Computer Systems (Computer Networks)

David Marchant

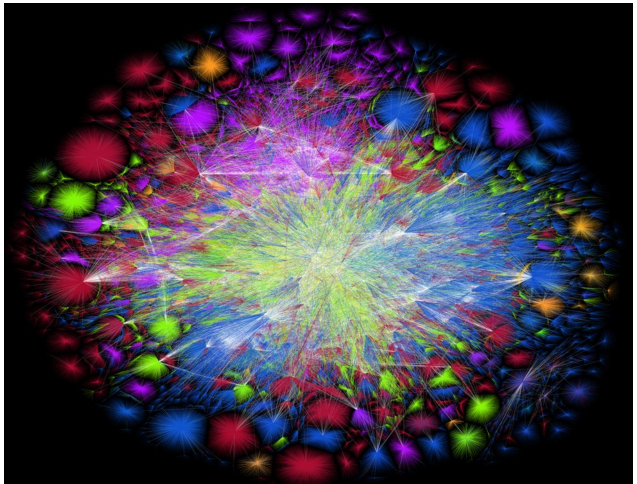
Based on slides compiled by Marcos Vaz Salles, with adaptions by Vivek Shah and Michael Kirkedel Thomsen

A quick bit about me

- PhD student in Computer Science from Scotland
- Jeg taler ikke dansk :(
- ‘David’ on the Discord
- Office: HCØ, building B, room 772-01-0-S06

Why study Computer Networks?

- How can we build networked applications?
- What are the protocols that power the Internet?
- How can we secure data transmission?



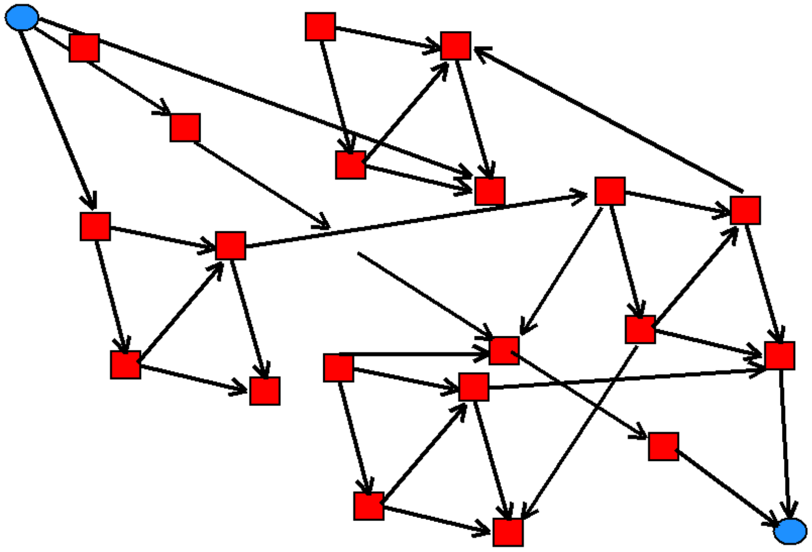
Map of the internet (2015), The Opte Project,

Computer Networks: What will we study?

- How can we build networked applications?
  - Application-level protocols, e.g., HTTP and content delivery
  - Programming with sockets
  - Resolving names with DNS

Computer Networks: What will we study?

- What are the protocols that power the Internet?
  - UDP: Basic transport
  - TCP: Reliable and ordered transport with flow and congestion control
  - IP: Addressing, forwarding, routing
  - Ethernet et al: Physical transmission



Largest portion of the course

Computer Networks: What will we study?

- How can we secure data transmission?
  - Cryptography
  - Authentication
  - HTTPS, IPSec: Securing protocols

What should we learn in this portion of the course?

- Describe the design of application-layer protocols such as HTTP and DNS.
- Implement networked applications making use of sockets.
- Explain the mechanisms used by transport-layer protocols to achieve multiplexing, reliability, flow control, and congestion control.
- Describe network setups involving subnets, NAT, and LAN segments as well as related interconnection hardware such as routers, switches, and hubs.
- Explain the use of cryptography and operational measures to secure network protocols and applications.
- Build for performance by using all the hardware we have, both locally and remotely.
- Utilise events as basis for applications.
- De-mystify cloud computing and its impacts on data center networking

Computer Networks: What will we study?

- Not only theoretical knowledge, but also skills
  - Programming with sockets
  - Implementing protocols
  - Using network tools
  - Building distributed applications

References & Course Materials

- Book
  - Computer Networking, 8th ed., James F. Kurose and Keith W. Ross, Pearson, ISBN 13: 978-0-135-92873-8
  - Computer Systems: A Programmers Perspective, 3rd ed., Randal Bryant and David O'Hallaron, ISBN 13: 978-0-134-09266-9

- Other references
  - Vast majority listed in the course schedule
  - Will keep updating them as we go

Teaching and Assignments

- Lectures
    - **Informal discussion oriented**.
    - **“The only stupid question is the one not asked”**
    - **Do not wait for course evaluation for hot fixes**
    - Slides are not a substitute for the book
- Assignments
    - **Assignments A3-A4**
        - Exercises on **concepts** as well
        - Programming of **distributed P2P file sharer**
        - **No plagiarism – Cite appropriately**

Acknowledgements

- Many of the slides in this course are based on or reproduce material kindly made available by Michael Freedman (Princeton), James Kurose & Keith Ross (RPI & NYU-Poly, textbook), Jerome Saltzer & M. Frans Kaashoek & Robert Morris (MIT), Randal E. Bryant & David R. Halloran (Computer Systems, textbook)
- Marcos Vaz Salles (Associate Professor, DIKU) for creating and compiling these slide sets

Questions so far?

What should we learn today?

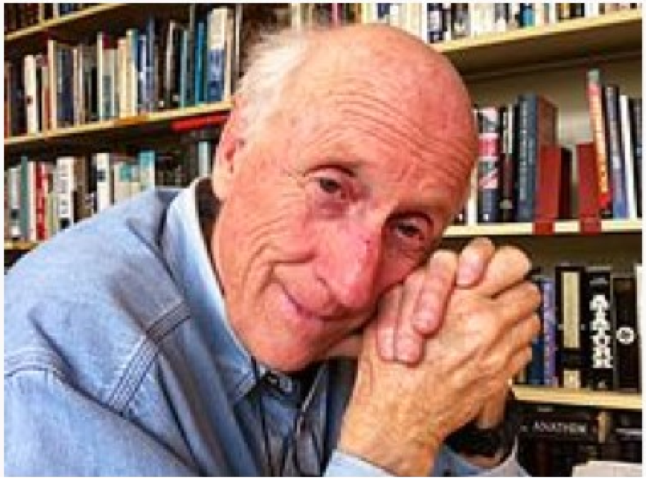
- Identify the key concepts in networking of protocols, layering, resource allocation, and naming.
- Describe what a protocol is and the main issues in protocol design.
- Explain the goal of layering in networked systems and the multiple layers in the Internet protocol suite.
- Explain how circuit switching and packet switching address the resource allocation problem in networks.
- Explain Time-Division Multiplexing (TDM) and Frequency-Division Multiplexing (FDM) and predict transmission time in circuit switched networks.
- Explain the advantages and challenges of packet switching and store-and-forward, and identify the components of transfer time, including processing, queueing, transmission, and propagation.
- Predict transfer time in such store-and-forward networks.
- Explain the notion of throughput in store-and-forward networks and predict throughput in specific scenarios.

Networking is Relevant

Source: Freedman

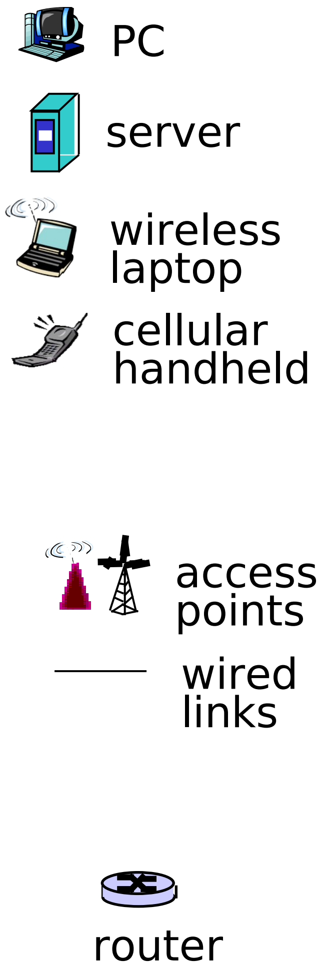
Information wants to be free because it has become so cheap to distribute, copy, and recombine... It wants to be expensive because it can be immeasurably valuable to the recipient. (1985)

Stewart Brand

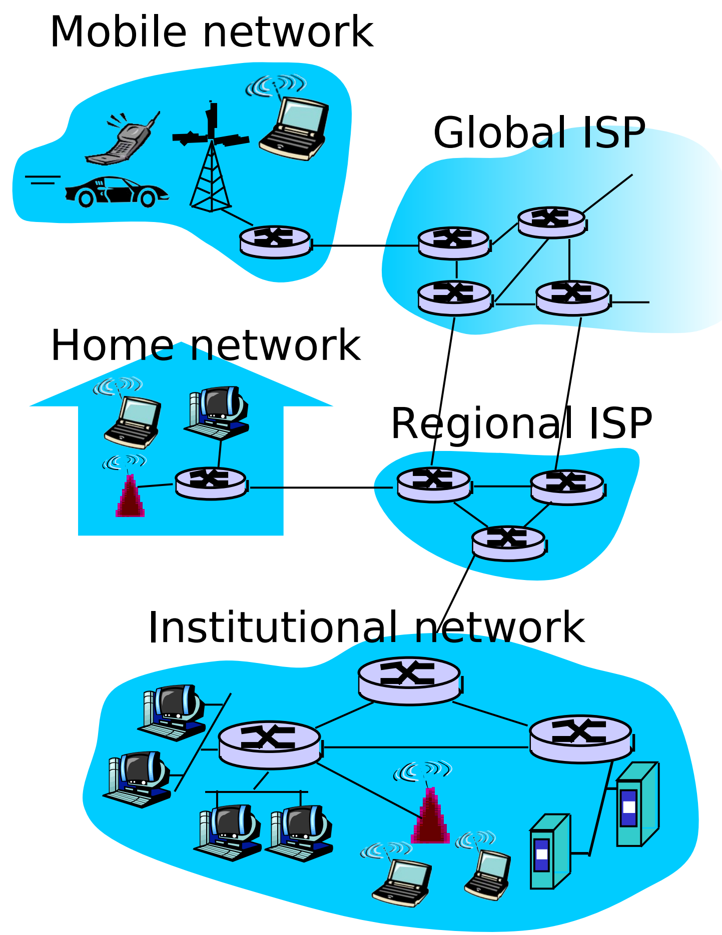


Source: Freedman

What’s the Internet: “nuts and bolts” view



- A network of networks
- Millions of connected computing devices: *hosts = end systems*
  - running *network apps*
- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate = *bandwidth*
- *routers:* forward packets (chunks of data)



Source: Kurose & Ross (partial)

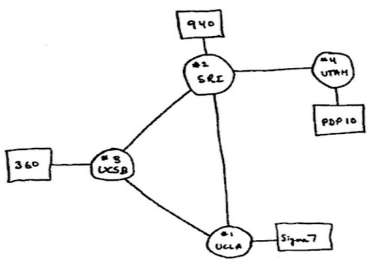
Key Concepts in Networking

- Protocols
  - Speaking the same language
  - Syntax and semantics
- Layering
  - Standing on the shoulders of giants
  - A key to managing complexity
- Resource allocation
  - Dividing scarce resources among competing parties
  - Memory, link bandwidth, wireless spectrum, paths
- Naming
  - What to call computers, services, protocols, …

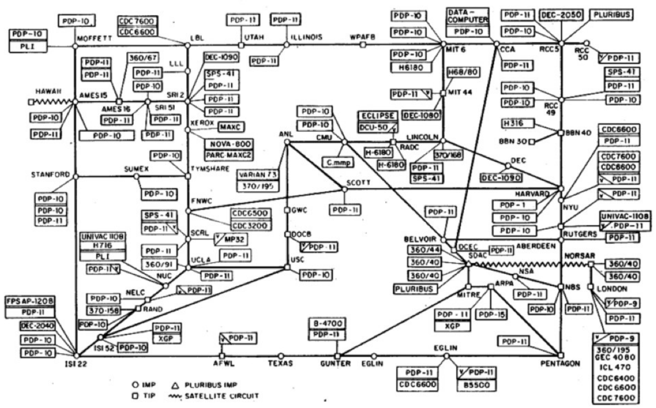
Source: Freedman

Key Concepts in Networking

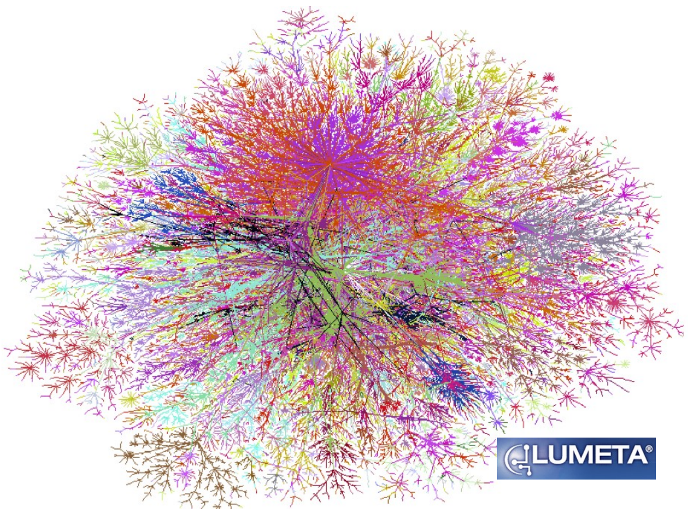
- Protocols
  - Speaking the same language
  - Syntax and semantics



1969



1977



1998

All speak IPv4
“ Internet Protocol version 4”

Source: Freedman

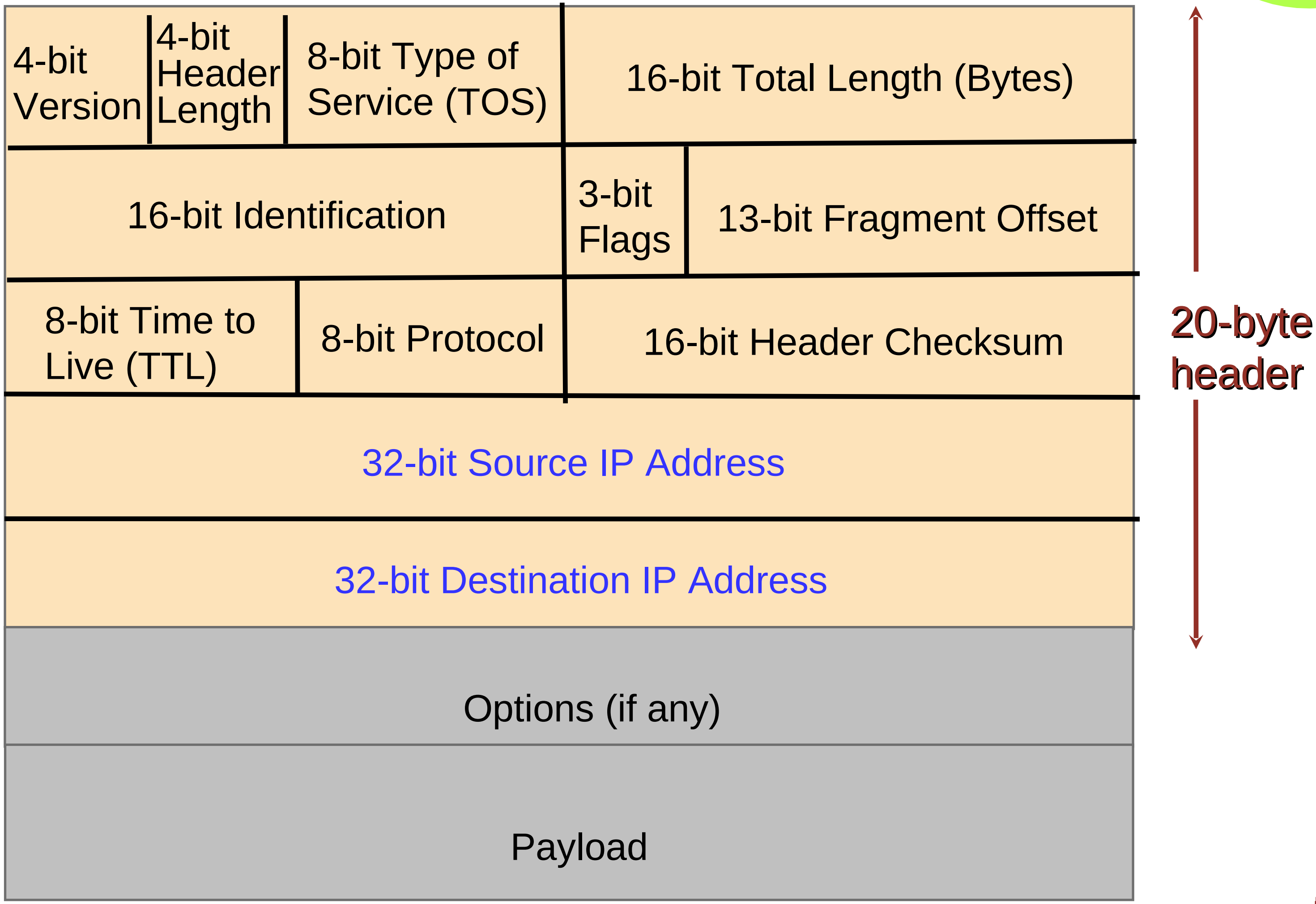
Protocol design is about tradeoffs

- **How should hosts and routers communicate?**
  - Standard protocol
  - Fast: Machine readable in hardware at line rates

- **Browsers, web servers, and proxies?**
  - Can be slower: software readable
  - Human readable
  - Extensible and forward-compatible
  - Not everybody might be familiar with extensions

Source: Freedman

IPv4 Packet



Source: Freedman

Example: HyperText Transfer Protocol

```
GET /courses/archive/spr09/cos461/ HTTP/1.1
Host: www.cs.princeton.edu
User-Agent: Mozilla/4.03
CRLF
```

Request

```
HTTP/1.1 200 OK
Date: Mon, 2 Feb 2009 13:09:03 GMT
Server: Netscape-Enterprise/3.5.1
Last-Modified: Mon, 21 Feb  2009 11:12:23 GMT
Content-Length: 42
CRLF
Site under construction
```

Response

Source: Freedman (partial)

Key Concepts in Networking

- Protocols
  - Speaking the same language
  - Syntax and semantics
- Layering
  - Standing on the shoulders of giants
  - A key to managing complexity

Source: Freedman

Key Concepts in Networking

Application Layer

Transport Layer

Network Layer

Link Layer

Physical Layer

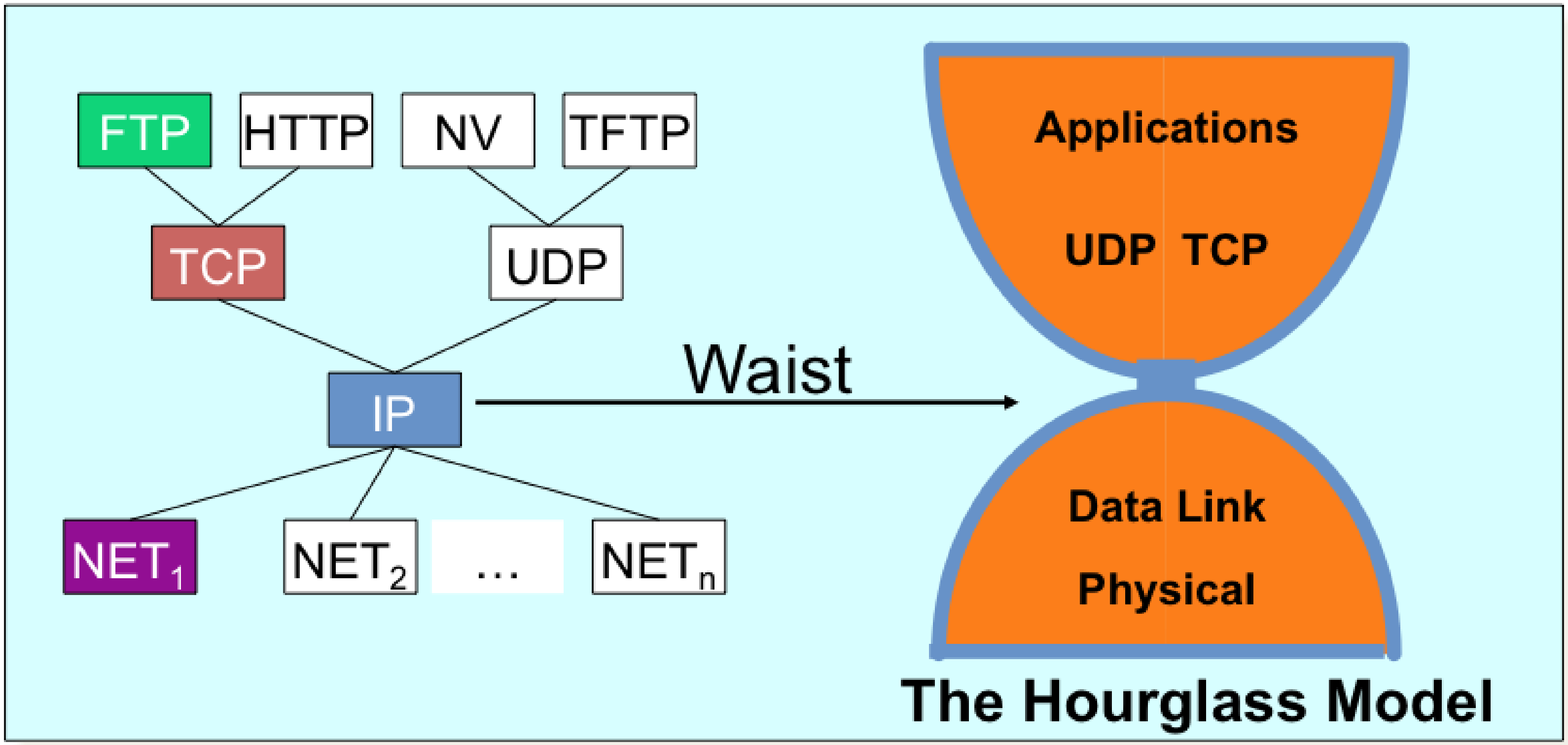
Layering = Functional Abstraction

- Sub-divide the problem
  - Each layer relies on services from layer below
  - Each layer exports services to layer above
- Interface between layers defines interaction
  - Hides implementation details
  - Layers can change without disturbing other layers

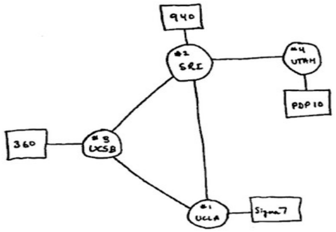
Application
Application-to-application channels
Host-to-host connectivity
Link hardware

Source: Freedman

The Internet Protocol Suite

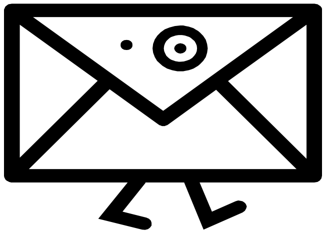


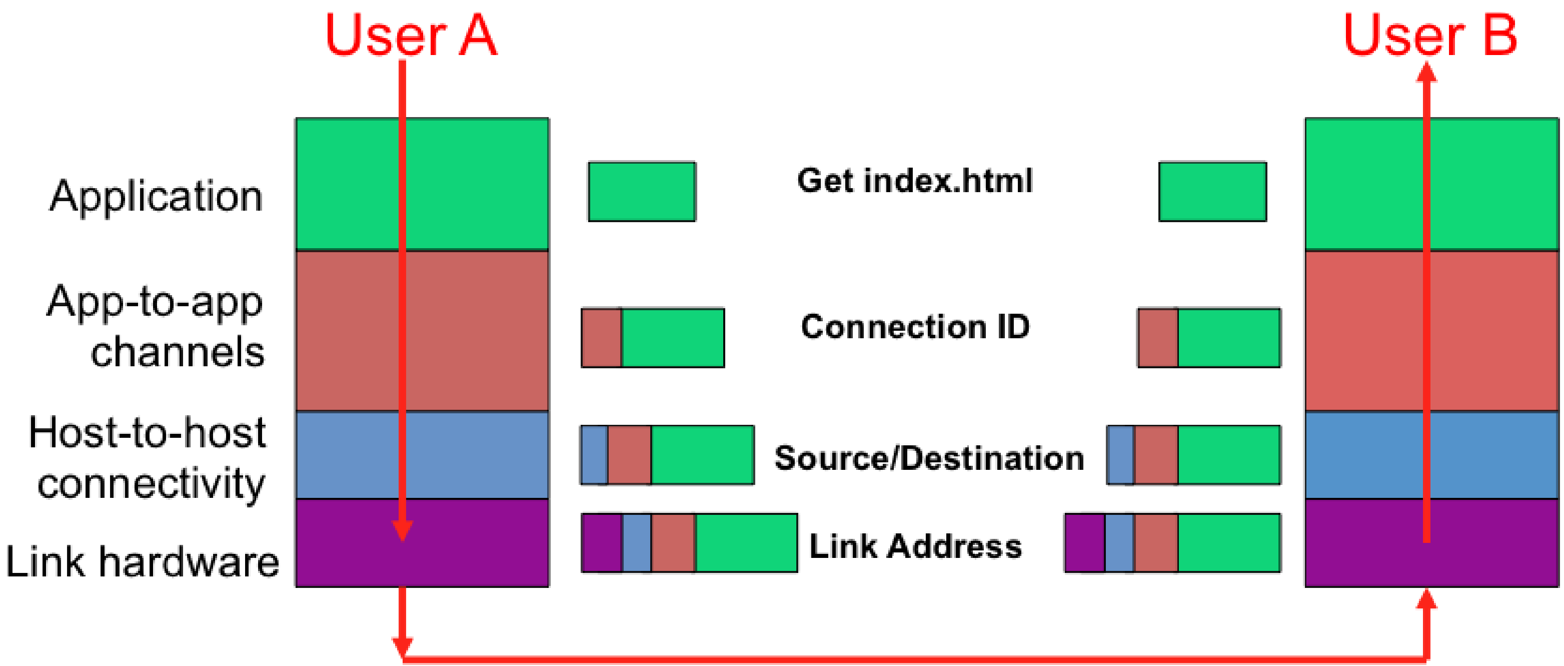
The waist facilitates interoperability



Source: Freedman

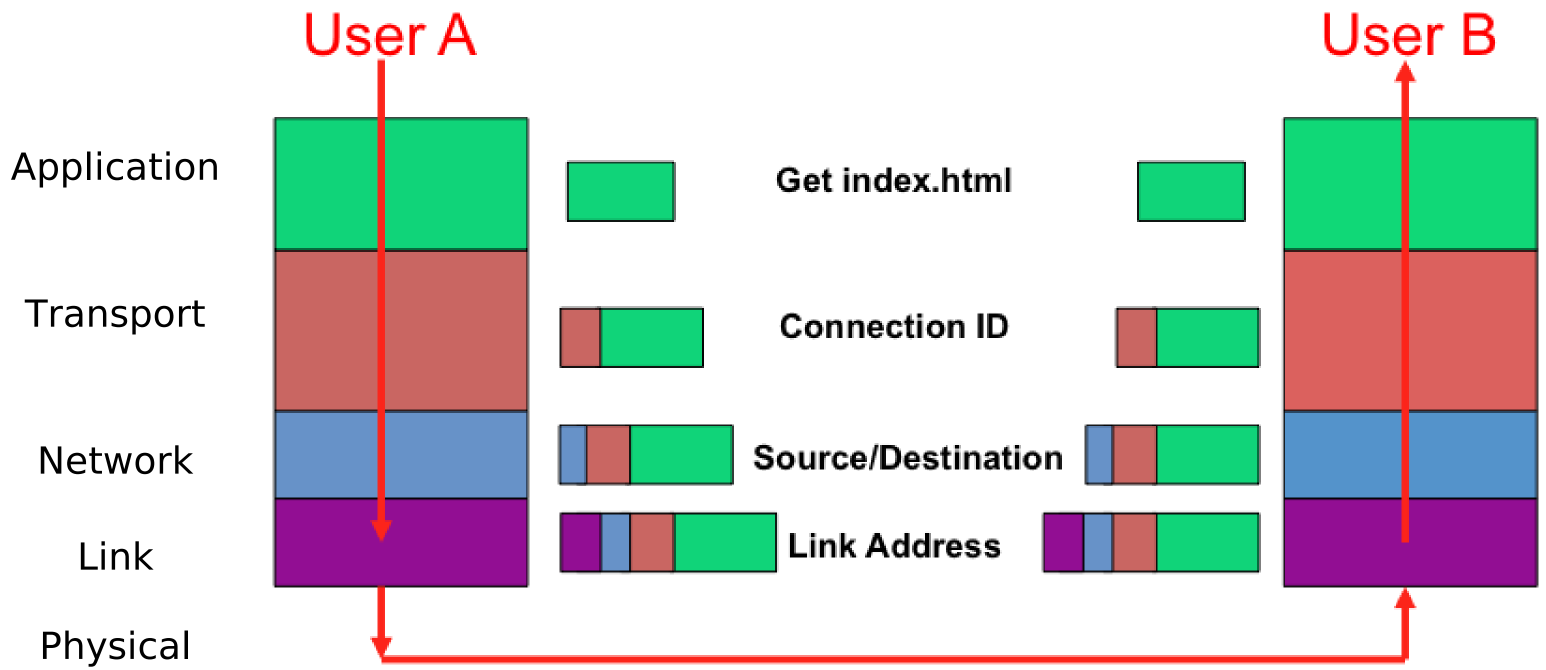
Layer Encapsulation in HTTP





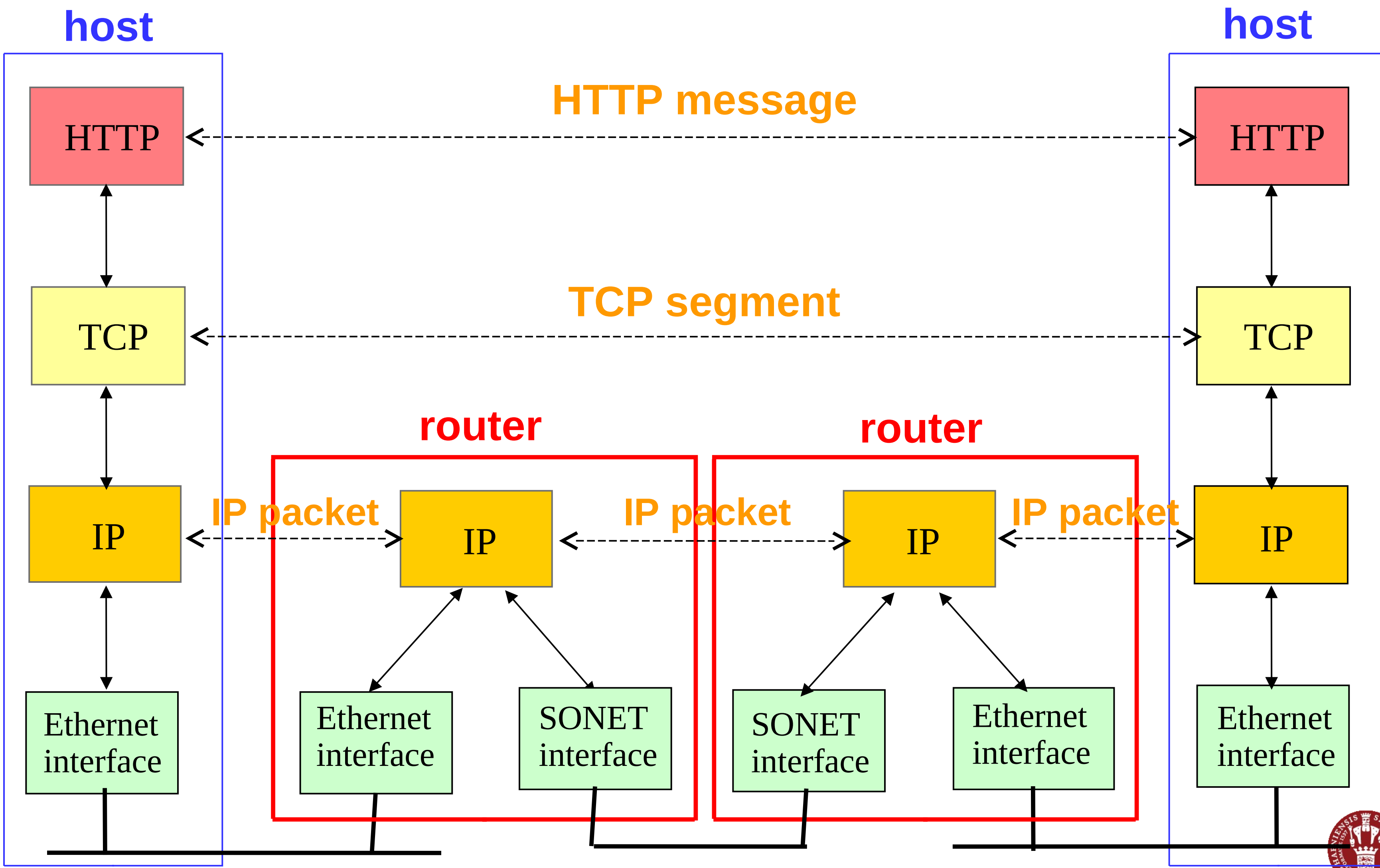
Source: Freedman

Layer Encapsulation in HTTP



Source: Freedman

IP Suite: End Hosts vs. Routers



Source: Freedman

Key Concepts in Networking

- Protocols
  - Speaking the same language
  - Syntax and semantics
- Layering
  - Standing on the shoulders of giants
  - A key to managing complexity
- Resource allocation
  - Dividing scarce resources among competing parties
  - Memory, link bandwidth, wireless spectrum, paths

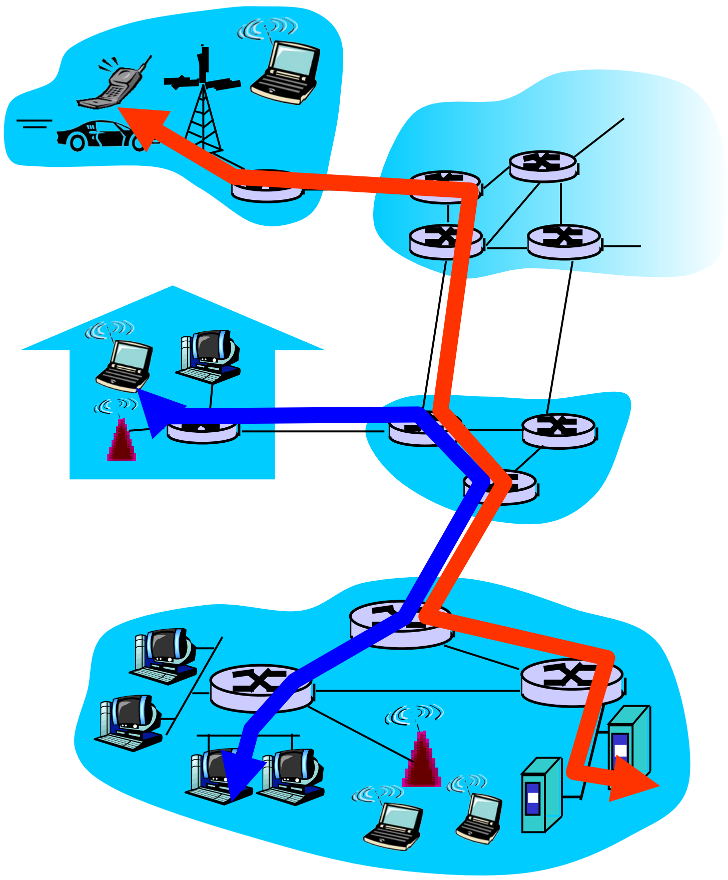
First Example: circuit vs. packet switching

Source: Freedman (partial)

Network Core: Circuit Switching

end-end resources reserved for "call"

- link bandwidth, switch capacity
- dedicated resources: no sharing
- circuit-like (guaranteed) performance
- call setup required



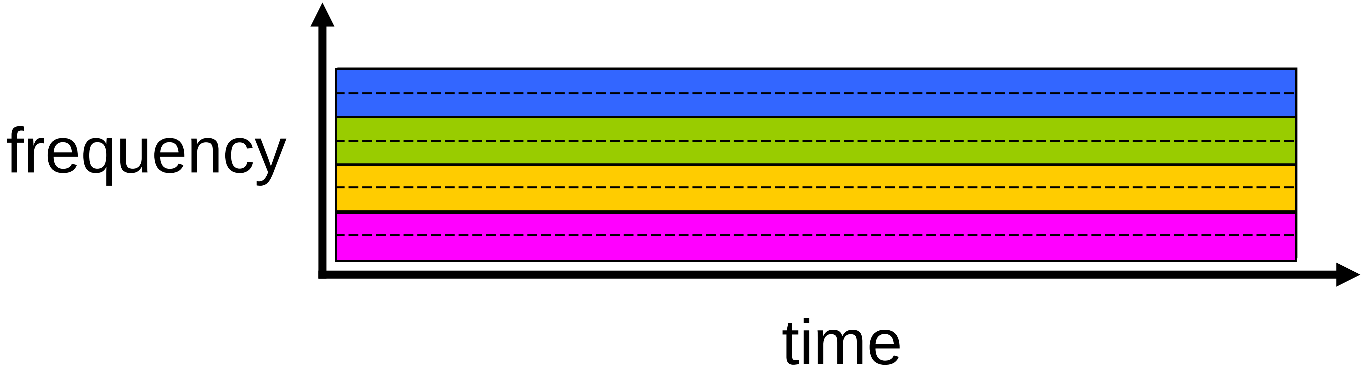
Source: Kurose & Ross

Circuit Switching: FDM and TDM

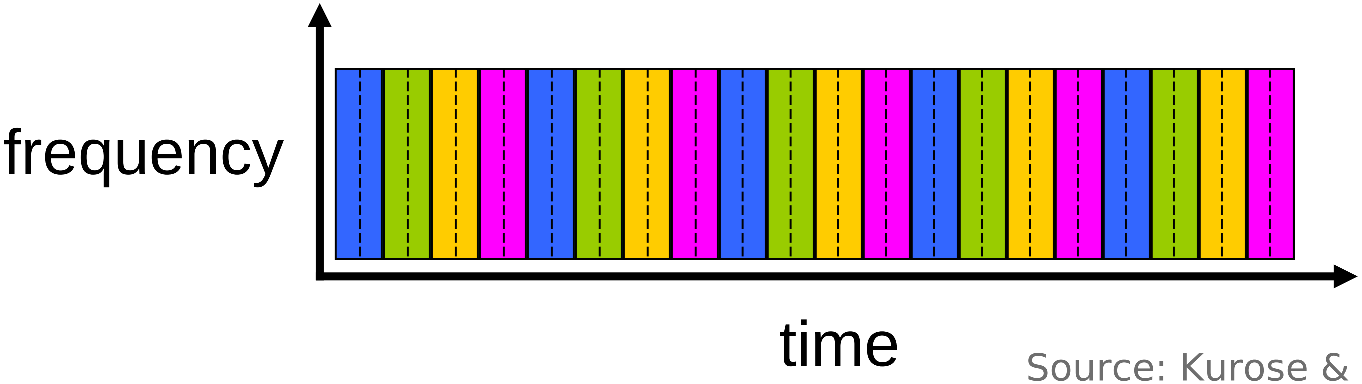
Example:

4 users

FDM



TDM



Source: Kurose & Ross

Numerical example

- How long does it take to send a file of 640,000 bits from host A to host B over a circuit-switched network?
  - all link speeds: 1536 Mbps
  - each link uses TDM with 24 slots/sec
  - 500 msec to establish end-to-end circuit
  - Note: 1 Mbps = 10^6 bps

- Let's work it out!
- Possible answers

  (a) 500 msec

  (b) 500.4 msec

  (c) 510 msec

  (d) 1 sec

Source: Kurose & Ross

Numerical example

- How long does it take to send a file of 640,000 bits from host A to host B over a circuit-switched network?
  - all link speeds: 1536 Mbps
  - each link uses TDM with 24 slots/sec
  - 500 msec to establish end-to-end circuit
  - Note: 1 Mbps = 10^6 bps

- Let's work it out!
- Possible answers

  (a) 500 msec

  (b) 500.4 msec

  (c) 510 msec (Assuming only 1 link)

  (d) 1 sec

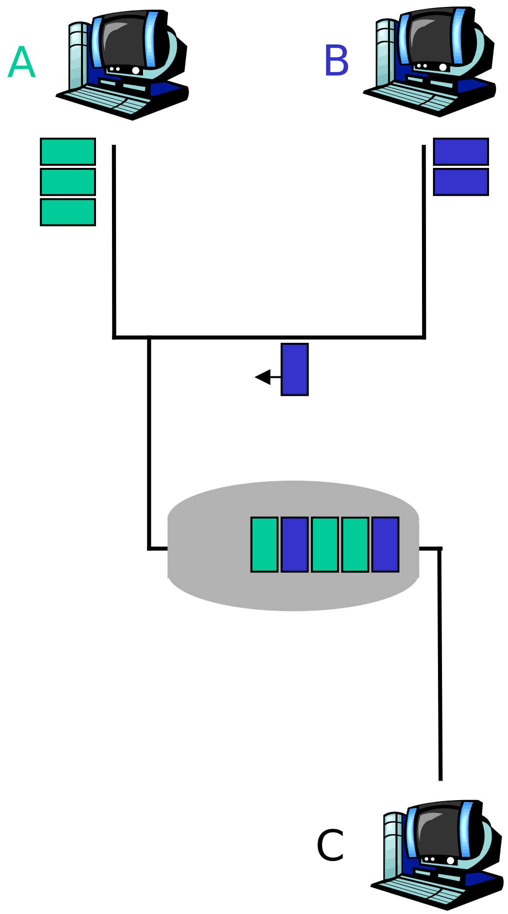
$$d = N(L/R)$$

Source: Kurose & Ross

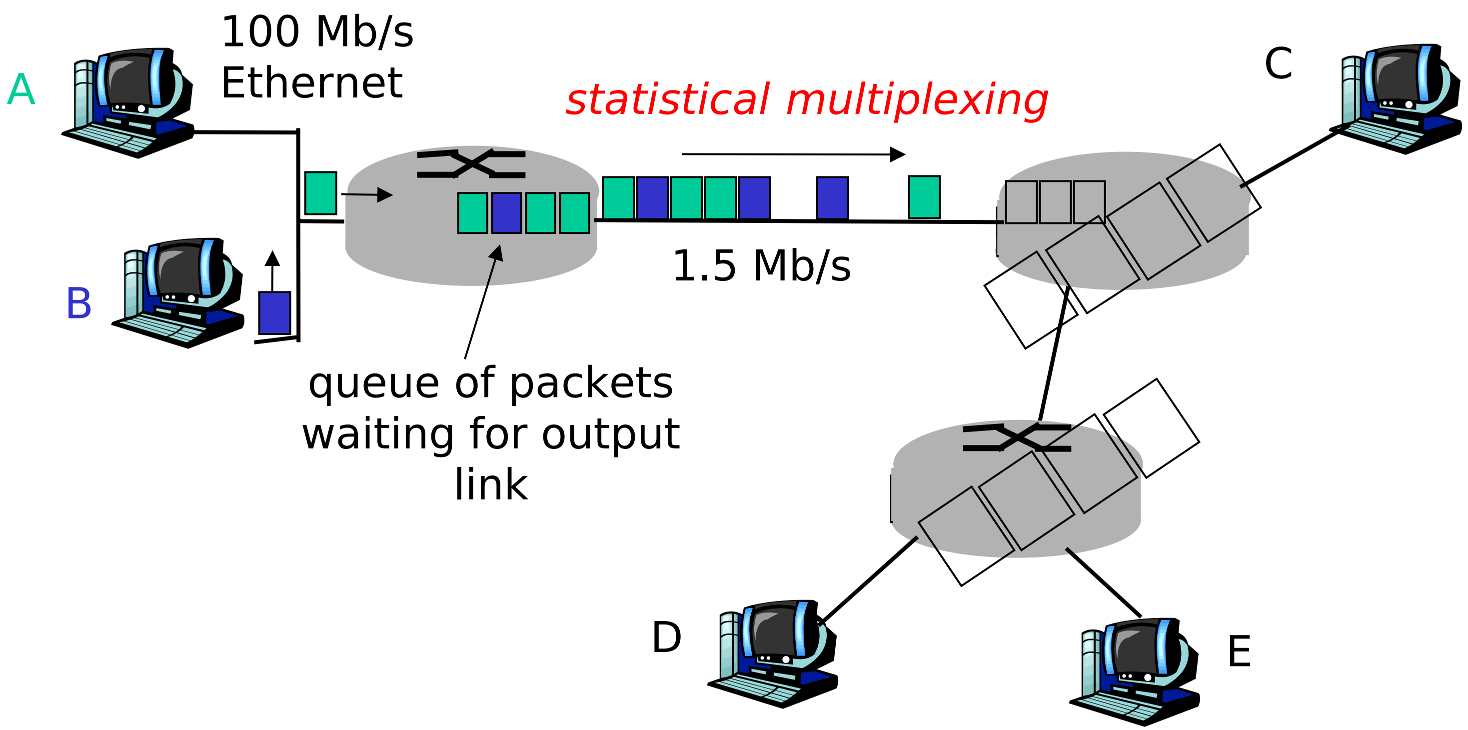
Network Core: Packet Switching

Data broken up into smaller 'packets'

- Can be sent seperately, each using full bandwidth
- Full bandwidth used on a single packet
- Packets from different sources are interspersed
- Resources only used when needed
- Unreliable transfer



Packet Switching: Statistical Multiplexing



- sequence of A & B packets has no fixed timing pattern
  - bandwidth shared on demand: statistical multiplexing.
- TDM: each host gets same slot in revolving TDM frame.

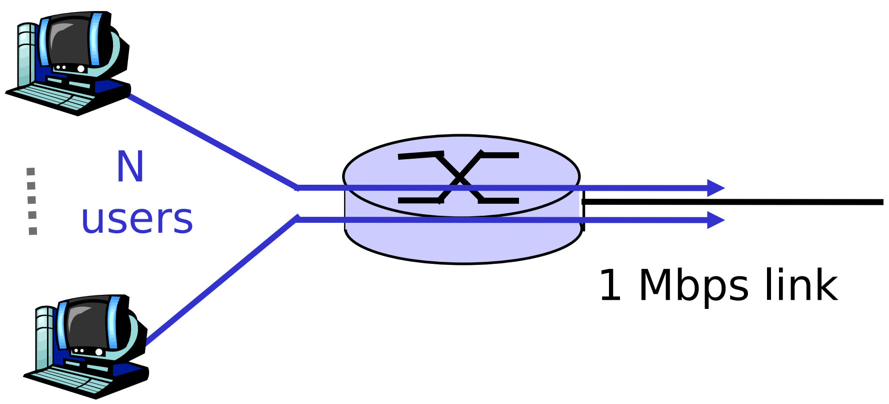
Source: Kurose & Ross

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

Q: How did we get value 0.0004?

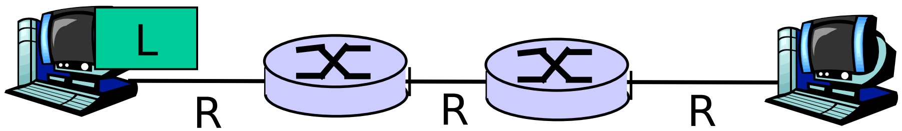
A: $P(N=10) = C(35,10)*P(A)^{10}*(1-P(A))^{(35-10)}$

$P(N>10) \rightarrow$ sum the above for N=10..35

Q: what happens if > 35 users ?

Source: Kurose & Ross (partial)

Packet-switching: store-and-forward



- takes L/R seconds to transmit (push out) packet of L bits on to link at R bps
- store and forward: entire packet must arrive at router before it can be transmitted on next link
- delay = 3L/R (assuming zero propagation delay)

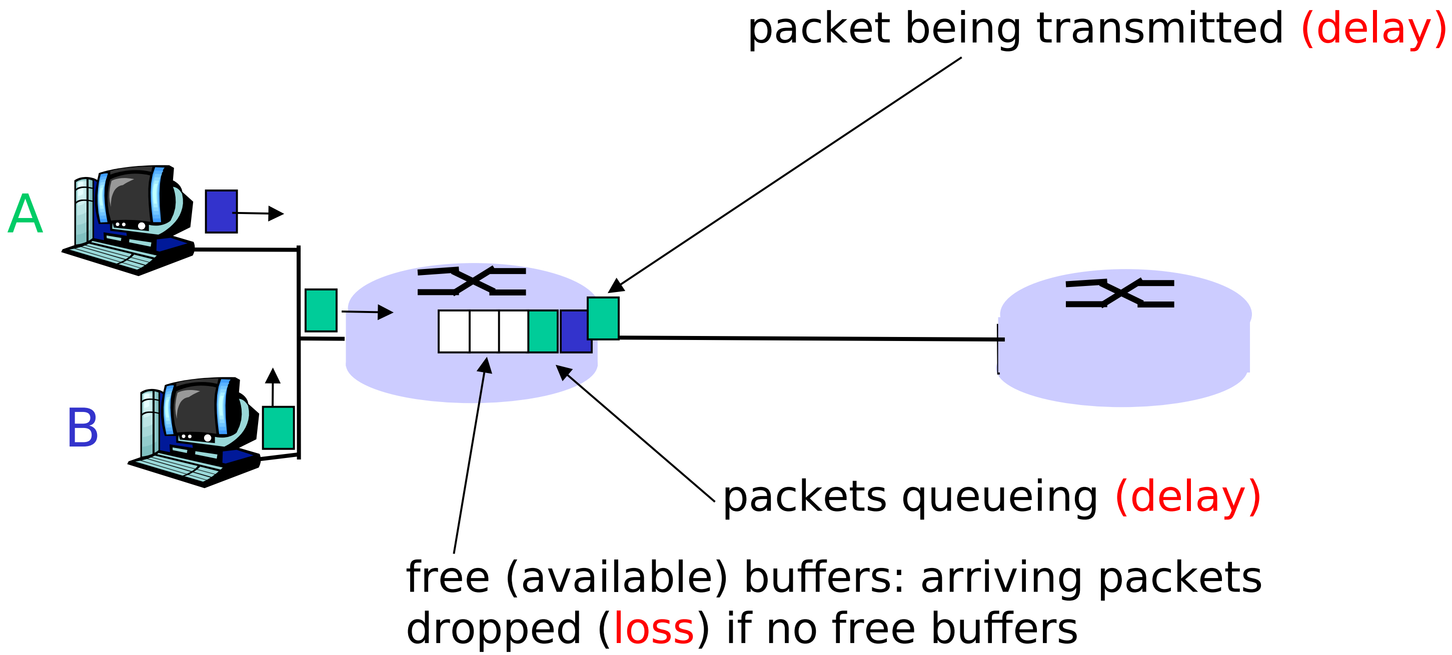
more on delay shortly …

- Example:
  - L = 7.5 Mbits
  - R = 1.5 Mbps
  - transmission delay =
    (a) 5 sec
    (b) 10 sec
    (c) 15 sec
    (d) 20 sec

Source: Kurose & Ross

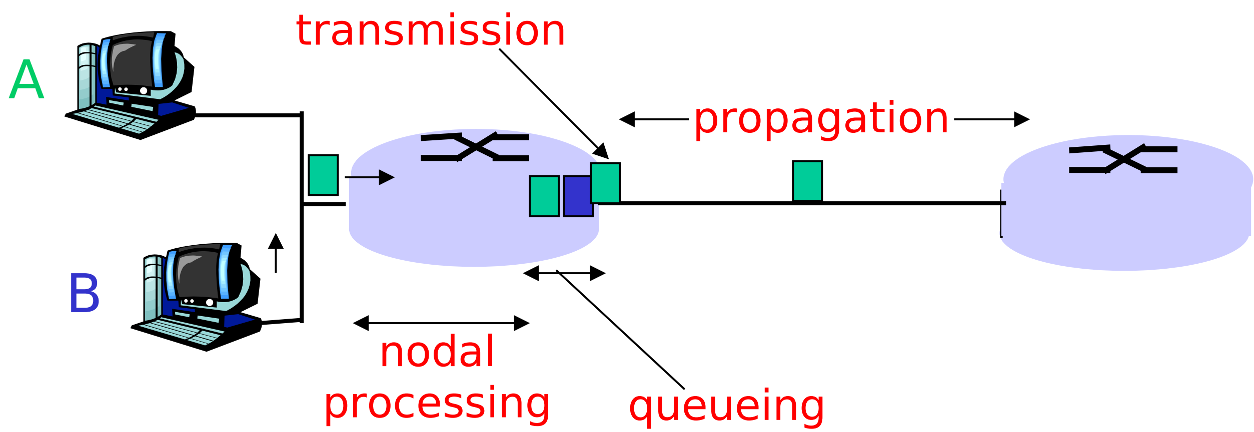
How do loss and delay occur?

- packets queue in router buffers
- packet arrival rate to link exceeds output link capacity
- packets queue, wait for turn



Source: Kurose & Ross

Four sources of packet delay



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

d_{proc}: nodal processing

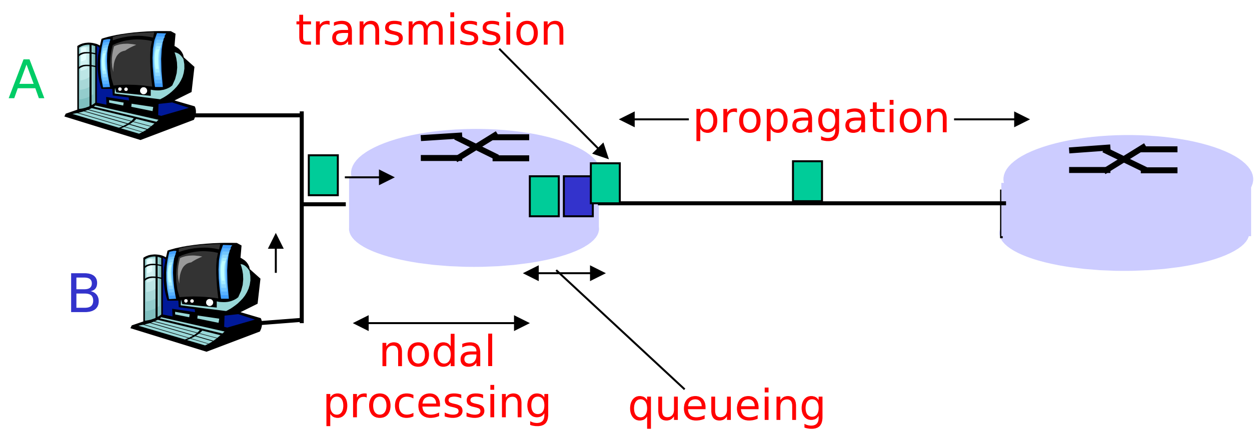
- check bit errors
- determine output link
- typically < msec

d_{queue}: queueing delay

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

Source: Kurose & Ross

Four sources of packet delay



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

d_{trans}: transmission delay

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

d_{prop}: propagation delay

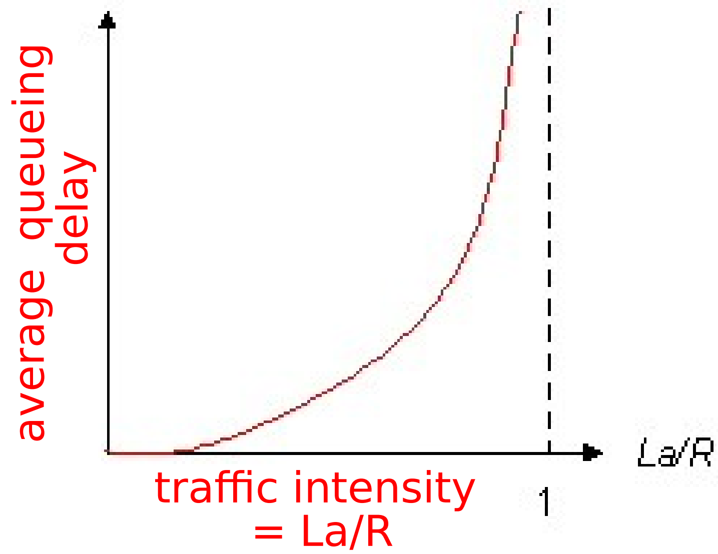
- d: length of physical link
- s: propagation speed in medium (~2×10^8 m/sec)
- $d_{prop} = d/s$

d_{trans} and d_{prop}
very different

Source: Kurose & Ross

Queuing delay

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



- La/R ~ 0: avg. queuing delay small
- La/R → 1: avg. queuing delay large
- La/R > 1: more “work” arriving than can be serviced, average delay infinite!

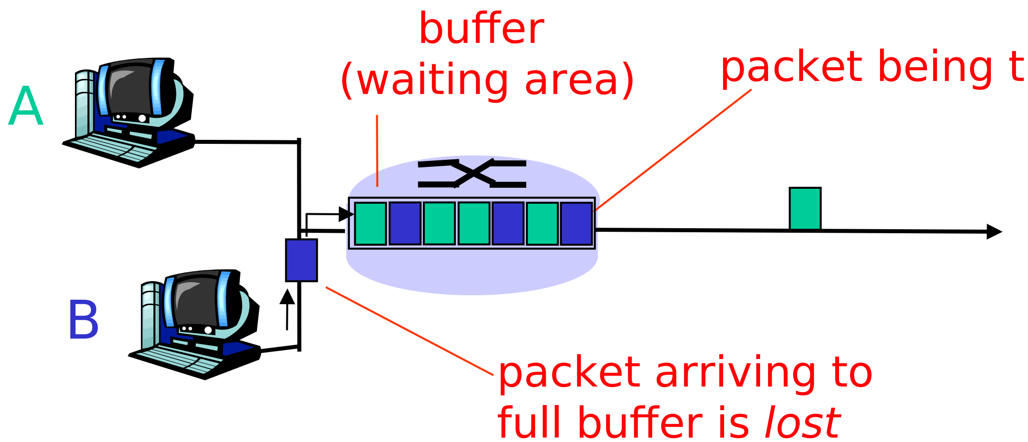

La/R ~ 0


La/R → 1

Source: Kurose & Ross

Packet Loss

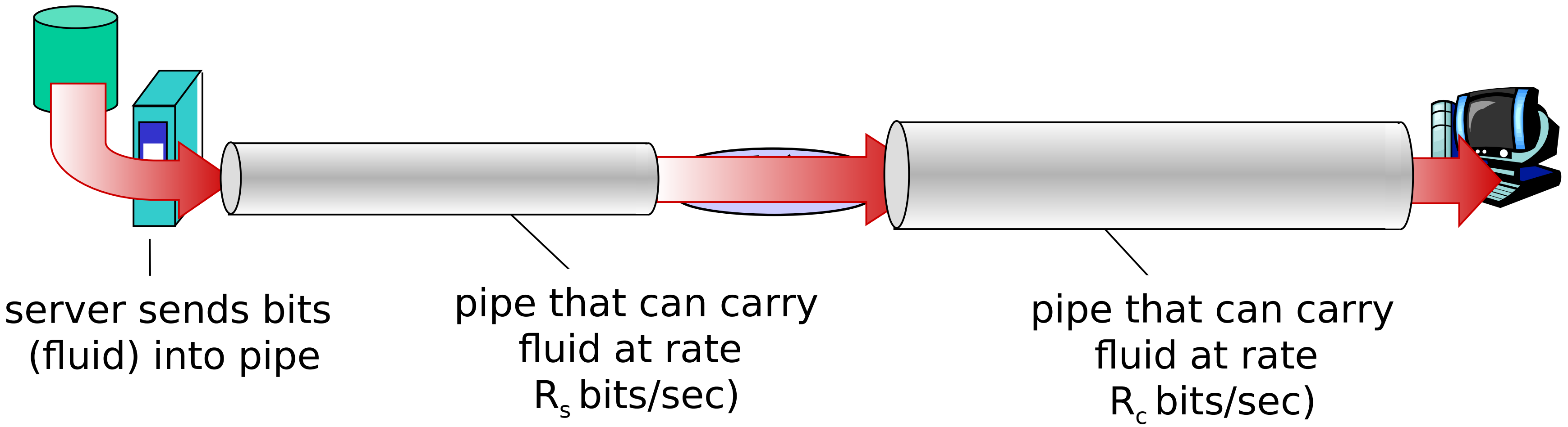
- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be re-transmitted by previous node, by source end system, or not at all



Source: Kurose & Ross

Throughput

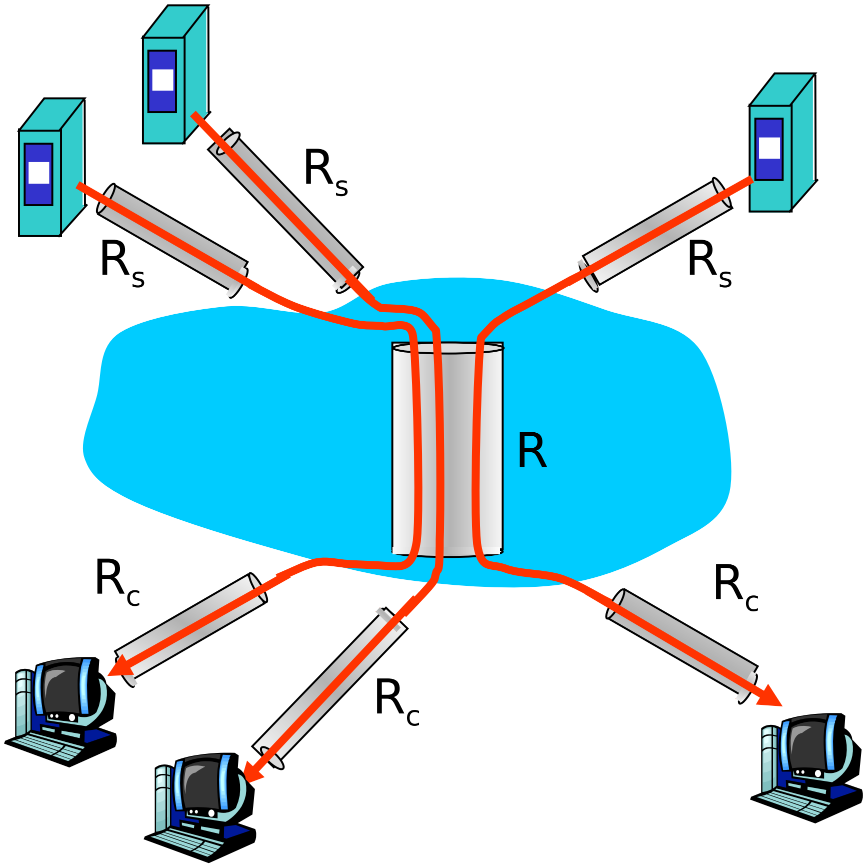
- throughput: rate (bits/time unit) at which bits transferred between sender/receiver
  - instantaneous: rate at given point in time
  - average: rate over longer period of time



Source: Kurose & Ross

Throughput: Internet scenario

- per-connection end-end throughput: $\min(R_c, R_s, R/3)$
- in practice: R_c or R_s is often bottleneck



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

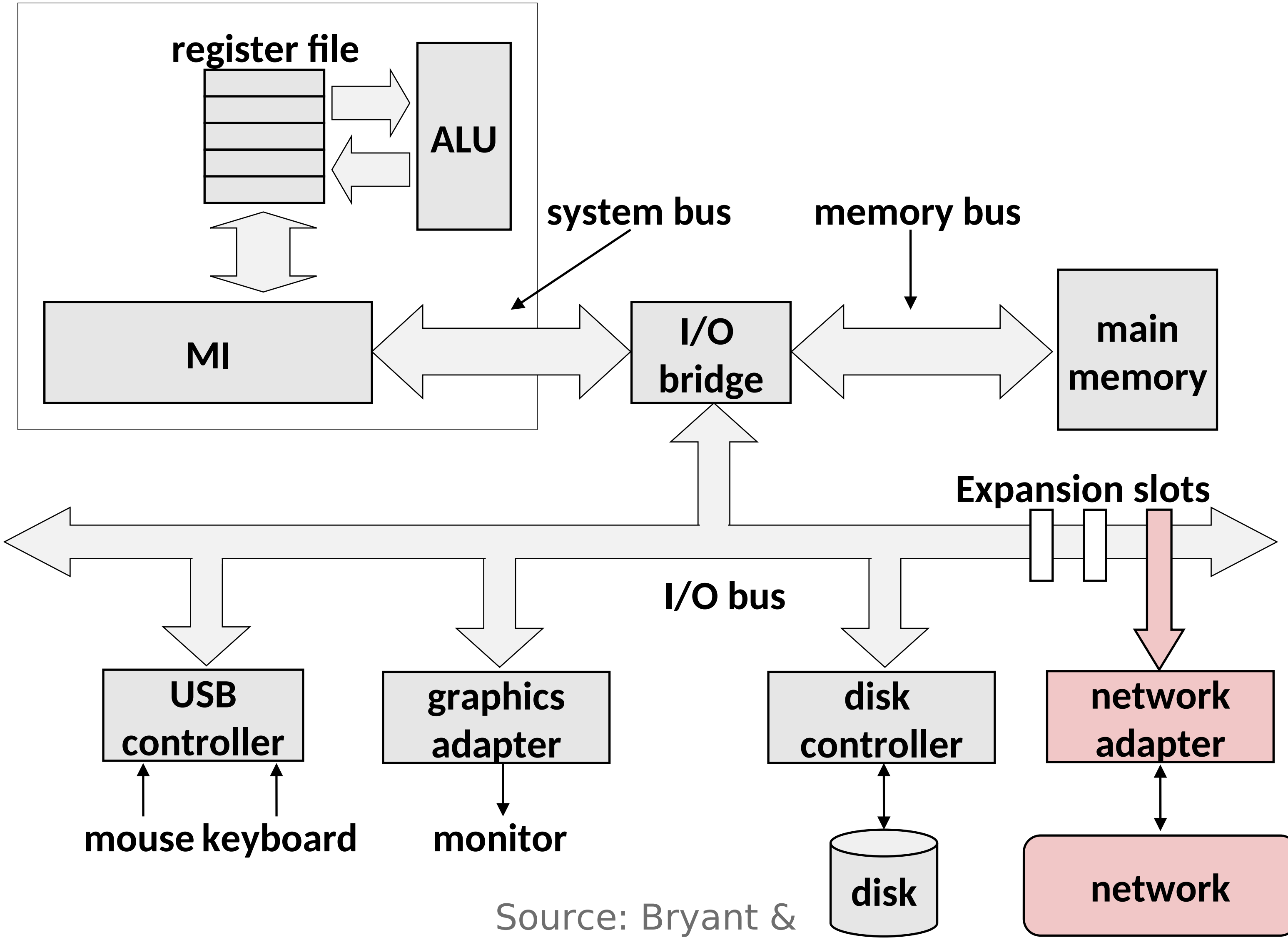
Source: Kurose & Ross

Summary

- Internet is a network of networks
  - Inter-operability, power to the edges
- Key concepts in networking
  - Protocols, layers, resource allocation, and naming
- Circuit switching
  - FDM
  - TDM
- Packet switching
  - Store-and-forward
  - Delay, loss, throughput

What's next ? What to program on the network ?

What's next ? How to program the network ?



Source: Bryant & Halloran