CSCI-1680
Wrap-up Lecture

Chen Avin

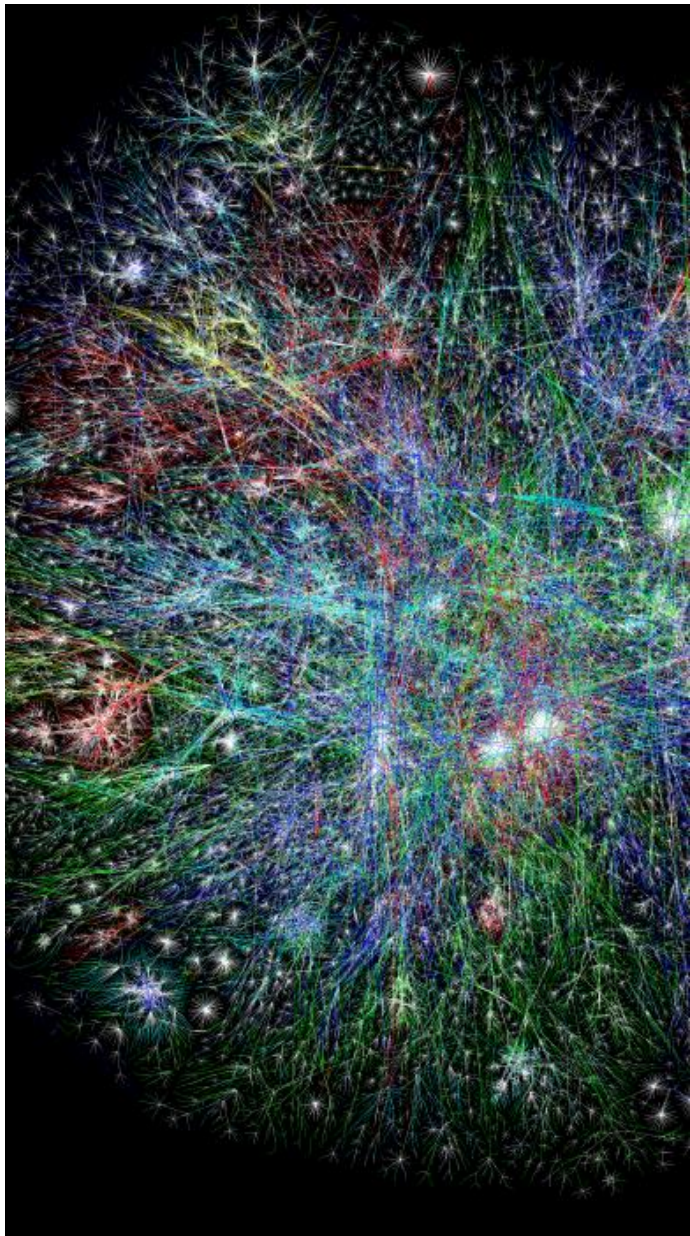
With some material from Jen Rexford

Administrivia

- **Today is the last class!**
- **Two more things to go:**
  - Final project, due this Friday
  - Final Exam: Thursday, Dec 17th, 2pm
- **How do you study?**
  - Any covered topic is fair game, but more emphasis on content given *after* midterm (TCP on)
  - Lecture slides, homeworks, plus relevant sections of the book
  - If in doubt, no topic not covered in class will be on the exam

What you (hopefully) learned from this course

- **Skill: Network programming**
  - C programming (most of you)
  - Socket programming
  - Server programming
  - Implementing protocols
- **Knowledge: How the Internet Works**
  - IP Protocol suite
  - Internet Architecture
  - Applications (Web, DNS, P2P, …)
- **Insight: key concepts**
  - Protocols
  - Layering
  - Naming



Introduction

- What is the Internet?
- Network edge
- Network core
- Network of networks
- Internet structure and ISPs
- Delay & loss in packet-switched networks
- Protocol layers, service models
- History of the Internet

Physical Layer

- **Modulation**
- **Encoding**

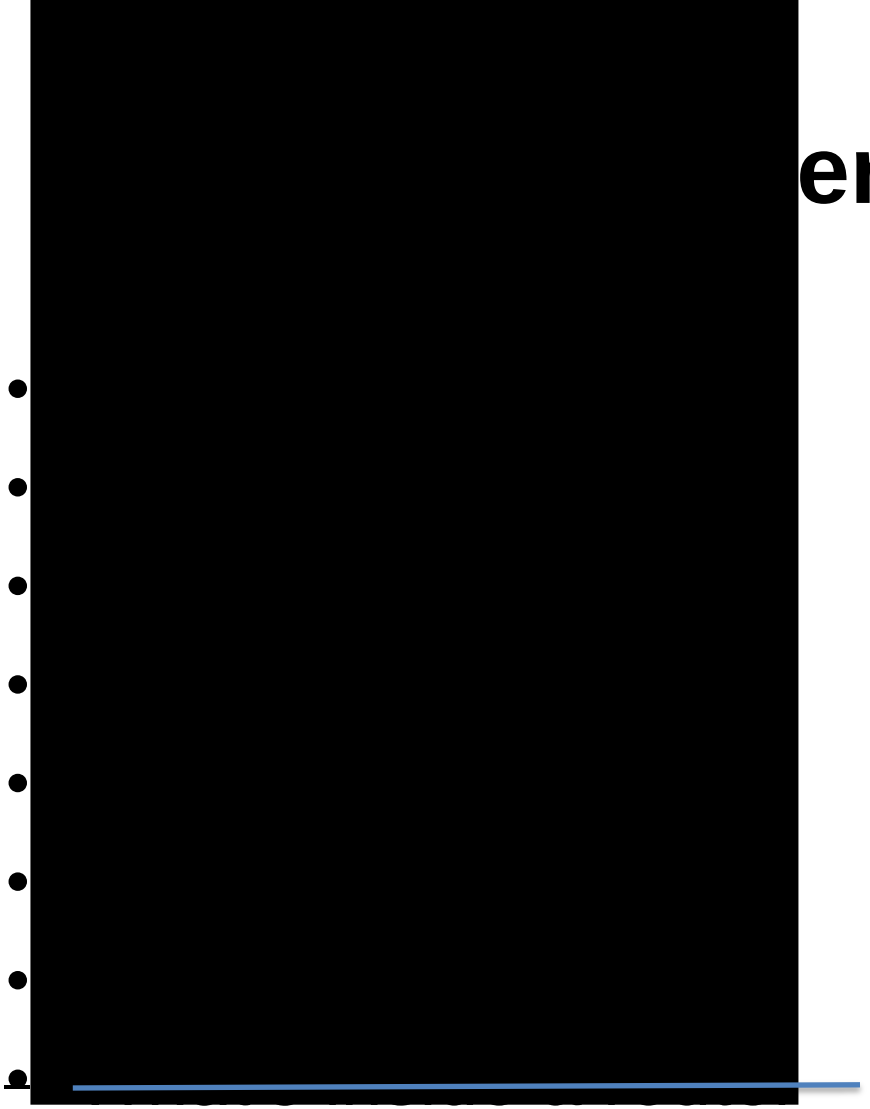
Link Layer

- **Framing**
- **Errors, reliability, performance**
- **Sliding window**
- **Medium Access Control (MAC)**
- **Case study: Ethernet**
- **Link Layer Switching**
  - STP

Network Layer - IP

- Philosophy and Overview
- Forwarding and Routing
- IPv4 Datagram format
- IPv4 addressing - CIDR
- ARP, DHCP, ICMP, NAT
- Tunneling
- IPv6

er - Routing



Network Layer - More

- **Multicasting**
  - Spanning tree
  - RPF
- **Mobile IP**
- **SDN**

Transport Layer

- Transport layer services
- Multiplexing/demultiplexing
- UDP
- Reliable data Transfer
- TCP

Transport Layer - TCP

- **Segment structure**
- **Reliable data transfer**
- **Flow control**
- **Connection management**
- **Congestion control**
- **Congestion avoidance**

Application layer

- Principles of network applications
- Web and HTTP
- Electronic Mail
- SMTP, POP3, IMAP
- DNS
- P2P applications
- Socket Programming

Wireless Networking

- **Background**
- **Wireless Link Characteristics**
- **IEEE 802.11 Wireless LAN**
- **MAC Protocol: CSMA/CA**
- **Mobility**
  - Direct and Indirect routing

Security

- **Classes of attacks**
- **Basic security requirements**
  - Confidentiality
  - Integrity
  - Authentication
  - Provenance
- **Simple cryptographic methods**
- **Cryptographic toolkit (Hash, Digital Signature, …)**
- **Certificate Authorities**
- **SSL / HTTPS**

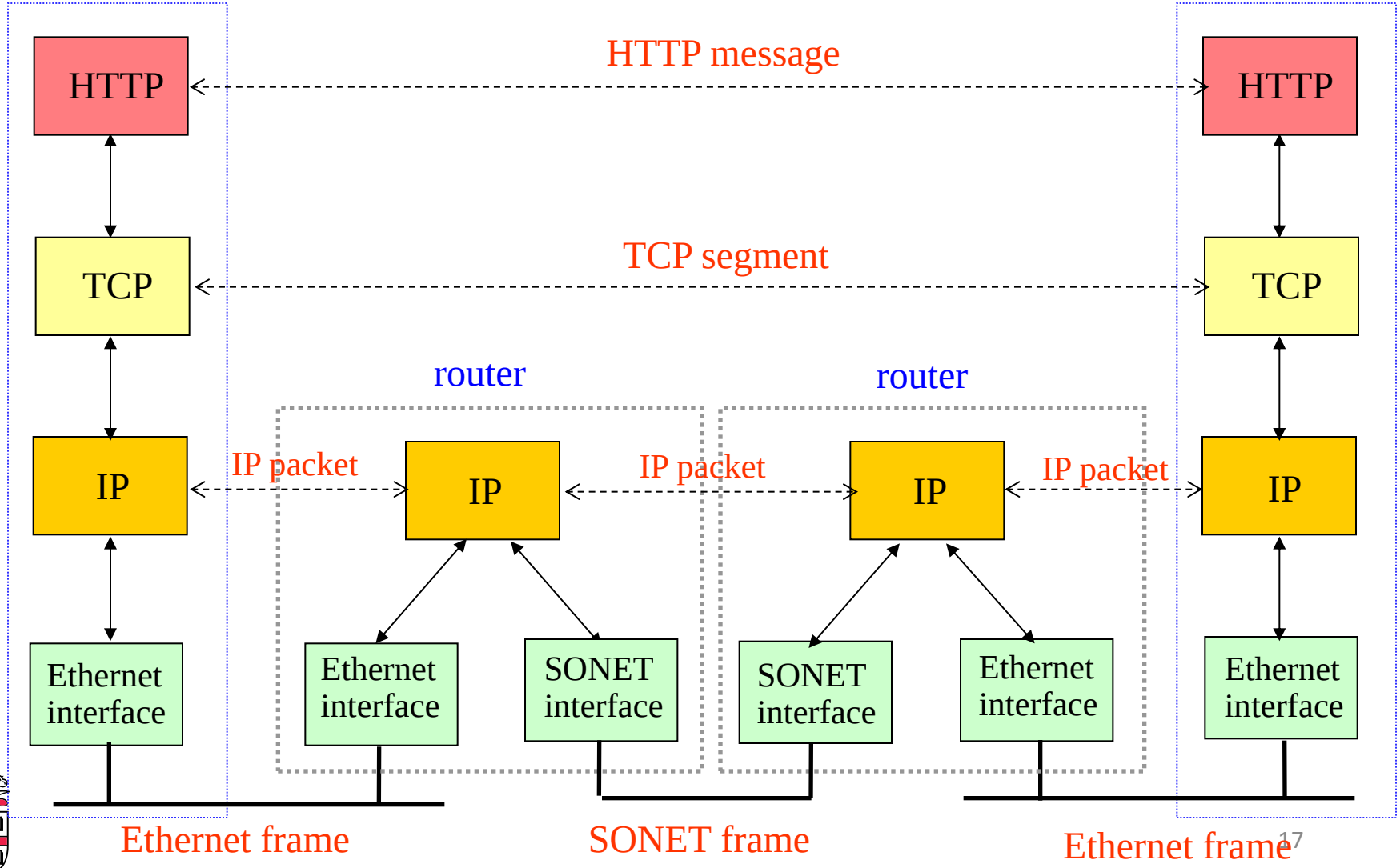
Networking Principles

- **We saw many layers and protocols, but some principles are common to many**
- **Some are general CS concepts**
  - Hierarchy
  - Indirection
  - Caching
  - Randomization
- **Some are somewhat networking-specific**
  - Layering
  - Multiplexing
  - End-to-end argument
  - Soft-state

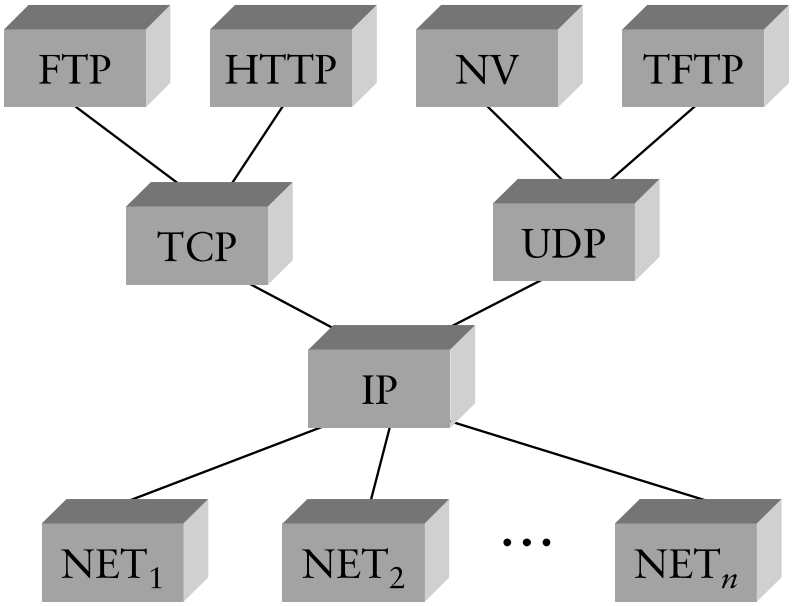
Layering

- **Strong form of encapsulation, abstraction**
- **Each layer has *three* interfaces:**
  - Services provided to upper layer
  - Protocol to communicate with peer at the same layer
  - Using the services of the lower layer
- **Provided interface hides all details of internal interface and lower layers**
- **Can be highly recursive**
  - E.g., IP over DNS, File system over Gmail!

Layering on the Internet



Layering: IP as a Narrow Waist



- **Many applications protocols on top of UDP & TCP**
- **IP works over many types of networks**
- **This is the "Hourglass" architecture of the Internet.**
  - If every network supports IP, applications run over many different networks (*e.g.*, cellular network)

Layering: Data Encapsulation

- **One layer's data is the (opaque) payload of the next**

Stream (Application)

Segments (TCP)

Packets (IP)

Frames (Ethernet)

Encoding: bits -> chips

Modulation: chips -> signal variations

Multiplexing: Access

- **Sharing a single channel**
- **E.g.,**
  - NAT: multiple nodes share a single IP address
    - De-multiplexing: NAT uses 5-tuple to disambiguate
  - SSH port forwarding
    - Only port 22 is open, can tunnel other ports
    - ssh other.host.com –L 5900:other.host.com:5900
  - VPN

Multiplexing: Reuse

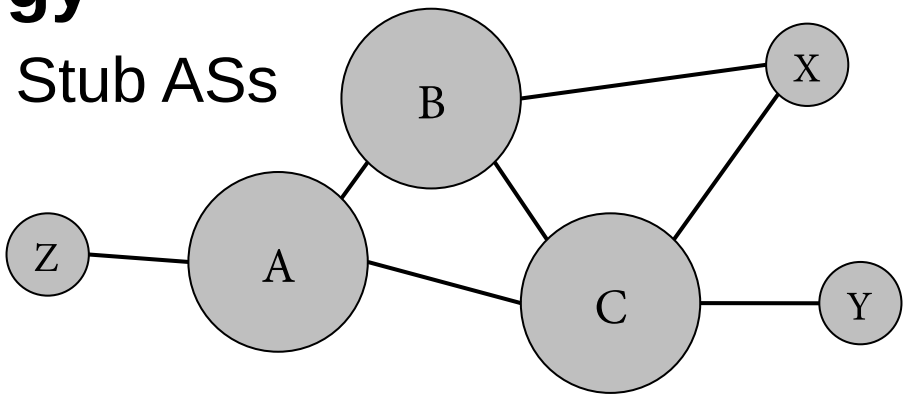
- **No need to re-implement functionality**
  - Several streams/flows can use the services of a protocol
- **E.g.:**
  - IP/ARP/AppleTalk on Ethernet: demux EtherType
  - TCP/UDP/DCCP/… on IP: demux Protocol ID
  - HTTP/SIP/SMTP/… on TCP/UDP: demux on Port
  - Multiple hosts on one HTTP server: demux on Host: field

Hierarchy Examples: IP Routing

- **IP Addressing**
  - Hierarchical assignment of address blocks
  - IANA -> Regional Internet Registries -> ISPs
  - Decentralized *control*
- **Topology**
  - (Roughly) correlated with addressing
  - Allows aggregation (CIDR)
    - Brown owns 128.148.0.0/16
  - Decreases size of routing tables!

Hierarchy Examples: IP Routing

- **AS-level Topology**
  - Separates intra and inter-domain routing
  - ASs have own economic interests
  - Delegation of control
    - Policy in inter-domain routing
    - Complete control of intra-domain routing
- **Hierarchical Topology**
  - Transit, Multi-homed, Stub ASs



Hierarchy Examples: DNS

- **Hierarchical name database**
- **Allows delegation of control**
  - Each organization controls a sub-tree
  - May delegate control
- **Allows scaling of the infrastructure**
  - A DNS server only needs to know about its sub-domains

Many Translations

- **DHCP: Given a MAC Address, assign an IP address**
  - Uses IP broadcast to find server
- **ARP: Given an IP address, find Ethernet MAC Addresses**
  - Uses Link Layer broadcast to find node
- **DNS: Given a Name, find an IP address**
  - Uses IP unicast/anycast to well known roots, to bootstrap
  - Relies on IP routing infrastructure, DNS hierarchy
- **DHT: Given a key, find a node**
  - Uses IP unicast plus efficient flat namespace routing

Caching

- **Duplicate data stored elsewhere**
  - Reduce latency for accessing the data
  - Reduce the load on other parts of the system
- **Often quite effective**
  - Locality of reference: temporal locality and small set of popular items
- **Examples:**
  - Web caching
  - DNS caching
  - ARP caching
  - Learning bridges

DNS Caching

- **What is cached?**
  - Mapping of names to IP addresses
  - Lookups that failed
  - IP addresses of name servers
- **Reduces latency**
- **Reduces load on hierarchy**
- **Why is it effective?**
  - Mostly read database
  - Doesn’t change very often
  - Popular sites are visited often

HTTP Caching

- **What is cached?**
  - Web objects
- **Where is it cached?**
  - Browser, proxy-cache, main memory on server
- **Reduces latency, load**
- **What contributes to high hit rates?**
  - Cacheable content (mostly static)
  - Sharing the cache among multiple users
  - Small amount of popular content

Randomization

- **Distributed adaptive algorithms**
- **Risk of synchronization**
  - Many parties respond to the same conditions in the same way
  - May lead to bad aggregate behavior
- **Randomization can de-synchronize**
  - Example: Ethernet backoff mechanism
  - Example: Random Early Drop

Interesting (extra) read: “The Synchronization of Periodic Routing Messages”, Sally Floyd and Van Jacobson, Sigcomm 1993

Soft State

- **State is stored in nodes by network protocols**
  - E.g., a mapping, routing entry, cached object
- **Key issue: how to deal with changes?**
- **Hard state: "valid unless told otherwise"**
  - "Managed" by originator of state
  - Kept consistent, explicit invalidation
- **Soft state: "valid if fresh"**
  - Removed by storing node on *timeout*
  - Periodically refreshed as needed
    - May need extra cost (on-demand revalidation or check)
  - Can be seen as a hint
- **Soft state reduces complexity**
  - At the cost of some unpredictability

Soft state examples

- **DNS Caching**
  - TTL
  - Can be wrong, check with origin on error
- **Alternative**
  - Origin keeps track of copies
  - Refresh copies on change in mapping
- **Cache coherence is hard**
  - And expensive at scale!
- **Others**
  - DHCP lease

But… There are BIG Challenges

- **Designed in a different environment, with different uses**
  - Identity / Accountability
  - Access model
  - Security
  - Challenges to openness

Thank you and Good Luck!

And see you around….