# Communication Networks

Prof. Laurent Vanbever

Communication Networks
Spring 2017

Laurent Vanbever
www.vanbever.eu

ETH Zürich (D-ITET)
February, 27 2017

Materials inspired from Scott Shenker & Jennifer Rexford

---

Last week on
Communication Networks

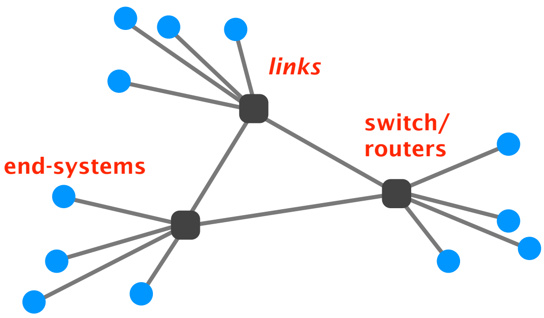
---

Communication Networks
Part 1: General overview

#1 What is a network made of?
How is it shared?
How is it organized?
How does communication happen?
How do we characterize it?

---

Networks are composed of three basic components




---

Communication Networks
Part 1: General overview

What is a network made of?
#2 How is it shared?
How is it organized?
How does communication happen?
How do we characterize it?

---

There exist two approaches to sharing:
reservation and on-demand

| | Reservation | On-demand |
|---|---|---|
| principle | reserve the bandwidth you need in advance | send data when you need |

---

In practice, the approaches are implemented using
circuit-switching or packet-switching

| | Reservation | On-demand |
|---|---|---|
| implem. | circuit-switching | packet-switching |

## Pros and cons of circuit switching

| advantages | disadvantages |
|---|---|
| predictable performance | inefficient if traffic is bursty or short |
| simple & fast switching<br>once circuit established | complex circuit setup/teardown<br>which adds delays to transfer |
| | requires new circuit upon failure |

## Pros and cons of packet switching

| advantages | disadvantages |
|---|---|
| efficient use of resources | unpredictable performance |
| simpler to implement<br>than circuit switching | requires buffer management and congestion control |
| route around trouble | |

## Communication Networks

Part 1: General overview

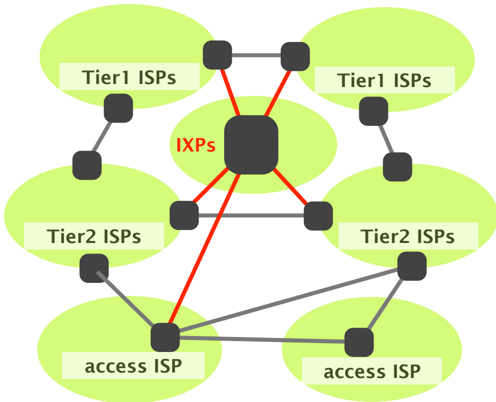
What is a network made of?

How is it shared?

#3 How is it organized?

How does communication happen?

How do we characterize it?




List any

technologies, principles, applications…

used after typing in:

> www.google.ch

and pressing enter in your browser

## You have a lot of networking knowledge already!

… and this, across all the layers

| | | |
|---|---|---|
| Transport | TCP | 72% |
| Application | DNS | 65% |
| Network | IP | 56% |
| Network | Router | 44% |
| Application | HTTP | 41% |
| Application | HTML | 30% |
| | Server | 26% |
| Link | Ethernet | 19% |
| Transport | UDP | 17% |
| | Encryption | 15% |

+72 other!

This week on

Communication Networks

## Communication Networks

Part 1: General overview

What is a network made of?

How is it shared?

How is it organized?

#4 How does communication happen?

#5 How do we characterize it?

## Communication Networks

Part 1: General overview

What is a network made of?

How is it shared?

How is it organized?

#4 How does communication happen?

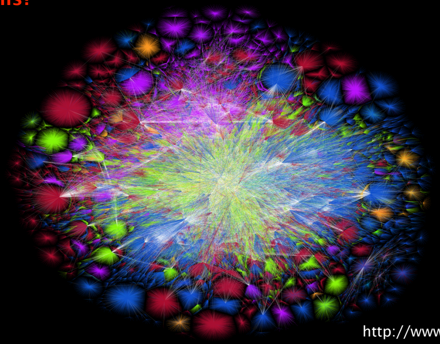
How do we characterize it?

The Internet should allow

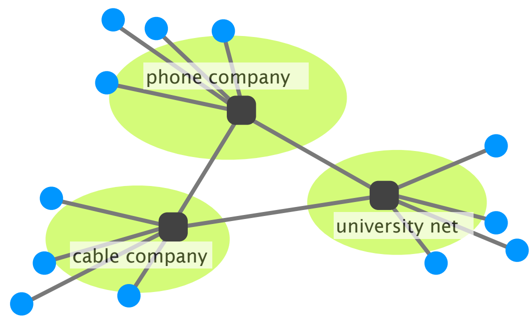
processes on different hosts
to exchange data

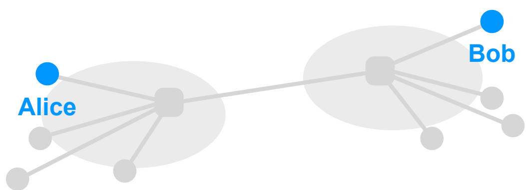
everything else is just commentary…

How do you exchange data in a network as complex as this?



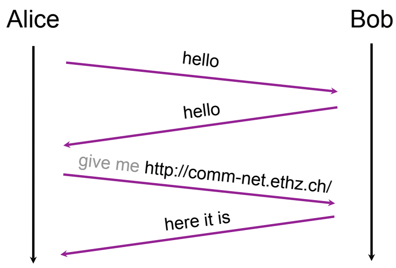
http://www.opte.org



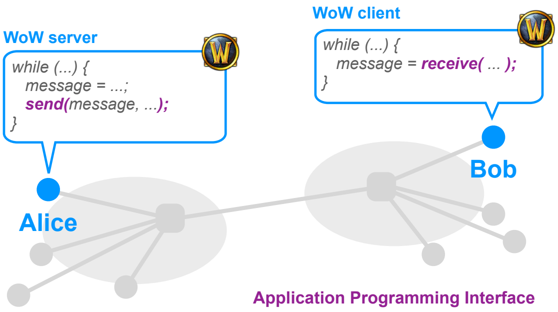

To exchange data, Alice and Bob use a set of network protocols

A protocol is like a conversational convention:
who should talk next and how they should respond

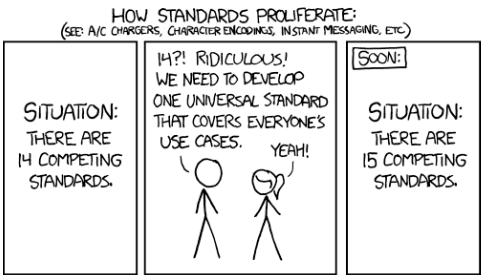



Sometimes implementations are not compliant…

Each protocol is governed
by a specific interface




In practice, there exists **a lot** of network protocols.
How does the Internet organize **this?**


https://xkcd.com/927/

Modularity is a key component
of any good system

| | |
|---|---|
| Problem | can't build large systems out of spaghetti code<br>hard (if not, impossible) to understand, debug, update |
| | need to bound the scope of changes<br>evolve the system without rewriting it from scratch |
| Solution | Modularity is how we do it<br>...and understand the system at a higher-level |

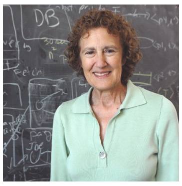


Photo: Donna Coveney

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

Internet communication can be decomposed
in 5 independent layers (or 7 layers for the OSI model)

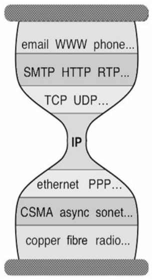
| | layer |
|---|---|
| L5 | Application |
| L4 | Transport |
| L3 | Network |
| L2 | Link |
| L1 | Physical |

Each layer provides a service to the layer above

| | layer | service provided: |
|---|---|---|
| L5 | Application | network access |
| L4 | Transport | end-to-end delivery (reliable or not) |
| L3 | Network | global best-effort delivery |
| L2 | Link | local best-effort delivery |
| L1 | Physical | physical transfer of bits |

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

**Applications**
…built on…
**Reliable (or unreliable) transport**
…built on…
**Best-effort global packet delivery**
…built on…
**Best-effort local packet delivery**
…built on…
**Physical transfer of bits**




## Each layer has a unit of data

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

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

| | layer | protocol |
|---|---|---|
| L5 | Application | HTTP, SMTP, FTP, SIP, … |
| L4 | Transport | TCP, UDP, SCTP |
| L3 | Network | IP |
| L2 | Link | Ethernet, Wifi, (A/V)DSL, WiMAX, LTE, … |
| L1 | Physical | Twisted pair, fiber, coaxial cable, … |

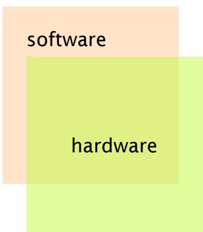
## The Internet Protocol (IP) acts as an unifying, network, layer

| | layer | protocol |
|---|---|---|
| L5 | Application | HTTP, SMTP, FTP, SIP, … |
| L4 | Transport | TCP, UDP, SCTP |
| L3 | Network | IP |
| L2 | Link | Ethernet, Wifi, (A/V)DSL, Cable, LTE, … |
| L1 | Physical | Twisted pair, fiber, coaxial cable, … |

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

| | layer |
|---|---|
| L5 | Application |
| L4 | Transport |
| L3 | Network |
| L2 | Link |
| L1 | Physical |

technology




redhat

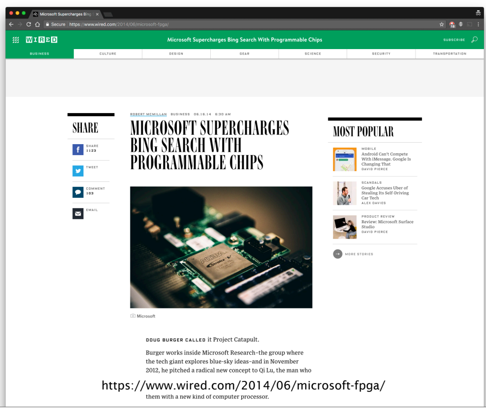
# Network stack challenges at increasing speeds

**The 100Gbit/s challenge**

**Jesper Dangaard Brouer**
**Red Hat inc.**

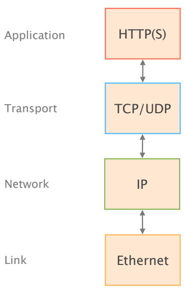
**Linux Conf Au, New Zealand, January 2015**

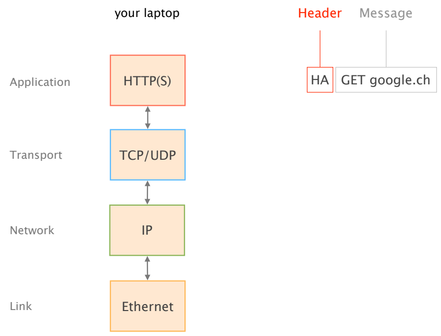
http://people.netfilter.org/hawk/presentations/LCA2015/net_stack_challenges_100G_LCA2015.pdf

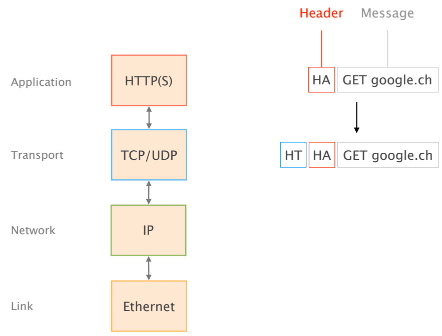


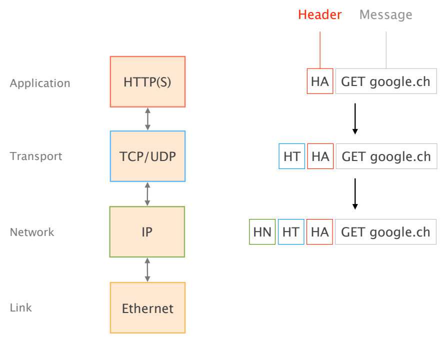

https://www.wired.com/2014/06/microsoft-fpga/

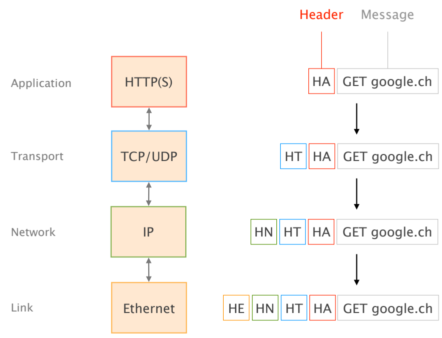
## Each layer takes messages from the layer above, and *encapsulates* with its own header and/or trailer



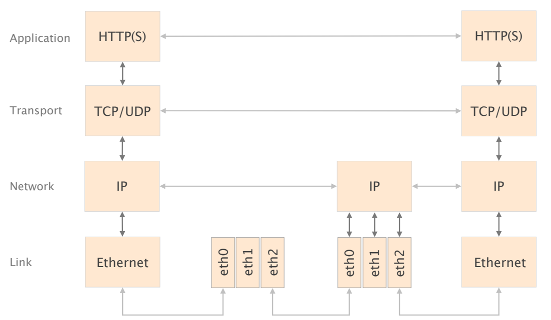


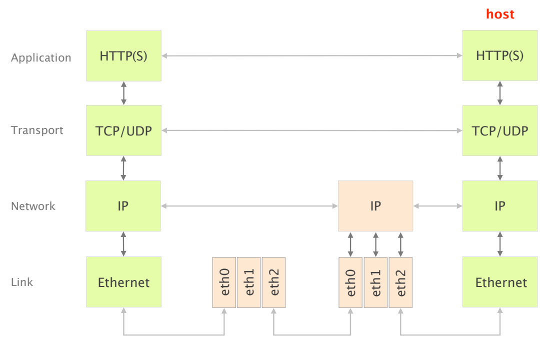



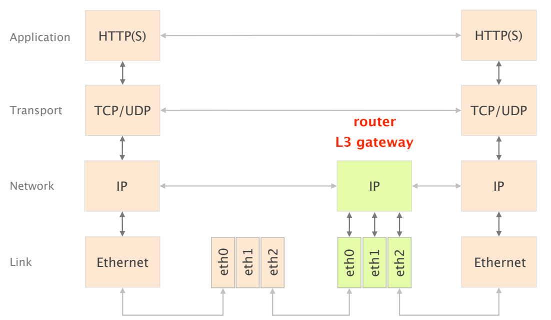

In practice, layers are distributed on every network device



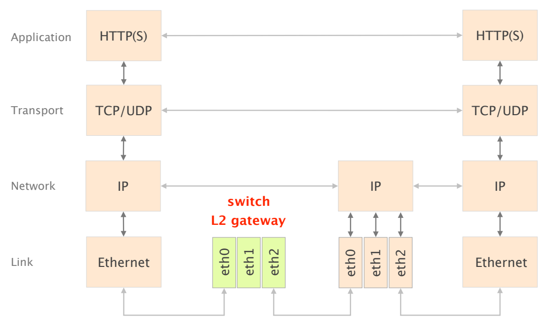

Since when bits arrive they must make it to
the application, all the layers exist on a host




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




Switches act as L2 gateway
as such they only implement L2



Let's see how it looks like in practice
on a host, using Wireshark

https://www.wireshark.org


Now, it's your turn

…to design a Internet protocol

instructions given in class

## Communication Networks

Part 1: General overview

What is a network made of?

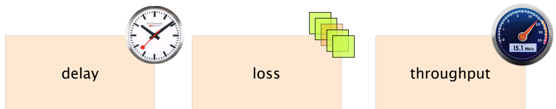
How is it shared?

How is it organized?

How does communication happen?

#5 How do we characterize it?

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

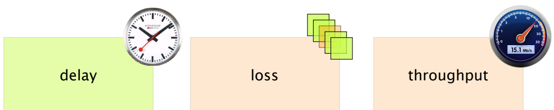



How long does it take for a packet to reach the destination

What fraction of packets sent to a destination are dropped?

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

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

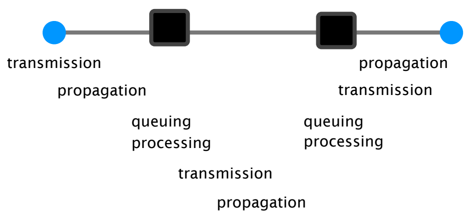



Each packet suffers from several types of delays
at *each node* along the path

transmission delay
⊕ propagation delay — due to link properties
⊕ processing delay
⊕ queuing delay — due to traffic mix & switch internals
= total delay

Overall, the main culprits for the overall delay are
the transmission, propagation and queuing delays

transmission delay
⊕ propagation delay
⊕ processing delay — *tend to be tiny*
⊕ queuing delay
= total delay



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

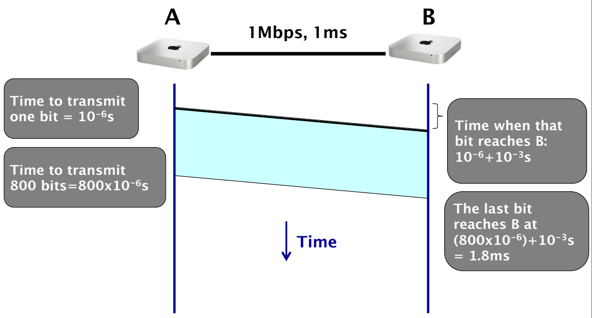
| | | | |
|---|---|---|---|
| Transmission delay [sec] | = | packet size / link bandwidth | [#bits] / [#bits/sec] |
| Example | | 1000 bits / 100 Mbps | 10 μsec |

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

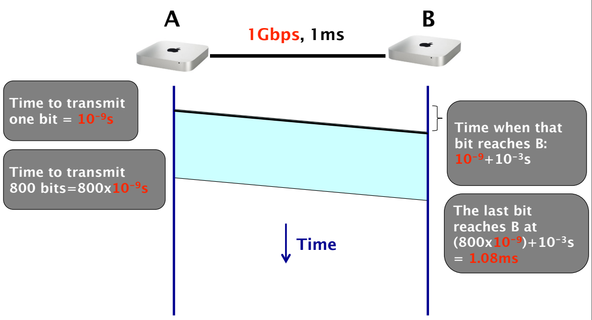
| | | | |
|---|---|---|---|
| Propagation delay [sec] | = | link length / propagation speed (fraction of speed of light) | [m] / [m/sec] |
| Example | | 30 000 m / $2.10^8$ m/sec (speed of light in fiber) | 150 μsec |

How long does it take for a packet to travel from A to B?
(not considering queuing for now)

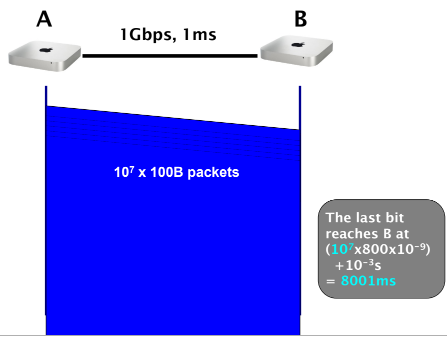
How long does it take to exchange 100 Bytes packet?




If we have a 1 Gbps link,
the total time decreases to 1.08ms




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




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

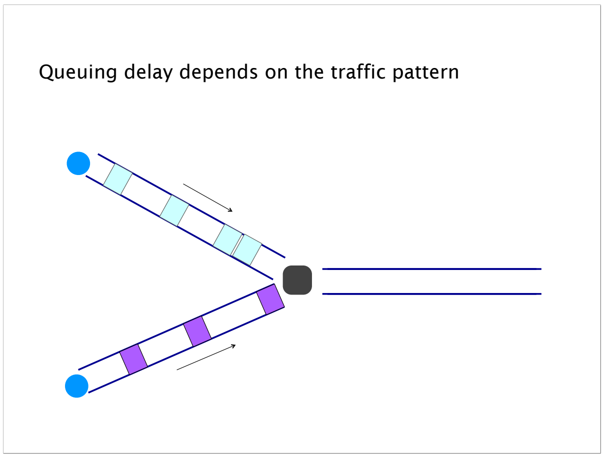
| | | |
|---|---|---|
| $10^7$x100B pkt | 1Gbps link | transmission delay dominates |
| 1x100B pkt | 1Gbps link | propagation delay dominates |
| 1x100B pkt | 1Mbps link | both matter |

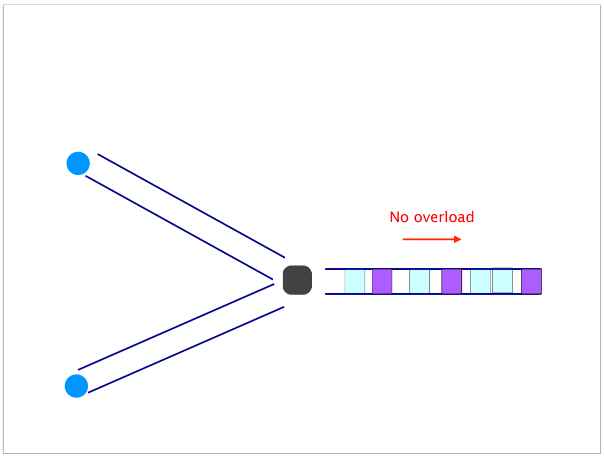
In the Internet, we can't know in advance which one matters!

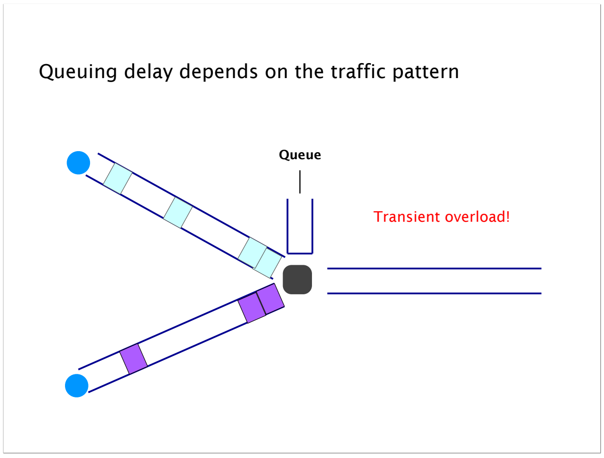
The queuing delay is the amount of time a packet waits (in a buffer) to be transmitted on a link

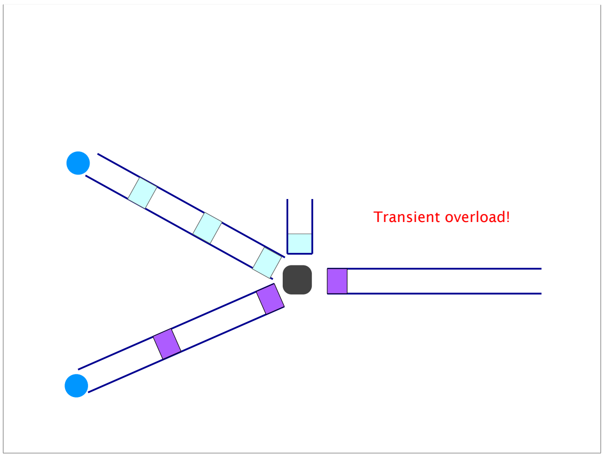
Queuing delay is the hardest to evaluate
as it varies from packet to packet

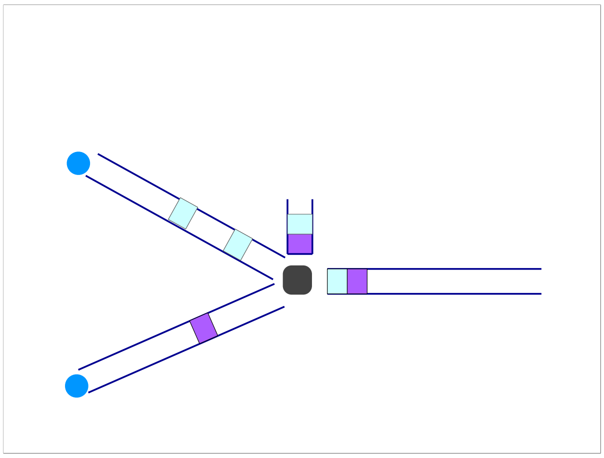
It is characterized with statistical measures
*e.g.*, average delay & variance, probability of exceeding *x*

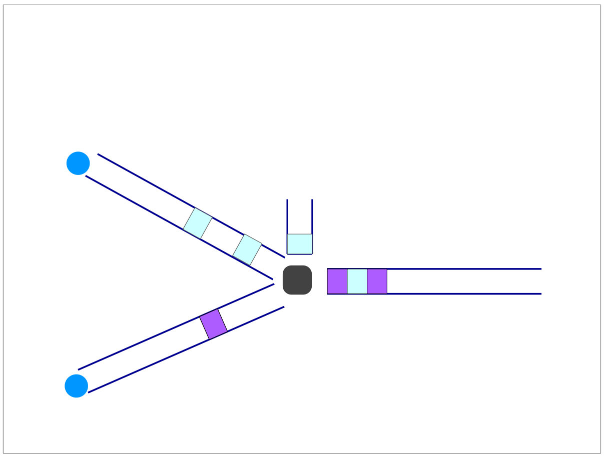

Queuing delay depends on the traffic pattern

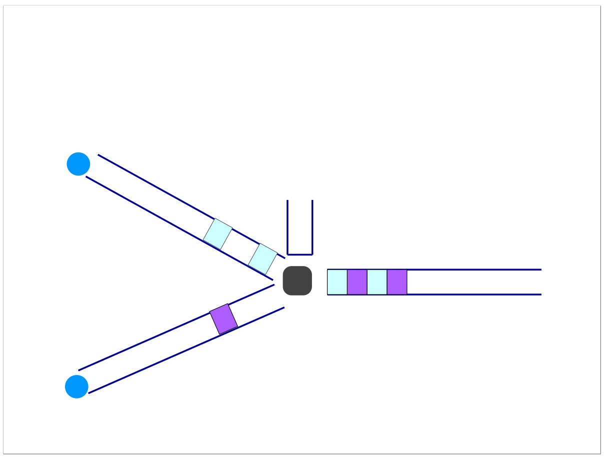

No overload

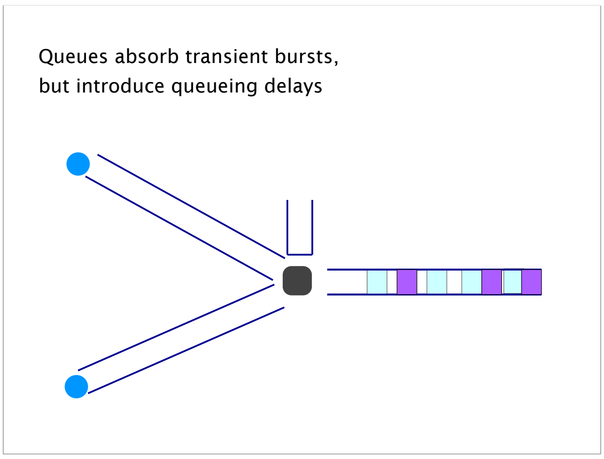

Queuing delay depends on the traffic pattern
Queue
Transient overload!


Transient overload!








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

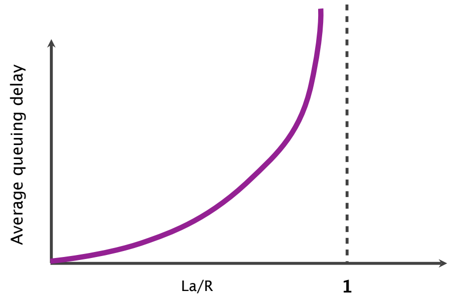
| | | |
|---|---|---|
| average packet arrival rate | *a* | [packet/sec] |
| transmission rate of outgoing link | *R* | [bit/sec] |
| fixed packets length | *L* | [bit] |
| average bits arrival rate | *La* | [bit/sec] |
| traffic intensity | *La/R* | |

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

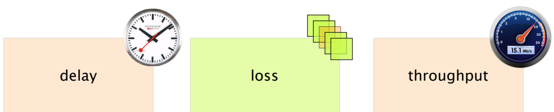
Golden rule

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

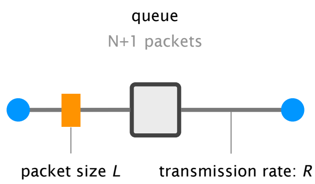
When the traffic intensity is <=1,
queueing delay depends on the burst size




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

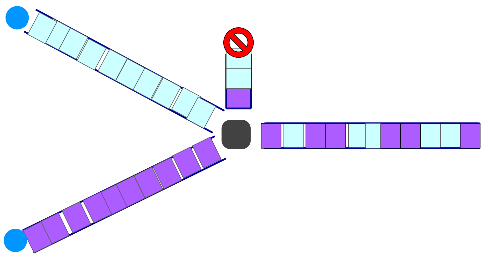



In practice, queues are not infinite.
There is an upper bound on queuing delay.

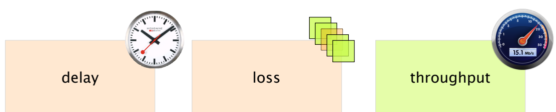



queuing delay upper bound: N*L/R

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



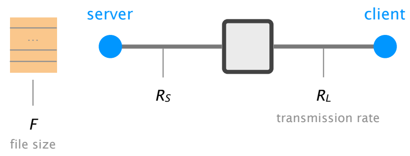
A network *connection* is characterized by its delay, loss rate and throughput



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

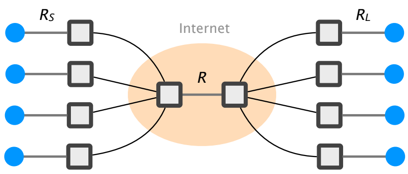
Average throughput [#bits/sec] = data size [#bits] / transfer time [sec]

To compute throughput, one has to consider the bottleneck link



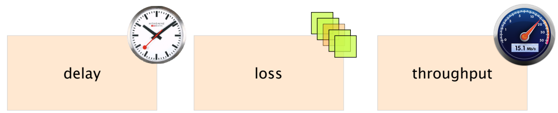

Average throughput min($R_S$, $R_L$)
= transmission rate of the bottleneck link

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



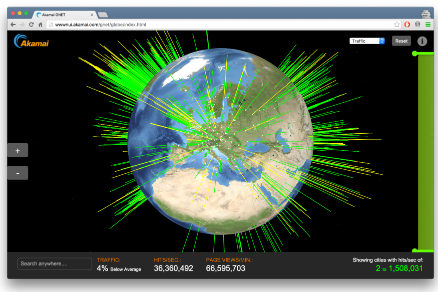

if 4*min($R_S$,$R_L$) > R the bottleneck is now in the core, providing each download R/4 of throughput

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

A brief overview of Internet history

The Internet history starts in the late 50's,
with people willing to communicate differently

Telephone network is *the* communication system
entirely based on circuit switching

People start to want to use networks for other things
defense, (not personal) computers, …

… but knew that circuit-switching will not make it
too inefficient for bursty loads and not resilient

## From this wish arose three crucial questions

Paul Baran
RAND

How can we design a more resilient network?
lead to the invention of packet switching

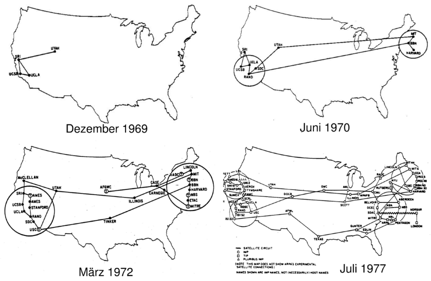
Len Kleinrock
UCLA

How can we design a more efficient network?
(also) lead to the invention of packet switching

Bob Kahn
DARPA

How can we connect all these networks together?
lead to the invention of the Internet as we know it

## The 60s saw the creation of packet switching and the Advanced Research Projects Agency Network




## The first message ever exchanged on the Internet was “lo”

Oct. 29 1969

Leonard Kleinrock @UCLA tries to log in a Stanford computer

UCLA — We typed the L… Do you see it?
*Yes! We see the L* — Stanford

We typed the O… Do you see it?
*Yes! We see the O*

We typed the G. system crashes

http://ftp.cs.ucla.edu/csd/first_words.html

## The 70s saw the creation of Ethernet, TCP/IP and the e-mail

| | |
|---|---|
| 1971 | Network Control Program<br>predecessor of TCP/IP |
| 1972 | Email & Telnet |
| 1973 | Ethernet |
| 1974 | TCP/IP<br>paper by Vint Cerf & Bob Kahn |

## In the 80s, TCP/IP went global

| | |
|---|---|
| 1983 | NCP to TCP/IP Flag day |
| 1983 | Domain Name Service (DNS) |
| 1985 | NSFNet (TCP/IP) succeeds to ARPANET |
| 198x | Internet meltdowns due to congestion |
| 1986 | Van Jabobson saves the Internet<br>(with congestion control) |

## The 90s saw the creation of the Web as well as the Internet going commercial

| | |
|---|---|
| 1989 | Arpanet is decommissioned |
| 1989 | Birth of the Web<br>Tim Berners Lee (CERN) — Swiss made |
| 1993 | Search engines invented (Excite) |
| 1995 | NSFNet is decommissioned |
| 1998 | Google reinvents search |

Next Monday on
Communication Networks

Routing!

## Communication Networks
Spring 2017

Laurent Vanbever
www.vanbever.eu

ETH Zürich (D-ITET)
February, 27 2017