Resumos RC 2023

#RC

Chapter 1 - Introduction

Network edge

What's the Internet: a service view

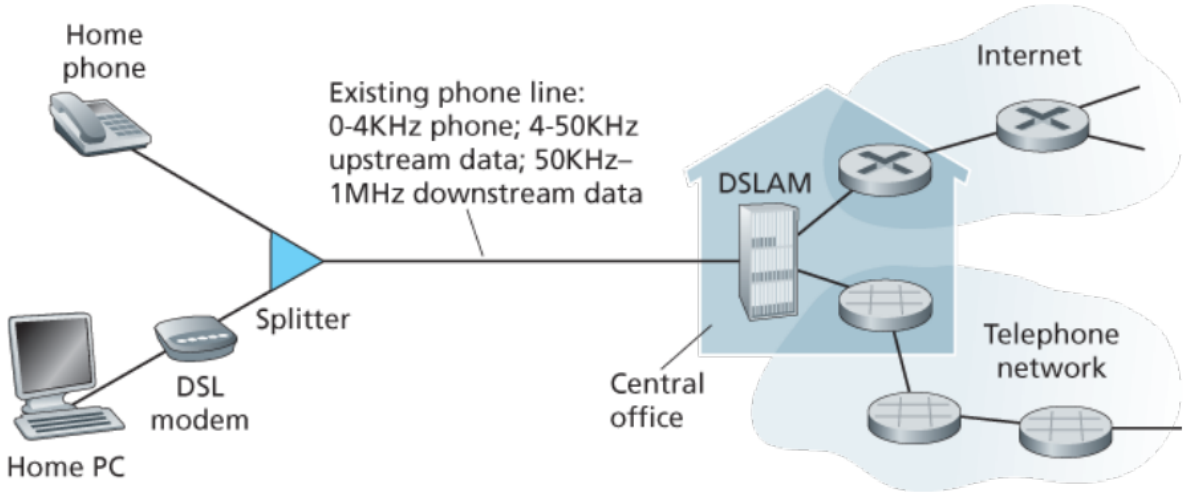
- infrastructure that provides services to applications
- provides programming interface to apps

Network structure:

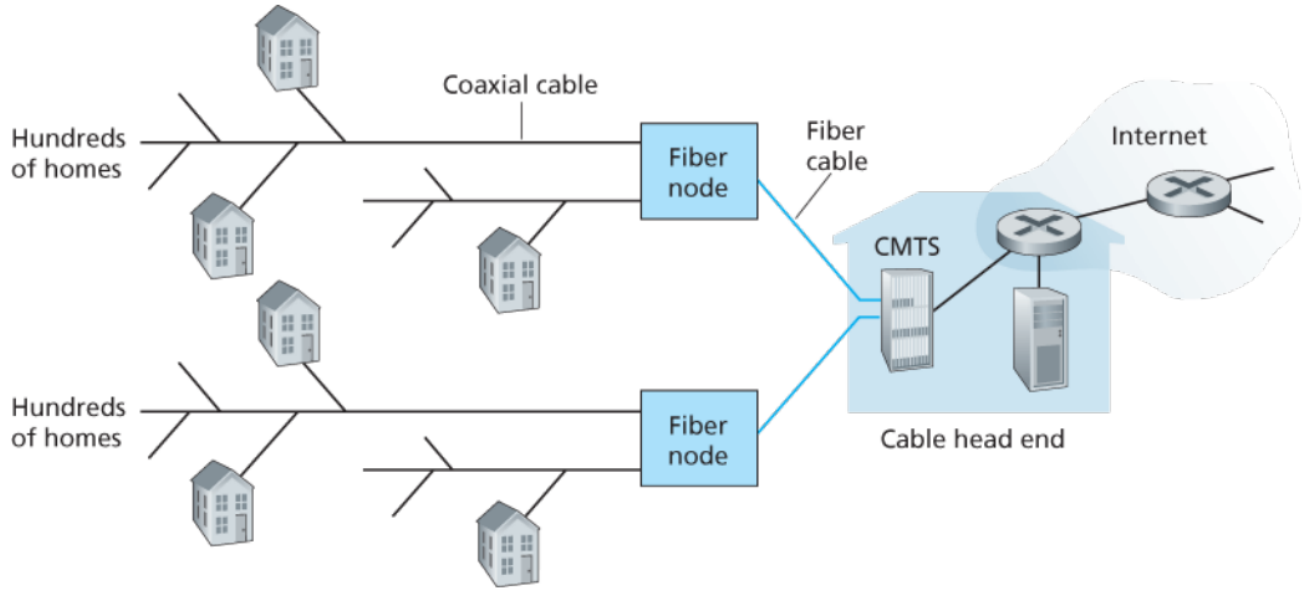
- Network edge: hosts - clients, servers
- Access networks, physical media (wired/ wireless communication links)
- Network core (interconnected routers)

A protocol defines the format and the order of messages exchanged between two or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message or other event.

DSL:



Cable Internet Access:

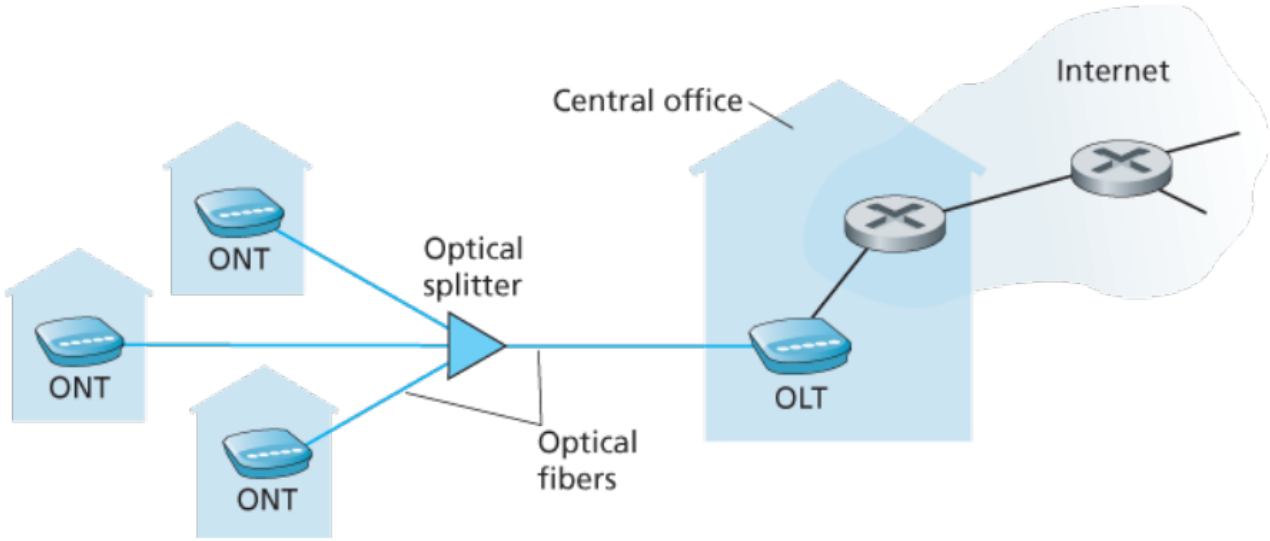


> This one is a hybrid fiber coax (HFC)

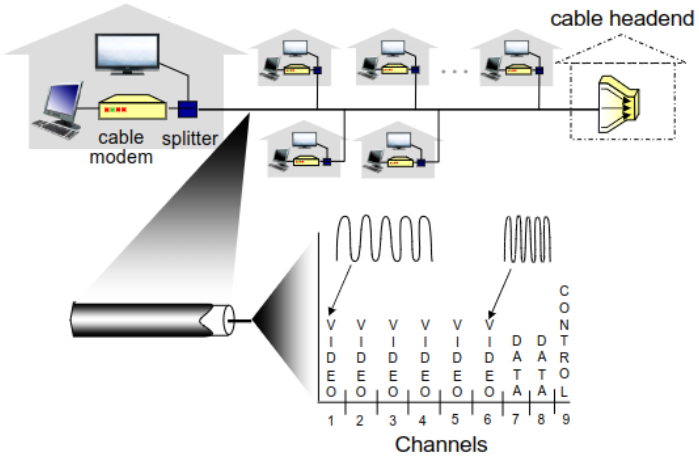
- twisted pair (TP)
- coaxial cable
- fiber optic cable

One important characteristic of cable Internet access is that it is a shared broadcast medium. In particular, every packet sent by the head end travels downstream on every link to every home and every packet sent by a home travels on the upstream channel to the head end. For this reason, if several users are simultaneously downloading a video file on the downstream channel, the actual rate at which each user receives its video file will be significantly lower than the aggregate cable downstream rate.
Because the upstream channel also shares, a distributed multiple access protocol is needed to coordinate transmissions and avoid collisions.

PON distribution architecture:



Acess network: cable network

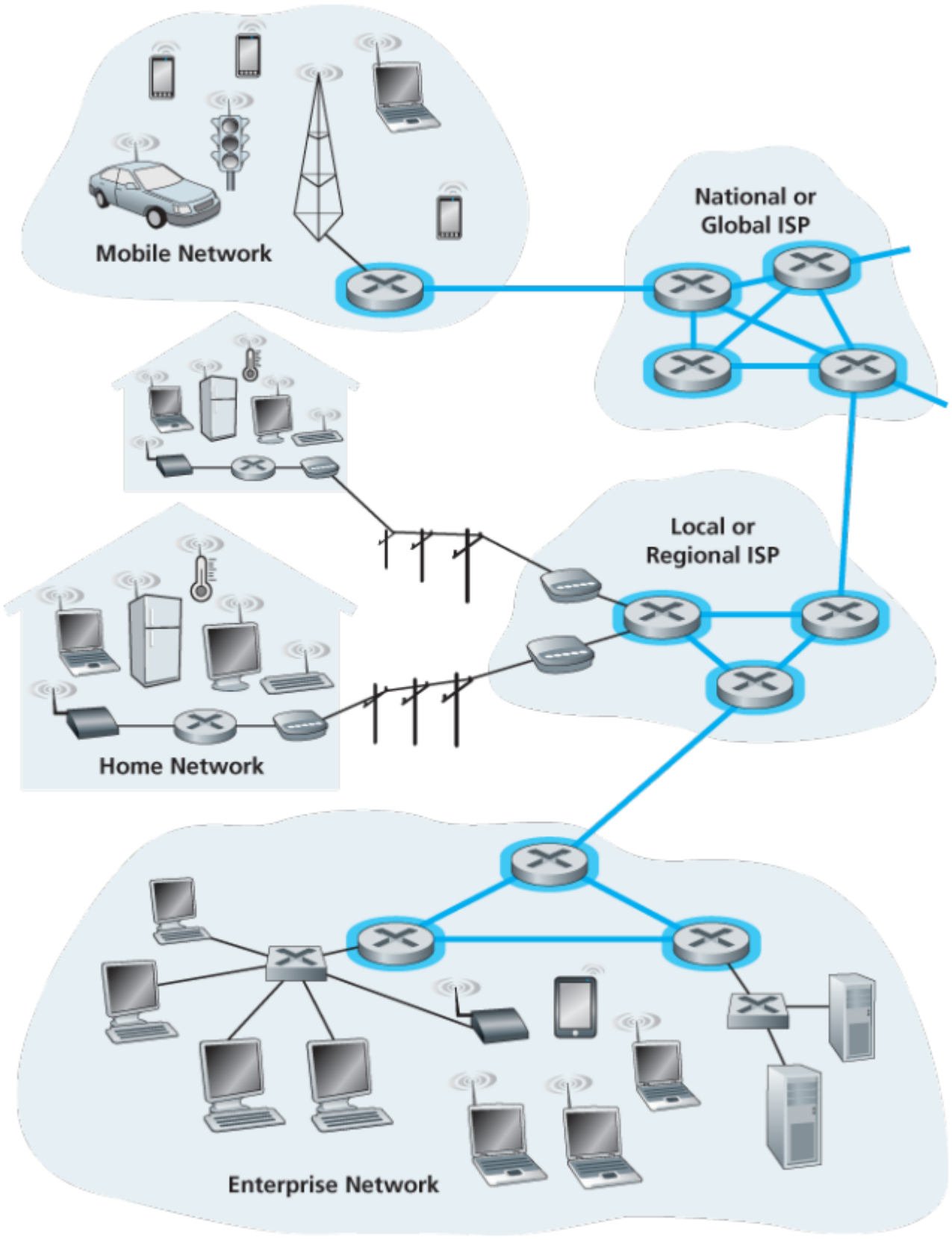


$$\text{packet transmission delay} = \text{time needed to transmit } L\text{-bit packet into link} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$$

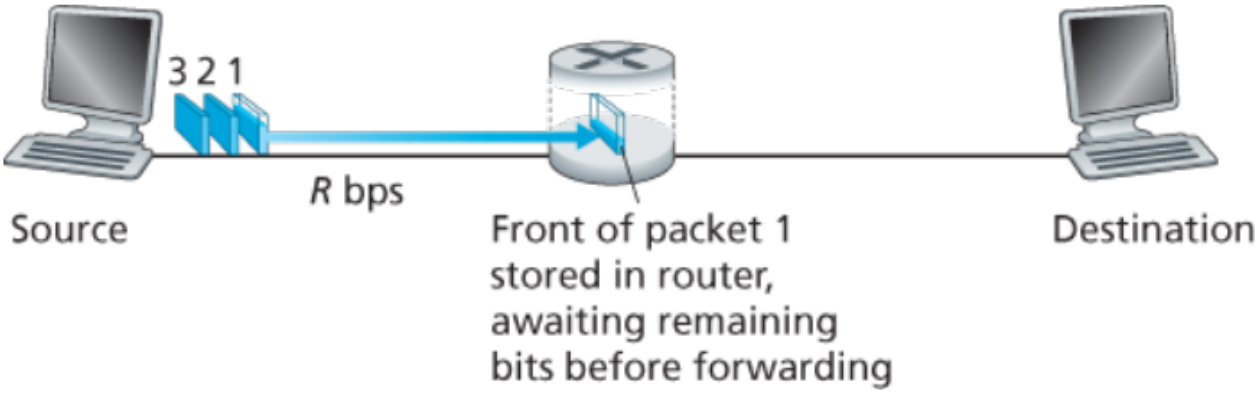
Packets are transmitted over each communication link at a rate equal to the *full* transmission rate of the link. So, **if a source end system or a packet switch is sending a packet of *L* bits over a link with transmission rate *R* bits/ sec, then the time to transmit the packet is L/R seconds.**

`packet of 10 bits + transmission rate of 20 bits/s => 10/20 s = 0.5 s`

Network core



Store-and-forward transmission means that the packet switch **must receive the entire packet before it can begin to transmit** the first bit of the packet onto the outbound link. To explore store-and-forward transmission in more detail, consider a simple network consisting of two end systems connected by a single router.

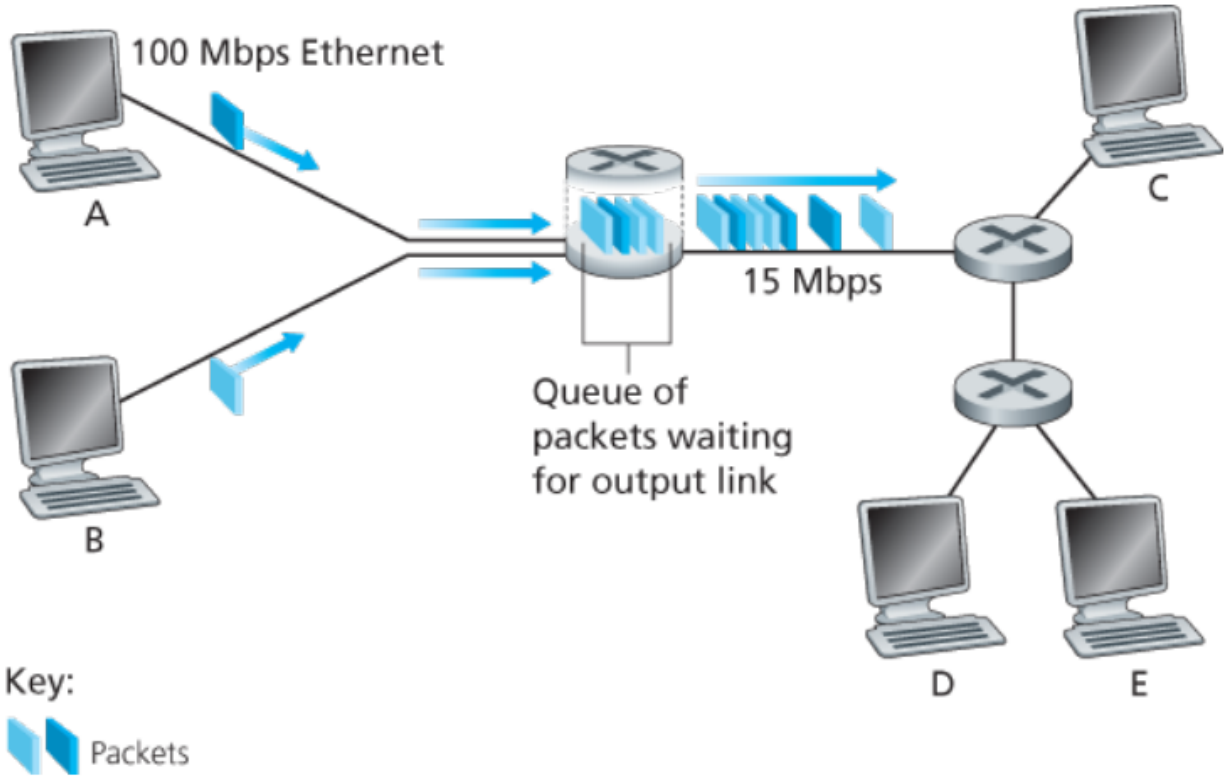


Only after the router has received all of the packet's bits can it begin to transmit (i.e., "forward") the packet onto the outbound link.

`time 4 * L/R to recieve all three packets`

Let's now consider the general case of sending one packet from source to destination over a path consisting of N links each of rate R (thus, there are *N-1 routers between source and destination*). Applying the same logic as above, we see that the end-to-end delay is: NLR

Each packet switch has multiple links attached to it. For each attached link, the packet switch has an output buffer (also called an output queue), which stores packets that the router is about to send into that link. The output buffers play a key role in packet switching. If an arriving packet needs to be transmitted onto a link but finds the link busy with the transmission of another packet, the arriving packet must wait in the output buffer. **Thus, in addition to the store-and-forward delays, packets suffer output buffer queuing delays.** These delays are variable and depend on the level of congestion in the network.



When a packet arrives at a router, the router examines the address and searches its forwarding table, using this destination address, to find the appropriate outbound link. The router then directs the packet to this outbound link.

Circuit switching vs Packet switching

In **circuit-switched** networks, the resources needed along a path (buffers, link transmission rate) to provide for communication between the end systems are **reserved** for the duration of the communication session between the end systems. In **packet-switched** networks, these resources are **not reserved**; a session's messages use the resources on demand and, as a consequence, may have to wait (that is, queue) for access to a communication link.
As a simple analogy, consider two restaurants, one that requires reservations and another that neither requires reservations nor accepts them. For the restaurant that requires reservations, we have to go through the hassle of calling before we leave home. But when we arrive at the restaurant we can, in principle, immediately be seated and order our meal. For the restaurant that does not require reservations, we don't need to bother to reserve a table. But when we arrive at the restaurant, we may have to wait for a table before we can be seated.

Example

Traditional telephone networks are examples of circuit-switched networks. Consider what happens when one person wants to send information (voice or facsimile) to another over a telephone network. Before the sender can send the information, the network must establish a connection between the sender and the receiver.
In the jargon of telephony, this connection is called a circuit. When the network establishes the circuit, it also reserves a constant transmission rate in the network's links (representing a fraction of each link's transmission capacity) for the duration of the connection. Since a given transmission rate has been reserved for this sender-toreceiver connection, the sender can transfer the data to the receiver at the guaranteed constant rate.

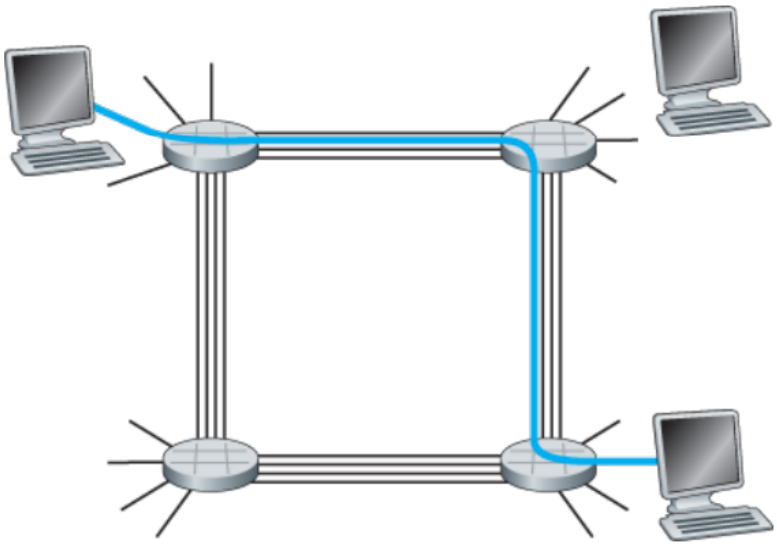
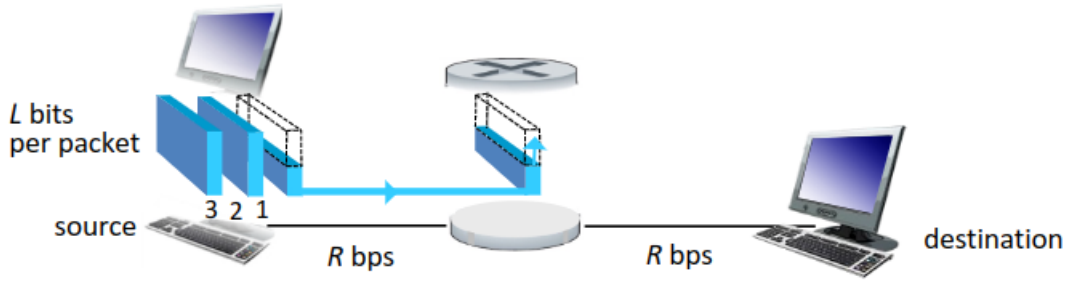


Figure 1.13 A simple circuit-switched network consisting of four switches and four links

Packet-switching: each packet is transmitted at full link capacity

Packet-switching: store-and-forward



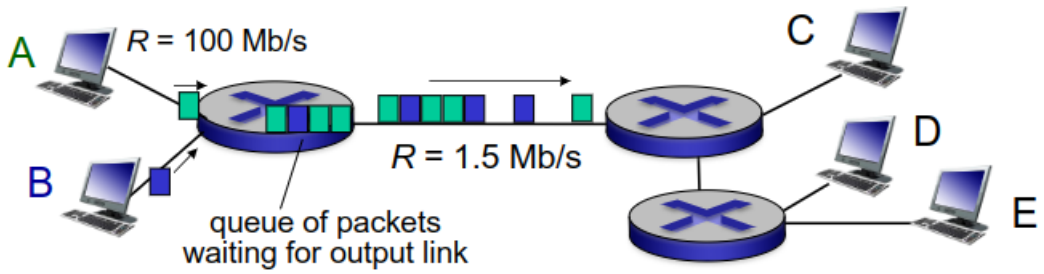
- takes *L*/*R* seconds to transmit (push out) *L*-bit packet into link at *R* bps
- *store and forward:* entire packet must arrive at router before it can be transmitted on next link
- end-end delay = 2*L*/*R* (assuming zero propagation delay)

one-hop numerical example:

- *L* = 7.5 Mbits
- *R* = 1.5 Mbps
- one-hop transmission delay = 5 sec

more on delay shortly …

Packet Switching: queueing delay, loss



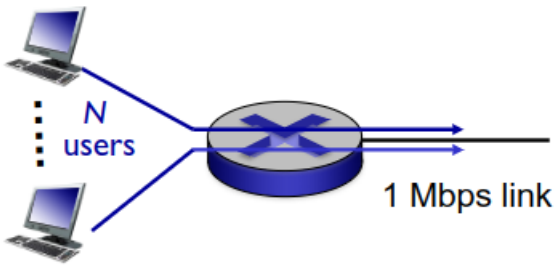
queuing and loss:

- if arrival rate (in bits) to link exceeds transmission rate of link for a period of time:
  - packets will queue, wait to be transmitted on link
  - packets can be dropped (lost) if memory (buffer) fills up

Packet switching versus circuit switching

packet switching allows more users to use network!

example:
- 1 Mb/s link
- each user:
  - 100 kb/s when "active"
  - active 10% of time



- *circuit-switching:*
  - 10 users
- *packet switching:*
  - with 35 users, probability > 10 active at same time is less than .0004 *

Q: how did we get value 0.0004?

Q: what happens if > 35 users ?

Exercises/ examples: https://gaia.cs.umass.edu/kurose_ross/interactive/

Delay, Loss and throughput in Packet-Switched Networks

As a packet travels from one node (host or router) to the subsequent node (host or router) along this path, the packet suffers from several types of delays at each node along the path. The most important of these delays are the nodal processing delay, queuing delay, transmission delay, and propagation delay; together, these delays accumulate to give a total nodal delay. The performance of many Internet applications—such as search, Web browsing, e-mail, maps, instant messaging, and voice-over-IP—are greatly affected by network delays. In order to acquire a deep understanding of packet switching and computer networks, we must understand the nature and importance of these delays.

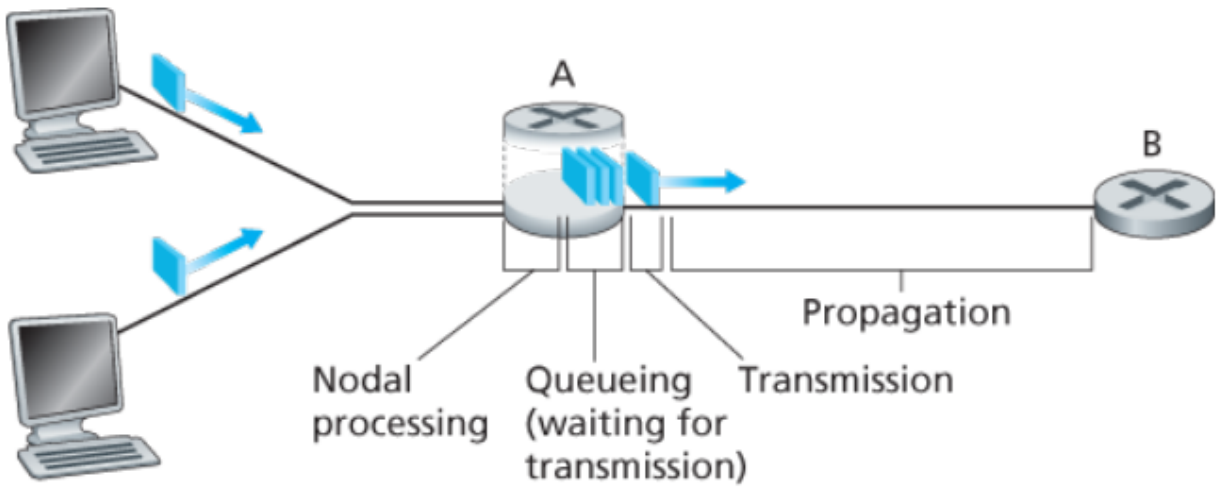


Figure 1.16 The nodal delay at router A

The time required to examine the packet's header and determine where to direct the packet is part of the **processing delay**. The processing delay can also include other factors, such as the time needed to check for bit-level errors in the packet that occurred in transmitting the packet's bits from the upstream node to router A. Processing delays in high-speed routers are typically on the order of microseconds or less.

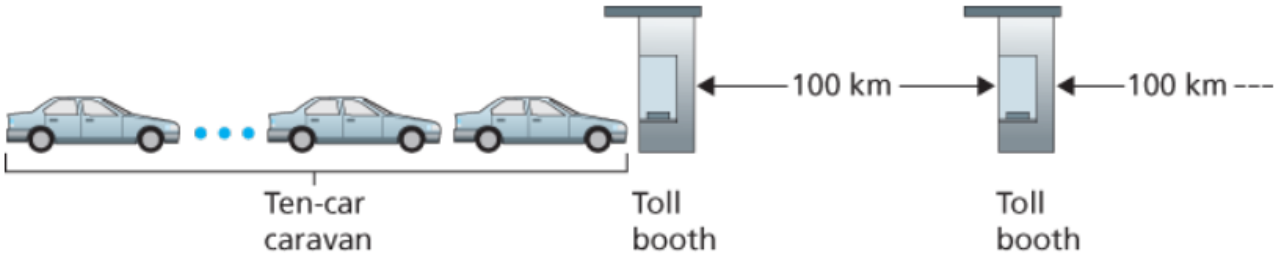
At the queue, the packet experiences a **queuing delay** as it waits to be transmitted onto the link. The length of the queuing delay of a specific packet will depend on the number of earlier-arriving packets that are queued and waiting for transmission onto the link. If the queue is empty and no other packet is currently being transmitted, then our packet's queuing delay will be zero.

Denote the length of the packet by L bits, and denote the transmission rate of the link from router A to router B by R bits/sec. For example, for a 10 Mbps Ethernet link, the rate is for a 100 Mbps Ethernet link, the rate is The **transmission delay** is L/R. This is the amount of time required to push (that is, transmit) all of the packet's bits into the link. Transmission delays are typically on the order of microseconds to milliseconds in practice.

Once a bit is pushed into the link, it needs to propagate to router B. The time required to propagate from the beginning of the link to router B is the propagation delay. The bit propagates at the propagation speed of the link. The propagation speed depends on the physical medium of the link (equal or a little less than the speed of light). The **propagation delay** is d/s, where d is the distance between router A and router B and s is the propagation speed of the link.

> Note
>
> Newcomers to the field of computer networking sometimes have difficulty understanding the difference between transmission delay and propagation delay. The difference is subtle but important. The transmission delay is the amount of time required for the router to push out the packet; it is a function of the packet's length and the transmission rate of the link, but has nothing to do with the distance between the two routers. The propagation delay, on the other hand, is the time it takes a bit to propagate from one router to the next; it is a function of the distance between the two routers, but has nothing to do with the packet's length or the transmission rate of the link.

Example:



Suppose that whenever the first car of the caravan arrives at a tollbooth, it **waits** at the entrance until the other nine cars have arrived and lined up behind it. (Thus the entire caravan must be stored at the tollbooth before it can begin to be forwarded.) The time required for the tollbooth to push the entire caravan onto the highway is 120s. This time is analogous to the transmission delay in a router. The time required for a car to travel from the exit of one tollbooth to the next tollbooth is 1h. This time is analogous to propagation delay. Therefore, the time from when the caravan is stored in front of a tollbooth until the caravan is stored in front of the next tollbooth is the sum of transmission delay and propagation delay—in this example, 62 minutes.

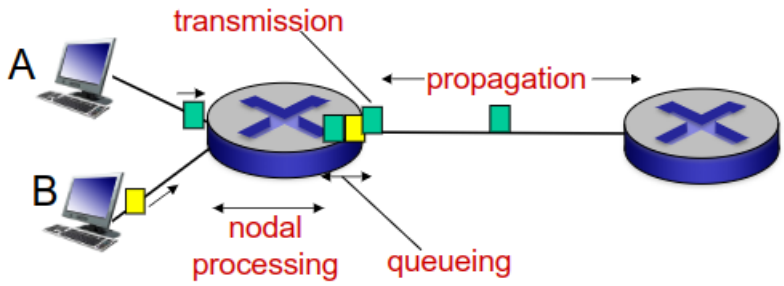
Let's explore this analogy a bit more. What would happen if the tollbooth service time for a caravan were greater than the time for a car to travel between tollbooths? For example, suppose now that the cars travel at the rate of 1,000 km/hour and the tollbooth services cars at the rate of one car per minute. Then the traveling delay between two tollbooths is 6 minutes and the time to serve a caravan is 10 minutes. In this case, the first few cars in the caravan will arrive at the second tollbooth before the last cars in the caravan leave the first tollbooth. **This situation also arises in packet-switched networks—the first bits in a packet can arrive at a router while many of the remaining bits in the packet are still waiting to be transmitted by the preceding router.**

- cars "propagate" at 100 km/hr
- toll booth takes 12 sec to service car (bit transmission time)
- car ~ bit; caravan ~ packet
- *Q:* How long until caravan is lined up before 2nd toll booth?

- time to "push" entire caravan through toll booth onto highway = 12*10 = 120 sec
- time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr)= 1 hr
- *A:* 62 minutes

- suppose cars now "propagate" at 1000 km/hr
- and suppose toll booth now takes one min to service a car
- *Q:* Will cars arrive to 2nd booth before all cars serviced at first booth?
  - *A: Yes!* after 7 min, first car arrives at second booth; three cars still at first booth

Four sources of packet delay



$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{proc}: nodal processing

- check bit errors
- determine output link
- typically < msec

d_{queue}: queueing delay

- time waiting at output link for transmission
- depends on congestion level of router

d_{trans}: transmission delay:

- *L*: packet length (bits)
- *R*: link *bandwidth (bps)*
- $d_{trans} = L/R$

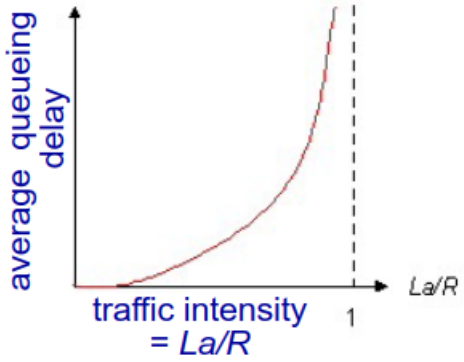
d_{prop}: propagation delay:

- *d*: length of physical link
- *s*: propagation speed (~$2x10^8$ m/sec)
- $d_{prop} = d/s$

d_{trans} and d_{prop} *very* different

Queueing delay (revisited)

- *R:* link bandwidth (bps)
- *L:* packet length (bits)
- a: average packet arrival rate (pkts/sec)



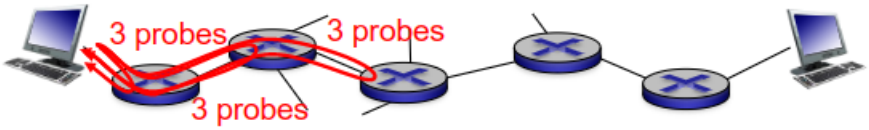
- *La/R* ~ 0: avg. queueing delay small
- *La/R* > 1: avg. queueing delay large
- *La/R* > 1: more "work" arriving than can be serviced, average delay infinite!


La/R ~ 0

La/R -> 1

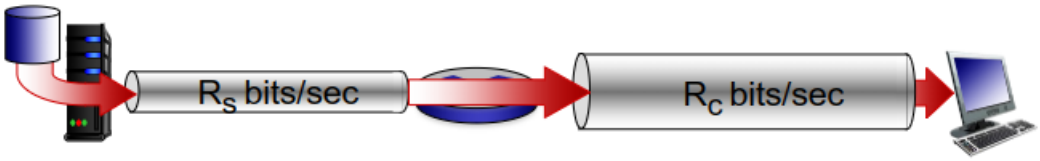
"Real" Internet delays and routes

- what do "real" Internet delay & loss look like?
- `traceroute` program: provides delay measurement from source to router along end-end Internet path towards destination. For all *i:*
  - sends three packets that will reach router *i* on path towards destination
  - router *i* will return packets to sender
  - sender times interval between transmission and reply.

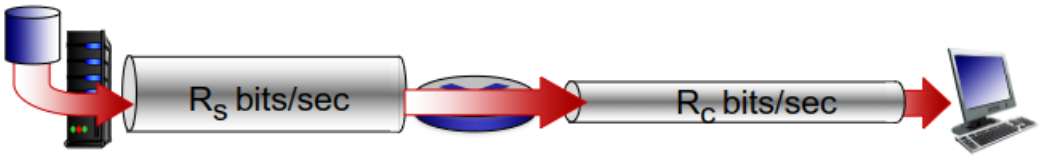


Throughput:

- $R_s < R_c$ What is average end-end throughput?

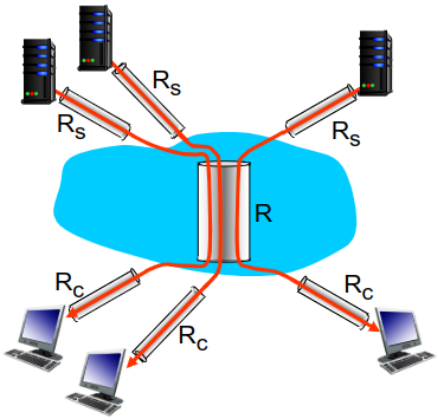


- $R_s > R_c$ What is average end-end throughput?



bottleneck link

link on end-end path that constrains end-end throughput



10 connections (fairly) share backbone bottleneck link *R* bits/sec

Protocol Layers

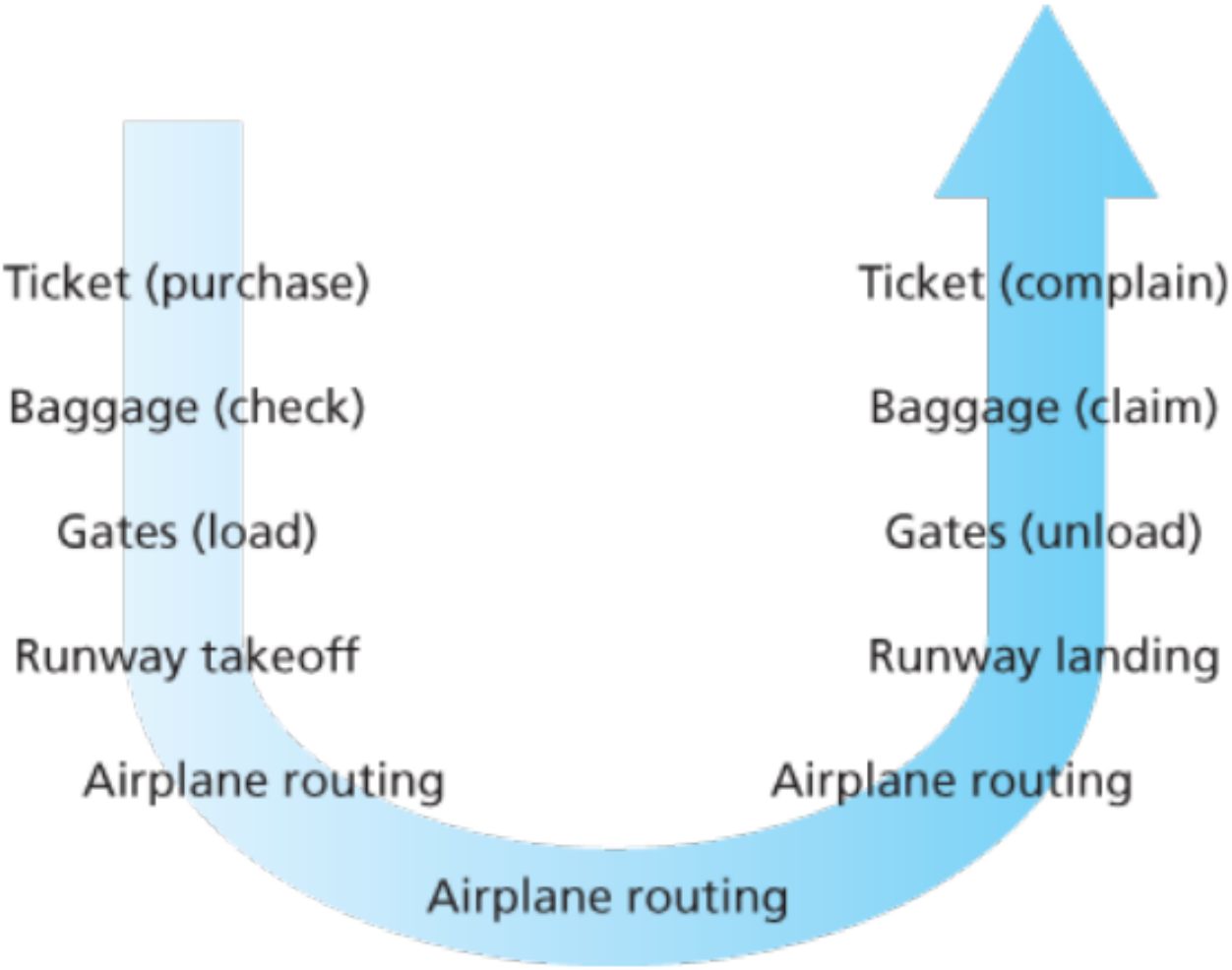
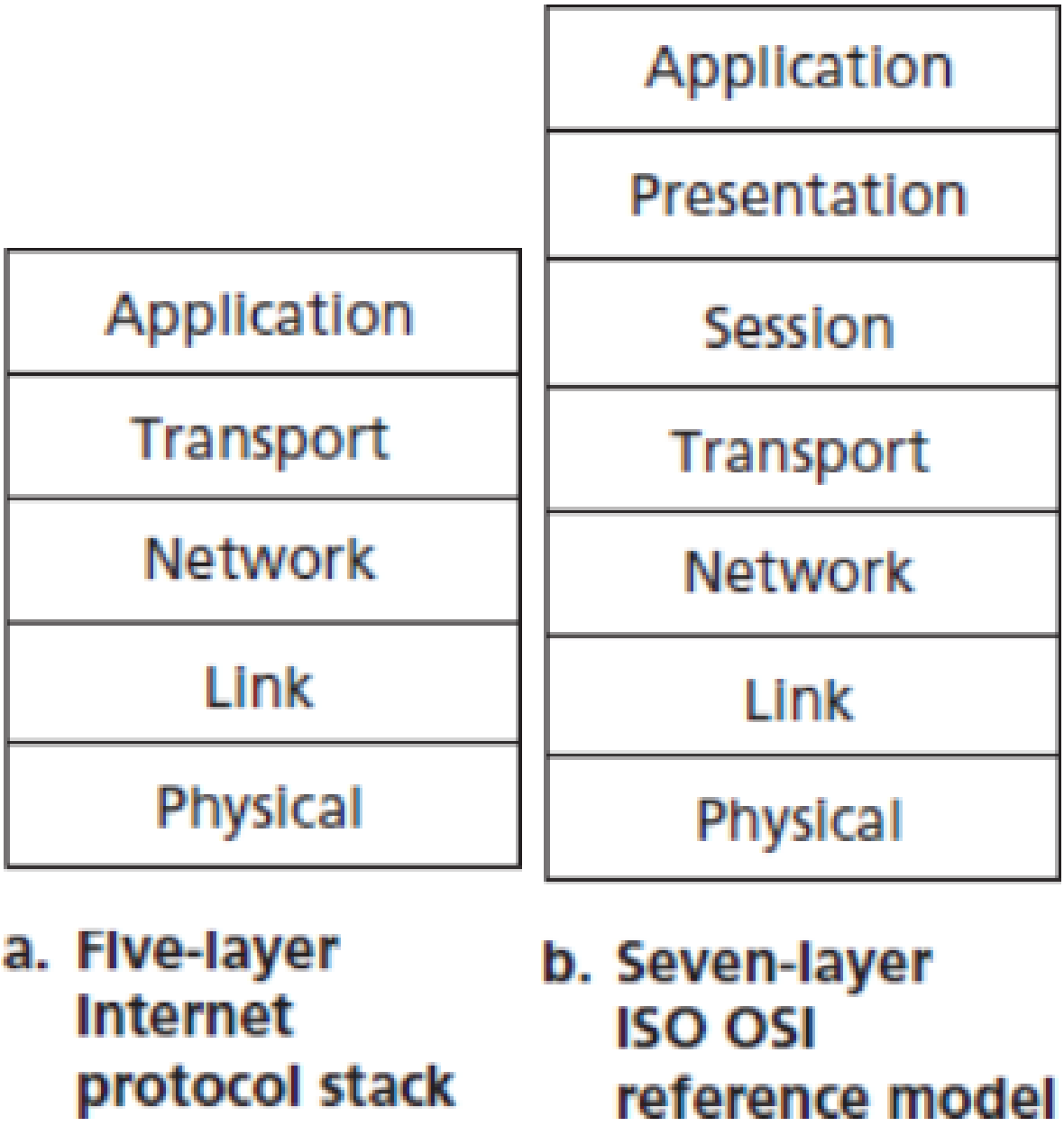


Figure 1.21 Taking an airplane trip: actions

Note that there is a ticketing function at each end; there is also a baggage function for already-ticketed passengers, and a gate function for already-ticketed and already-baggagechecked passengers.

For passengers who have made it through the gate (that is, passengers who are already ticketed, baggage-checked, and through the gate), there is a takeoff and landing function, and while in flight, there is an airplane-routing function.

If the gate functions were changed (for instance, to have people board and disembark by height), the remainder of the airline system would remain unchanged since the gate layer still provides the same function (loading and unloading people); it simply implements that function in a different manner after the change. For large and complex systems that are constantly being updated, **the ability to change the implementation of a service without affecting other components of the system is another important advantage of layering.**



The **application layer** is where network applications and their application-layer protocols reside. The Internet's application layer includes many protocols, such as the HTTP protocol (which provides for Web document request and transfer), SMTP (which provides for the transfer of e-mail messages), and FTP (which provides for the transfer of files between two end systems).

The Internet's **network layer** is responsible for moving network-layer packets known as datagrams from one host to another. The Internet transport-layer protocol (TCP or UDP) in a source host passes a transport-layer segment and a destination address to the network layer, just as you would give the postal service a letter with a destination address. The network layer then provides the service of delivering the segment to the transport layer in the destination host. The Internet's network layer includes the celebrated IP protocol, which defines the fields in the datagram as well as how the end systems and routers act on these fields.

The Internet's network layer routes a datagram through a series of routers between the source and destination. To move a packet from one node (host or router) to the next node in the route, the network layer relies on the services of the link layer. In particular, at each node, the network layer passes the datagram down to the link layer, which delivers

the datagram to the next node along the route. At this next node, the link layer passes the datagram up to the network layer. `[...]` (ethernet, wifi, etc.)

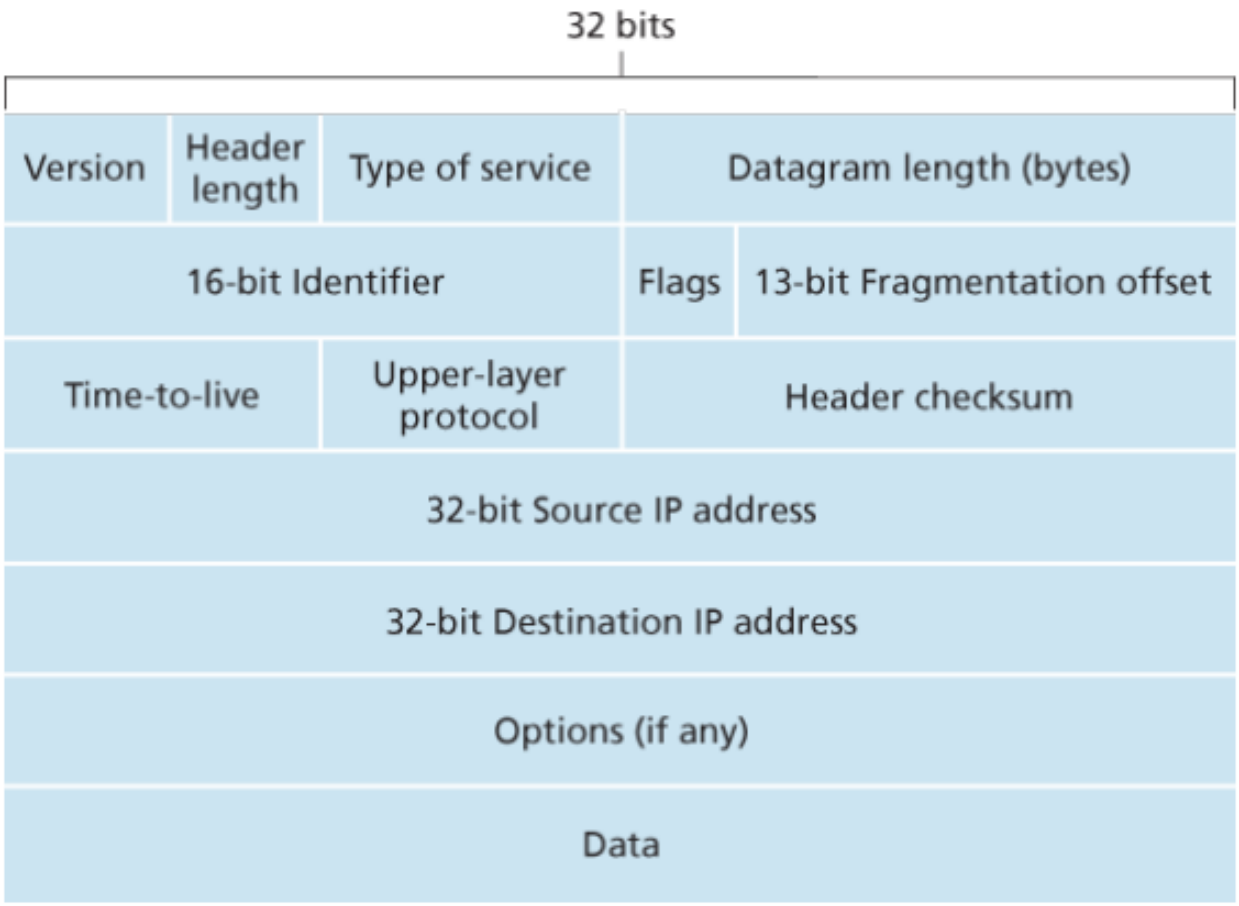


Figure 4.16 IPv4 datagram format

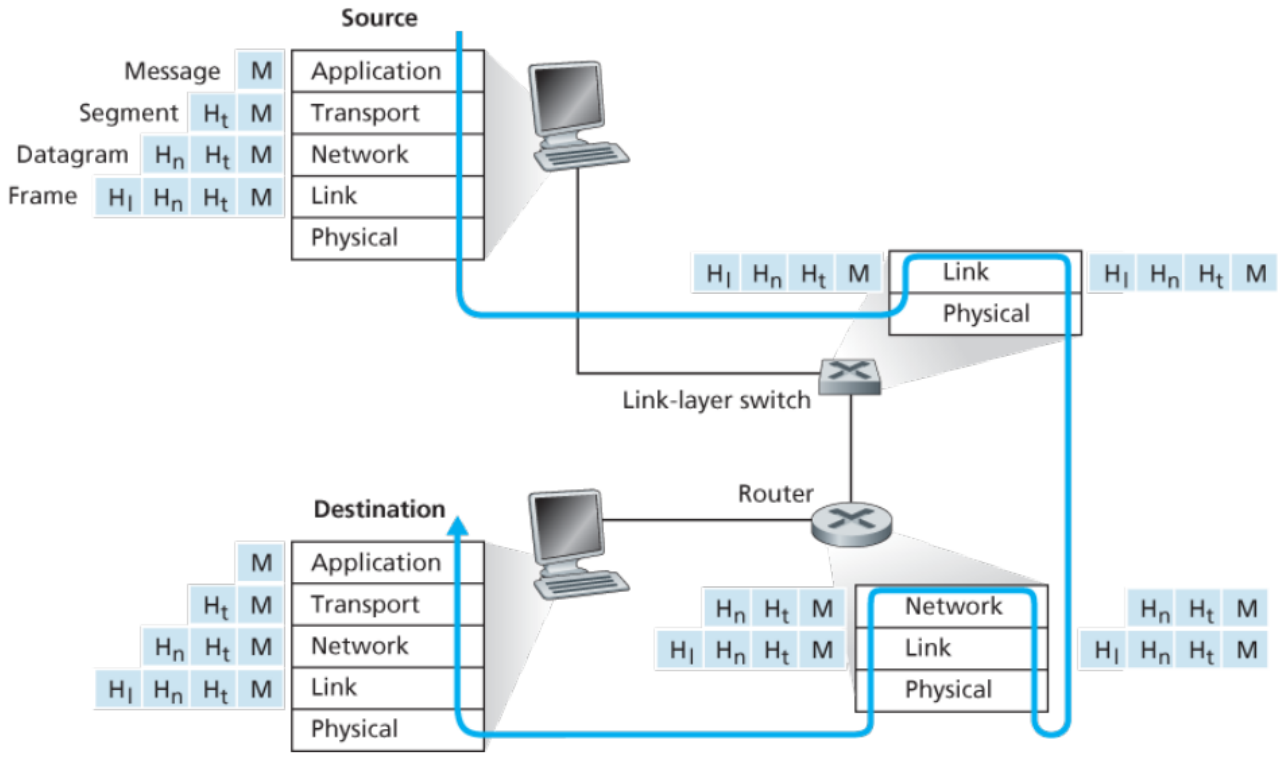
- **Version number.** These 4 bits specify the IP protocol version of the datagram. By looking at the version number, the router can determine how to interpret the remainder of the IP datagram. Different versions of IP use different datagram formats. The datagram format for IPv4 is shown in Figure 4.16. The datagram format for the new version of IP (IPv6) is discussed in Section 4.3.5.
- **Header length.** Because an IPv4 datagram can contain a variable number of options (which are included in the IPv4 datagram header), these 4 bits are needed to determine where in the IP datagram the payload (e.g., the transport-layer segment being encapsulated in this datagram) actually begins. Most IP datagrams do not contain options, so the typical IP datagram has a 20-byte header.
- **Type of service.** The type of service (TOS) bits were included in the IPv4 header to allow different types of IP datagrams to be distinguished from each other. For example, it might be useful to distinguish real-time datagrams (such as those used by an IP telephony application) from non-real-time traffic (for example, FTP). The specific level of service to be provided is a policy issue determined and configured by the network administrator for that router. We also learned in Section 3.7.2 that two of the TOS bits are used for Explicit Congestion Notification.
- **Datagram length.** This is the total length of the IP datagram (header plus data), measured in bytes. Since this field is 16 bits long, the theoretical maximum size of the IP datagram is 65,535 bytes. However, datagrams are rarely larger than 1,500 bytes, which allows an IP datagram to fit in the payload field of a maximally sized Ethernet frame.
- **Identifier, flags, fragmentation offset.** These three fields have to do with so-called IP fragmentation, a topic we will consider shortly. Interestingly, the new version of IP, IPv6, does not allow for fragmentation.
- **Time-to-live.** The time-to-live (TTL) field is included to ensure that datagrams do not circulate forever (due to, for example, a long-lived routing loop) in the network. This field is decremented by one each time the datagram is processed by a router. If the TTL field reaches 0, a router must drop that datagram.

- **Time-to-live.** The time-to-live (TTL) field is included to ensure that datagrams do not circulate forever (due to, for example, a long-lived routing loop) in the network. This field is decremented by one each time the datagram is processed by a router. If the TTL field reaches 0, a router must drop that datagram.
- **Protocol.** This field is typically used only when an IP datagram reaches its final destination. The value of this field indicates the specific transport-layer protocol to which the data portion of this IP datagram should be passed. For example, a value of 6 indicates that the data portion is passed to TCP, while a value of 17 indicates that the data is passed to UDP. For a list of all possible values,

see [IANA Protocol Numbers 2016]. Note that the protocol number in the IP datagram has a role that is analogous to the role of the port number field in the transport-layer segment. The protocol number is the glue that binds the network and transport layers together, whereas the port number is the glue that binds the transport and application layers together. We'll see in Chapter 6 that the link-layer frame also has a special field that binds the link layer to the network layer.

- **Header checksum.** The header checksum aids a router in detecting bit errors in a received IP datagram. The header checksum is computed by treating each 2 bytes in the header as a number and summing these numbers using 1s complement arithmetic. As discussed in Section 3.3, the 1s complement of this sum, known as the Internet checksum, is stored in the checksum field. A router computes the header checksum for each received IP datagram and detects an error condition if the checksum carried in the datagram header does not equal the computed checksum. Routers typically discard datagrams for which an error has been detected. Note that the checksum must be recomputed and stored again at each router, since the TTL field, and possibly the options field as well, will change. An interesting discussion of fast algorithms for computing the Internet checksum is [RFC 1071]. A question often asked at this point is, why does TCP/IP perform error checking at both the transport and network layers? There are several reasons for this repetition. First, note that only the IP header is checksummed at the IP layer, while the TCP/UDP checksum is computed over the entire TCP/UDP segment. Second, TCP/UDP and IP do not necessarily both have to belong to the same protocol stack. TCP can, in principle, run over a different network-layer protocol (for example, ATM) [Black 1995]) and IP can carry data that will not be passed to TCP/UDP.
- **Source and destination IP addresses.** When a source creates a datagram, it inserts its IP address into the source IP address field and inserts the address of the ultimate destination into the destination IP address field. Often the source host determines the destination address via a DNS lookup, as discussed in Chapter 2. We'll discuss IP addressing in detail in Section 4.3.3.

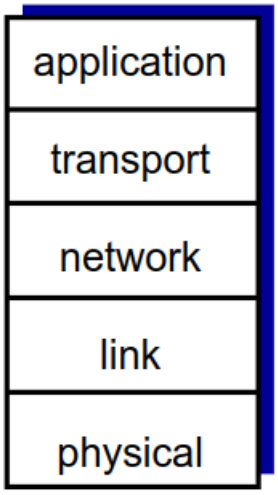
While the job of the **link layer** is to move entire frames from one network element to an adjacent network element, the job of the physical layer is to move the individual bits within the frame from one node to the next. The protocols in this layer are again link dependent and further depend on the actual transmission medium of the link (for example, twisted-pair copper wire, single-mode fiber optics). For example, Ethernet has many physical-layer protocols: one for twisted-pair copper wire, another for coaxial cable, another for fiber, and so on. In each case, a bit is moved across the link in a different way.



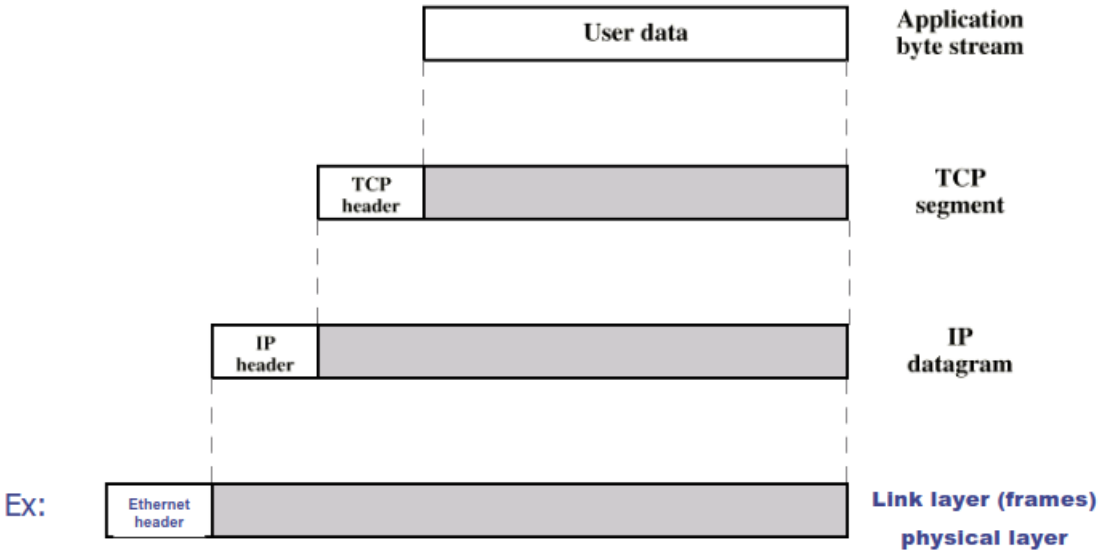
A useful **analogy** here is the sending of an interoffice memo from one corporate branch office to another via the public postal service. Suppose Alice, who is in one branch office, wants to send a memo to Bob, who is in another branch office. The memo is analogous to the application-layer message. Alice puts the memo in an interoffice envelope with Bob's name and department written on the front of the envelope. The interoffice envelope is analogous to a transport-layer segment—it contains header information (Bob's name and department number) and it encapsulates the application-layer message (the memo). When the sending branch-office mailroom receives the interoffice envelope, it puts the interoffice envelope inside yet another envelope, which is suitable for sending through the public postal service. The sending mailroom also writes the postal address of the sending and receiving branch offices on the postal envelope. Here, the postal envelope is analogous to the datagram—it encapsulates the transport layer segment (the interoffice envelope), which encapsulates the original message (the memo). The postal service delivers the postal envelope to the receiving branch-office mailroom. There, the process of de-encapsulation is begun. The mailroom extracts the interoffice memo and forwards it to Bob. Finally, Bob opens the envelope and removes the memo.

- *application:* supporting network applications
  - FTP, SMTP, HTTP
- *transport:* process-process data transfer
  - TCP, UDP
- *network:* routing of datagrams from source to destination
  - IP, routing protocols
- *link:* data transfer between neighboring network elements
  - Ethernet, 802.11 (Wi-Fi), PPP

physical: bits "on the wire"



Estratégia: encapsular a unidade dados na camada prot. inferior



Chapter 4 - Network Layer

- datagram network provides network-layer connectionless service
- virtual-circuit network provides network-layer connection service

Virtual circuits

"source-to-dest path behaves much like telephone circuit"
- performance-wise
- network actions along source-to-dest path

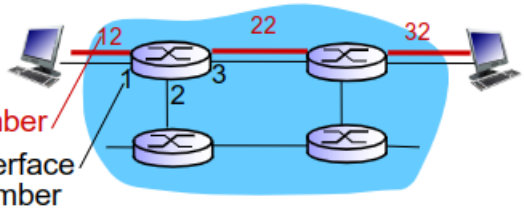
- call setup, teardown for each call *before* data can flow
- each packet carries VC identifier (not destination host address)
- every router on source-dest path maintains "state" for each passing connection
- link, router resources (bandwidth, buffers) may be allocated to VC

a VC consists of:

1. *path* from source to destination
2. *VC numbers*, one number for each link along path
3. *entries in forwarding tables* in routers along path

- packet belonging to VC carries VC number (rather than dest address)
- VC number can be changed on each link.
  - new VC number comes from forwarding table

VC forwarding table

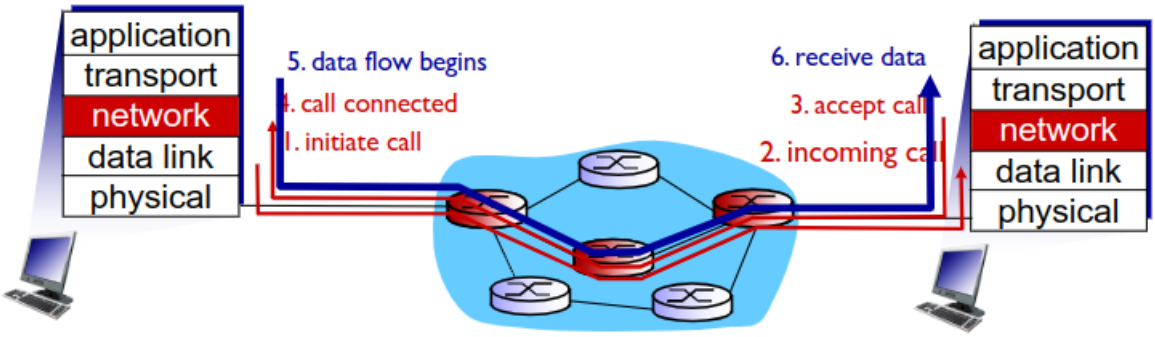


forwarding table in northwest router:

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
…	…	…	…

VC routers maintain connection state information!

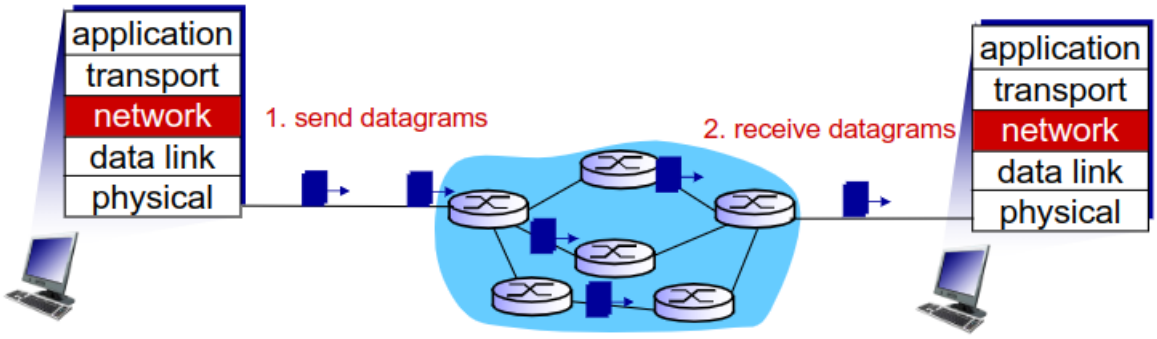
Signaling protocols in virtual circuits:



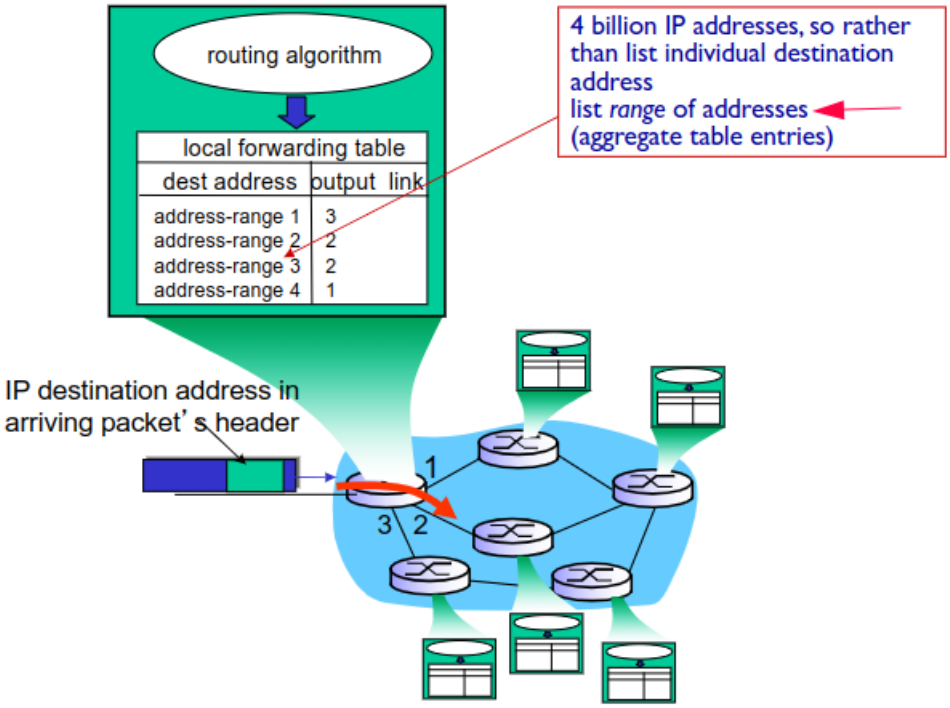
> (not used in today's internet at network layer)

Datagram networks

- no call setup at network layer
- routers: no state about end-to-end connections
  - no network-level concept of "connection"
- packets forwarded using destination host address



Datagram forwarding table



longest prefix matching

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

examples:

DA: 11001000 00010111 00011110 10100001 which interface?

DA: 11001000 00010111 00011000 10101010 which interface?

1º example -> 2º interface

2º example -> 1º interface

Datagram or VC network: why?

Internet (datagram)

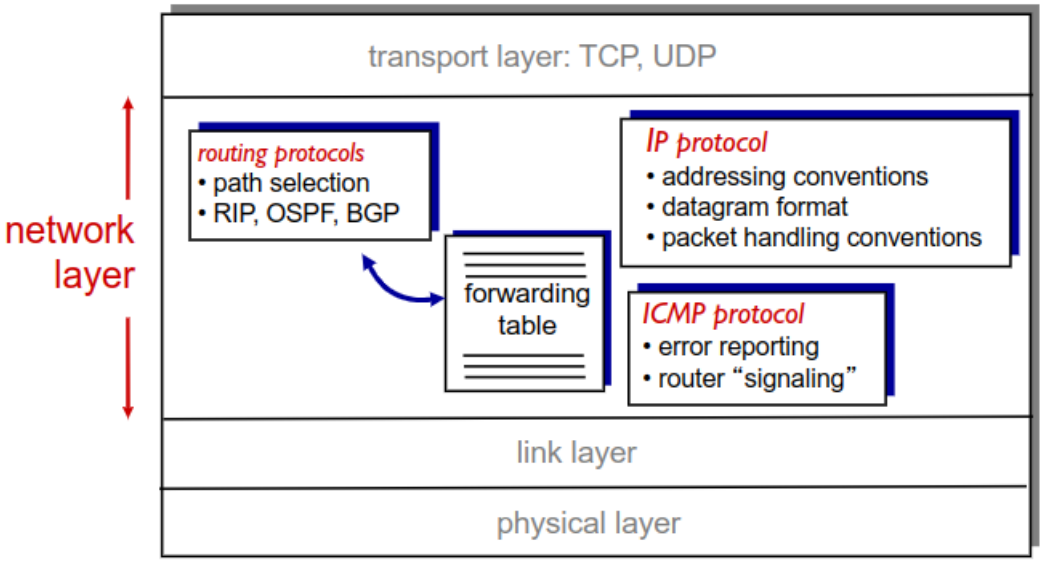
- data exchange among computers
  - "elastic" service, no strict timing requirements
- many link types
  - different characteristics
  - uniform service difficult
- "smart" end systems (computers)
  - can adapt, perform control, error recovery
  - ***simple inside network, complexity at "edge"***

ATM (VC)

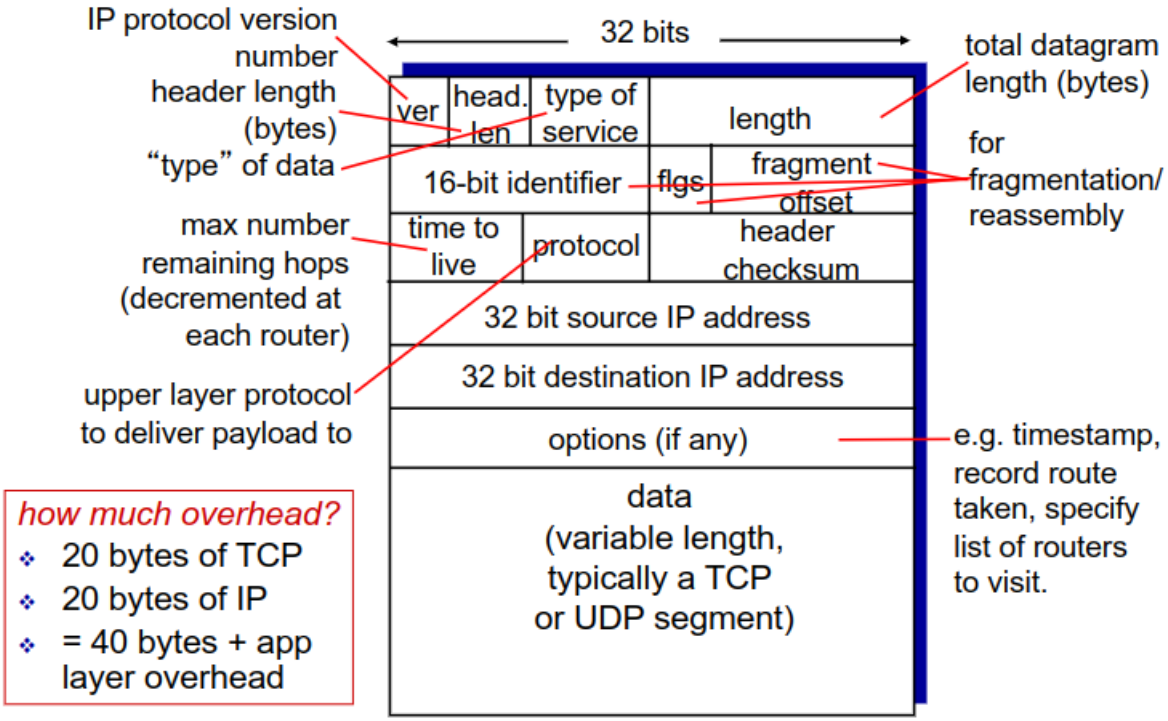
- evolved from telephony
- human conversation:
  - strict timing, reliability requirements
  - need for guaranteed service
- "dumb" end systems
  - telephones
  - ***complexity inside network***

Função	Rede de Datagramas	Rede de Circuitos Virtuais (VC)
Estabelecimento prévio da conexão (ou circuito)	Não é necessário	É necessário
Endereçamento	Endereço de origem e destino em cada PDU	PDUs contêm o identificador do circuito
Routing / Forwarding	PDUs são encaminhados de forma independente entre si	A rota é estabelecida inicialmente e todos os PDUs utilizam essa rota
Informação de estado	não é necessária	necessária por VC
Falha de um elemento de rede	não é normalmente problemática	todos os VC são terminados
Controlo de tráfego e Controlo de congestão	difícil	fácil, se os recursos atribuídos são suficientes

IP: Internet Protocol



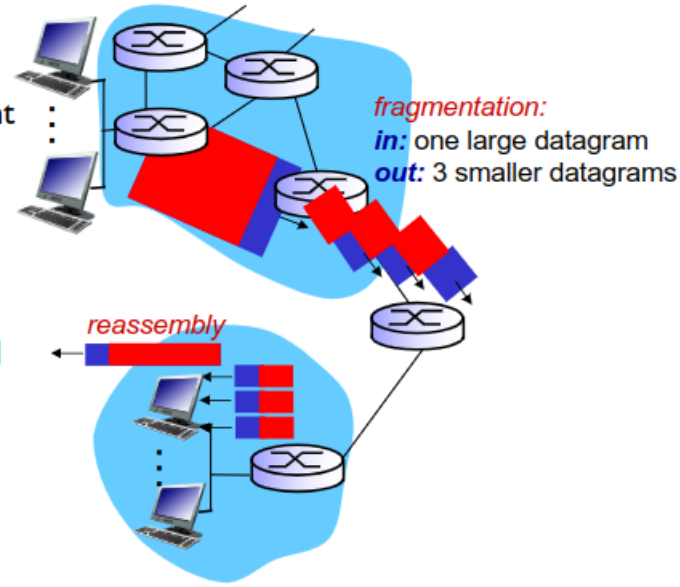
IP datagram format



how much overhead?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

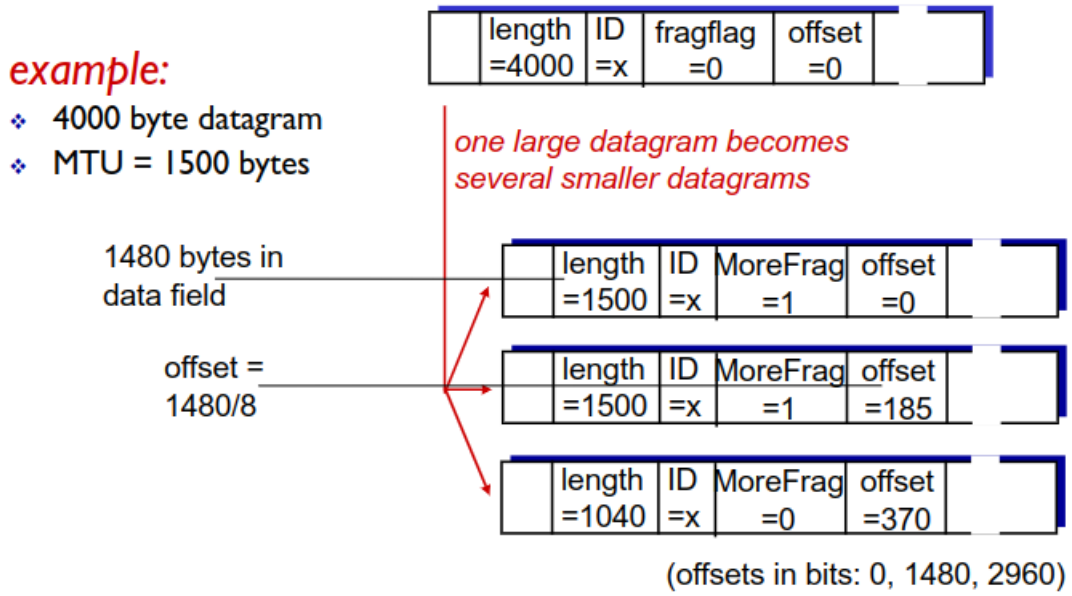
- network links have MTU (max transfer unit) size – the largest possible link-level frame
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify order of related fragments



Campos manipulados na fragmentação IPv4:

- *identification* - identifica fragmentos pertencentes ao mesmo datagrama original
- *more fragments* - *flag* que determina se há mais fragmentos e também saber se o fragmento é o último
- *may fragment* - identificação da possibilidade ou não do datagrama ser fragmentado pela rede
- *fragment offset* - *offset* dos dados do fragmento relativamente ao datagrama original

Em IPv6, por defeito, não está prevista fragmentação!



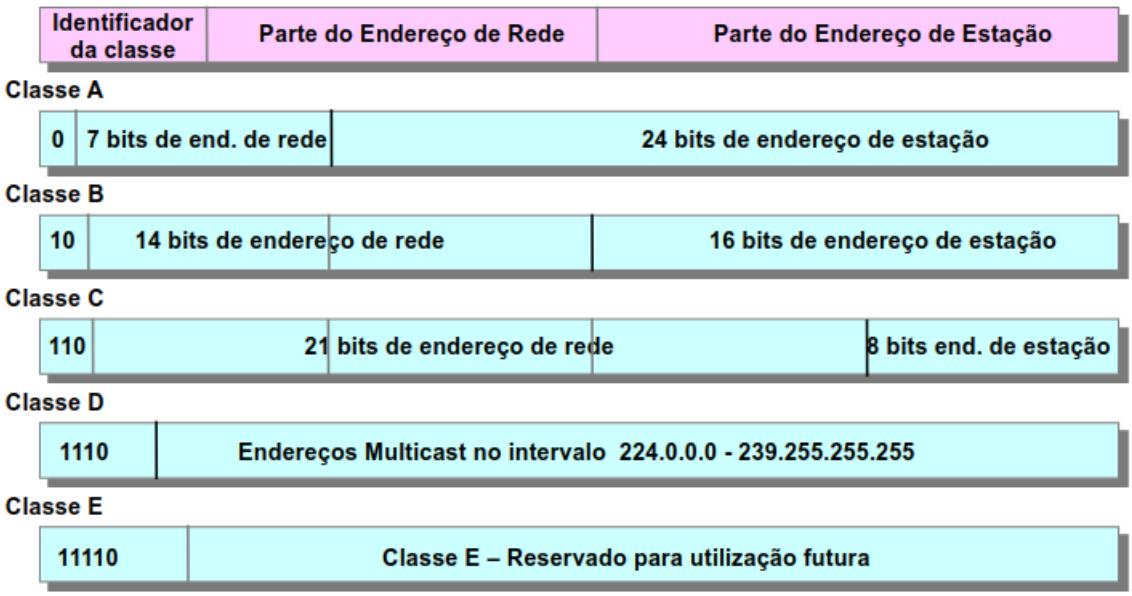
\20 byte header in each packet, so the original has 3980 bytes of data: $1480 + 1480 + 1020 = 3980$.
Dividing the offset by 8 allows it to fit in 13 bits instead of 16. This means every packet but the last must contain a number of data bytes that is a multiple of 8, which isn't a problem.

> The fragment offset is measured in Units of 8 bytes each. It has 13 bit field in the IP header. As said in the RFC page 17
> "*This field indicates where in the datagram this fragment belongs.The fragment offset is measured in units of 8 octets (64 bits). The first fragment has offset zero.*"

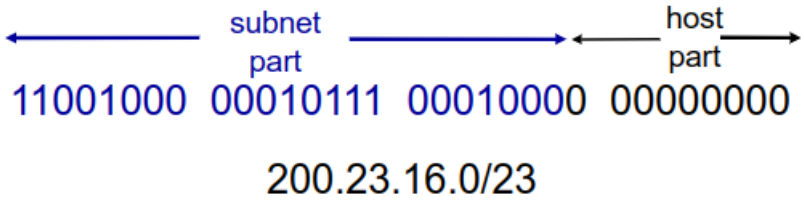
IP addressing

IPv4: 32-bit unsigned binary value
`xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx` (*dot decimal notation*)
One part identifies the network, and the other identifies the host:
`<network id><host id>`

Internet assigned number authority:

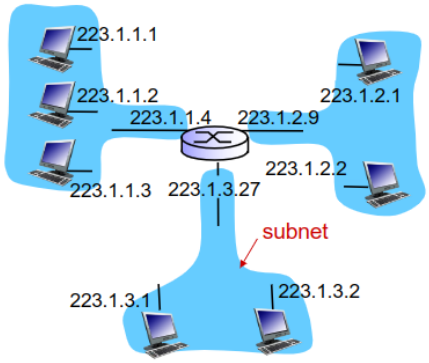


CIDR: Classes **InterDomain** Routing



What is a subnet?

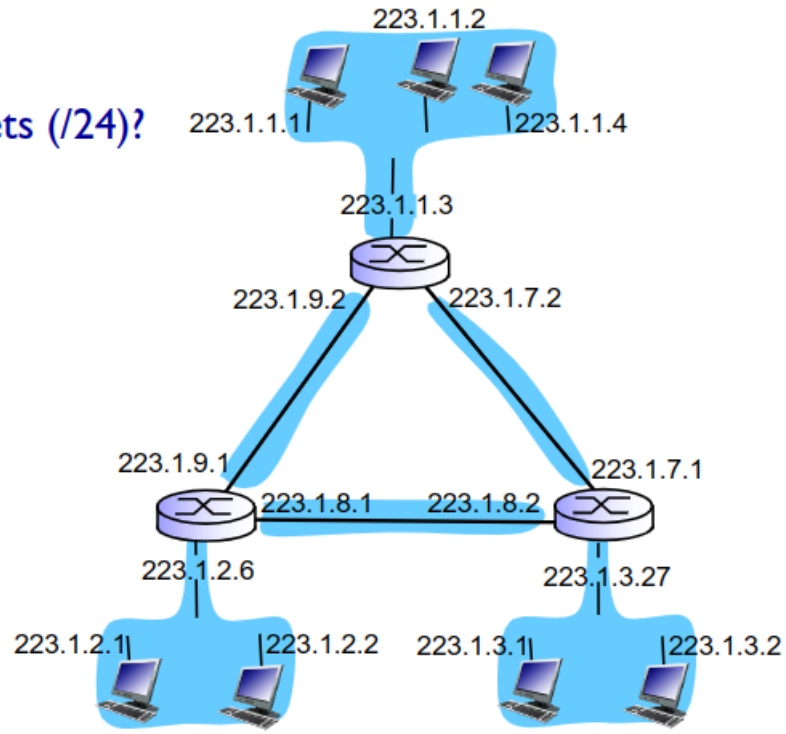
- device interfaces with same subnet part of IP address
- can physically reach each other without intervening router



network consisting of 3 subnets (/24)

Subnets

how many subnets (/24)?



> 6 subnets

- enables better address organization and management
- allows to introduce more hierarchy levels for routing
- however, reduces addressing space

Private addresses (without global IP connectivity):

192.168.0.0 - 192.168.255.255 / 16
172.16.0.0 - 172.31.255.255 /12
10.0.0.0 - 10.255.255.255 /8

> Note: host with many interfaces is called *multihome*

Routing

Routing table:

> netstat -nr

destination	next_hop	netmask	flags	use	interface
default	192.110.1.254	0.0.0.0	UG	102410	tu0
192.110.1.0	192.110.1.240	255.255.255.0	UH	234576	tu0
.....	.....	.....	.....	.....	.....
192.168.1.0	192.110.1.253	255.255.255.0	UG	124586	tu0

```
arch :: ~ % netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG        0 0          0 wlan0
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0 wlan0
```

Forwarding algorithm:

Entrega (forwarding):

- É facilitada pelo endereçamento hierárquico
- O endereço IP é: **a.b.c.d/m = X.Y** (rede.estação)

1) usar **máscara** para extrair o endereço de rede **X**
2) procurar entrada que melhor se ajuste a **X**
   se **X** é local, entregar na interface **X.Y** (entrega directa)
   senão usar **X** para determinar o próximo salto (*next hop*);
3) A entrada por defeito (**0.0.0.0/0 ou default)** ajusta-se a todos os **X**

Supernetting:

Tratar das redes diretamente ligadas, tráfego interno, e rota por defeito! A rota por defeito tem de garantir acesso ao à rede onde o router está ligado. Neste caso é RTR. Nas redes internas só precisamos de saber que são acessíveis via RTR2 (neste caso). Especificamos isso através do IP de RTR2.

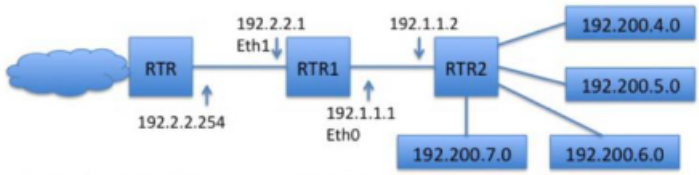


Tabela de encaminhamento de RTR1 - sem Supernetting

Destino	Próximo Nó	Máscara	Interface
192.2.2.0	192.2.2.1	255.255.255.0	Eth1
192.1.1.0	192.1.1.1	255.255.255.0	Eth0
192.200.4 (0000 0100).0	**192.1.1.2**	**255.255.255.0**	**Eth0**
192.200.5 (0000 0101).0	**192.1.1.2**	**255.255.255.0**	**Eth0**
192.200.6 (0000 0110).0	**192.1.1.2**	**255.255.255.0**	**Eth0**
192.200.7 (0000 0111).0	**192.1.1.2**	**255.255.255.0**	**Eth0**
Default	192.2.2.254	0.0.0.0	Eth1
192.200.4(0000 0100).0	**192.1.1.2**	**255.255.252 (11111100).0**	**Eth0**

As conexões internas partilham o mesmo próximo salto, o que permite fazer agregação de rotas. Para agregação de notas precisamos de alterar a máscara. Se olharmos para a parte em binário, há uma parte comum. Logo os dois últimos bits são irrelevantes para a forma de encaminhar. Abdicando deles, não criamos ambiguidades, mas temos uma só entrada na tabela de encaminhamento com a máscara 255.255.252.0. Reduz-se 4 entradas para 1 entrada.

IP routing: static vs. dynamic

Encaminhamento (routing):

a) Estático - baseado em rotas pré-definidas
- as rotas permanecem fixas
- reduz o tráfego na rede
- esquema simples mas pouco flexível

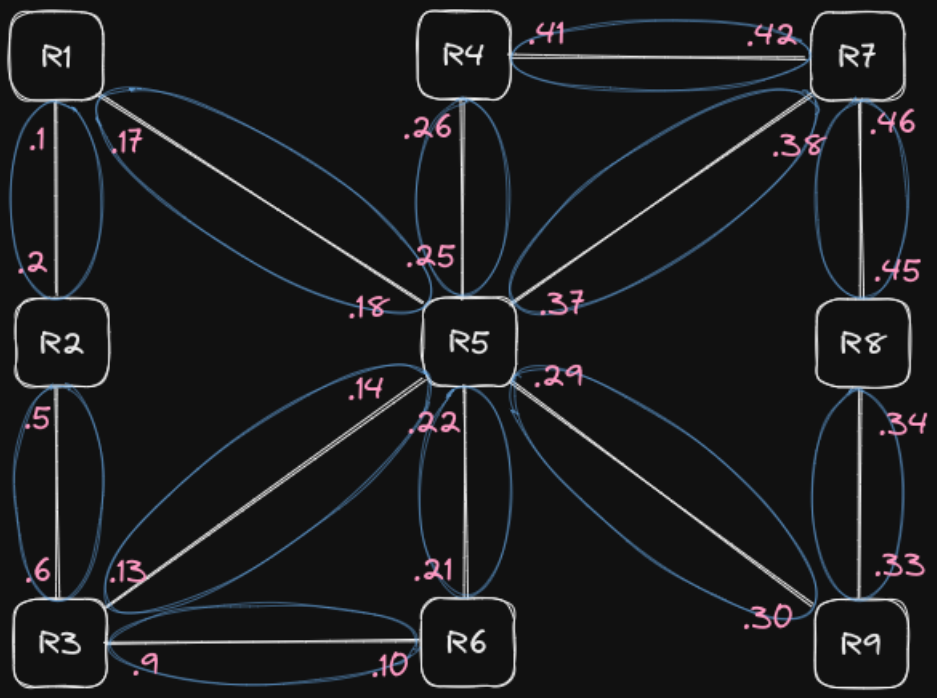
b) Dinâmico - rotas atualizadas ao longo do tempo
- os routers trocam informação de routing entre si
- esta actualização dinâmica de rotas é obtida através de protocolos específicos de encaminhamento (routing):
  - » RIP (Routing Information Protocol, OSPF (Open Shortest Path First), BGP (Border Gateway Protocol), etc.
- grande flexibilidade e adaptação (automática) a falhas ou mudanças na configuração de rede
- o tráfego de actualização pode causar sobrecarga na rede

- Computação dinâmica das rotas:
  - centralizada - cada router, conhecendo a topologia da área, determina o melhor caminho para os possíveis destinos dessa área
    - e.g. Dijkstra's Algorithm
  - distribuída - cada router envia informação das melhores rotas que conhece aos routers seus vizinhos (redes a que dá acesso)
    - e.g. Bellman-Ford Algorithm
- Princípio utilizado
  - Vector Distância (*Vector Distance*)
    - e.g. Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP) (desenvolvido pela Cisco) (distributed)
  - Estado das ligações (*Link State*)
    - e.g. Open Shortest Path First (OSPF) (uses Dijkstra's algorithm)

- Um router pode conhecer rotas estáticas e/ou dinâmicas para um mesmo destino, aprendidas por protocolos distintos.
- Como é seleccionada a "melhor" rota?
  - distância – indicador administrativo que permite estabelecer uma relação de preferência entre rotas aprendidas por protocolos de routing distintos.
  - métrica – indicador que traduz o custo de fazer forwarding por uma determinada interface, permitindo estabelecer uma relação de preferência entre rotas aprendidas pelo mesmo protocolo de routing.
- Um domínio administrativo com uma politíca de encaminhamento bem definida é designado Sistema Autónomo (e.g. ISPs têm atribuído um #AS único)

Supernetting & subnetting exercises:
`(https://i.imgur.com/ULDXU87.png)`

200.1.1.0/26



12 subredes
200.1.1.0/26 -> 6 bytes para host -> 2^6 endereços
64 endereços / 12 subredes = 5.333 ~= 4 bits para cada subrede

R1 200.1.1.0/30
R2 200.1.1.4/30
R3 200.1.1.8/30
(...)

Tabela de R1

Destination	Gateway	Mask
0.0.0.0	200.1.1.18/30	0.0.0.0
192.1.1.0/24	200.1.1.2/30	255.255.255.0
200.1.1.0 (1)	200.1.1.1/30 => 0.0.0.0	255.255.255.252
200.1.1.4	200.1.1.2/30	255.255.255.252 (2)
200.1.1.0	200.1.1.18/26	255.255.255.192 (3)
200.1.1.16 (1)	200.1.1.17/30	255.255.255.252

(1) Diretamente ligadas
- Rota por defeito: 0.0.0.0
- Rotas diretamentes ligadas: .0 e .16

(2) para descongestionar R5, vai logo para R2

(3) encaminhar gama de endereços para R5:

8: 00001000
12: 00001100

NAT

NAT: network address translation

motivation: local network may use just one IP address as far as outside world is concerned:

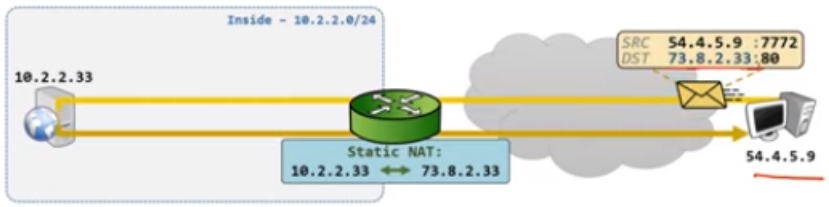
- range of addresses not needed from ISP: one public IP address can be used for all devices (then private addresses)
- can change addresses of devices in local network without notifying outside world
- can change ISP without changing addresses of devices in local network
- devices inside local net not explicitly addressable, visible by outside world (a security plus)

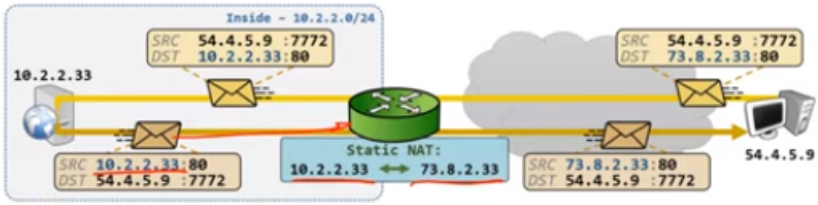
https://youtu.be/fm_NLOCi1Ns

Notes:
static NAT - explicit mapping between an IP address to another IP address
goal: makes internal resources externally accessible

static NAT translates the IP dest address in the package that is recieved; the ports did not change. outbound packet => translate the source
static NAT is bidirectional, doesnt matter who initiated
static NAT does not conserve any IP address

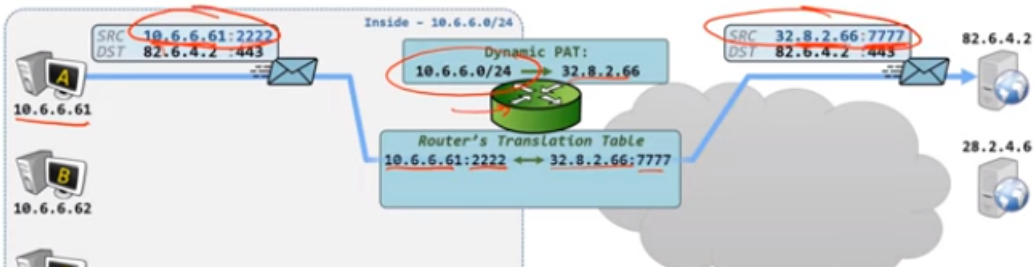
=> impossível, o IP é privado





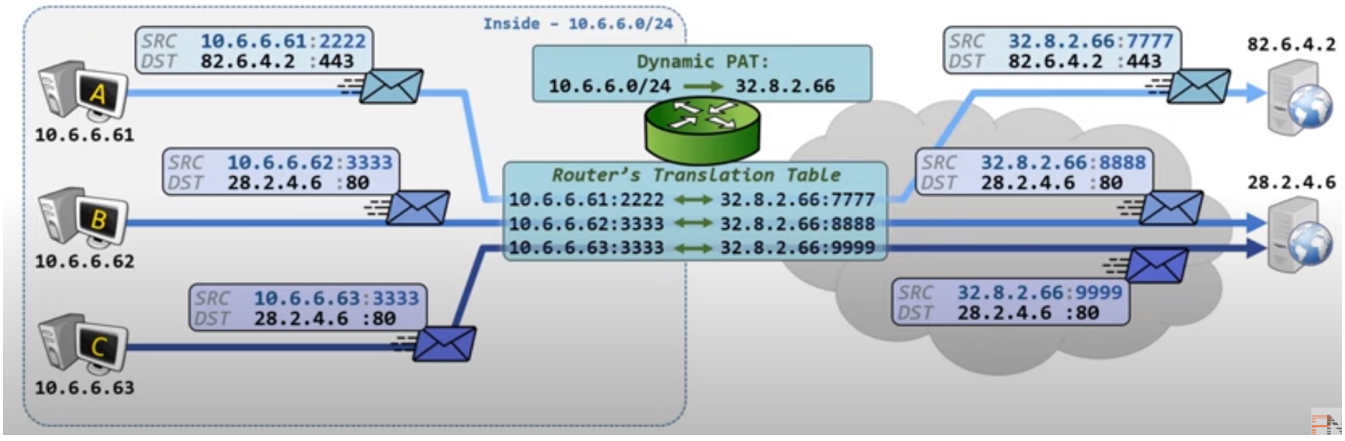
https://youtu.be/2t8-WRZWCq8

Dynamic NAT / PAT
Dynamic: the translation device chooses POST attributes



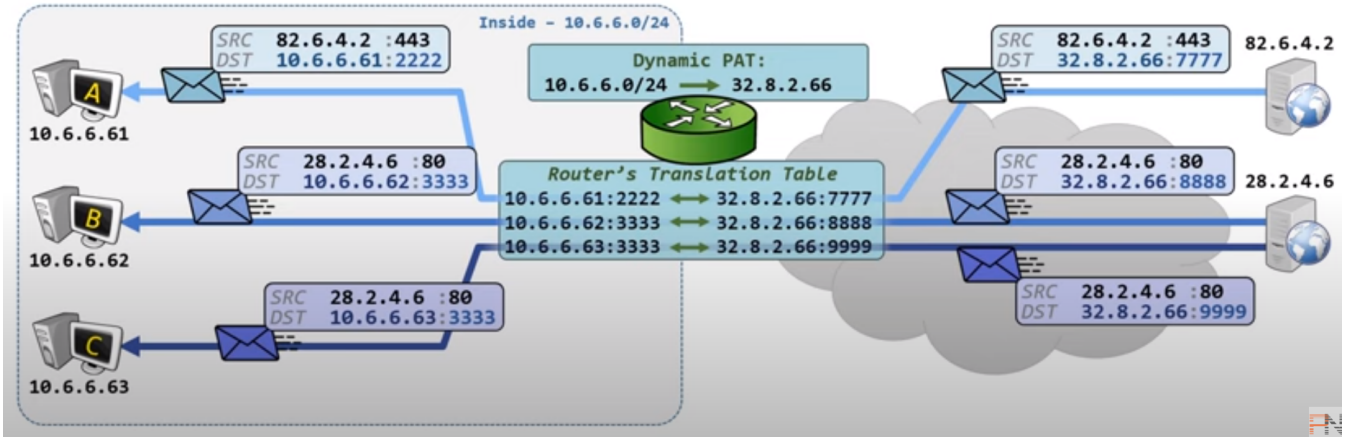
65 000 unique ports, thus max 65 000 concurrent connections per public IP address (more IP addresses could be added to the dynamic NAT pool)

Port 433 -> https request
Port 80 -> http request
Host A randomly selected port 2222
Host B randomly selected port 3333
Two host could generate the same source port value
=> source port re-randomized: assures each host has unique IP:Port on the outside; ensures router can "un-translate" packet back to original host

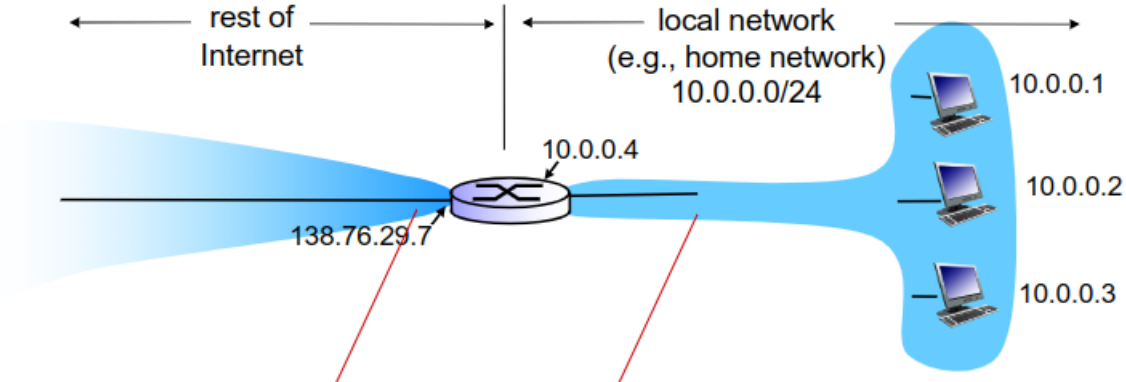


The trafic cannot be initiated by the outside host. PAT is unidirectional

Replies:

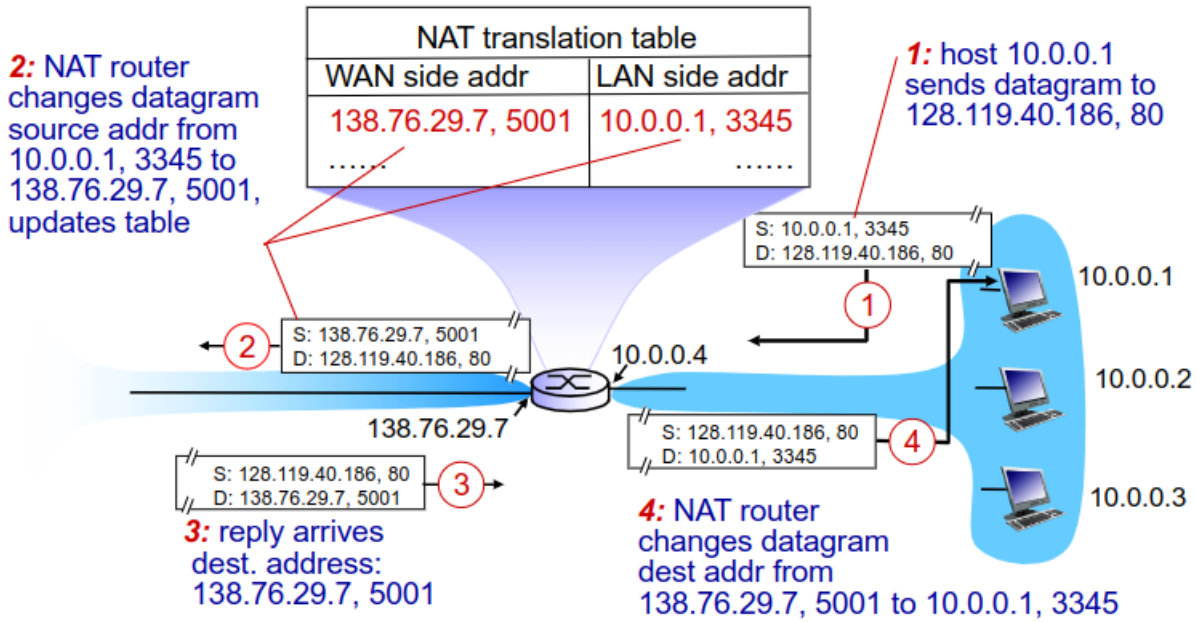


- Summary:
  - Admin defines set of pre and post translation attributes
  - Device determines actual post-translation IP Address and Port
  - Allows many hosts with Private IPs to share one Public IP
    - Greatest potential for IP address conservation
  - Translation Device assures unique IP:Port on the outside
    - ~65,000 concurrent connection per shared IP
  - Dynamic PAT is Unidirectional
    - Traffic must be initiated from Inside



e.g. all datagrams *leaving* local network have *same* single source NAT IP address: 138.76.29.7, different source port numbers (aka PAT)

datagrams with source or destination in this network have 10.0.0.0/24 address for source, destination (as usual)



IPv6

IPv6: motivation

- *initial motivation:* 32-bit address exhaustion…
- additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS (Quality of Service)

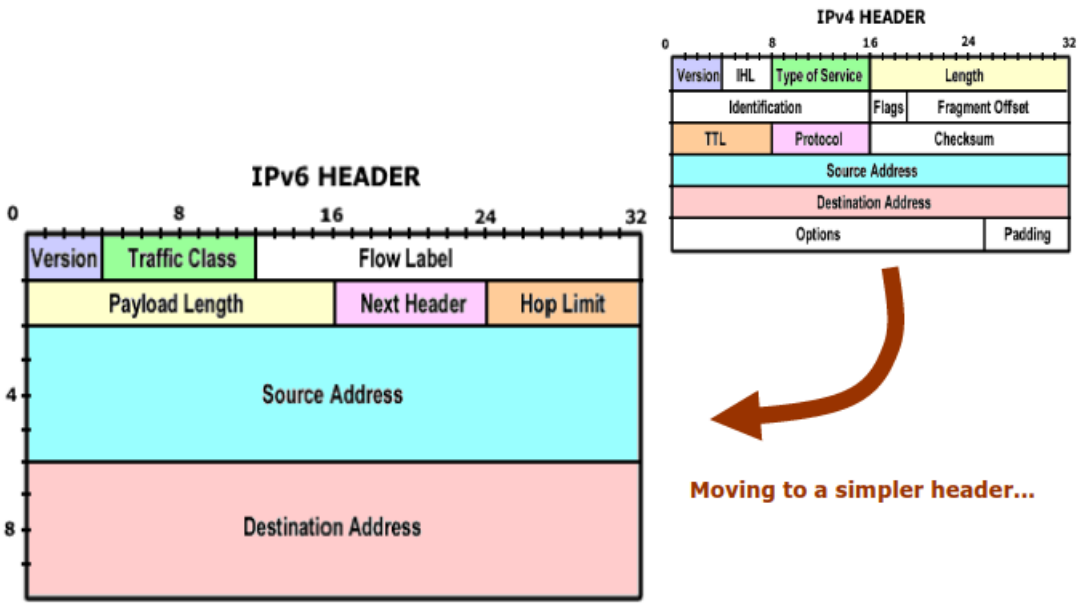
IPv6 datagram format:

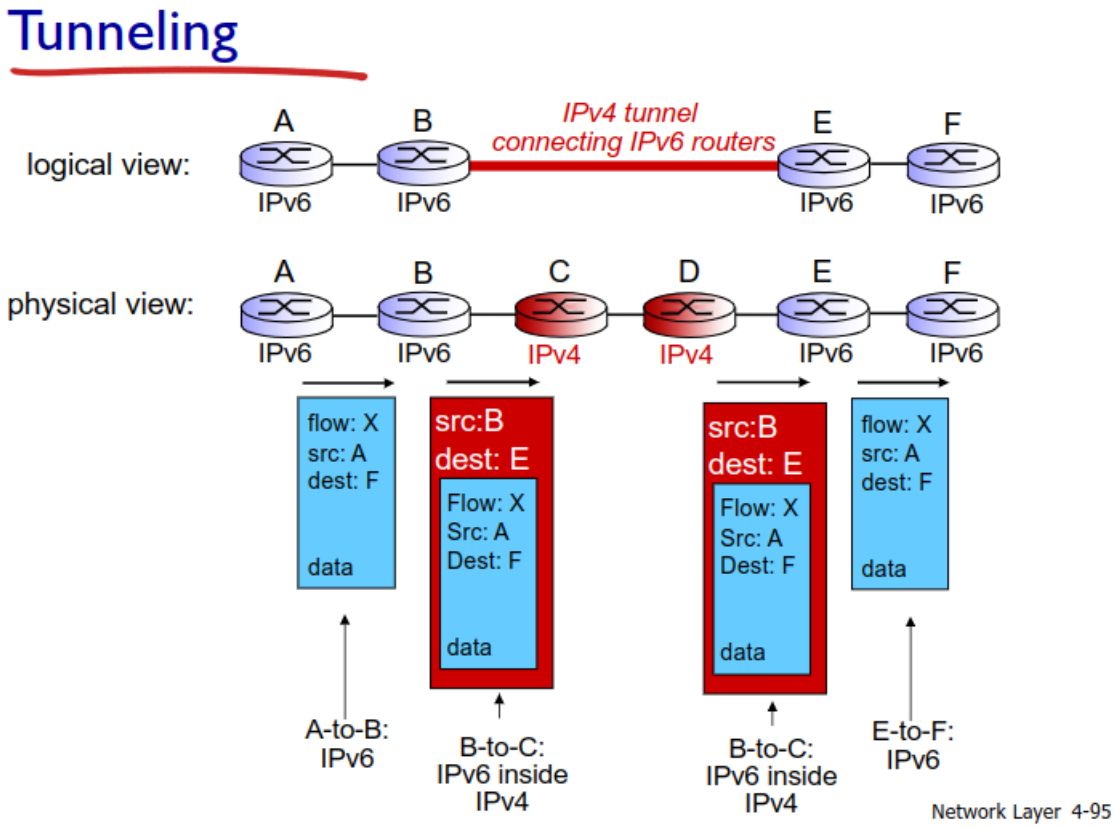
- fixed-length 40 byte header
- no fragmentation allowed, by default

Por defeito, fragmentação não é implementada em IPv6.
(Mas pode com *next header*).
Pretende-se que as aplicações produzam trafego de maneira que os pacotes não sejam demasiado grandes a nivel IP.

- *checksum:* removed entirely to reduce processing time at each hop
- *options:* allowed, but outside of header, indicated by "Next Header" field
- *ICMPv6:* new version of ICMP
  - additional message types, e.g. "ICMP Packet Too Big"
  - other control messages to support multicast, mobile IP, etc.

IPv6 é independente do layer 2.





Chapter 6 - Link Layer and LANs

Terminology:

- hosts and routers: **nodes**
- communication channels that connect adjacent nodes along communication path: **links** (wired/ wireless/ LANs)
- layer-2 packet: **frame**, encapsulates datagram

Data-link layer has the responsability of transferring datagram from one node to physically adjacent node over a link.

- datagram transferred by different link protocols over different links:
  - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- each link protocol provides different services
  - e.g., may or may not provide reliable data transfer (rdt) over each link

transportation analogy:

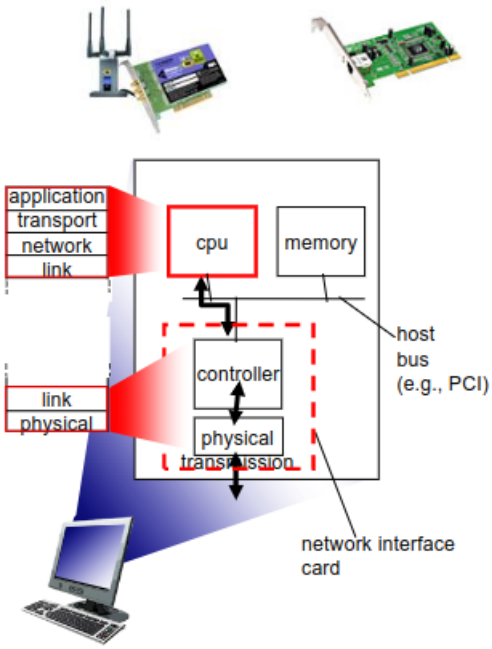
- trip from Princeton to Lausanne
  - car: Princeton to JFK
  - plane: JFK to Geneva
  - train: Geneva to Lausanne
- tourist = datagram
- transport segment = communication link
- transportation mode = link layer protocol
- travel agent = routing algorithm

Link layer services:

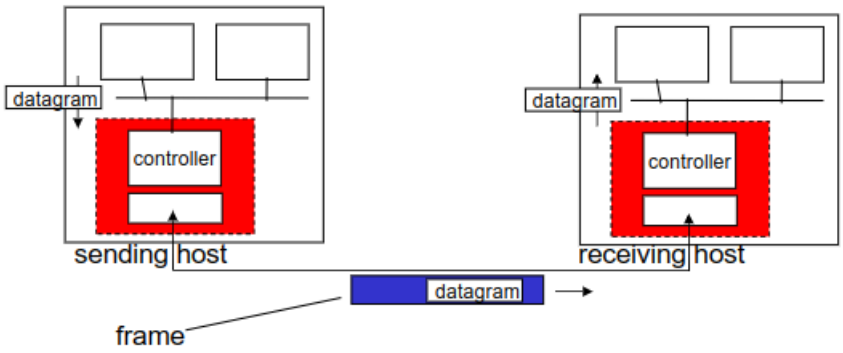
- **framing**, link acess:
  - encapsulate datagram into frame, adding header, trailer
  - obtain channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, destination
  - "encapsular o que o nível 3 passa numa unidade de dados nível 2"
- reliable delivery between adjacent nodes
- flow control
  - Serviço genérico que pode existir no nível 2. Regula a cadência entre nós adjacente. Um que envia e outro que recebe.
- error control
  - 2 tarefas : deteção e correção. Posso receber uma entrega não fiável e simplesmente descartar, sem corrigir. Códigos de correção são muito pouco usados porque possuem overhead muito grande. Utilizam-se então códigos de retransmissão. Redes cabeladas têm muitos poucos erros e é possível que nem tenham código de deteção de erros. Redes WiFi, como o meio é muito mais suscetível a erros, o código de deteção de erros está sempre presente.
- half-duplex and full duplex
  - with half duplex, nodes at both ends of link can transmit, but not at same time

Where is the link layer implemented?

- in each and every host
- link layer implemented in "adaptor" (aka network interface card NIC) or on a chip
  - Ethernet card, 802.11 card; Ethernet chipset
  - implements link, physical layer
- attaches into host's system buses
- combination of hardware, software, firmware



Adaptors communicating

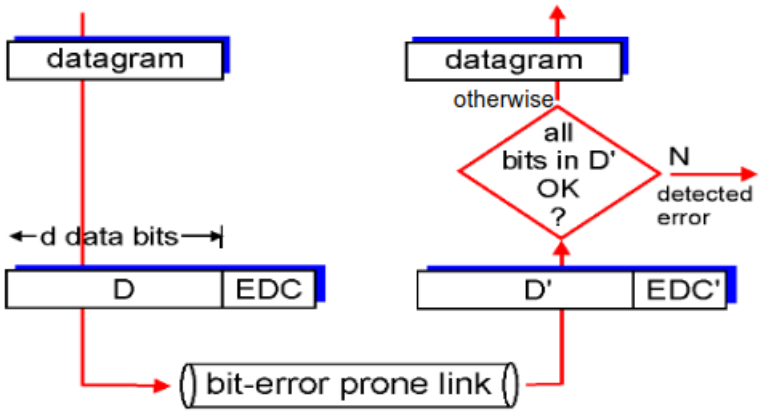


- sending side:
  - encapsulates datagram in frame
  - adds (eventually) error checking bits, reliability (rdt), flow control, etc.
- receiving side
  - if in use, looks for errors, rdt, flow control, etc.
  - extracts datagram, passes to upper layer at receiving side

Error detection

EDC= Error Detection and Correction bits (redundancy)
D = Data protected by error checking, may include header fields

- Error detection not 100% reliable!
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction

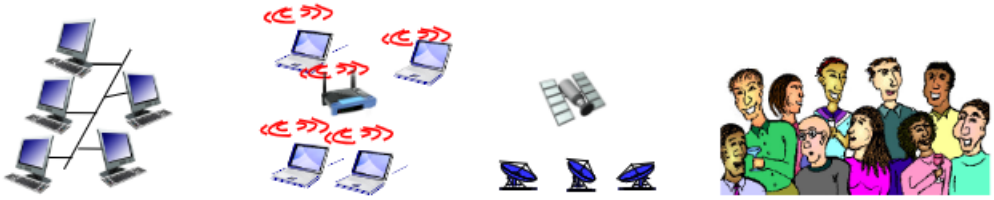


> single bit parity, two-dimensional bit parity, etc.

Multiple access links, protocols

two types of "links":

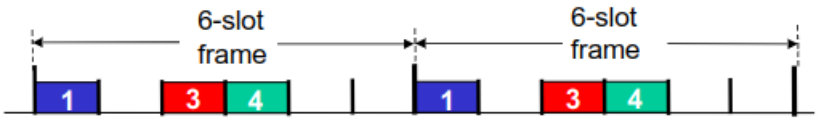
- point-to-point
  - (PPP for dial-up access; HDLC for point-to-point or point-to-multipoint)
  - point-to-point link between Ethernet switch, host
- broadcast (shared wire or medium)
  - old-fashioned Ethernet
  - upstream HFC (Hybrid Fiber Coax)
  - 802.11 wireless LAN



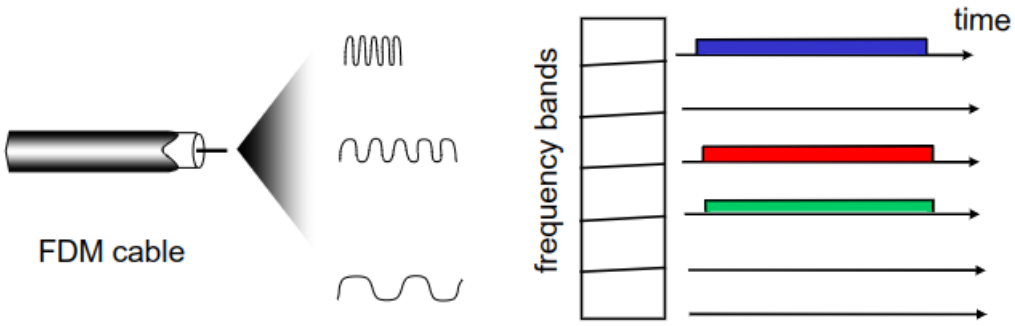
shared wire (e.g., cabled Ethernet) | shared RF (e.g., 802.11 WiFi) | shared RF (satellite) | humans at a cocktail party (shared air, acoustical)

Channel partitioning MAC protocols

- **TDMA** - time division multiple access
  - channel capacity is divided in time frames, and then time frames in time slots
  - access to channel in "**rounds**"
  - each station gets fixed length slot (length = pkt transmition time) in each round
  - unused slots go idle
  - Pros: no collisions. Cons: time wait, may waste capacity.



- **FDMA** - frequency multiple access
  - channel spectrum divided into frequency bands
  - each station assigned fixed frequency band
  - unused transmission time in frequency bands goes idle
  - same TDMA pros and cons

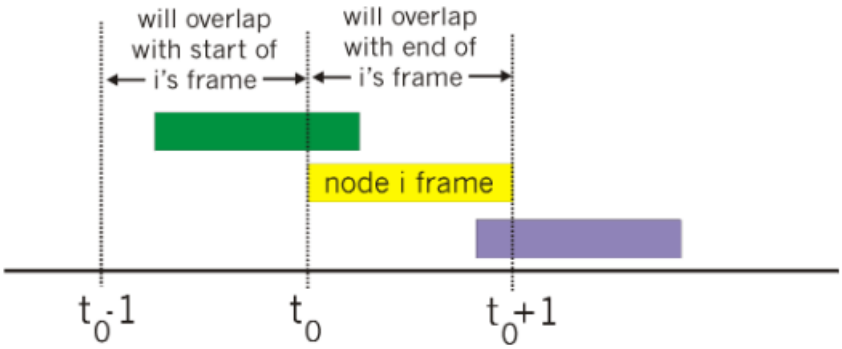


TDMA vs FDMA

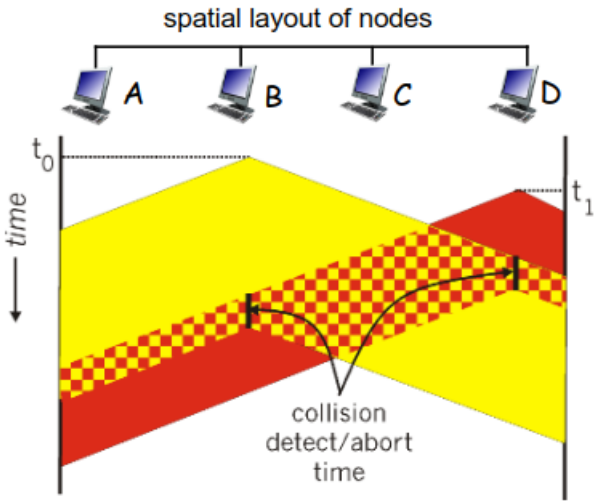
- the choice between TDMA and FDMA depends on the specific application requirements and network conditions
- TDMA may be preferable in scenarios:
  - there are strict requirements for latency and delay, or
  - where the channel is susceptible to interference
- FDMA may be preferable in scenarios:
  - where simplicity and flexibility are more important, or
  - where there are a large number of users that need to be supported

- **Random access protocols**
  - when node has packet to send:
    - transmit at full channel data rate R
    - no a priori coordination among nodes
  - two or more transmitting nodes -> "collision"
  - random access MAC protocol specifies:
    - how to detect collisions
    - how to recover from collisions (e.g. via delayed retransmissions)
  - examples of random access MAC protocols
    - ALOHA; sloted ALOHA (not covered in RC)
    - CSMA, CSMA/CD; CSMA/CA
- **Pure (unslotted) ALOHA**
  - when frame first arrives -> transmit immediately
  - collision probability increases:
    - frame sent at t_0 collides with other frames sent in $[t_0 - I, t_0 + I]$

- max efficiency: 18%



- **CSMA** - listen before transmit
  - if channel sensed idle: transmit entire frame
  - if channel sensed busy: defer transmission
  - (human analogy: don't interrupt others :sociedade:)
  - **Collisions**
    - collisions can still occur: propagation delay means two nodes may not hear each other's transmission
    - collision: entire packet transmission time wasted
- **CSMA/CD**: (collision detection) carrier sensing, deferral as in CSMA
  - collisions detected withing short time
  - colliding transmissions aborted, reducing channel wastage
  - collision detection:
    - easy in wired LANs: measure signal strengths, compare transmitted, recieved signals
    - difficult in wireless LANs: recieved signal strength overwhelmed by local transmission strength; difficult to recieve (sense collisions) when transmitting due to weak recieved signals; can't sense all collisions in any case: hidden terminal, fading.
  - (human analogy: the polite conversationalist)

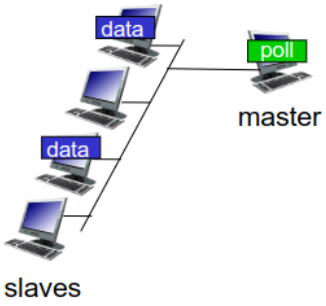


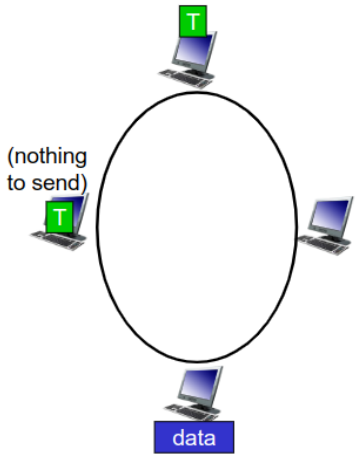
Ethernet CSMA/CD algorithm

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel idle, starts frame transmission. If NIC senses channel busy, waits until channel idle, then transmits.
3. If NIC transmits entire frame without detecting another transmission, NIC is done with frame !
4. If NIC detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, NIC enters binary (exponential) backoff:
   - after m^{th} collision, NIC chooses K at random from $\{0,1,2,\ldots,2^m-1\}$. NIC waits $K \cdot 512$ bit times, returns to Step 2
   - longer backoff interval with more collisions

> Binary backoff: m^1, m^2, ..., m^n (longer interval bakcoff with more collisions)

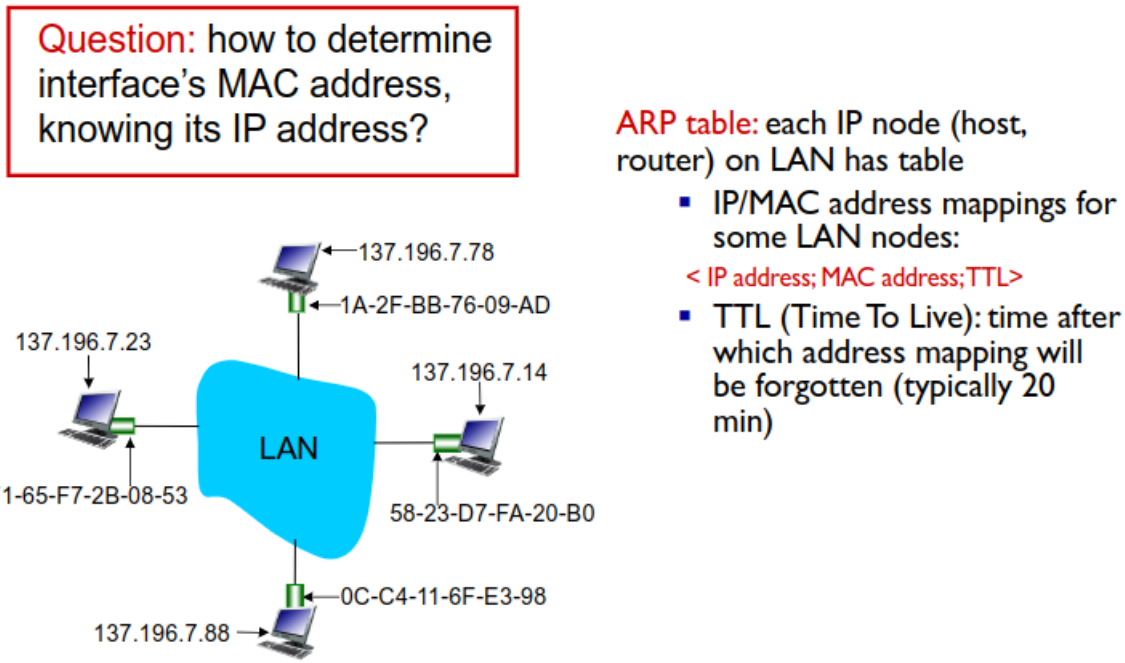
- **"Taking turns"** MAC protocols
- *polling*:
  - master node "invites" slave nodes to transmit in turn
  - typically used with "dumb" slave devices
  - concerns:
    - polling overhead
    - latency
    - single point of failure (master)
- *token passing*: (p.ex. bluetooth)
  - control token passed from one node to next sequentially
  - token message
  - concerns:
    - token overhead
    - latency
    - single point of failure (token)





MAC addresses and ARP

- function: used **locally** to get frame from one interface to another physically-connected interface (same network, in IP-addressing sense)
- 48 bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable (first 4 bytes)



> address mapping can be forgotten (!)

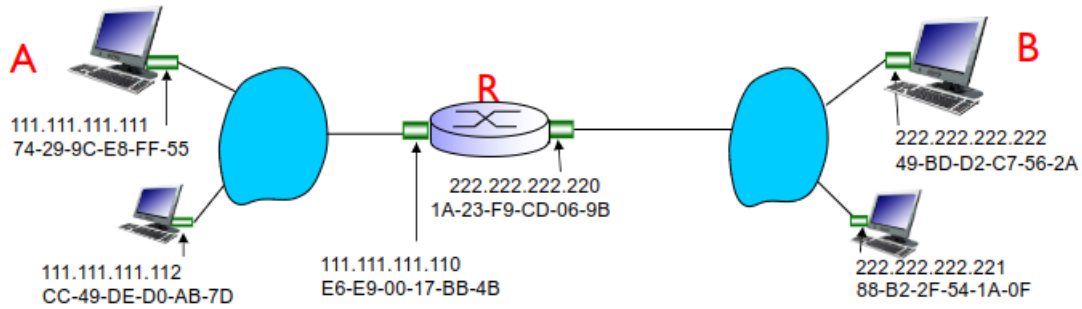
```
root@PC1A:/tmp/pycore.33753/PC1A.conf# arp -a
? (192.168.106.1) at 00:00:00:aa:00:00 [ether] on eth0
root@PC1A:/tmp/pycore.33753/PC1A.conf#
```

- A wants to send datagram to B
  - B's MAC address not in A's ARP table
- A broadcasts ARP query packet, containing B's IP address
  - destination MAC address = FF-FF-FF-FF-FF-FF
  - all nodes on LAN recieve ARP query
- B recieves ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's MAC address (unicast - it's already known from the recieved ARP packet)
- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed
- ARP is "plug-and-play"
  - nodes create their ARP tables without intervantion from net administrator

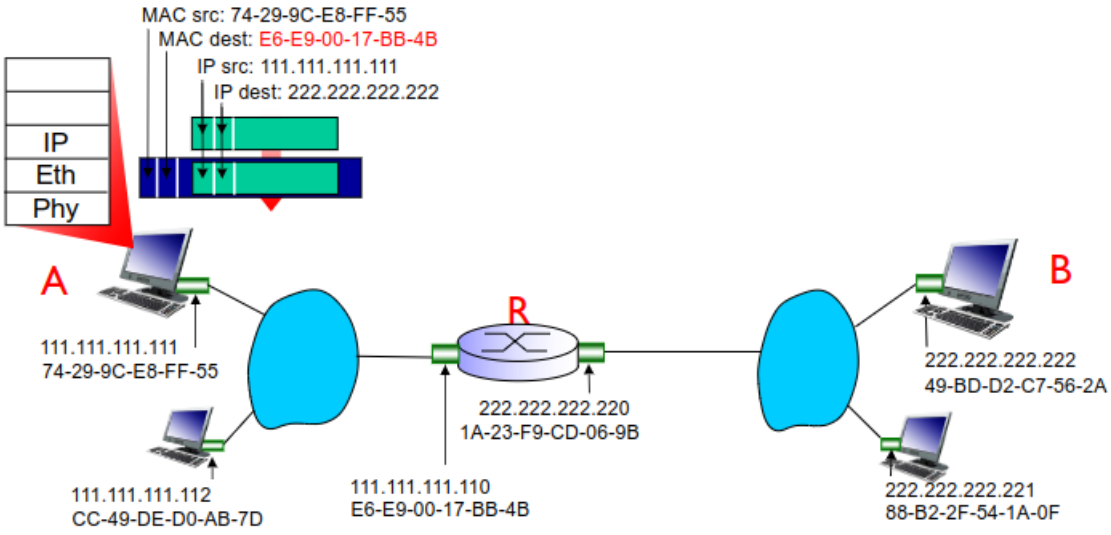
Addressing: routing to another LAN

walkthrough: send datagram from A to B via R

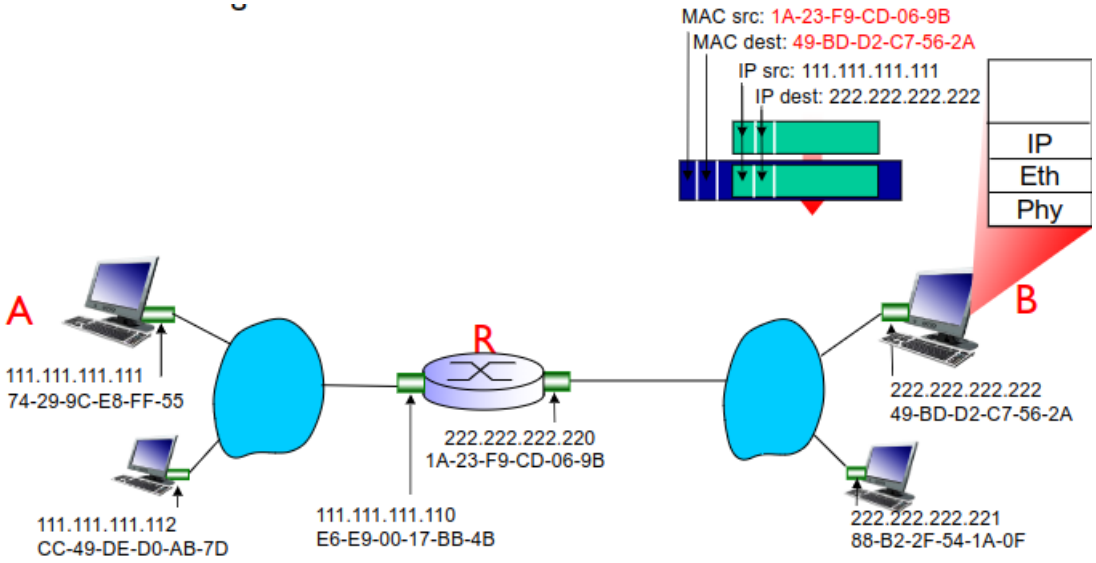
- focus on addressing – at IP (datagram) and MAC layer (frame)
- assume A knows B's IP address
- assume A knows IP address of first hop router, R (how?)
- assume A knows R's MAC address (how?)



- A creates IP datagram with IP source A, destination B
- A creates link-layer frame with R's MAC address as destination address, frame contains A-to-B IP datagram

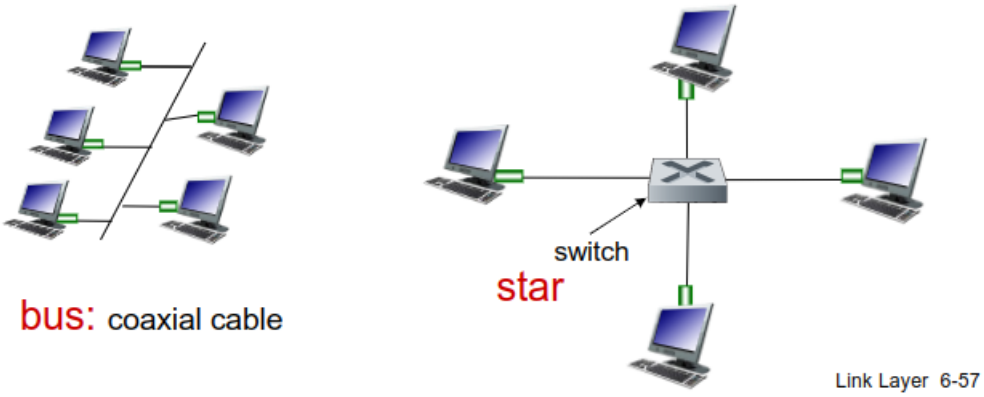


- frame sent from A to R
- frame recieved at R, datagram removed, passed up to IP
- R forwards datagram with IP to source A, destination B
- R creates link-layer frame with B's MAC address as destination address frame contains A-to-B IP datagram

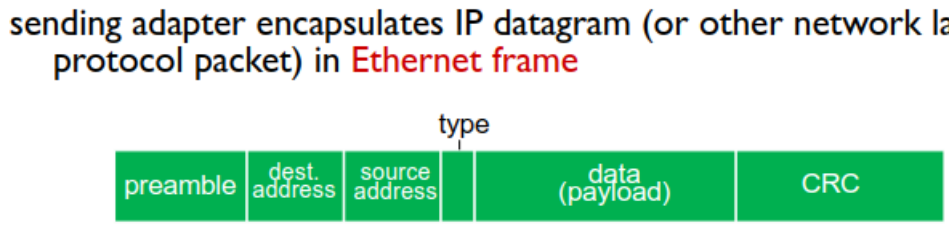


Ethernet

- bus: all nodes in same collision domain (can collide with each other)
- star: prevails today
  - active switch in center
  - each "spoke" runs a separate Ethernet protocol (nodes do not collide with each other)



Ethernet frame structure



- preamble:
  - 7 bytes with pattern `10101010` followed by one byte with pattern `10101011`
  - used to synchonize reciever, sender *clock rates*
- addresses: 6 byte source, destination MAC addresses

    - if adapter recieves frame with matching destination address, or with broadcast address (e.g. ARP packet), it passes data in frame to network layer protocol
    - otherwise, adapter discards frame
- type: indicates higher layer protocol (mostly IP but others possible, e.g., ARP, Novell IPX, AppleTalk)
- CRC: cyclic redundancy check at receiver
    - aka Frame Check Sequence (FCS)
    - error detected: frame is dropped

ConnectionLess: Não orientadas à conexão. Basta que sinta um meio em silencio para enviar as tramas; não há qualquer negociação entre um MAC de origem e um MAC de destino.

Unreliable: Não há confirmações entre NIC's dizendo se a trama chegou bem ou não. Se algo corre mal, as NIC's descartam a trama e depois tentam recuperar de colisão.

Ethernet switch

- link-layer device: takes an active role
    - store, forward Ethernet frames
    - examine incoming frame's MAC address, selectively forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment
- transparent
    - hosts are unaware of presence of switches
- plug-and-play, self-learning
    - switches do not need to be configured

- A: each switch has a switch table, each entry:
    - (MAC address of host, interface to reach host, timestamp)
    - looks like a routing table!

MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

Swtich: frame filtering/forwarding
When frame received at switch:

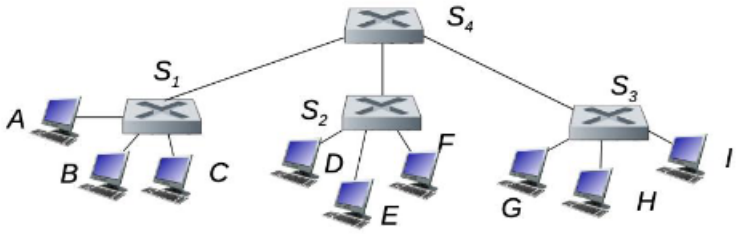
1. record incoming link, MAC address of sending host

2. index switch table using MAC destination address
3. **if** entry found for destination
   then {
   if destination on segment from which frame arrived
   then drop frame
   else forward frame on interface indicated by entry
   }
   else flood; // forward on all interfaces *except* arriving interface

Quando uma trama é recebida:

1. Regista por qual interface é que chegou a trama e o endreço MAC de quem envia;
2. Indexa a tabela usando o MAC destino. E depois ou esse endereço existe ou não na tabela;
3. Se a entrada existir, ele vai primeiro validar se o destino está num segmento diferente daquele onde chega a trama; (situações com topoligias com múltiplos switches, ter a certeza que a trama não é reenviada no link onde está a chegar;
4. Se sim, então descarta a trama;
5. Se não, faz o forward da trama pela interface indicada na entrada (MAC destino);
6. Se a entrada não existir na tabela, faz **flood**. Envia a trama para todas as interfaces à exceção daquela onde chega a trama.

Switches interconectados:



S1, S2, S3 e S4 são redes comutáveis.

Se quisermos enviar uma trama de A para G, aplica-se na mesma o **algoritmo de auto aprendizagem.**

À medida que o tráfego vai fluindo, a tabela de comutação de S4 pode ter numa só entrada o A, o B e o C na porta 1. O que as tabelas têm é uma lista de endereços MAC que são atingidos a partir de uma determinada porta.

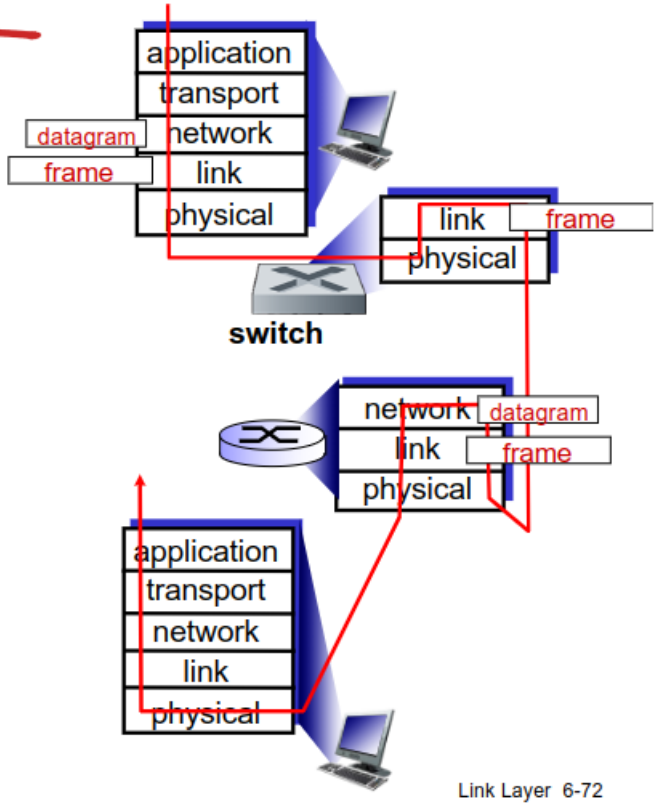
Switches vs. routers

both are store-and-forward:

- routers: network-layer devices (examine network-layer headers)
- switches: link-layer devices (examine link-layer headers)

both have forwarding tables:

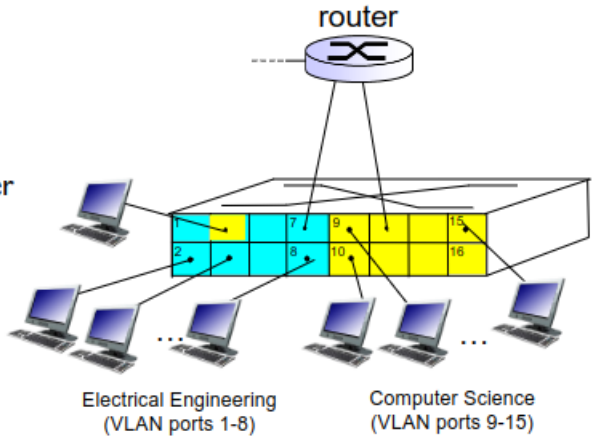
- routers: compute tables using routing algorithms, IP addresses
- switches: learn forwarding table using flooding, learning, MAC addresses



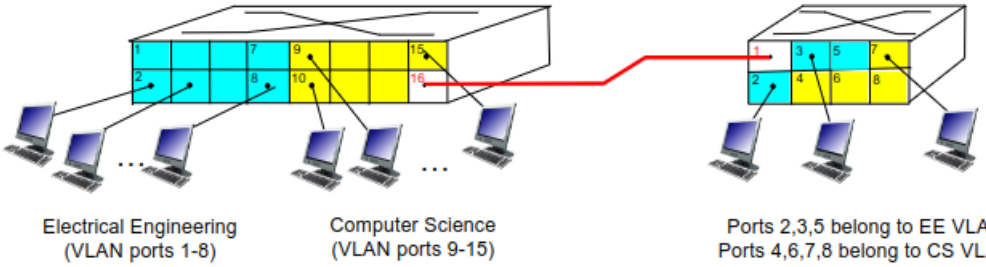
VLANs

port-based VLAN: switch ports grouped (by switch management software) so that **single physical switch operates as multiple virtual switches**.

- **traffic isolation:** frames to/from ports 1-8 can only reach ports 1-8
  - can also define VLAN based on MAC addresses of endpoints, rather than switch port
- **dynamic membership:** ports can be dynamically assigned among VLANs

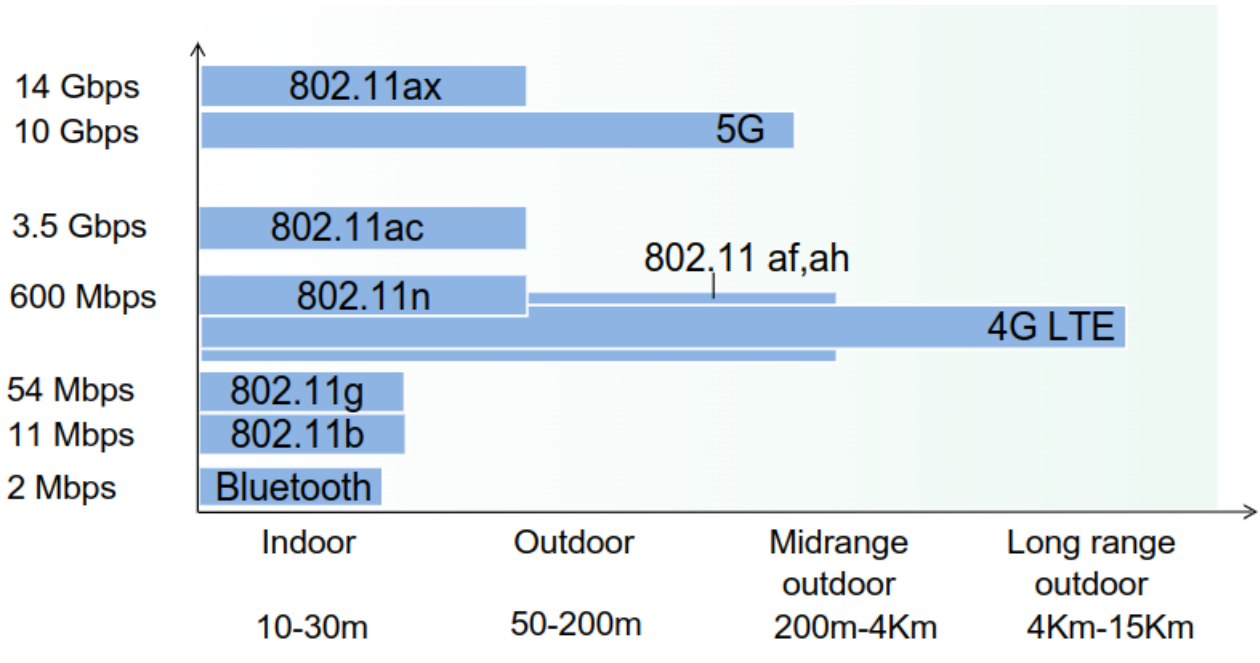


- **forwarding between VLANS:** done via routing (just as with separate switches)
  - in practice vendors sell combined switches plus routers



- **trunk port:** carries frames between VLANS defined over multiple physical switches
  - frames forwarded within VLAN between switches can't be vanilla 802.1 frames (must carry VLAN ID info)
  - 802.1q protocol adds/removed additional header fields for frames forwarded between trunk ports

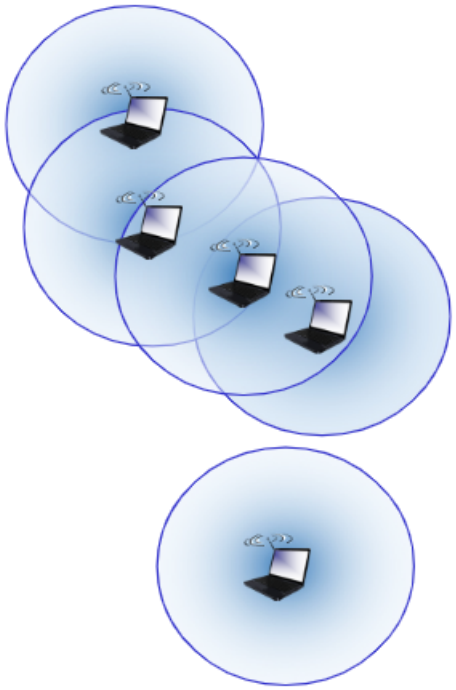
Chapter 7 - Wireless and Mobile Networks



(5 GHz tem menor alcance do que 2.4 GHz)

Ad hoc mode -> single hop

Elements of a wireless network



ad hoc mode

- no base stations
- nodes can only transmit to other nodes within link coverage
- nodes organize themselves into a network: route among themselves

Wireless network taxonomy

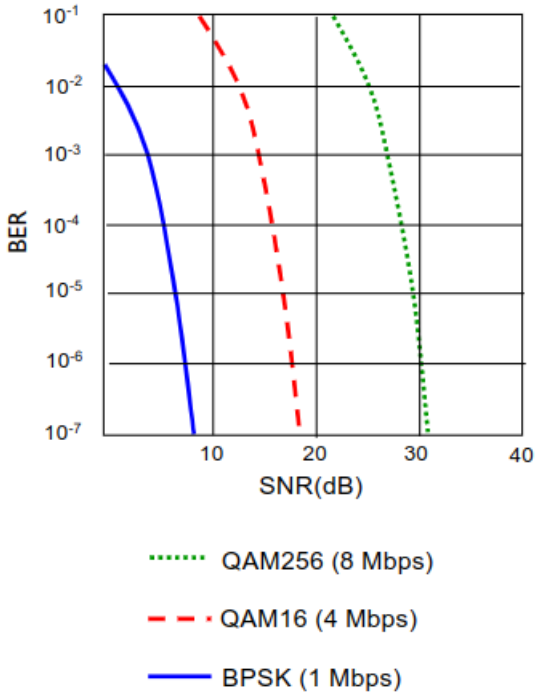
	single hop	multiple hops
infrastructure (e.g., APs)	host connects to base station (WiFi, WiMAX, cellular) which connects to larger Internet	host may have to relay through several wireless nodes to connect to larger Internet: *mesh net*
no infrastructure	no base station, no connection to larger Internet (Bluetooth, ad hoc nets)	no base station, no connection to larger Internet. May have to relay to reach other a given wireless node MANET, VANET

Wired Link Characteristics:

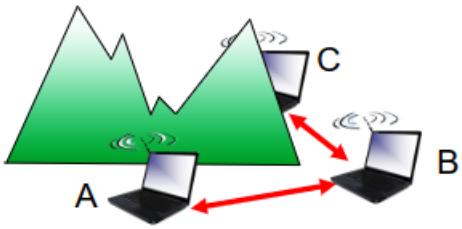
- *decreased signal strength*: radio signal attenuates as it propagatges through matter (path loss)
- *interference from other sources*: standardized wireless network frequencies (e.g. 2.4 GHz) shared by other devices (e.g., Wi-Fi, cellular, motors) interfere as well
- *multipath propagation*: radio signal reflects off objects ground, arriving at destination at slightly different times

SNR em DBm (BER -> Bit Error Ratio)

- SNR: signal-to-noise ratio
  - larger SNR – easier to extract signal from noise (a "good thing")
- *SNR versus BER tradeoffs*
  - *given physical layer:* increase power -> increase SNR->decrease BER
  - *given SNR:* choose physical layer that meets BER requirement, giving highest throughput
    - SNR may change with mobility: dynamically adapt physical layer (modulation technique, rate)

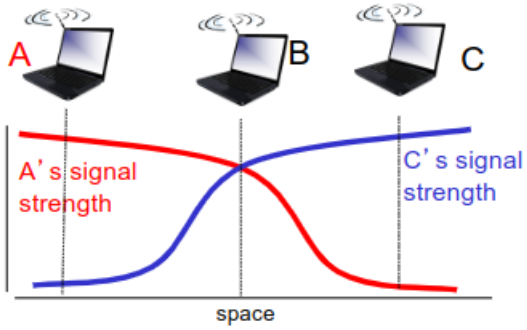


Multiple wireless senders and receivers create additional problems (beyond multiple access):



Hidden terminal problem

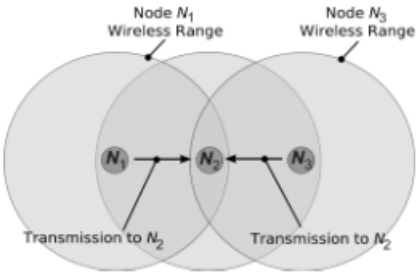
- B, A hear each other
- B, C hear each other
- A, C cannot hear each other means A, C unaware of their interference at B



Signal attenuation:

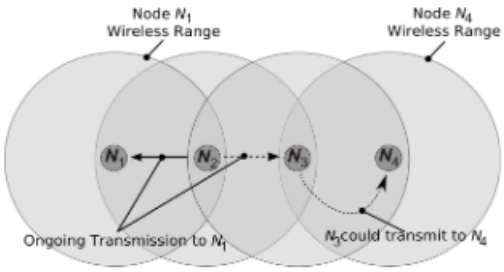
- B, A hear each other
- B, C hear each other
- A, C cannot hear each other interfering at B

Sentir o canal no transmissor não fornece informação acerca do canal no receptor



Hidden node problem / Fading

N1 e N3 não se escutam mutuamente devido a obstáculos ou atenuação: os seus pacotes colidem em N2



Exposed node problem

N1 e N4 poderiam ser receptores simultâneos mas os respectivos emissores N2 e N3 estão em zona de alcance

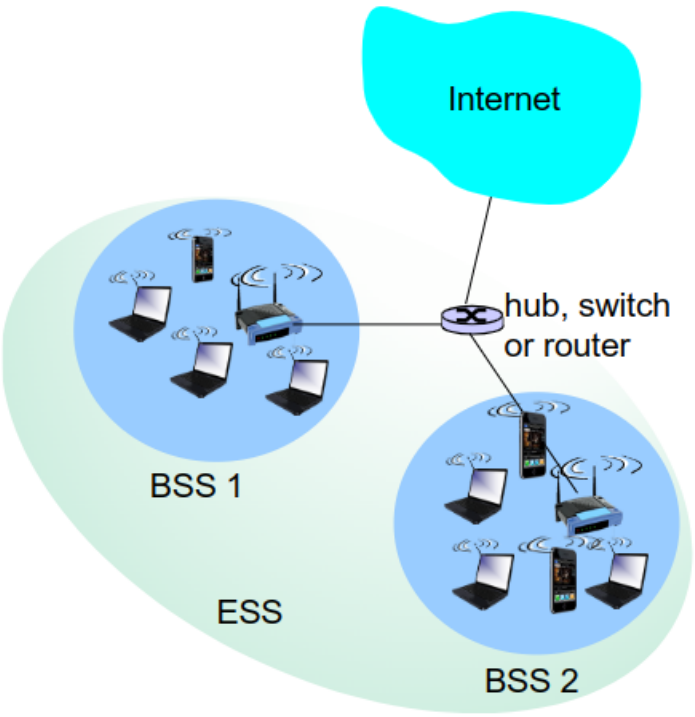
Dois principais problemas

Problema menos grave que o anterior
→reduz a utilização
→menos estudado

Wireless, Mobile Networks 6-16

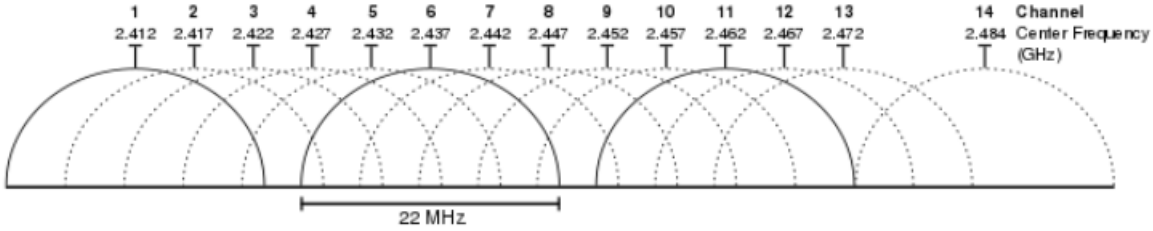
N3 não está em pode de transmitir ou receber tramas.
Se N4 tentar transmitir para N3, falha e faz backoff (tenta mais tarde).

802.11 LAN architecture



- wireless host communicates with base station
  - base station = access point (AP)
- Basic Service Set (BSS) (aka "cell") in infrastructure mode contains:
  - wireless hosts
  - access point (AP)
  - ad hoc mode: hosts only
- Extended Service Set (ESS) includes one or more BSSs

- spectrum divided into channels at different frequencies
  - AP admin chooses frequency for AP
  - interference possible: channel can be same as that chosen by neighboring AP!



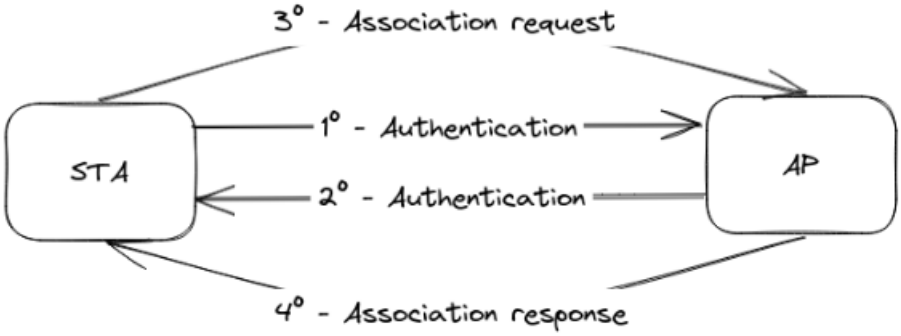
Example: IEEE 802.11b: 2.4GHz-2.48GHz spectrum is divided into 11 or 14 channels (country dependent). IEEE 802.11g/n is divided into 13 channels.

802.11: Channels, association

- arriving host: must associate with an AP
  - scans channels, listening for *beacon frames* containing AP's name (SSID) and MAC address
  - selects AP to associate with
  - then may perform authentication before association
  - then typically run DHCP to get IP address in AP's subnet

A STA envia uma trama de autenticação para o AP (fornecimento de credenciais em caso de shared key authentication - open system não precisa), e de seguida o AP envia uma trama de autenticação para esse STA, aceitando ou rejeitando a autenticação.

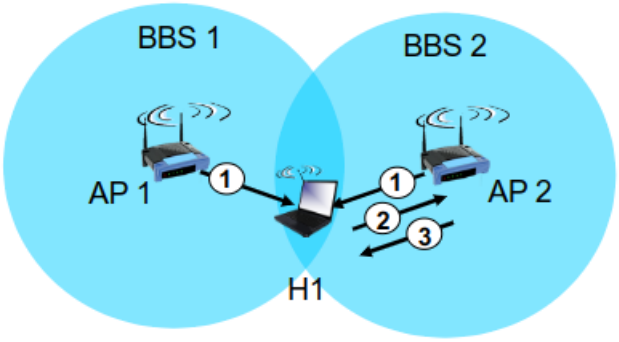
Após a autenticação bem-sucedida, a fase de associação ocorre, o que permite que o STA fique oficialmente associado ao AP e obtenha acesso aos serviços e recursos da rede (ligação lógica entre ambos). Assim, a associação permite que o AP guarde informação sobre cada dispositivo (enviadas na association request) para que os pacotes sejam entregues corretamente. (A association response inclui um association ID, que a STA irá utilizar).

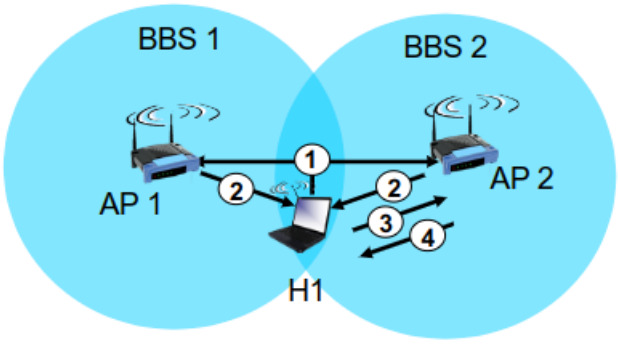


Beacon -> trama que indica as condições de funcionamento do AP.
Beacon frames são transmitidos periodicamento para "anunciar" a presença da rede *wireless* , e para sincronização dos dispositivos na rede.

Passive scanning -> AP espera pelos beacons
Active scanning -> AP faz pedido em broadcast

802.11: passive/active scanning





passive scanning:

(1) Beacon frames sent from APs
(2) Association Request frame sent: H1 to selected AP
(3) Association Response frame sent from selected AP to H1

active scanning:

(1) Probe Request frame broadcast from H1
(2) Probe Response frames sent from APs
(3) Association Request frame sent: H1 to selected AP
(4) Association Response frame sent from selected AP to H1

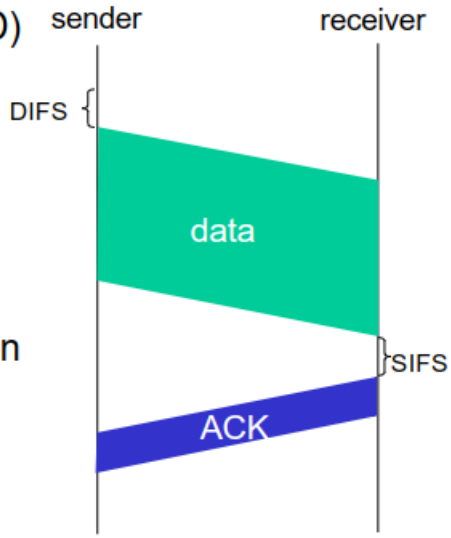
802.11 sender

1 if sense channel idle for **DIFS** then
- sets timer t and transmit entire frame (no CD)
- if no ACK within t, increase random backoff interval
- otherwise transmission successful

2 if sense channel busy then
- start random backoff time
- timer counts down while channel idle
- when the timer expires senses channel again

802.11 receiver

- if frame received OK
  - return ACK after **SIFS** (ACK needed due to hidden terminal problem)



DIFS – DCF (Distributed Coordinated Function) Interframe Space (standard-dep, from 28 to 50 μs)
SIFS – Short Interframe Space (standard-dependent, usually from 10 to 16 μs)

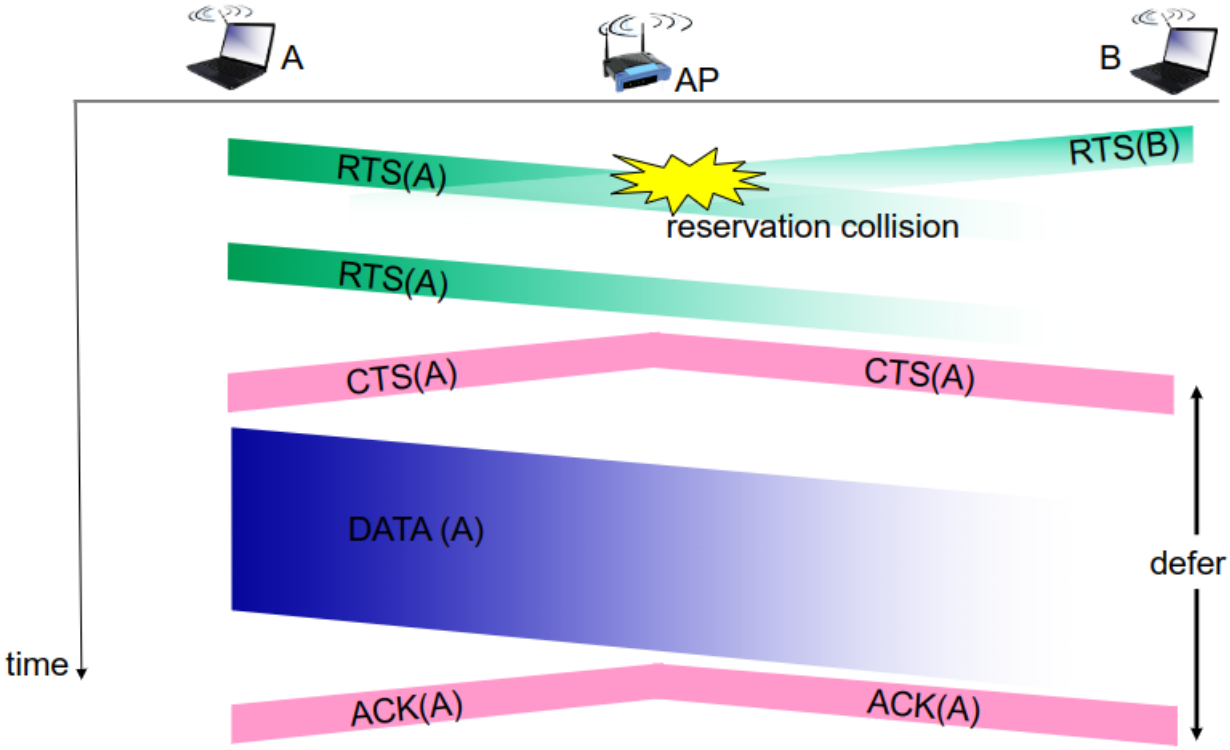
> ACK -> acknowledgement
> SIFS < DIFS => **garantia que possa ouvir o meio**

Avoiding collisions

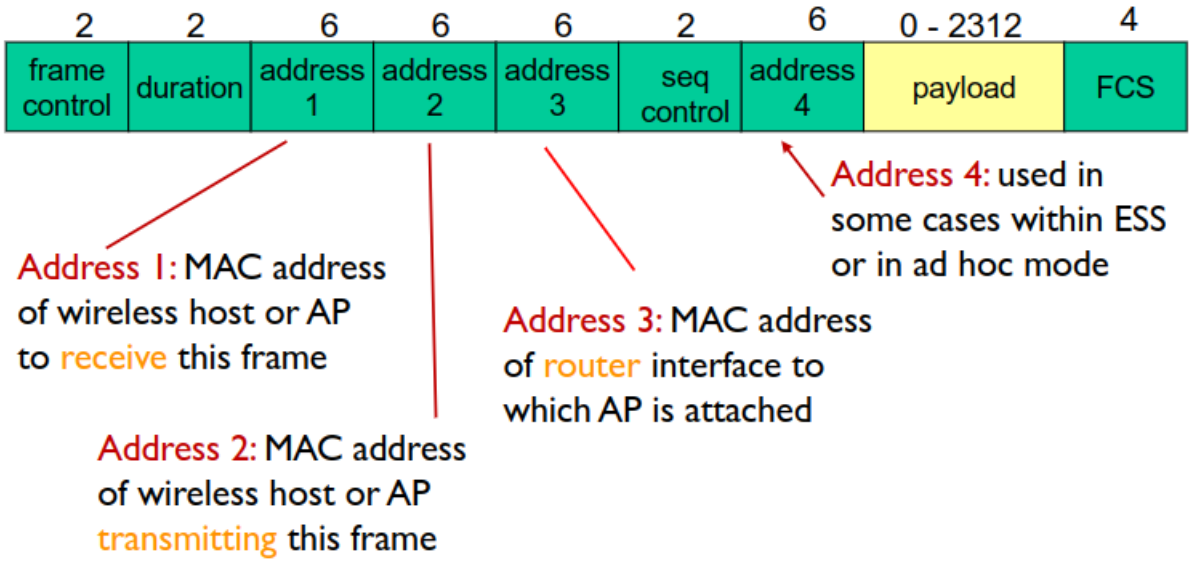
ideia: allow sender to "reserve" channel rather than random access of data frames: avoid collisions of long data frames

- sender first transmits small request-to-send (RTS) packets to base station (BS) using CSMA
  - RTSs may still collide with each other (but they're short)
- BS broadcasts clear-to-send CTS in response to RTS
- CTS heard by all nodes
  - sender transmits data frame
  - other stations defer transmissions

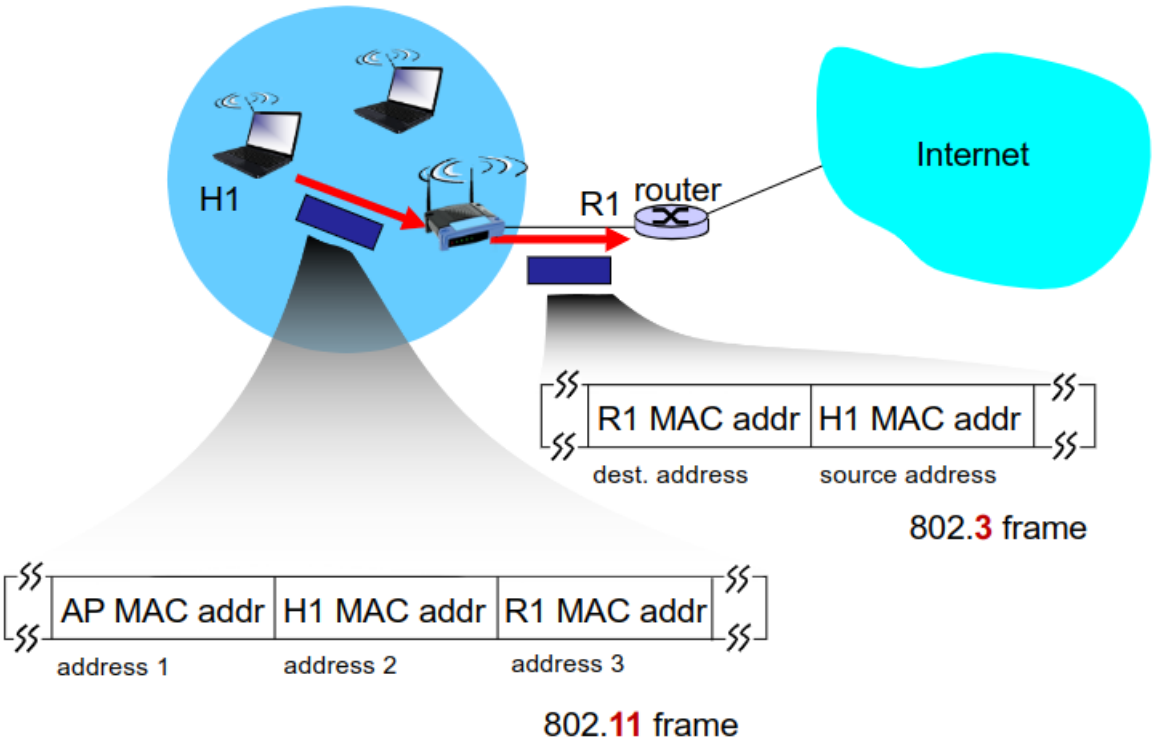
avoid data frame collisions completely using small reservation packets!

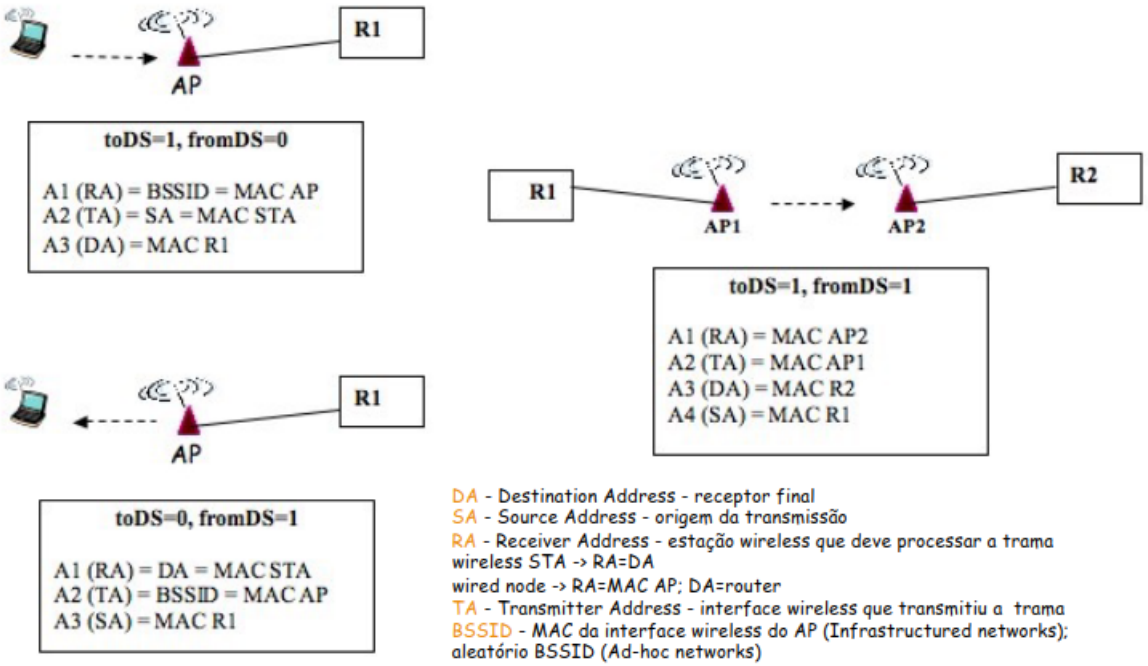


802.11 frame: addressing



802.11 frame: addressing





Portanto, "os 4 endereços são necessários" (i.e. podem ser todos únicos) no caso em que toDS == 1 e fromDS == 1.

Summary:

- The addresses semantics depends on type of frame and directionality
- addr1 – MAC address the receiver (varies); always read
- addr2 – MAC address of the transmitter
- addr3 – MAC address of the router (if to/from DS)

toDS	fromDS	addr1	addr2	addr3	addr4	obs.
0	0	DA	SA	BSSID	-	ad hoc
0	1	DA	BSSID	SA	-	do AP
1	0	BSSID	SA	DA	-	para AP
1	1	RA	TA	DA	SA	dentro DS

Frame types and subtypes:

Type 00 – Management frames
e.g. Beacon, Association request, Probe request, etc.

Type 01 – Control frames
e.g. RTS, CTS, ACK, etc.

Type 10 – Data frames
e.g. Data

> Note:
> When BER becomes too high, switch to lower transmission rate with lower BER