

# Telecommunication Networks

Sándor Laki

ELTE-Ericsson Communication Networks Laboratory

*ELTE FI – Department Of Information Systems*

[lakis@elte.hu](mailto:lakis@elte.hu)

<http://lakis.web.elte.hu>



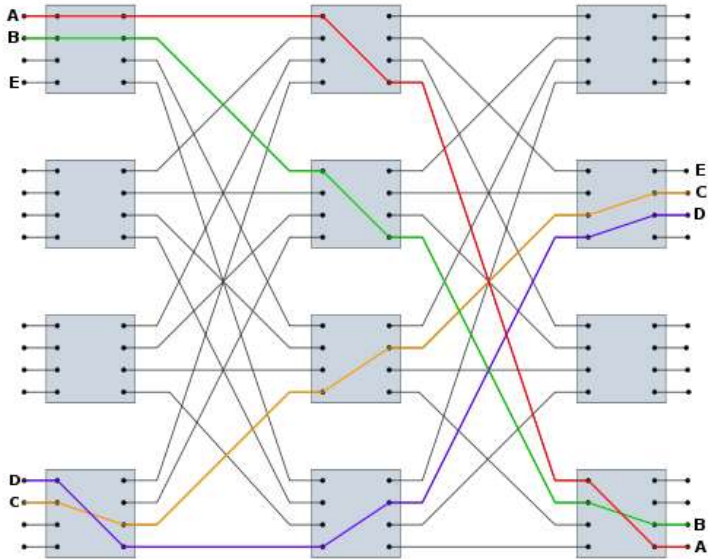
Eötvös Loránd  
University

*Based on the slides of Laurent Vanbever.  
Further inspiration: Scott Shenker & Jennifer Rexford & Phillipa Gill*

# Megvalósítások

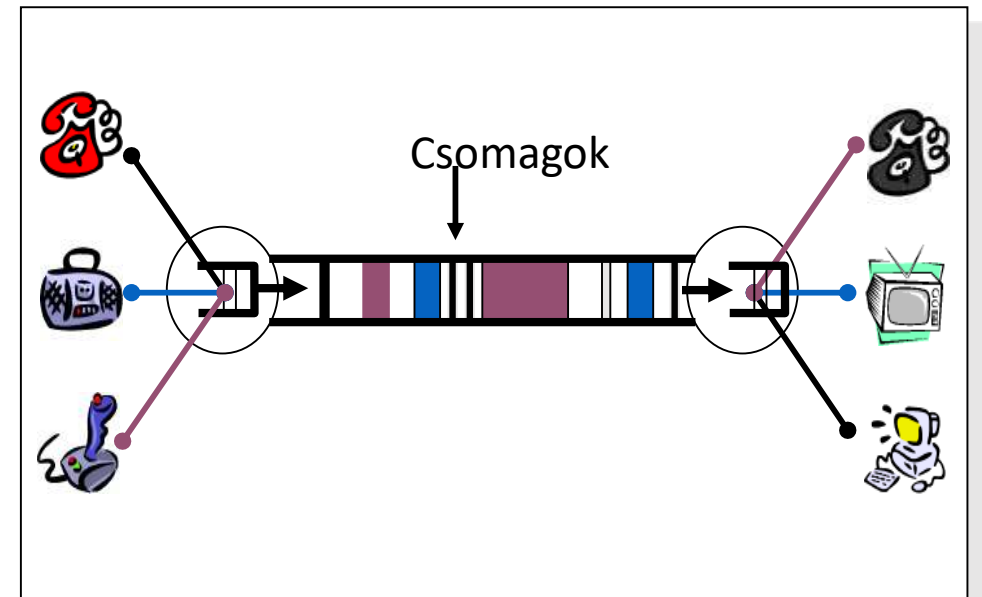
## Előre foglalással

Áramkörkapcsolt hálózat  
Pl. vezetékes telefon



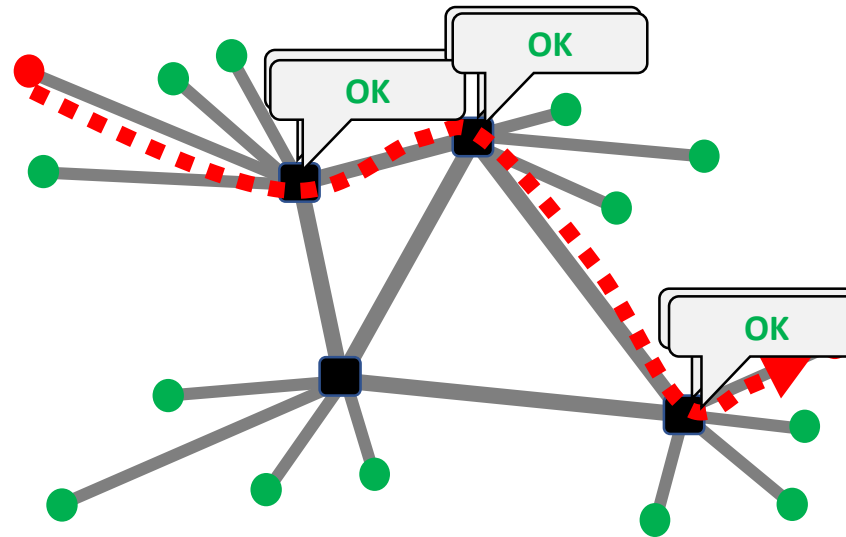
## Igény szerinti

Csomagkapcsolt hálózat  
Pl. Internet



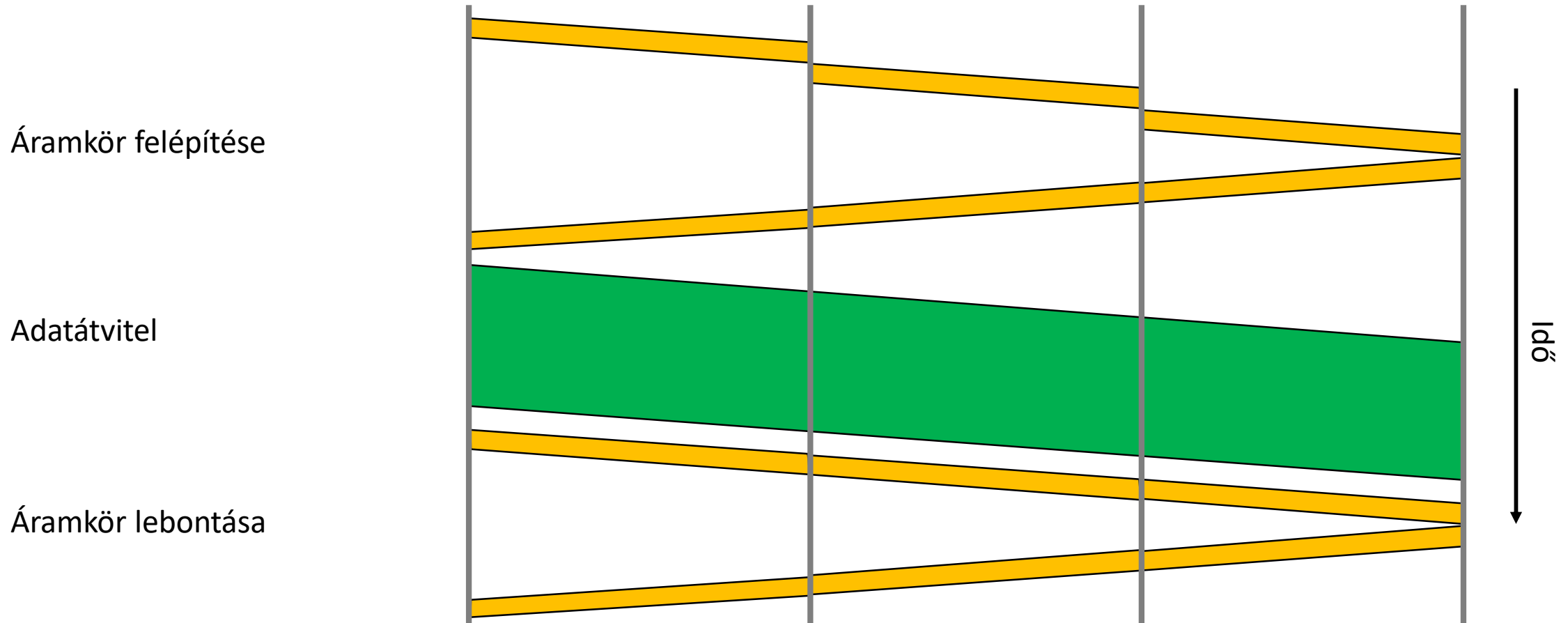
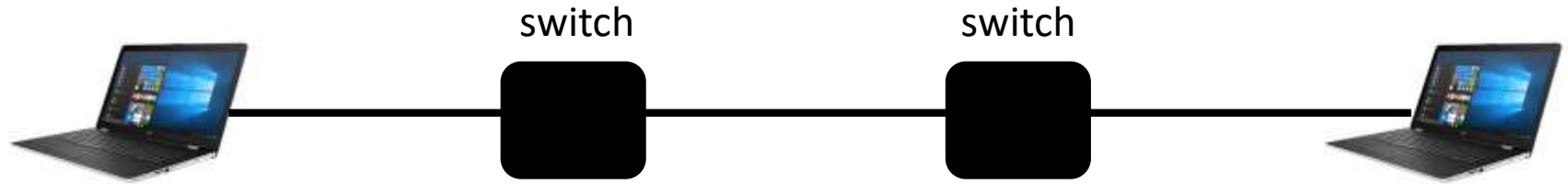
# Az áramkörkapcsolt hálózat alapja a **Resource Reservation Protocol**

Erőforrás-foglaló protokoll



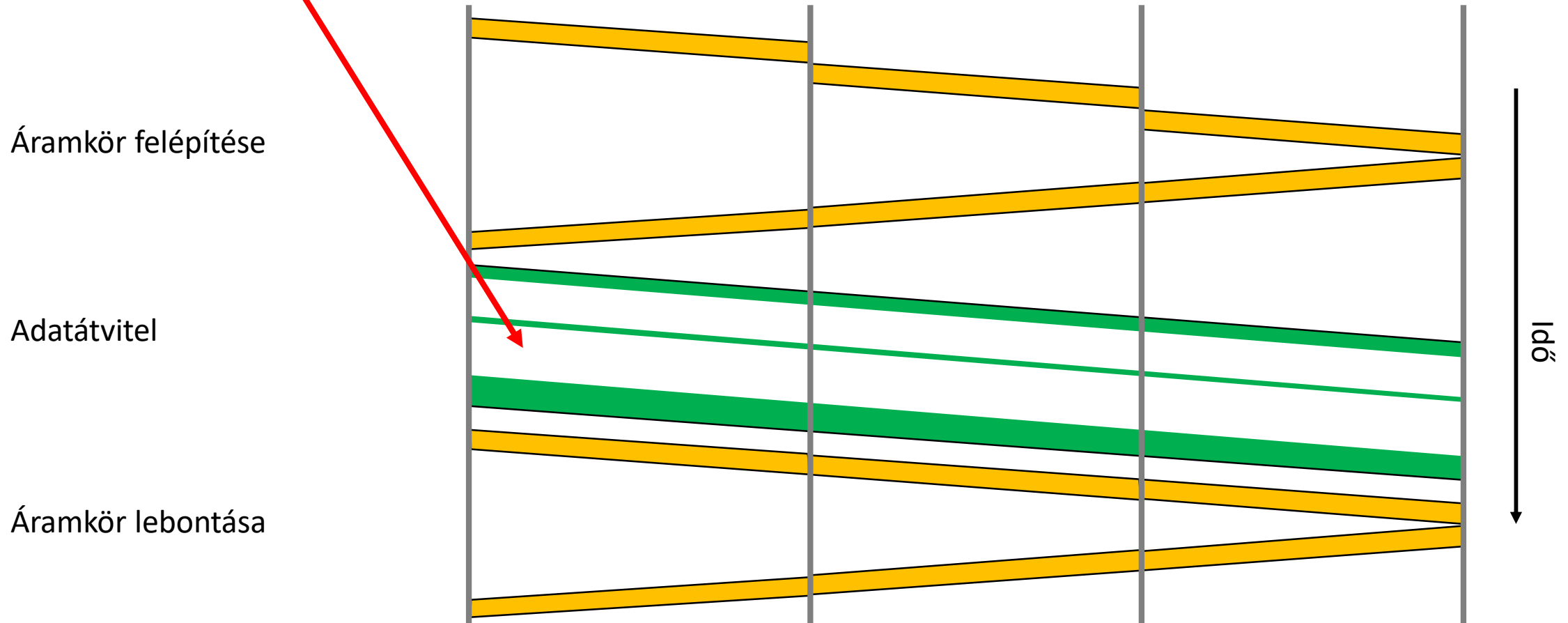
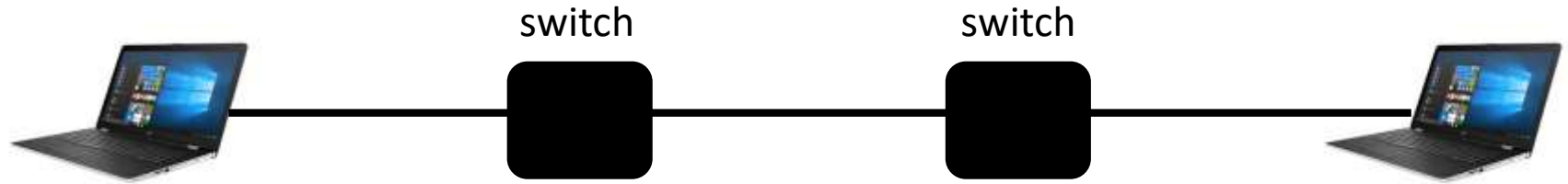
1. A **forrás foglalási kérést küld** 10 Mbps igényről a célállomásnak
2. A switchek **kialakítják az „áramkört”**
3. A forrás megkezdi az **adatküldést**
4. A forrás **áramkör-lebontó üzenetet** küld a cél felé (teardown)

# Adatátvitel áramkörkapcsolt hálózaton

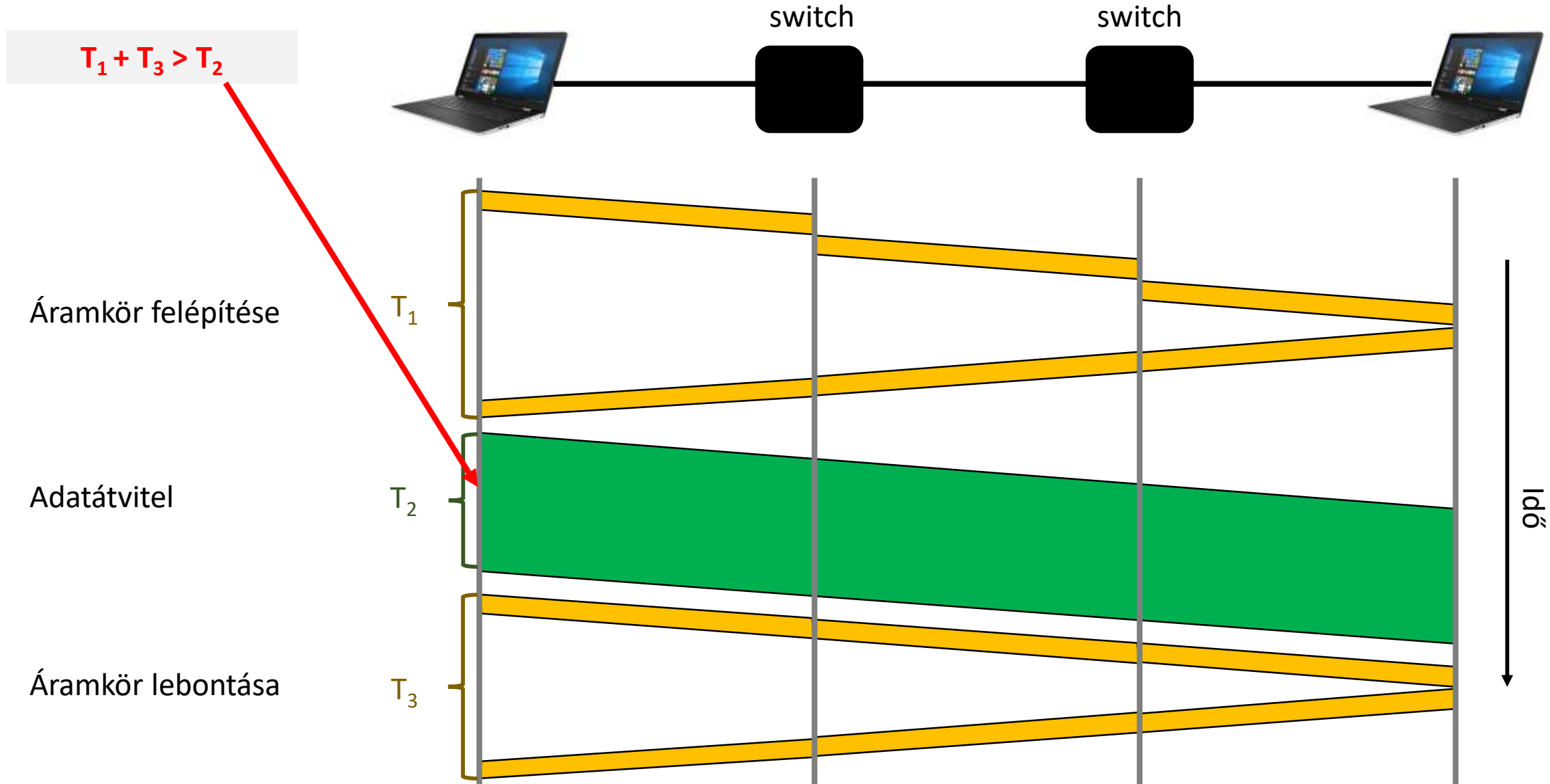


# Löketszerű forgalom - Rossz teljesítmény

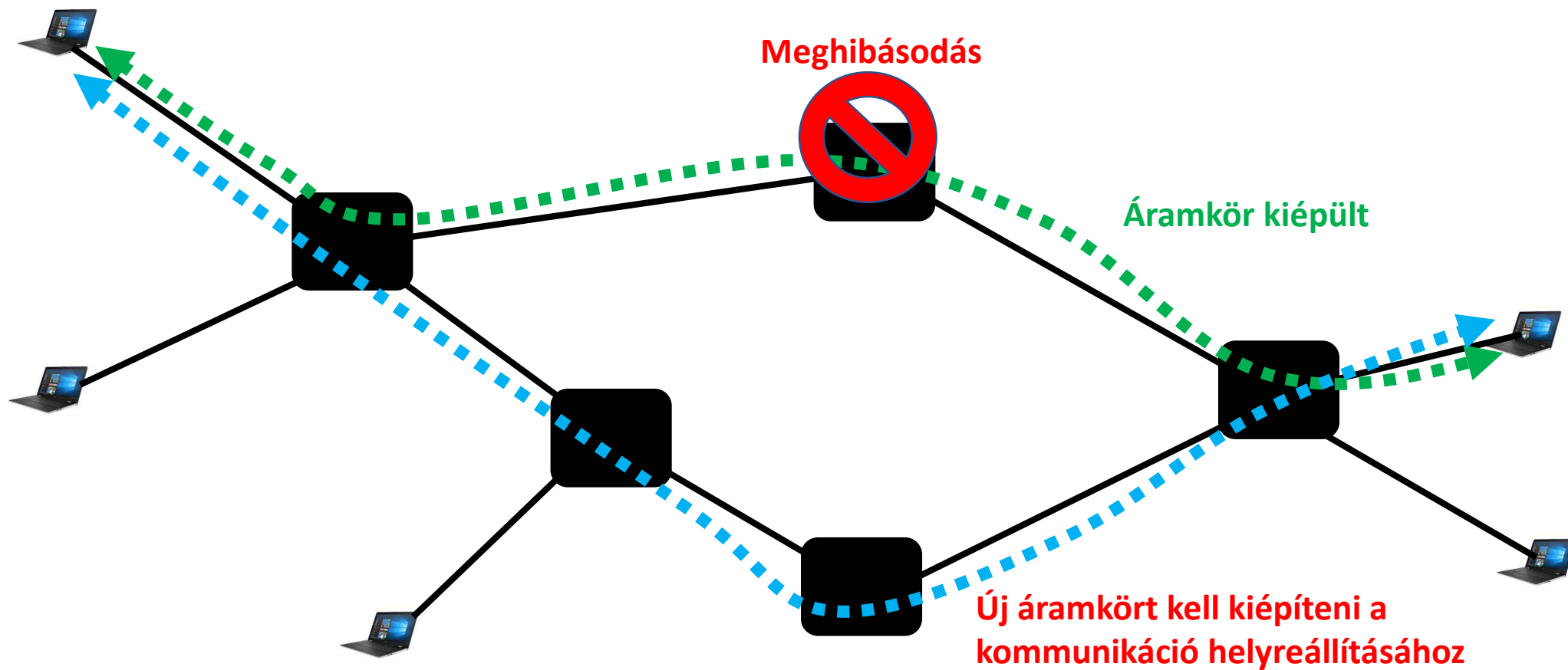
A löketszerű forgalom miatt az áramkör az idő nagy részében kihasználatlan.



# Rövid üzenetváltás – Rossz teljesítmény



# További probléma a **meghibásodott** switch kikerülése (reroute)



# Érvek/Ellenérvek

## Előnyök

Kiszámítható teljesítmény

Egyszerű és gyors kapcsolás  
Miután kiépült az áramkör

## Hátrányok

Alacsony hatékonyság  
Löketszerű forgalom  
Rövid folyamatok

Bonyolult áramkör felépítés/lebontás  
Megnövekedett késleltetés

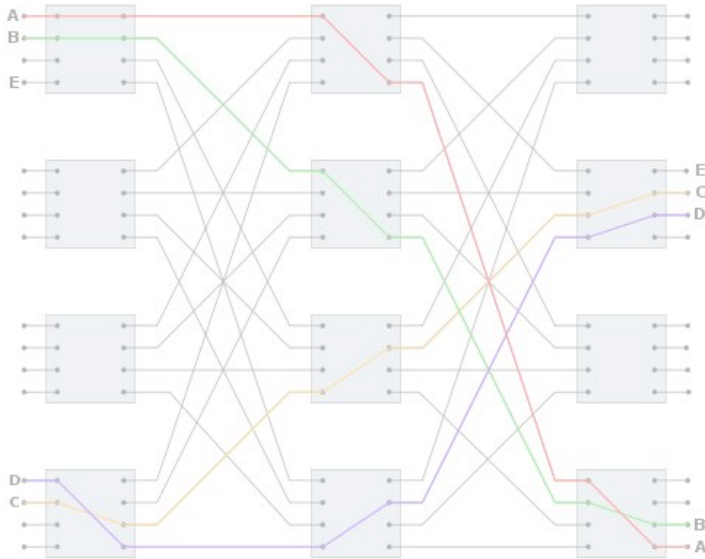
Hiba esetén új áramkör szükséges



# Megvalósítások

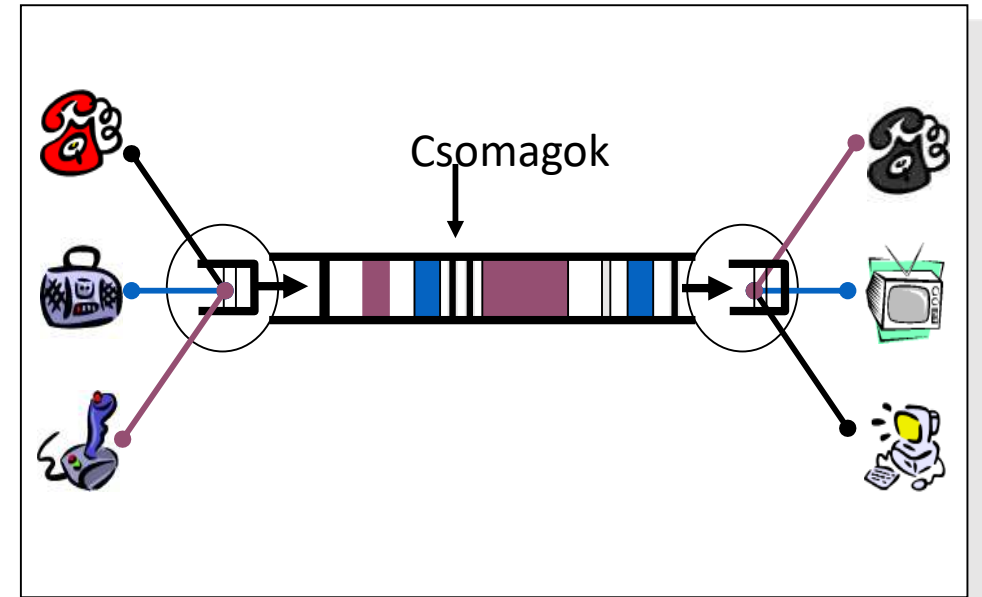
Előre foglalással

Áramkörkapcsolt hálózat  
Pl. vezetékes telefon



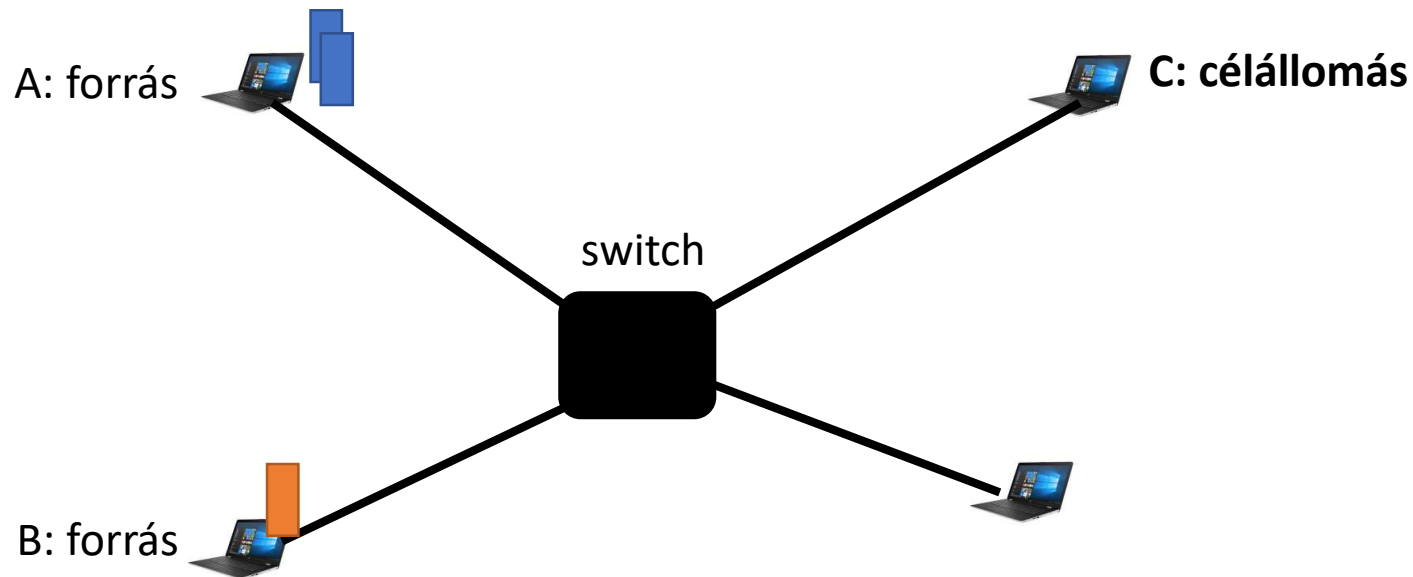
Igény szerinti

Csomagkapcsolt hálózat  
Pl. Internet



# Csomagkapcsolt hálózatok

**Az adatátvitel egyedi csomagokban történik.**



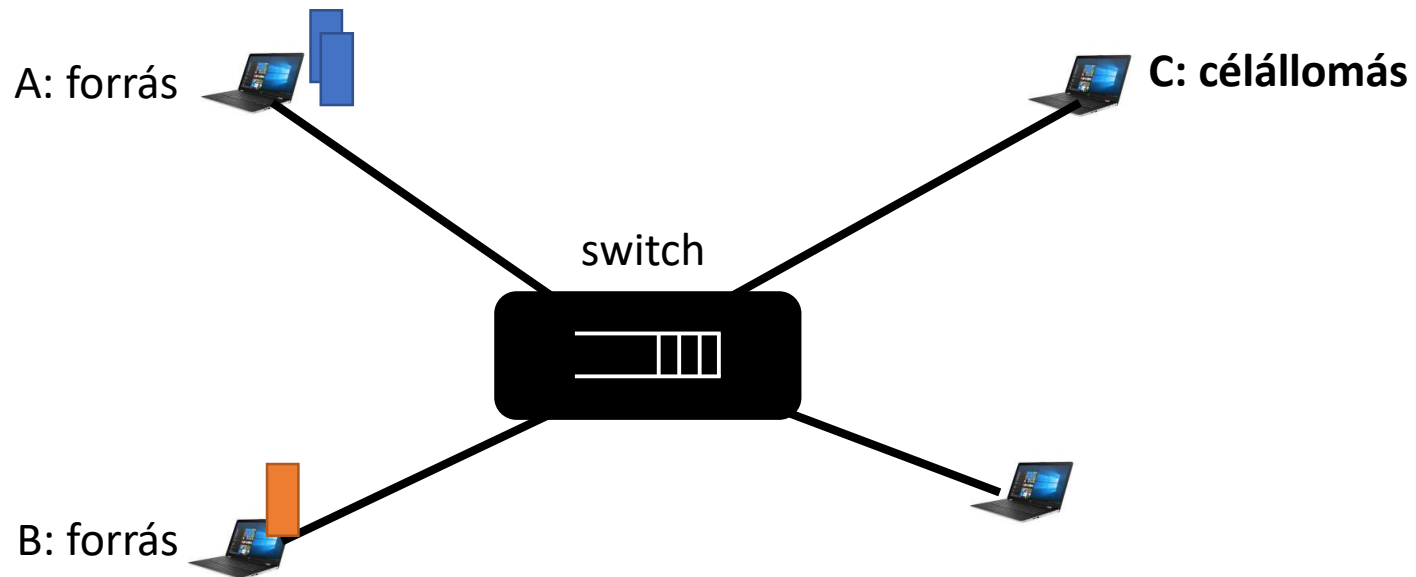
**Minden csomag tartalmazza a cél címét/azonosítóját (most C).**

**Nincs globális koordináció, azaz a csomagok zavarhatják egymást.  
(Id. egyszerre érkeznek be a switchhez)**

**Pufferelés szükséges a löketek kezeléséhez.**

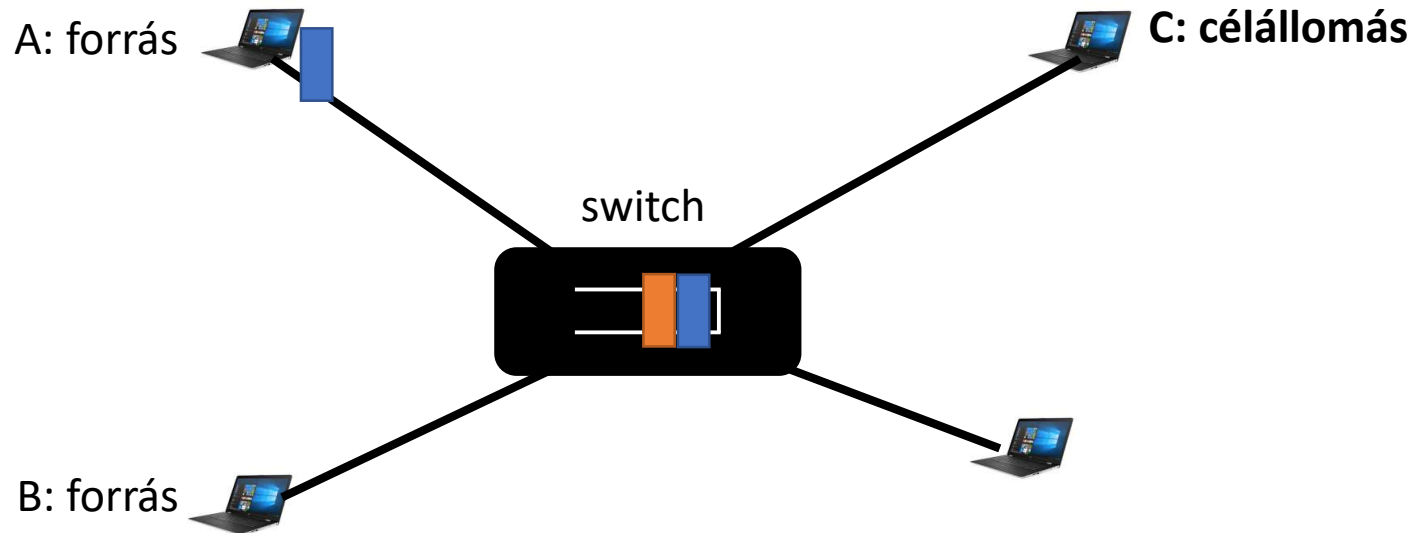
# Csomagkapcsolt hálózatok

## Pufferelés az átmeneti túlterhelések kezeléséhez

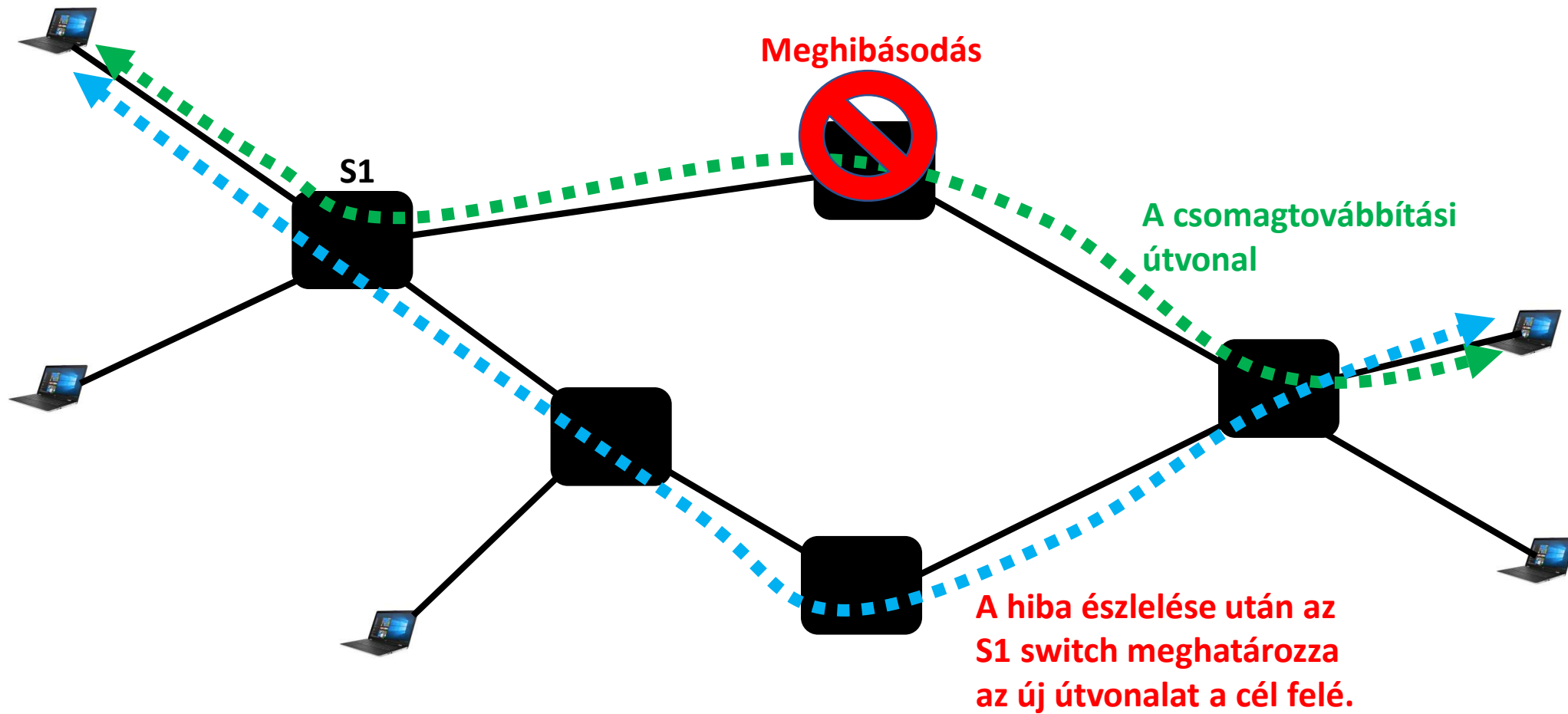


# Csomagkapcsolt hálózatok

## Pufferelés az átmeneti túlterhelések kezeléséhez



# Hiba tolerancia



# Érvek ellenérvek

## Előnyök

Hatékony erőforrásgazdálkodás

Egyszerű megvalósítás

Hibatolerancia

## Hátrányok

Kiszámíthatatlan teljesítmény

Szükséges puffer-kezelés és torlódás-vezérlés

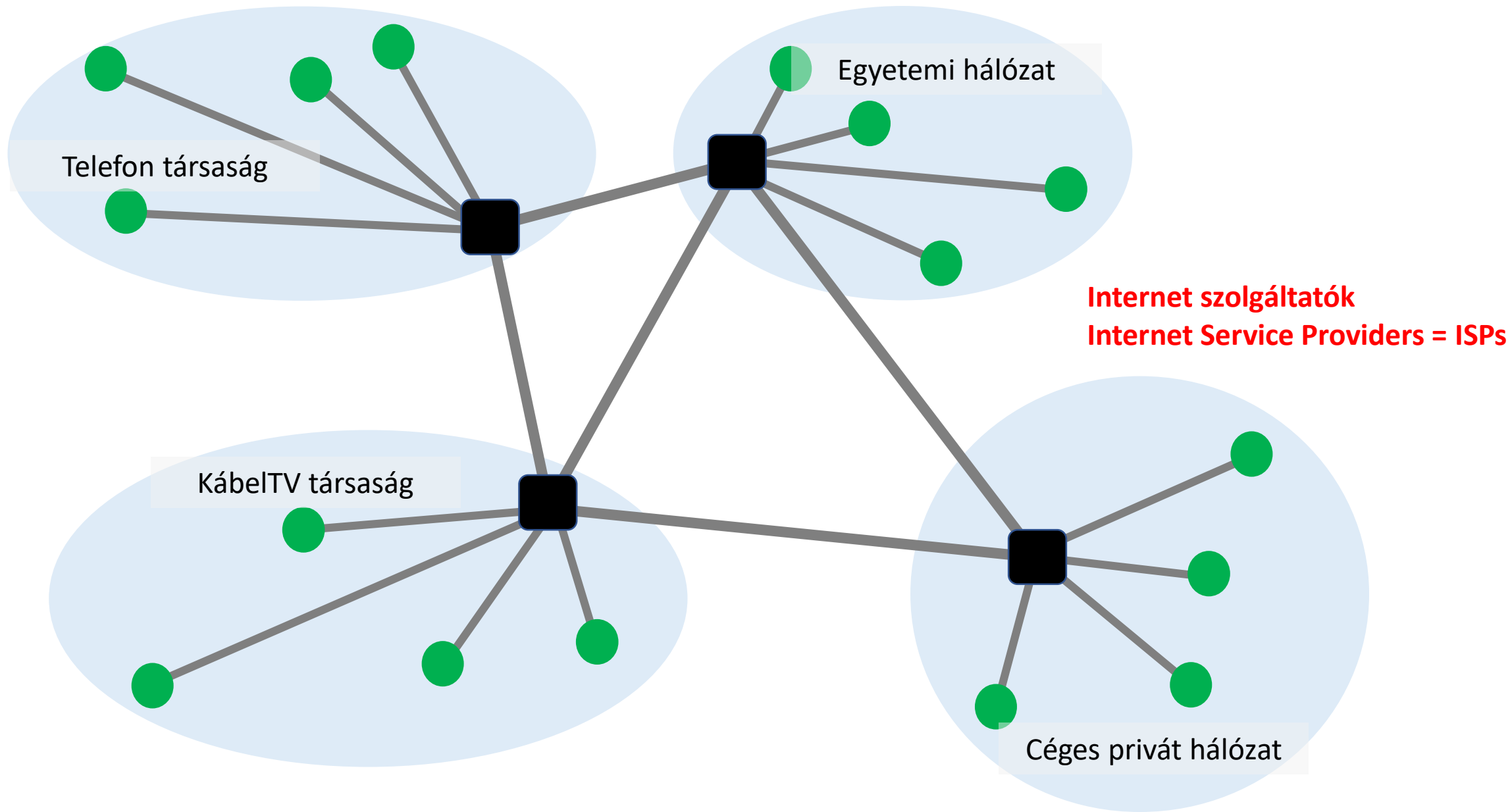
# Az Internet csomagkapcsolt

Rugalmasság és hatékonyság

# Áttekintés

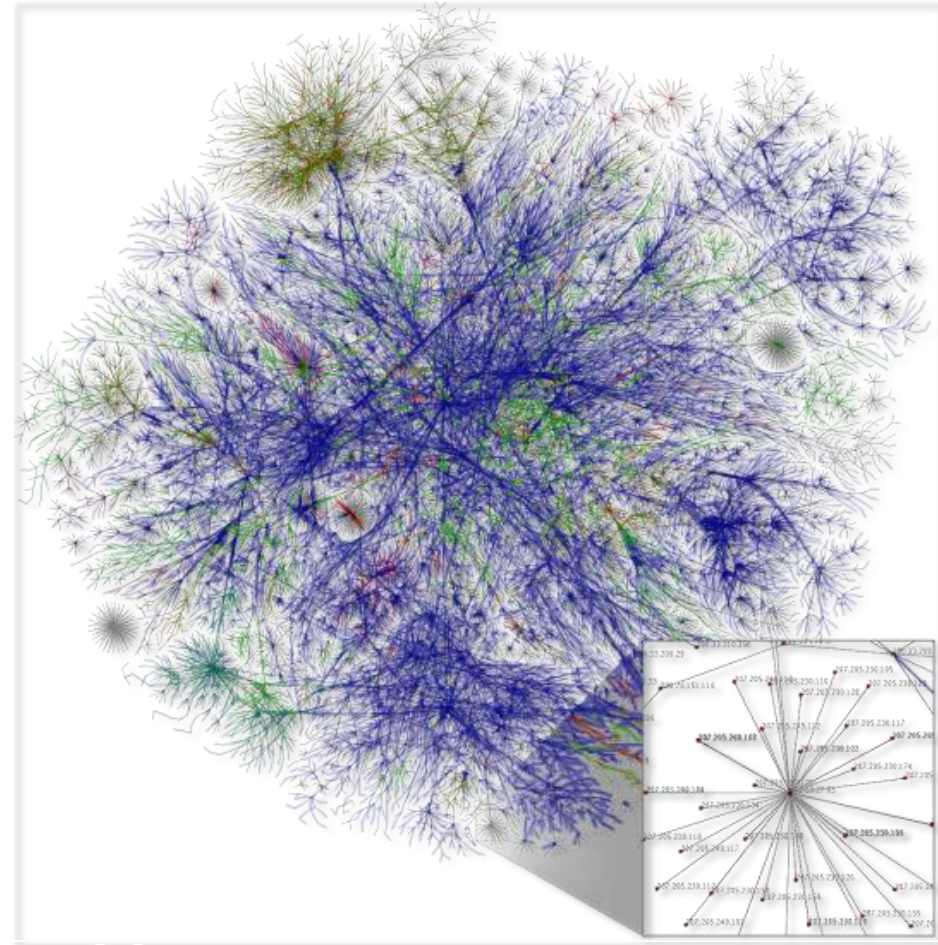
Hogyan szervezzük a hálózatot?



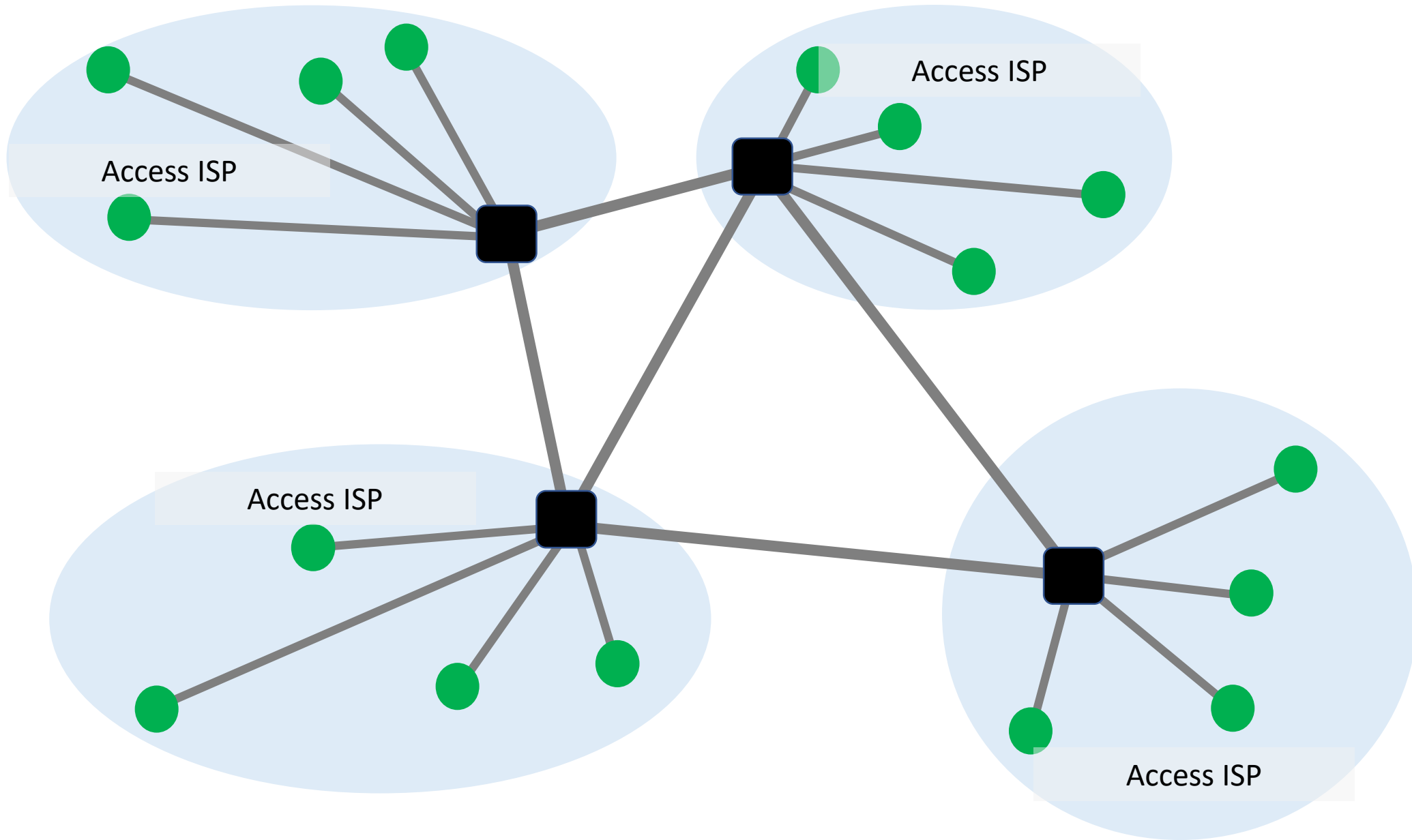


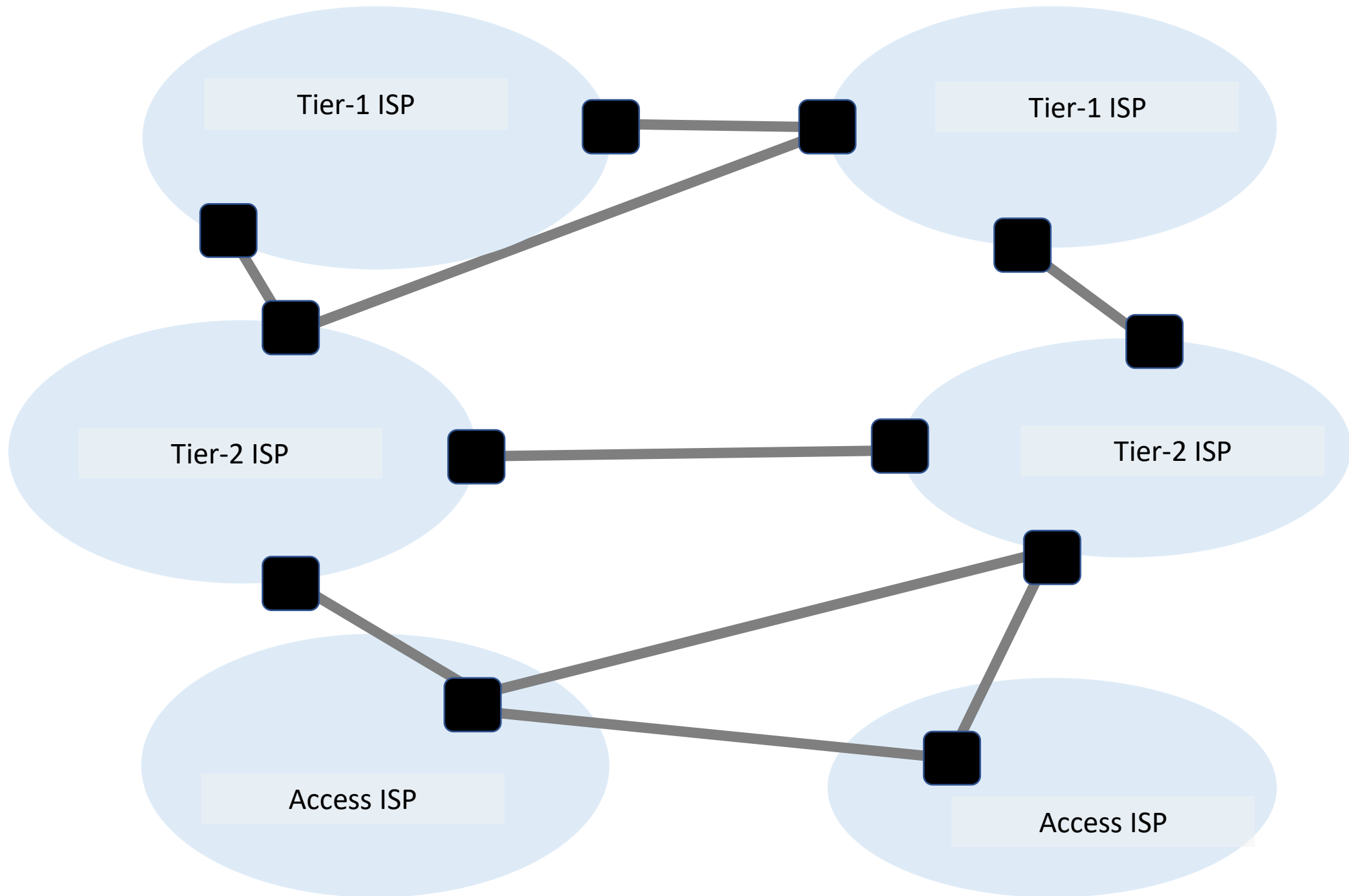
# Mi az internet?

- Hálózatok hálózata
- A világra kiterjedő nyitott WAN
- Jellemzői
  - rendszerfüggetlenség;
  - nincs központi felügyelet;
  - építőelemei a LAN-ok;
  - globális;
  - olyan szolgáltatásokat nyújt, mint a **World Wide Web**, e-mail vagy fájlátvitel.



# ISP – Internet szolgáltató





# Az Internet hierarchikus struktúrája

szolgáltató-vásárló (provider-customer) viszonyok

## Tier-1

nemzetközi

Nincs szolgáltatója

## Tier-2

nemzeti

Tier-3 szolgáltatóknak nyújt átjárást

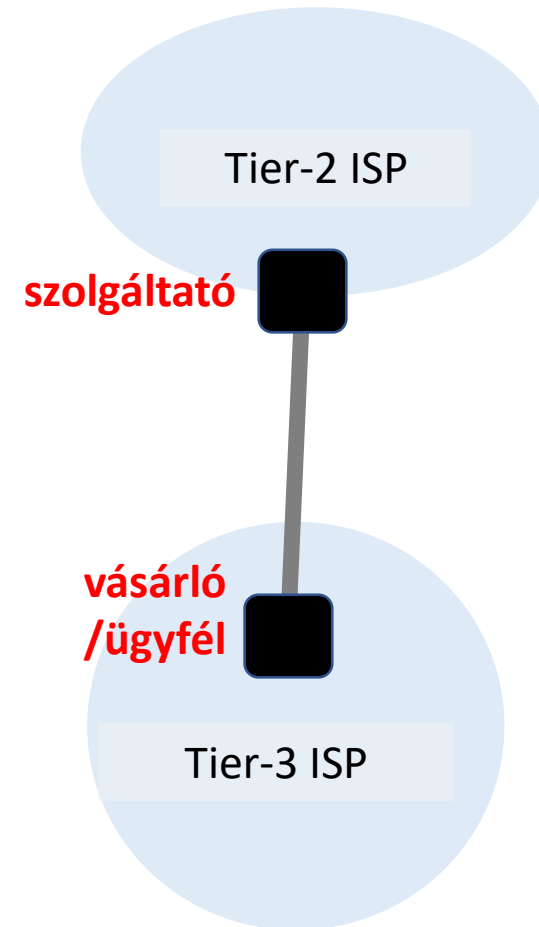
Legalább egy szolgáltatója van

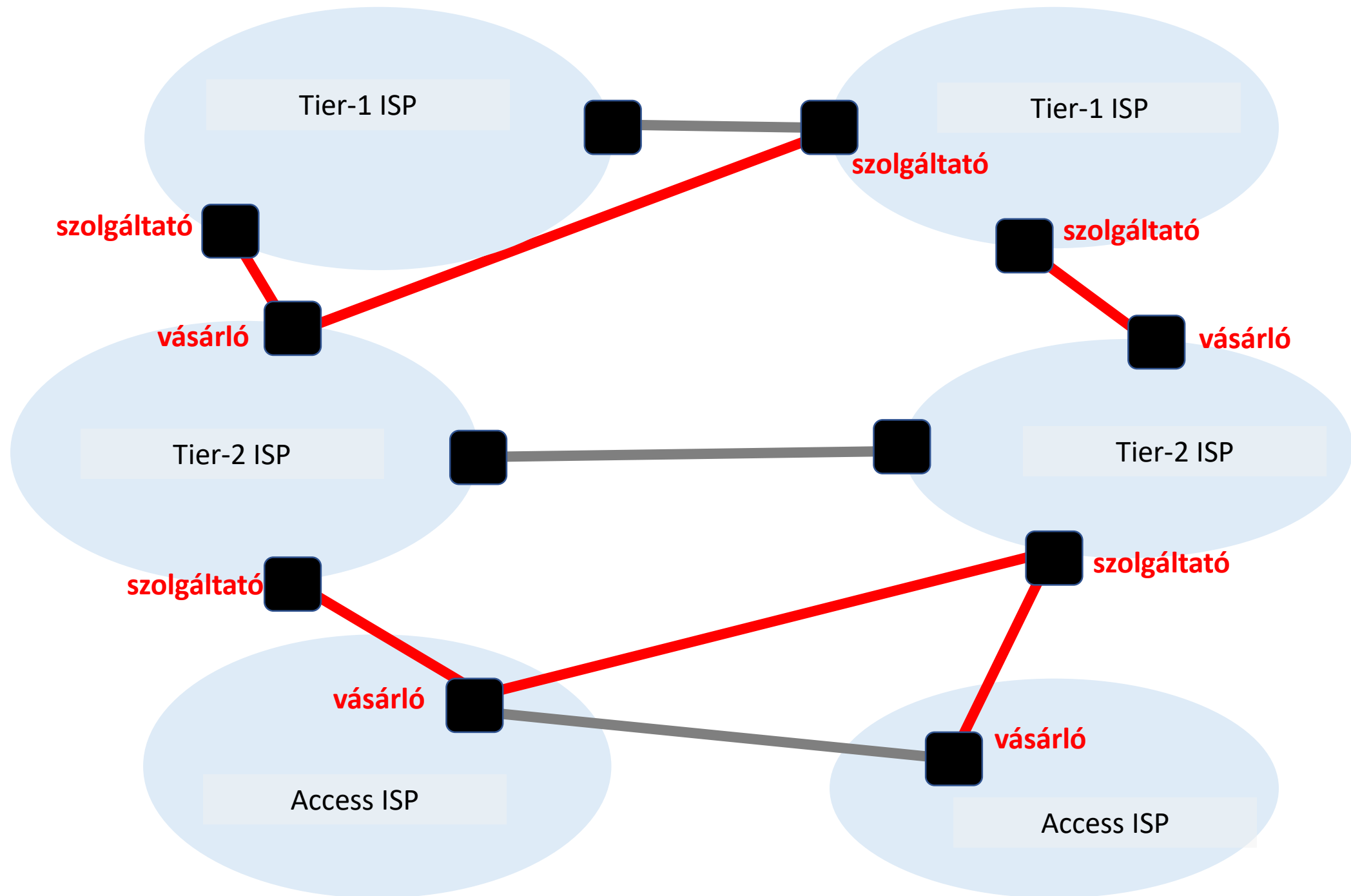
## Tier-3

helyi

Nem nyújt átjárást más szolgáltatóknak

Legalább egy szolgáltatója van





# Hálózatok eloszlása a Tier-ekben

**~50.000 hálózat összesen**

## **Tier-1**

nemzetközi

Nincs szolgáltatója

**pár tucat**

## **Tier-2**

nemzeti

Tier-3 szolgáltatóknak nyújt átjárást

Legalább egy szolgáltatója van

**több ezer**

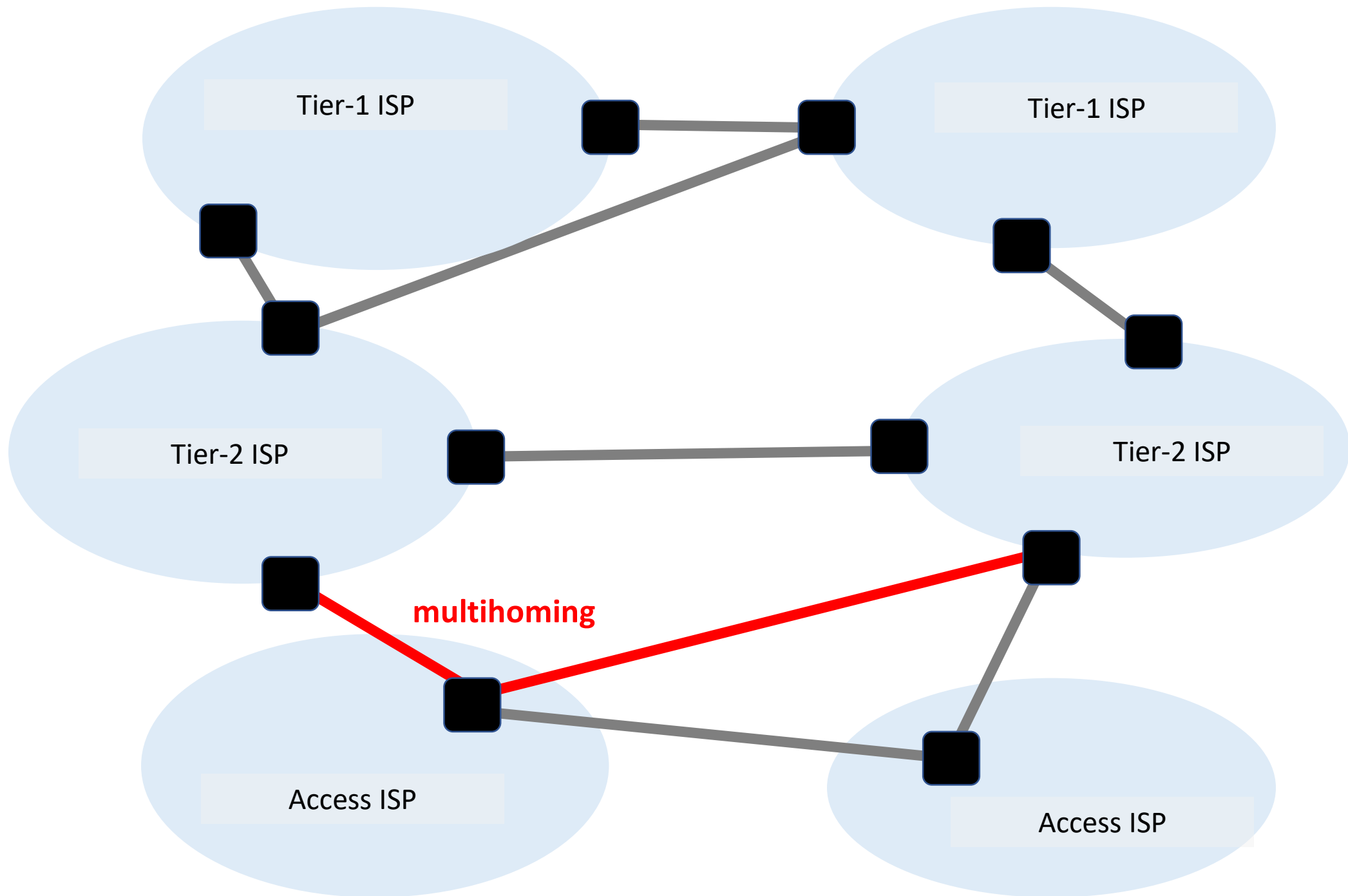
## **Tier-3**

helyi

Nem nyújt átjárást más szolgáltatóknak

Legalább egy szolgáltatója van

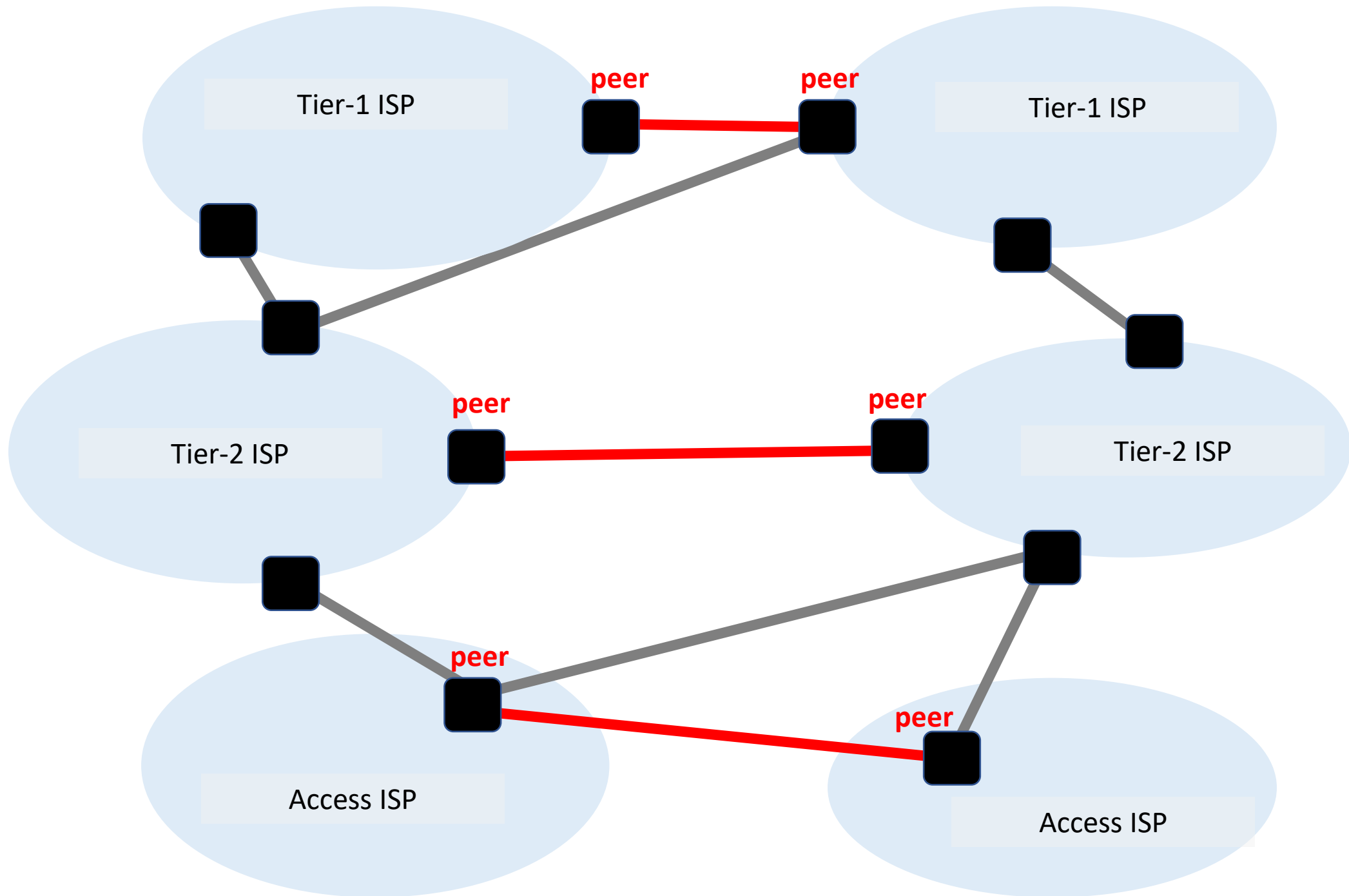
**85-90%**





Némely hálózat között közvetlen kapcsolat is létezik  
– csökkenti a szolgáltatónak fizetendő számlát

**Ezt hívjuk „peering”-nek – ez egyfajta kölcsönös kapcsolat...**



# A szomszédos hálózatok egyesével való összekapcsolása túl költséges lenne

## **Infrastruktúra költségek**

Fizikai linkek kiépítése vagy bérlése

## **Sávszélesség költségek**

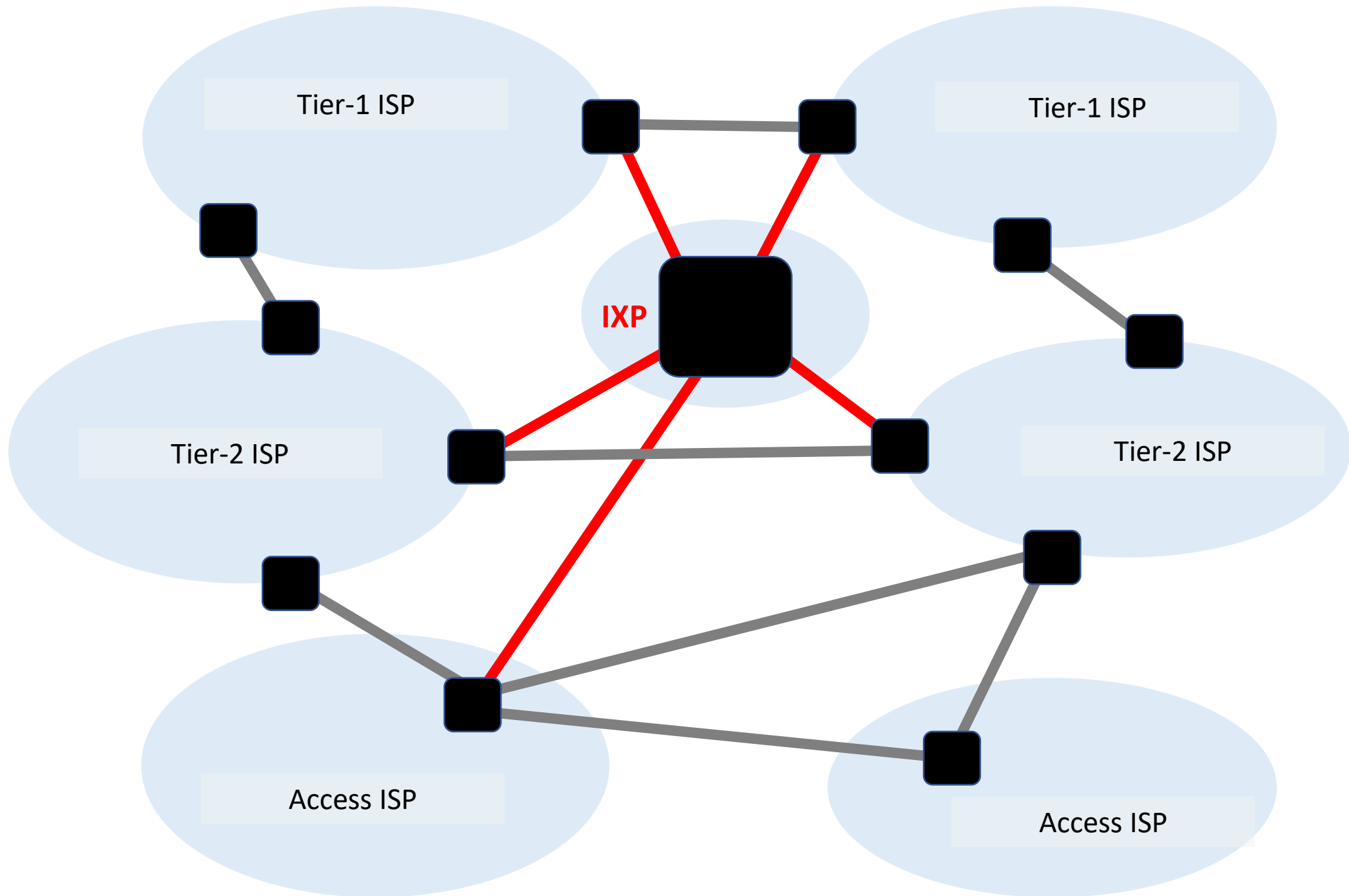
A linkek nem feltétlenül lesznek teljesen kihasználva

## **Humán költségek**

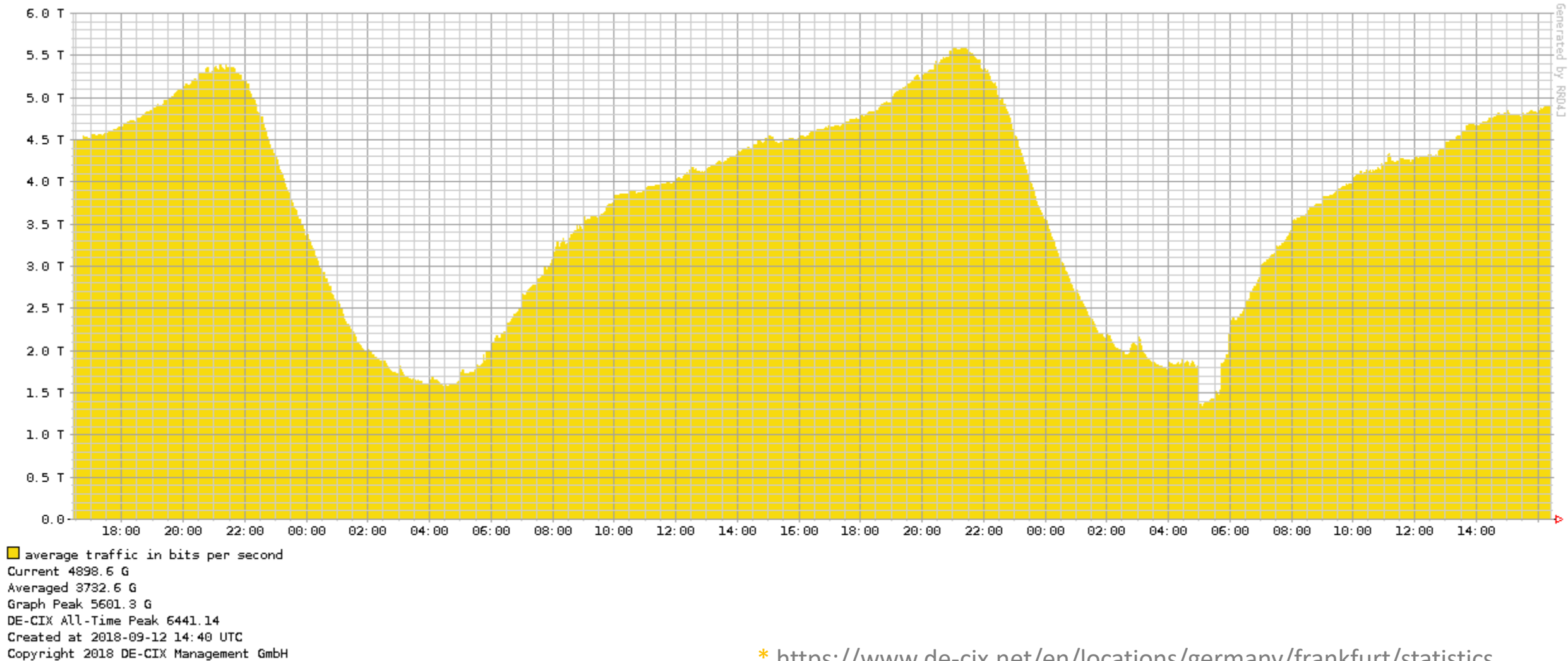
Minden kapcsolatot egyedi módon kell kezelni

A problémát az úgynevezett Internet eXchange Pontok (IXP) oldják meg

**Az IXP-k lehetővé teszik több hálózat összekapcsolását  
egy fizikai (földrajzi/topológiai) helyen.**



# Egy IXP két napja – DE-CIX Frankfurt



\* <https://www.de-cix.net/en/locations/germany/frankfurt/statistics>

# This week

How does communication happen?

How do we characterize it?

# Briefly...

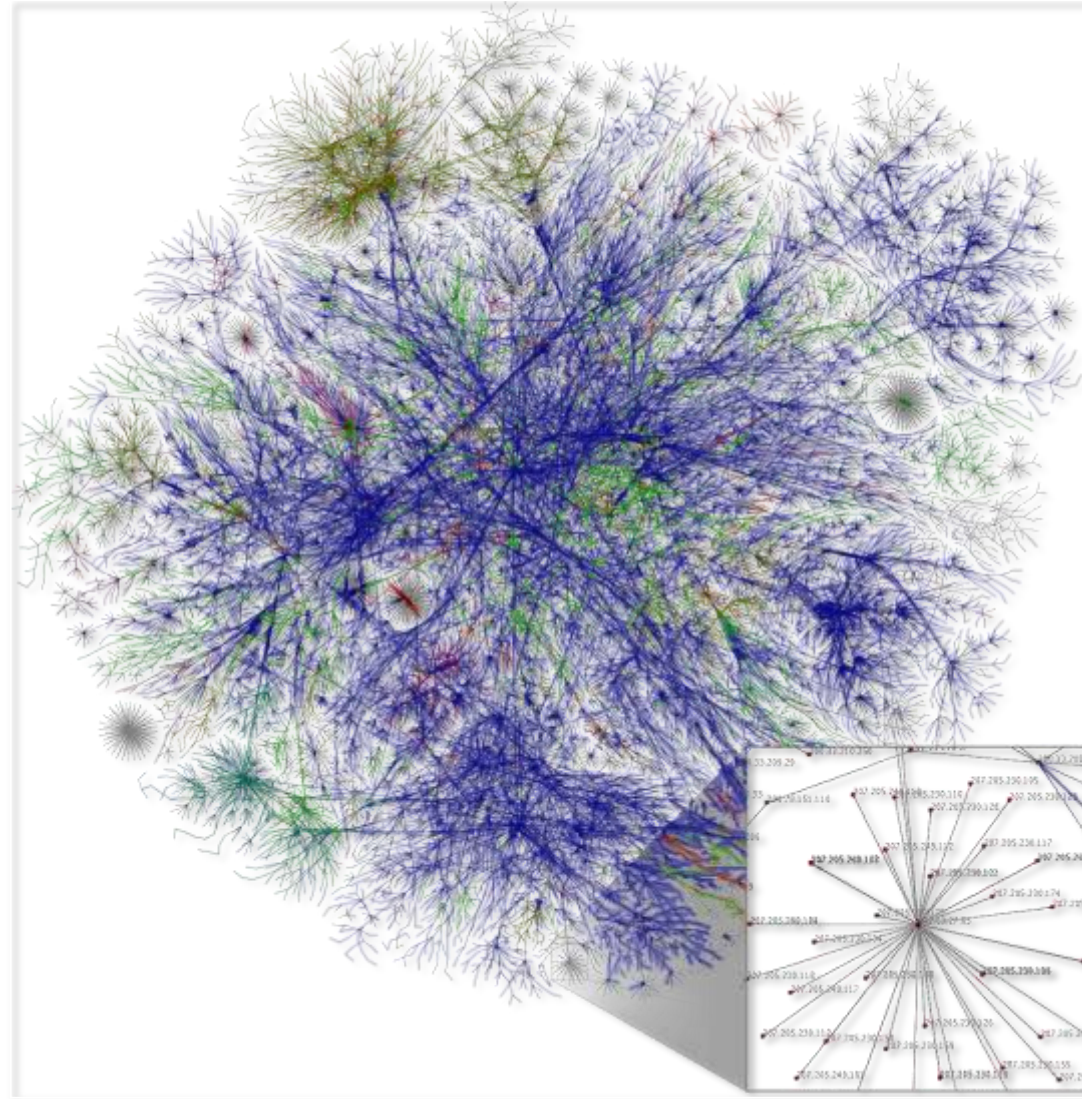
*The Internet should allow*

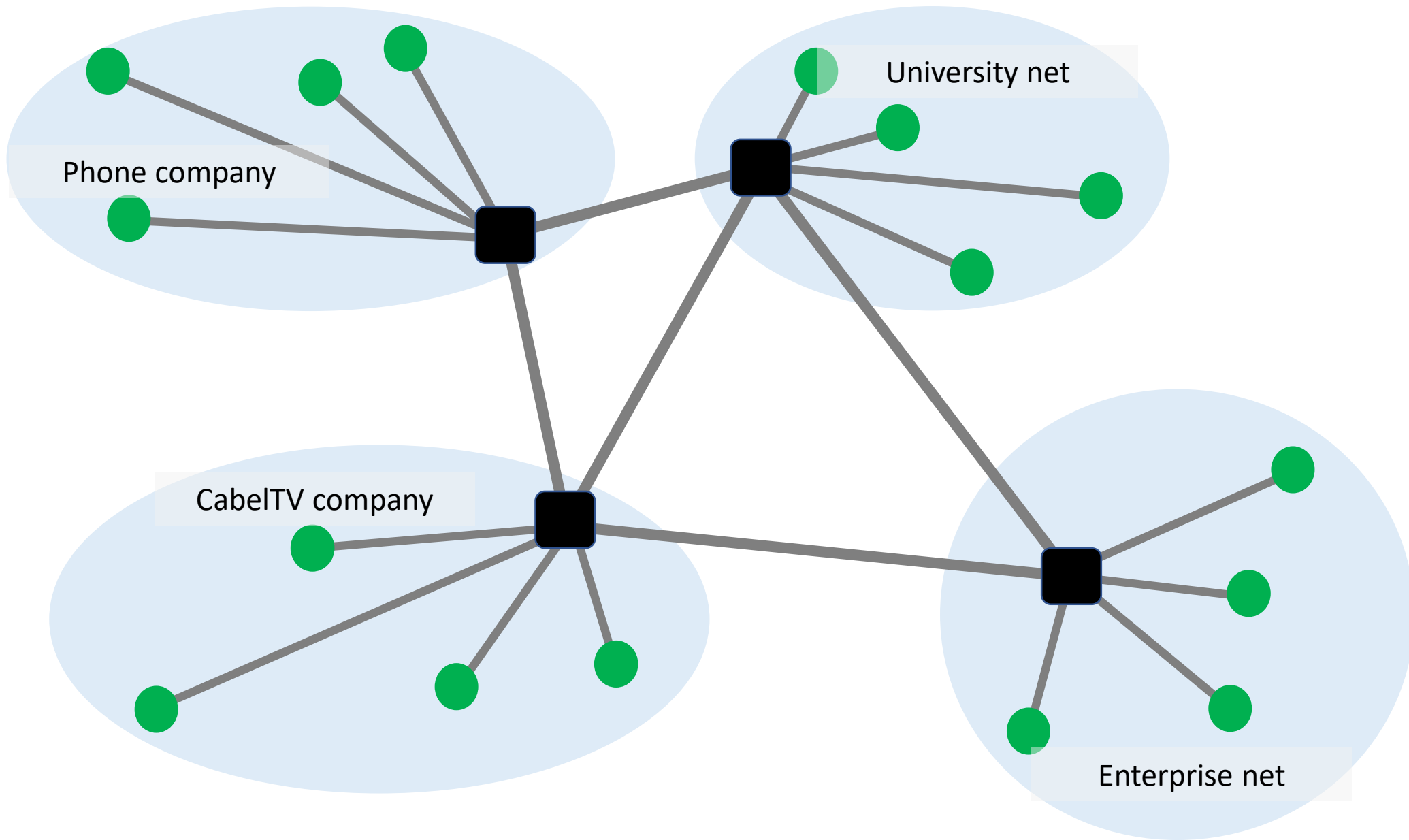
**processes on different hosts**  
**to exchange data**

*everything else is just commentary...*

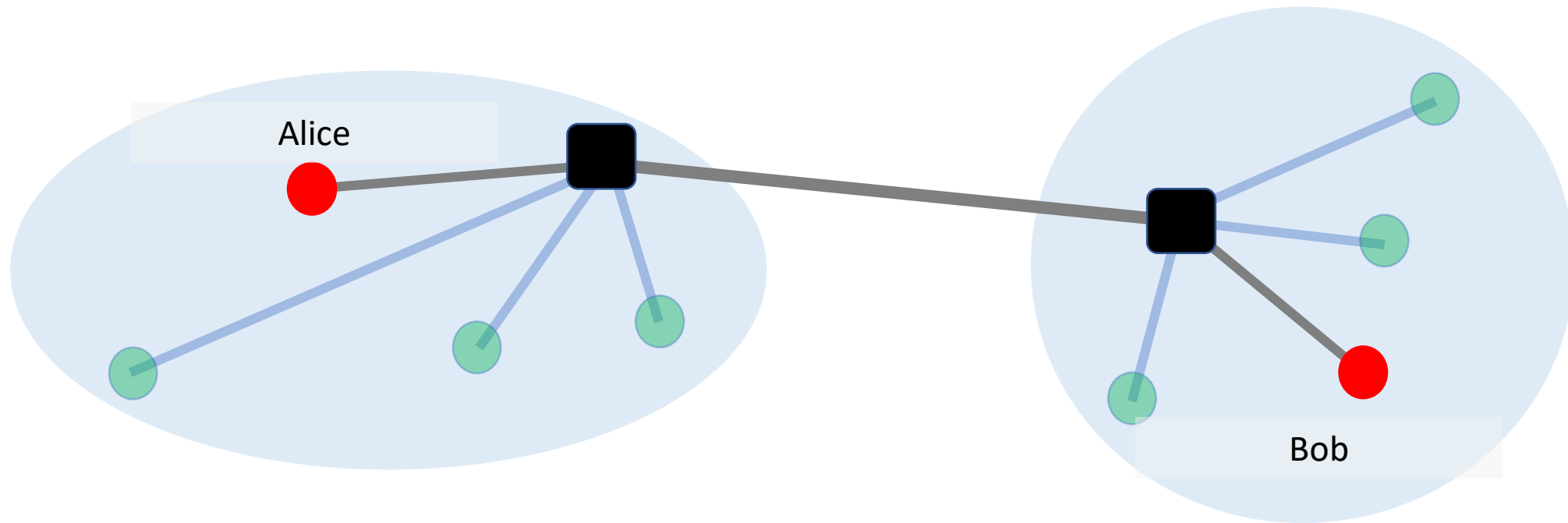


Ok, but how to do that in a complex system like the Internet?





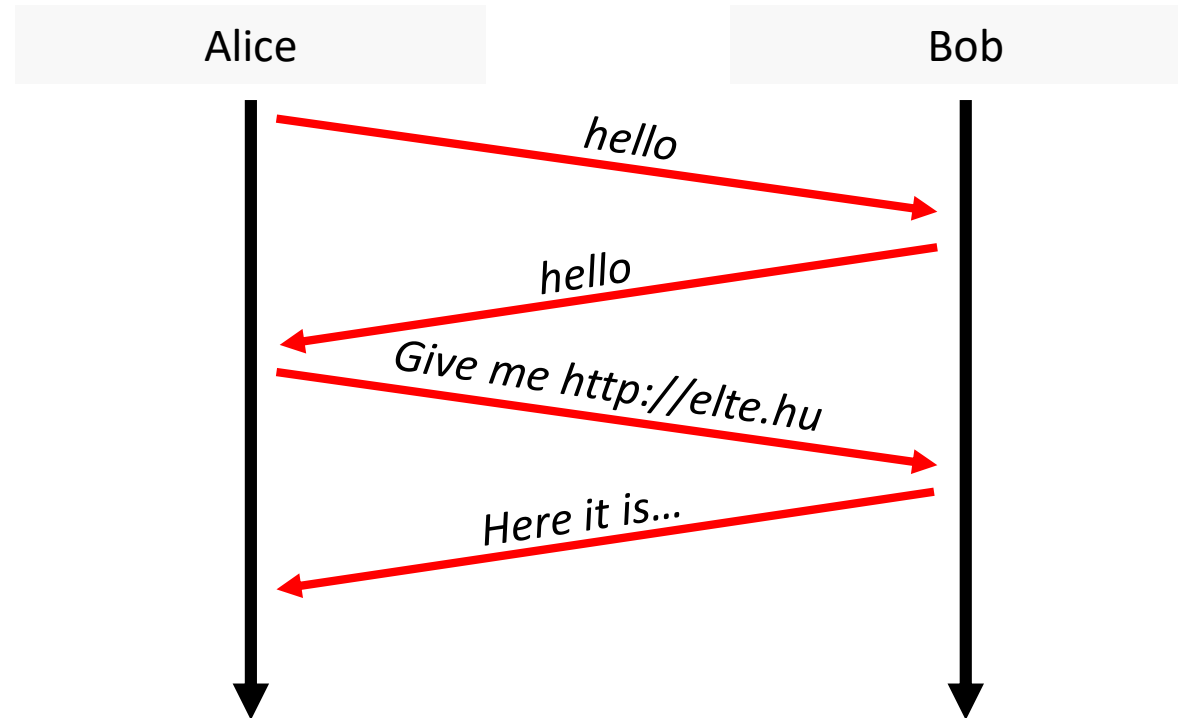
To exchange data, Alice and Bob use  
**a set of network protocols**



# A protocol is like a conversational convention

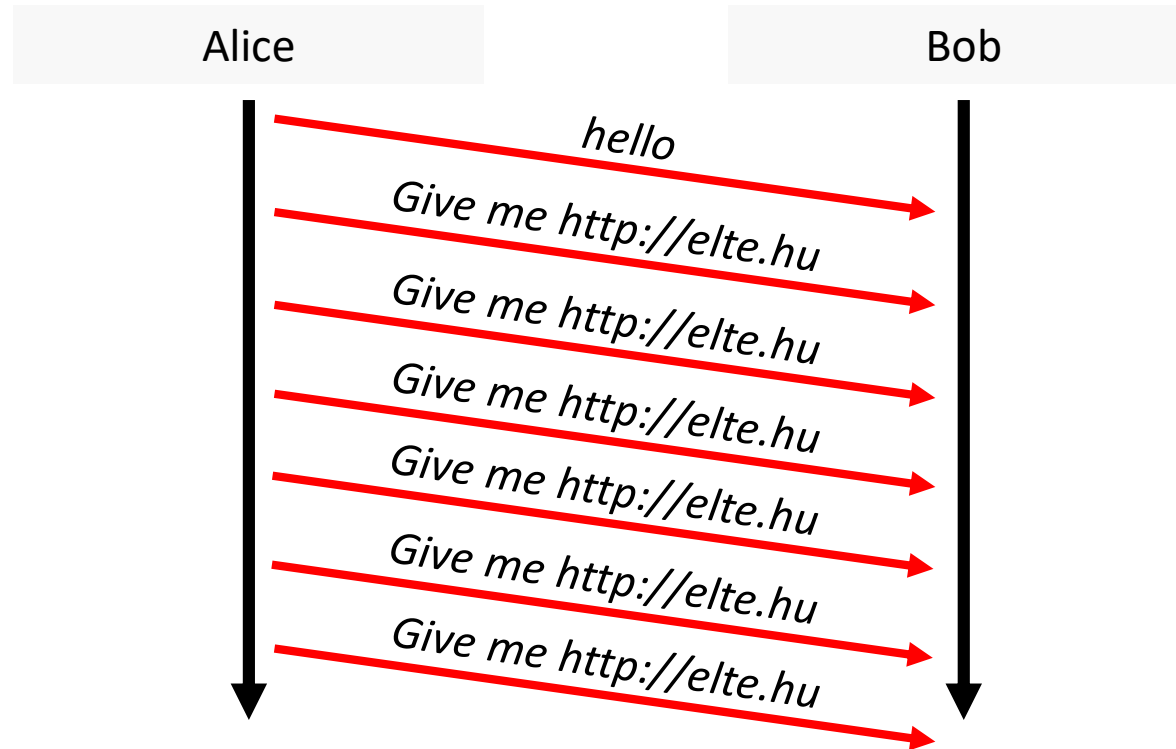
The protocol defines the order and rules the parties should follow

*Who should talk next and how to respond...*

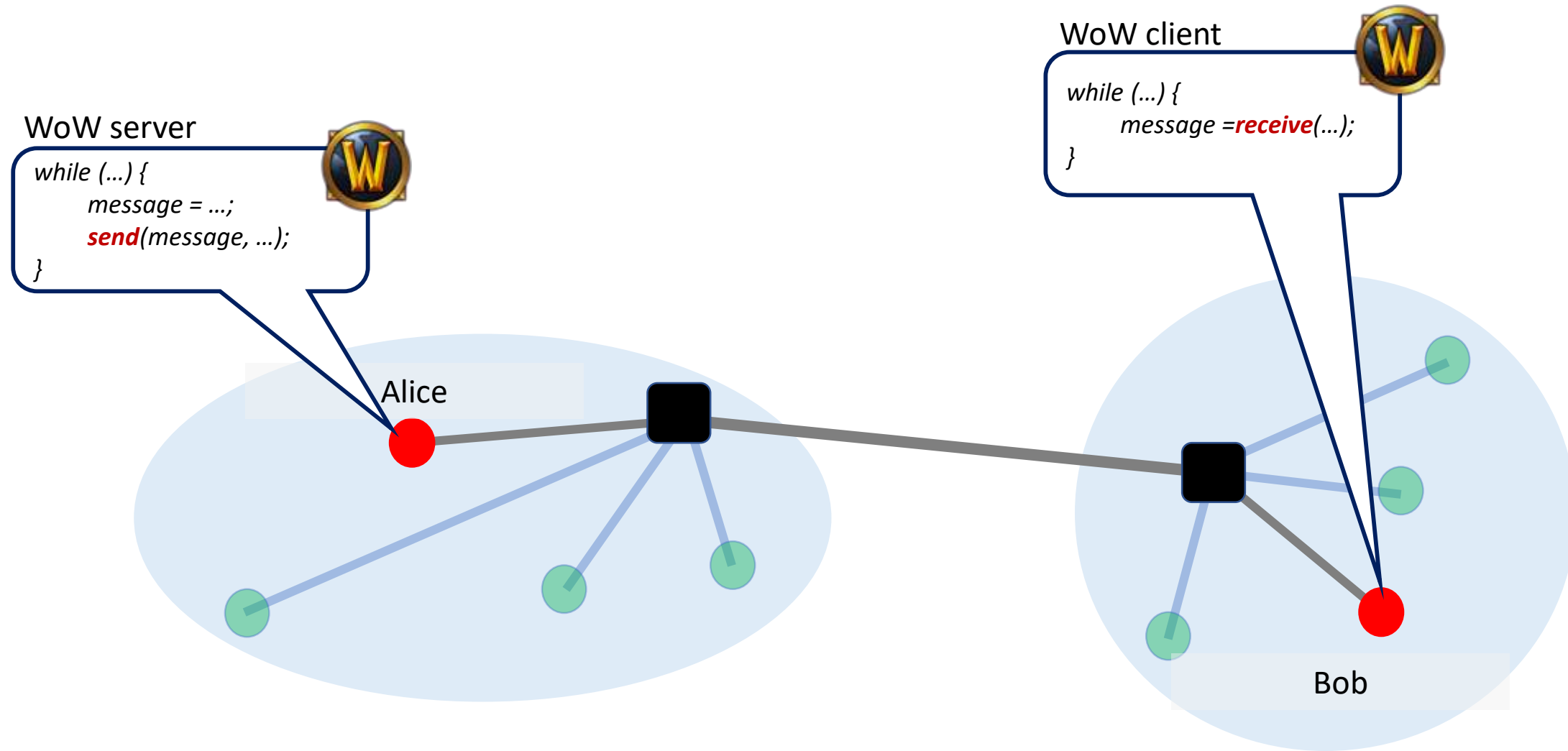


# There are other kind of implementations...

Gimme, gimme, gimme  
a web site after Midnight



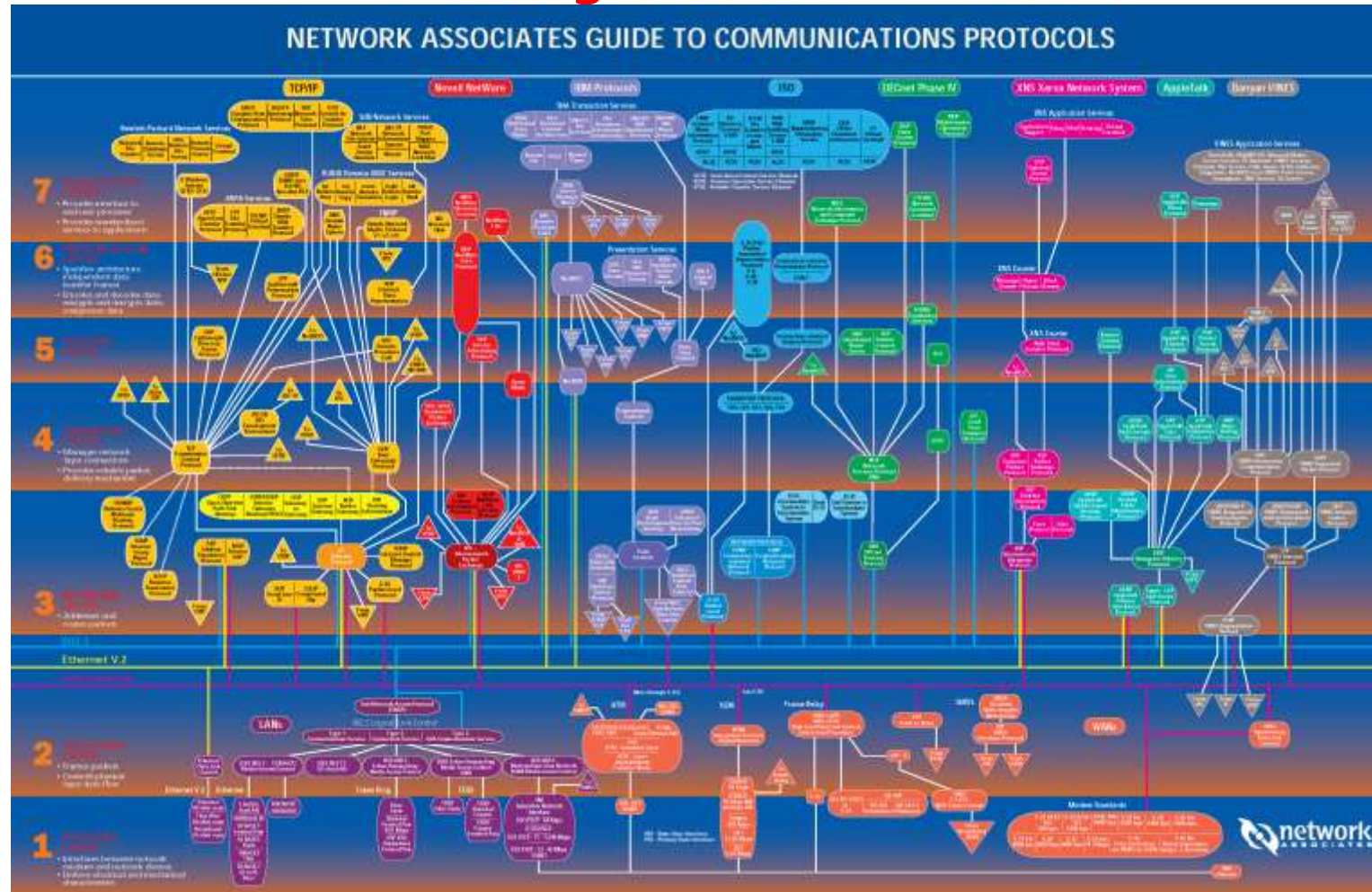
# Each protocol is governed by a specific API





# In practice, many existing protocols...

*How does the Internet **organize this**???*



HOW STANDARDS PROLIFERATE:  
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)

SITUATION:  
THERE ARE  
14 COMPETING  
STANDARDS.

14?! RIDICULOUS!  
WE NEED TO DEVELOP  
ONE UNIVERSAL STANDARD  
THAT COVERS EVERYONE'S  
USE CASES.



SOON:

SITUATION:  
THERE ARE  
15 COMPETING  
STANDARDS.



# Modularity is a key component of any good system

## **Problem**

**can't build large systems out of spaghetti code**

*hard (if not, impossible) to understand, debug, update*

**need to bound the scope of changes**

*evolve the system without rewriting it from scratch*

## **Solution**

**Modularity is how we do it**

*...and understand the system at a higher-level*



„Modularity,  
based on abstraction,  
is **the way** things get done”

*Barbara Liskov, MIT*

To provide structure to the design of network protocols,  
network designers organize **protocols** in layers

**and the network hardware/software  
that implement them**

# Hálózatok modelljei

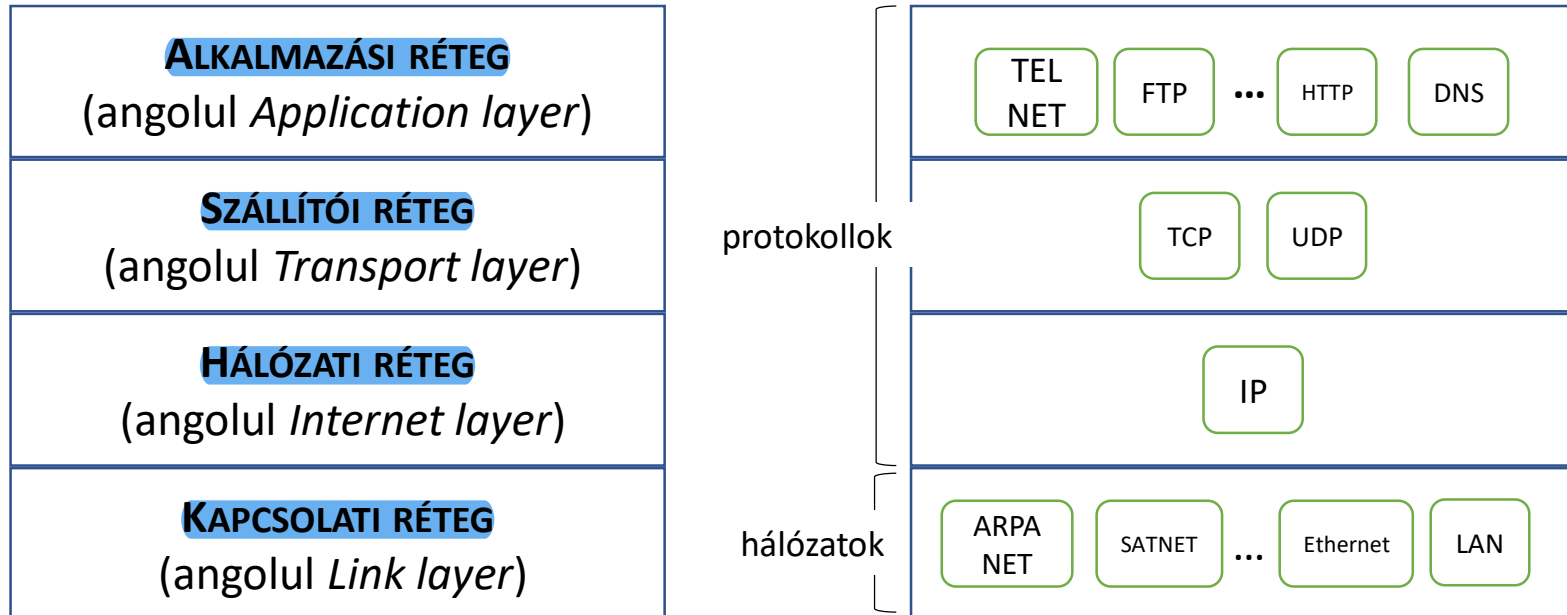
- Internet rétegmodelljei

- TCP/IP modell: 4 réteget különböztet meg. 1982 márciusában az amerikai hadászati célú számítógépes hálózatok standardja lett. 1985-től népszerűsítették kereskedelmi felhasználásra. (*Interop*)
- Hibrid TCP/IP modell: 5 réteget különböztet meg (*Tanenbaum, Stallings, Kurose, Forouzan*)

- Nyílt rendszerek hálózatának standard modellje

- *Open System Interconnection Reference Model*: Röviden OSI referencia modell, amely egy 7-rétegű standard, koncepcionális modellt definiál kommunikációs hálózatok belső funkcióit. (*ISO/IEC 7498-1*)

# TCP/IP modell (RFC 1122)



# TCP/IP modell rétegei („bottom-up”)

- **Kapcsolati réteg / Host-to-network or Link layer**
  - nem specifikált
  - a LAN-tól függ
- **Internet réteg / Internet or Network layer**
  - speciális csomagformátum
  - útvonal meghatározás (routing)
  - csomag továbbítás (angolul *packet forwarding*)
- **Szállítói réteg / Transport layer**
  - **Transport Control Protocol**
    - megbízható, kétirányú bájt-folyam átviteli szolgáltatás
    - szegmentálás, folyamfelügyelet, multiplexálás
  - **User Datagram Protocol**
    - nem megbízható átviteli szolgáltatás
    - nincs folyamfelügyelet
- **Alkalmazási réteg / Application layer**
  - Szolgáltatások nyújtása: Telnet, FTP, SMTP, HTTP, NNTP, DNS, SSH, etc.

# Rétegek jellemzése



- Szolgáltatás
  - Mit csinál az adott réteg?
- Interfész
  - Hogyan férhetünk hozzá a réteghez?
- Protokoll
  - Hogyan implementáljuk a réteget?

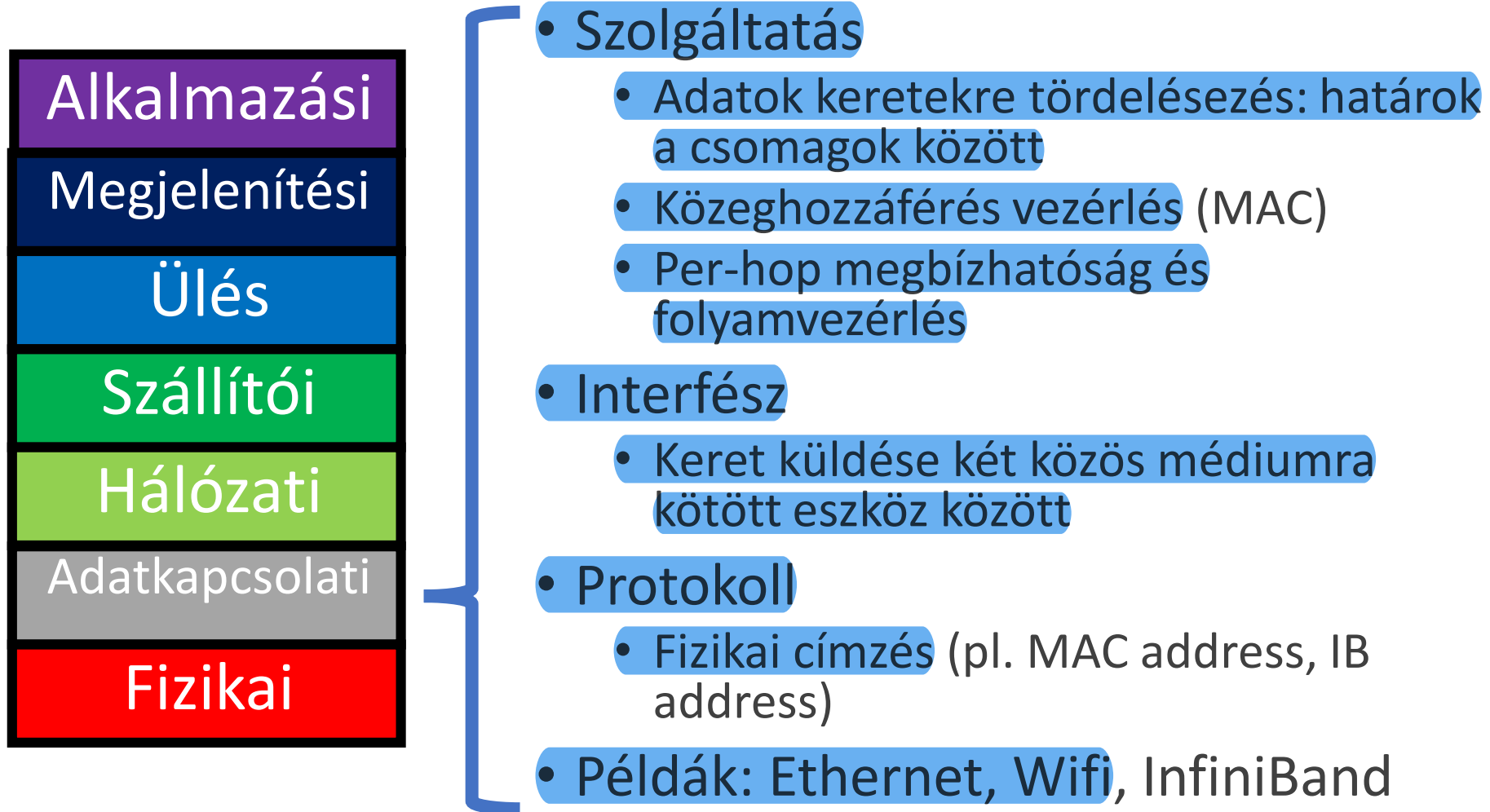
# Fizikai réteg



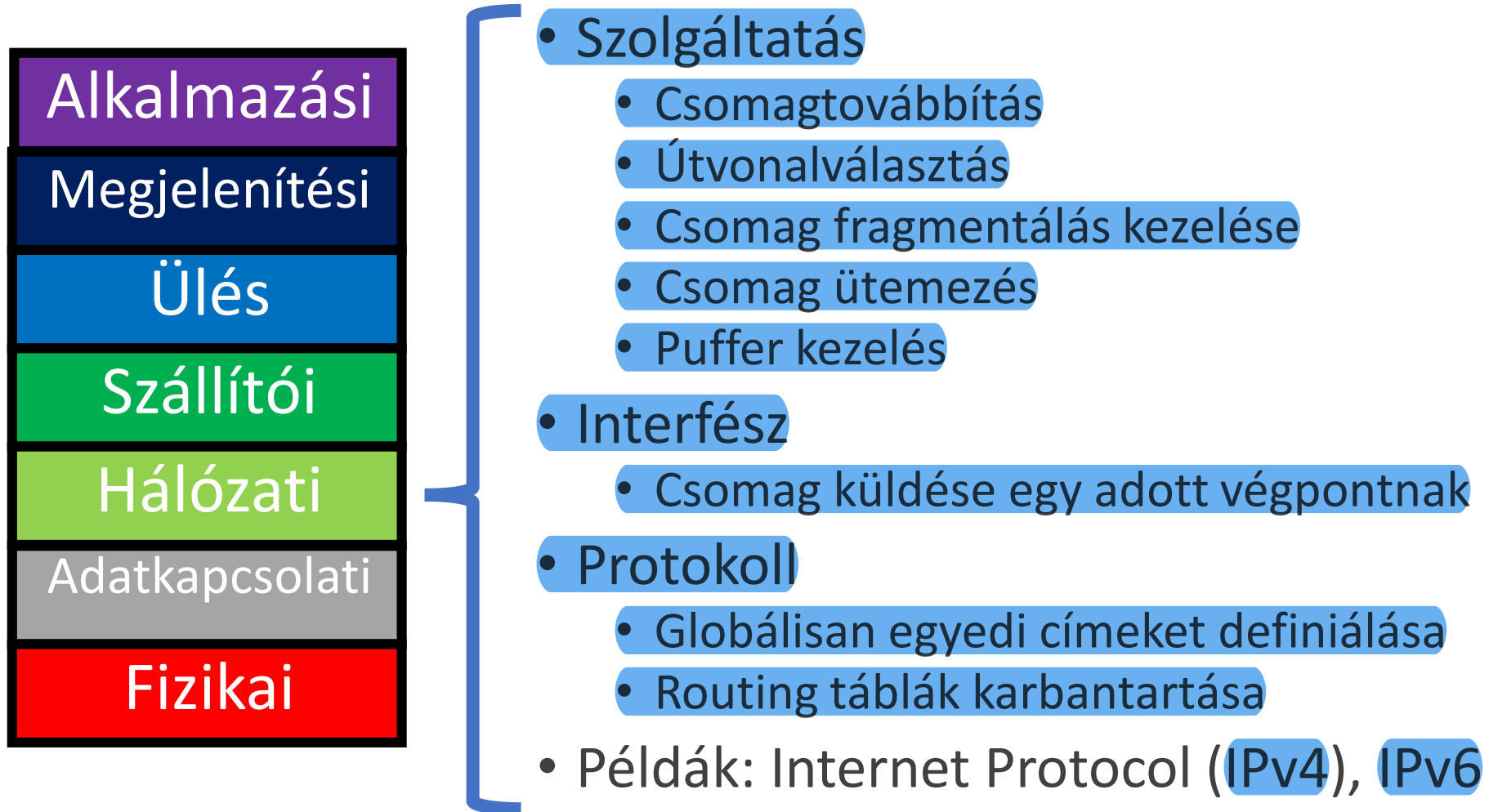
- Szolgáltatás
  - Információt visz át két fizikailag összekötött eszköz között
  - definiálja az eszköz és a fizikai átviteli közeg kapcsolatát
- Interfész
  - Specifikálja egy bit átvitelét
- Protokoll
  - Egy bit kódolásának sémája
  - Feszültség szintek
  - Jelek időzítése
- Példák: koaxiális kábel, optikai kábel, rádió frekvenciás adó



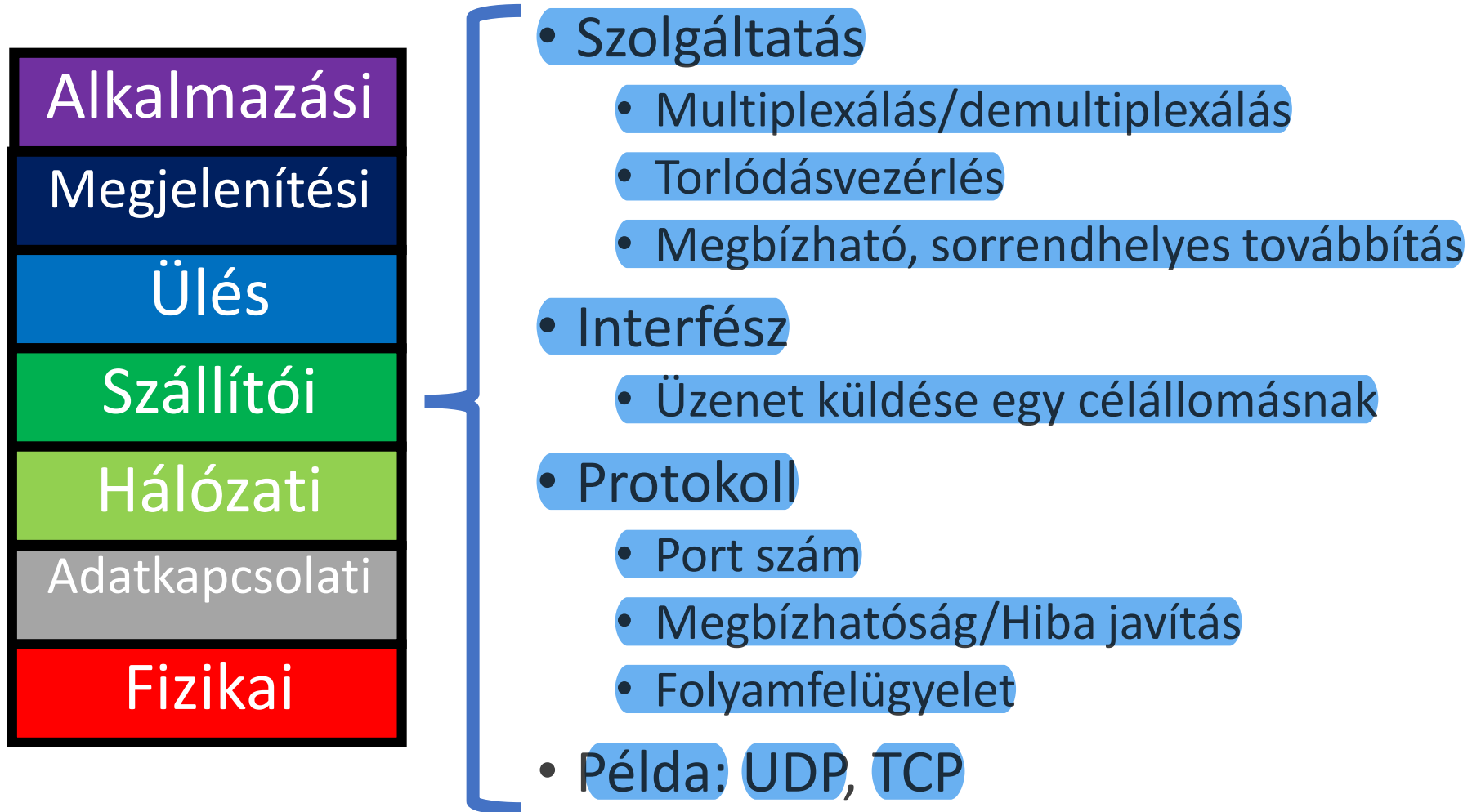
# Adatkapcsolati réteg



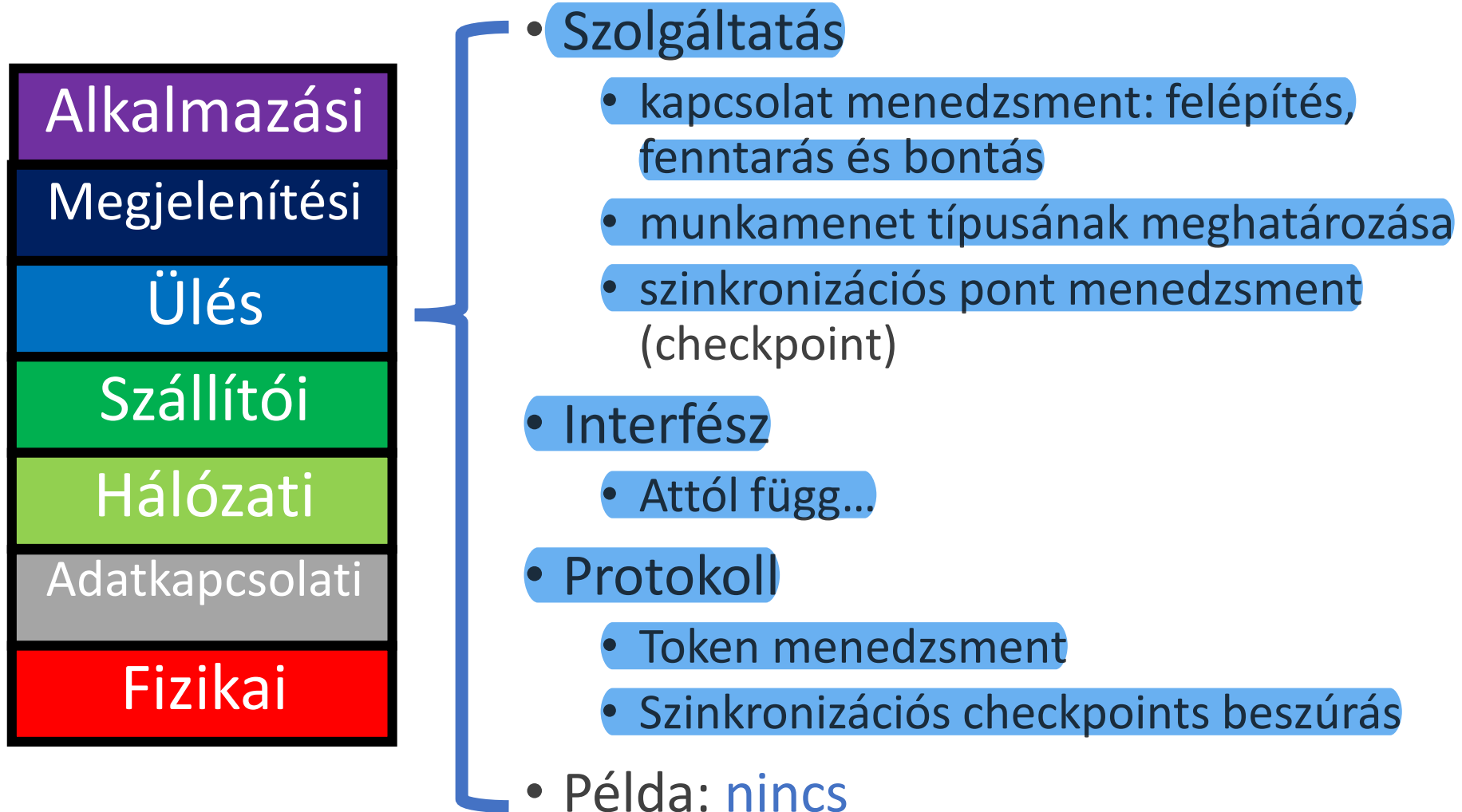
# Hálózati réteg



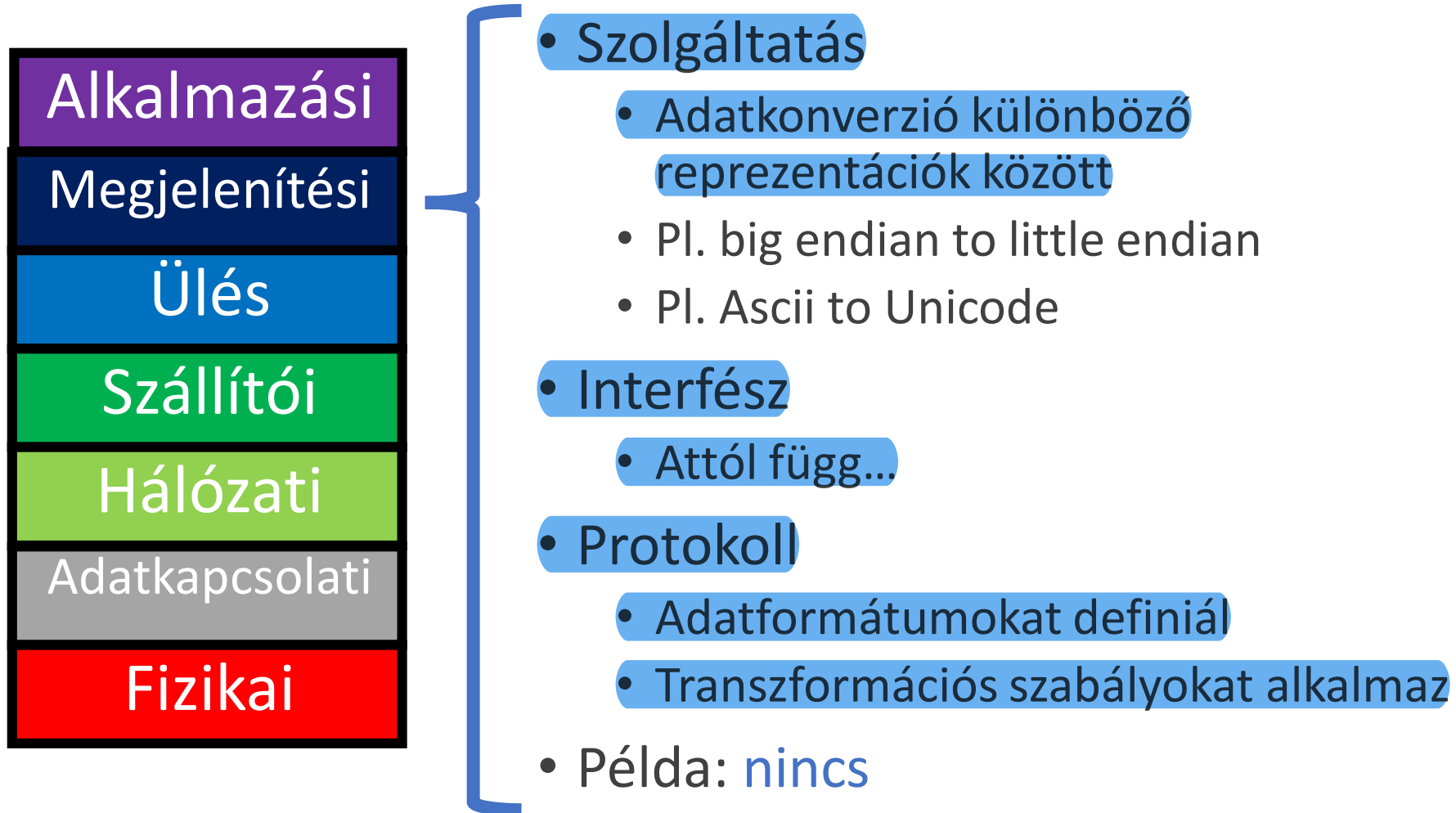
# Szállítói réteg



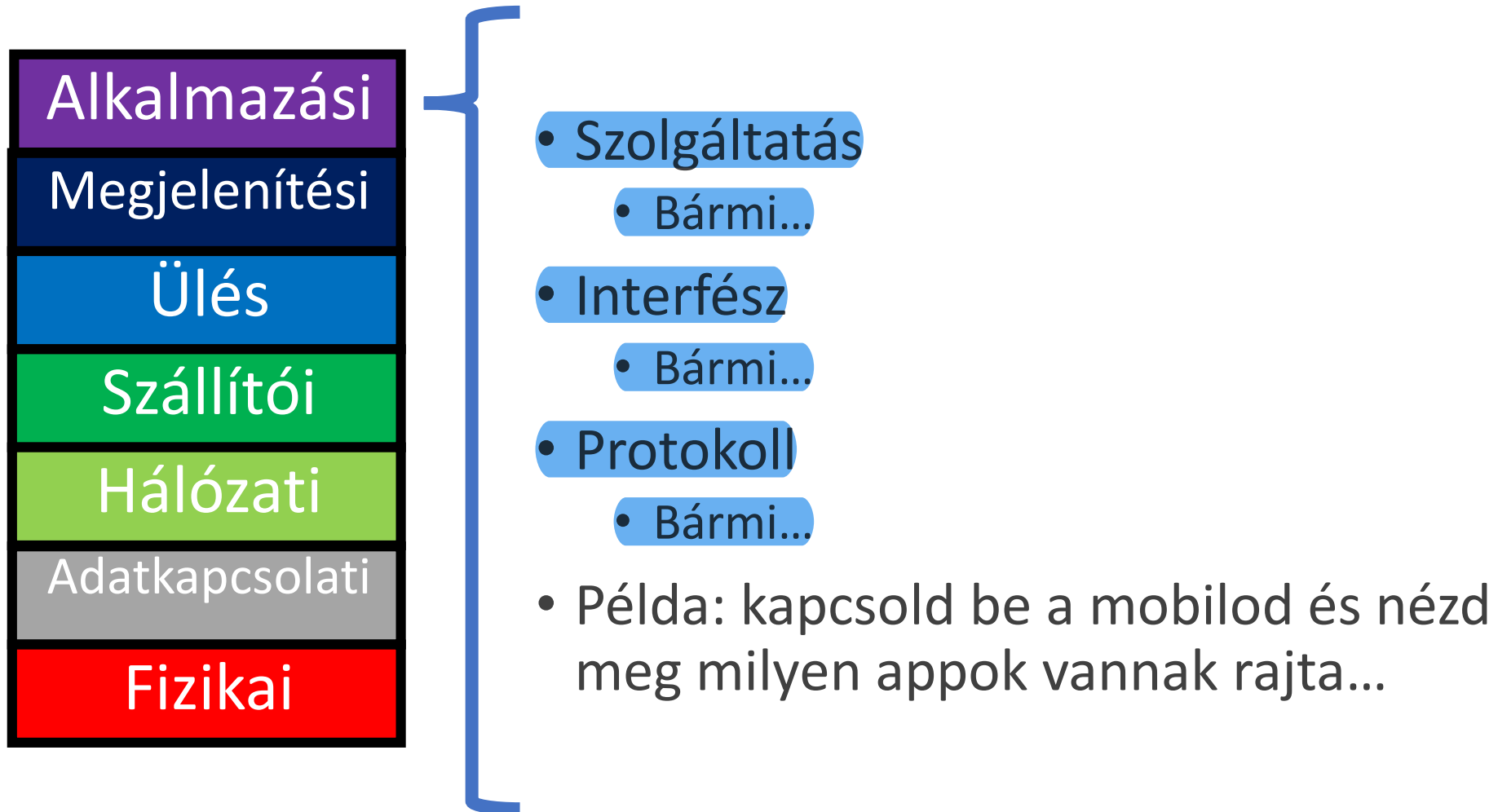
# Ülés v. Munkamenet réteg



# Megjelenítési réteg

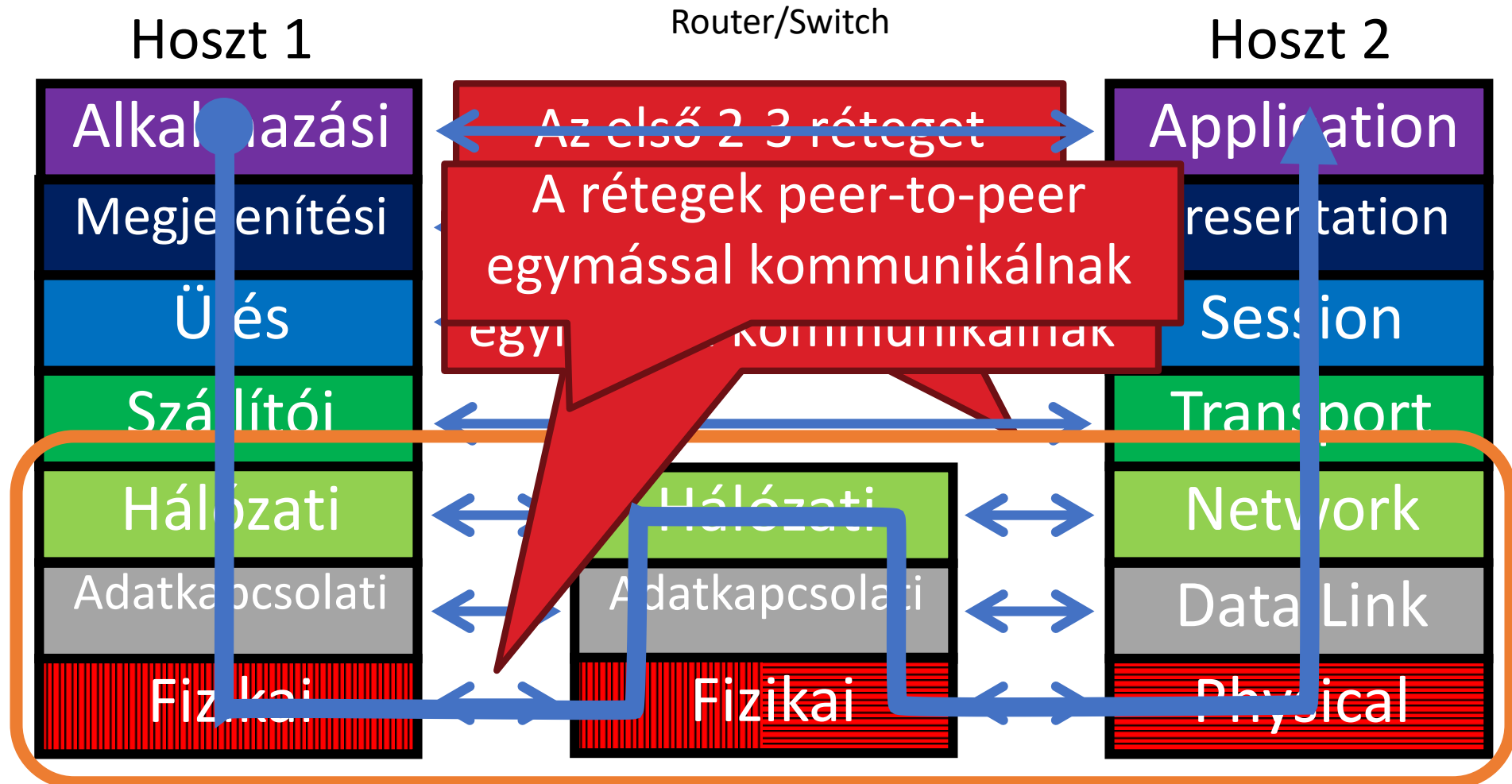


# Alkalmazási réteg



# ISO OSI modell

## OSI: Open Systems Interconnect Model



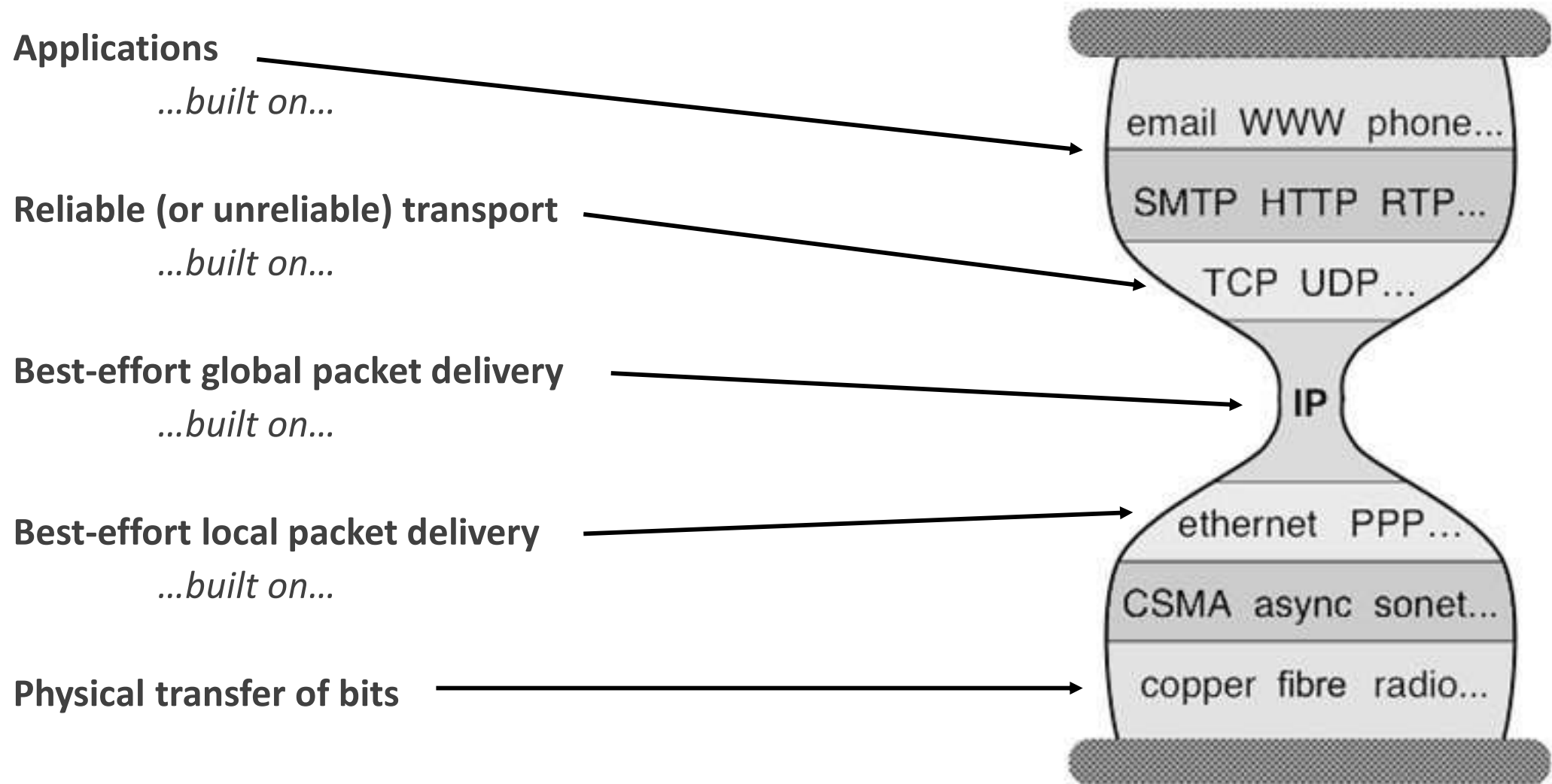
# Hybrid model – 5 layers

Each layer provides a service to the layer above

<u>layer</u>	<u>service provided</u>
<b>L5 Application</b>	<i>high level network access</i>
<b>L4 Transport</b>	<i>end-to-end delivery (reliable or not)</i>
<b>L3 Network</b>	<i>global best-effort delivery</i>
<b>L2 Link</b>	<i>local best-effort delivery</i>
<b>L1 Physical</b>	<i>physical transfer of bits</i>



Each layer provides a service to the layer above  
by using the services of the layer directly below it



Each layer has a **unit of data**  
(aka **protocol data unit**)

	layer	role ( <b>PDU</b> )
<b>L5</b>	<b>Application</b>	exchanges <b>messages</b> between processes
<b>L4</b>	<b>Transport</b>	transports <b>segments</b> between end-systems
<b>L3</b>	<b>Network</b>	moves <b>packets</b> around the network
<b>L2</b>	<b>Link</b>	moves <b>frames</b> across a link
<b>L1</b>	<b>Physical</b>	moves <b>bits</b> across a physical medium

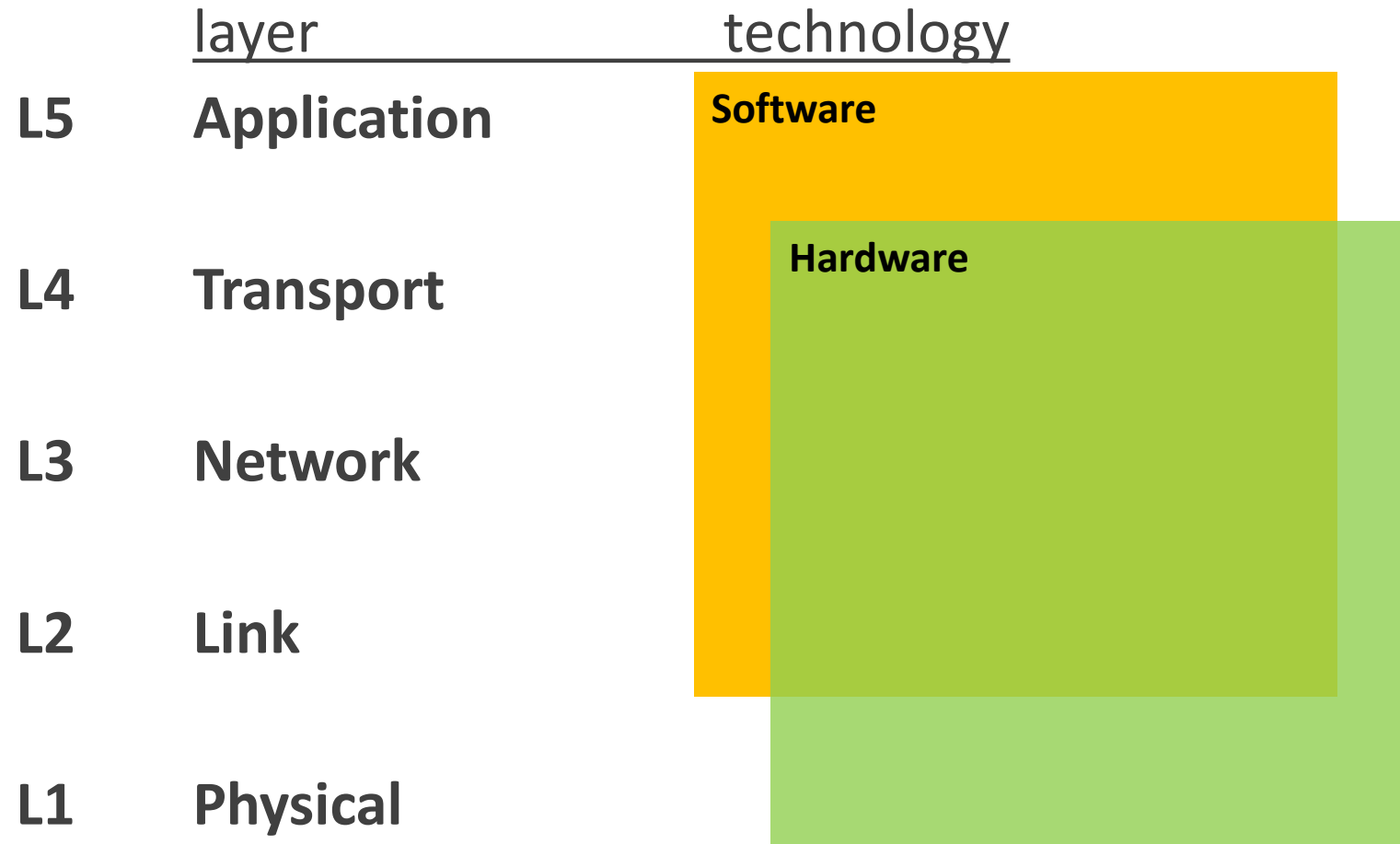
Each layer (except for L3) is implemented with different protocols

	<u>layer</u>	<u>protocols</u>
<b>L5</b>	<b>Application</b>	HTTP, SMTP, FTP, SIP, ...
<b>L4</b>	<b>Transport</b>	TCP, UDP, SCTP
<b>L3</b>	<b>Network</b>	IP
<b>L2</b>	<b>Link</b>	Ethernet, Wifi, ADSL, WiMAX, LTE, ...
<b>L1</b>	<b>Physical</b>	Twisted pair, fiber, coaxial cable, ...

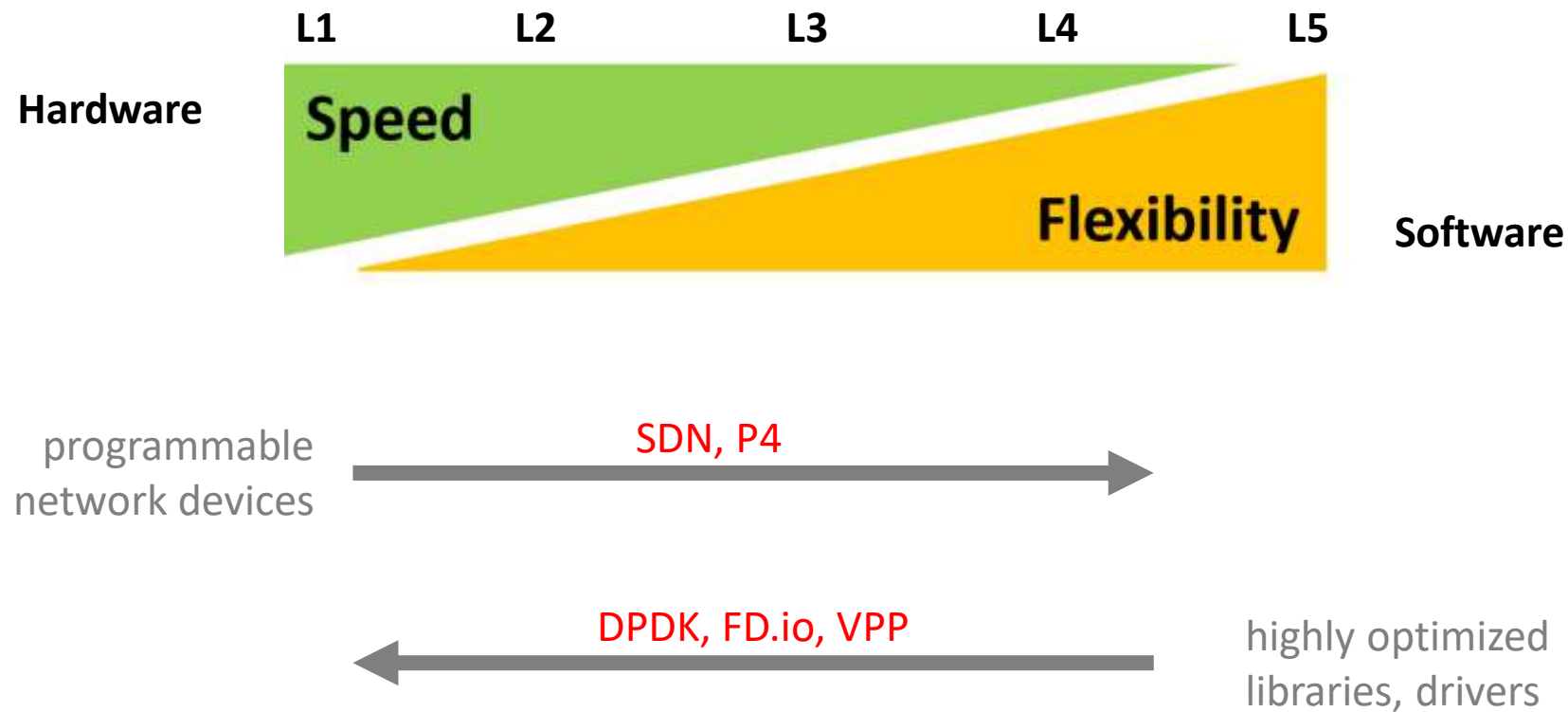
# The Internet Protocol (IP) is the glue acting as a unifying network layer

	<u>layer</u>	<u>protocols</u>
L5	Application	HTTP, SMTP, FTP, SIP, ...
L4	Transport	TCP, UDP, SCTP
L3	Network	IP
L2	Link	Ethernet, Wifi, ADSL, WiMAX, LTE, ...
L1	Physical	Twisted pair, fiber, coaxial cable, ...

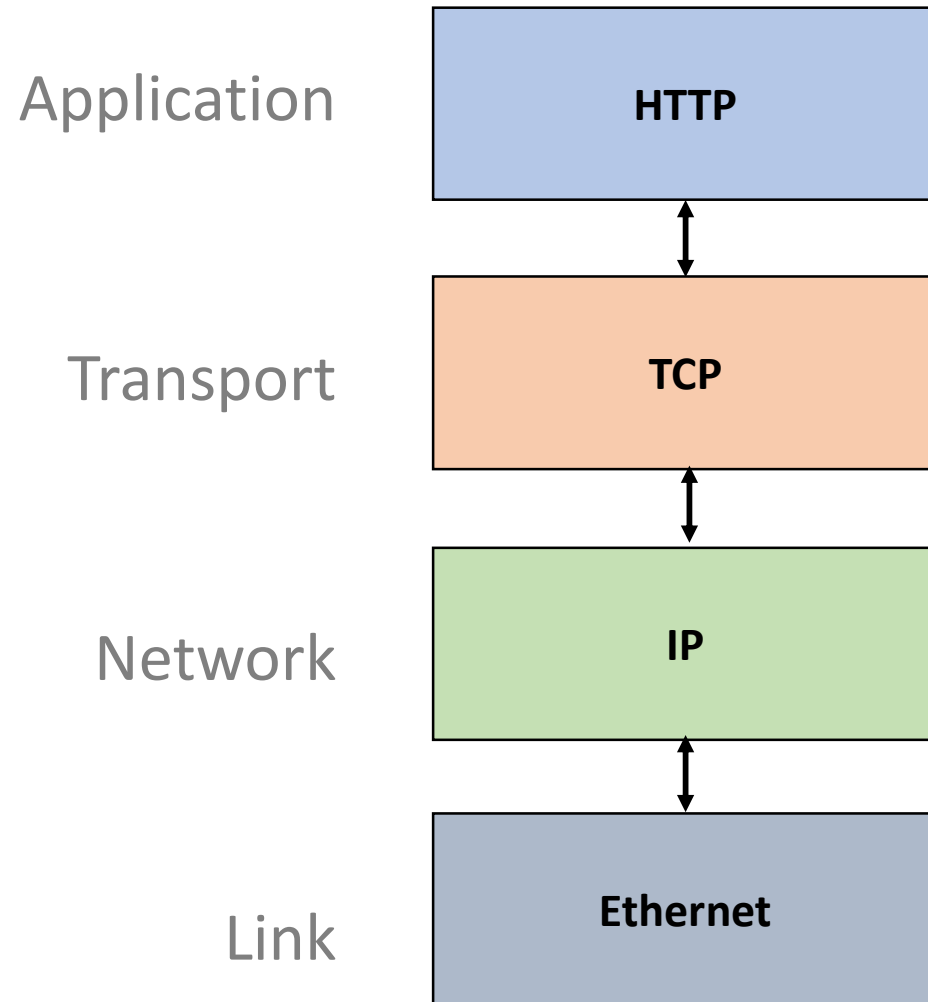
Each layer is implemented with  
different protocols and technologies



# Software and hardware advancements



Each layer takes messages from the layer above,  
and *encapsulates* with its own header and/or trailer



Application

**HTTP**

Transport

**TCP**

Network

**IP**

Link

**Ethernet**



Header

**HA**

Message

**GET google.com**





Application

**HTTP**

Transport

**TCP**

Network

**IP**

Link

**Ethernet**



Header

Message

HA

**GET google.com**

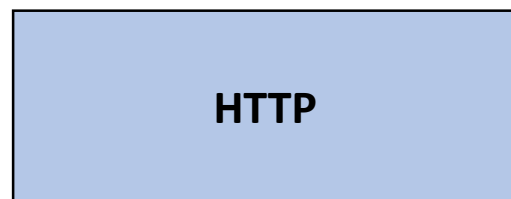


HT

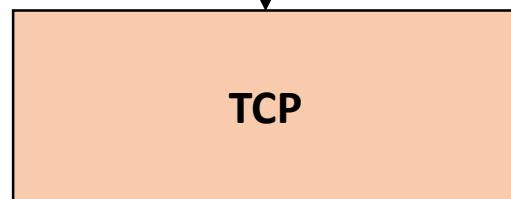
HA

**GET google.com**

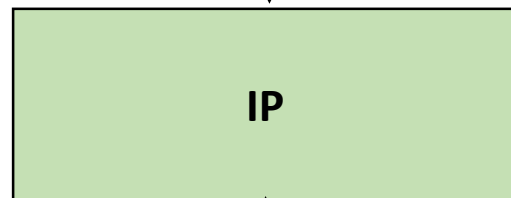
Application



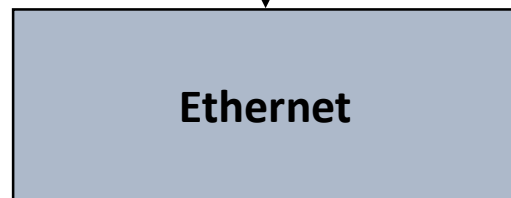
Transport



Network



Link



Header

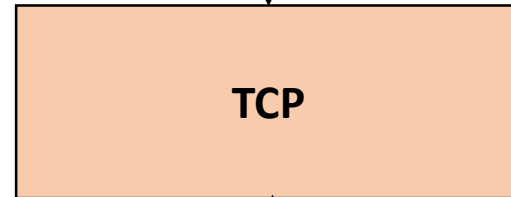
Message



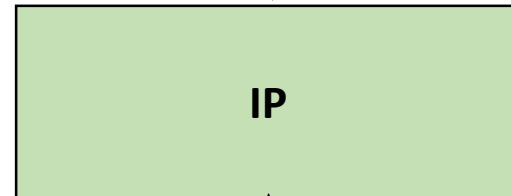
Application



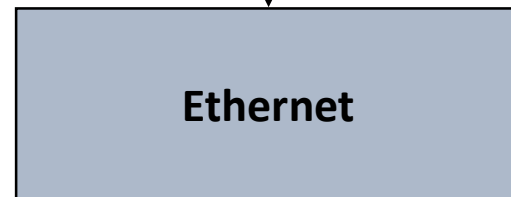
Transport



Network

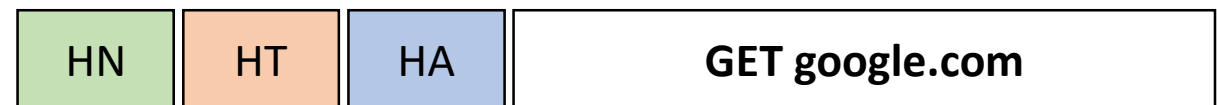


Link

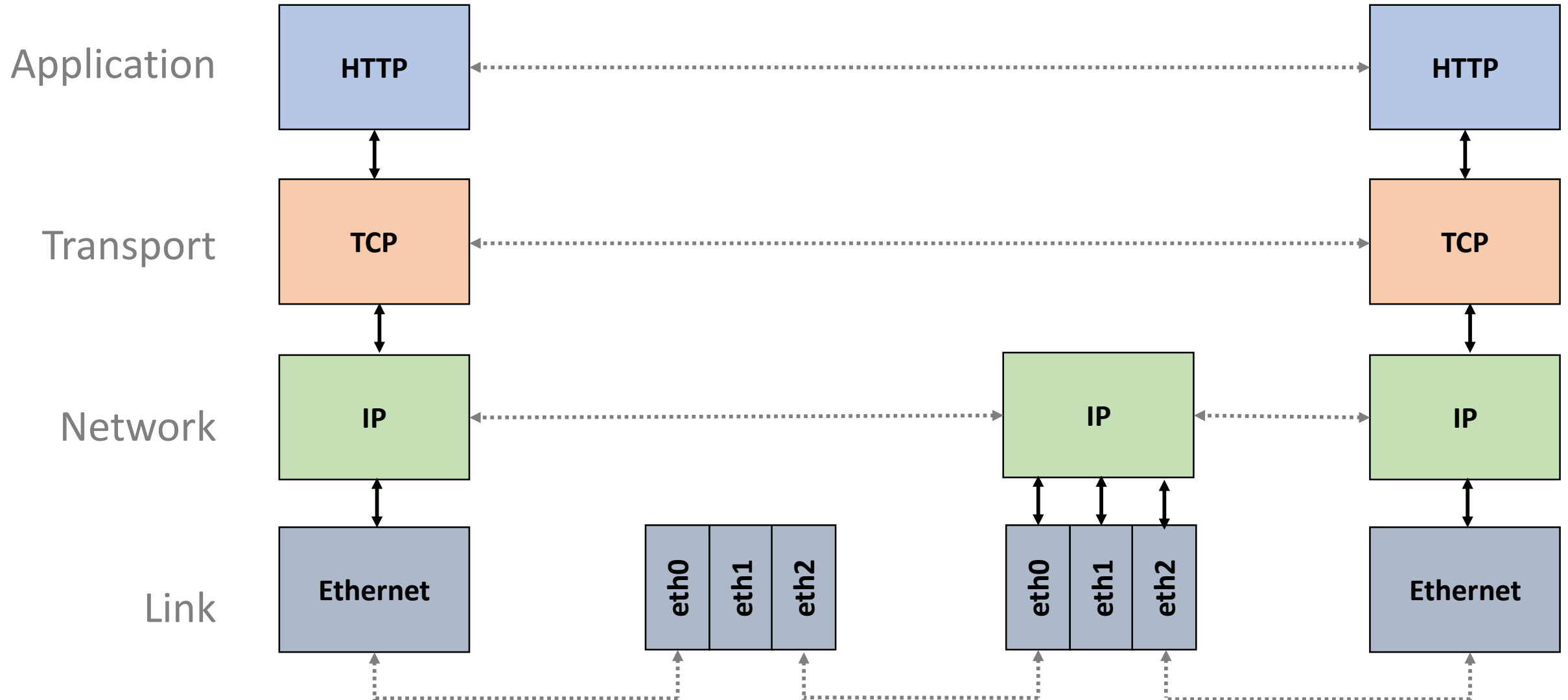


Header

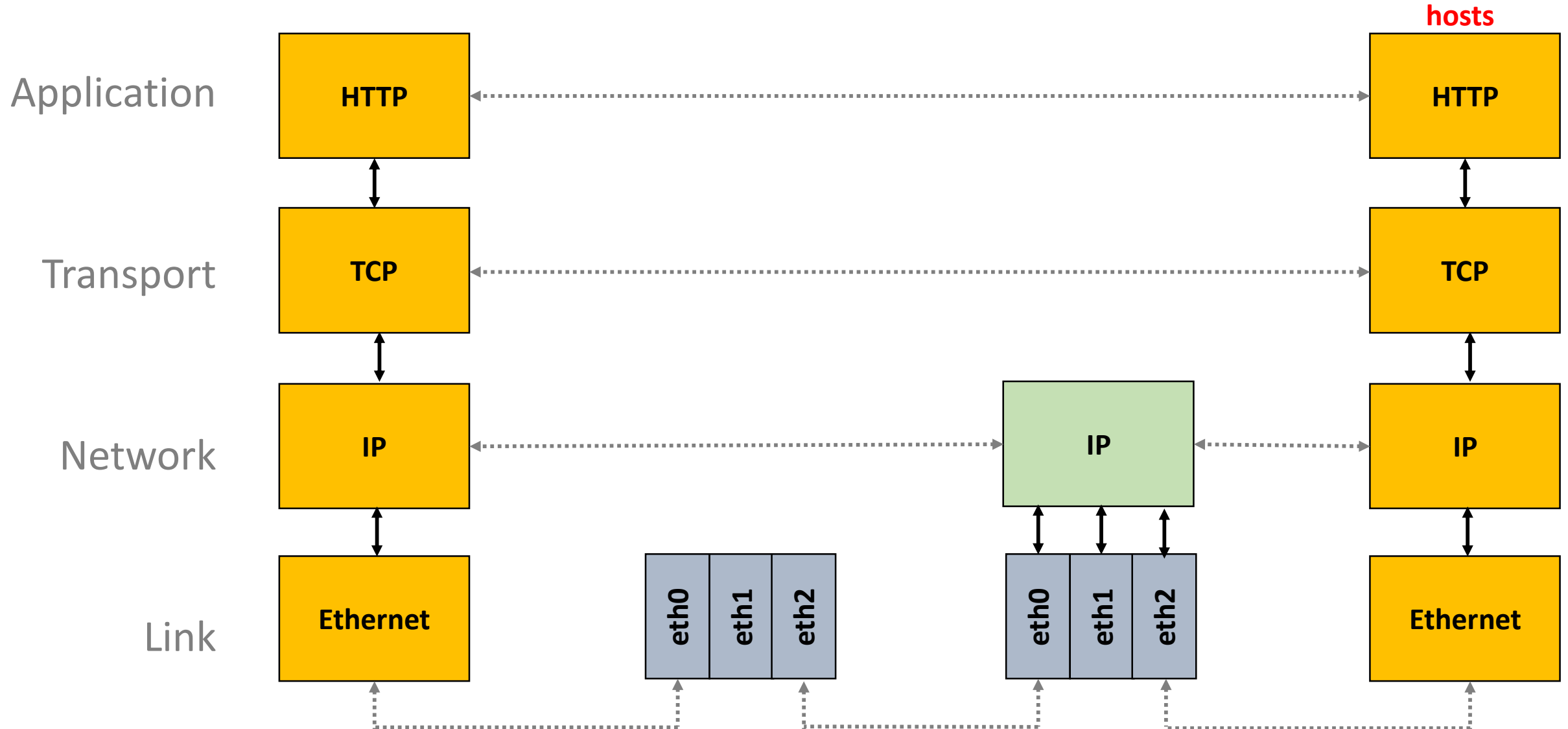
Message



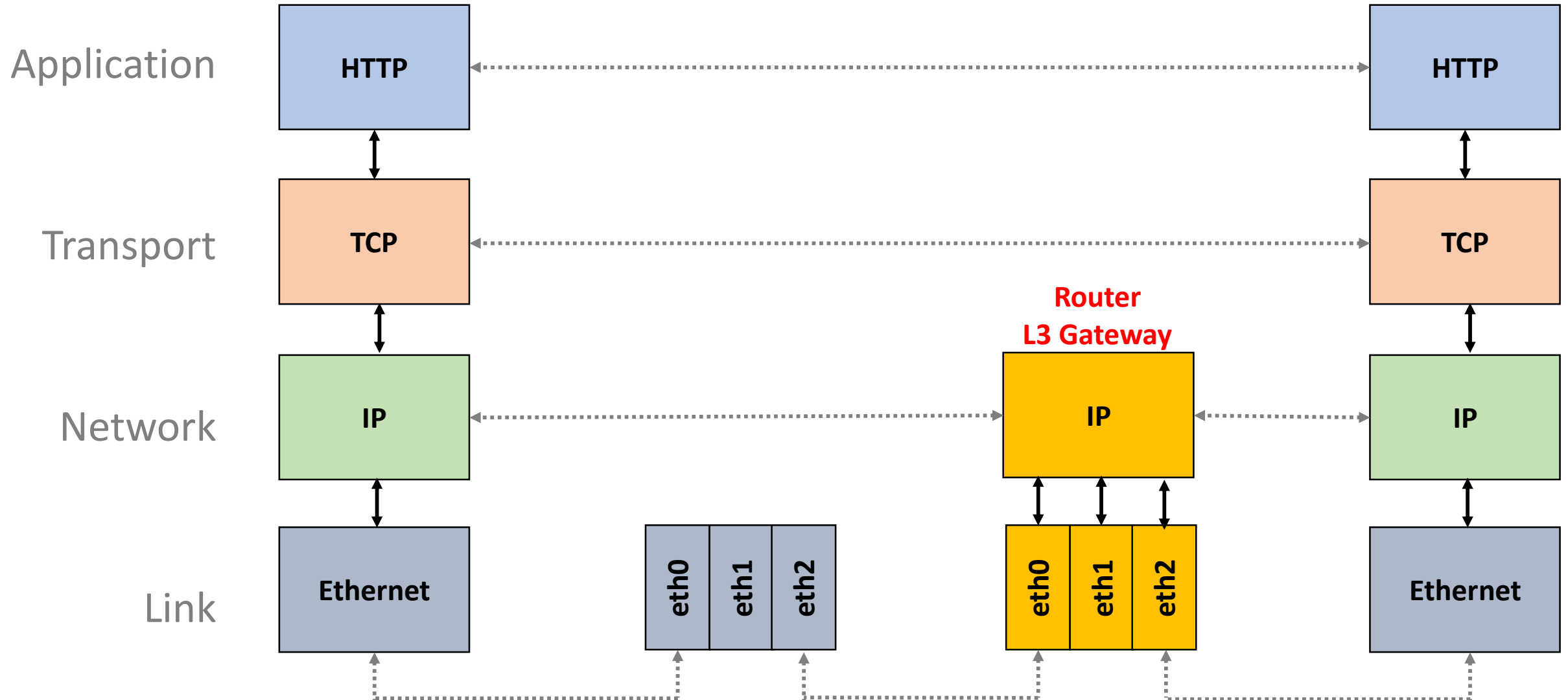
In practice, layers are **distributed**  
on every network device



Since when bits arrive they must make it to the application, **all the layers exist on a host**

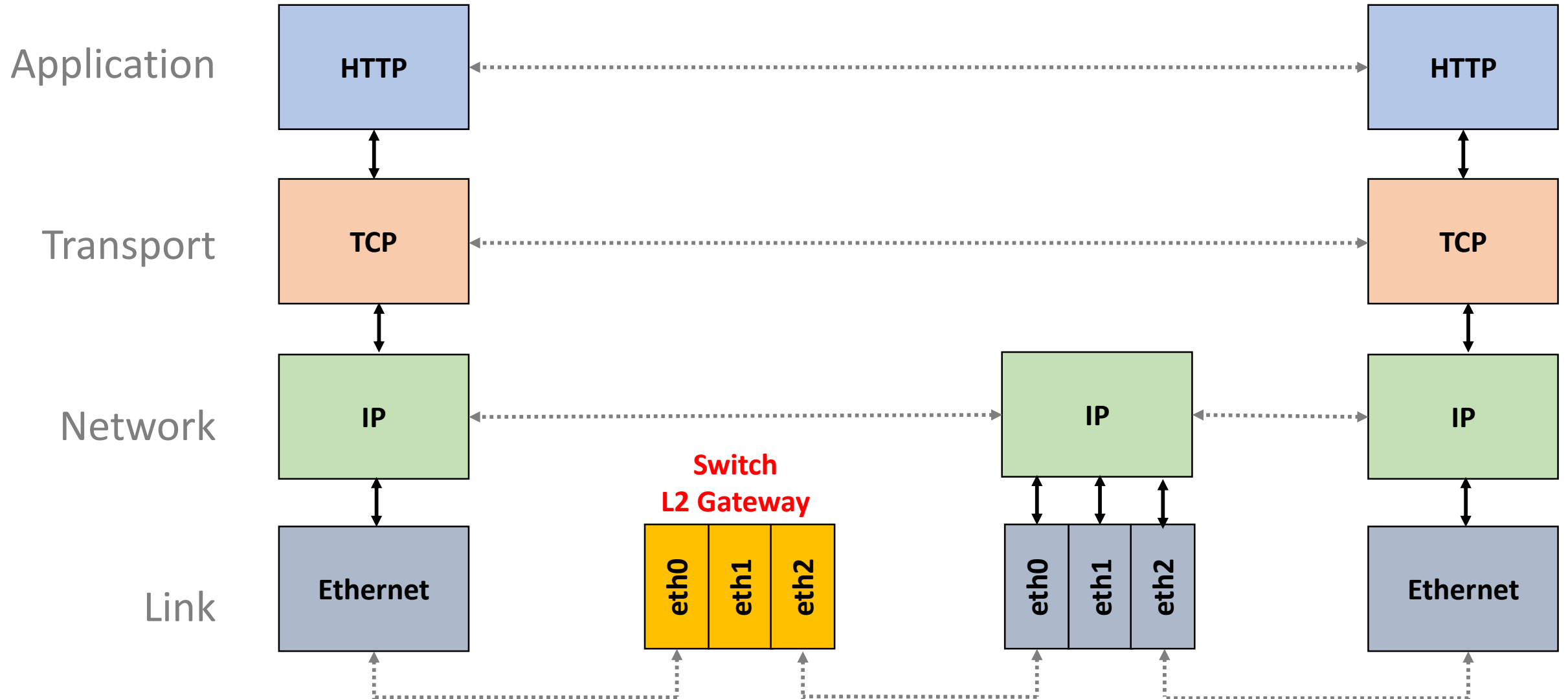


# Routers act as **L3 gateway** as such they implement L2 and L3



Switches act as L2 gateway

as such they only implement L2



# Overview

How do we characterize the network?



A network *connection* is characterized by its *delay*, *loss rate* and *throughput*



How long does it take for a packet to reach the destination

What fraction of packets sent to a destination are dropped?

At what rate is the destination receiving data from the source?

# Delay



Each packet suffers from several types of delays at *each node* along the path

**transmission** delay

+ **propagation** delay

+ **processing** delay

+ **queueing** delay

due to **link properties**

due to **traffic mix & switch internals**

---

= **total delay**

Each packet suffers from several types of delays  
at *each node* along the path

**transmission** delay

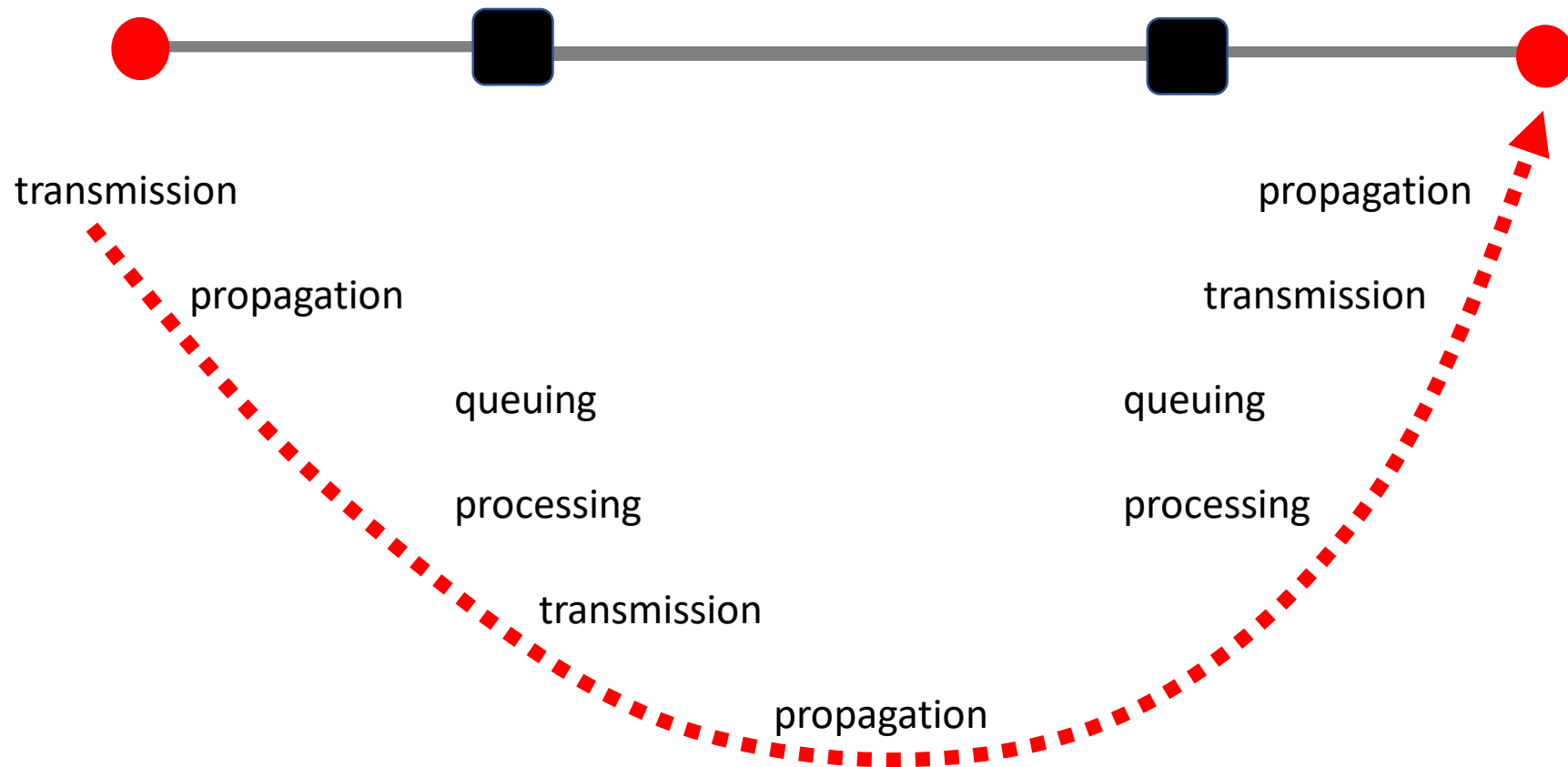
+ **propagation** delay

~~+ **processing** delay~~ **tend to be tiny**

+ **queueing** delay

---

= **total** delay



The transmission delay is the amount of time required to push all of the bits onto the link

$$\begin{array}{lcl} \text{Transmission delay} & = & \frac{\text{packet size}}{\text{link bandwidth}} \\ \text{[sec]} & & \begin{array}{l} \text{[#bits]} \\ \text{[#bits/sec]} \end{array} \end{array}$$

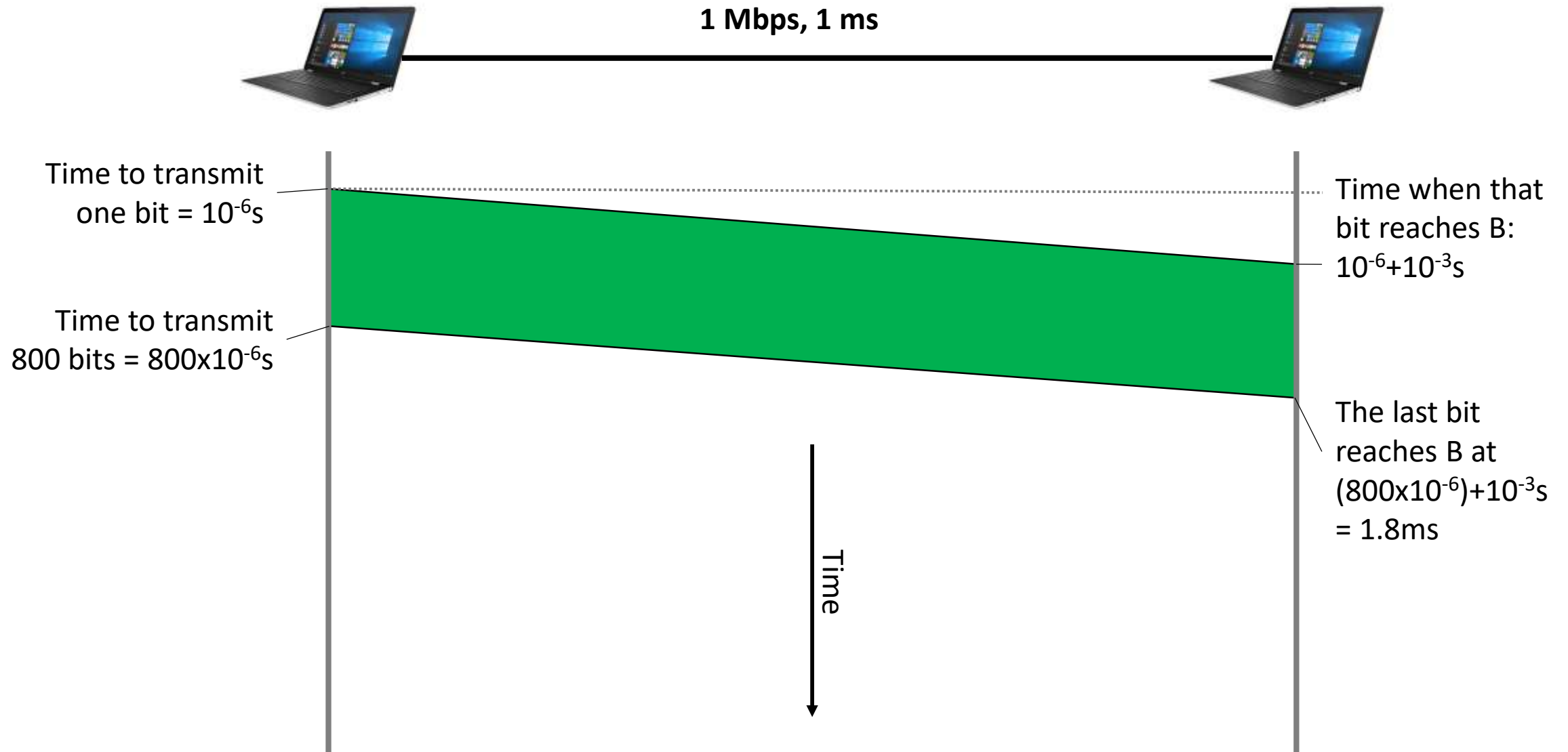
$$\begin{array}{lcl} \text{Example} & = & \frac{1000 \text{ bits}}{100 \text{ Mbps}} = 10 \mu\text{sec} \end{array}$$

The propagation delay is the amount of time required for a bit to travel to the end of the link

$$\text{Propagation delay} = \frac{\text{link length}}{\text{signal propagation speed}}$$

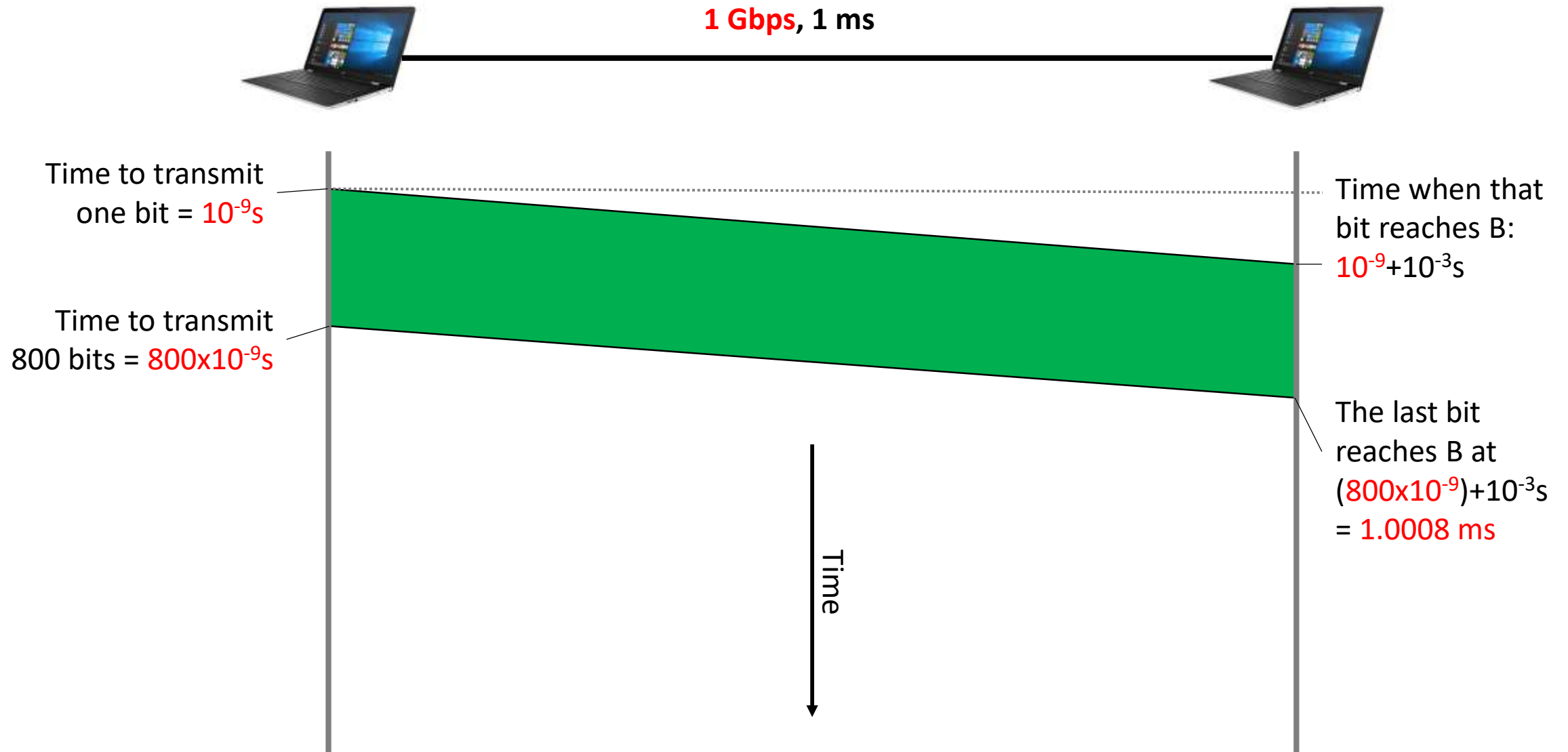
Example  $= \frac{30000 \text{ m}}{2 \times 10^8 \text{ m/sec (speed of light in fiber)}} = 150 \text{ } \mu\text{sec}$

# How long does it take to exchange 100 Bytes packet?





If we have a 1 Gbps link,  
the total time decreases to **1.0008 ms**



# If we now exchange a 1GB file split in 100B packets



**1 Gbps, 1 ms**



**$10^7 \times 100\text{B}$  packets**

Time when that  
bit reaches B:

**$10^{-9} + 10^{-3}\text{s}$**

The last bit  
reaches B at  
 **$(10^7 \times 800 \times 10^{-9}) + 10^{-3}\text{s}$**   
 **$= 8001\text{ ms}$**

Different transmission characteristics imply  
different tradeoffs in terms of which delay dominates

$10^7 \times 100\text{B}$ pkts	1Gbps link	<i>transmission delay dominates</i>
1x100B pkt	1Gbps link	<i>propagation delay dominates</i>
1x100B pkt	1Mbps link	<i>both matter</i>

**In the Internet, we cannot know in advance which one matter!**

The queuing delay is the amount of time a packet waits (in a buffer) to be transmitted on a link

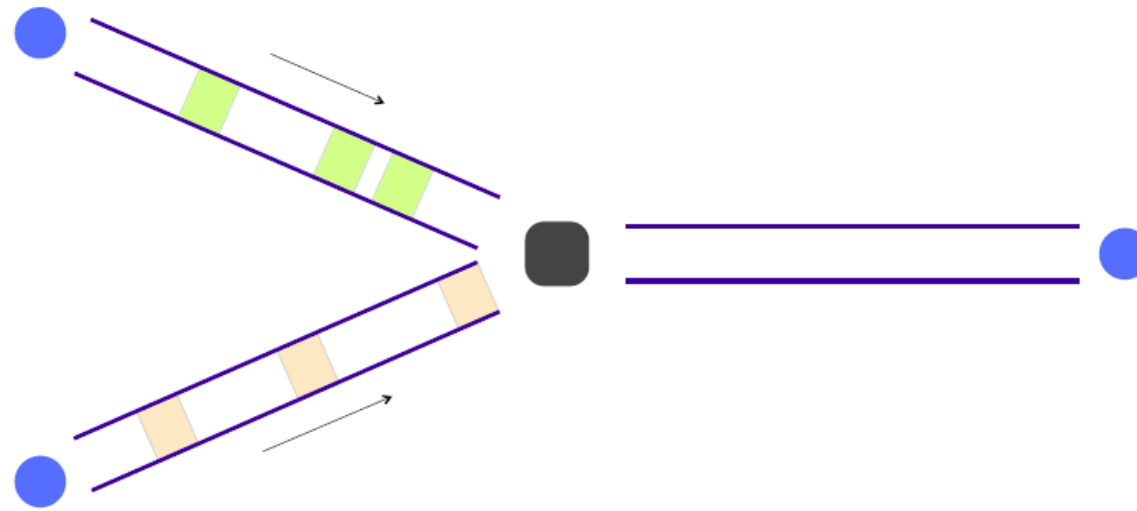
**Queuing delay is the hardest to evaluate**

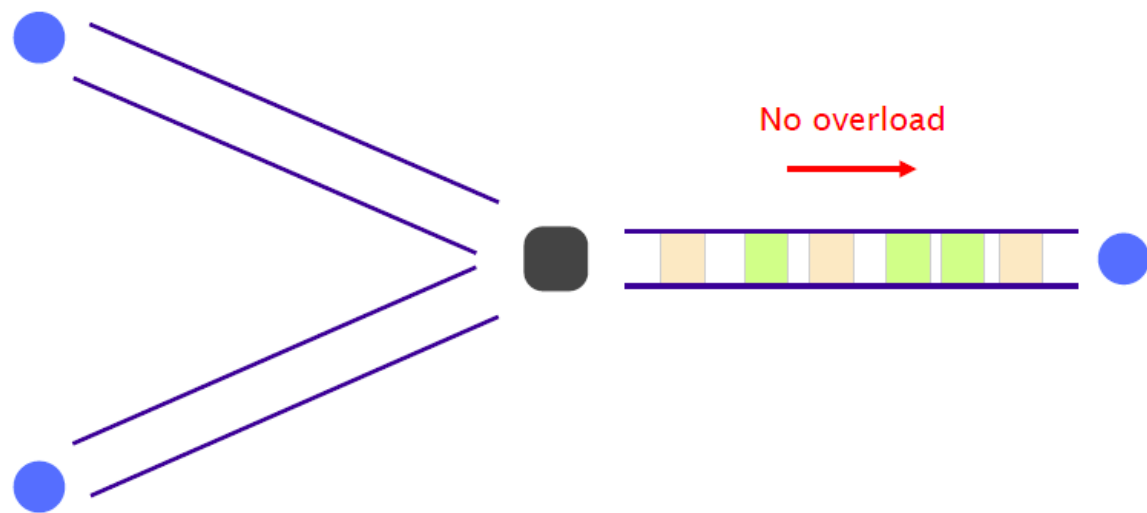
as it varies from packet to packet

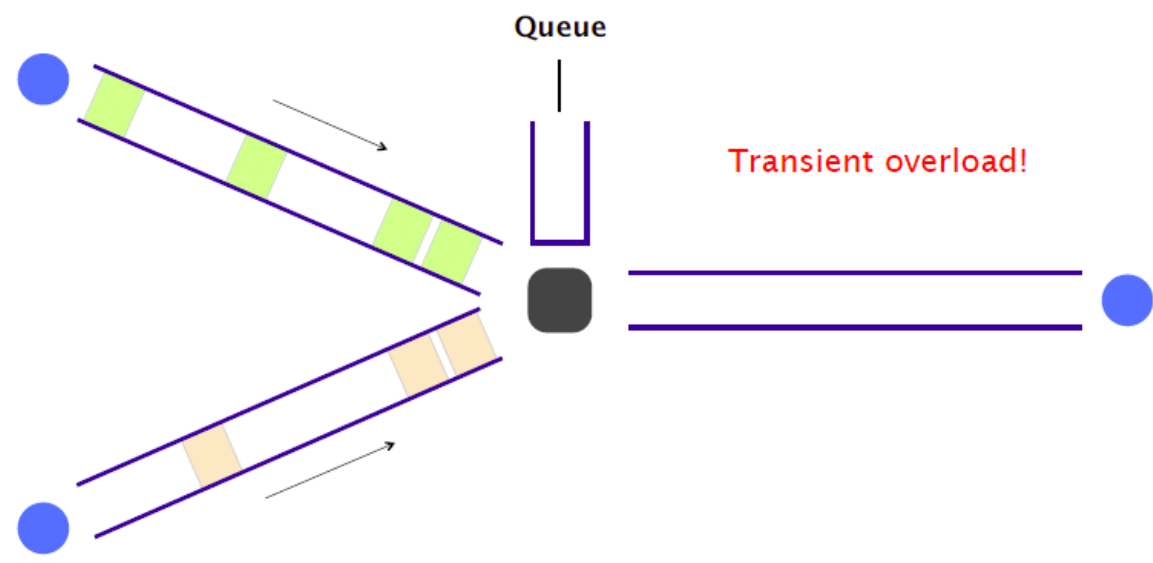
**It is characterized with statistical measures**

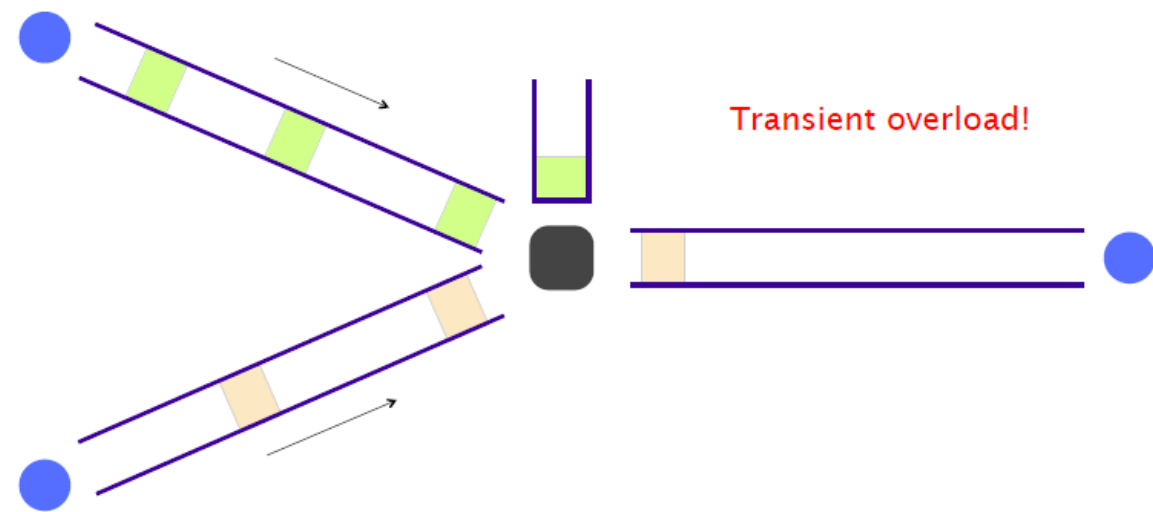
*e.g.*, average delay & variance, probability of exceeding  $x$

Queuing delay depends on the traffic pattern

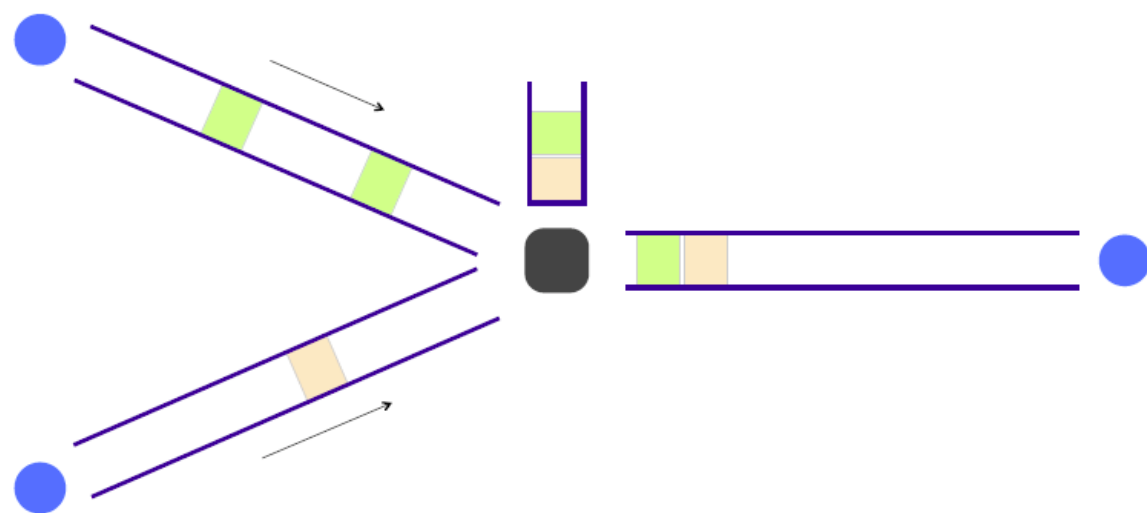


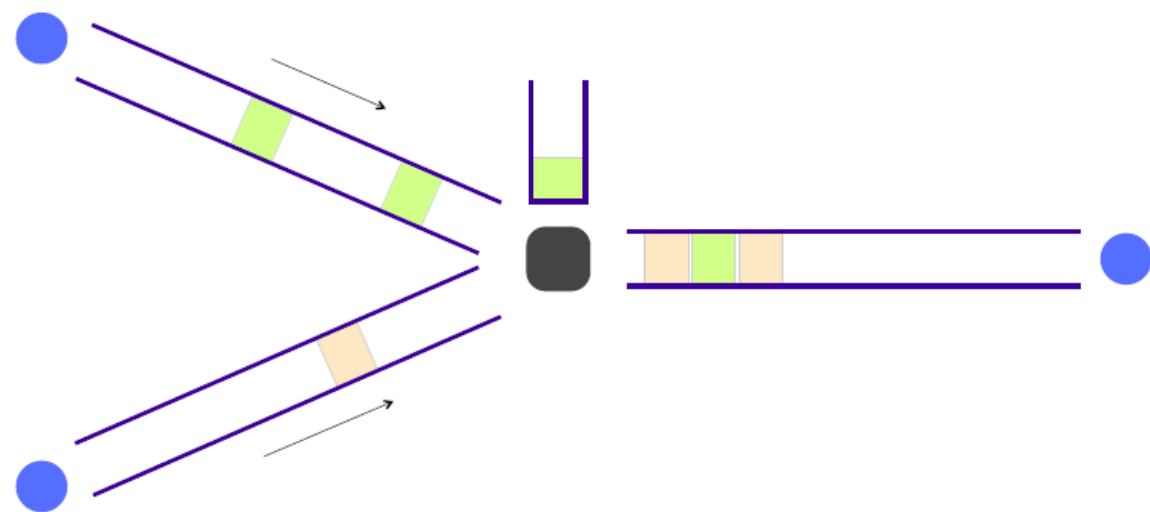


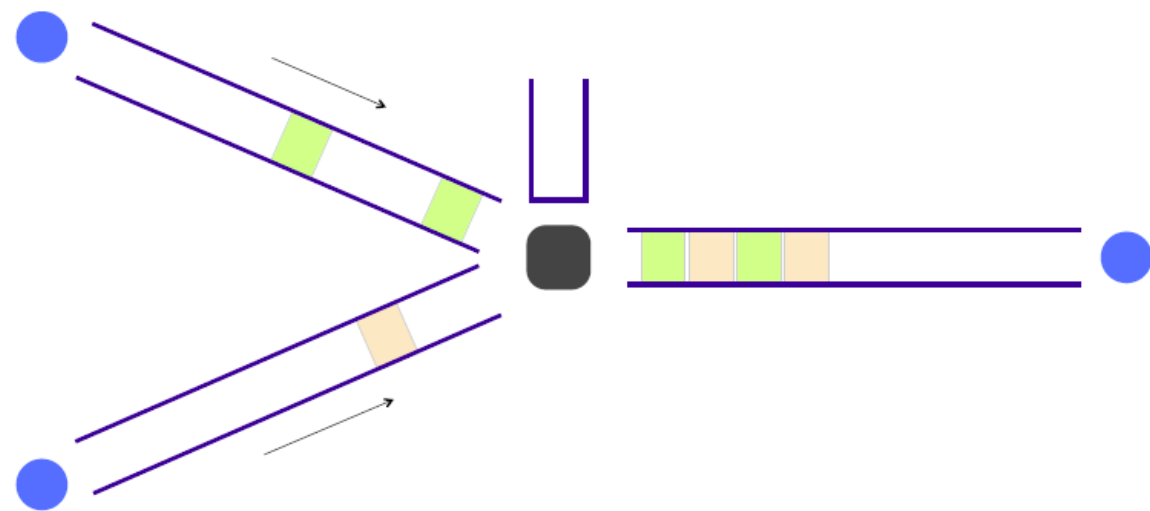




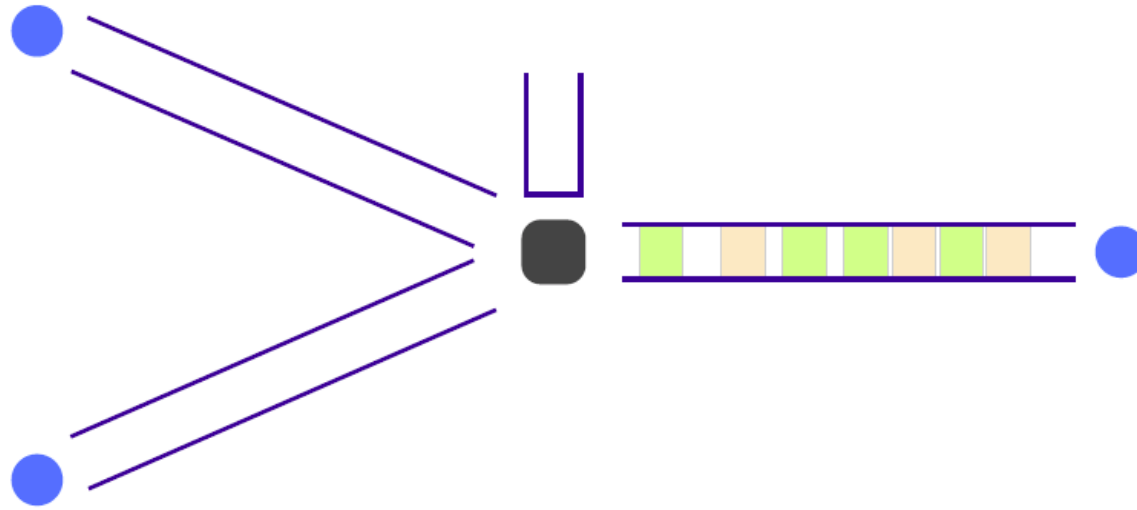








Queues absorb transient bursts,  
but introduce queueing delays



The time a packet has to sit in a buffer before being processed depends on the traffic pattern

Queueing delay depends on:

**arrival rate** at the queue

**transmission rate** of the outgoing link

traffic **burstiness**

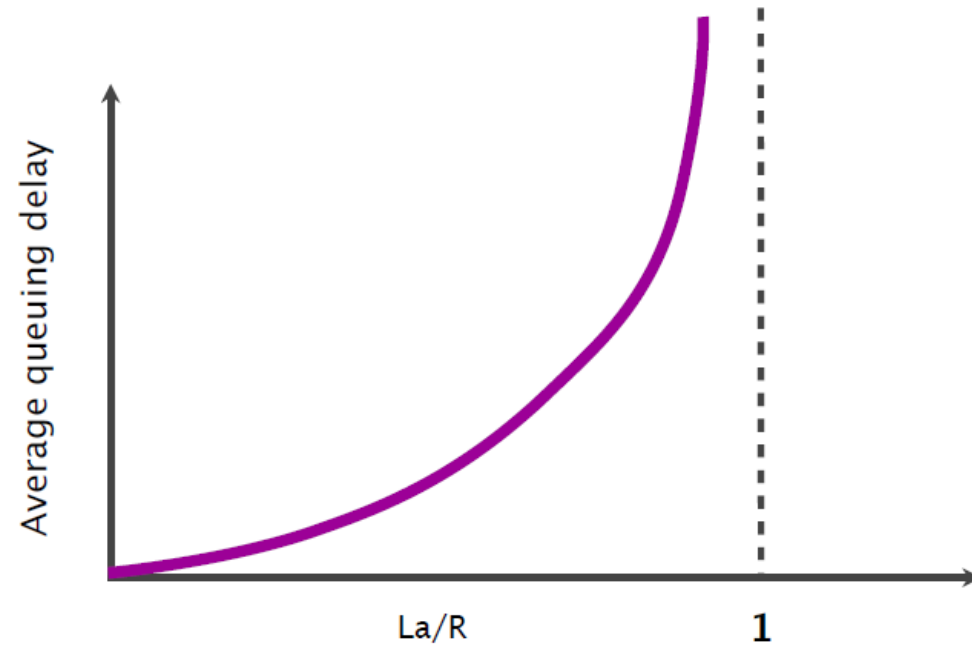
- average packet arrival rate  $a$  [packet/sec]
  - transmission rate of outgoing link  $R$  [bit/sec]
  - fixed packets length  $L$  [bit]
- 
- average bits arrival rate  $La$  [bit/sec]
  - traffic intensity  $La/R$

When the traffic intensity is  $>1$ , the queue will increase without bound, and so does the queuing delay

Golden rule

**Design your queuing system,  
so that it operates far from that point**

When the traffic intensity is  $\leq 1$ ,  
queueing delay depends on the burst size

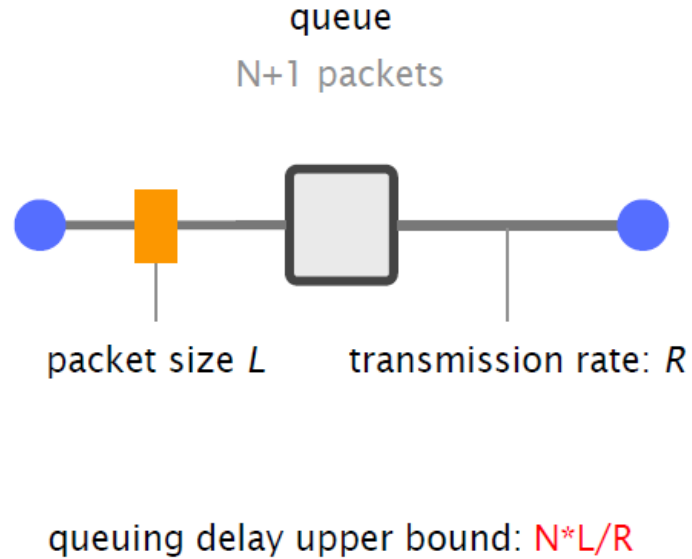




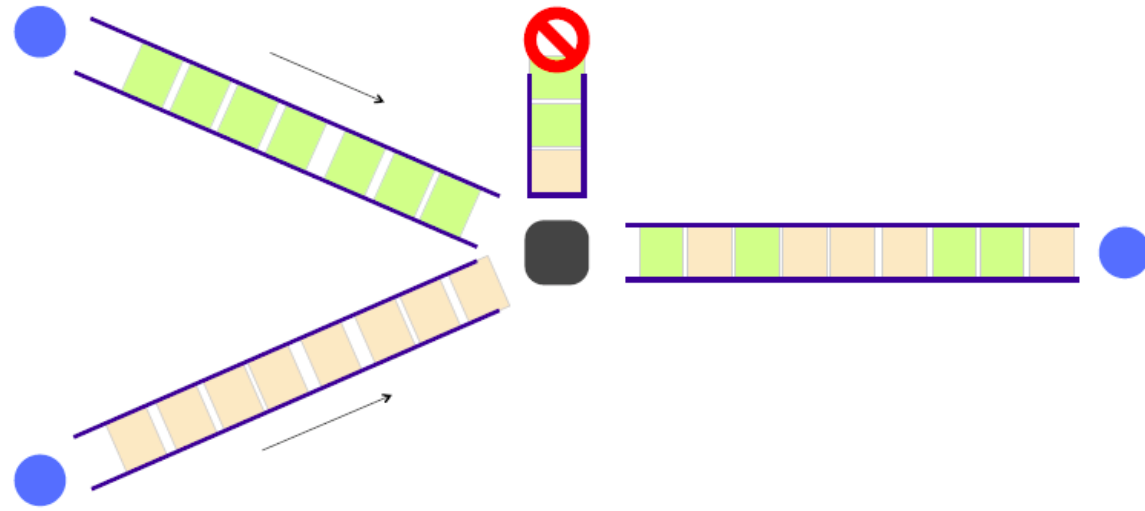
# Loss



In practice, queues are not infinite.  
There is an upper bound on queuing delay.



If the queue is persistently overloaded, it will eventually drop packets (loss)



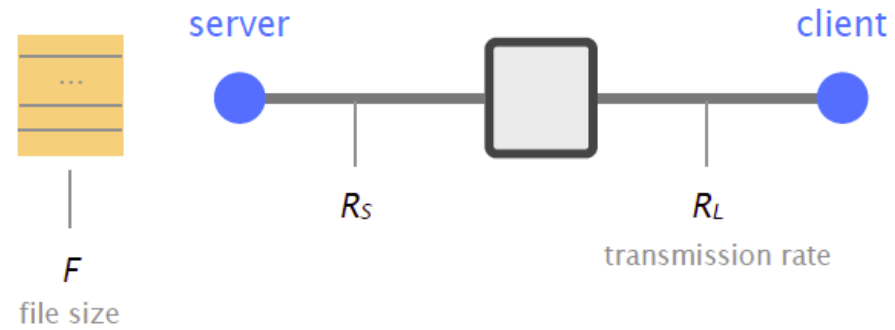
# Throughput



The throughput is the instantaneous rate at which a host receives data

$$\begin{array}{lcl} \text{Average throughput} & = & \frac{\text{data size}}{\text{transfer time}} \\ \text{[#bits/sec]} & & \begin{array}{l} \text{[#bits]} \\ \text{[sec]} \end{array} \end{array}$$

# To compute throughput, one has to consider the bottleneck link

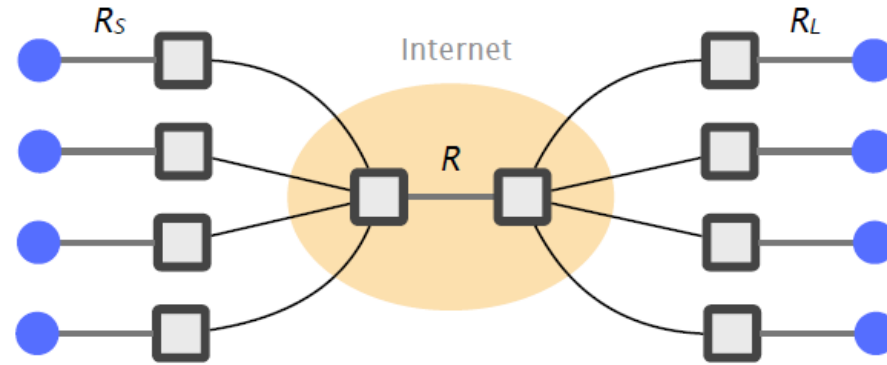


Average throughput

$$\min(R_S, R_L)$$

= transmission rate  
of the bottleneck link

To compute throughput, one has to consider the bottleneck link... **and the intervening traffic**



if  $4 * \min(R_s, R_L) > R$

the bottleneck is now in the core,  
providing each download  $R/4$  of throughput

As technology improves, throughput increase &  
delays are getting lower except for propagation  
(speed of light)



Because of propagation delays,  
Content Delivery Networks move content closer to you



\* <http://wwwnui.akamai.com/gnet/globe/index.html>

To be continued...