CcompSci 356 Computer Network Architecture
Lecture 27: A brief introduction to Quality of Service and Final review

Xiaowei Yang
xwy@cs.duke.edu

Overview

- Introduction to Quality of Service (30 mins)
- Review (30 mins)
- Pop quiz (~10 mins)
- Course evaluation (~5 mins)

Internet Quality of Service
[PD] Chapter 6.5

Internet Quality of Service

Motivation

- Internet currently provides one single class of **"best-effort" service**
  - No assurance about delivery
- Many existing applications are ***elastic***
  - Tolerate delays and losses
  - Can adapt to congestion
- "Real-time" applications may be ***inelastic***

Inelastic Applications

- Continuous media applications
  - **Lower and upper limit** on acceptable performance
  - Below which video and audio are not intelligible
  - Internet telephones, teleconferencing with high delay (200 - 300ms) impair human interactions

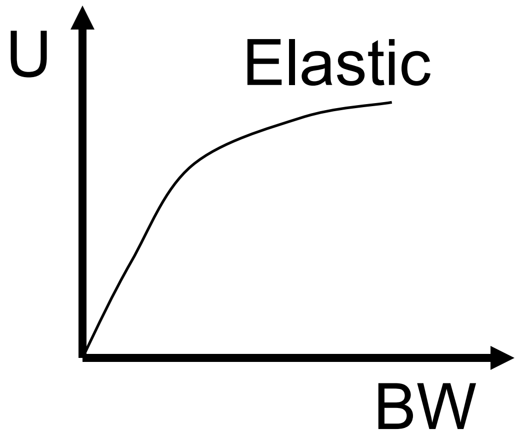
- Hard real-time applications
  - Require **hard limits on performance**
  - E.g., industrial control applications
    - Internet surgery

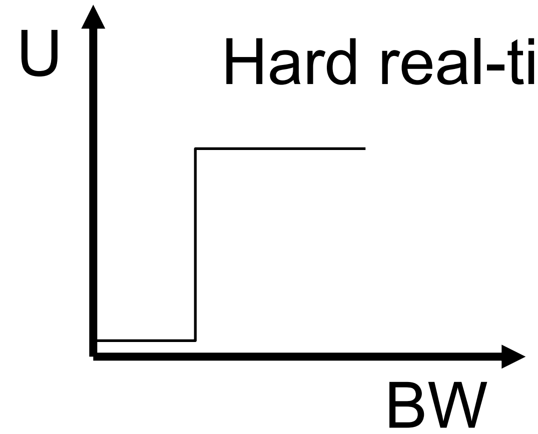
Design question #1: Why a New Service Model?

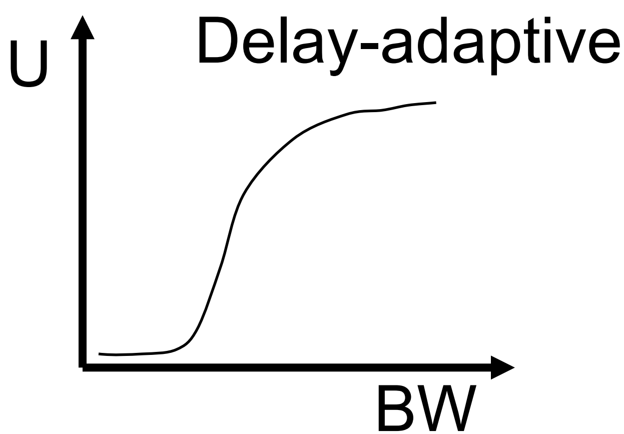
- What is the **basic objective** of network design?
  - Maximize total bandwidth? Minimize latency? Maximize ISP's revenues?
  - **the designer's choice: Maximize social welfare:** the total **utility** given to users (why not profit?)

- What does utility vs. bandwidth look like?
  - Must be non-decreasing function
  - Shape depends on application

Utility Curve Shapes





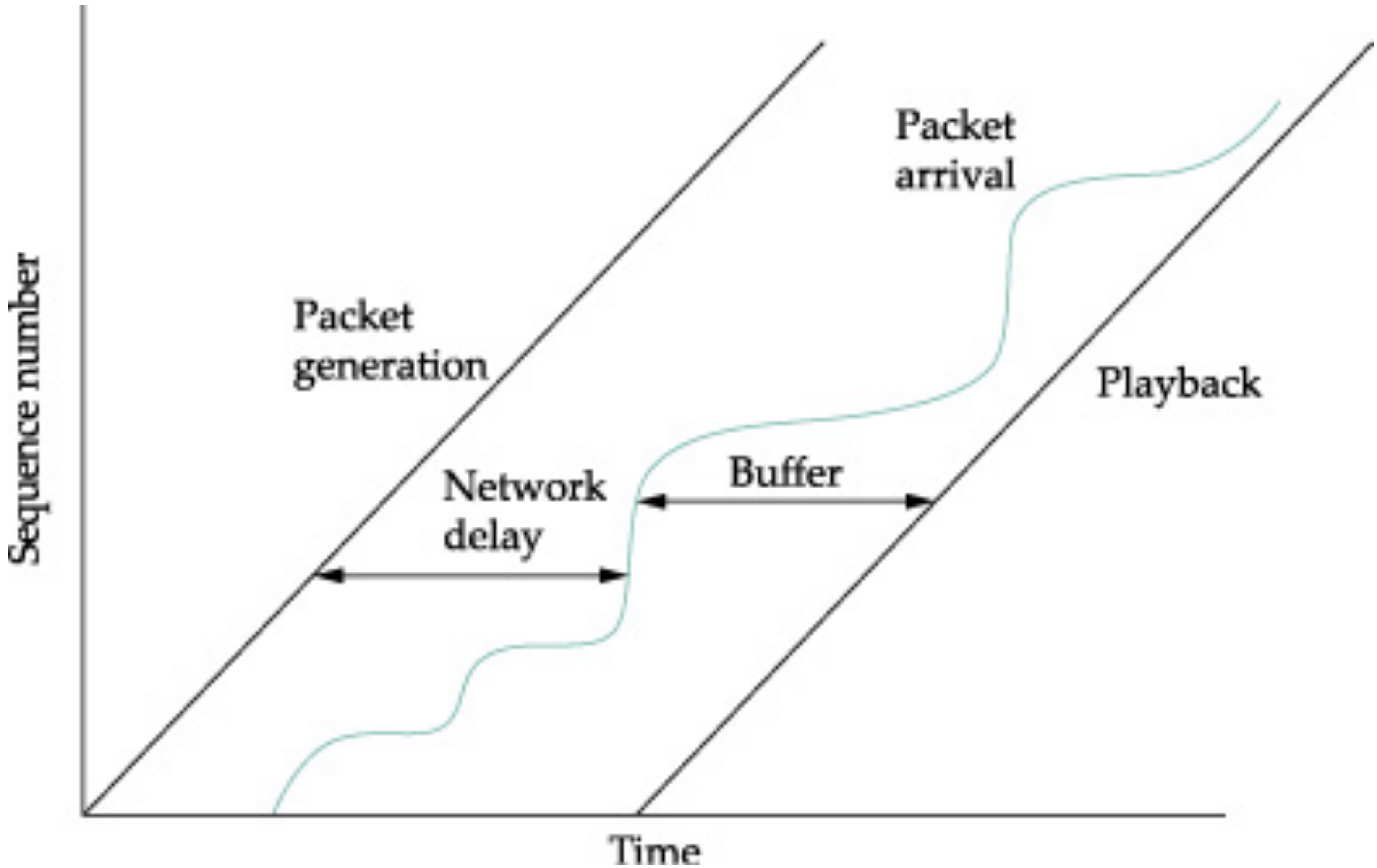


- Stay to the right and you are fine for all curves

Playback Applications

- Sample signal → packetize → transmit → buffer → playback
  - Fits most multimedia applications

- Performance concern:
  - Jitter: variation in end-to-end delay
    - Delay = fixed + variable = (propagation + packetization) + queuing
- Solution:
  - Playback point – delay introduced by buffer to hide network jitter



Sequence number
Packet generation
Network delay
Buffer
Packet arrival
Playback
Time

Characteristics of Playback Applications

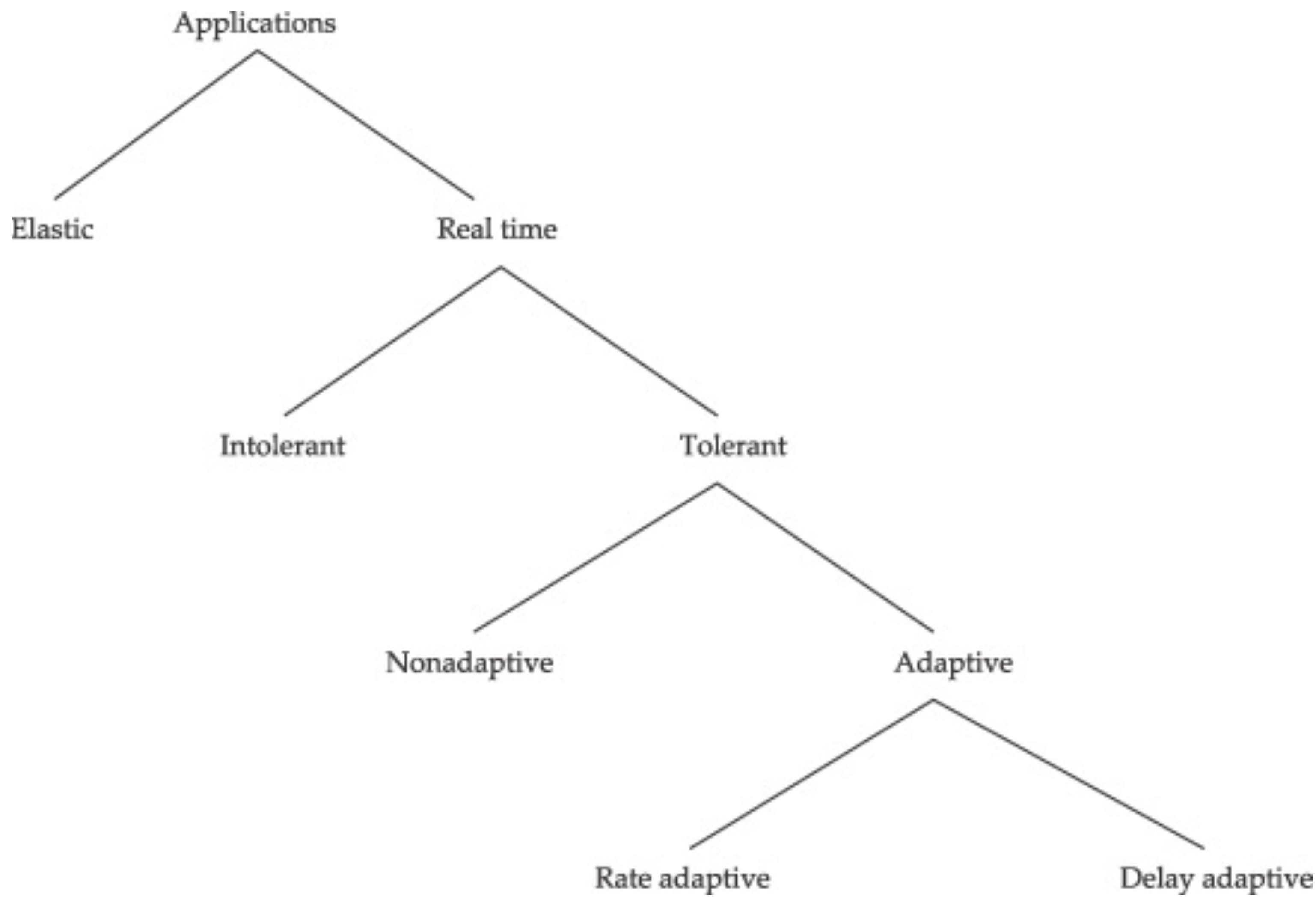
- In general lower delay is preferable
- Doesn't matter when packet arrives as long as it is before playback point
- Network guarantees (e.g., bound on jitter) would make it easier to set playback point
- Applications can tolerate some loss

Applications Variations

- Rigid and adaptive applications
  - Delay adaptive
    - Rigid: set fixed playback point
    - Adaptive: adapt playback point
      - E.g. Shortening silence for voice applications
  - Rate adaptive

- Loss tolerant and intolerant applications

- Four combinations


Applications
Elastic
Real time
Intolerant
Tolerant
Nonadaptive
Adaptive
Rate adaptive
Delay adaptive

Applications Variations

Really only two classes of applications

1) **Intolerant and rigid**
2) **Tolerant and adaptive**

Other combinations make little sense

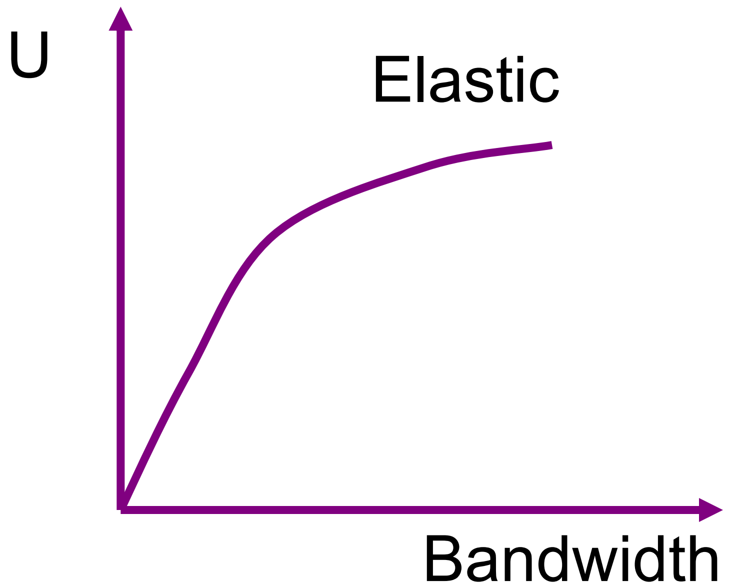
3) Intolerant and adaptive
   - Cannot adapt without interruption
4) Tolerant and rigid
   - Missed opportunity to improve delay

Design question 2: How to maximize $V = \sum U(s_i)$

- Choice #1: add more pipes

- Choice #2: fix the bandwidth but offer different services
  - Q: can differentiated services improve V?

If all users' utility functions are elastic



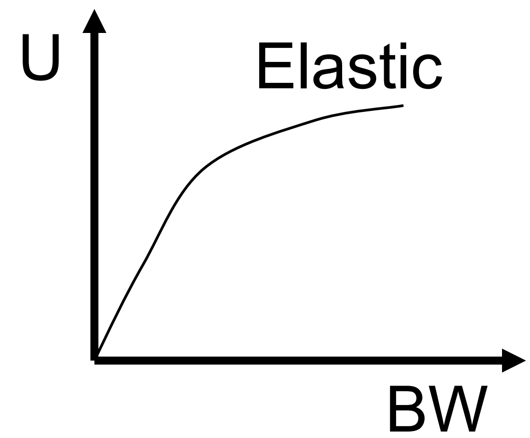
Does equal allocation of bandwidth maximize total utility?

- $\sum s_i = B$
- Max $\sum U(s_i)$

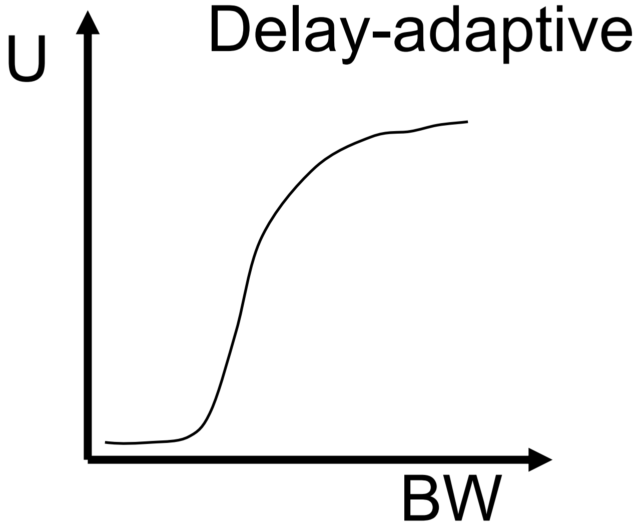
Design question: is Admission Control needed?

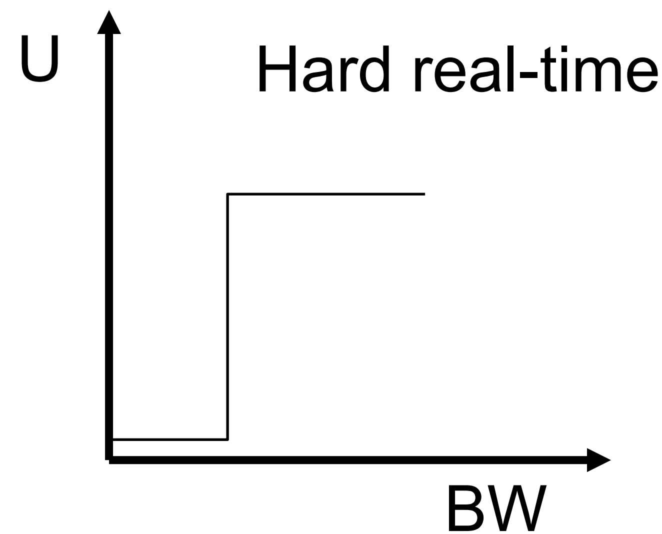
- If U(bandwidth) is concave
  → elastic applications
  - Incremental utility is decreasing with increasing bandwidth
    - $U(x) = \log(x^{p)}$
    - $V = n\log(B/n)^{p} = \log B^{p} n^{1-p}$
  - Is always advantageous to have more flows with lower bandwidth
    - No need of admission control;

  This is why the Internet works! And fairness makes sense



Utility Curves – Inelastic traffic

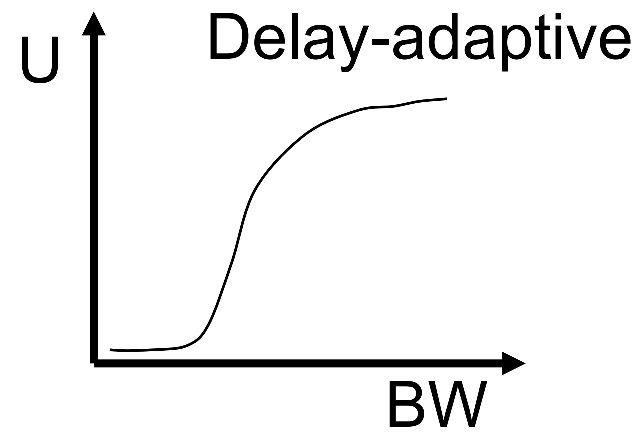




Does equal allocation of bandwidth maximize total utility?

Is Admission Control needed?

- If U is convex → inelastic applications
  - U(number of flows) is no longer monotonically increasing
  - Need admission control to maximize total utility
- **Admission control** → deciding when the addition of new people would result in reduction of utility
  - Basically avoids overload



Incentives

- Who should be given what service?
  - Users have incentives to cheat
  - Pricing seems to be a reasonable choice
  - But usage-based charging may not be well received by users

Over provisioning

- Pros: simple
- Cons
  - Not cost effective
  - Bursty traffic leads to a high peak/average ratio
    - E.g., normal users versus leading edge users
  - It might be easier to block heavy users

Comments

- End-to-end QoS has not happened
- Why?
- Can you think of any mechanism to make it happen?

Approaches to QoS

- Fine-grained:
  - Integrated services
    - RSVP

- Coarse-grained:
  - Differentiated services

Review

What this course is about?

- Computer network architecture
- “Architecture is both the process and the product of planning, designing, and constructing buildings and other physical structures.”
- “Architectural works, in the material form of buildings, are often perceived as cultural symbols and as works of art. ”
- https://en.wikipedia.org/wiki/Architecture

- By Dhirad, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=113235

- By Livioandronico2013 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=54296849

What is “network architecture”

- “Network architecture is the design of a communication network.”

- “It is a framework for the specification of a network's physical components and their functional organization and configuration, its operational principles and procedures, as well as data formats use.”

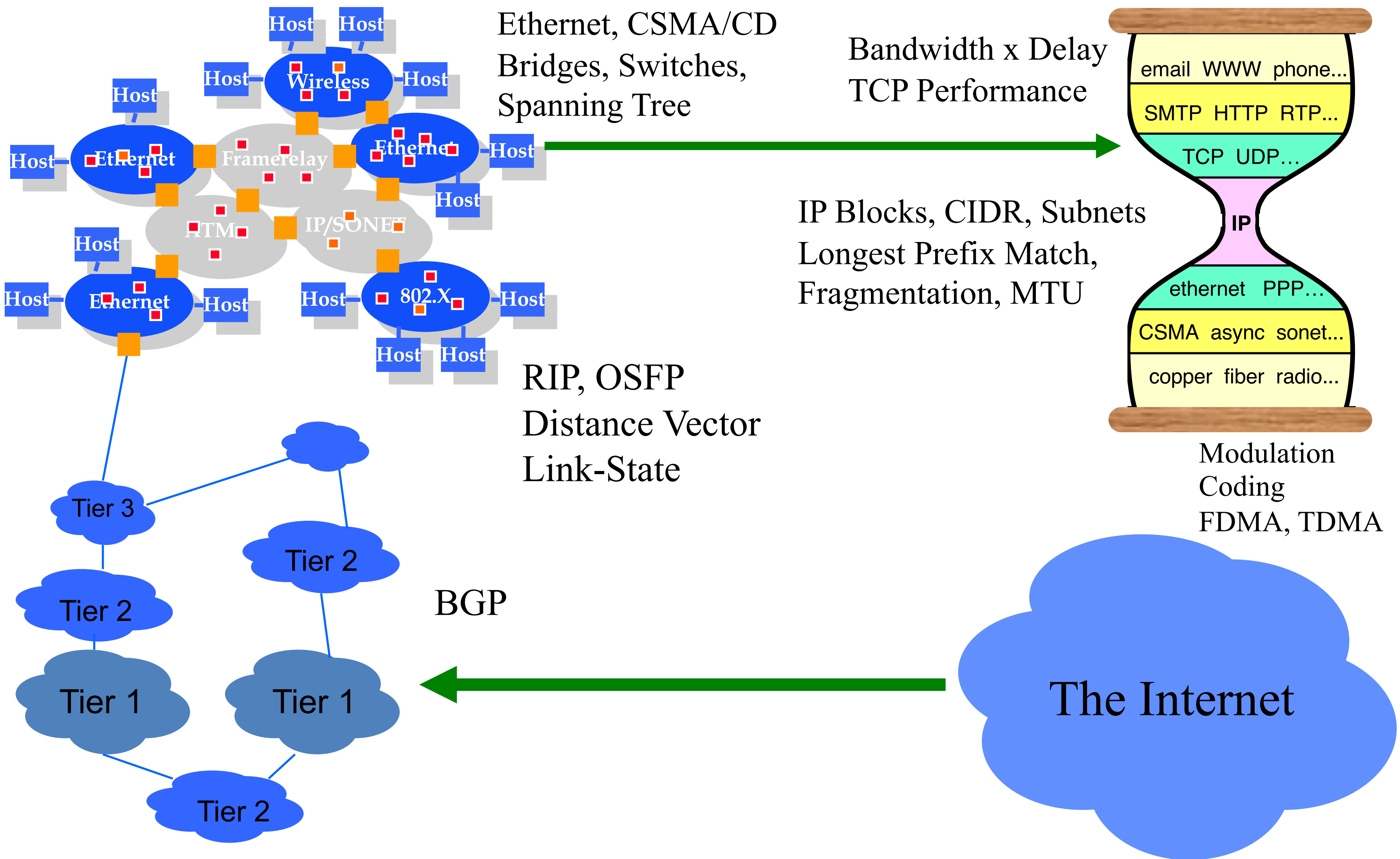
What I hope to teach

- The Internet architecture as an example of network architectures
  - Teach by examples
- Computer networks are just one type of computer systems
  - Share similar design principles

Challenges and solutions

- Complexity
  - Large number of components
  - Large number of interconnections
  - Many irregularities
  - Parties of conflicting interests
  - …
- Coping with complexity
  - Modularity, abstraction, layering, and hierarchy

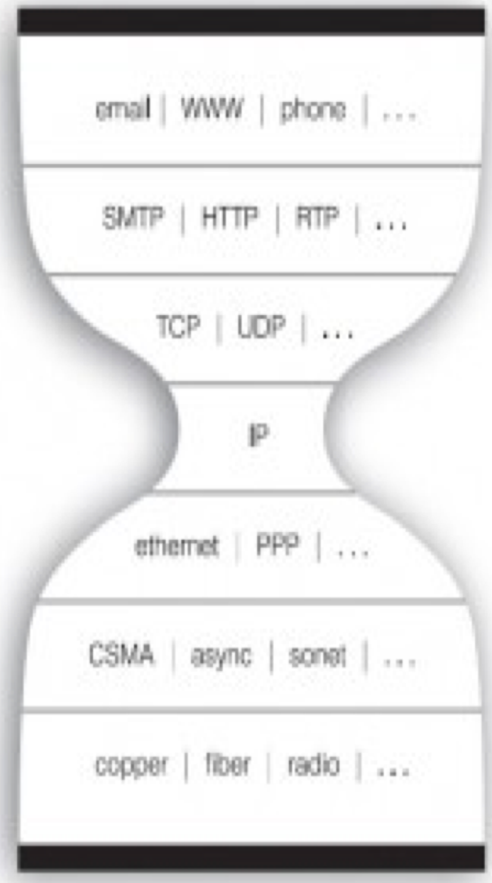
What the Internet looks like



Unique challenges faced by a communication network

- Scalability: how to support a large number of components
- Reliability: how to obtain reliable delivery from unreliable components
- Performance: low latency, high throughput
- Resource sharing among multiple users
- Routing and addressing

Functions/Concepts at different layers



Hourglass architecture of the Internet

Application protocols vs Applications

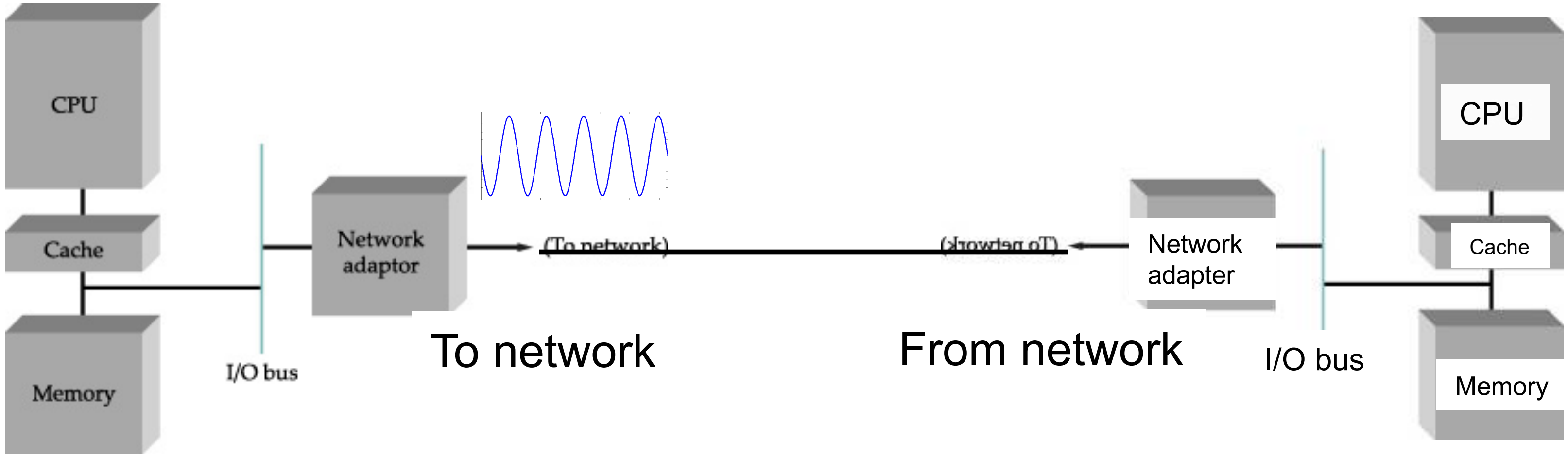
Reliable transport, multiplexing

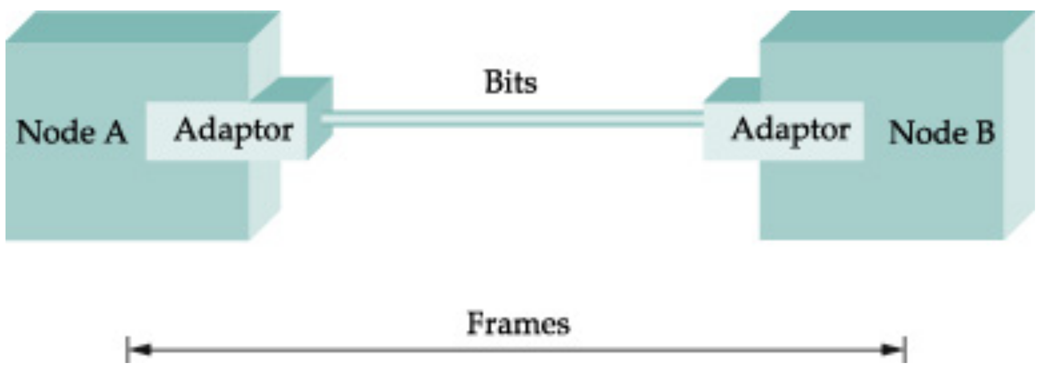
Forwarding, Routing, Addressing

Encoding, framing, error detection, reliability
media sharing, switching

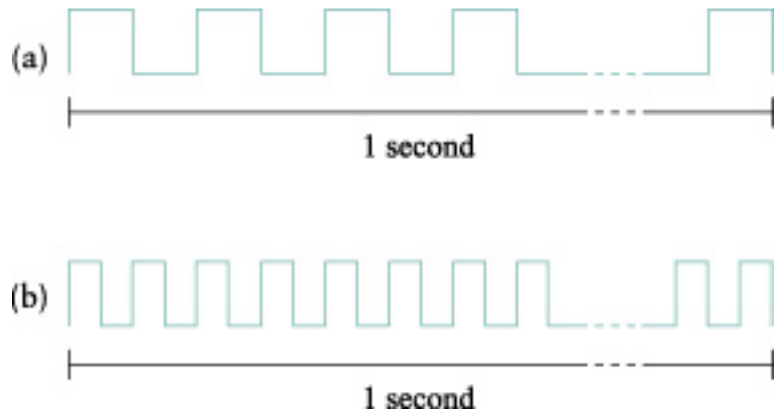
Bandwidth, latency, throughput,
delay-bandwidth product

Physical properties of a link





Bandwidth



- Bandwidth of a link refers to the number of bits it can transmit in a unit time
  - A second of time as distance
  - Each bit as a pulse of width

- Fast link
- Slow link

Latency to transmit a packet

- Has four components
  - Link propagation delay
  - Transmission/serialization latency
  - Queuing delay
  - Processing delay (often ignored)

Round trip time (RTT)

- Time to send a packet and receive an acknowledgement

How to determine the “optimal” sliding window size

- What does “keep the pipe full” mean?

Mechanisms at Different layers

- Link layer
  - Encoding
    - NRZ, NRZI, Manchester, 4B/5B
  - Framing
    - Byte-oriented, bit-oriented, time-based
    - Bit stuffing
  - Error detection
    - Parity, checksum, CRC
  - Reliability
    - FEC, sliding window

Link layer continued

- Multi-access link
  - Ethernet
    - Collision Sense Multiple Access/Collision Detection (CSMA/CD)
  - WIFI
    - Carrier-sense multiple access with collision avoidance (CSMA/CA)
    - Cannot send and receive at the same time
    - Must send when channel is idle
    - RTS/CTS

Link layer continued

- Virtual circuit switching
  - ATM
- Datagram switching
  - Ethernet learning bridges
    - Spanning tree algorithm
- Source routing

The network layer

- The Internet Protocol
- Classless Interdomain Routing (CIDR)
  - Addressing format
  - Subnet, network prefix
- Forwarding
  - Longest prefix matching

The network layer continued

- Routing
  - Distance vector
  - Link state
  - BGP

- Auxiliary functions
  - ARP, ICMP, DHCP, NAT, IP Tunnel

- Multicast

The transport layer

- UDP
  - Datagram, connectionless, multiplexing multiple applications
- TCP
  - Reliable, byte stream

TCP

- Connection establishment
- Reliability
  - Sliding window
  - Loss recovery
    - Time out, duplicate acks, selective ACKs
- Flow control
- Congestion control and avoidance

TCP congestion control

1. Probing for the available bandwidth
   – slow start (cwnd < ssthresh)

2. Avoid overloading the network
   – congestion avoidance (cwnd >= ssthresh)

Slow Start

- Initial value: **Set cwnd = 1 MSS**
  - Modern TCP implementation may set initial cwnd to 2
- When receiving an ACK, cwnd+= 1 MSS
  - If an ACK acknowledges two segments, cwnd is still increased by only 1 segment.
  - Even if ACK acknowledges a segment that is smaller than MSS bytes long, cwnd is increased by 1.
  - Question: how can you accelerate your TCP download?

Congestion Avoidance

- If cwnd >= ssthresh then each time an ACK is received, increment cwnd as follows:
  - cwnd += MSS * (MSS / cwnd) (cwnd measured in bytes)

- So *cwnd* is increased by one MSS only if all *cwnd*/MSS segments have been acknowledged.

Slow Start

- Initial value: Set **cwnd = 1 MSS**
  - Modern TCP implementation may set initial cwnd to 2
- When receiving an ACK, cwnd+= 1 MSS
  - If an ACK acknowledges two segments, cwnd is still increased by only 1 segment.
  - Even if ACK acknowledges a segment that is smaller than MSS bytes long, cwnd is increased by 1.
  - Question: how can you accelerate your TCP download?

Congestion Avoidance

- If cwnd >= ssthresh then each time an ACK is received, increment cwnd as follows:
    - cwnd += MSS * (MSS / cwnd) (cwnd measured in bytes)

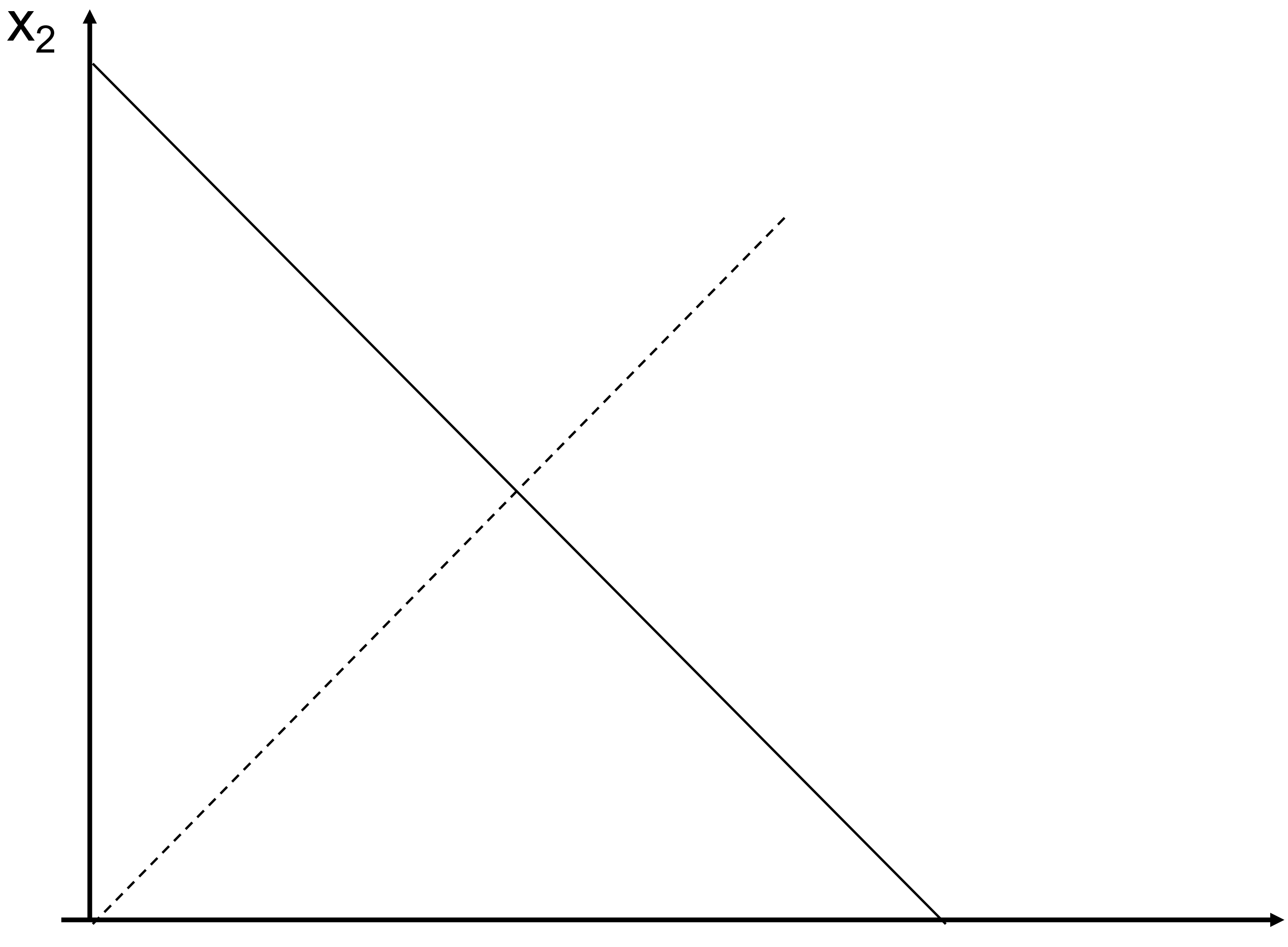
- So *cwnd* is increased by one MSS only if all *cwnd*/MSS segments have been acknowledged.

TCP congestion control modeling

$$x_i(t+1) = \begin{cases} a_{\mathrm{I}} + b_{\mathrm{I}} x_i(t) & \text{if } y(t) = 0 \Rightarrow \text{Increase}, \\ a_{\mathrm{D}} + b_{\mathrm{D}} x_i(t) & \text{if } y(t) = 1 \Rightarrow \text{Decrease}. \end{cases}$$

- Four sample types of controls
- AIAD, AIMD, MIAD, MIMD

Phase plot

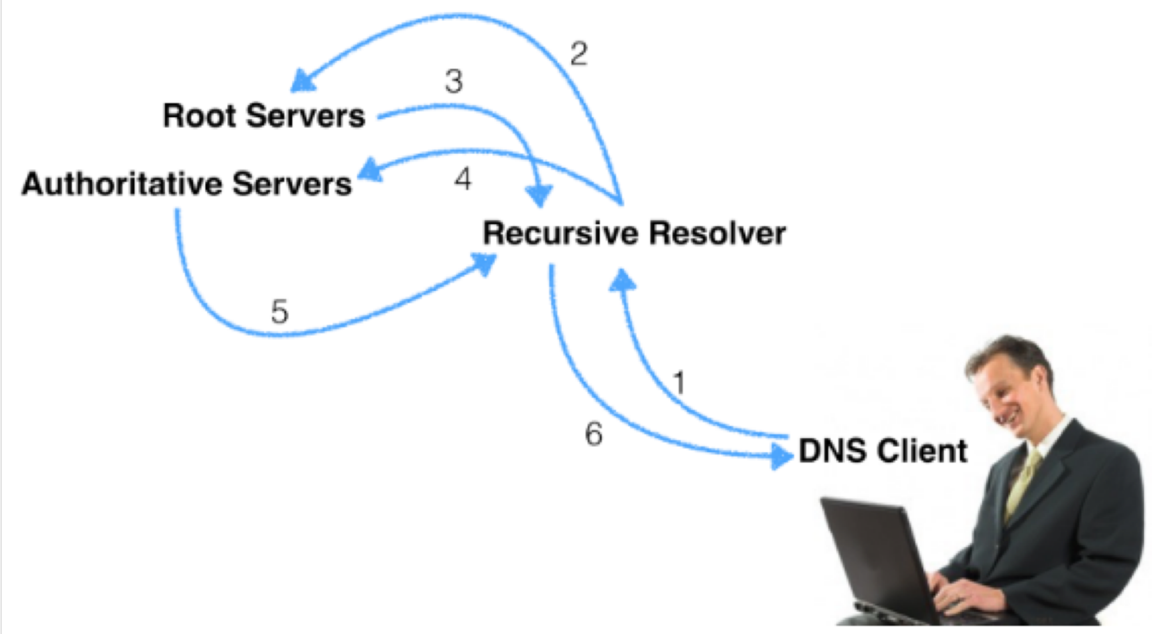


Application layer

- Domain Name System (DNS)
- Socket interface
- Application protocols vs applications
  - Email
- Overlay networks

Figure out the DNS server hierarchy

- dig +norecurse @a.root-servers.net NS www.cnn.com



- https://ns1.com/articles/using-dig-trace
- dig +trace www.cnn.com
- dig +trace turner-tls.map.fastly.net.

Looking forward

- Graduate networking class
  - Datacenter networking
  - Future Internet architectures
  - SDN, NFV

What to expect in the final

- Networking knowledge
  - Short questions, true or false questions
  - Computation questions
- Application of networking knowledge
  - Design knowledge

Pop quiz and course evaluation