Introduction to Computer Networks and the Internet COSC 264

Introduction to Networking and Communications

Dr. Andreas Willig

Dept. of Computer Science and Software Engineering
University of Canterbury, Christchurch

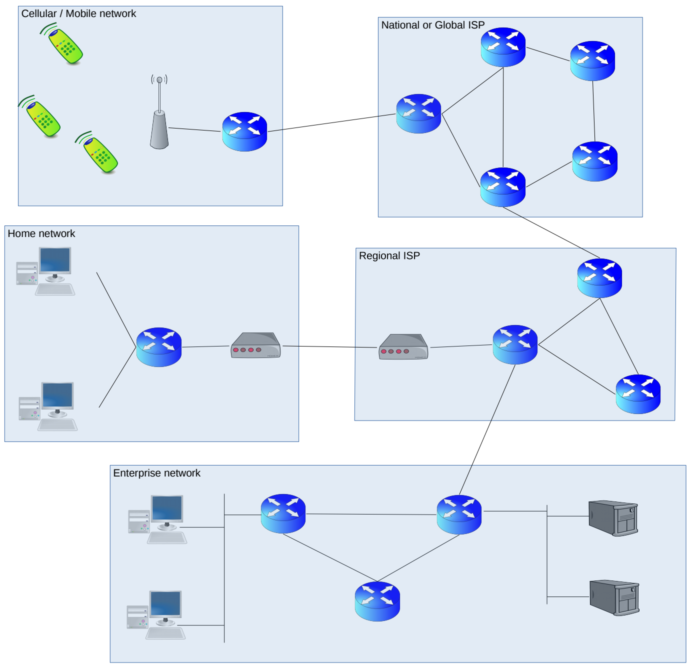
UoC, 2020

Scope

- High-level introduction to networking and networked applications
- Focus on Internet, more on principles rather than technologies
- Broad-based survey, many details clarified later
- References:
  - We do not use a single textbook, much of the material is built on [3]
  - Standard networking textbooks include: [6], [3], [4], [5]

Outline

A First Look at the Internet



A First Look (2)

- The Internet connects billions of **end systems** or **hosts**, like:
  - Desktop computers
  - Web servers, file servers
  - Smartphones, tablets
  - Networked devices like sensors, fridges, cars, ...
- End hosts are attached to **networks**, either:
  - to local area networks like Wireless LANs or Ethernet
  - to provider networks through point-to-point links or dialup lines
  - to mobile broadband networks through a wireless interface
- These networks are connected together using special devices called **routers** or **switches**, which themselves are connected through networks or point-to-point links
- The Internet is a **network of networks**

A First Look (3)

- End hosts and routers / switches are connected to each other through point-to-point links, cellular or local area networks, which can use different technologies, for example:
  - Wireless links: WiFi / WLAN, mobile networks, satellites
  - Wired links: Ethernet with coaxial or STP cable, optical links
- The link technologies differ in their characteristics, for example:
  - they have different transmission rates (in bits/s)
  - they have a different length / range
  - they have different error rates / reliability
- Switches and routers can be operated by individuals and companies, as well as *Internet Service Providers* (ISP)
- ISPs together provide global connectivity

A First Look – Packet Switching

- Information is transmitted in units of **packets**, e.g. a file or web page is split into several such packets
- A packet carries data and extra information, e.g. addressing fields specifying the sender and receiver of the packet
- Routers and switches receive packets on an incoming link and use their addressing fields to find the right outgoing link and thus a path to the destination (**routing**)
- Different packets between the same source and destination can take different routes
- This principle of splitting information into individual packets and routing these individually is called **packet switching**
  - Internet is similar to road or train network or postal network
- The POTS (the "Plain Old Telephone System") operates on a different principle called circuit-switching

A First Look – Network Structure

- End systems / hosts form the **edge** of the network
- **Access networks**
  - Denotes local network or link connecting end hosts to next router
  - Refers sometimes also to the boundary between networks operated by a customer (which then is the access network) and an ISP
  - Examples:
    - ADSL/VDSL or telephone dialup (yuck!) for residential access
    - Local area networks like Wireless LAN or Ethernet on a campus
    - 3G / 4G / 5G broadband networks
  - Offer data rates in the range of a few Mbps to a few Gbps
- **Provider / core networks**:
  - Internet service providers come in different sizes
  - The smaller ones offer Internet access to end customers
  - The larger ones offer global connectivity to other (smaller) ISPs or very large and global customers
  - Offer data rates in the range of several Gbps to Tbps to Pbps

A First Look – Transmission Media

- Data transmission is ultimately based on the propagation of **signals** or **waves** over a **transmission medium**
- Main cases of interest:
  - **Acoustic signals** propagate through air or water, represented as variation of pressure over time
  - **Electromagnetic signals** (e.g. radio frequency, infrared, optical) propagate through space (free space or cables), represented as variation of amplitude, phase or polarity over time
- Acoustic signals propagate at speed of sound, electromagnetic signals propagate at speed of light, both are medium-dependent
- A sender generates a carrier signal and **modulates** it according to the information that needs to be transmitted
- The transmission medium distorts and attenuates the signal, the receiver needs to guess the information from the incoming (distorted, attenuated) signal (**demodulation**)

A First Look – Guided Transmission Media

- In guided media a data signal can only propagate in a confined space, e.g. along a cable
- Twisted-pair copper wire
  - Used as telephone cabling and in some Ethernet variants
  - Relatively susceptible to electromagnetic interference (e.g. from electrical / magnetical devices)
  - Only connects two stations
  - Data rates and ranges: up to a few tens of Gbps (Ethernet) over short ranges (tens of meters)
- Coaxial cables
  - Used in cable television and cable-based Internet access
  - Better shielding
  - More difficult to handle (hard to bend around corner)
  - Several stations can be attached
  - Data rates / distances: tens of Mbps, few hundreds of meters

A First Look – Guided Transmission Media (2)

- Fiber cables
  - Used in Internet core and high-speed backbones
  - Difficult to handle
  - Only connects two stations
  - Immune against electromagnetic interference, very low error rates
  - Data rates / distances: Tens of Gbps to Tbps, tens to few hundreds of kilometers

A First Look – Unguided Transmission Media

- In unguided media a data signal can propagate in all directions
- Acoustic transmission through air or water:
  - Data rates / distances: tens to hundreds Kbps, hundreds of m / a few km
- Terrestrial radio channel
  - Uses radio frequencies (RF bands extend from 20 kHz to 300 GHz, practically used bands are below 100 GHz)
  - Mostly used for access networks
  - Supports mobility of end hosts and people
  - Can propagate through walls, can save expensive (!) cabling
  - Examples: Wireless LAN, mobile broadband
  - Data rates / distances: up to few Gbps over tens of meters for WLAN, hundreds of Mbps over hundreds of meters to few kilometers for mobile broadband

A First Look – Unguided Transmission Media (2)

- Satellite radio channel
  - Used to link two far away ground stations without cabling, or to provide broadcast services (like TV) or telephone services to large or sparsely populated areas, high-speed Internet access
  - Geostationary satellites (at 36,000 km distance) appear to be stationary, low-earth orbit (LEO) satellites provide only intermittent connectivity and many of them are required
  - Data rates: up to hundreds of Mbps, a few Gbps
  - Often very expensive for anything other than broadcast
- Infrared and optical wireless options exist but are not common

Outline

Networked Applications

- The Internet enables many distributed applications, e.g.:
  - Sharing of resources (printers, files) within a department
  - Email
  - Remote access to computers
  - World-wide web
  - Voice-/video conferencing
  - Video streaming, TV on demand
  - File transfers, peer-to-peer file sharing (legal or otherwise)
  - Social networks
  - Rapid spreading of viruses, trojans and other threats
  - Internet of Things / networked sensors
  - Online multi-player games
- Applications are usually programs running on end systems, routers or switches are rarely involved
- We are particularly interested in the **Quality-Of-Service** (QoS) and **Quality-Of-Experience** (QoE) requirements of applications

Outline

QoS and QoE

- QoS metrics aim to characterize the performance of a network independently of applications
- Typical QoS metrics in the Internet include
  - End-to-end delay: time for a packet to reach receiver?
  - Packet loss rate: what fraction of packets is being lost in transit?
  - Achievable data rate / throughput
- Typical metrics for (fixed or mobile) telephone networks include:
  - Call setup time
  - Call blocking probability
- The term QoE is used in different ways:
  - It refers holistically to the overall experience when using a service
  - For applications like audio or video conferencing it relates to perception-oriented performance, e.g. speech quality, impact and visibility of distortions, smooth movements in a video etc

Delay

- Delay measures:
  - the time between generation of a packet at a source host and its reception at the destination (**end-to-end delay**), or
  - the time between generation of a packet at a source host and reception of some response at the source host (**round-trip delay**)
- Some applications require bounds on delay (**delay-sensitive**)
  - Can you name some?
- Components of end-to-end delay:
  - Propagation delay
  - Transmission delay
  - Processing delay in hosts and routers
  - Number of routers along the path
  - Queueing delays in hosts and routers

Delay – Propagation Delay

- Information is imparted on carrier signals
- Carrier signals travel at medium-dependent propagation speed v:
  - Speed of sound in the air: $v \approx 340$ m/s
  - Speed of sound in water: $v \approx 1,500$ m/s
  - Speed of light in free space: $v \approx 300,000$ km/s
  - Speed of light in a cable: $v \approx 200,000$ km/s
- A signal propagating at speed v (in m/s) takes time

$$\frac{d}{v}$$

  to travel a distance of d m
- This travel time is called **propagation delay**
- **Question**: what is the propagation delay of an acoustic signal through air at a distance of 1 km?

Delay – Transmission Delay

- Digital information is modulated at a certain rate onto a carrier signal, governing the speed of information transmission
- This rate is called the **data rate** and specified in bps (bits per second) or Bps (bytes per second)
- When the data rate is R bps and we want to transmit a block of l bits of information, the time it takes to transmit these l bits is:

$$\frac{l}{R}$$

  and is called the **transmission delay**
- **Question**: what determines the data rate of Morse code?

Delay – Processing Delay

- Each station (end host or router) a packet traverses needs some time to carry out packet processing
- This time is called the **processing delay**

Delay – Queueing Delay

- Routers can receive and transmit packets over several lines
- For each incoming packet a router must make a decision to which outgoing line the packet needs to go
- There can arrive several packets for the same outgoing line at about the same time
- The router can start transmission for one of these packets and stores the remaining ones in a **queue**
- Packets in the queue are usually served in FIFO order
- Fix one particular packet: If the packet enters the queue it needs to wait until all previous packets in the queue have been transmitted, only then its own transmission starts
- This waiting time is called **queueing delay**, it is random, as it depends on how many other packets are ahead in the queue

Bit Error Rate

- Signal transmission is subject to distortion, noise and attenuation
- The receiver has to guess the transmitted bits from a distorted and attenuated signal and might guess wrongly
- The rate at which the receiver guesses bits wrongly is called the **bit error rate** or **bit error probability**
- It is often assumed that bit errors happen randomly, statistically independent of each other and happening at the same rate
  - This assumption is often at odds with reality
- **Question**: under these assumptions and assuming the bit error rate is $p \in [0, 1]$, what is the probability of at least one bit error in a packet of length l bits?
- Packets are often equipped with a **checksum** to detect bit errors – if the checksum is wrong then the packet is discarded
- **Question**: is this always the right reaction to a bit error?

Packet Losses

- Packets can be lost for various reasons:
  - End hosts crash
  - Routers crash
  - Packets experience bit errors and are discarded
  - Hosts or routers drop packets due to lack of buffer space
  - Protocols have internal / implementation errors
- Packet losses are often measured as **packet loss rate**, i.e. as percentage of packets (over some time horizon) that gets lost
- How to deal with lost packets?
  - We can retransmit them if they are really needed
    - **Question**: what is needed for that?
  - We can simply ignore the packet loss (only in some applications)

Throughput

- Some choices have to be made in the definition of throughput
- Throughput measures amount of data received per unit time
- What is a suitable unit of time?
  - Very short time intervals: **instantaneous throughput**
  - Longer time intervals: **average throughput**
- Do we include any protocol overheads or retransmissions?
  - If these are excluded, we often speak of **goodput**
- Many applications require some minimum throughput/goodput

Note

These imperfections – delays, packet losses, limited throughput, packet reordering – are what makes the field of networking in the first place!

Typical QoS requirements

Application	Data loss	Bandwidth	Delay-sensitive
File transfer	No loss	Elastic	No
Email	No loss	Elastic	No
Web pages	No loss	Mostly elastic, minimum rate desirable	No
Internet telephony, videoconferencing	Loss-tolerant	Audio/Voice: few kbps – 1 Mbps, Video: 10 kbps – 5 Mbps	$\leq$ 200 - 250 ms
Streaming audio / video	Loss-tolerant	Same as above	A few seconds
Interactive games	Loss-tolerant	few kbps – 1 Mbps	few hundreds ms
Instant messaging	No loss	Elastic	Not very, depends

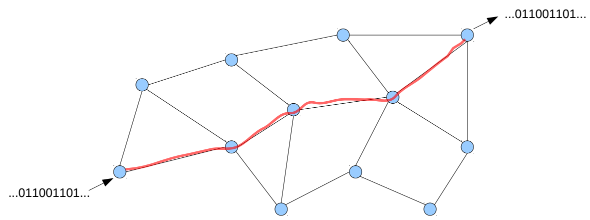
- Table paraphrased from [3, Fig. 2.4]
- The term “elastic” means: these applications have no strict minimum-bandwidth requirements, they are just fine with what they can get

Outline

Communication Patterns

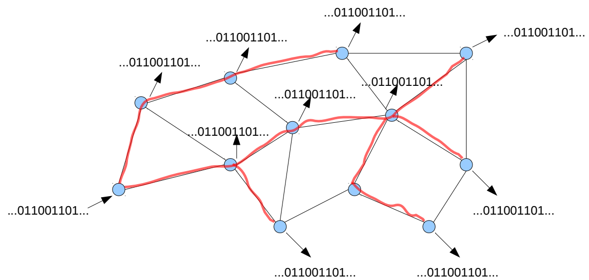
- Guiding question: who communicates with whom?
- We discuss the three most important patterns:
  - Unicast
  - Broadcast
  - Multicast
- There are further patterns, e.g. anycast, convergecast

Communication Patterns – Unicast



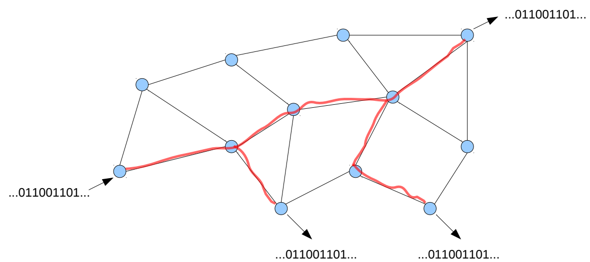
- Only two nodes in the network involved
- One is the transmitter, the other the receiver, but nodes can also have both roles
- Goal: reproduce exactly at the chosen receiver the bit stream sent by the transmitter
- Examples: phone connections, viewing a web page

Communication Patterns – Broadcast



- One node as sender, all other as receivers
- Goal: reproduce exactly at **all** stations in the network the bit stream sent by the transmitter
- Examples: Radio, TV, . . .

Communication Patterns – Multicast



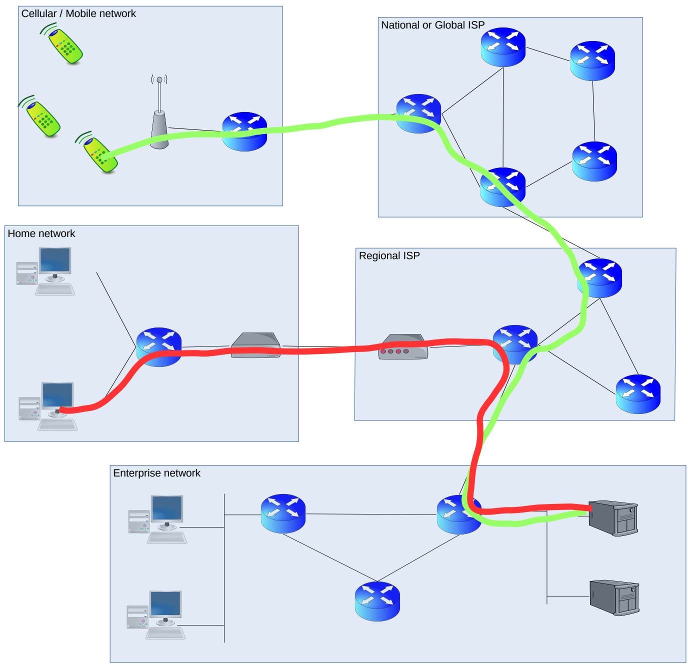
- One node as sender, several, but not all others as receivers
- Goal: reproduce exactly at **some** stations (self-declared or invited members of the multicast group) in the network the bit stream sent by the transmitter
- Often, in multicast groups all nodes can act as sender
- Example: Internet chat, phone conferences

Outline

Networked Application Paradigms

- There are a few fundamental application paradigms that applications are being built on
- These include:
  - Client-server
  - Peer-to-Peer
  - Publish-Subscribe (not discussed here)

Client/Server Paradigm



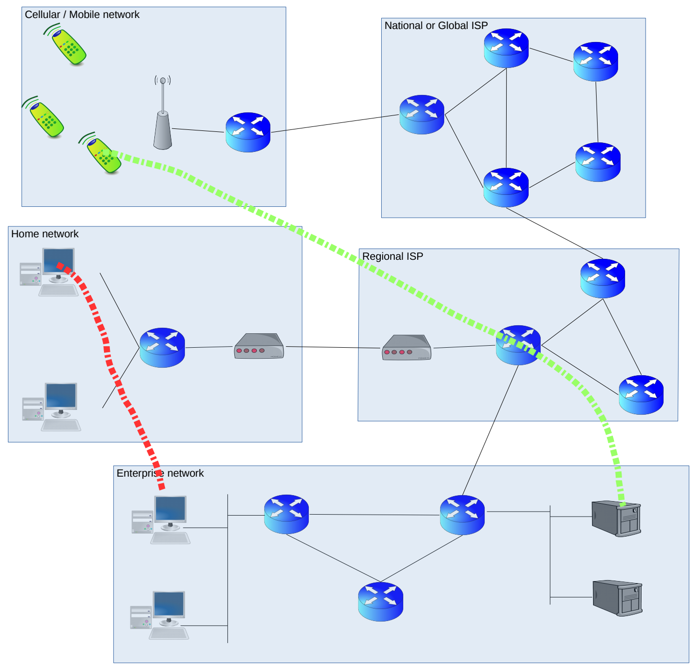
Client/Server Paradigm (2)

- Broadly speaking, there are two roles:
  - A **server** offers service (data, resources), provides it upon request
  - A **client** wants to use a service and submits requests
- Server is well-known for offering this service, waits for client requests all the time
- Example applications: web, email, FTP, airline reservation systems, remote login
- Client application programs:
  - run on an end host
  - request service from a server and proceed when response is there
  - need to know server is available and how to reach it (addressing)
  - Afterwards a client application can end
  - Example: web browser requesting a web page

Client/Server Paradigm (3)

- Server applications:
  - run on an end host
  - need an address that is known to clients
  - are ready to accept service requests and respond to them
  - usually run all the time and support several clients
- Note that:
  - Clients normally don't communicate with each other
  - When n different clients request the same service (e.g. a particular file), the server has to deliver it n times
- Server infrastructure:
  - Most often servers are dedicated and powerful machines
  - Running a single server introduces **single point of failure**
  - For this reason, and to better cope with high demand, often the service is provided by a **server farm**, which to the outside appears to be a single server available under one address (**virtual server**)
  - Example: large web services (e.g. Wikipedia, Google, YouTube)
  - Such heavy-duty server infrastructure is costly

Peer-to-Peer Paradigm



Peer-to-Peer Paradigm (2)

- In Peer-to-Peer (P2P) networks there is:
  - (Almost) No dedicated server providing services exclusively
  - No strict separation of roles, each host sends and receives data
  - No requirement for hosts to be always on
- Rough idea:
  - Suppose there are already n hosts having a certain piece of data
  - The $n+1$-st host requesting it identifies somehow some or all of the n hosts and requests different chunks of the data piece from these
  - In return, the $n+1$-st host agrees to provide data to the $n+2$-nd host, the $n+3$-rd host, and so on
- Many P2P systems are hybrid:
  - The actual data transfer follows the above approach
  - There are in addition centralized servers keeping track of involved hosts, so that host $n+1$ can find some or all of the first n hosts

Peer-to-Peer Paradigm (3)

- P2P systems can scale better than server-based systems
  - Each contributing host can provide service capacity
  - The more hosts, the more service capacity is available
- Main application: P2P file transfer, e.g. BitTorrent

Outline

Traffic: A Key Network Design Factor

- Any network is expected to carry a certain class of traffic
- Examples:
  - The POTS (plain old telephone system) carries voice traffic
  - The Internet carries WWW, P2P, interactive video, ...

Important point

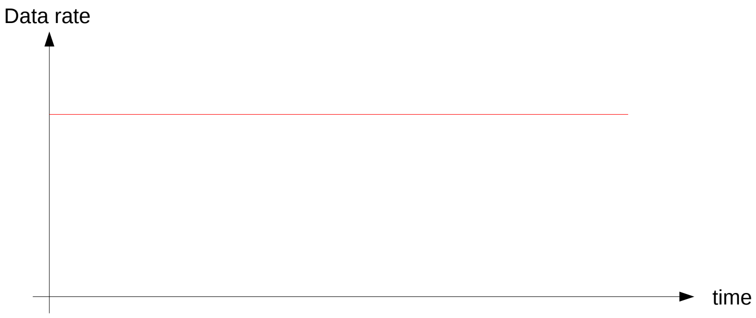
The design of a network is strongly influenced by the traffic it is supposed to carry!

Outline

Voice/CBR Traffic

- In today's POTS voice is transmitted digitally, the analog voice signal is A/D-converted with fixed sampling rate and resolution
  - ISDN: 8 kHz sampling rate, 8 bit resolution $\Longrightarrow$ 64 kbit/s
  - Cellular phone systems add voice compression, e.g. the voice coders in GSM generate between 4.75 and 12 kbit/s
- The voice data is generated continuously at a *fixed rate*
  - This is not always true, some systems transmit nothing when speaker is not active, we then have on-off behaviour
- This is called a **continuous bit rate (CBR)** data stream
- Other examples of CBR data: CBR video, periodic sensor measurements, ...

Voice/CBR Traffic (2)



- This fixed rate must be continuously provided by the network, otherwise users get dissatisfied with voice quality
- Network must support bounded round-trip delay, otherwise users get annoyed (boundary is at $\approx$ 250 ms)
  - See [2] for information on "acceptable" voice transmission
- Phone calls last about three minutes on average

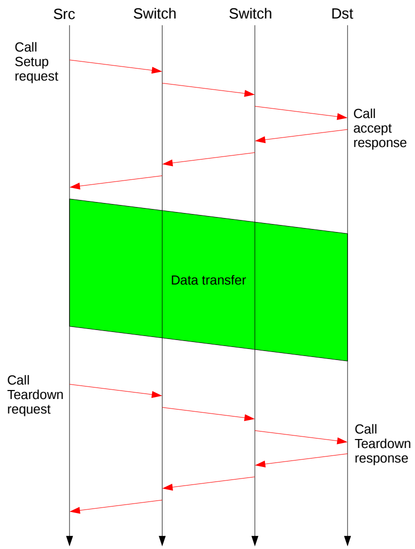
Circuit Switching

- A good way to support CBR traffic is to establish a dedicated "circuit" or "connection" between end points for duration of the call
  - Think of this as a "virtual private cable"
- The lifetime of a connection encompasses three phases:
  - **Connection setup**: identify route, set aside resources (buffers, processing capacity, bandwidth) in switching elements and links, so that resources are *guaranteed*
  - **Connection usage**: use the established connection to transmit CBR data – the pre-reserved resources guarantee that this connection is not influenced by other connections
  - **Connection teardown**: free the reserved resources
  - **Question**: How do you trigger these steps in the POTS?
- Switching elements in CS networks are called **switches**

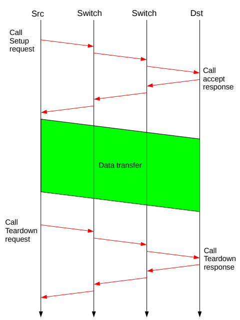
Circuit Switching – Important Properties

- A routing decision is made only once (at connection setup) and never/rarely modified
  - Routing: finding path through network along which circuit is set up
- A connection has its resources guaranteed
- Any bandwidth not used by a connection cannot be re-used by other connections, depending on data rate dynamics this can result in poor utilization
- Connection setup takes time, it does not pay off when only very little data needs to be transmitted
- Connection setup may fail when no route or insufficient resources are available in the network
- Most complex step is connection setup, connection usage is a low-complexity task for switches

Simplified Timeline for Circuit-Switching

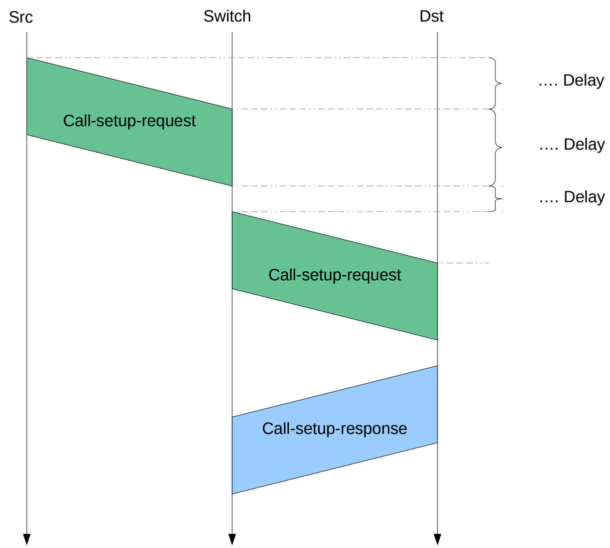


Simplified Timeline for Circuit-Switching (2)



- A `call-setup-request` travels from source to destination (traversing the switches along chosen route), destination accepts by sending `call-setup-response`
- The call setup could also be rejected by a switch or destination, e.g. due to lack of resources in switches or destination being busy in another call
- Each switch and the destination also require some processing time
- Once the call is established, the switches will forward data with negligible delay
- To disestablish connection, we also go through a request / response message exchange

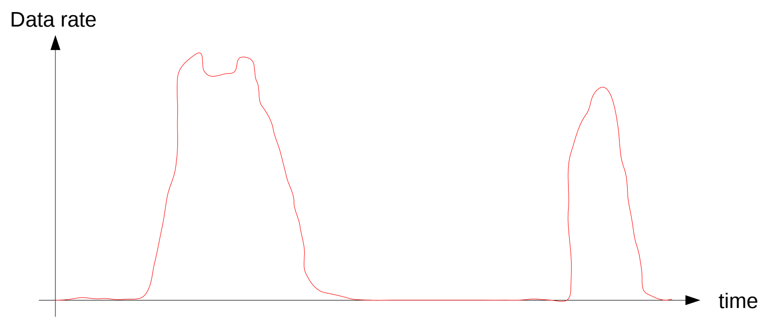
Timeline for Circuit-Switching – Delays Involved



- **Question**: Can you identify the different types of delay involved here?

Outline

Data Traffic



- Many data applications naturally have time-varying rates
- Called **Variable-Bit-Rate (VBR)** or **bursty** traffic
- WWW is an example: users alternate between clicking and thinking/reading

Data Traffic over Circuit-Switched Networks

- CS-networks are not well suited to VBR traffic:
  - Reserved rate is not well utilized during traffic pauses
  - Reserved rate might be too small during traffic peaks
  - No re-use of underutilized connections by other connections

Conclusion

A more flexible networking mechanism is needed for data traffic!!!

Packet Switching

- Data flows are segmented into **packets**
- Packets are basic unit of transmission, not connections
- A packet consists of:
  - A **packet header** containing meta-information about the packet, e.g. address fields (see below)
  - The **packet payload**
  - Possibly a **packet trailer** for error detection / correction
- Packets are transmitted individually and independently
- There is no notion of a connection, packets can be sent immediately without having to set up any state / resource reservation in the network
- Analogy: letter transfer in postal network, envelopes correspond to packet headers
- The Internet is a packet-switched network!

Packet Switching – Consequences

- Let us call a sequence of packets between the same source-destination-pair a **flow**
- Each packet is routed individually, different packets in the same flow can take different routes
  - *Each switching element* (called **router** in packet-switched networks) makes a routing decision *for each packet*
  - Each packet must include information facilitating routing, e.g. header fields for source and destination address
  - Routers do not attempt to identify flows, nor do they store any per-flow state
  - Packets do not necessarily arrive in the same order as they have been sent (packet reordering) [1]
- Many flows can share a link, bandwidth not utilized by one flow can be used by others

Packet Switching – Consequences (2)

- Lack of resource reservation means: there are no(t many) guarantees for packet delivery
  - Internet/IP “best effort” service: packet is delivered – or not
  - IPs lack of guarantees is compensated (in parts) by TCP

Important Point

Routers in packet-switched networks perform more complex processing during information transfer than switching fabrics in circuit-switched networks!

Packet Switching – Congestion

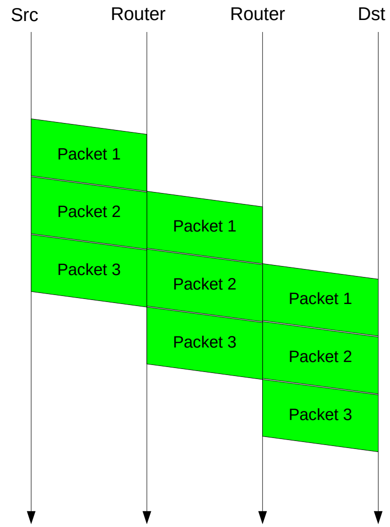
- Since flow data rates and routes often cannot be predicted in advance, routers *buffer* some packets to prevent packet dropping in temporary overload situations
- Routers only have a finite amount of memory, and when overload situation sustains, packet dropping is inevitable, this is called **congestion**
- Important question: which packets to drop?
- **Congestion control** schemes either try to avoid congestion or to deal with it in a graceful manner, e.g. by:
  - Subject packet flows to **admission control**
  - Sending signals to traffic sources to reduce data rates
  - Make good decisions about which packets to drop (e.g. based on priorities) – applications must help here!!
  - Modify the **pricing** during congestion

Packet Switching – Choice of Packet Size

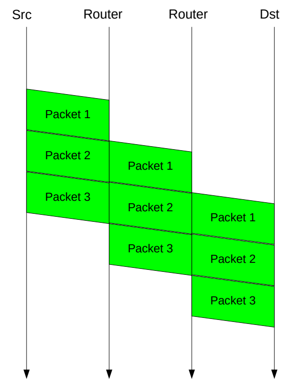
Packet header	Payload	Trailer

- Packet overheads (header, trailer) have a fixed size
- Payload size is variable (within bounds)
- Tradeoff:
  - Small payload size leads to high overhead ratio
  - Small payload size leads to reduced susceptibility to errors
- Packet size limits can be technology- or application-driven
  - Example: too long packets might block important alarm messages for an unacceptably long time (**blocking times**)

Simplified Timeline for Packet-Switching



Simplified Timeline for Packet-Switching (2)



- Compared to circuit-switching no connection establishment and teardown is required
- Source can send packets back to back
- Routers only forward packets when they have them received in full (and there can be some further processing and queueing delay as well)
- Routers can transmit a packet in parallel to receiving and processing packets on other links
- This allows for **pipelining**

Outline

UC UNIVERSITY OF CANTERBURY

Protocols and Services

- Packets in the Internet can get lost, re-ordered, delayed, modified or being tampered with
- Stations (end hosts, routers) can implement procedures to repair these problems
- These procedures run in a distributed fashion (as different stations need to cooperate) and are called **protocols**
- Protocols are rules and procedures underlying data transfer, they:
  - control transmission and reception
  - specify what packets look like
  - specify which packets are sent when (response to other packets)
  - specify what happens when expected responses do not occur
  - are described in standards
  - can make use of other protocols for their services
- Examples from the Internet:
  - TCP and IP provide reliable and ordered packet delivery
  - HTTP is used to request and retrieve information in the web

UC UNIVERSITY OF CANTERBURY

Protocols and Services (2)

- A protocol design problem:
  - Suppose you have a friend in a very remote corner of the world
  - You want to send your friend Lev Tolstoy's book "War and Peace"
  - You want your friend to read it **completely** and in the **right order**
  - You do not have any Internet, but both of you have an infinite supply of postcards, stamps and time
  - You can assume that one page of the book fits on one postcard
- **Question**: How would you achieve that?
  - Devise a set of protocol rules to follow
  - What is a more abstract service provided by your protocol?

UC UNIVERSITY OF CANTERBURY

Protocols and Services (3)

- We normally do not work with protocols directly (unless we implement them) but we use their service through an **application programming interface (API)**
- Example:
  - The **service** offered by the TCP/IP protocol is: reliable and in-order transfer of a stream of bytes over the Internet
  - The API is the so-called socket interface (discussed later)
  - The protocol is the complex set of procedures carried out by TCP and IP to deal with lost packets, to bring packets into the right order, to route packets towards their destination and so forth
- **Question**: assume you want to send a postcard – what could be the service, what the API and what is the protocol?

UC UNIVERSITY OF CANTERBURY

[1] Jon C. R. Bennett, Craig Partridge, and Nicholas Shectman.
Packet Reordering is Not Pathological Network Behaviour.
IEEE/ACM Transactions on Networking, 7(6):789–798, December 1999.

[2] Olivier Hersent, David Gurle, and Jean-Pierre Petit.
IP Telephony – Packet-based multimedia communications systems.
Addison-Wesley, Harlow / England, London, 2000.

[3] James F. Kurose and Keith W. Ross.
Computer Networking – A Top-Down Approach Featuring the Internet.
Addison-Wesley, Boston, fourth edition, 2001.

[4] Larry L. Peterson and Bruce S. Davie.
Computer Networks – A Systems Approach.
Morgan Kaufmann, San Francisco, fourth edition, 2007.

[5] William Stallings.
Data and Computer Communications.
Pearson, tenth edition, 2013.

[6] Andrew S. Tanenbaum and David J. Wetherall.
Computer Networks.
Prentice-Hall, Englewood Cliffs, New Jersey, fifth edition, 2010.