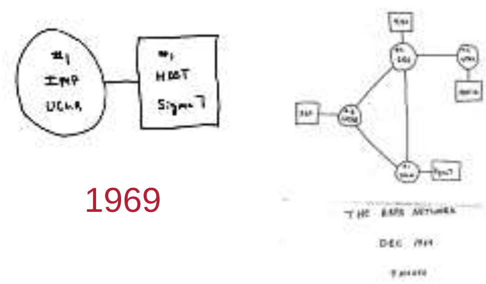
Internet Technology

01. Introduction

Paul Krzyzanowski
Rutgers University
Spring 2016

How did we get from this…



1969

Images courtesy of Computer History Museum.

… to this?

1973

Image courtesy of Computer History Museum.

… to this?

2015

Image courtesy of the opte project – opte.org

The early history of data communication

Computer networks existed before the Internet

- Remote sensors
  - 1951: Cape Cod System: network long-range & short-range radars to a Whirlwind I computer
- Remote terminals & printers (teletypes)
  - 1940: Dartmouth College, NH to New York
- Connected computers
  - 1960: Semi-automatic business research environment (SABRE)
    - Airline reservations
    - Teleprinters at American Airlines ticketing offices & two IBM 7090 mainframes

Data communication before computers

Data communication through history established the basic principles that we need for digital data networks

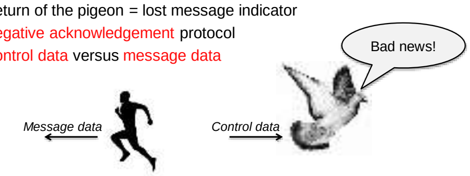
- – Broadcasting, unicasting
- – Control data vs. Message data
- – Synchronization
- – Relays
- – Acknowledgements
- – Error notification
- – Message encoding
- – Encryption
- – Flow control

Earliest forms of transmission

- Walk (or run) to deliver a message
- Point-to-point (unicast) delivery
- Professional runners are documented back in 2900 BC
- If you need it faster, use a horse

Negative acknowledgements

- What if the messenger is attacked?
  - – The message is lost
- 2350 BC: King Sargon of Akkad (Mesopotamia)
  - – Messengers carry a homing pigeon
  - – If the messenger is attacked, he would release the pigeon
  - – Return of the pigeon = lost message indicator
  - – Negative acknowledgement protocol
  - – Control data versus message data



Message relays

- c. 480 BC: Guard posts installed at regular intervals
- Created the opportunity for relaying messages (repeaters)
  - – New messengers or horses can replaced incoming tired ones

2,340 years later… the U.S. Pony Express service set up almost 200 horse exchange stations every 5-20 miles along the route from St. Joseph, Missouri to Sacramento, CA

Broadcasts

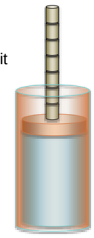
- Point-to-point transit was not always possible
  - – Speed
  - – Danger
  - – Impassible terrain or water (e.g., coordinate ships)

Broadcasts

- Fire-based signaling
  - – 1184 BC: signal the fall of Troy ("beacon to beacon")
  - – Size of flame may convey a message
    - 1455: Scottish signal approach of the English
    - Gauging size can be tricky
    - … so the basic use was simply to signal an event
  - – Or number of flames
    - "One if by land…"
  - – Only a tiny number of predefined messages can be encoded
  - – Limited to line of sight and ability to see
  - – Longer distances need relays

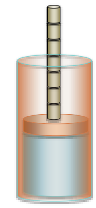
Encoding: Water-clock optical communication

- c. 350 BC Aeneas Tacticus (Αινείας ο Τακτικός)
  – Greek military scientist & cryptographer
- Both sides have a jar with the same amount of water in it
- A graduated rod floating in cork encodes 30 messages
  – 1 = “Cavalry arrived in the country”
  – 2 = “Heavy infantry”
  – 3 = “Light armed infantry”
  – …
- Protocol for communication
  – Sender raises a torch (synchronization)
  – Receiver raises a torch (synchronization)
  – Sender & receiver unplug a cork at the bottom of the jar
  – Sender raises a torch again when the rod reaches the desired message
  – Receiver plugs the jug & raises a torch (positive acknowledgement)
  – Receiver reads the message # on the rod and consults a table (encoding)



Water-clock optical communication

- Control signals (torches) are separated from data signals (positions on the rod)
- Message encoding = messages as positions on the rod
- Full alphabet could be encoded but transmission of meaningful messages would be very slow



Message Encoding: Polybius

- 5x5 grid of letters
  – 5 tablets with 5 letters per tablet
- One set of 5 torches identifies tablet
- Another identifies the letter
- Torches remain lit.
  Unneeded torches hidden behind a screen
- Ability to send arbitrary messages (but tedious)
- First optical telegraph

Longer transmission distance: telescopes

- Telescope invented in early 1600s
- Led to increased interest in long-distance communication
- 1616: Franz Kessler
  – Use a telescope to view a flashing light
  – A shutter in front of a burning torch flashes a number, which corresponds to a letter

Robert Hooke’s telegraphic communication

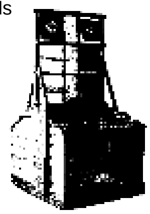
- Long-distance visual communication system: 1684
- Wooden frame holds symbols (flow control) and letters of the alphabet (data)
- Flow control symbols included
  – "I am ready to communicate” *synchronization*
  – "I am ready to observe” *synchronization*
  – "I shall be ready presently” *delay*
  – "I see plainly what you shew” *acknowledgement*
  – "Shew the last again” *error report*
  – "Not too fast” *rate control*
  – "Shew faster” *rate control*
  – "Answer me presently. Dixi." (Dixi = Latin for "I have spoken") *end-of-text*
  – "Make haste to communicate this to the next correspondent” *priority*

French Optical Telegraph: version 1 (1791)

- 1791: Claude Chappe
- Link between Brûlon and Parcé (17 km)
  – Two synchronized pendulum clocks, numbers 1-10 on the face plate
  – Two large flipboards: one side white, the other black
  – Two telescopes
  – Flipboards used for synchronization
    1. Flip the board to synchronize the clocks
    2. Sender waits until the clock points to the desired number
    3. Sender then flips the board. The receiver looks at the number on his clock

French Optical Telegraph: version 2 (1792)

- Use a wood frame holding 5 sliding panels
- Panels produce a 5-bit code
  - $2^5 = 32$ possible combinations
- Much more efficient data transmission
  - Adopted in the U.K.



French Optical Telegraph: version 3 (1793)

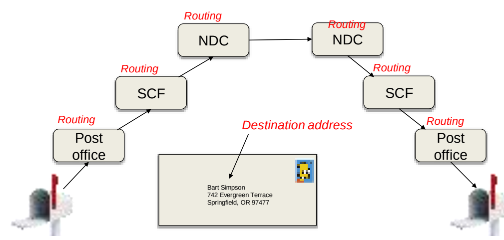
- Semaphore system
  - Large horizontal beam (regulator)
    - Rotates to 4 positions
  - Smaller wing (indicator) at each end
    - 8 positions
- Total # of symbols =
  - 8 x 8 x 4 = 256
  - 8x more efficient!
- 15 stations by 1794; 556 stations by 1852
- By 1840s, nearly every European country had optical telegraph lines
- 1-3 symbols per minute ≈ 8.5 bits per second!

Electromagnetic telegraphy

- England and U.S. circa 1837
- Not immediately attractive to countries with optical telegraph networks
  - Wires could be severed and destroy a communication link
  - Requires electricity (batteries)
  - Requires capital expenditure and land rights to lay wires
- Semaphores were particularly attractive for ship-ship and ship-to-shore communication (pre-radio)
- Increased transmission speeds
  - Up to 35 – ~60 words per minute ≈ 30-50 bits per second

Postal Service: Store & Forward Transmission

- Sectional Center Facility (SCF): serves a geographic area defined by one or more 3-digit ZIP code prefixes
- Network Distribution Center (NDC): 21 NDCs serve as the "backbone" of the network



And more…

- 1878: Bell Telephone Company opens 1st switching office
  - Plugboard switches – manual circuit switching
- 1890: Connections between switching offices
- 1896: Wireless telegraph
- 1926: Telephone crossbar switches – "circuit switching"
- 1939: Pulse-code modulation (PCM) – digital voice
- 1946: First car phone
- 1946: Frequency division multiplexing to support 1,800 telephone circuits on three pairs of coax cables
- 1956: Transatlantic cable
- 1958: 300 bps modem (modulator/demodulator)
- 1960s: 2400 bps modem
- 1963: ASCII (American Code for Information Exchange)

Data Networking

Early vision for the Internet

- 1961: First paper on packet switching
  - Leonard Kleinrock, MIT
- 1962: "Intergalactic Network" by J.C.R. Licklider of MIT
  - Early vision of the Internet
  - Globally interconnected computers
  - Access data & programs from any computer
- 1965: First wide-area network
  - A computer in Massachusetts is connected to a computer in California over a low-speed dial-up line

This led to…

1968: ARPANET

- Funded by DARPA
  (Defense Advanced Research Projects Agency)
- Experiment in resource sharing & packet switching
- Provide high bandwidth links between academic, industrial, and government research labs working on defense projects
- Key component: Interface Message Processor (IMP)
  - Precursor of router
  - RFP won by Bolt Beranek and Newman (BBN)

Photo from https://en.wikipedia.org/wiki/Interface_Message_Processor

1968: ARPANET

- 1969: ARPANET goes live
  - UCLA: 1st node on ARPANET
  - Stanford Research Institute (SRI): 2nd node (Doug Engelbart's lab)
    - SRI kept track of host name to address mapping and directory of RFCs
- Four hosts on ARPANET by the end of 1969
- 1970: Network Control Program (NCP) created
  - Allows application-to-application communication
  - Precursor of TCP/IP
- Now we can write applications that use networking!
- 1972: first killer app – email

Key design principles

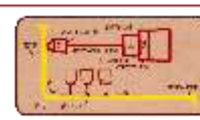
Key Principles of the ARPANET evolved into the Internet

1. Support interconnection of networks
   - No changes needed to the underlying physical network
2. Assume unreliable communication; design for best effort
   - If a packet does not get to the destination, it will be retransmitted
3. Routers & gateways will connect networks
   - They will not store information about the flow of packets
4. No global (centralized) control of the network

ARPANET host-to-host protocol

- NCP (Network Control Program)
  - Assumed reliable networks
- Evolved to TCP (Transmission Control Protocol)
  - Handled retransmission & in-order delivery
- Evolved to
  - IP for host-to-host communication
  - TCP for reliable application-to-application communication
  - UDP for unreliable application-to-application communication
- Early applications:
  - Email, file transfer (ftp), chat (rtalk), remote login (rlogin), packet voice, remote file access

The Internet (ARPANET) evolves



- 1973: Ethernet developed
  - Commercially introduced in 1980
  - 11 years after ARPANET!
- 1983: 4.2 BSD UNIX
  - UCB integrated networking (**sockets**) into UNIX
  - Became the dominant network API for all operating systems
- 1985: Domain name system
  - 1984: top-level domains defined (.gov, .edu, .com, .mil, .org)
- 1988: Scalable routing
  - Interior Gateway Protocol (IGP) within a region of the Internet
  - Exterior Gateway Protocol (EGP) to connect regions together
    → Border Gateway Protocol (BGP)
- Open standards, documented as RFCs by the Internet Engineering Task Force (IETF)

The Internet goes commercial

- ARPANET demonstrated the value
  … but was available only for DoD partners
  - Other networks were created
  - NSFNET, CSNET, BITNET
- NSFNET (1985) chose to use TCP/IP
  - Available for non-defense computer science research
  - NSFNET did not permit commercial traffic on its backbone
  - Private networks emerged to handle commercial traffic
- NFSNET goes commercial
  - Late 1980s goal: make infrastructure not reliant on public funding
  - Seek commercial, non-academic customers
  - 1990: ARPANET decomissioned
  - 1995: NSFNET defunded

Some Killer Apps on the Internet

- 1982: email
- 1989: World Wide Web invented by Tim Berners-Lee
  - HTTP, HTML, Web server, Web browser
  - Mosaic browser (1993) → Netscape → Mozilla
- 1995: amazon.com
- 1995: eBay
- 1998: Google
- 1999: Napster
- 2003: Skype
- 2005: Facebook, YouTube
- 2006: Twitter
- 2012: Amazon web services

A look at the Internet

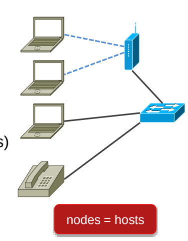
What is the Internet?

A global network of networks based on the IP family of protocols

The edge of the network

Local Area Network (LAN)

- Communications network
- Small area
  - Home, building, set of buildings
- Same, sometimes shared, transmission medium
- Usually high data rate (10 Mbps-10 Gbps)
- Low latency
- Devices are peers
  - Any device can initiate a data transfer with any other device



node = host = device on the network running an application: clients & servers

Connecting Nodes to LANs

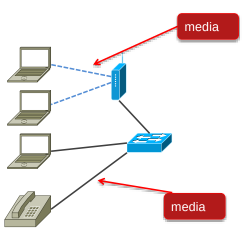
Connecting Nodes to LANs

Adapter
- expansion slot (PCIe, Thunderbolt dongle, USB dongle)
- usually integrated onto main board

E.g., Intel 82579V Gigabit Ethernet Controller

Network adapters are referred to as
Network Interface Controller (NICs) or adapters

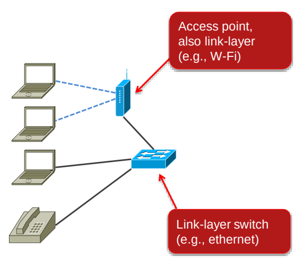
Local Area Network



Common media
- Unshielded Twisted Pair (UTP)
  - Example: 8-wire Category 6 ethernet cable
- Radio frequency
  - 802.11n frequency hopping in the 2.4 GHz and 5 GHz bands
- Coaxial cable
  - Cable TV, digital interconnect from carriers
- Optical Fiber

Media: Data communication links: fiber, radio frequency, copper wire

Local Area Network



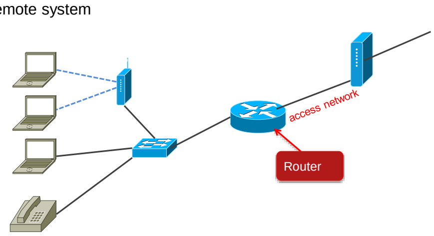
Hub:
- Device that acts as a central point for LAN cables
- Take incoming data from one port & send to all other ports

Switch
- Moves data from input to output port
- Analyzes packet to determine destination port and makes a virtual connection between the ports.

Link-layer switches: create a physical network (e.g., Ethernet, Wi-Fi)

Beyond the LAN: Access Networks

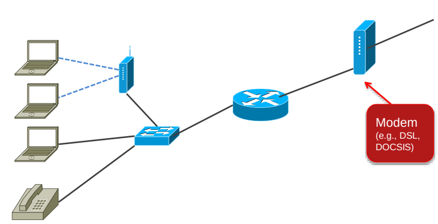
- Connect a LAN to a router: the first step in the path to the destination on a remote system



Access Network = link between a customer's network and the ISP

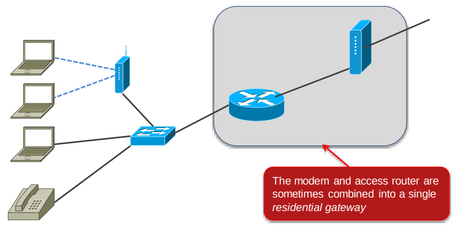
Router: control the movement of packets between physical networks

Beyond the LAN: Access Networks



Modem: modulator-demodulator – converts data between different analog formats (e.g., phone lines, cable TV, fiber)

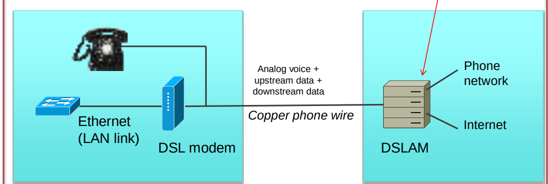
Beyond the LAN: Access Networks



Residential gateway: connects a home or office network to an Internet service provider

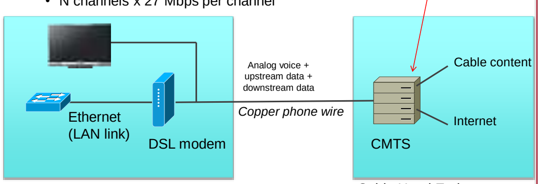
Access Link: DSL

- DSL: Digital Subscriber Line
  - Use phone wiring
  - Voice: 0 – 4 kHz
  - Upstream channel: 4 kHz – 50 kHz
  - Downstream channel: 50 kHz – 1 MHz



Access Link: Cable Internet

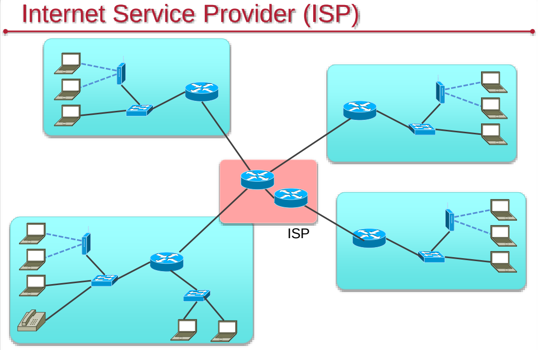
- DOCSIS Cable Modem
  - Coax cable
  - Downstream (shared channel)
    - M channels x 38 Mbps per channel = channel bonding
  - Upstream (timeslots – time division multiplexing, TDM)
    - N channels x 27 Mbps per channel



Other access links

- Fiber to the home
  - Example: Verizon FiOS
- Cellular: LTE (Long-Term Evolution)

Internet Service Provider (ISP)



ISP: Organization that provides access to the Internet

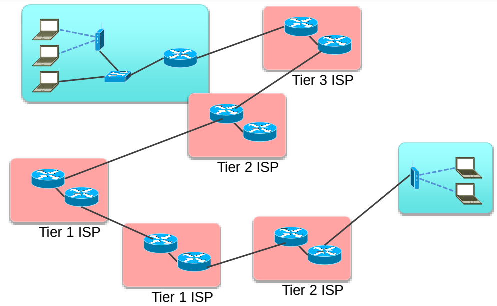
ISP Tiers

- One ISP does not connect to all the LANs of the world!
- Thousands of ISPs
  - They purchase internet service from ISPs above them
- At the top: the Internet Backbone
  - Main data routes between core routers on the Internet
  - Lots of redundancy

ISP Tiers

- Tier 1 = top-level ISPs
  - Peering agreements with each other
    - Peering = forward/receive traffic at no cost
  - Have access to the entire Internet routing table
  - Global Tier 1 ISPs
    - Own the infrastructure that forms the backbone
    - may peer on multiple countries/continents
    - May own/lease transoceanic fiber
  - Examples: AT&T, Level 3, Cogent Communications, CenturyLink, NTT
- Tier 2 ISPs
  - Peers with some networks but purchases transit from Tier 1 networks and other Tier 2 networks
- Tier 3 ISPs
  - Focus on retail and consumer markets
  - Purchases transit from Tier 1 & Tier 2 ISPs
  - Direct coverage limited to a region of a country

Routing



A packet may pass through many networks – within and between ISPs

Switching

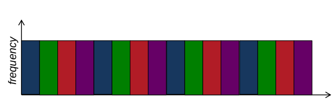
How do nodes share a network?

- Dedicated connection – no sharing: physical circuit
- Talk on different frequencies: broadband
  - Range of frequencies: FDM (Frequency Division Multiplexing)
- Take turns: baseband
  - Short fixed time slots: circuit switching
    - TDM (Time Division Multiplexing)
    - Circuit switching: performance equivalent to an isolated connection
  - Variable size time slots: packet switching
    - *Statistical multiplexing* for network access
    - Easily support many-to-many communication

- Packet switching is the dominant means of data communication

Circuit Switching

FDM: Frequency Division Multiplexing



TDM: Time Division Multiplexing

Circuit Switching

- Connection setup (control message)
  - Establishes a path (route) from the source to the destination
  - The node is informed that the path is set up and data transmission can take place
- Data transmission
  - Same route taken for all the data
  - The path and switching resources remain allocated whether data is flowing or not
- Teardown
  - When the transmission is done, the sender releases the circuit
- Circuit switching offers
  - Guaranteed (fixed) bandwidth
  - Constant latency

Packet Switching

- Packet switching
  - Messages are broken into chunks of data called packets
  - Packets travel from the source node to a destination via packet switches: switches and routers
  - Each packet contains a destination address
  - Available bandwidth ≤ channel capacity
  - Variable latency

Why is circuit switching awesome?

- Routing decisions do not have to be made for each packet – just once at circuit setup
  - … but circuit setup takes time
- Data can be routed out as its being received by a router
  - Data for a specific flow arrives on pre-determined channel
  - No need to buffer it (no congestion)
- Guaranteed bandwidth

Why is circuit switching NOT awesome?

- Network capacity is dedicated to the connection whether it is used or not
- Example (from text)
  - Users share a 1 Mbps link
  - Suppose a user is active 10% of the time
  - Circuit switching
    - If a 1-second frame is divided into 10 time slots of 100 ms, each user would be allocated one time slot per frame
    - Maximum # of simultaneous users = 10
  - Packet switching
    - Probability that a user is active is still 0.10
    - Assume 35 users (> 3x than in circuit switching)
    - What is the probability that ≤10 users are active simultaneously?
      - 10 users will fill channel capacity

Packet Switching Example: working it out …

- 35 users; each user is active 10% of the time (p=0.1)
- Probability that exactly *n* users are active?
  - Use the *probability mass function* of the *binomial distribution*
    - We want exactly *n* successes (active) = p^n
    - And 35 - *n* failures (inactive) = $(1 - p)^{35 - n}$
    - There are $\binom{35}{n}$ different ways of distributing 35 successes ← Binomial coefficient

$$= \frac{35!}{n!\,(35 - n)!}$$

  - P(exactly *n* users active) = $\binom{35}{n} p^n (1 - p)^{35 - n}$
- Probability that ≤ *n* users are active simultaneously
  = P(0) + P(1) … + P(*n*) = 0.2503 + 0.0973 + ... ← Cumulative distribution function
  = 0.99958

Why is circuit switching NOT awesome?

- Packet switching
  - Probability that a user is active is still 0.10
  - Assume 35 users (> 3x than in circuit switching)
  - P(exactly n users active simultaneously) = ${}^{35}C_n * (0.1)^n * (0.9)^{(35-n)}$
  - P(≤10 users active simultaneously) = P(0) + P(1) + … + P(10) = 0.99958 = 99.96%
- *Essentially the same performance as circuit switching for >3x the users!*
  - Also, users can see better performance than with circuit switching if the packet network is lightly loaded for periods of time

Delay analysis

Units in Networking

Time (delay)

Unit	Abbr	value	seconds
second	s	1	1 s
millisecond	ms	1,000 ms = 1 s	1×10^{-3} s
microsecond	μs	1,000 μs = 1 ms	1×10^{-6} s
nanosecond	ns	1,000 ns = 1 μs	1×10^{-9} s

Data Size (payload)

Unit	Abbr	value
bit	b	
byte	B	1 byte = 8 bits
kilobits	Kb	1 Kb = 1,000 bits (10^3 bits) = 125 bytes
megabit	Mb	1 Mb = 10^6 bits
gigabit	Gb	1 Gb = 10^9 bits

Channel Capacity (bitrate)

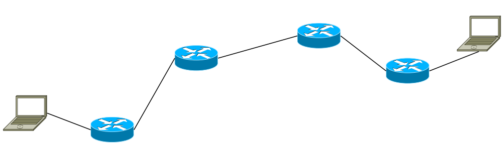
Unit	Abbr	value
bits per second	bps	
kilobits per second	Kbps	1,000 bits per second = 10^3 bps
megabits per second	Mbps	10^3 Kbps = 10^6 bps
gigabits per second	Gbps	10^3 Mbps = 10^6 Kbps = 10^9 bps

Delay & throughput in packet-switched networks

Network Delay

A packet passes through routers on its way to its destination

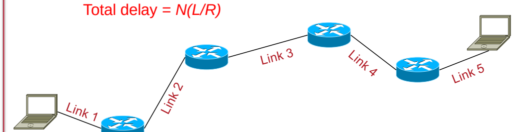
Each router & network link introduces a delay



Store & Forward Overhead

With packet switching

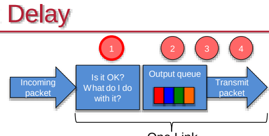
- A packet must be fully received before it is sent out
- In the simplest case, if
  R = rate in bits per second and *L = length of packet in bits*
- It takes L/R seconds to transmit a packet to the first link.
- Another L/R seconds to transmit to the next link
- *N* links ⇒ *N-1* routers:
  Total delay = *N(L/R)*



Sources of Network Delay

Per-link:

1. Processing delay
2. Queuing delay
3. Transmission delay
4. Propagation delay



1. Processing delay

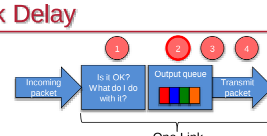
Time to examine the packet's header, check for errors, determine where to route it (output link)

Typical delay: several microseconds

Sources of Network Delay

Per-link:

1. Processing delay
2. Queuing delay
3. Transmission delay
4. Propagation delay



2. Queuing delay

On packet-based networks,
only one packet may be transmitted onto a link at a time

Packets awaiting transmission will wait in a queue

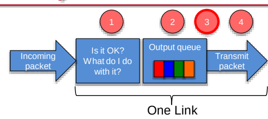
Queuing delay = function of:
 # packets waiting to be transmitted
 size of those packets
 speed at which bits can be transmitted

Typical delay: 0 to several milliseconds

Sources of Network Delay

Per-link:

1. Processing delay
2. Queuing delay
3. Transmission delay
4. Propagation delay



3. Transmission delay

Time to get the entire packet onto the link

Transmission delay = packet size ÷ link speed
If *R = rate in bits per second* and *L = length of packet in bit*
Transmission delay = *L/R*

Example:

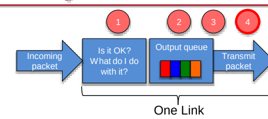
Time to transmit a 1500 byte packet (maximum size of regular ethernet frame) on a 1 Gbps link takes (1500×8) ÷ 10^9 = 0.000012 s = 12 μs

On a 10 Mbps link, the same packet would take 1.2 ms to transmit

Sources of Network Delay

Per-link:

1. Processing delay
2. Queuing delay
3. Transmission delay
4. Propagation delay



4. Propagation delay

Once the data is on the network, time to get to the next router or end node

Propagation delay = distance × signal propagation speed in medium

Wireless = speed of light (c) = 3.00 × 10^8 m/s
Unshielded twisted pair (UTP) = 0.59c = 1.77 × 10^8 m/s
Single mode (long distance) optical fiber = 0.68c = 2.04 × 10^8 m/s

Example:

Optical fiber: NYC to London delay = (5,576 × 10^3 m) × (2.04 × 10^8 m/s) = 27.3 ms

Queuing Delay: Traffic Intensity

- Queuing delay can range from insignificant to huge
- What determines the delay?
  - Rate at which traffic arrives
  - How bursty the traffic is (variation in arrival time)
  - Transmission rate of the outbound link (how quickly the packets can get out)

Let

a = average rate of packet arrival (packets per second)
L = size of packet (bits; assume all packets are equal)
R = transmission rate (bits per second)

Then

L/R = # seconds that it takes to transmit a packet
La/R = **traffic intensity**: useful in estimating queuing delays

Queuing Delay: Traffic Intensity

- If *La/R* > 1
  - Packets arrive faster than they can be transmitted
  - Queue will grow
  - Eventually, packets will have to be dropped ⇒ packet loss

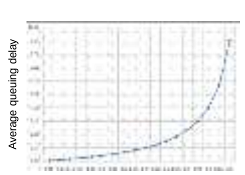
> *a* = average rate of packet arrival (packets per second)
> *L* = size of packet (bits; assume all packets are equal)
> *R* = transmission rate (bits per second)
>
> ***La/R* = traffic intensity**

- If *La/R* ≤ 1
  - Packets can be transmitted at the same speed or faster than they arrive
  - BUT … packets may arrive in bursts
  - If *N* packets arrive at once
    - First packet: no queuing delay (takes *L/R* time to transmit)
    - Second packet: queuing delay = *L/R* (time to transmit the 1st packet)
    - Third packet: queuing delay = *2L/R* (time to transmit the 2 previous packets)
    - Nth packet: queuing delay = *(N-1)L/R*

Queuing Delay: Traffic Intensity

In reality, packets arrive randomly

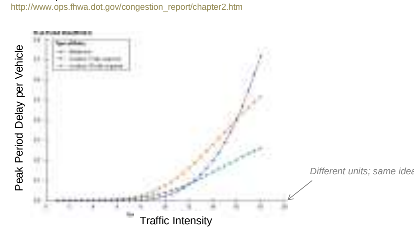
- As traffic intensity approaches 1
  - There will be times when packets come in faster than they can be transmitted
  - The average queue length (and hence delay) increases rapidly



Queuing Delay: Traffic Intensity

It's the same for cars as it is for packets

Relationship of Incident and Bottleneck Delay to Traffic Intensity
Traffic Congestion and Reliability
U.S. Department of Transportation
Federal Highway Administration
Office of Operations
http://www.ops.fhwa.dot.gov/congestion_report/chapter2.htm



Nodal delay

Total delay per node (router) =

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

d_{proc} = processing delay (typically a few microseconds)

d_{queue} = queuing delay (depends on congestion)

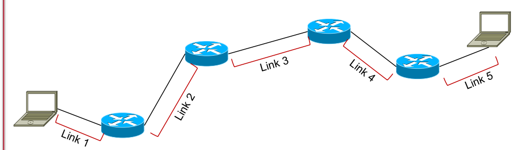
d_{trans} = transmission delay (L/R)

d_{prop} = a few microseconds to a few milliseconds

End-to-end delay

- We looked at a single link
- With *N-1* routers, we have *N* links
- Total end-to-end delay (*ignoring queuing delays*) =

$$d_{end\text{-}to\text{-}end} = N(d_{nodal}) = N(d_{proc} + d_{trans} + d_{prop})$$



The end