Introduction to CN

Introduction and basics of Networking

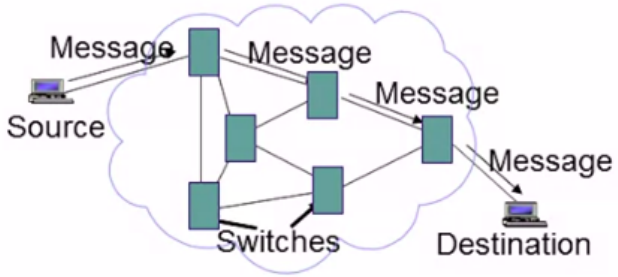
Telegraph service

Elements of Telegraph Networks

- Digital transmission
  - Text messages converted into symbols
  - Transmission system designed to convey symbols
- Multiplexing
  - *Framing* needed to recover text characters
- Message Switching
  - Messages contain source & destination *addresses*
  - *Store-and-Forward:* messages forwarded hop-by-hop across network
  - *Routing* according to destination address

Electric Telegraph Networks

- Electric telegraph networks exploded
  - Message switching & Store-and-Forward operation
  - Key elements: Framing, Multiplexing, Addressing, Routing, Forwarding



Moore's Code

Digital Communications

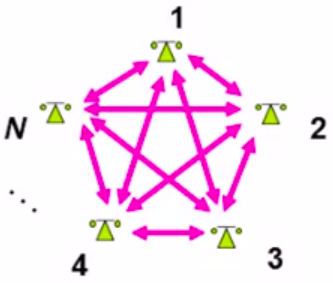
- Morse code converts text message in sequence of dots & dashes
- Use transmission system designed to convey dots and dashes

	Morse Code		Morse Code		Morse Code		Morse Code
A	·—	J	·———	S	···	2	··————
B	—···	K	—·—	T	—	3	···——
C	—·—·	L	·—··	U	··—	4	····—
D	—··	M	——	V	···—	5	·····
E	·	N	—·	W	·——	6	—····
F	··—·	O	———	X	—··—	7	——···
G	——·	P	·——·	Y	—·——	8	———··
H	····	Q	——·—	Z	——··	9	————·
I	··	R	·—·	1	·————	0	—————

N-Square Problem in Networking

The N^2 Problem

- Initially, p2p direct communications - for *N* users to be fully connected *directly*
  - Requires too much space for cables
  - Inefficient & costly since connections not always on



$N = 1000$
$N(N-1)/2 = 499500$

Telephone Networking

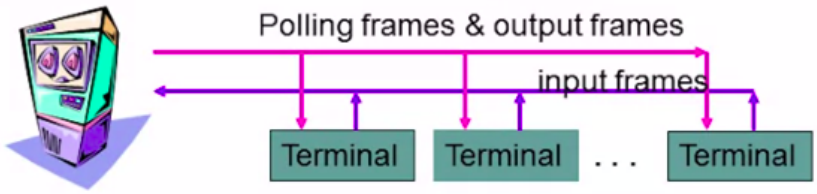
Elements of Telephone Networks

- Digital transmission & switching
  - Digital voice; Time Division Multiplexing
- Circuit switching – Connection oriented
  - User signals for call setup and tear-down
  - Route selected during connection setup
  - End-to-end connection across network
  - Signaling coordinates connection setup
- Hierarchical Network Structure
  - Decimal numbering system
  - Hierarchical structure; simplified routing; scalability

Medium Access Control

Medium Access Control

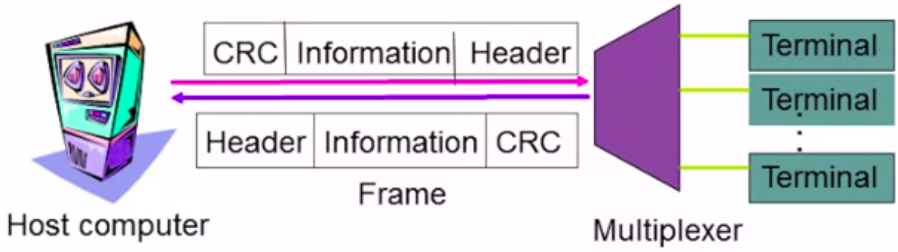
- Dedicated communication lines were expensive
- Terminals generated messages sporadically
- Frames carried messages to/from attached terminals
- Address in frame header identified terminal
- *Medium Access Controls* for sharing a line in arbitrated manner
- Example: Polling protocol on a multi-drop line



Mulitplexing

Multiplexing

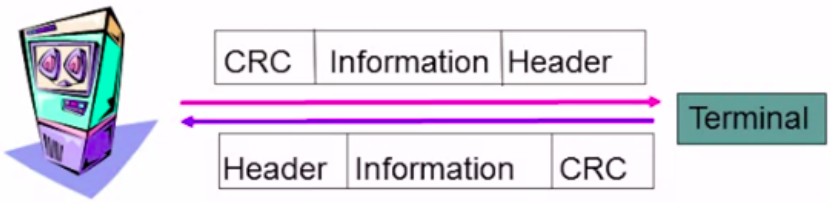
- Multiplexer allows a line to carry *frames* to/from multiple terminals
- Frames are buffered at *multiplexer* until line becomes available, i.e. store-and-forward
- Header carries other *control* information for framing



Error Control Protocol

Error Control Protocol

- Communication lines introduced errors
- Error checking codes used on frames
  - "Cyclic Redundancy Check" (CRC) calculated based on frame header and information payload, and appended
  - Header also carries ACK/NAK control information
- Retransmission requested when errors detected



Packet Switching

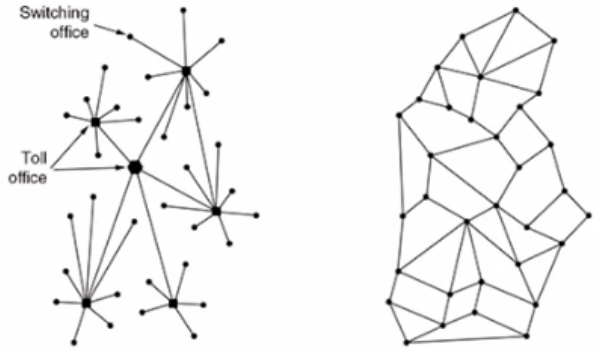
Packet Switching

- Network should support multiple applications
  - Transfer arbitrary message size
  - Low delay for interactive applications
  - Store-and-forward operation could induce high delay on interactive messages
- Packet switching introduced
  - Network transfers packets using store-and-forward
  - Packets have maximum length
  - Break long messages into multiple packets
  - By switching, packets delivered (and reassembled) at destination

The ARPANET

The ARPANET

- The vulnerability of the telephone system was a concern.



(a) Telephone system structure; (b) Distributed switching system structure

The ARPANET Design

- Connection-less packet transmission
- Packets are encapsulated in frames
- Error control uses check bits
- Destinations identified by unique addresses
- Routing tables at the packet switches
- Messages are segmented into packets
- End-to-end congestion control
- Flow control prevents buffer overflow

Protocols and Services

Layers, Services & Protocols

- The overall communications process between machines connected across one or more networks is very complex
- ***Layering*** partitions related communications functions into groups that are manageable
- Each layer provides a ***service*** to the layer above
- Each layer operates according to a ***protocol***

Protocols

- A *protocol* is a set of precise & unambiguous rules that governs
  - how two or more communicating entities in a layer are to interact
  - *Messages* that can be sent and received
  - *Actions* that are to be taken when a certain event occurs

The purpose of a protocol is to provide a service to the layer above

DNS

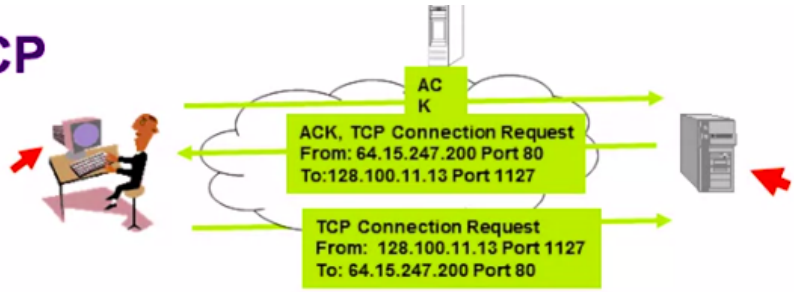
Example: DNS Protocol

- DNS protocol is an application layer protocol
- DNS is a distributed database that resides in multiple machines in the Internet
- DNS protocol allows queries of different types
- DNS usually involves short messages and so uses service provided by UDP

  Well-known port 53

TCP

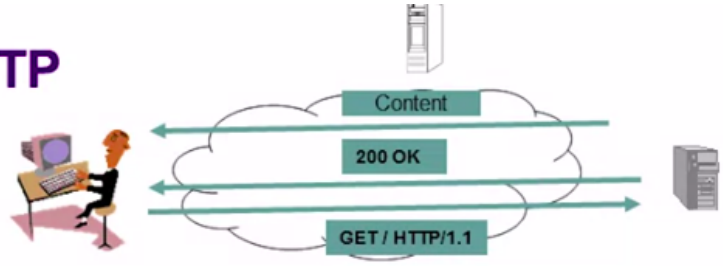
2. TCP



- Browser software uses HTTP to send request for document
- HTTP server waits for requests by listening to a well-known port number (80 for HTTP)
- HTTP client sends request messages through an "ephemeral port number," e.g. 1127
- HTTP needs a Transmission Control Protocol (TCP) connection between the HTTP client and HTTP server to transfer messages reliably

HTTP

3. HTTP



- HTTP client sends its request message: "GET comm.html …"
- HTTP server sends a status response: "200 OK"
- HTTP server sends requested file
- Browser displays document
- Clicking a link sets off a chain of events across the Internet!

Example

Example: HTTP

- HTTP is an application layer protocol
- Retrieves documents on behalf of a browser application program
- HTTP specifies fields in request messages and response messages
  - Request types; Response codes
  - Content type, options, cookies, …
- HTTP specifies actions to be taken upon receipt of certain messages