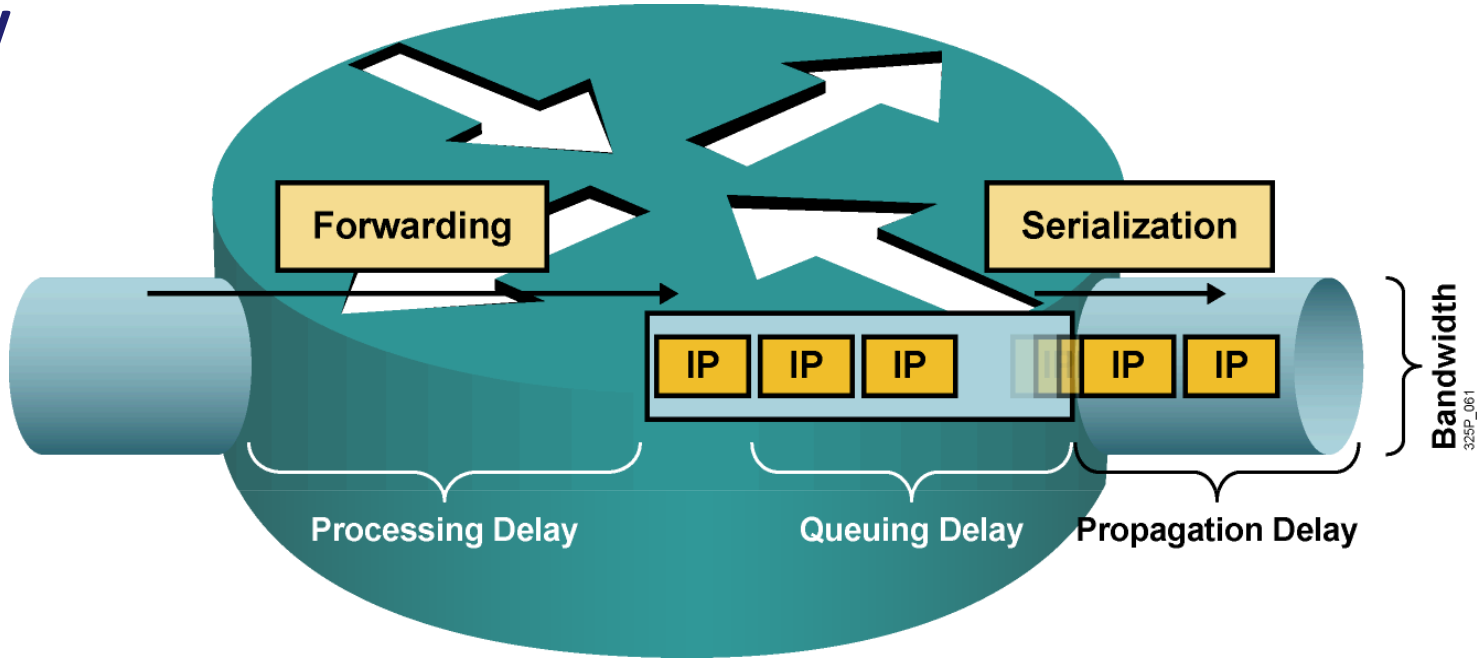
Increasing Available Bandwidth



- Upgrade the link (the best but also the most expensive solution).
- Improve QoS with queuing mechanisms to forward the important packets first.
- Compress the payload of Layer 2 frames (takes time).
- Compress IP packet headers.

Quality of Service in IP Networks

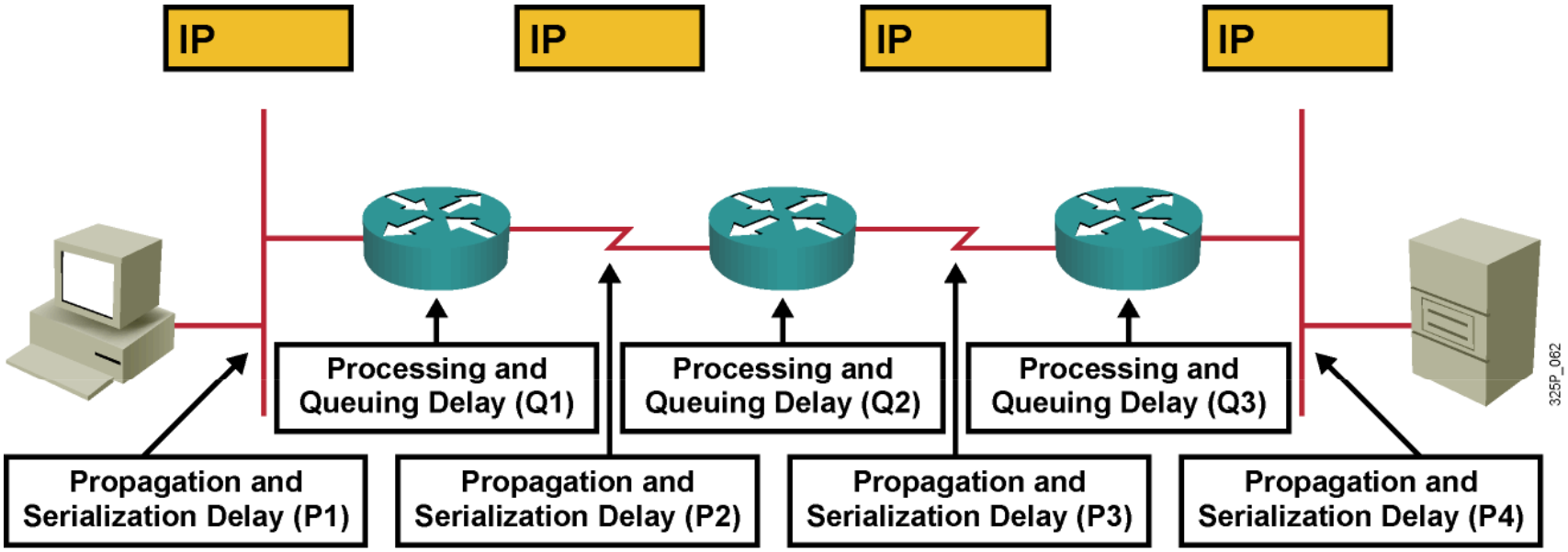
Types of Delay



- Processing delay: The time it takes for a router to take the packet from an input interface, examine the packet, and put the packet into the output queue of the output interface.
- Queuing delay: The time a packet resides in the output queue of a router.
- Serialization delay: The time it takes to place the “bits on the wire.”
- Propagation delay: The time it takes for the packet to cross the link from one end to the other.

Quality of Service in IP Networks

The Impact of Delay and Jitter on Quality



Delay = P1 + Q1 + P2 + Q2 + P3 + Q3 + P4 = *x* ms

- End-to-end delay: The sum of all propagation, processing, serialization, and queuing delays in the path
- Jitter: The variation in the delay.
- In best-effort networks, propagation and serialization delays are fixed, while processing and queuing delays are unpredictable.

Quality of Service in IP Networks

The Impacts of Packet Loss:

- Telephone call: “I cannot understand you. Your voice is breaking up.”
- Teleconferencing: “The picture is very jerky. Voice is not synchronized.”
- Publishing company: “This file is corrupted.”
- Call center: “Please hold while my screen refreshes.”

Types of Packet Drops:

- Tail drops occur when the output queue is full. Tail drops are common and happen when a link is congested.
- Other types of drops, usually resulting from router congestion, include input drop, ignore, overrun, and frame errors. These errors can often be solved with hardware upgrades.

Quality of Service in IP Networks

Example: QoS Requirements for Voice

- Voice calls, either one-to-one or on a conference connection capability, require the following:
  - ≤ 150 ms of one-way latency from mouth to ear (per the ITU G.114 standard)
  - ≤ 30 ms jitter
  - ≤ 1 percent packet loss
  - 17 to 106 kbps of guaranteed priority bandwidth per call (depending on the sampling rate, codec, and Layer 2 overhead)
  - 150 bps (plus Layer 2 overhead) per phone of guaranteed bandwidth for voice control traffic

Quality of Service in IP Networks

QoS Methodologies

There are three key methodologies for implementing QoS:

- **Best-Effort (BE) QoS is essentially no QoS. Traffic is routed on a first-come, first-served basis. Sensitive traffic is treated nodifferently than normal traffic.**

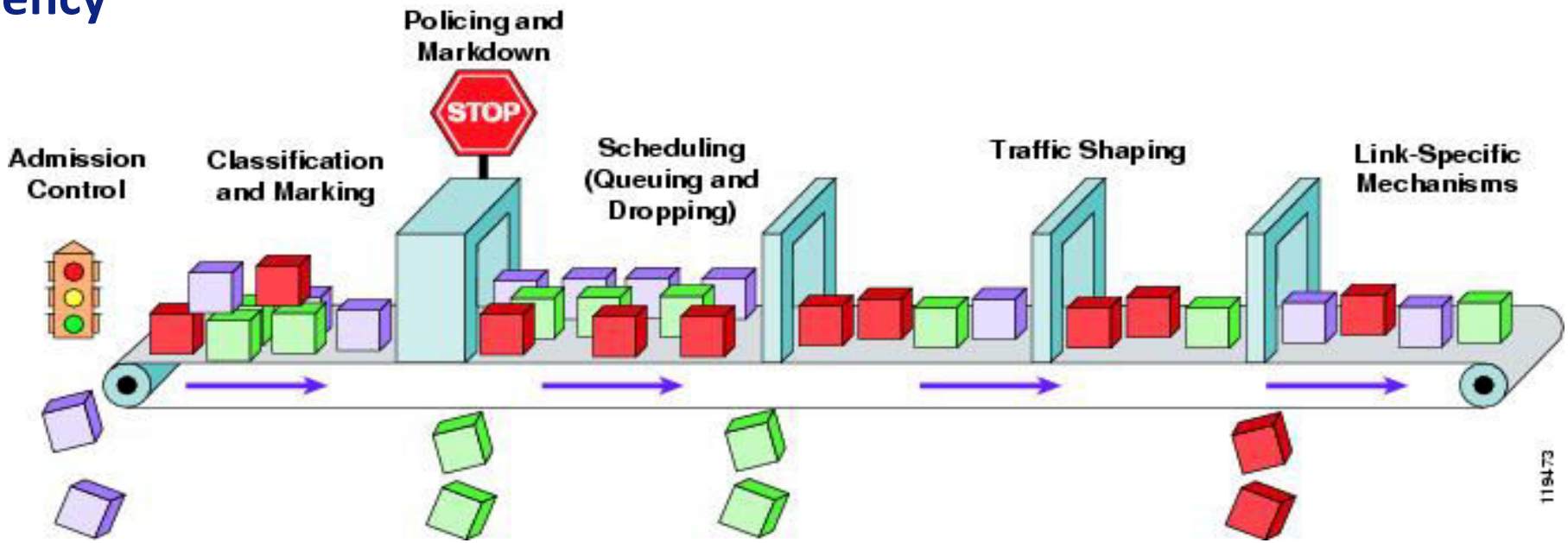
- **Integrated Services (IntServ) QoS is also known as end-to-end QoS. With the IntServ model, applications ask to the network for an explicit resource reservation per flow.**

- **Differentiated Services (DiffServ) QoS was designed to be a scalable QoS solution. Traffic types are organized into specific classes, and then marked to identify their classification. Policies are then created on a per-hop basis (PHB) to provide a specific level of service.**

Quality of Service in IP Networks

QoS Tools:

- Classification and Marking
- Queuing
- Queue Congestion Avoidance
- Traffic shaping and traffic policing
- Link efficiency



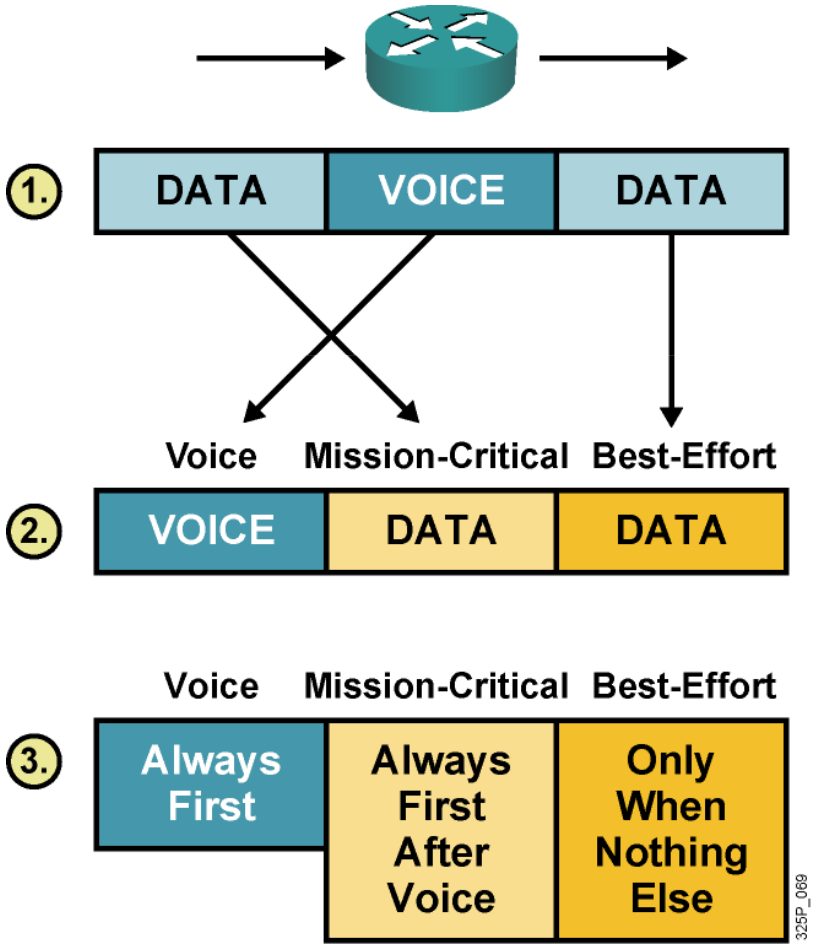
Classifying and Marking Traffic

DiffServ QoS involves three steps:

Step 1: Identify types of traffic and their requirements. Once classification has occurred, traffic should be marked.

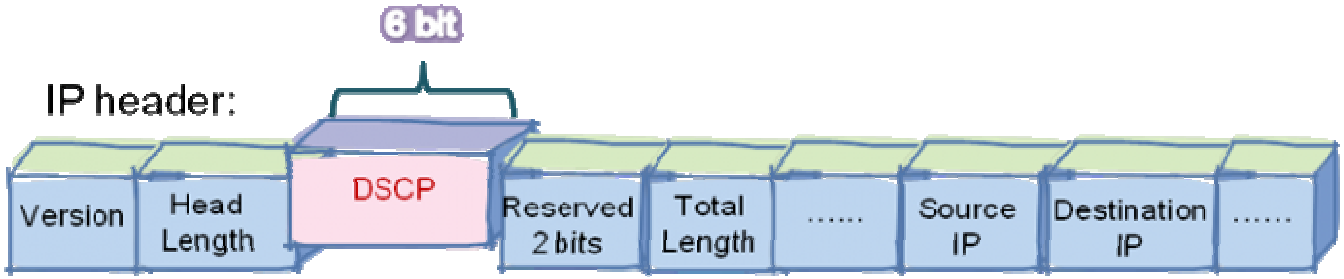
Step 2: Divide traffic into classes or classes of service.

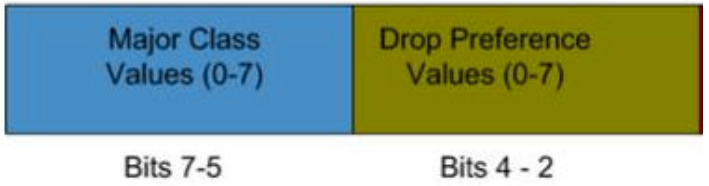
Step 3: Define QoS policies for each class.



Quality of Service in IP Networks

- **Layer-3 Marking**
- **It is accomplished using the 8-bit Type of Service (ToS) field, part of the IP Header.**
- **In Diffserv, the definition of this entire field was changed. It is now called the "DS“ (Differentiated Services) field and the upper 6 bits contain a value called the "DSCP“ (Differentiated Services Code Point).**





- **The first three bits identify the Class Selector of the packet. The following three bits identify the Drop Precedence of the packet.**
- **These values determine the per-hop behavior (PHB) received by each classification of traffic.**

Quality of Service in IP Networks

Classes of Services

The IETF recommends the following commonly defined classes of service:

- **Best Effort (BE) – This is the default class of service.**
- **Expedited Forwarding (EF) – This class of service is dedicated to low-delay, low-packet-loss, and low-jitter and is typically used for voice and video.**
- **Assured Forwarding (AF) – The AF class of service assures packet delivery under defined conditions. Packets are transmitted as long as the traffic does not exceed a subscribed rate.**

Application	L3 Classification		
	IPP	PHB	DSCP
Routing	6	CS6	48
Voice	5	EF	46
Video Conferencing	4	AF41	34
Streaming Video	4	CS4	32
Mission-Critical Data	3	AF31	26
Call Signaling	3	CS3	24
Transactional Data	2	AF21	18
Network Management	2	CS2	16
Bulk Data	1	AF11	10
Best Effort	0	0	0
Scavenger	1	CS1	8

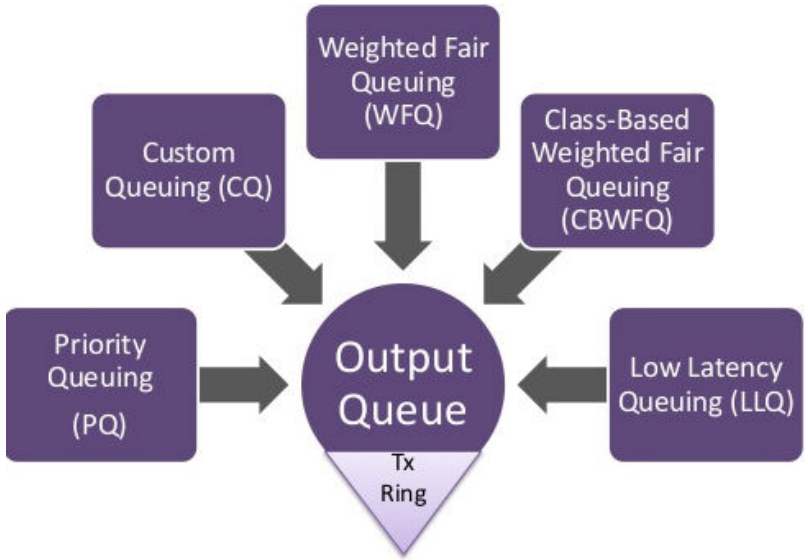
Quality of Service in IP Networks

Queuing

- A queue is used to store traffic until it can be processed or serialized. Both switch and router interfaces have ingress (inbound) queues and egress (outbound) queues. If the queue becomes saturated, new packets will be dropped (tail drop).

- In order to provide a preferred level of service for high-priority traffic, some form of software queuing must be used. Software queuing techniques can include:

  - First-In First-Out (FIFO) (default)
  - Priority Queuing (PQ)
  - Custom Queuing (CQ)
  - Weighted Fair Queuing (WFQ)
  - Class-Based Weighted Fair Queuing (CBWFQ)
  - Low-Latency Queuing (LLQ)



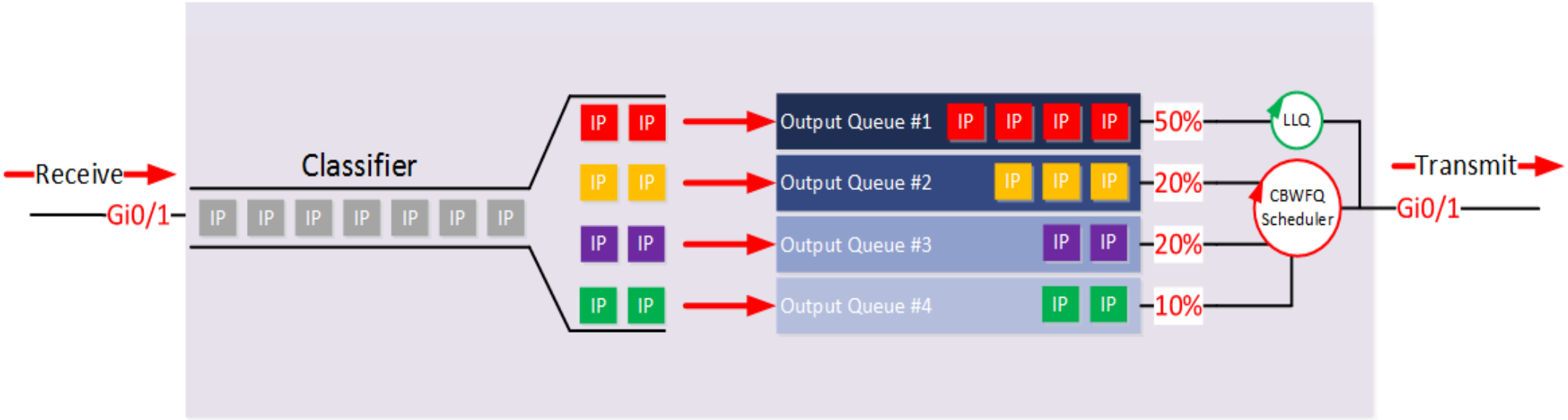
CBWFQ and LLQ Queuing

- The CBWFQ scheduler guarantees a minimum percentage of a link's bandwidth to each class/queue. If all queues have a large number packets, each queue gets the percentage bandwidth implied by the configuration. However, if some queues are empty and do not need their bandwidth for a short period, the bandwidth is proportionally allocated across the other classes. CBWFQ is used for data applications.

- LLQ looks and acts just like CBWFQ in most regards, except it adds the capability for some queues to be configured as low-latency queues. LLQ schedules these specific queues as strict-priority queues. In other words, LLQ always services packets in these priority queues first. LLQ is used for the highest-priority traffic, which is especially suited for voice over IP (VoIP).

CBWFQ and LLQ Queuing

- LLQ (Low Latency Queuing) is an extension of CBWFQ (Class Based Weighted Fair Queuing) where we add a priority queue to the scheduler. Here's an illustration to help you visualize this:

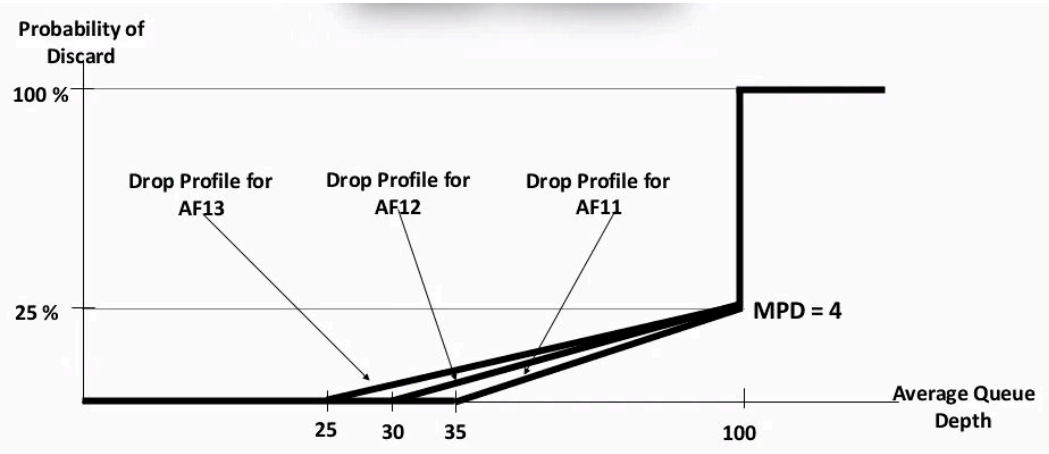


Quality of Service in IP Networks

Queue Congestion Avoidance

- Router queues are susceptible to congestion. Common causes for congestion include:
  - The speed of an ingress interface is higher than the egress interface.
  - The combined traffic of multiple ingress interfaces exceeds the capacity of a single egress interface.
  - The switch/router CPU is insufficient to handle the size of the forwarding table.
- If an interface's queue buffer fills to capacity, new packets will be dropped. This condition is referred to as tail drop.

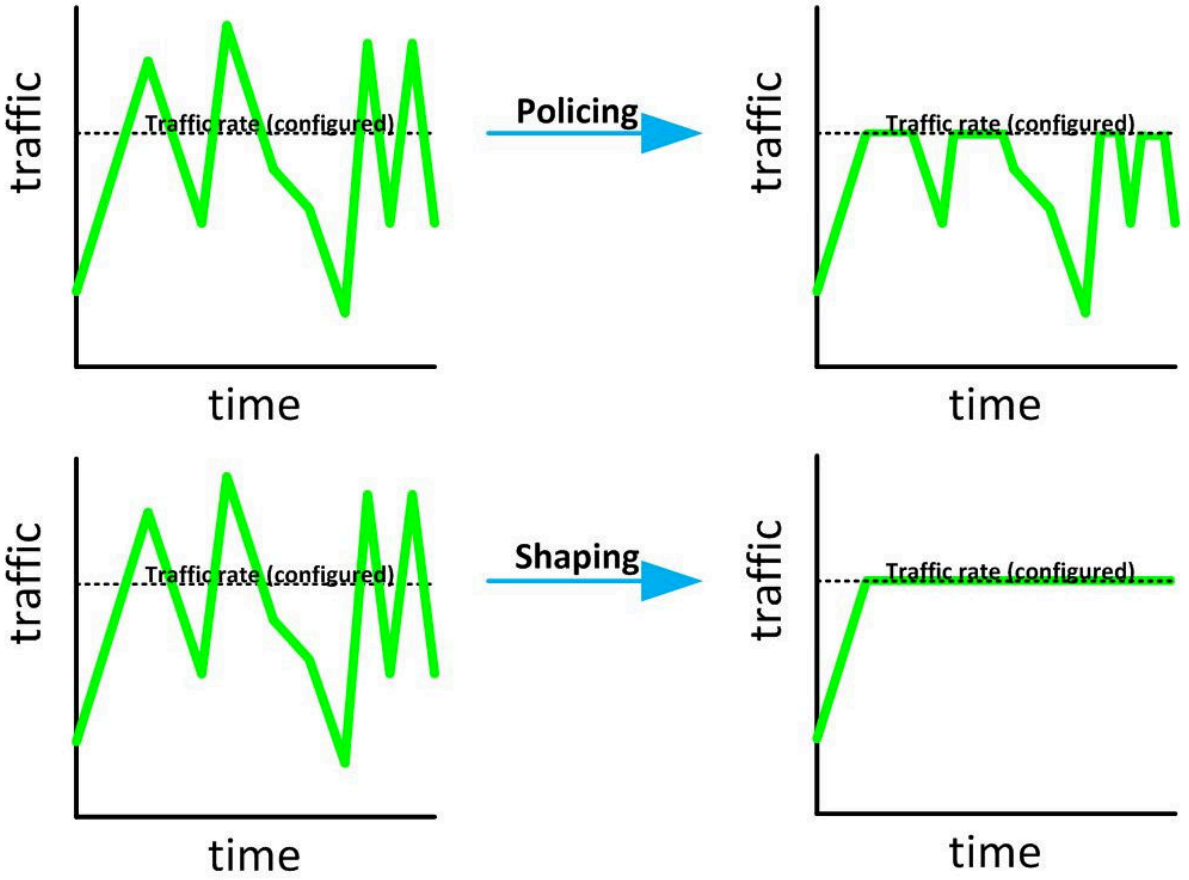
- QoS also provides a mechanism to drop lower priority traffic before higher priority traffic, during periods of congestion. This is known as Weighted Random Early Detection (WRED).



Traffic Shaping and Policing

Policing - traffic exceeding the configured threshold is dropped (or remarked to a lower class of service). It is usually used in the inbound traffic.

Shaping - buffers excess (burst) traffic to transmit during non-burst periods. It is used in the outbound traffic.



Quality of Service in IP Networks

Link Efficiency

- Compression increases the bandwidth available on a link, by reducing the size of data. Compression algorithms shrink data by exploiting blank spaces and repeating patterns within headers and payloads.

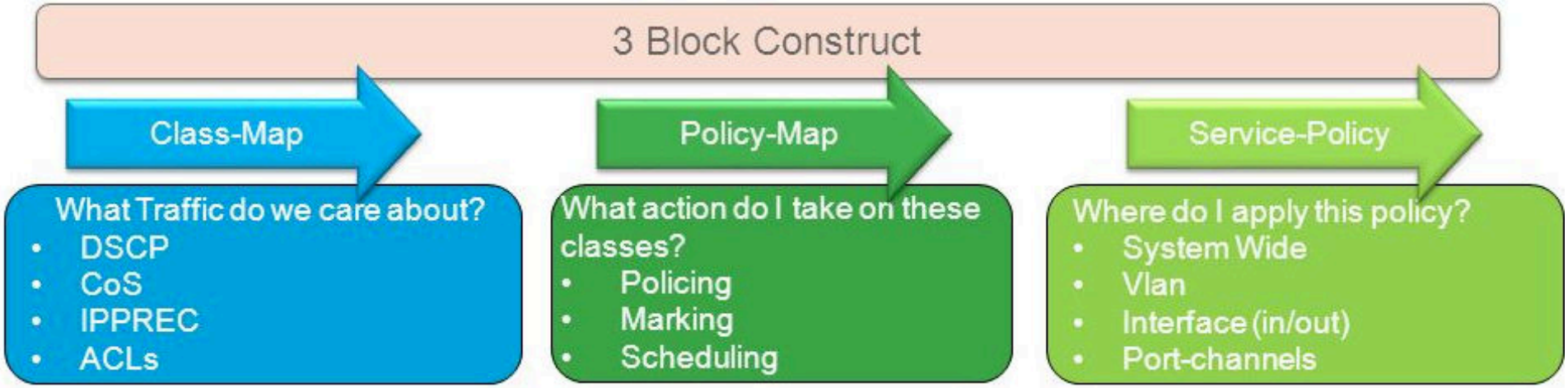
- Header compression: the Layer-3 and Layer-4 headers of a packet will be compressed. This is most efficient for traffic that has a small payload, such as VoIP.

The two most common types of header compression are as follows:

- TCP Header Compression
- RTP Header Compression (cRTP) - compresses the RTP, UDP, and IP headers of a packet. cRTP can reduce the collective header size from 40 bytes to 2 bytes.

Cisco Modular QoS CLI (MQC)

- The MQC is an improved command-line implementation of QoS in Cisco Devices
- There are three steps to configuring QoS using MQC:
  - Classify traffic using a class-map.
  - Define a QoS policy using a policy-map.
  - Apply the policy to an interface, using the service-policy command.



MQC - Classify traffic using a class-map

- Each class is identified using a class map. A traffic class contains three major elements:

1. A case-sensitive name
2. A series of match commands
3. An instruction on how to evaluate the match commands if more than one match command exists in the traffic class
    - Match all: All conditions have to succeed.
    - Match any: At least one condition must succeed.

```
access-list 11 permit 10.0.76.0 0.0.0.255
access-list 12 permit 10.0.78.0 0.0.0.255
access-list 13 permit 10.0.77.0 0.0.0.255
access-list 118 permit udp any eq 57 any eq 57
access-list 126 permit tcp any any range 1963 1978
access-list 127 permit tcp any range 1963 1978 any
access-list 146 permit udp any range 16384 19000 any
```

REDDIG CONFIG

```
class-map match-any EF
 match protocol sip
 match access-group 146
class-map match-all AF4
 match access-group 13
class-map match-any AF3
 match access-group 11
 match access-group 12
 match access-group 126
 match access-group 127
 match protocol gre
class-map match-any AF2
 match access-group 118
class-map match-any AF1
```

MQC - Define a QoS policy using a policy-map

A policy map defines a traffic policy, which configures the QoS features associated with a traffic class that was previously identified using a class map.

A traffic policy contains three major elements:

1. A case-sensitive name
2. A traffic class
3. The QoS policy that is associated with that traffic class

A wide variety of policy actions are available.

Example:

```
policy-map QoS-Policy
  class VoIP
    priority 100
  class Application
    bandwidth 25
  class class-default
    fair-queue
```

REDDIG CONFIG

```
policy-map SETDSCP
 class EF
  set ip dscp ef
 class AF1
  set ip dscp af11
 class AF2
  set ip dscp af21
 class AF3
  set ip dscp af31
 class AF4
  set ip dscp af41
```

Quality of Service in IP Networks

MQC - Apply the policy to an interface

- Attach the specified service policy map to the input or output interface.
- Service policies can be applied to an interface for inbound or outbound packets.

router(config-if)# `service-policy {input | output} policy-map-name`

REDDIG CONFIG

```
interface GigabitEthernet0/0
 ip address 10.100.30.101 255.255.255.0
<output omitted>
service-policy output SETDSCP
```

```
interface GigabitEthernet0/1.100
 encapsulation dot1Q 100
 ip address 10.0.80.253 255.255.255.0
<output omitted>
 service-policy input SETDSCP
```

Quality of Service in IP Networks

- QOS - Basic Verification Commands
  - Display the class maps

```
router#
show class-map
```

  - Display the policy maps

```
router#
show policy-map
```

  - Display the applied policy map on the interface

```
router#
show policy-map interface type number
```

Traffic Tunnelling

Traffic Tunnelling

Tunneling Overview

- Tunnels provide a way to transport protocols that the underlying network does not support. There are several reasons why this may be:
  - The network infrastructure doesn't support the protocol being used
  - The network infrastructure cannot route the packets due to lack of routing information or addressing types (public addressing vs. private addressing)
  - The network infrastructure doesn't support the traffic type (multicast or broadcast)

- In REDDIG Network, we use two types of tunnels:

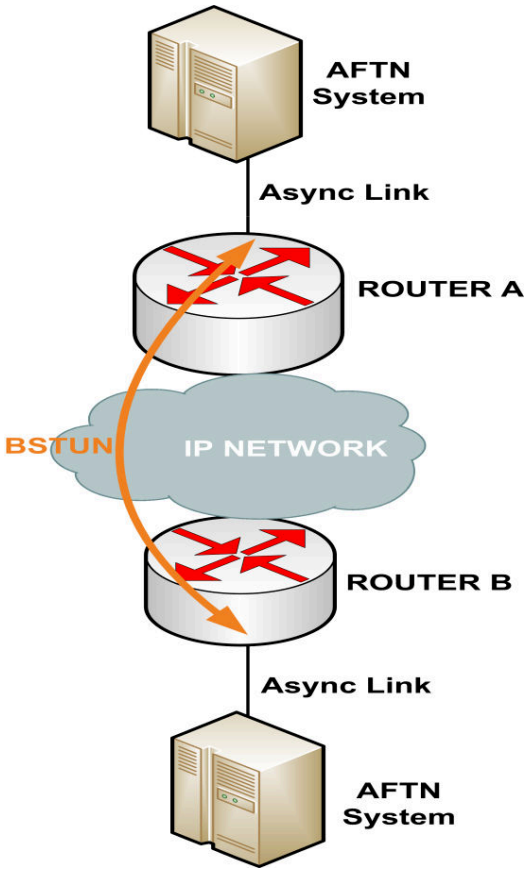
  - Block Serial Tunnel (BSTUN) for async protocols (AFTN)
  - Generic Routing Encapsulation (GRE) for multicast routing support (RADAR)

Traffic Tunnelling

Block Serial Tunnel (BSTUN)

- BSTUN enable enterprises to transport polled asynchronous traffic over the same network that supports their multiprotocol traffic, eliminating the need for separate facilities.

- BSTUN Operations: At the downstream router, traffic from the attached asynchronous device is encapsulated in IP. The asynchronous traffic can be routed across arbitrary media to the host site where the upstream router supporting these protocols removes the IP encapsulation headers and presents the original traffic to the remote device over a serial connection



Traffic Tunnelling

BSTUN in REDDIG Network – Configuration and Monitoring

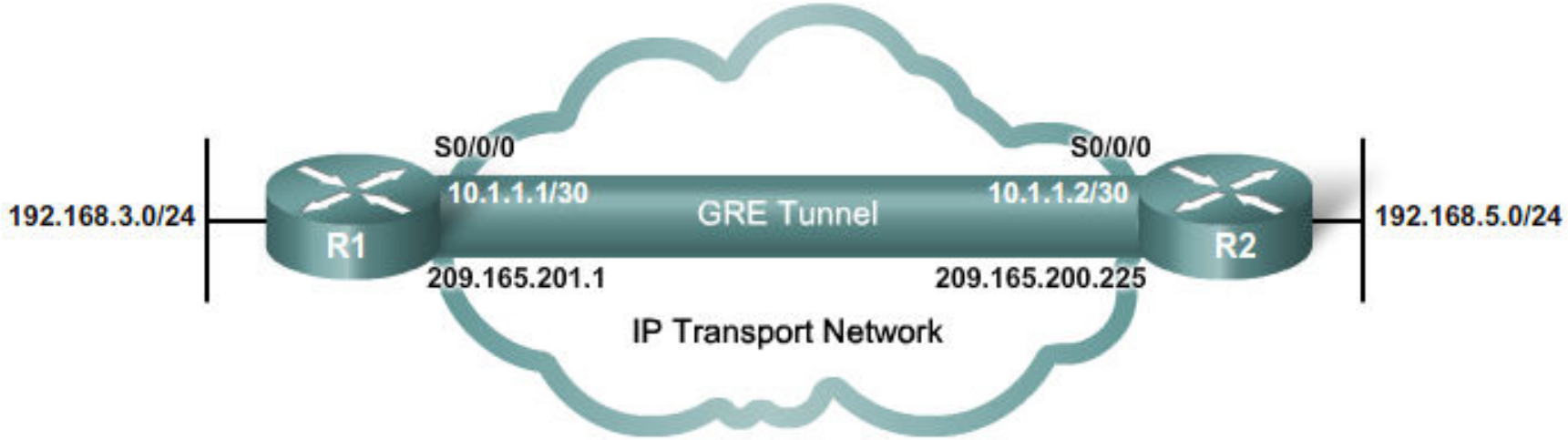
```
SBCT-CISCO-VSAT-1-A-V15#
interface Loopback0
 ip address 10.130.30.1 255.255.255.255
!
bstun protocol-group 1 async-generic
bstun protocol-group 2 async-generic
bstun protocol-group 3 async-generic
bstun protocol-group 10 async-generic
!
interface Serial0/0/2
 physical-layer async
 description AFTN_SAEZ_Ezeiza
 no ip address
 encapsulation bstun
 delay 600000
 bstun group 10
 bstun route all tcp 10.130.20.1 tcp-queue-max 500
 asp role primary
 asp addr-offset 0
 hold-queue 4000 in
 hold-queue 4000 out
```

```
SBCT-CISCO-VSAT-1-A-V15#sh bstun
This peer: 10.130.30.1

  Serial0/0/0 -- AFTN_SLLP_LaPaz (group 1 [async-generic])
route transport address       dlci lsap state     rx_pkts tx_pkts   drops
all       TCP       10.130.25.1              open      721450  162453      0


  Serial0/0/1 -- AFTN_SGAS_Asuncion (group 2 [async-generic])
route transport address       dlci lsap state     rx_pkts tx_pkts   drops
   all    TCP      10.130.55.1              open       52830  581112     14


  Serial0/0/2 -- AFTN_SAEZ_Ezeiza (group 10 [async-generic])
route transport address       dlci lsap state     rx_pkts tx_pkts   drops
  all       TCP      10.130.20.1              open      603587 1282627      7


  Serial0/0/3 -- AFTN_SUMU_Montevideo (group 3 [async-generic])
route transport address       dlci lsap state     rx_pkts tx_pkts   drops
  all       TCP      10.130.65.1              open      148110  448883      4
```

Traffic Tunnelling

Generic Routing Encapsulation (GRE)

- GRE is an OSI Layer 3 tunneling protocol:
  - Encapsulates a wide variety of protocol packet types inside IP tunnels
  - Creates a virtual point-to-point link to Cisco routers at remote points over an IP internetwork
  - Uses IP for transport
  - Uses an additional header to support any other OSI Layer 3 protocol as payload (for example, IP, IPX, AppleTalk)



Traffic Tunnelling

Why Use GRE Tunnels?

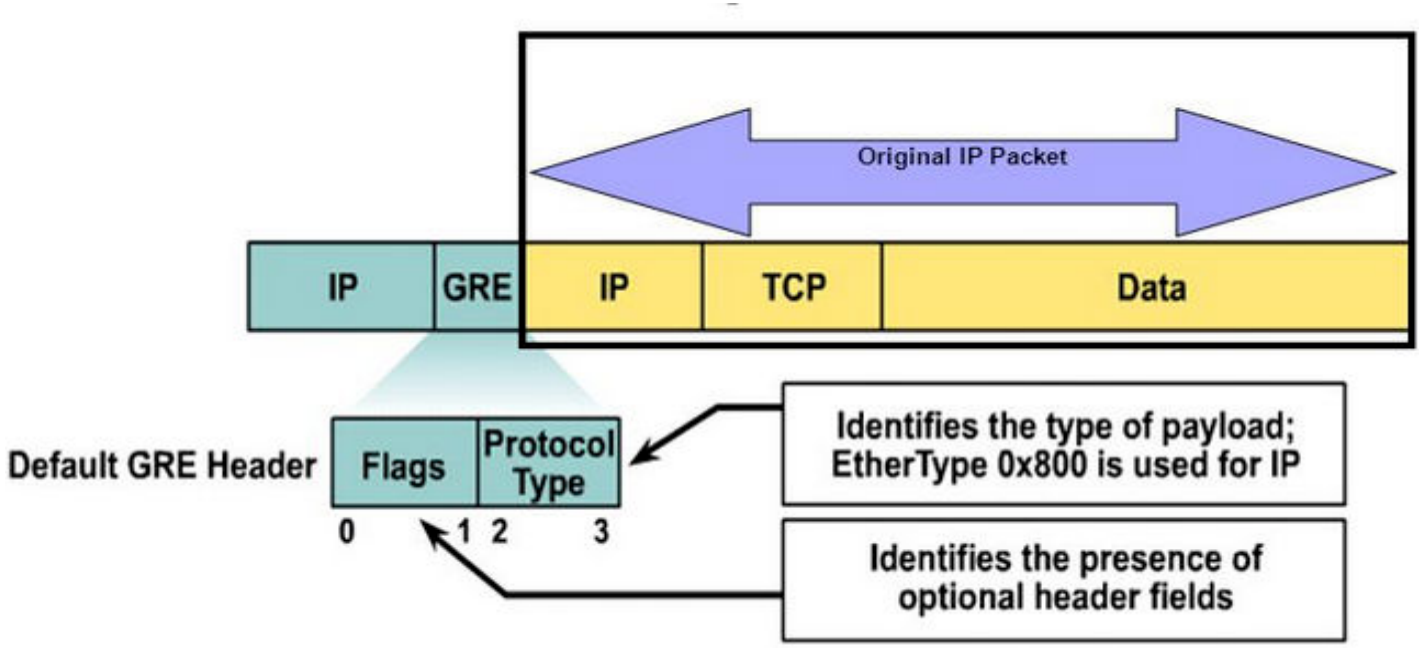
GRE can be used to solve these problems:

- **GRE supports multicast traffic allowing hello messages generated by an IGP to be transported through the GRE tunnel across the underlying infrastructure as a unicast packet. IPsec can then be used to encrypt all traffic flowing through the GRE tunnel.**
- **GRE configuration creates a logical direct connection between two sites over the underlying infrastructure. This means the control plane of the IGP believes it is directly connected to the neighbor with which it is exchanging hellos and therefore can form the adjacency.**

Traffic Tunnelling

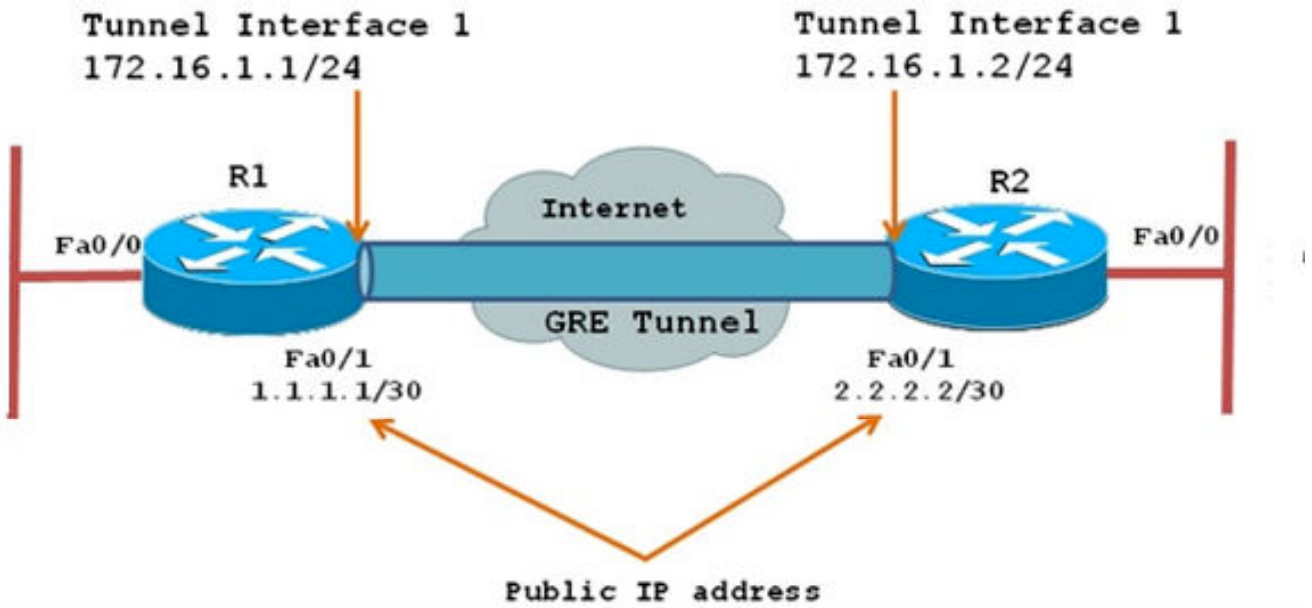
GRE - Encapsulation Overhead

- GRE itself is stateless. It does not include any flow-control mechanisms, by default.
- GRE does not include any strong security mechanisms to protect its payload.
- The GRE header, along with the tunneling IP header, creates at least 24 bytes of additional overhead for tunneled packets.



Traffic Tunnelling

GRE Configuration Example



ROUTER 1: R1	ROUTER 2: R2
R1(config)# interface Tunnel1	R2(config)# interface Tunnel1
R1(config-if)# ip address 172.16.1.1 255.255.255.0	R2(config-if)# ip 172.16.1.2 255.255.255.0
R1(config-if)# ip mtu 1400	R2(config-if)# ip mtu 1400
R1(config-if)# ip tcp adjust-mss 1360	R2(config-if)# ip tcp adjust-mss 1360
R1(config-if)# tunnel source 1.1.1.1	R2(config-if)# tunnel source 2.2.2.2
R1(config-if)# tunnel destination 2.2.2.2	R2(config-if)# tunnel destination 1.1.1.1

Traffic Tunnelling

GRE Tunnels in REDDIG Network – Configuration and Monitoring

```
interface GigabitEthernet0/1
 vrrp 120 ip 10.120.30.254

interface Tunnel1
 ip address 10.10.1.2 255.255.255.252
 keepalive 10 3
 tunnel source 10.100.30.254
 tunnel destination 10.100.25.254

interface Tunnel55
 ip address 10.30.55.1 255.255.255.252
 ip pim sparse-dense-mode
 shutdown
 tunnel source 10.120.30.254
 tunnel destination 10.120.55.254
 service-policy input SETDSCP
 service-policy output SETDSCP

ip route 10.130.25.1 255.255.255.255 Tunnel1
ip mroute 10.0.129.0 255.255.255.0 Tunnel55
```

```
BCT-CISCO-VSAT-1-A-V15#sh int tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 10.10.1.2/30
  MTU 17916 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive set (10 sec), retries 3
  Tunnel source 10.100.30.254, destination 10.100.25.254
  Tunnel protocol/transport GRE/IP
Tunnel TTL 255, Fast tunneling enabled
  Tunnel transport MTU 1476 bytes
Input queue: 0/75/0/0 (size/max/drops/flushes);
  Queueing strategy: fifo
    5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     9852581 packets input, 533892401 bytes, 0 no buffer
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     3924566 packets output, 234420597 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
```