Networking Fundamentals

François Fluckiger

CERN

CERN School of Computing 2019

Objectives of the Lecture Series

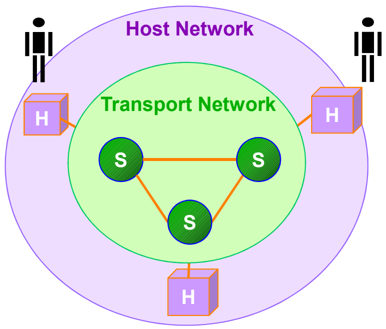
- Introduction to networking **fundamentals**

Parts

1. Transport Network: The Essentials
2. Focus on End-Systems

Over simplified Topological Model

- **Transport Network**
  - Formed of switches
- **Host Network***
  - Where the service is provided (client) or delivered (server)



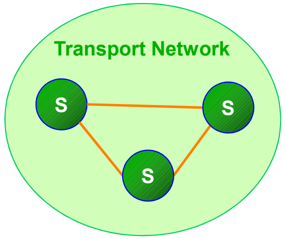
* *In the Internet, historical Name given to **End-Systems**, e.g. client or server computers*

Part 1

Transport Network: The essentials

Transport Network

- Multiple **Technologies**
  - Telephone, X.25, ISDN, ATM, IP
- Based on **Architectural Principles**



Transport Technology Architectures

Requirements

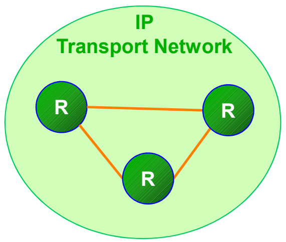
- Robustness / Survivability
- Scalability
- Predictability
- Performance Guarantees
- Per-Call Charging

Architecture

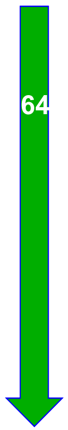
1. Centralization
2. Network Intelligence
3. States
4. Addressing
5. Traffic prioritization
6. Security Boundaries

IP Transport Network

- **Packet Switching Technology**
  - Like X.25, ATM



The Birth of Packet Swicthing



Paul Baran
(1926 – 2011)

Invent
Packet Switching

End of Tyranny of Calls

64 — Packet Switching

- Before Packet Switching
  All **Telecom Nets** (Telegraph, Telephone)
  based on **circuit switching**
- Can't do anything before placing a **call**
- Then, you're charged **per call**

ARPANET starting

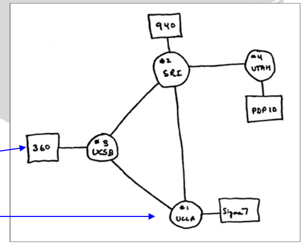
64 — Packet Switching

68 — DARPA contract for ARPANET

69 — First 4-node Network released

US DoD contract to
- UCLA
- Company BBN

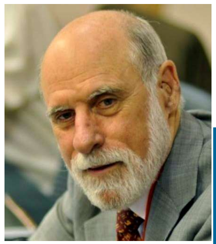
End-system:	**HOST**
Nodes:	**IMP**



Towards today's technology

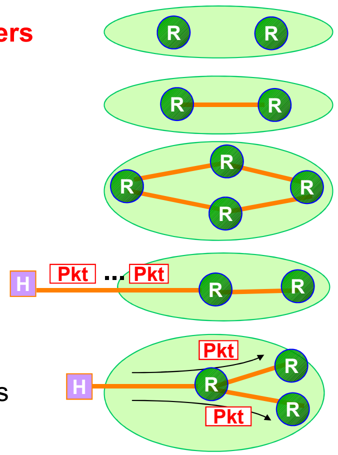
64 — Packet Switching

74 — **Bob Khan** and **Vint Cerf** Invent **TCP/IP**



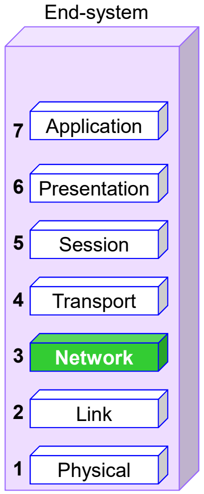
IP Networks: Back to Basics

- Formed of **switches** called **routers**
- Routers interconnected by **links**
- Topology usually **meshed**
- Hosts chop data stream into blocks called **packets**
- Routers switch individual packets



IP

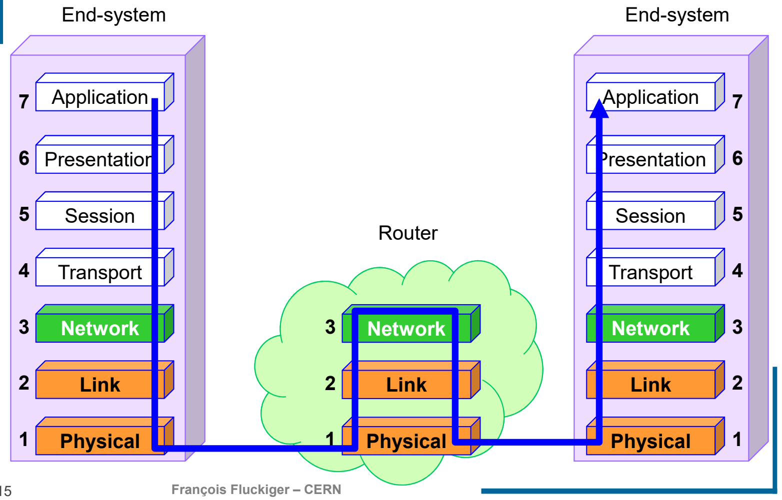
- **Layer 3 (Network) protocol**
- **Specifies**
  - Format
  - addressing
  - routing
- **The highest-level protocol understood by routers**



N-star slides

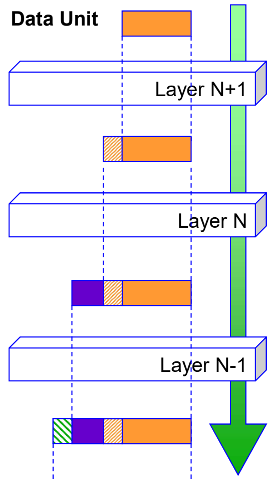
F as Fundamental

Protocols understood by routers



Layering principle (emission)

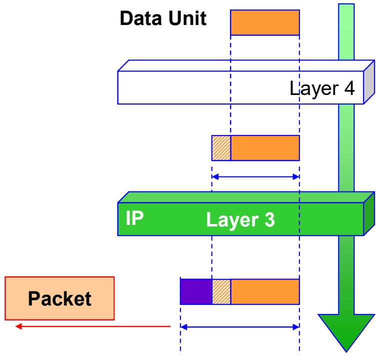
- Each protocol layer **N** adds a **Header** to the data unit received from layer **N+1** (1)
- Header contains **control** information; e.g. :
  - Numbering of the data unit
  - Coding of the destination
  - Codes for error detection
  - Priority of the data unit



(1) and segment the data unit if necessary

Packets

- Each **data unit** constructed at a given layer has a specific name
- Data Unit at layer 3 (IP) is called a **Packet**



Historical IP: Main Features

- **IP is a ConnectionLess (CL) protocol *(stateless)***
  - all packets routed independently
  - packets carry full destination address
  - packets may be lost, miss-ordered
  - **all packets had (initially) same priority**
- **Opposite = Connection-Oriented (CO) *(stateful)***
  - no information sent before a hard connection is set up

Stateful / Stateless Networks

Stateful	**Stateless**
Telephone	Post office Road Network
“λ on-demand” ISDN ATM *Frame Relay* *SNA* *X.25*	IP Ethernet

Stateful vs Stateless

\+

Stateful	**Stateless**
Traffic more predictable	**No call set-up delay before sending a packet**
Easier for network to reserve resources	**Routing possibly more dynamic**
QoS easier to guarantee	**Higher Resilience**

IP, HTTP Stateless Regular Behavior

- **IP router**
  - take a packet, forward it, forget it …
  - take a packet, forward it, forget it …
- **HTTP server**
  - take a request, serve it, forget it …
  - take a request, serve it, forget it …

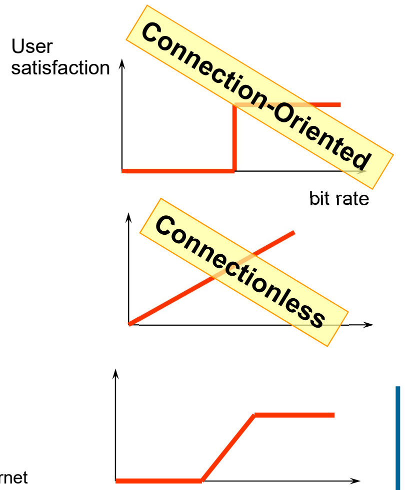
Predicting Load?

Types of Applications

- ***Constant Bit Rate (CBR)***
  - **Traditional real-time applications -** e.g. PABXs
- ***Available Bit Rate (ABR)***
  - **Traditional bulk data applications -** e.g. file transfer
- ***Variable Bit Rate (VBR)***
  - **Modern real-time applications -** e.g. compressed audio, video

Quality of Service and bit rate

- **CBR applications**
- **ABR applications**
- **VBR applications**



From Scott Shenker,
Fundamental Design Issues for the Future Internet

IP Architecture (in 84)

Requirements

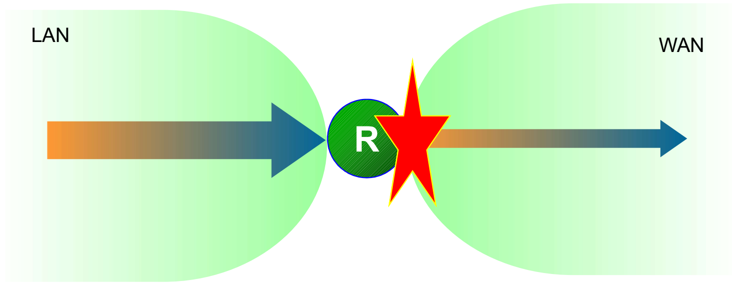
- Robustness / Survivability — **Maximum**
- Scalability — **Good**
- Predictability — **Fair**
- Performance Guarantees — **None**
- Per-call charging — **None**

Architecture

1. Centralization — **None**
2. Network Intelligence — **Minimum**
3. States — **Stateless**
4. Addressing — **Fixed size, numerical**
5. Traffic prioritization — **None**
6. Security Boundaries — **Within end-systems**

Why congestions?

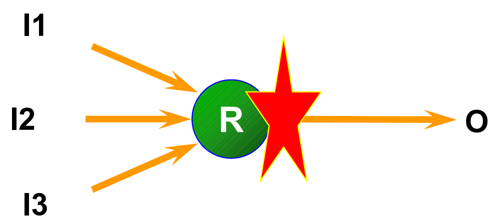
Fast / Slower junctions



e.g. **LAN – WAN Boundary**

Why congestions?

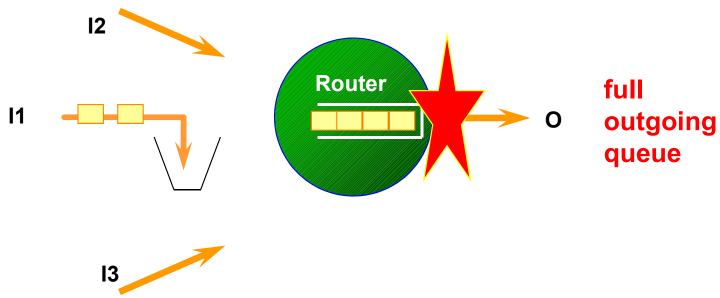
The N-to-1 problem



Σ Input traffic > Output capacity

Why congestions?

Lack of resources (generally buffers) in routers



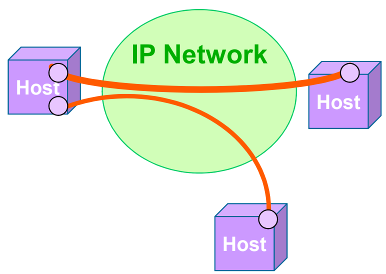
Initial IP : shortcomings

- Support of **real-time** applications

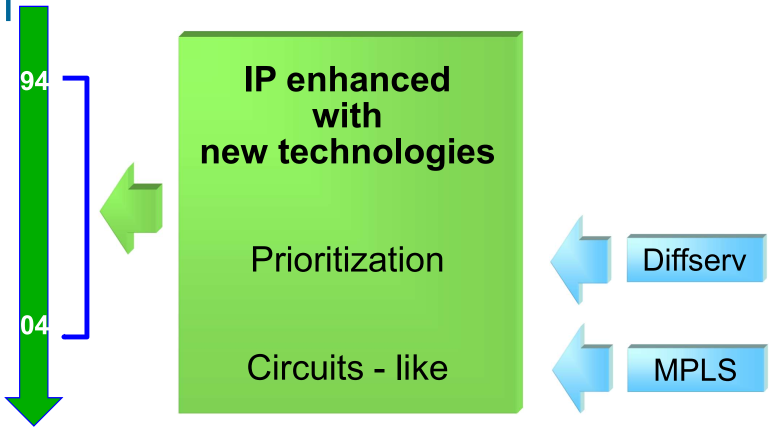
  e.g.

  - Internet Telephony
  - Real-Time video conferencing

Per-flow guarantees

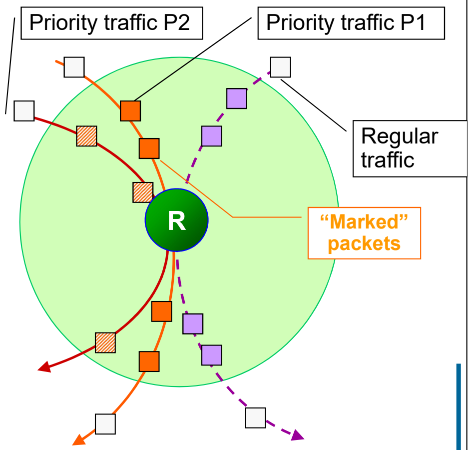


Turning IP into a Universal Technology



Diffserv Principle

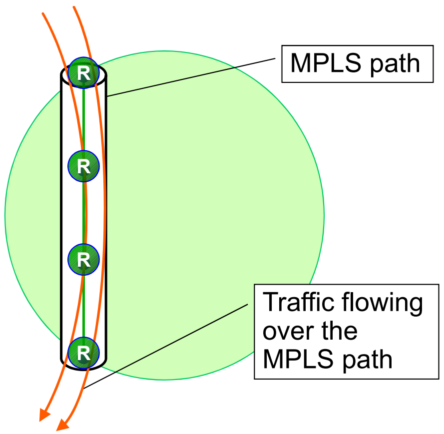
- **Priority Mark inserted** before Packets enter the "QoS core"
- Simple examination of mark provides priority



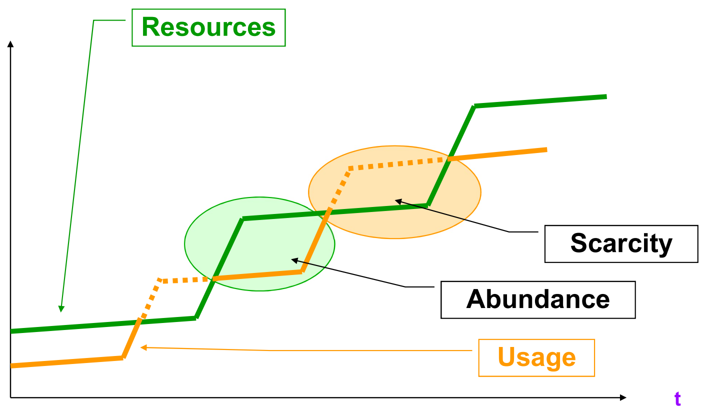
MPLS (MultiProtocol Label Switching)

- **Create a "circuit"** (called **MPLS path**)
- **Force all traffic with**
  - same destination
  - same Qos requirement

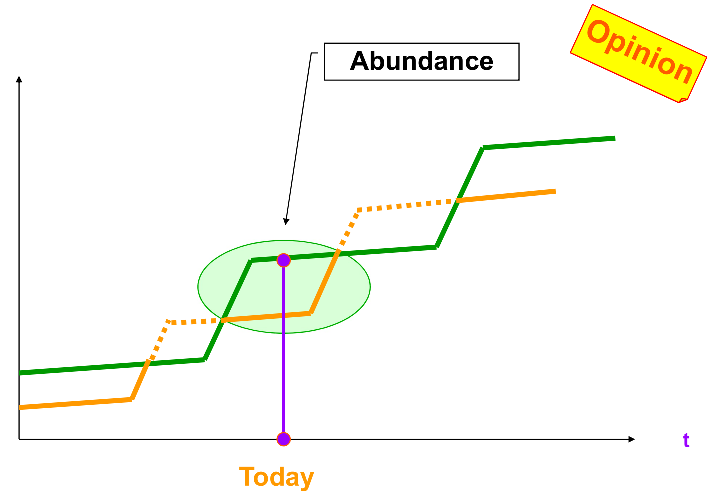
to flow over the path



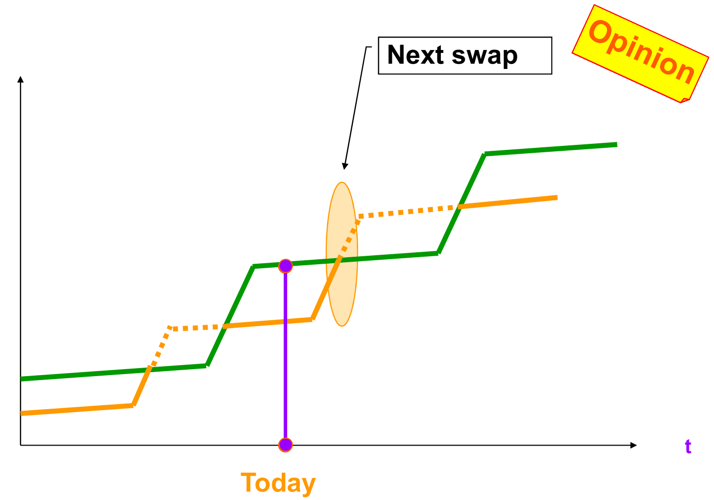
Resources and Usage



Resources and usage

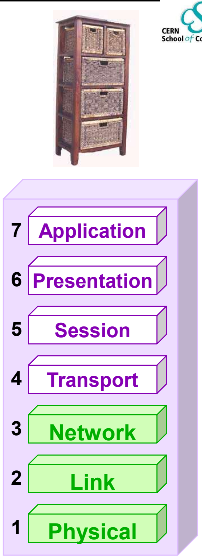


Resources and usage

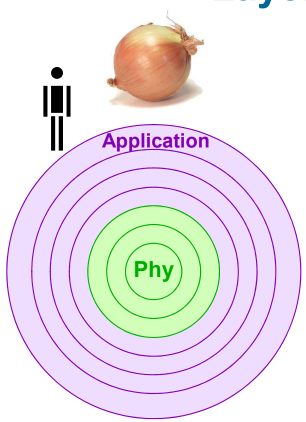


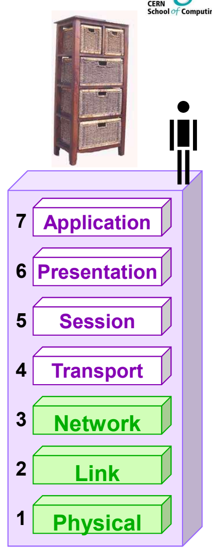
Metaphors

- Functions clustered into layers
- Each layer has only two neighbors
- Changing one layer => **changing two interfaces only**



Layering principle

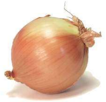


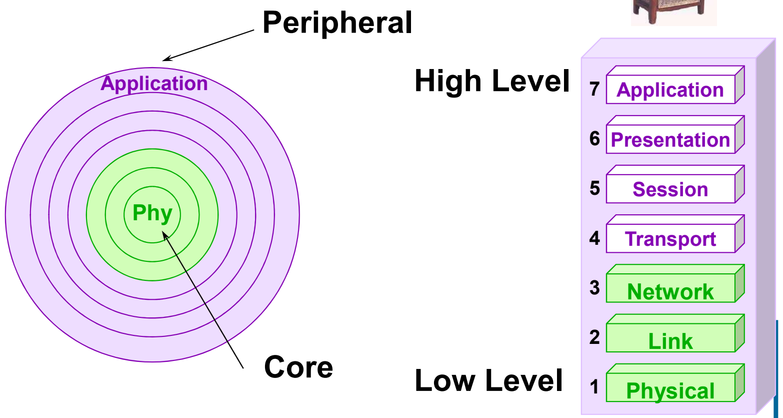


Graphical Presentation from Norway, 1976

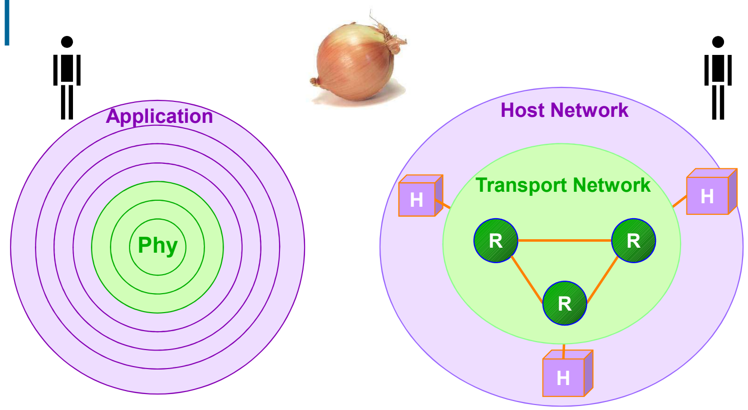
Predating OSI

Technologies





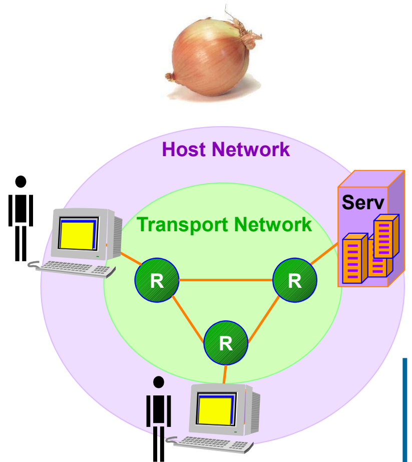
Topological Representations



The Dangers of the Host/Node Divide

1. Overlooking the **Client /Server** distinction

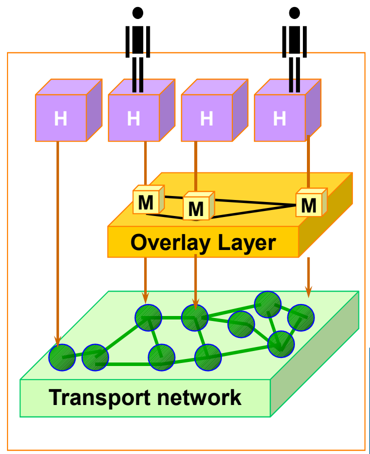
2. Neglecting the role of **Overlay Layers**



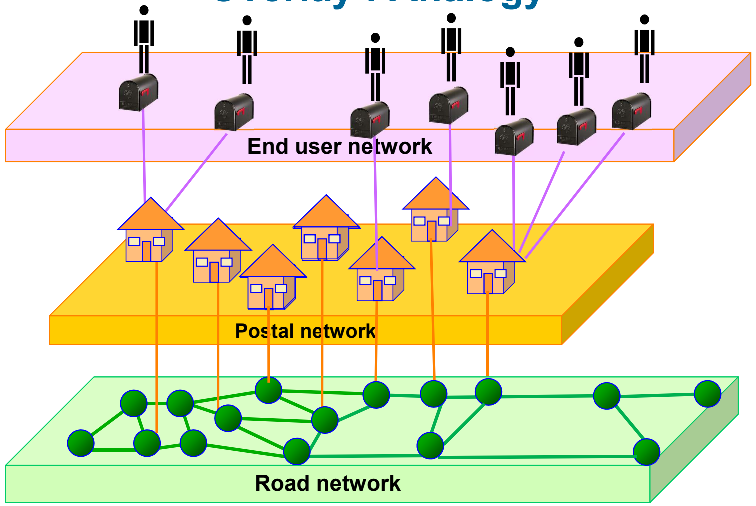
What is Overlay Layer?

A set of **intermediary** systems

- **invisible** to the **end-user**
- **on-top** of transport network
- which all **conspire** to deliver a specific service
- forming a **topology**
- essential but … **not compulsory**

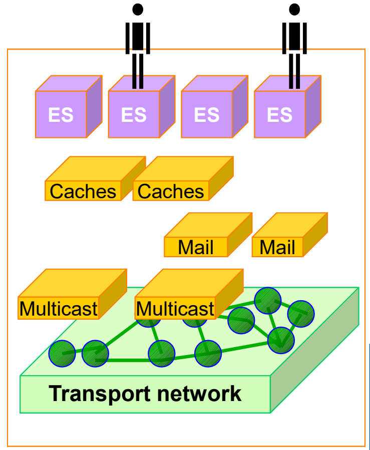


Overlay : Analogy



Examples of Overlay Layers

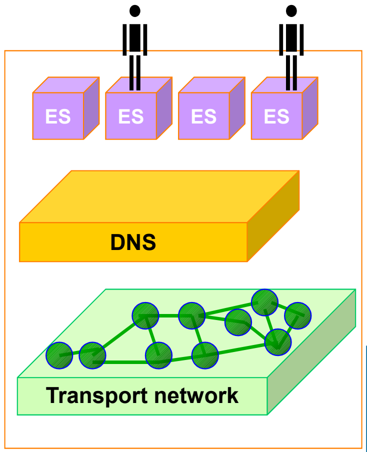
- IP multicast
- Email Relays
- Web caches



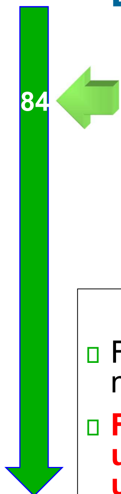
The Most Successful Overlay

The DNS

- Universal
- Invisible
- Well managed
- Unchallenged



End of Tyranny of Numbers



Paul Mockapetris

invents the DNS

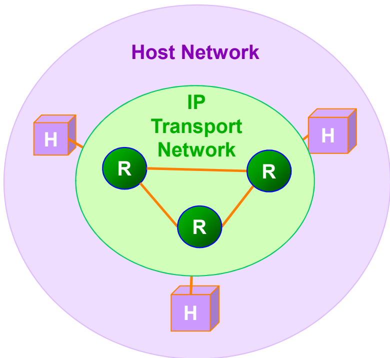
University of Southern California

DNS: Domain Name Server

- Fully distributed mechanism to translate names into (IP) addresses
- **First ever global naming system to uniquely identify any "object" in the universe**

Summing up : IP Philosophy

- **Dumb Routers**
  - Congestion risks
  - No flow control
  - No security check
- **Clever Hosts** *
  - Loss Detection/Recovery
  - Flow control
  - Security check



** Historical Name given to End-Systems, e.g. client or server computers*

End of

Part 1

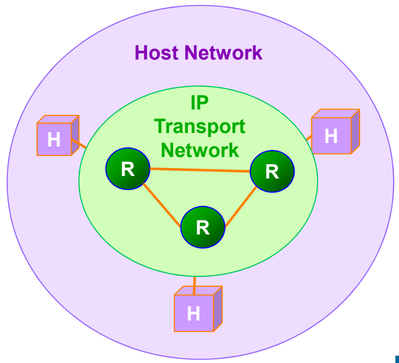
Transport Network: The Essentials

Part 2

Focus on End-Systems

IP Philosophy

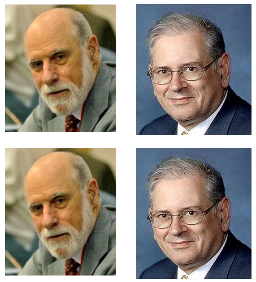
- **Dumb Routers**
  - Congestion risks
  - No flow control
  - No security check
- **Clever Hosts** *
  - Loss Detection/Recovery
  - Flow control
  - Security check



** Historical Name given to End-Systems, e.g. client or server computers*

Towards today's technology

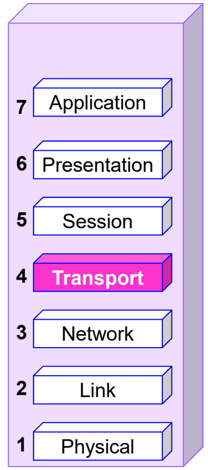
- 71 — Arpanet adopts NCP*
- 74 — TCP/IP Invented
- 78 — TCP/IP Split
- 83 — Arpanet converts to TCP/IP



* NCP : Network Control Protocol

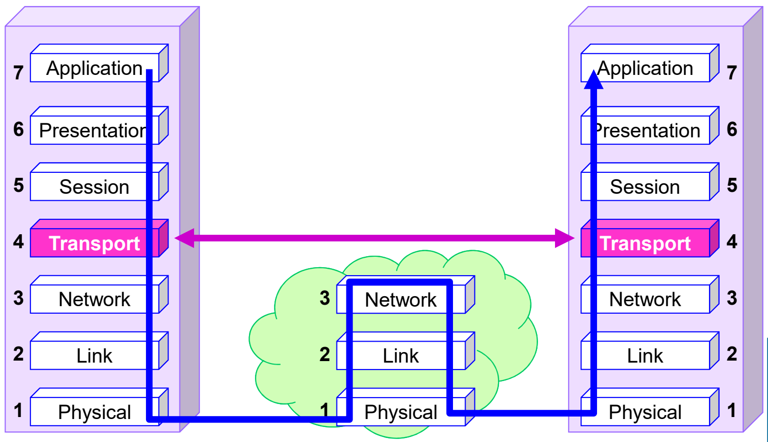
TCP*

- **Layer 4** (Transport) protocol
- Specifies
  - Mechanisms for flow control, error detection, error recovery
- The lowest protocol used exclusively by Hosts



* *Transport Control Protocol*

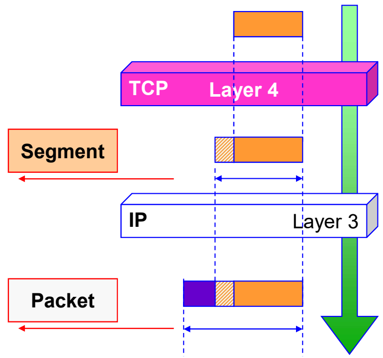
Protocols understood by hosts only



TCP Segments

- Each **data unit** generated at a given layer has a specific name
- Data Unit at layer 4 (TCP) is called a **Segment**

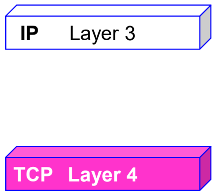
  (sometimes also called **block)**
- Data Unit at layer 3 (IP) is called a **Packet**)



Is “TCP/IP” Connectionless or Connection-Oriented?

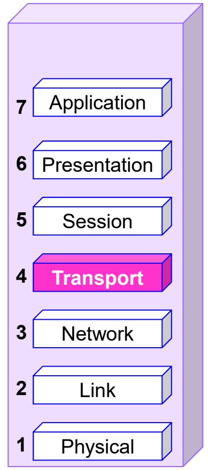
It is “Both” !

- **IP is connectionless**
  - Network permission **not** needed to transmit data
- **TCP is connection-oriented**
  - **No** data transmitted before **Remote Host permission**



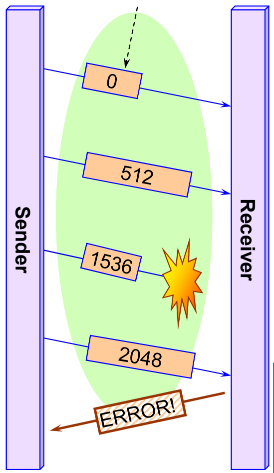
What does TCP Provide

1. **Connection service** (Hand check)
2. **Error services**
   - 2.1 Detection of loss, duplicated, out of sequence packets
   - 2.2 Correction of errors
3. **Flow control** between receiver/server



2.1 Error Detection

- Segments carry a **sequence number**
- Receiver can **detect** out-of-sequence segments (i.e. missing or duplicated)
- … and **Notify the Error to the sender**

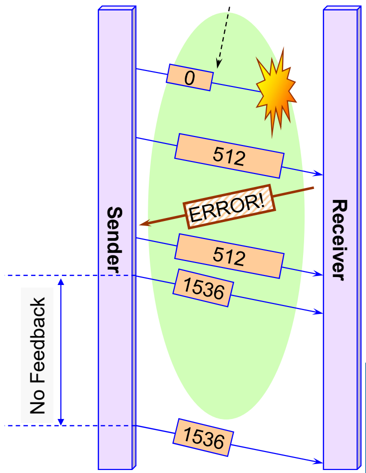


2.2 Error Recovery

- By **retransmission**

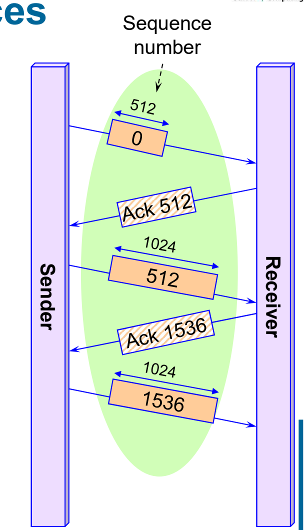
a. Upon **Error Notification** from Receiver
Good

b. Upon **Sender Decision** if no feedback from receiver
Bad



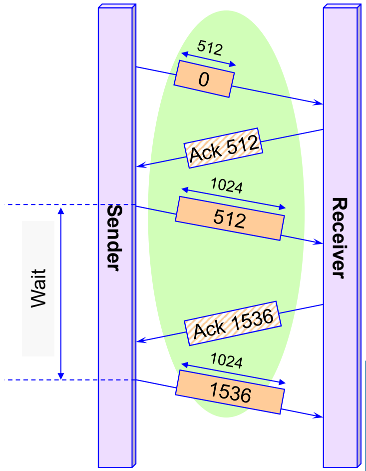
2.1 Error Services

- **Error detection**
  - Sequence number = **order of the first octet of segment** in the data stream
  - Receiver **Acknowledges** segments
  - e.g. "ACK 512 means":
    " *I am now ready to receive octet #512 and beyond, because I correctly received all octets up to #511*"



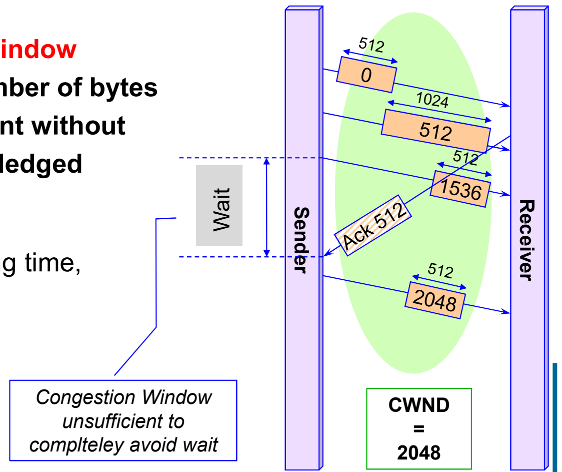
2.1 Error Services

- **Problem with Ack**
  - Sender must **wait** for ACK of segment N before sending segment N+1
- **Solution**
  - **Windowing** = anticipation on the Acks



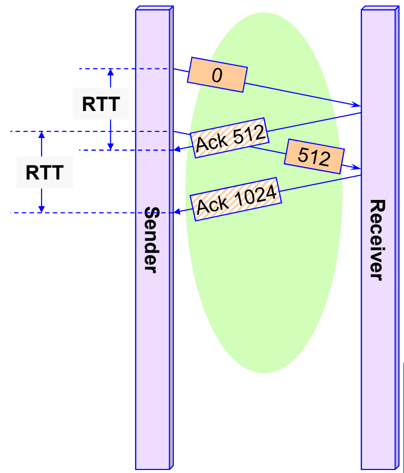
2.1 TCP Windowing

- **Congestion Window (CWND) = number of bytes that can be sent without being acknowledged**
- Reduces waiting time, hopefully to 0



2.2 Error Recovery - TCP Timeout

- **Question: *which value for the Timeout*?**
- **TCP** measures permanently the **Round Trip Time (RTT)**
  - RTT = Average Time between segment emission and ACK reception
- **Timeout = β x RTT**
  - Choice of β delicate (simple choice: β = 2)

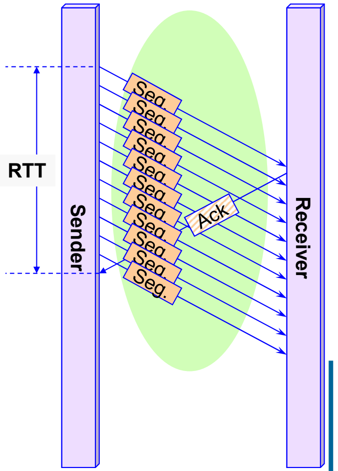


Sender Buffer Size

- Sender must **keep all sent segments** until acknowledged

- **Question:**

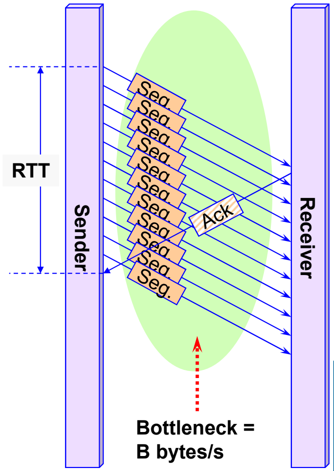
  What is the optimal buffer size to keep all segments?



Optimal Buffer Size

- If **bottleneck** of the Bandwidth between Sender and Receiver is **B** byte/second
  - Sender can send up to **B** bytes / second over **RTT** seconds
  - Maximum number of bytes to store = **B x RTT**
- Called the **Bandwidth*Delay Product** (**BDP**)

BDP = B x RTT



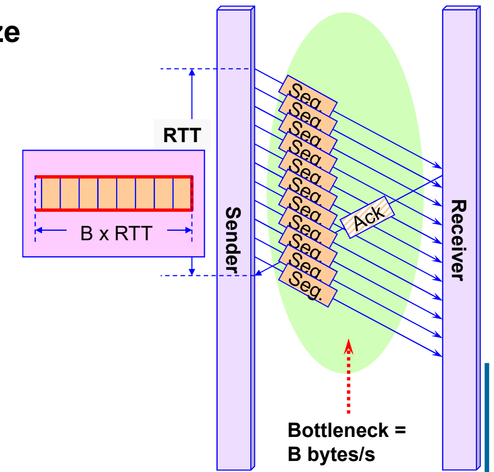
Optimal Buffer Size

- **Optimal buffer size**

  =

  Bandwidth*Delay Product

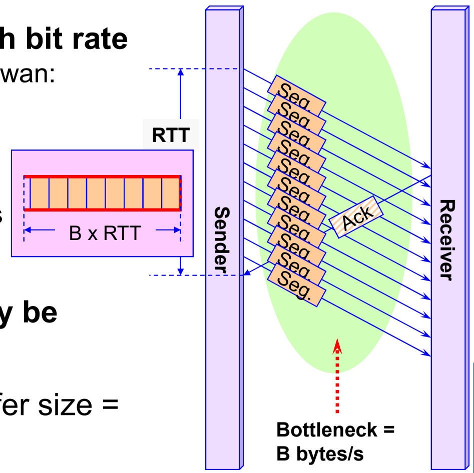
- **BDP = B x RTT**



BDP over long distance links

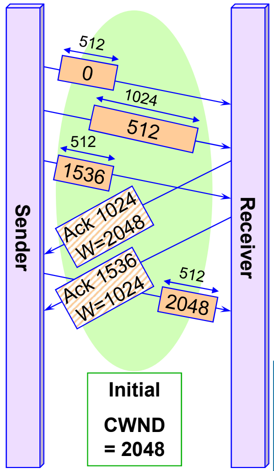
- **Long and very high bit rate links** e.g. Geneva-Taiwan:
  - RTT ~330ms
  - B = 1 Gbps
  - BDP ~40 Mbytes
- **In case packet may be dropped**

  Recommended buffer size = **2 BDP**



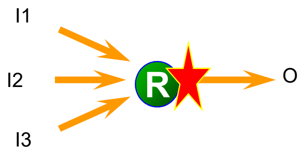
3 TCP mechanism for Flow control

- Window size CWND **not fixed**
- **Server** may reduce if detecting network congestion
- **Receiver** may reduce if sender sending to fast
- Value of Window = remaining incoming **buffer space** in Receiver

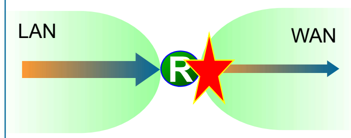


Congestions

The N-to-1 problem

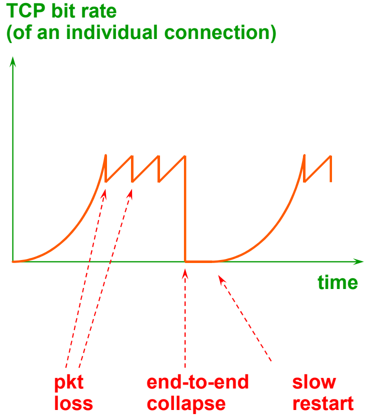


Fast / Slower junctions



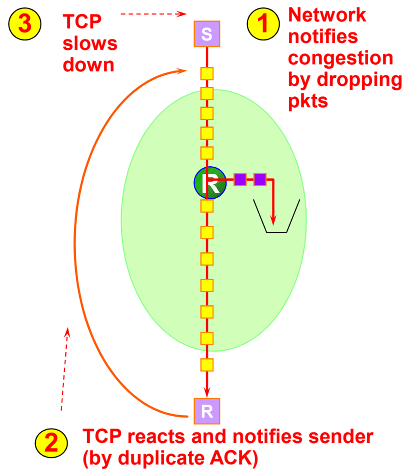
TCP congestion avoidance (TCP-Reno)

- **Called "slow start"**
  - Rate **doubles** every round-trip time
- If packet loss, sender halves window
  - Then window increases **linearly**
- In case of **end-to-end collapse** (sender time out exhausted), sender resumes with **slow start**



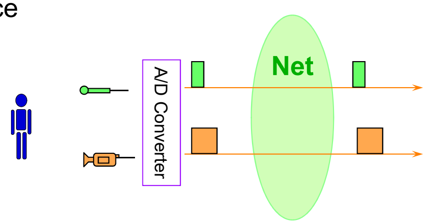
F F Congestion Notification in Internet

- Current Internet: **no explicit congestion notification** from network to source
- Notification technique: ***Network drops packets***
- TCP reacts and slows down
- TCP said to be "***congestion-indication responsive***"



Real-time media transmission

- Assume now the source of packets in an
  - analog to digital converter
  - connected to a microphone and a movie camera

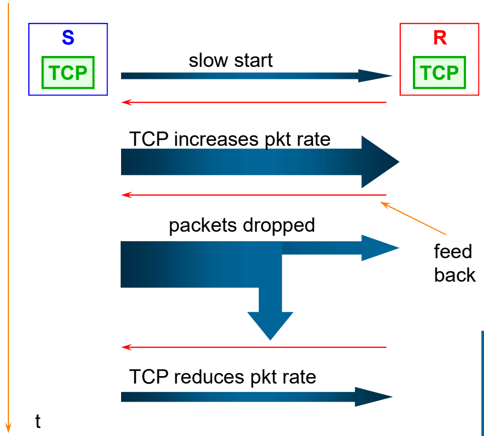


Audio and video packets

- This is **real-time audio/video**

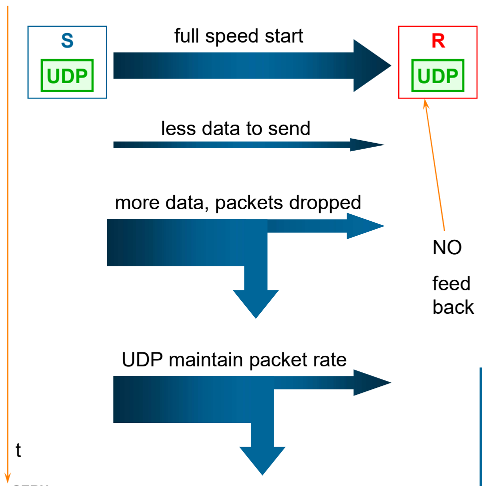
TCP behavior

- **Slow start**
- Sender aware of packets dropped
- Sender reduces bit rate when packet dropped



UDP* behavior

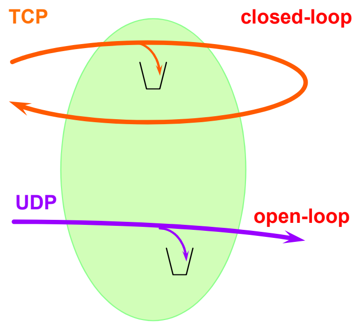
- **UDP** sends blindly to a receiver
- **No feedback** from the receiver
- Sender unaware whether packets are dropped/lost



* *User Datagram Protocol*

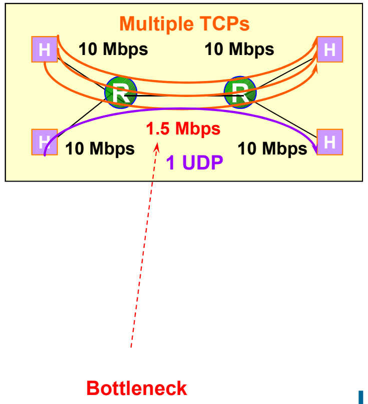
Unresponsive flows

- **Un**responsive flows do not react to **congestion indication** (pkt loss)
- Can create
  - **bandwidth starvation** inflicted to well-behaved responsive traffic



Unfair competition

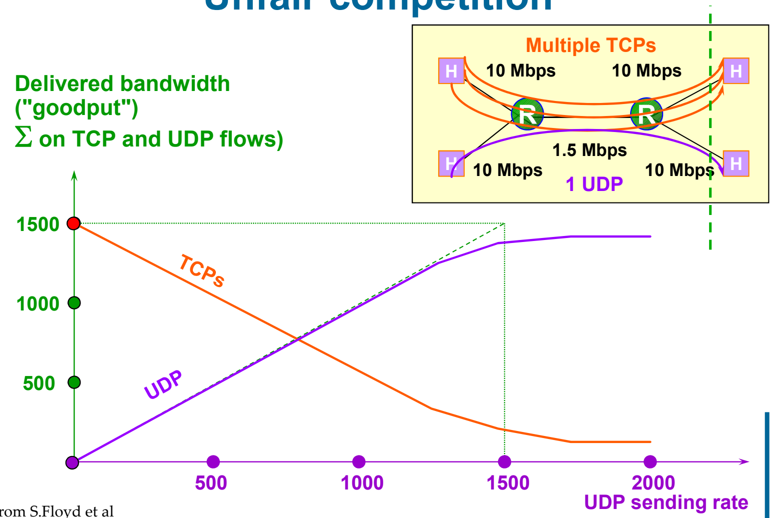
- Case
  - **2 LANs* (10 Mbps)**
  - interconnected at 1.5 Mbps via a pair or routers
- Competition between
  - **3 TCP** connections and
  - **1 UDP** connection



* *Local Area Networks*

From S.Floyd et al

Unfair competition



From S.Floyd et al

End of

Part 2

Focus on End Systems

Acronyms used in these lectures

- **ABR** *Available Bit Rate*
- **ACK** *Acknowledgement*
- **ATM** *Asynchronous Transfer Mode*
- **BDP** *Bandwidth Delay Product*
- **BE** *Best Effort*
- **CBR** *Constant Bit Rate*
- **CL** *Connectionless Mode*
- **CO** *Connection-oriented Mode*
- **CWND** *Congestion Window*
- **Diffserv** *Differentiated Services*
- **HTTP** *Hyper Text Transport Protocol*
- **IP** *Internet Protocol*

Acronyms used in these lectures

- **ISDN** *Integrated Services Data Network*
- **LAN** *Local area Network*
- **MPLS** *Multiprotocol Label Switching*
- **NCP** *Network Control Protocol*
- **QoS** *Quality of Service*
- **RTT** *Round Trip Time*
- **TCP** *Transport Control Protocol*
- **UDP** *User Datagram Protocol*
- **VBR** *Variable Bit Rate*
- **WAN** *Wide-Area Network*

Further reading ...

- **Internetworking with TCP/IP, vol 1**
  Douglas E. Commer, Prentice Hall, ISBN 0-130-183806
- **Computer Networks, Ed. 4**
  Andrew Tannenbaum, Prentice Hall, ISBN 0-130-661023
- **Understanding Networked Multimedia**
  Francois Fluckiger, Prentice Hall, ISBN 0-131-90992-4