Part 1: Introduction

Goal:
- Review of how the Internet works
- Overview
  - Get context
  - Get overview, “feel” of the Internet
- Application layer protocols and addressing
- Network layer / Routing
- Link layer / Example
What’s the Internet: “Nuts and bolts” view

- Millions of connected computing devices: *hosts, end-systems*
  - PC’s workstations, servers
  - PDA’s, phones, toasters
  - running *network apps*

- *Communication links*
  - Fiber, copper, radio, satellite

- *Routers*: forward packets (chunks) of data through network
Example access net: Home network

Typical home network components:
- ADSL or cable modem
- router/firewall
- Ethernet
- Wireless access points
What’s the Internet: “Nuts and bolts” view

- **Protocols**: control sending, receiving of messages
  - E.g., TCP, IP, HTTP, FTP, PPP

- **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:
  - WWW, email, games, e-commerce, database, voting,
  - More?

- Communication services provided:
  - Connectionless
  - Connection-oriented

- Cyberspace [Gibson]:
  
  "a consensual hallucination experienced daily by billions of operators, in every nation, ...."
Principles of the Internet

- Edge vs. core (end-systems vs. routers)
  - Dumb network
  - Intelligence at the end-systems
- Different communication paradigms
  - Connection oriented vs. connection less
  - Packet vs. circuit switching
- Layered system
- Network of collaborating networks
A closer look at network structure

- **Network edge:** applications and hosts
- **Network core:**
  - Routers
  - Network of networks
- **Access networks, physical media:** Communication links
The network edge

- **End systems (hosts):**
  - Run application programs
  - E.g., WWW, email
  - At “edge of network”

- **Client/server model**
  - Client host requests, receives service from server
  - E.g., WWW client (browser)/server; email client/server

- **Peer-2-peer model:**
  - Host interaction symmetric
  - E.g.: File sharing
Network edge: Connection-oriented service

**Goal:** data transfer between end systems

- **Handshake:** setup (prepare for) data transfer ahead of time
  - Hello, hello back human protocol
  - *Set up “state”* in two communicating hosts

- TCP – Transmission Control Protocol
  - Internet’s connection-oriented service

**TCP service** [RFC 793]

- **Reliable, in-order** byte-stream data transfer
  - Loss: acknowledgements and retransmissions

- **Flow control:**
  - Sender won’t overwhelm receiver

- **Congestion control:**
  - Senders “slow down sending rate” when network congested
Network edge: Connectionless service

**Goal:** Data transfer between end systems
- Same as before!

- **UDP** – User Datagram Protocol [RFC 768]: Internet’s connectionless service
  - Unreliable data transfer
  - No flow control
  - No congestion control
The network core

- Mesh of interconnected routers
- The fundamental question: How is data transferred through net?
  - Circuit switching: Dedicated circuit per call: telephone net
  - Packet switching: Data sent through net in discrete “chunks”
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
Network core: Packet switching

Each end-end data stream divided into *packets*

- Users’ A, B packets *share* network resources
- Each packet uses full link bandwidth
- Resources used *as needed*
Network core: Packet switching

Packet-switching versus circuit switching: Human restaurant analogy
Network core: Packet switching

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
  - Transmit over link
  - Wait turn at next link
Delay in packet-switched networks

Packets experience delay on end-to-end path

- Four sources of delay at each hop
  - Nodal processing
    - Check bit errors
    - Determine output link
  - Queueing
    - Time waiting at output link for transmission
    - Depends on congestion level at router

A

transmission

B

propagation

nodal processing

queueing
Delay in packet-switched networks

Transmission delay:
- $R = \text{link bandwidth (bps)}$
- $L = \text{packet length (bits)}$
- Time to send bits into link $= \frac{L}{R}$

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

Note: $s$ and $R$ are very different quantities!
Queueing delay

- \( R \) = link bandwidth (bps)
- \( L \) = packet length (bits)
- \( A \) = average packet arrival rate

Traffic intensity = \( \frac{L}{R} \)

- \( \frac{L}{R} \sim 0 \): average queueing delay small
- \( \frac{L}{R} \rightarrow 1 \): delays become large
- \( \frac{L}{R} > 1 \): more “work” arriving than can be serviced => average delay infinite!
Packet switching vs. circuit switching

Is packet switching a “slam dunk winner?”

- Great for bursty data
  - Resource sharing
  - No call setup
- Excessive congestion: Packet delay and loss
  - Protocols needed for reliable data transfer, congestion control
- Key question: How to provide circuit-like behavior?
  - Bandwidth guarantees needed for audio/video apps
Packet-switched networks: Routing

- **Goal:** Move packets among routers from source to destination

- **Datagram network:**
  - *Destination address* determines next hop
  - Routes may change during session
  - Analogy: Drive towards dst. and ask for directions

- **Virtual circuit network:**
  - Fixed path determined at *call setup time*, remains fixed through call
  - Each packet carries tag (virtual circuit ID); Tag determines next hop
  - Routers maintain per-call state
  - Analogy: Fixed driving instructions
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*
Protocol “layers”

Networks are complex!
- 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 in our discussion of networks?
Why layering?

Dealing with complex systems:
- Explicit structure allows identification, relationship of complex system’s pieces
  - Layered reference model for discussion
- Modularization eases maintenance, updating of system
  - Change of implementation of layer’s service transparent to rest of system
  - E.g., change in gate procedure does not affect rest of system

- Layering considered harmful?
Protocol “Layers”

Networks are complex!

- 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?
Why layering?

Dealing with complex systems:
- Explicit structure allows identification, relationship of complex system’s pieces
  - Layered reference model for discussion
- 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
- Can layering be considered harmful?
Internet protocol stack

- **Application**: Supporting network applications
- **Transport**: Host-host data transfer
- **Network**: Uniform format of packets, routing of datagrams from source to destination
- **Link**: Data transfer between neighboring network elements
- **Physical**: Bits “on the wire”
Layering: *Logical* communication

Each layer:
- Distributed
- “Entities” implement layer functions at each node
- Entities perform actions, exchange messages with peers
Layering: *Logical* communication

E.g., transport

- Take data from application
- Add addressing, reliability check info to form “datagram”
- Send datagram to peer
- Wait for peer to ack receipt
- Analogy: post office

[Diagram of network layers with arrows showing data flow and acknowledgment]
Layering: *Physical* communication
Protocol layering and data

Each layer takes data from above
- Adds header information to create new data unit
- Passes new data unit to layer below
Internet protocol stack

- **Application**: Supporting network applications
- **Transport**: Host-host data transfer
- **Network**: Uniform format of packets, routing of datagrams from source to destination
- **Link**: Data transfer between neighboring network elements
- **Physical**: Bits “on the wire”
What makes the Internet so sexy

- Applications can be deployed by anybody that is connected to the Internet (Fundamentally different to the Telephone world)

- Multi-service network:
  Everything over the Internet
  - Every application protocol over IP
  - IP over any network technology
TCP/IP Protocol structure

Application
(Processes)

Transport
(Hosts)

Network

Link

- Telnet
- FTP
- HTTP
- SMTP
- DNS
- TCP
- UDP
- IP
- FDDI
- ATM
- Tokenring
- Ethernet
Internet structure: Network of networks

- Roughly hierarchical
- At center: “tier-1” ISPs (e.g., UUNet, BBN/Genuity, Sprint, AT&T), national/international coverage
  - Treat each other as equals

Diagram:
- Tier-1 providers interconnect (peer) privately
- Tier-1 providers also interconnect at public network access points (NAPs)
Internet structure: Network of networks

- “Tier-2” ISPs: smaller (often regional) ISPs
  - Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs

- Tier-2 ISP pays tier-1 ISP for connectivity to rest of Internet
- tier-2 ISP is customer of tier-1 provider

Tier-2 ISPs also peer privately with each other, interconnect at NAP
Internet structure: Network of networks

- “Tier-3” ISPs and local ISPs
  - Last hop ("access") network (closest to end systems)

Local and tier-3 ISPs are customers of higher tier ISPs connecting them to rest of Internet
Example Tier-1 ISP: Sprint
Internet structure: Network of networks

- A packet passes through many networks!

Try traceroute!
Internet philosophy

- Interconnect networks
  - Across multiple different technologies

- Requirements on these networks
  - Minimal assumption about network capabilities => some data should be transmitted from time to time
    ⇒ "Best Effort" paradigm

- High survivability
  - Mesh
    - Intermediate nodes do not maintain per host state
    => "End-to-end" paradigm

- Layering with simple application interface
  - Sockets
Internet design philosophy

In order of importance:

- Connect existing networks
- Survivability
- Support multiple types of service
- Must accommodate a variety of networks
- Allow distributed management
- Allow host attachment with a low level of effort
- Be cost effective
- Allow resource accountability
Internet terminology

- internet
  - Collection of packet switched networks interconnected by routers running the IP/TCP protocols
- The „Internet“
  - Public Internet (in contrast to Intranets)
- End system == host attached to network
- Router = gateway = intermediate system
  - Routes packets between its attached networks
- Firewall (device placed at network boundaries)
  - Restricts packet flows to improve security
Internet terminology (2.)

- **Name**
  - Identifies an object (e.g., an end system)

- **Address**
  - Identifies where an object is located (here IP address)

- **Name to address translation (and vice versa)**
  - Links names and addresses
    - mail.zrz.TU-Berlin.DE ↔ 130.149.4.15

- **Interface**
  - Network attachment (≥ 1 interfaces = multihomed)

- **Route**
  - How to get to an objects location

- **Network prefix**
  - Set of addresses