Chapter 1
Introduction
Chapter 1: Introduction

Our goal:

❖ get “feel” and terminology
❖ more depth, detail later in course
❖ approach:
  ▪ use Internet as example

Overview:

❖ what’s the Internet?
❖ what’s a protocol?
❖ network edge; hosts, access net, physical media
❖ network core: packet/circuit switching, Internet structure
❖ performance: loss, delay, throughput
❖ security
❖ protocol layers, service models
❖ history
Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network *edge*
   - end systems, access networks, links

1.3 Network *core*
   - *circuit switching, packet switching*, network structure

1.4 Delay, loss and *throughput* in packet-switched networks

1.5 Protocol layers, service models

1.6 Networks under attack: security

1.7 History
What's the Internet: "nuts and bolts" view

- 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)
“Fun” internet appliances (Internet of Things)

- IP picture frame
  http://www.ceiva.com/
- Web-enabled toaster + weather forecaster
- Slingbox: watch, control cable TV remotely
- Internet refrigerator
- Internet phones

Introduction 1-5
What’s the Internet: “nuts and bolts” view

- **protocols** control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, Skype, Ethernet

- **Internet: “network of networks”**
  - loosely hierarchical
  - public Internet versus private intranet

- **Internet standards**
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force
What’s the Internet: a service view

- **communication infrastructure** enables distributed applications:
  - Web, VoIP, email, games, e-commerce, file sharing

- **communication services** provided to apps:
  - reliable data delivery from source to destination
  - “best effort” (unreliable) data delivery
What’s a protocol?

**human protocols:**
- “what’s the time?”
- “I have a question”
- introductions

... specific msgs sent
... specific actions taken when msgs received, or other events

**network protocols:**
- machines rather than humans
- all communication activity in Internet governed by protocols

protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt
What's a protocol?

A human protocol and a computer network protocol:

Q: Other human protocols?
Chapter 1: roadmap

1.1 What is the Internet?
1.2 Network edge
   ✷ end systems, access networks, links
1.3 Network core
   ✷ circuit switching, packet switching, network structure
1.4 Delay, loss and throughput in packet-switched networks
1.5 Protocol layers, service models
1.6 Networks under attack: security
1.7 History
A closer look at network structure:

- **network edge:** applications and hosts
- **access networks, physical media:** wired, wireless communication links
- **network core:**
  - interconnected routers
  - network of networks
The network edge:

❖ end systems (hosts):
  - run application programs
  - e.g. Web, email
  - at “edge of network”

❖ client/server model
  - client host requests, receives service from always-on server
  - e.g. Web browser/server; email client/server

❖ peer-peer model:
  - minimal (or no) use of dedicated servers
  - e.g. Skype, BitTorrent
Access networks and physical media

Q: How to connect end systems to edge router?

- residential access nets
- institutional access networks (school, company)
- mobile access networks

Keep in mind:
- *bandwidth* (bits per second) of access network?
- shared or dedicated?
Dial-up Modem

- uses existing telephony infrastructure
  - home directly-connected to central office
- up to 56Kbps direct access to router (often less)
- can’t surf, phone at same time: not “always on”
Digital Subscriber Line (DSL)

- uses existing telephone infrastructure
- up to 1 Mbps upstream (today typically < 256 kbps)
- up to 8 Mbps downstream (today typically < 1 Mbps)
- dedicated physical line to telephone central office

Existing phone line:
- 0-4KHz phone; 4-50KHz upstream data;
- 50KHz-1MHz downstream data

DSLAM: DSL Access Multiplexer

- DSL: Digital Subscriber Line
  - uses existing telephone infrastructure
  - up to 1 Mbps upstream (today typically < 256 kbps)
  - up to 8 Mbps downstream (today typically < 1 Mbps)
  - dedicated physical line to telephone central office
Residential access: cable modems

- uses cable TV infrastructure, rather than telephone infrastructure
- HFC: hybrid fiber coax
  - asymmetric: up to 30Mbps downstream, 2 Mbps upstream
- network of cable, fiber attaches homes to ISP router
  - homes share access to router
  - unlike DSL, which has dedicated access
Residential access: cable modems

Diagram: http://www.cabledatacomnews.com/cmic/diagram.html
Cable Network Architecture: Overview
Cable Network Architecture: Overview

cable headend

cable distribution
network (simplified)

home

setting Top Box

Home Environment

TV

Cable Modem

PC

Coax

Splitter

10 Mbps Ethernet
Optical links from central office to the home

Two competing optical technologies:
- Passive Optical network (PON)
- Active Optical Network (PAN)

Much higher Internet rates; fiber also carries television and phone services
**Ethernet Internet access**

- Typically used in companies, universities, etc.
- 10 Mbps, 100 Mbps, 1 Gbps, 10 Gbps, 100 Gbps Ethernet
- Today, end systems typically connect into Ethernet switch

![Diagram of Ethernet Internet access](image)
Wireless access networks

- shared wireless access network connects end system to router
  - via base station aka “access point”
- wireless LANs:
  - 802.11b/g (WiFi): 11 or 54 Mbps
- wider-area wireless access
  - provided by telco operator
  - ~1Mbps over cellular system (EVDO, HSDPA)
  - next up (?): WiMAX (10’s Mbps) over wide area
Home networks

Typical home network components:

- DSL or cable modem
- router/firewall/NAT
- Ethernet
- wireless access point
Physical Media

- **bits**: propagate between transmitter/rcvr pairs
- **physical link**: what lies between transmitter & receiver
- **guided media**:
  - signals propagate in solid media: copper, fiber, coax
- **unguided media**:
  - signals propagate freely, e.g., radio

**Twisted Pair (TP)**

- **two insulated copper wires**
  - Category 3: traditional phone wires, 10 Mbps Ethernet
  - Category 5: 100Mbps Ethernet
Physical Media: coax, fiber

Coaxial cable:
- two concentric copper conductors
- bidirectional
- baseband:
  - single channel on cable
  - legacy Ethernet
- broadband:
  - multiple channels on cable
  - HFC

Fiber optic cable:
- glass fiber carrying light pulses, each pulse a bit
- high-speed operation:
  - high-speed point-to-point transmission (e.g., 10's-100's Gbps)
- low error rate: repeaters spaced far apart; immune to electromagnetic noise
Physical media: radio

- signal carried in electromagnetic spectrum
- no physical “wire”
- bidirectional
- propagation environment effects:
  - reflection
  - obstruction by objects
  - interference

Radio link types:

- terrestrial microwave
  - e.g. up to 45 Mbps channels
- LAN (e.g., WiFi)
  - 11Mbps, 54 Mbps
- wide-area (e.g., cellular)
  - 3G cellular: ~ 1 Mbps
- satellite
  - Kbps to 45Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay
  - geosynchronous versus low altitude
Chapter 1: roadmap

1.1 What is the Internet?
1.2 Network edge
   ❖ end systems, access networks, links
1.3 Network core
   ❖ circuit switching, packet switching, network structure
1.4 Delay, loss and throughput in packet-switched networks
1.5 Protocol layers, service models
1.6 Networks under attack: security
1.7 History
The Network Core

- mesh of interconnected routers/switches
- the fundamental question: how is data transferred through net?
  - circuit switching: dedicated circuit (end-2-end path through network) per call: telephone net
  - packet-switching: data sent thru net in discrete “chunks” (no dedicated end-2-end resources)
Network Core: Circuit Switching

end-end resources reserved for “call”

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

3 Phases-
- Set-up
- Transfer
- Disconnect
Network Core: Circuit Switching

- network resources (e.g., bandwidth, time) divided into “pieces”
- pieces allocated to calls
- resource piece *idle* if not used by owning call (no sharing)

- dividing link bandwidth into “pieces”
  - frequency division multiplexing
  - time division multiplexing
Public Circuit Switched Network
Circuit Switching: FDM and TDM

Freq. Division Multiplexing

Time Division Multiplexing (TDM)

Example:
4 users

Introduction 1-32
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: 1.536 Mbps
- each link uses TDM with 24 slots/sec
- 500 msec to establish end-to-end circuit

Let’s work it out!

Repeat Problem with TDM → FDM
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: 1.536 Mbps
- each link uses TDM with 24 slots/sec \(\iff\) frame length = 1 sec.
- 500 msec to establish end-to-end circuit

Transmission delay = \((640,000/1.536\times10^6)\) sec

= 0.417 sec.

1 slot = 1/24 = 0.0417 sec.

# TDM frames required to send the whole file = 10!

Hence delay = 10 sec. + 0.5 (set-up) = 10.5 sec.
Network Core: Packet Switching

- Each end-end data stream divided into packets
  - User A, B packets share network resources
  - Each packet uses full link bandwidth
  - Resources used as needed

Resource contention:
- Aggregate resource demand can exceed amount available
- **Congestion**: Packets queue, wait for link use
- **Store and forward**: Packets move one hop at a time
  - Node receives complete packet before forwarding

Bandwidth division into “pieces”
- Dedicated allocation
- Resource reservation

Introduction 1-35
Packet Switched Network
Datagram Network

(Datagram = Packet Switched)

Diagram showing the flow of datagrams through a network.
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.
Inside a packet switch/router
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)

**Example:**
- $L = 7.5$ Mbits
- $R = 1.5$ Mbps
- transmission delay = 15 sec

more on delay shortly ...
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
- packet switching:
  - with 35 users, probability
    > 10 active at same time
    is less than .0004

Q: how did we get value 0.0004?
Q: what happens if > 35 users?
Packet switching versus circuit switching

Is packet switching a “slam dunk winner?”

❖ **great for bursty data**
  - resource sharing
  - simpler, no call setup

❖ **but during congestion:** packet delay and loss
  - protocols needed for reliable data transfer, congestion control

❖ **Q: How to provide circuit-like behavior?**
  - Quality of Service guarantees needed for audio/video apps (media streaming, real-time interactive apps like games)
  - still an unsolved problem (chapter 7)
Chapter 1: roadmap

1.1 What is the Internet?
1.2 Network edge
   - end systems, access networks, links
1.3 Network core
   - circuit switching, packet switching, network structure
1.4 Delay, loss and throughput in packet-switched networks
1.5 Protocol layers, service models
1.6 Networks under attack: security
1.7 History
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

```
free (available) buffers: arriving packets dropped (loss) if no free buffers
```

```
packet being transmitted (delay)
```

```
packets queueing (delay)
```
Four sources of packet delay

- nodal processing
  - check bit errors
  - determine output link
  - typically $\ll$ msec

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

$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$
Four sources of packet delay

\[ d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}} \]

\( d_{\text{trans}} \): transmission delay:
- L: packet length (bits)
- R: link bandwidth (bps)
- \( d_{\text{trans}} = L/R \)

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

\( d_{\text{trans}} \) and \( d_{\text{prop}} \) very different
Caravan analogy (packet = caravan)

- cars “propagate” at 100 km/hr
- toll booth takes 12 sec to service car (transmission time)
- car~bit; caravan ~ packet
- Q: How long until caravan is lined up before 2nd toll booth?
  - time to “push” entire caravan through toll booth onto highway = 12*10 = 120 sec
  - time for last car to propagate from 1st to 2nd toll both:
    100km/(100km/hr)= 1 hr
  - A: 62 minutes
Caravan analogy (more)

- cars now “propagate” at 1000 km/hr
- toll booth now takes 1 min to service a car
- Q: Will cars arrive at 2nd booth before all cars serviced at 1st booth?
  - A: Yes! After 7 min, 1st car arrives at second booth; three cars still at 1st booth.
  - 1st bit of packet can arrive at 2nd router before packet is fully transmitted at 1st router! (see Ethernet applet at AWL Web site)
Queueing delay (revisited)

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

Utilization \[ U = \frac{(av)aarrival rate}{(av)service rate} = \frac{a}{R/L} \]

- $La/R = U \sim 0$: avg. queueing delay small
- $La/R = U \rightarrow 1$: avg. queueing delay large
- $La/R = U > 1$: more “work” arriving than can be serviced, average delay infinite!
Packet loss

- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all
“Real” Internet delays and routes

- What do “real” Internet delay & loss look like?
- **Traceroute program**: provides delay measurement from source to router along end-end Internet path towards destination. For all $i$:
  - sends three packets that will reach router $i$ on path towards destination
  - router $i$ will return packets to sender
  - sender times interval between transmission and reply.
“Real” Internet delays and routes

traceroute: gaia.cs.umass.edu to www.eurecom.fr

Three delay measurements from gaia.cs.umass.edu to cs-gw.cs.umass.edu

trans-oceanic link

* means no response (probe lost, router not replying)
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

![Diagram](Image)

- Server sends bits (fluid) into pipe
- Pipe that can carry fluid at rate $R_s$ bits/sec
- Pipe that can carry fluid at rate $R_c$ bits/sec
Throughput (more)

- $R_s < R_c$  
  What is average end-end throughput?

- $R_s > R_c$  
  What is average end-end throughput?

**bottleneck link**

link on end-end path that constrains end-end throughput
Throughput: Internet scenario

- per-connection end-end throughput: \( \min(R_c, R_s, R/10) \)
- in practice: \( R_c \) or \( R_s \) is often bottleneck

10 connections (fairly) share backbone bottleneck link \( R \) bits/sec
Internet Structure: network of networks

❖ Network Edge users/clients connect to Wide-Area Internet via Local Access Networks

Hierarchy of inter-connects

Campus backbone network: high speed links.

Gateway to Wide Area Network

Campus Access Network

Backbone

Dept. LAN
Wide Area Internet Architecture

Autonomous domain: under a single administration

Various local networks - small office & home (SOHO) users, institutions etc. connect to the Wide-Area Internet through ISP
Internet structure: network of networks

- **at center:** small # of well-connected large networks
  - “tier-1” commercial ISPs (e.g., Verizon, Sprint, AT&T, Qwest, Level3), national & international coverage
  - large content distributors (Google, Akamai, Microsoft)
  - treat each other as equals (no charges)

Tier-1 ISPs & Content Distributors, interconnect (peer) privately … or at Internet Exchange Points IXPs
Tier-1 ISP: e.g., Sprint

POP: point-of-presence

to/from backbone

peering

to/from customers
Internet structure: network of networks

“tier-2” ISPs: smaller (often regional) ISPs

- connect to one or more tier-1 (provider) ISPs
  - each tier-1 has many tier-2 customer nets
  - tier 2 pays tier 1 provider

- tier-2 nets sometimes peer directly with each other (bypassing tier 1), or at IXP
Internet structure: network of networks

- a packet passes through many networks from source host to destination host
Chapter 1: roadmap

1.1 What is the Internet?
1.2 Network edge
   - end systems, access networks, links
1.3 Network core
   - circuit switching, packet switching, network structure
1.4 Delay, loss and throughput in packet-switched networks
1.5 Protocol layers, service models
1.6 Networks under attack: security (omit)
1.7 History
Protocol “Layers”

Networks are complex, with many “pieces”:

- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

**Question:**
Is there any hope of organizing structure of network?

Or at least our discussion of networks?
Organization of air travel

- ticket (purchase)
- baggage (check)
- gates (load)
- runway takeoff
- airplane routing

- ticket (complain)
- baggage (claim)
- gates (unload)
- runway landing
- airplane routing

❖ a series of steps
**Layering of airline functionality**

<table>
<thead>
<tr>
<th>Layers: each layer implements a service</th>
</tr>
</thead>
<tbody>
<tr>
<td>❖ via its own internal-layer actions</td>
</tr>
<tr>
<td>❖ relying on services provided by layer below</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer</th>
<th>Functions</th>
<th>Layer</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ticket (purchase)</td>
<td>ticket (complain)</td>
<td>ticket</td>
<td></td>
</tr>
<tr>
<td>baggage (check)</td>
<td>baggage (claim)</td>
<td>baggage</td>
<td></td>
</tr>
<tr>
<td>gates (load)</td>
<td>gates (unload)</td>
<td>gate</td>
<td></td>
</tr>
<tr>
<td>runway (takeoff)</td>
<td>runway (land)</td>
<td>takeoff/landing</td>
<td></td>
</tr>
<tr>
<td>airplane routing</td>
<td>airplane routing</td>
<td>airplane routing</td>
<td></td>
</tr>
</tbody>
</table>

**Layers:**
- **departure airport**
- **intermediate air-traffic control centers**
- **arrival airport**
Why layering?

Dealing with complex systems:

- explicit structure allows identification, relationship of complex system's pieces, simplifies design, implementation
  - layered reference model

- modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
    (e.g., change in gate procedure doesn't affect rest of system)

- Disadvantages?
Internet protocol stack

- **application**: supporting network applications
  - FTP, SMTP, HTTP
- **transport**: process-process data transfer
  - TCP, UDP
- **network**: routing of datagrams from source to destination
  - IP, routing protocols
- **link**: data transfer between neighboring network elements
  - Ethernet, 802.111 (WiFi), PPP
- **physical**: bits “on the wire”
ISO/OSI reference model

- **presentation**: allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- **session**: synchronization, checkpointing, recovery of data exchange
- Internet stack “missing” these layers!
  - these services, *if needed*, must be implemented in application
  - needed?

<table>
<thead>
<tr>
<th>application</th>
<th>presentation</th>
<th>session</th>
<th>transport</th>
<th>network</th>
<th>link</th>
<th>physical</th>
</tr>
</thead>
</table>

Introduction 1-68
Hourglass Protocol Architecture

- IP – thin waist of the hourglass Internet protocol architecture - ONLY 1 protocol at the network layer (IP)!

- **Multiple** higher-layer protocols
- **Multiple** lower-layer protocols
TCP/IP Protocol Suite

- HTTP
- SMTP
- DNS
- RTP

**Diverse communications**

**Reliable stream service**
- TCP
- UDP

**Best-effort connectionless packet transfer**

**Network interface 1**
**Network interface 2**
**Network interface 3**

**Applications**
- Distributed applications

**Network layer**
- IP
- (ICMP, ARP)

**Transport layer**
- TCP
- UDP

**Application layer**
- HTTP
- SMTP
- DNS
- RTP

**Diverse communications**
Layering: logical communication

E.g.: transport

- take data from app
- add addressing, reliability check info to form “datagram”
- send datagram to peer
- wait for peer to ack receipt
- analogy: post office ??
Layering: physical communication
Layer n in one machine interacts with layer n in another machine to provide a service to layer n +1.

The entities comprising the corresponding layers on different machines are called peer processes.

The machines use a set of rules and conventions called the layer-n protocol.

Layer-n peer processes communicate by exchanging Protocol Data Units (PDUs).
- The data passed to the layer below is a **Service Data Unit (SDU)**
- SDU’s are **encapsulated** in PDU’s
- Service Access Points (SAP): interface between 2 adjacent layers
A layer may impose a limit on the size of a data block that it can transfer for implementation or other reasons.

Thus a layer-n SDU may be too large to be handled as a single unit by layer-(n-1).

Sender side: SDU is segmented into multiple PDUs.

Receiver side: SDU is reassembled from sequence of PDUs.
Multiplexing

- **Sharing** of layer n service by *multiple* layer n+1 users
- Multiplexing tag or ID required in each PDU to determine which users an SDU belongs to
Internetworking

- Key to building a “network of networks” (internet)
  - operating over *multiple, coexisting, different* network technologies
  - providing ubiquitous connectivity through IP packet transfer
  - achieving huge economies of scale

- To provide *universal communication services*
  - independent of underlying network technologies
  - providing common interface to user applications

- To provide *distributed applications*
  - Rapid deployment of new applications
Covered a “ton” of material!

- Internet overview
- what’s a protocol?
- network edge, core, access network
  - packet-switching versus circuit-switching
  - Internet structure
- performance: loss, delay, throughput
- layering, service models

You now have:

- context, overview, “feel” of networking
- more depth, detail to follow!