Overview of Computer Networking

What is Computer Networking?

Logical separation of tasks in digital systems

Computation: Local operations (ALU, load, store, branch, OS, …)

Communication: Data exchange between computation units





```
Local computation
  Request information  --communication-->  Accept request
                                            Process request
                                            Local computation
  Receive information  <--communication--   Send response
Local computation
```


What is Computer Networking?

Logical separation of tasks in a digital system

Computation: Local operations (ALU, load, store, branch, OS, …)

Communication: Data exchange between computation units



Making this work

Rules — lots of rules!

Special hardware

Special software



```
Local computation
  Request information  --communication-->  Accept request
                                            Process request
                                            Local computation
  Receive information  <--communication--   Send response
Local computation
```


Approaches to Networking

What's required

Understanding how people and machines communicate

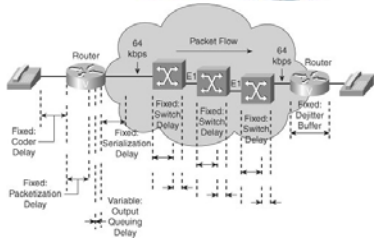


What's technically possible

Network topology (graph theory)

Message encoding (information theory)

Speed and delay (performance theory)



Historical engineering solutions

Division of labor

Hierarchy (top-down)

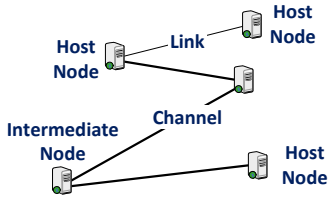
Security





Topology

Computer network as directed or undirected graph



Node

- Host node
  - Network edge — user systems
  - Computer, workstation, …
- Intermediate node
  - Hardware/software systems for data communication
  - Modem, hub, switch, concentrator, multiplexor, router, …

Link

- Transmission path between neighboring nodes

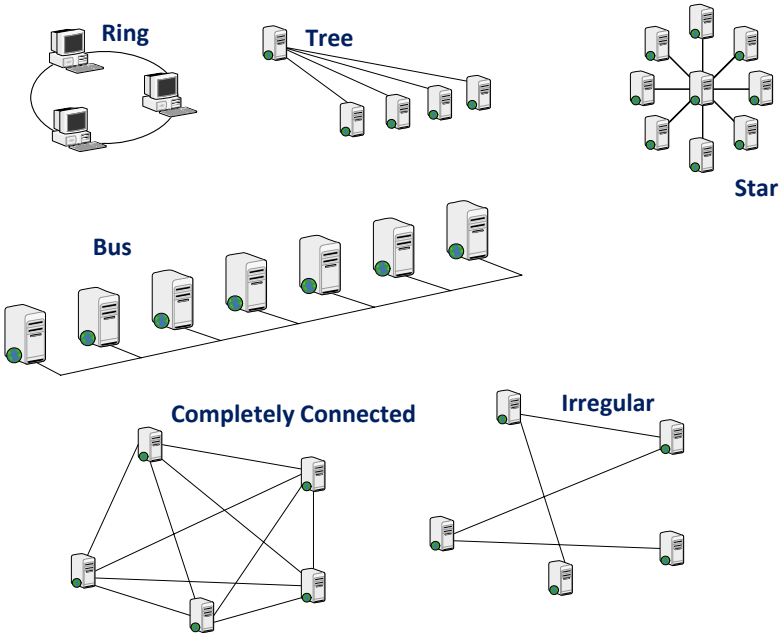
Hop

- Data transfer between neighboring nodes over one link

Channel

- Transmission path between nodes
- May include intermediate nodes

Network Topologies



How People (and Machines) Communicate



Requirements

- Language
- Medium
- Names
- Rules of conversation (protocols)

Preferences

- Keep it simple
- Work with minimum details necessary for specific task
- Obtain details dynamically as needed

Models

- Define roles in computation process
- Define roles in communication process
- Define rules of behavior for each role

Models

Typical roles in computation

- Application program
- Calling function / called function
- OS service
- Class or object

Typical roles in communication

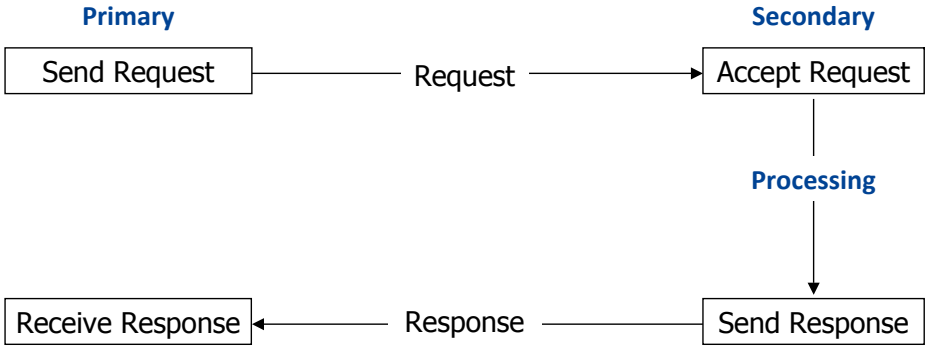
Primary	initiates request and accepts response
Secondary	responds to request
Symmetric	swap roles Primary ⟷ Secondary
Balanced	both roles Primary and Secondary

Example — client/server model

Client and Server	Concurrent application programs / threads
Client	Initiates request to server (Primary)
Server	Responds to client request (Secondary)

Transaction Model

Transaction → request + response



General model with many cases

Familiar examples

Client / Server transaction	Browser requests page from website
Procedural transaction	**`main()`** calls **`function(x)`**

Layered Systems

System divided into logical layers

Within layer

Subsystems interact tightly

Example

```
// subsystems: i, a[i],  b[i], c[i]
for ( i = 0 ; i < 1024 ; i++){
       a[i] = b[i] + c[i] ;
}
```

Between layers

Subsystems interact through programming interface

Example

`main()`

Calling function, Primary

`f(x)`

Called function, Secondary

```
// subsystems: main(), f(x)
main(){
       y = f(x) ;
}
f(x){
       return y;
}
```


Standard Agent Relationships

Agent

Software or hardware entity

Peer relationship

Two+ independent agents at same layer in layered model

Examples

- Independent user application layer programs
  - Microsoft Word + PowerPoint
  - Web Client (browser) + Web Server (website)
- Independent OS layer programs
  - USB driver
  - WiFi driver

Service relationship

`main()` calls **`function(x)`**

Microsoft Word calls printer driver

Application program opens socket (OS call)

Peer-to-Peer Transaction

Peer-to-Peer (P2P)

Transaction between agents of equal level or status

Usually **CLIENT / SERVER** model (not necessarily)

Example

Web service

Browser and web server — application programs (equal status)

Request

Browser (web client) sends page request to web server

Response

Web server sends page content to browser



Primary — web client **Secondary — web server**

Protocol Examples

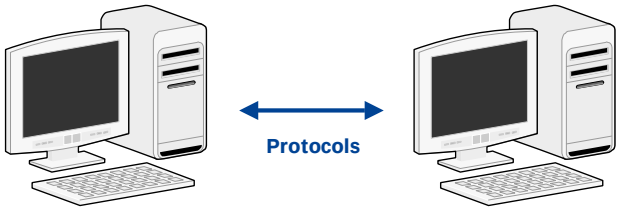
Transaction protocols

Hypertext Transfer Protocol (**HTTP**)

- Browser requests web page from web server
- Web server provides page as response

Post Office Protocol version 3 (**POP3**)

- Client system requests email messages from email server
- Email server provides messages as a response



Service Transaction

Service

Transaction between agents of unequal level or status

Example

User program makes OS call to open file

- User program is application running above OS
- OS performs performs low-level services for applications

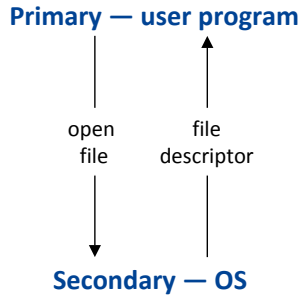
Request

- Application program issues OS call

Response

- OS opens file and returns file descriptor





Service Transaction Example

Calling function → Called function

Request

- Caller invokes called function with parameter

Response

- Called function returns with result

```
user(){
        local work
        response = provider(parameters)
        local work
}
provider(parameters){
        local work
        return response
}
```

Service transaction
Service request
+
Service response

General Layered Service Model

Task divided into layers

Layer n

Provider to layer n + 1

User to layer n – 1

Interface

Boundary between layers

Simple example

Two service transactions

- Layer 3 calls layer 2
- Layer 2 calls layer 1

Layer 2

- Provider to layer 3
- User to layer 1

```
layer_3(){
        local work
        response-2 = layer_2(p3-2)
        local work
}
layer_2(p3-2){
        local work
        response-1 = layer_1(p2-1)
        local work
        return response-2
}
layer_1(p2-1){
        local work
        return response-1
}
```

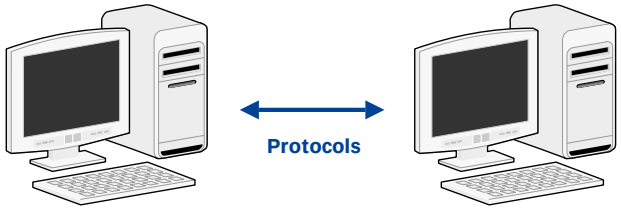
Protocol

Protocol

Rules for transaction between peers

Examples

- Syntax
- Semantics
- Synchronization
- Procedures
- Algorithms
- Naming



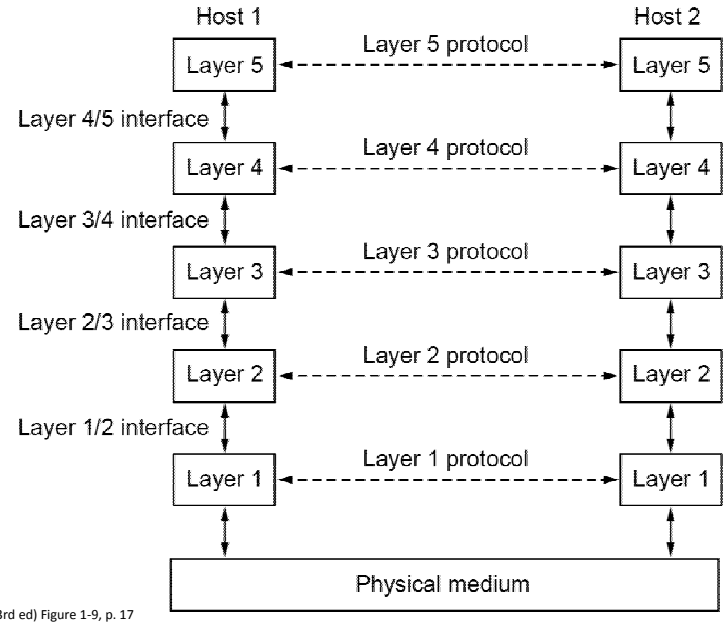
Layered communication

Communication task divided into layers

Protocol stack

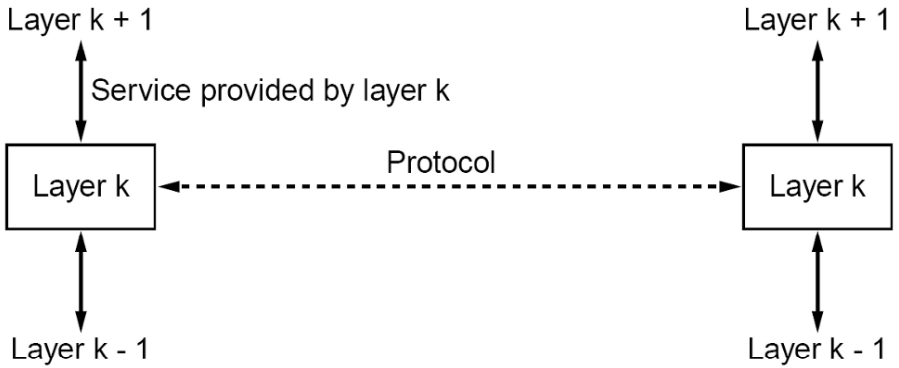
Specific peer-to-peer protocol defined at each layer

Protocol Stack



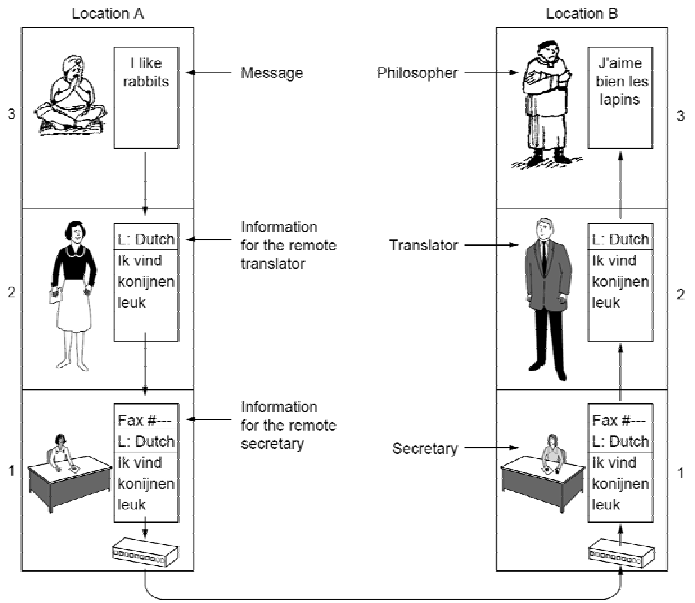
Tanenbaum (3rd ed) Figure 1-9, p. 17

Services and Protocols



The relationship between a service and a protocol

Protocol Stack Example



Tanenbaum (3rd ed) Figure 1-10, p. 19

The philosopher-translator-secretary architecture

Layered Protocol Model

Layer n protocol

Performs **VIRTUAL COMMUNICATION** between layer n peers

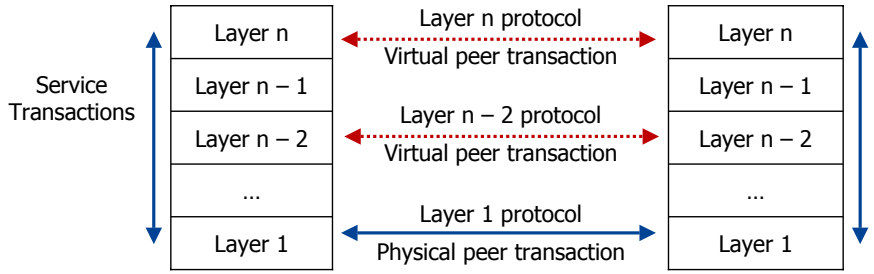
Exchanges layer n information with layer n peer

Layer n service

Receives request from layer n + 1

Passes request to layer n – 1 for communication service

Receives response from layer n – 1



Encapsulation — Protocol Headers

Layer n – 1 protocol

Receives service request from layer n

Request = message to layer n peer agent

Adds layer n – 1 **HEADER**

Header = message to layer n – 1 peer agent

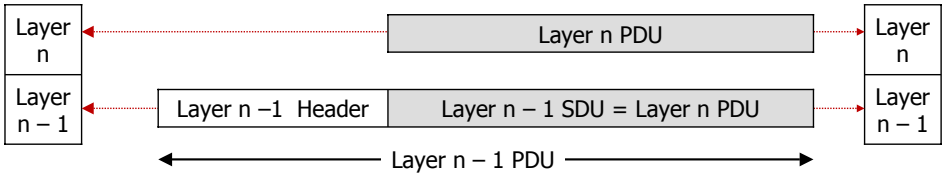
Protocol Data Unit (PDU) at layer n – 1

Message output from layer n – 1 protocol

Layer n PDU + layer n – 1 header

Service Data Unit (SDU) at layer n – 1

Layer n PDU = random data for layer n – 1



Functional Analysis of Communication

Open System Interconnection Model (OSI)

Layer	Function	Description
7	Application	Exchange of data between user applications
6	Presentation	Syntax and semantics of exchanged data
5	Session	Identification, separation, and continuity of multiple ongoing data transactions between software agents
4	Transport	Reliable end-to-end data exchange between host nodes Prevents data loss, errors, repetitions, ordering errors
3	Network	End-to-end data routing between host nodes via multiple hops
2	Data Link	Control of data transmission between neighboring hardware agents (one hop)
1	Physical	Data transmission between neighboring hardware agents on physical channels (electrical, optical, radio, …)

Example of OSI Functional Layers

Hypothetical OSI web browser

Layer	Example Functions
Application	Browser provides GUI — requests web pages by URL
Presentation	Encoding standard for Hebrew (Windows, UTF, ISO, …)
Session	Web page includes multiple graphic files Each file requested and received as separate conversation
Transport	Each request/response checked for errors and completeness Each requested file provided to session layer without errors
Network	Find route to web server by network address File requests/data exchanged with server by network address
Data Link	Data bytes exchanged between host computer and next-hop data communication hardware
Physical	Data bits exchanged with next-hop data communication hardware on physical channels

Internet Functional Model

OSI Layer	OSI Function	Internet Layer	Comment
7	Application	Application	Application provides presentation service and some session service (transactions)
6	Presentation		
5	Session	Transport	Internet session management can be: Reliable — with transport service Unreliable — without transport service
4	Transport		
3	Network	Network	End-to-end data routing as in OSI
2	Data Link	Infrastructure	Internet protocols do not discuss physical data transmission
1	Physical		

Ref: http://tools.ietf.org/html/rfc4949

Example of Internet Functional Layers

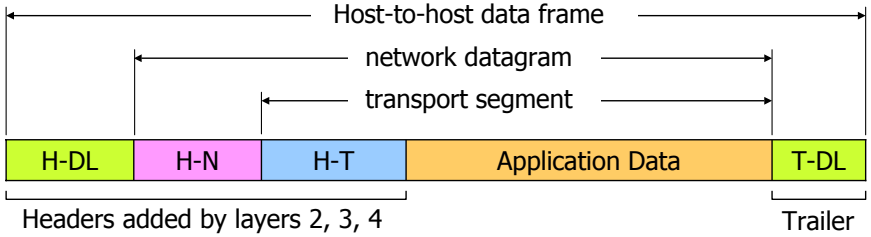
Typical web browser

Layer	Example Functions
Application	Browser provides GUI — requests web pages by URL Translate (DNS) URL into network address (IP) for web server Encoding standard for Hebrew (Windows, UTF, ISO, …) Web page includes graphic files Each file requested/received as separate conversation (HTTP)
Transport	Each file request conversation identified for error control (TCP) Each requested file provided to session layer without errors
Network	File requests/data exchanged with server by network routing (RIP, OSPF, IGRP, BGP) Transfer data across network by network address (IP)
Infrastructure	Network layer messages sent to Internet data communication equipment

Internet PDUs

Protocol Data Unit (PDU)

Layer	Message	PDU
Application	Data	Message
Transport	Header	Segment
Network	Header	Datagram
Data Link	Header + Trailer	Frame
Physical	Bits	Signal



Internet Endpoints

Network Endpoint

Address of **SOFTWARE AGENT** running in **HARDWARE AGENT**

Network Address + Port

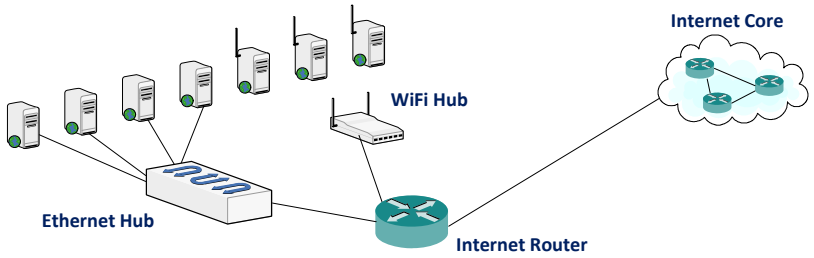
System Level	Layer	Communication ID	
User	Application	Socket	Associates file descriptor with network endpoint
Operating System	Transport	Port	Software address identifies program exchanging data
	Network	Network (IP) Address	Identifies computing node in global network
Hardware	Data Link	Hardware Address	Identifies hardware device (node) in local network
	Physical	Attachment	Physical connection

Well-known ports

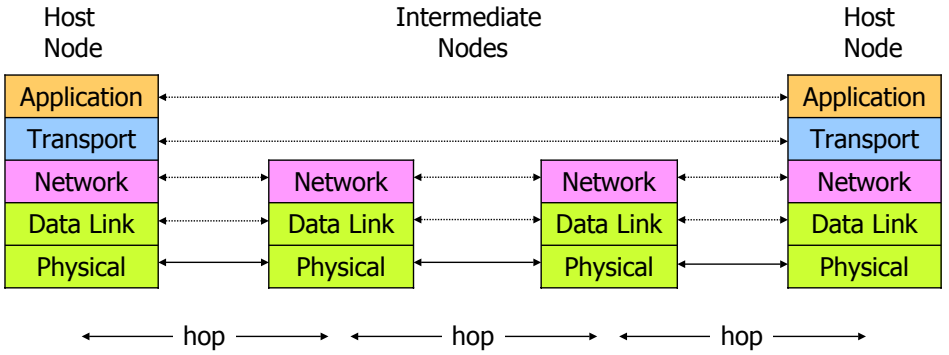
Standard services defined on ports 0 – 1023

Data Communication Equipment (DCE)

Layer	DCE	Function
Network	Router	Receives Network Datagrams in Data Link Frames Sends Datagrams in Data Link Frames to next hop on path to destination
Data Link	Switch (Hub)	Manages physical transmission layer Exchanges Frames among neighboring hardware agents
Physical	Network Interface Card	Modulator/demodulator (modem) Transmits and receives digital bits over physical medium



Internet Hops



Host nodes

- Application data (message) sent to Transport for reliable exchange
- Transport segment sent to Network for addressing and routing

Intermediate nodes

- Examine Network datagrams for addressing and routing
- Treat Transport segment as meaningless data

Network Zoo

Many network types with specific protocol stacks

Wide Area Networks (WAN)

- Public Switched Telephone Network (PSTN)
  - Local loop, backbone, PDH/SDH, ESS, ISDN
  - Public Switched Data Network (PSDN) — X.25
- Broadband Integrated Network
  - ATM, B-ISDN, Frame Relay
- Cellular
  - 2.5G (GPRS/EDGE), 3G (UMTS, CDMA2000), 4G (WCDMA)

Local Area Networks (LAN < 2 km)

Ethernet, WiFi, VLAN, token ring, token bus, FDDI, …

Personal Area Network (PAN < 20 m)

Bluetooth, ZigBee, IrDA, …

Commercial network protocol stacks

SNA, DECnet, Windows Networking, AppleNet, Netware, …

So, what is 'The Internet'?

Internet = Inter-Networking

Protocols for connecting heterogeneous networks

Autonomous System (AS)

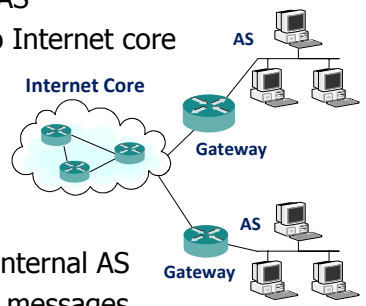
Any network running its own protocol stack

Internet Gateway

- Runs network-specific protocol stack on AS
- Runs Internet protocols on connection to Internet core

Internet core

- Backbone network of Internet routers
- Connected by dedicated links



Typical implementation

- Hosts run network-specific protocols on internal AS
- Hosts use Internet protocols for external messages
- No difference at infrastructure level

Intranet?

Intranet

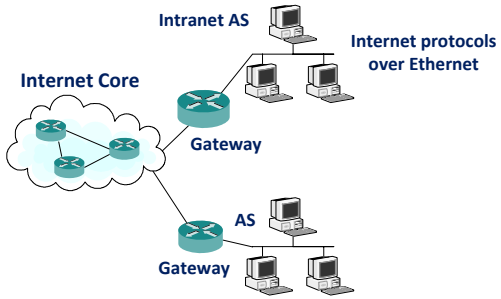
- Using internet protocols in AS

Pure intranet

- Internet protocols above Ethernet/WiFi LAN

Windows network

- Uses Internet protocols for transport and addressing
- Uses Microsoft protocols for message syntax, node location, …

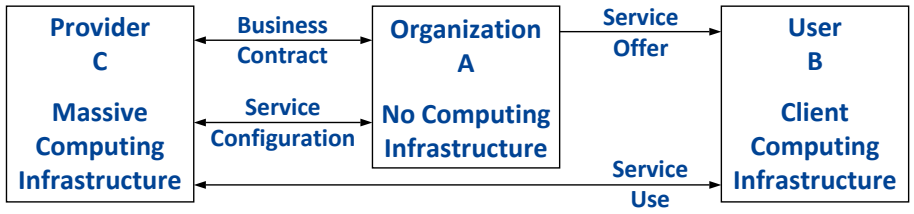


Hey! Hey! You! You! Get Off of My Cloud

words and music: Mick Jagger and Keith Richards

Cloud ≠ Internet ≠ Network

- Network
  - Collection of agents with single defined protocol stack
- Internet
  - Collection of agents using inter-networking protocols at layers 3 & 4
- Cloud
  - Business model
  - Organization A rents computing service from provider C
  - Organization A offers service to user B via provider C network



Why Cloud Computing?

Outsourcing service model

- User gets computing services from service provider
  - Service Level Agreement (SLA) guarantees customer service
  - Provider handles operations+administration+maintenance (OAM)
- Business advantages to organization
  - Economies of scale — large provider can do it cheaper
  - Cuts labor/capital costs from balance sheet → happy investors

Based on standard technologies

- Cloud service organized from conventional resources
  - Hardware + software + network
  - Provider offers menu of services
- Not a fundamentally different computing technology

Unique technological issues

- Service reliability — provider committed to SLA
- Optimization of provider-side resource configuration
- Optimization of user-side resource configuration

Service Configuration in Cloud Computing

Infrastructure as a service (IaaS)

- Organization sees virtual hardware environment
  - Real hardware or hypervisor / system virtual machine
- Organization installs OS → installs software → user runs jobs

Platform as a service (PaaS)

- Organization sees virtual OS environment
  - OS on single hardware platform or virtual OS
- Organization installs software → user runs jobs

Software as a service (SaaS)

- Organization sees virtual application software environment
  - Applications running on private OS or "sandboxed" on shared OS
  - Sandbox — private execution environment per application instance
- User runs jobs

Storage as a service (STaaS)

- User sees virtual mounted storage device

Centralize → Decentralize → Centralize → ?

1950s — 60s

- Centralized mainframe computer + multiple OS instances over hypervisor
- Timesharing OS serves multiple users
- User sees OS environment via dumb terminal (thin client)

1970s

- User applications offloaded to minicomputers + timesharing services
- User sees timeshared OS environment via dumb terminal

1980s

- User applications offloaded to personal workstations (PC)
- User sees single-user OS environment running locally

1990s

- Network single user workstations
- User sees single-user OS environment running locally

2000s

- Centralized control of local OS environment by IT departments

2010s

- Cloud + netbook / tablet / smart phone — dumb terminal with high-res GUI

Issues in Cloud Computing

Cost

- Provider issues
  - Economies of scale ⇒ lower cost per compute job
- Organization issues
  - Capital + OAM costs → operating costs
  - Lower start-up costs ⇒ operating debt

Reliability

- Provider issues
  - Redundant infrastructure → continuity + disaster recovery
  - Centralized management of OAM, security, performance
  - Virtualization → serve multiple users on physical server
  - Multitenancy → provide multiple sandboxed application instances on OS
- User sees guaranteed service

Agility

- Organization / provider reconfigure service as needed
- Growth, load balancing, time-zone serving

Cloud Ownership

Public cloud

- Service provider as public utility — sells / rents computing service
  - Initial providers leverage large existing infrastructure
  - Amazon, Microsoft, Google, IBM
- Menu of services at fixed prices

Private cloud

- Cloud infrastructure for private organization
- Managed internally or outsourced
- Isolates service developers from implementation issues
  - Standard development platform
- Requirements for economic justification
  - Large organization
  - Technology-based services
  - Frequent new service
  - Example — internet content provider

Programming in the Cloud

Depends on environment

- IaaS — Organization sees virtual hardware environment
- PaaS — Organization sees virtual OS environment
- SaaS — Organization sees virtual application software environment

IBM Bluemix

- SaaS from IBM
- Free accounts for students using **name@hadassah.ac.il** address
- Bluemix DevOps Services
  - Develop, track, plan, and deploy software on IBM cloud service
  - Collaboration tools — Git, Jazz SCM, GitHub
  - Build application → deploy to IBM cloud
  - Supports
    - Arduino, C, C#, C++, CSHTML, Embedded, JavaScript (ejs) Erlang, Go, HTML, abstraction markup language (Haml) Jade, Java, JSON, Lua Objective-C PHP, Python, Ruby, Swift, Virtual, Basic (vb) VMHTML, XHTML, XML, Xquery, yaml, Launch, file Dockerfile, gitignore, git config, cfignore
  - "You can go from source code to a running app in minutes."

Some Internet Protocols

Application layer transactions

Hypertext Transfer Protocol (HTTP)

Transport layer

Transport Control Protocol (TCP)

Reliable transport service

User Datagram Protocol (UDP)

Unreliable transport service

Network layer

Internet Protocol (IP)

Node addressing

Internet Control Message Protocol (ICMP)

Messages about messaging

Routing protocols (RIP, OSPF, IGRP, BGP)

Learn network topology for message forwarding

RFC — Internet standard

Protocol	RFC
HTTP	2616
TCP	793
UDP	768
IP	791
ICMP	792

What Internet Protocols Do
Some examples

Hypertext Transfer Protocol (HTTP)

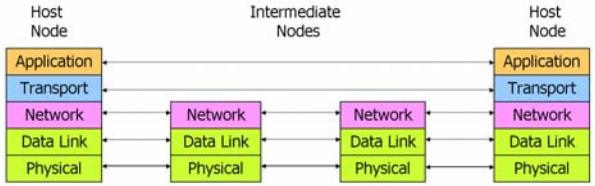
Application layer transactions

Requests

Get	Retrieve file by name
Post	Replace file by name
Delete	Delete file by name

Responses

Data	Contents of requested file
Status	Status of transaction



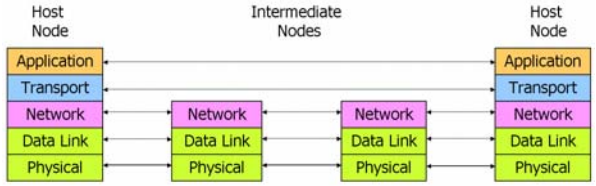
What Internet Protocols Do
Some examples

Domain Name Service (DNS)

Translates node name to Internet address (and vice versa)

Example

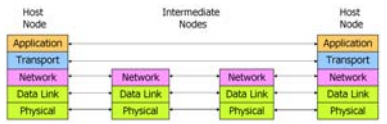
```
c:\> nslookup www.hadassah.ac.il
Name:    www.hadassah.ac.il
Address: 212.179.79.228
```



What Internet Protocols Do
Some examples

Transport Control Protocol (TCP)

Reliable transport service



Sender

Label source and destination software by port number

Number outgoing segments

Wait for **ACK** (acknowledgment) for outgoing segments

Retransmit segments if no **ACK** before timeout

Negotiate segment size (for error and congestion control)

Receiver

Check completeness and order of incoming segments

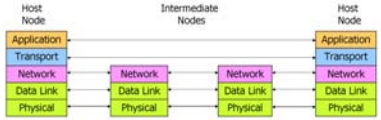
Check incoming segments for errors

Send **ACK** for good segments

Provide good incoming segment to destination software

What Internet Protocols Do
Some examples

Internet Protocol (IP)



- Best effort network service
- No guarantee of delivery

IP version 4 address

- Four octets 0.0.0.0 to 255.255.255.255 (many reserved addresses)

Sender

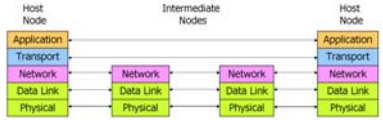
- Attach source and destination network addresses to segment
- Route IP datagram to next hop along route

Receiver

- Intermediate node — route IP datagram to next hop along route
- Host node — provide segment to transport layer

Network Infrastructure
Layers 1 + 2 — bits, bytes, signals, cables, electronics

Scale



- Wide Area Network (WAN < earth)
- Local Area Network (LAN < 2 km)
- Personal Area Network (PAN < 30 m)

Medium

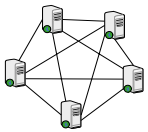
Copper wire and cable Electrical signals	Requires legal right to install cables
Optical fiber Light wave signals	
Open space Radio wave signals	Requires legal right to transmit radio

Traffic statistics

- Constant Bit Rate (CBR) — peak data rate = average data rate
- Variable Bit Rate (VBR) — peak data rate > average data rate

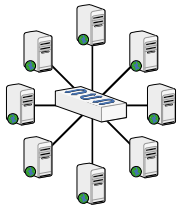
Connectivity = Medium + Topology

Point-to-point



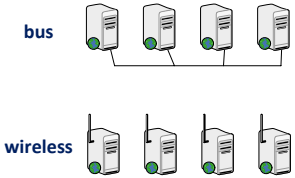
- Dedicated link from node to node
- Fastest and most complex

Switch



- Dedicated link from node to switch
- Switch connects nodes on request
  - Non-blocking provides $n \times (n - 1)$ connectivity
  - Blocking provides $n \times m$ connectivity ($m < n - 1$)

Shared medium



- Nodes share medium access
- Contention
  - Nodes compete for access
- Polling
  - Central controller polls nodes

Physical Transmission

Serial data rate at physical layer

- Bits per second = bps = b/s
- Bytes per second = B/s
- 1 B/s = 8 b/s

Capacity (bandwidth)

- Maximum data rate on medium
- Fixed by transmitter / medium / receiver
- Limits
  - Speed of circuits
  - Signal to noise ratio (SNR)



Physical Transmission

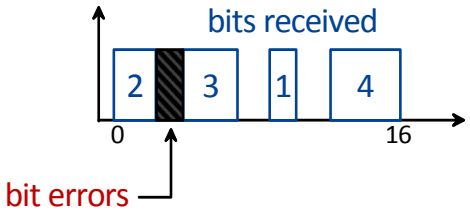
Throughput

$$\text{throughput} = \frac{\text{error-free data received per second}}{\text{capacity}}$$

Takes account of

- Utilization = % time transmitter sending
- Errors ⇒ re-transmission ⇒ more data on same capacity
- Delays ⇒ less data received on same capacity

utilization = 10 / 16 = 62.5%



Baud Rate
Symbols per second

Symbol

Physical signal that encodes bits

Symbol rate (Baud rate)

Symbols transmitted per second

Bit transmission rate

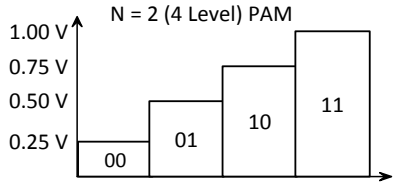
Bits transmitted per second = (symbols / second) × (bits / symbol)

Example

Pulse amplitude modulation (PAM)

Define 2^N electrical levels from 0 to 11…1

Each symbol (level) transmits N data bits



Baud Rate
Symbols per second

33 kbps dial-up modem

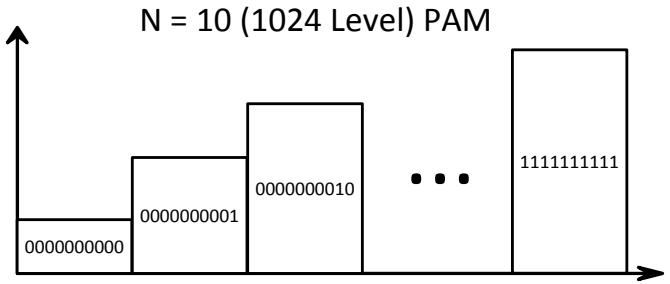
Define 2^{10} = 1024 electrical symbols (max for SNR on phone line)

Baud rate = 3300 symbols / second

Bits transmitted per second

Data rate = (3300 symbols / second) × (10 bits / symbol)

= 33,000 bps



Data Concentration

High capacity link

No single node can utilize link capacity

Example

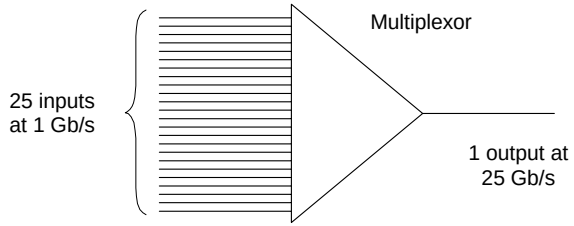
- Optical fiber cable with 4 fibers at 25 Gbps = 100 Gbps

Multiplexing

Combine multiple nodes onto one link

Example

- Optical fiber with 25 Gbps data rate
- Combine 25 nodes transmitting at 1 Gbps



Multiplexing Methods

Frequency Division Multiplexing (FDM)

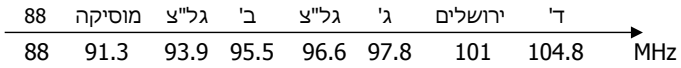
Divide available frequencies (bandwidth) among nodes

Nodes transmit simultaneously on different frequencies

Example

FM radio uses 88 MHz to 108 MHz = 20 MHz bandwidth

Divide 20 MHz into 100 channels = 200 kHz per FM channel



Multiplexing Methods

Time Division Multiplexing (TDM)

Divide capacity into time slots

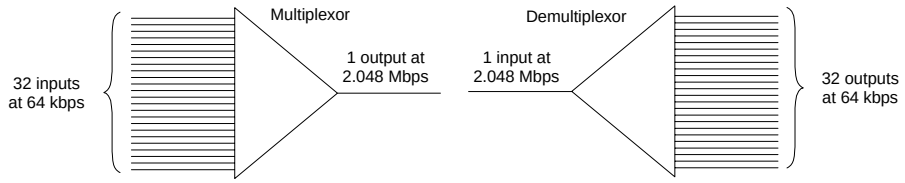
Node transmits in assigned time slot

Example

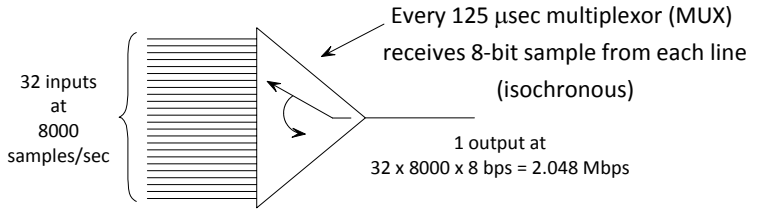
E1 digital line transmits at 2048 kbps

Divide 2048 kbps line into 32 time slots = 64 kbps per node

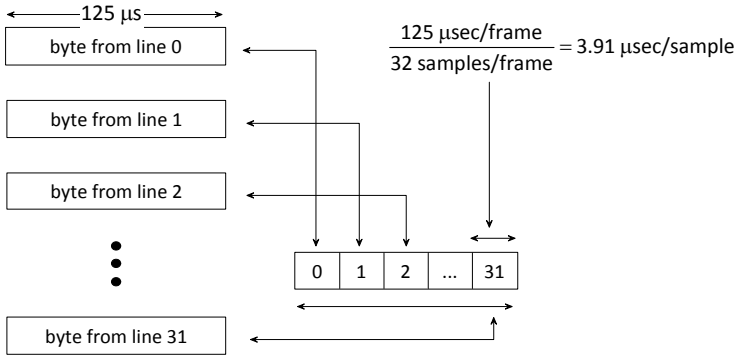
32 x 64 kbps = 2048 kbps = 2.048 Mbps



E1 Multiplex



$$\frac{1}{8000 \text{ samples/second}} = 125 \text{ µs/sample}$$



$$\frac{125 \text{ µsec/frame}}{32 \text{ samples/frame}} = 3.91 \text{ µsec/sample}$$

Mixed Multiplexing
GSM Cellular

Time Division Multiple Access (TDMA)

Used on GSM / UMTS phones — 2G and 3G

Combines FDM and TDM

Frequency Division Multiplexing (FDM)

GSM bands = 25 MHz

Divide 25 MHz into 125 channels = 200 kHz per channel

Transmit 270 kbps over 200 kHz channel

Time Division Multiplexing (TDM)

Divide 270 kbps into 8 times slots = 33 kbps per user

33 kbps = 23 kbps for voice + 10 kbps control

Data Statistics — CBR

Constant Bit Rate (CBR)



- Isochronous data
  - Equal time interval between bits
  - Bits per second = constant

Average data rate

- Average data rate = peak data rate = minimum data rate

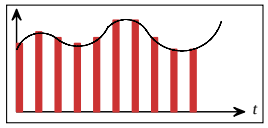
Example

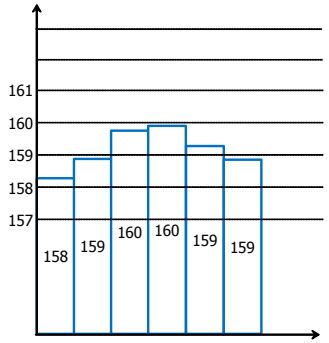
- Uncompressed digital audio
- Sample analog signal every T seconds
- Round-off sample to N-bit number from 0 to $2^N - 1$
- Digital audio stream at N / T bps

Digital Voice on Telco Telephone

Sample analog voice signal every 0.125 ms

- 0.125 ms per voice sample ⇒ 8000 voice samples / second





Round-off sample to 8-bit data

- Data ∈ {0, 1, 2, ... , 255}
- Sample = {158.276, 158.879, 159.724, 159.821, 159.312, 158.791}
- Data = {158, 159, 160, 160, 159, 159}

DS-0 stream

- (8000 samples / second) × (8 bits / sample) = 64 kbps
- 64 kbps digitized voice (no compression)

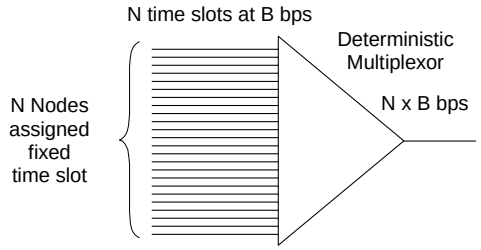
Multiplexing Statistics

Deterministic multiplexing (CBR)

- N Nodes = N time slots
- Node reserves fixed time slot
  - Guaranteed transmission capacity
  - Node transmits in assigned time slot

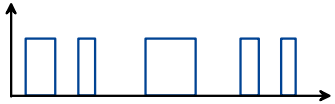
Example

- E1 multiplex for wired telephone — 32 x 64 kbps = 2048 kbps
- E2 multiplex — 4 x 2048 kbps = 8192 kbps



Data Statistics — VBR

Variable Bit Rate (VBR)



- Bursty data
  - Peak data rate B > average data rate λ
  - Assume packets are independent (Poisson statistics)

$P(k,T,\lambda)$ = probability of k bits arriving in T seconds when average rate = λ

$$P(k,T,\lambda) = \frac{(\lambda T)^k}{k!} e^{-\lambda T}$$

Example

- Data sent by time-of-day client
  - Request time-of-day (1000 bits) once every hour (3600 seconds)
  - Average data rate = 1000 bits / 3600 seconds = 0.28 bps
- Peak data rate = 55 Mbps on 802.11g WiFi
  - Peak data rate 55 Mbps > average data rate = 0.28 bps

Multiplexing Statistics

Statistical multiplexing (VBR)

M nodes > N time slots

Bursty data

Average data rate λ < peak data rate B

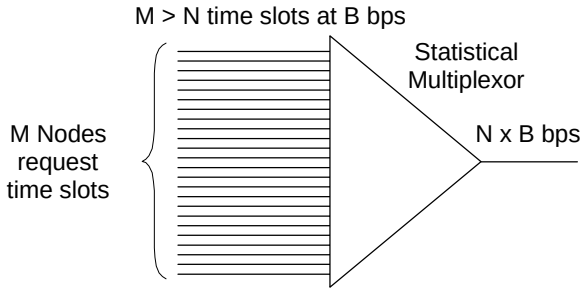
Average traffic rate = M x λ < capacity rate = N x B

Actual traffic < capacity ⇒ OK

Actual traffic > capacity ⇒ data delayed or lost

Example

Internet routers



Overflow in VBR

Overflow

Actual traffic > capacity

Short time (a few time slots) ⇒ data delayed

Long time (many time slots) ⇒ buffer overflow ⇒ data lost

Overflow probability

Average traffic rate = M x λ

Average data arriving in time T = M x λ x T

Capacity rate = N x B

Data capacity in time T = N x B x T

Overflow in time T

Actual data arriving in time T > N x B x T

N x B x T + 1 or N x B x T +2 or N x B x T +3 or ...

Independent outcomes

$$P(\text{overflow}) = P(\text{NBT}+1 \; or \; \text{NBT}+2 \; or \; ...) = \sum_{k=\text{NBT}+1}^{\infty} \frac{(\text{M}\lambda\text{T})^k}{k!} e^{-(\text{M}\lambda\text{T})}$$

Overflow Example

Average traffic on network

λ = 10 packets / second per node

M = 10 nodes

Average packets in 0.1 second = M x λ x T

= 10 nodes x (10 packets / second per node) x 0.1 second

= 10 packets

Maximum traffic on network (capacity)

B = 25 packets / second per node

N = 4 nodes

Maximum packets in 0.1 second = N x B x T

= 4 nodes x (30 packets / second per node) x 0.1 second

= 12 packets

Overflow condition for T = 0.1 second

Overflow if actual traffic > N x B x T

$$P(\text{overflow}) = \sum_{k=13}^{\infty} \frac{(10)^k}{k!} e^{-(10)} = 0.21 = 21\%$$

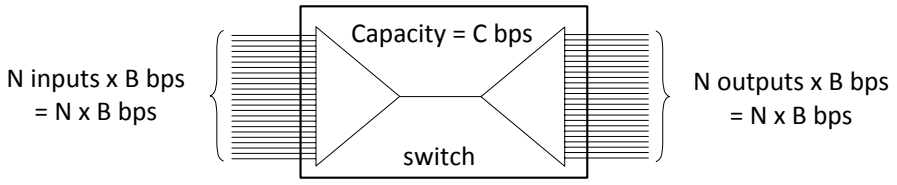
Switching

Switch

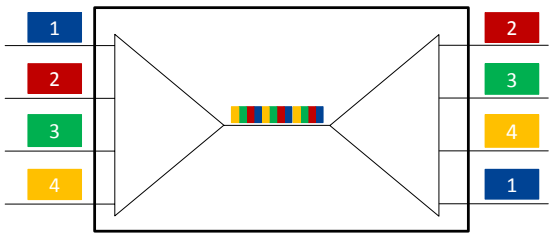
Multiplexor + Demultiplexor

Data at **input_port$_i$ → output port$_j$**

i,j = 0, 1, 2, ... , N - 1



Example



Circuit Switching

Deterministic multiplexing

Capacity C = N × B

Dedicated (reserved) link

input_port$_i$ → output port$_j$

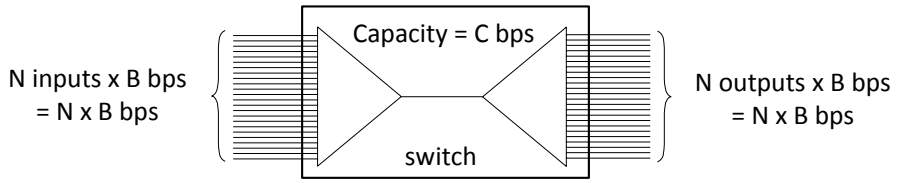
No competition

Guaranteed capacity B — if used or not

Example

Bezeq phone call

64 kbps from telephone to telephone (even if no one speaks)



Packet Switching

Statistical multiplexing

Capacity C = M × B < N × B

Dynamical time slot assignment (on request)

input_port$_i$ → output port$_j$

Competition

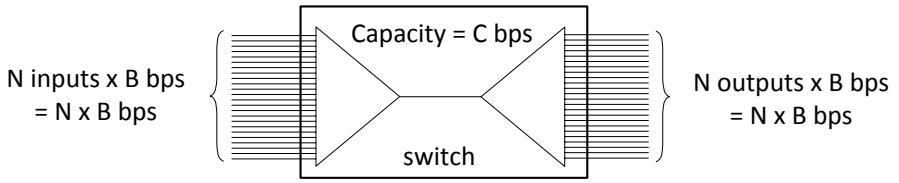
More ports than capacity

Demand > capacity ⇒ delay

Example

Internet router

Packet queue — first come first served



Connection Types

Connection

State machine associated with data exchange

Connection-oriented

First set-up data channel

Multiple data transactions associated with connection state

Monitor channel state during data exchange

Close channel after data exchange

Example — **phone call**

Enter number → answer call → extended conversation → disconnect

Connectionless

Transmit data with no prior channel set-up

No channel state defined by nodes

Each message independent

Example — **email message**

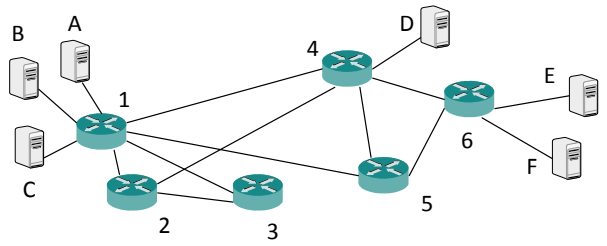
Send email → hope message arrives → hope message is found / read

Datagram Service

Network of routers and links

Packet switching

Connectionless



Each datagram

Has source and destination address in header

Data Link header or Network header

Routed individually through network

Datagrams may follow separate routes

Example

src = B dest = F	data

B → 1 → 4 → 6 → F

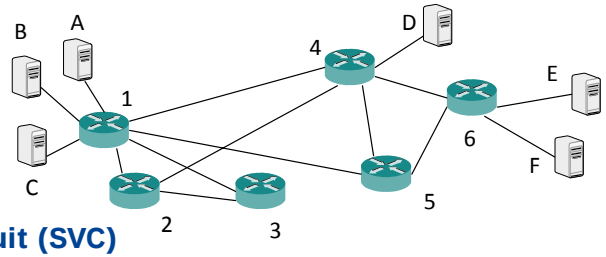
B → 1 → 5 → 6 → F

Switched Virtual Circuit (SVC)

Network of switches and links

Circuit switching or packet switching

Connection-oriented



Switched Virtual Circuit (SVC)

Set-up / close messages carry source and destination addresses

Example

Set-up VC – 1: B → 1 → 4 → 6 → F

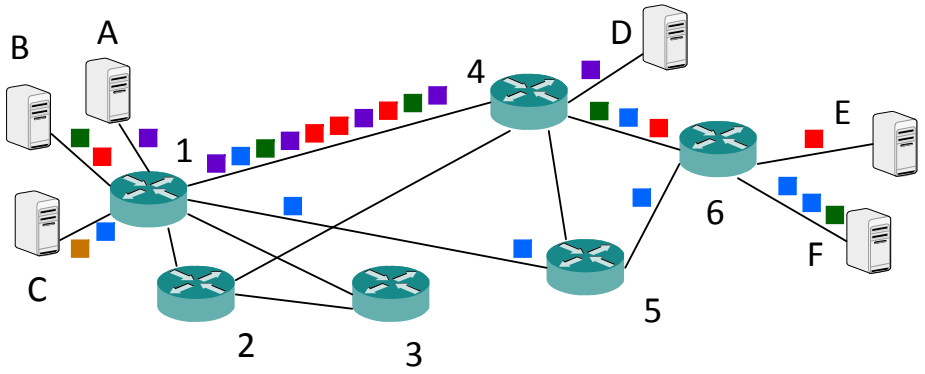
Packet routing by VC ID in header (layer 2 or layer 3)

Every packet follows same VC route

Example

VC – 1	data

Switching Example



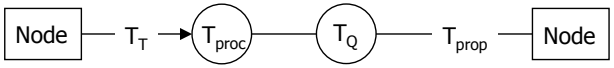
A to D — circuit mode (deterministic SVC)

B to E — packet mode (statistical SVC)

B to F — packet mode (statistical SVC)

C to F — packet mode (datagram service)

Transmission Delay



Transmission delay T_T

T_T = Time to inject bits into line = (bits in packet) / (bits per second)

Example: 1000 Mb / 100 Mbps = 10 sec

Processing delay T_{proc}

Packet process time in intermediate node

SVC with fixed route ⇒ shorter delay than datagram routing

Propagation delay T_{prop}

T_{prop} = (length of cable) / (signal speed)

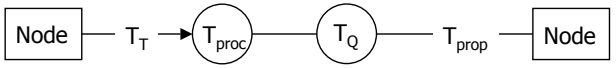
Example: 4 km / (2×10^8 km/s) = 2×10^{-8} sec << 10 sec

Queuing delay T_Q

Time packet waits in buffer for previous packets (congestion)

T_Q = (service time per packet) × (packets waiting in buffer)

Example of Queuing Delay



Queuing delay T_Q

T_Q = (service time per packet) × (packets waiting in buffer)

Packets waiting in buffer = 1 / (1 – utilization)

Queuing delay example

Service time per packet = 10 ms / packet

Service rate = 1 / (10 ms / packet) = 100 packets / second

Average traffic = S = 85 packets / second

Utilization = (85 packets / second) / (100 packets / second) = 0.85

Buffer level = 1 / (1 – 0.85) = 6.67

T_Q = (10 ms / packet) × 6.67 packets = 67 ms

C = switch capacity = service rate = 100 packets / second

Demand > 100 buffer ⇒ overflow ⇒ excess delay

$$P(\text{demand} > C) = \sum_{k=C+1}^{\infty} P(\text{demand} = k) = \sum_{k=C+1}^{\infty} \frac{S^k}{k!} e^{-S} = \sum_{k=101}^{\infty} \frac{85^k}{k!} e^{-85} = 0.05$$

Error Control

Bit error

Data 1 received as 0 or data 0 received as 1

$$\text{Bit Error Rate (BER)} = \frac{\text{bit errors in received data}}{\text{bits in received data}}$$

Packet Loss

Congestion or buffer overflow → packet discarded

$$\text{Packet loss rate} = \frac{\text{packets lost}}{\text{packets transmitted}}$$

Error detection

Error correction code / redundancy code / checksum

Checksum transmitted with data in header / trailer

Receiver compares independent hash with transmitted code

Error control

Required

- Discard corrupt packet

Optional

- Retransmit discarded / missing packets

Network Scale

Private network

Small Office / Home Office (SOHO)

- Small number of computers in a few rooms
- Simple Ethernet / WiFi LAN

Enterprise

- Many nodes in large building / campus
- Complex Intranet

Access network

Provide user connection to Internet core

Infrastructure provider manages layers 1 and 2

Internet Service Provider (ISP) manages layers 3 and 4

Internet core

Network of routers and links at layer 3

Infrastructure provider manages links at layers 1 and 2

Links are typically built over complex network systems

Private Networks

Simple Ethernet / WiFi LAN

Ethernet switching hub

- 4 to 16 nodes
- Full connectivity (non-blocking)
- 10 / 100/ 1000 Mbps

WiFi hub

- More nodes lowers performance
- Nodes compete to transmit to hub
- 11 / 54 / 100+ Mbps

Complex Intranet

Multiple LAN hubs

Hubs connected

- Directly (bridging)
- Indirectly (routing)

Non-Private Networks

Access + core

Service infrastructure

Routing + accounting nodes in office buildings

Link infrastructure

Cables + radio channels on public / private property

- Legal and licensing issues

Controlled by companies in cable businesses

- Telephone companies (Telco)
- Cable TV companies
- Electric companies
- Railroads companies

Choices for small business Intranet at 3 locations

Pay service provider monthly

Or

- Purchase LAN hubs and routers
- Lease cables from Telco

Telephone Network
It's everywhere

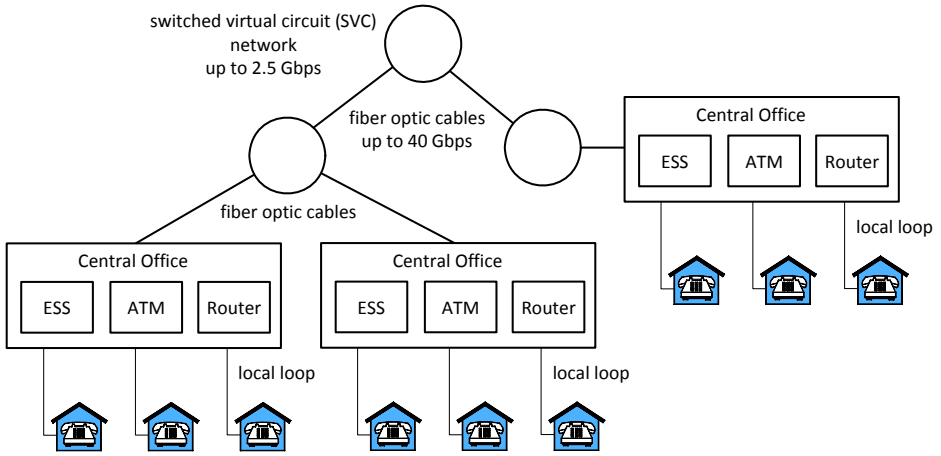
Local loop
- Wired connection to most buildings
- Can carry 1 Mbps (up to 4 km) to 25 Mbps (up to 300 m)
- Voice network
  - Analog voice channel from 300 to 3300 Hz
  - Digitized voice at 64 kbps

Local presence (central office) in every neighborhood
- Local loop attached to non-blocking switches

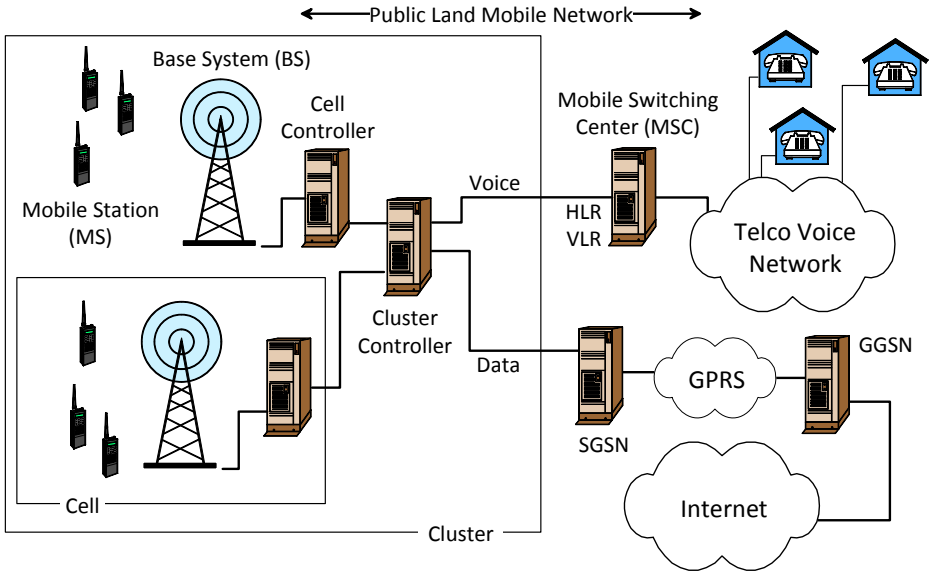
Tree network of switches
- Central offices connect to regional offices on fiber optic backbone

Global broadband switched virtual circuit (SVC) network
- Circuit mode switches (ESS7) for 64 kbps voice
- Circuit / Packet mode layer 2 switches (ATM) up to 2.5 Gbps
- Private routers throughout network for Internet traffic

Telephone Network



Cellular Network
Wireless to base station — uses Telco network for WAN service



SOHO Access Networks

Dial-up modem (modulator / demodulator)
- Converts digital bits from computer to analog signals for phone line
- User modem connects to ISP modem by phone call
- 56 kbps downstream / 33 kbps upstream

Digital Subscriber Line (DSL)
- FDM on local loop
- Voice channel connected to telephone voice network
- Data channel — 15 Mbps downstream / 750 kbps upstream
  - ATM link between DSL modem and Telco central office
  - Datagrams routed to ISP on Telco router network

Cable modem
- FDM on TV cable
- TV channels connected to TV
- Data channel — 30 Mbps downstream / 2 Mbps upstream (shared)
  - Ethernet link between cable modem and cable head office
  - Datagrams routed to ISP on Telco router network

Enterprise Access Networks

Leased line

Telco line to DCE on customer premises
2.048 Mbps to 40 Gbps
Carrier Ethernet — Ethernet extensions for metropolitan networks

Asynchronous Transfer Mode (ATM)

Telco system for broadband switched virtual circuits (SVC)
Optimized for multimedia transmission
Layer 2 ATM switch on customer premises
Telco line up to 2.5 Gbps

Frame Relay (FR)

Telco system for broadband permanent virtual circuits (PVC)
Layer 2 FR switch on customer premises
Telco line up to 45 Mbps

WiMax

Wireless metropolitan network
Applies cellular technology for 40 Mbps data

Internet Core

Internet backbone

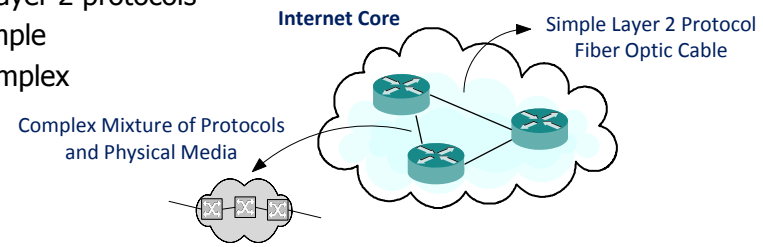
Collection of core routers and fast links

Core router

Fast router with very high I/O capacity
Up-to-date routing protocols
Handle multiple layer 1 and layer 2 protocols

Fast links

Various layer 2 protocols
Some simple
Some complex



Documentation

Standards

Formal documentation of systems, algorithms, protocols
Adopted by international committees
Record technical background and implementation requirements

Standards organizations

ISO	International Standards Organization Organization of governmental standards organizations
ITU-T	International Telecommunications Union - Telecommunications Sector United Nations standards organization (formerly CCITT)
ANSI	American National Standards Institute US government standards organization
IEEE	Institute of Electrical and Electronics Engineers
ACM	Association of Computing Machinery
IETF	Internet Engineering Task Force The Internet Society inherited Internet from US government in 1989 Internet standards called RFC (request for comment) Available at http://www.ietf.org/rfc.html