THE UNIVERSITY OF SYDNEY

Lecture 8

Application Layer

ELEC 3506/9506 Communication Networks

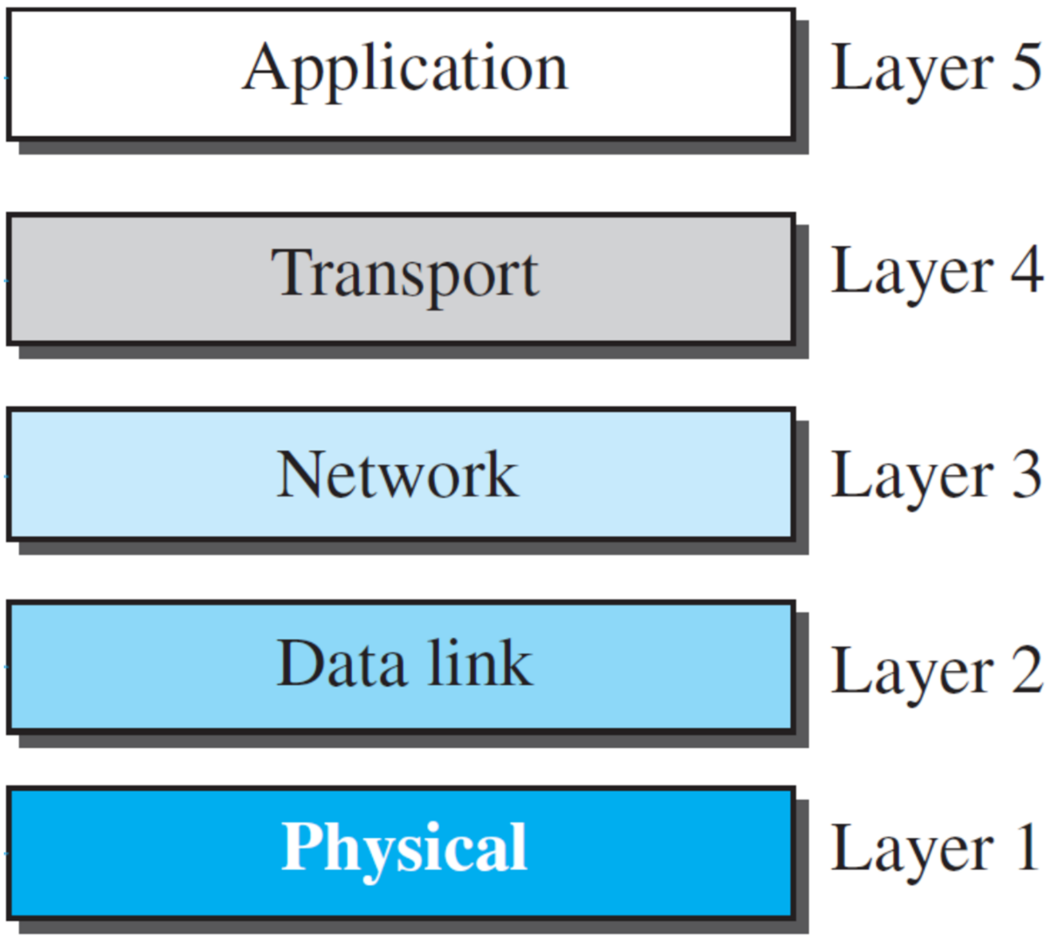
Dr Wibowo Hardjawana

School of Electrical and Information Engineering

Topics of the Day

- Application Layer Overview
- Hyper Text Transfer Protocol (HTTP)
- File Transfer Protocol (FTP)
- E-Mail: SMTP, POP, IMAP
- Domain Name System (DNS)
- Simple Network Management Protocol (SNMP)
- NETCONF and YANG

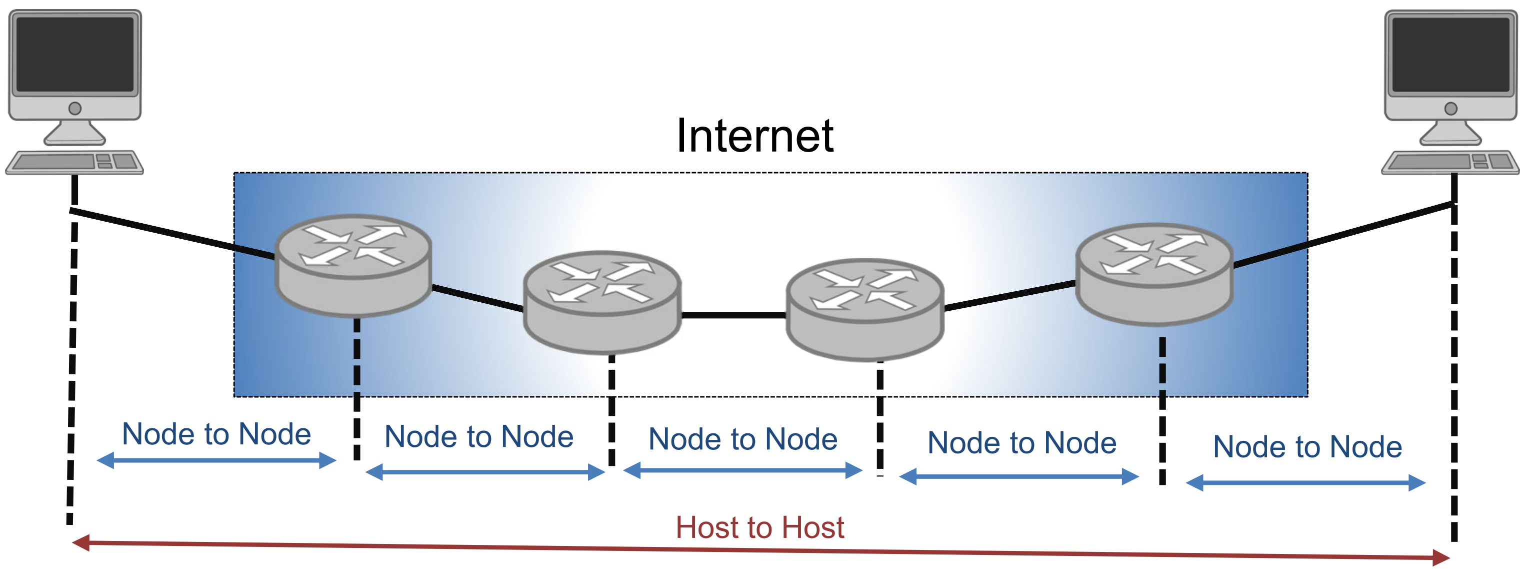
Layers in the TCP/IP Protocol Suite



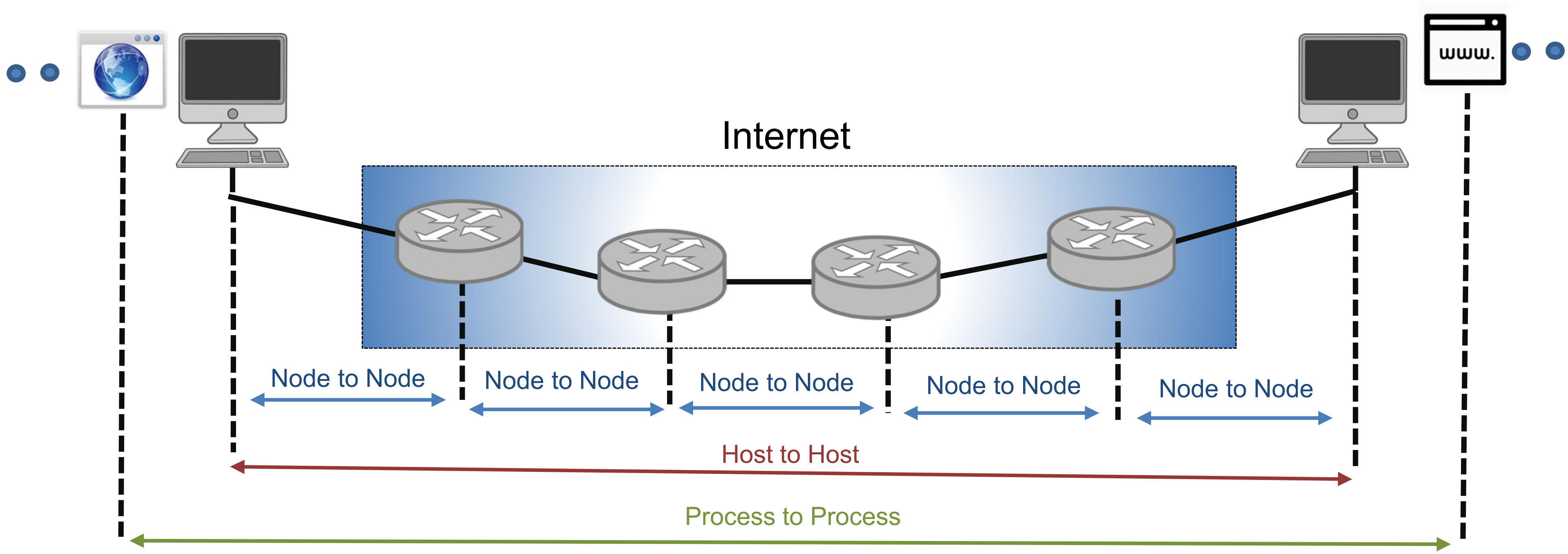
- **Application Layer:**
  - Providing services to Internet users
  - Receiving services from transport layer
  - The other four layers make the services at application layer possible

Transport Layer vs Other Layers

- Data Link Layer (Layer 2) – Node to Node Communication.
- Network Layer (Layer 3) – Host to Host Communication.

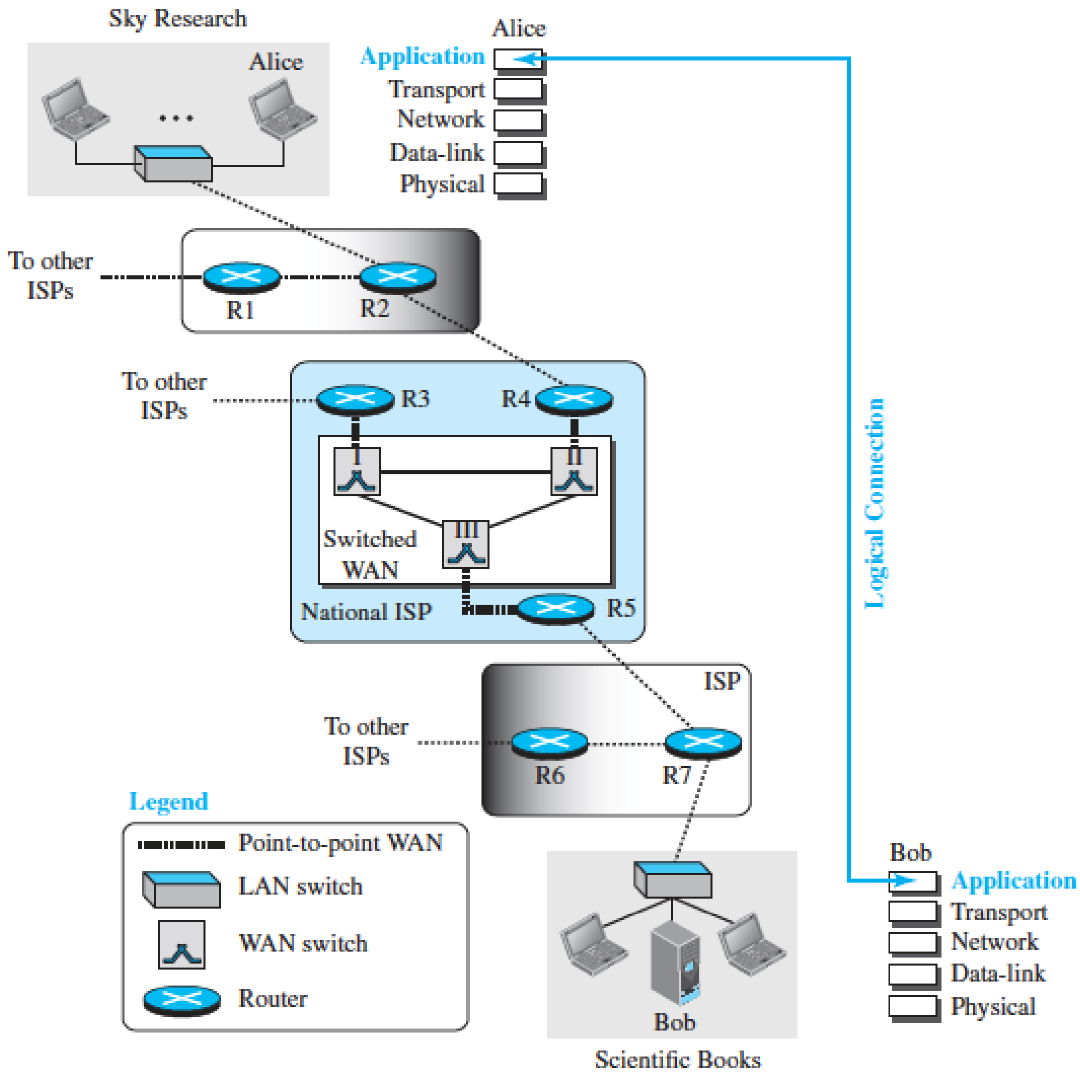


Transport Layer



- **Transport layer**: provides logical communication between applications/**processes** running on different hosts.

Logical Connection at Application Layer



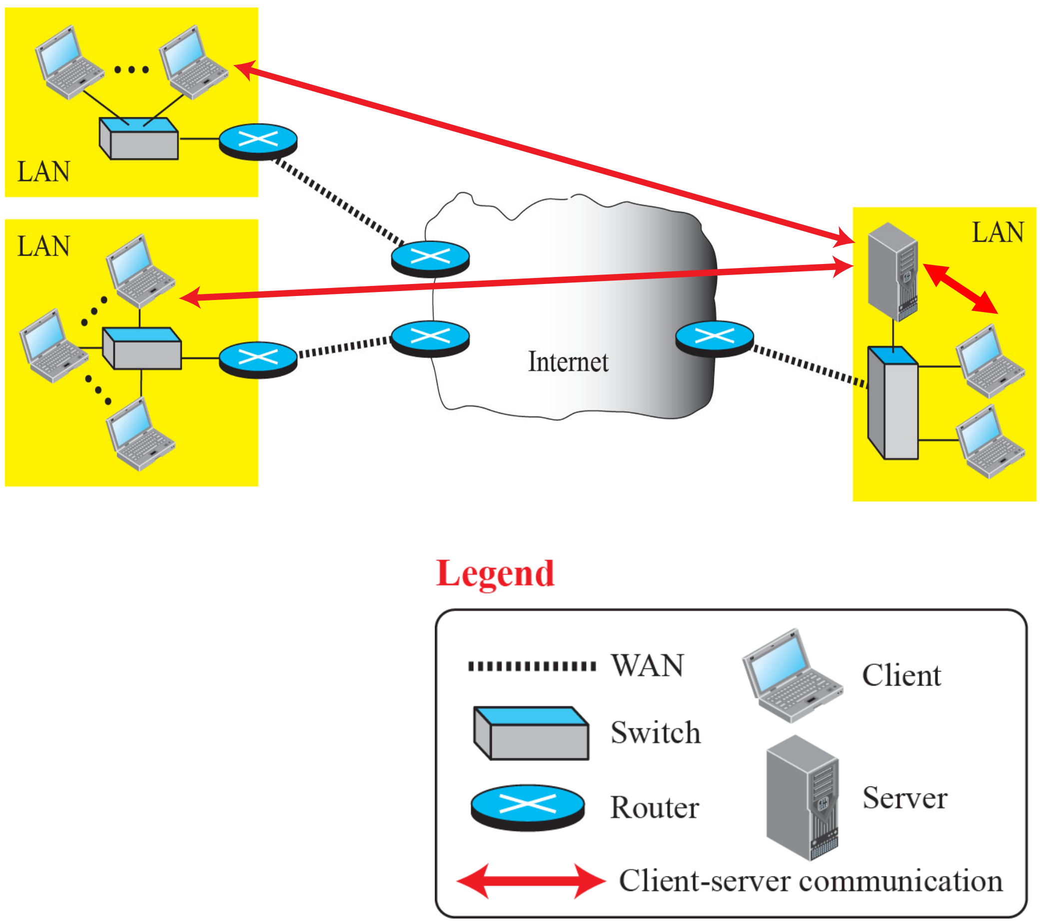
Add/Remove Application Layer Protocols

- The layered architecture makes the Internet flexible to add/remove/replace protocols in each layer
- If a protocol is added to each layer, it should be designed such that it uses the services provided by one of the protocols at the lower layer
- If a protocol is removed from a layer, care should be taken to change the protocol at the next higher layer that uses the services of the removed protocol
- Application layer protocols do not provide services to any other protocol in the suite. Thus, they can be removed easily
- New protocols can be added easily to application layer as long as they can use the services provided by the transport layer protocols

Application Layer Paradigms

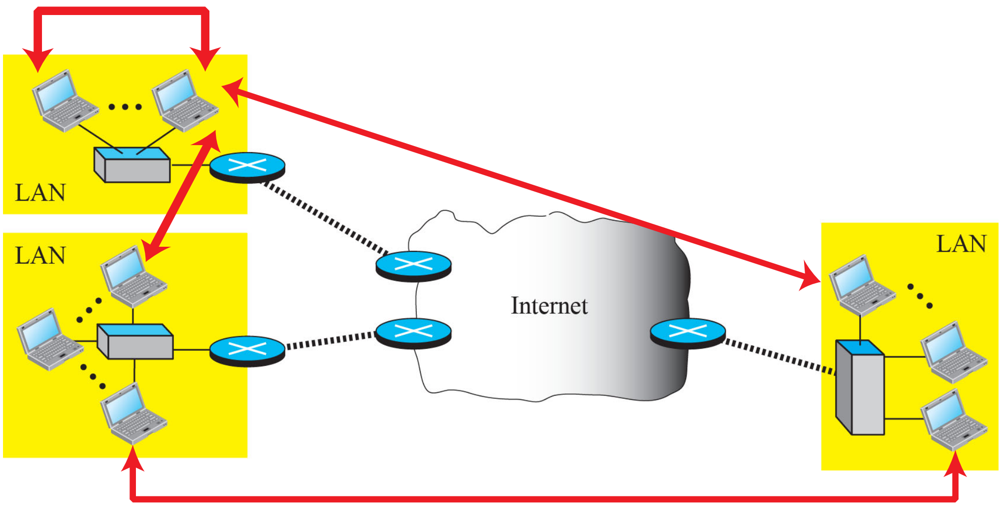
- Client-Server Paradigm (Traditional)
  - Service is provided by the server process, which must be running at all the time
  - **Client**: initiates contact with server, requests service from server
  - **Server**: provides requested service to client (router for Application Layer)
  - Examples: World Wide Web, file transfer protocol (FTP), email
- Peer-to-Peer (P2P) Paradigm (New)
  - No central server
  - Responsibility is shared between peers
  - A computer can both provide or receive services
  - Examples: Skype, BitTorrent, IPTV, Internet telephony

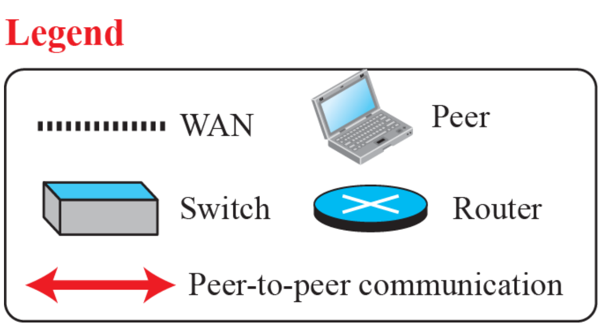
Client-Server Paradigm



- Requires powerful server due to concentration of load
- Server may breakdown
- Only suitable for services that can return some type of income

Peer-to-Peer Paradigm





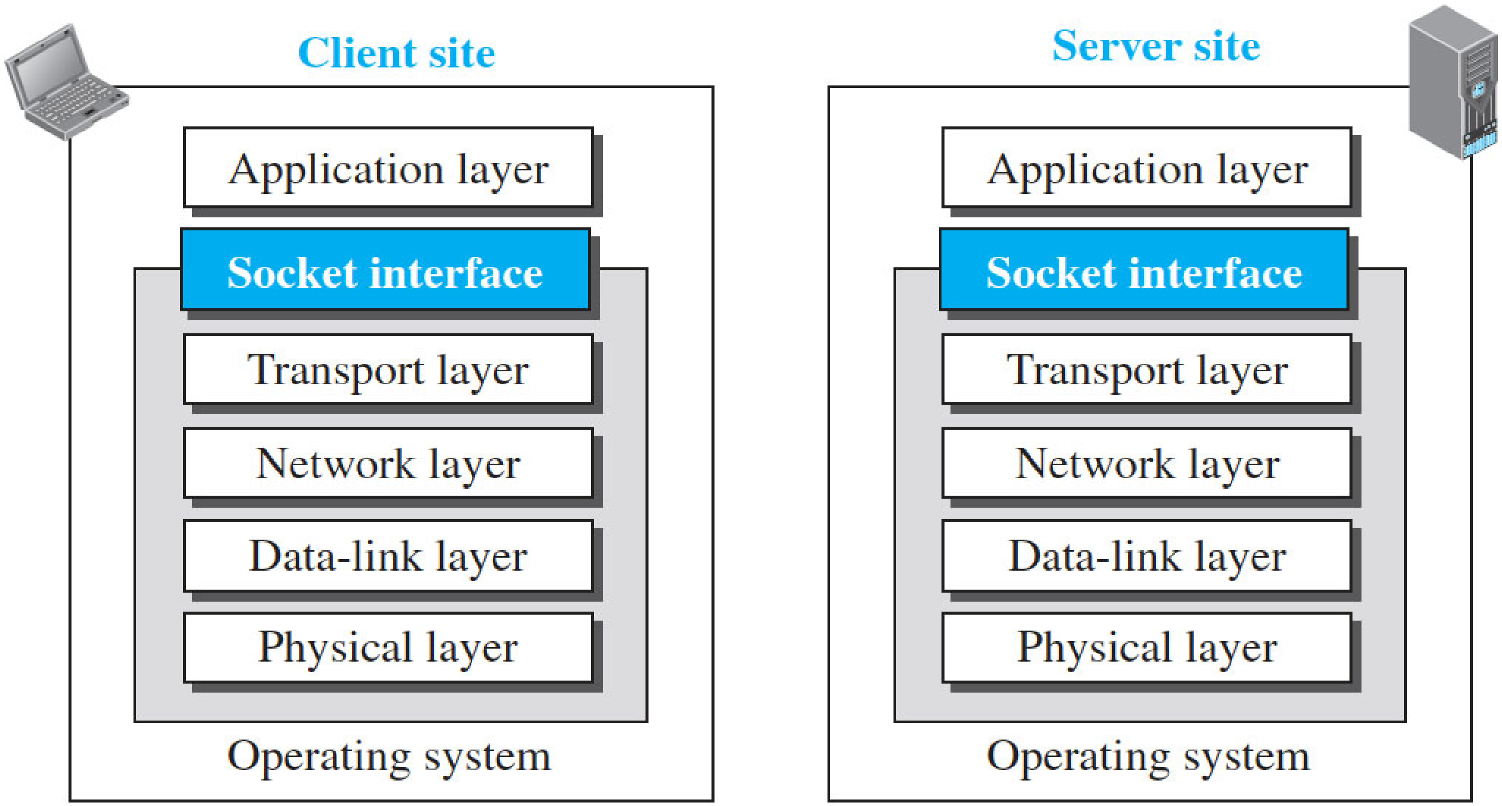
- Scalable & cost-effective
- Cons:
  - Security
  - Applicability

THE UNIVERSITY OF SYDNEY

Client-Server Programming

- We need a set of instructions to tell the lowest four layers of the TCP/IP suite to:
  - open the connection
  - send and receive data from the other end
  - close the connection
- Application Programming Interface (API) between:
  - the process at the application layer and
  - the operating system that encapsulates the first four layers of the TCP/IP protocol suite
- Socket interface: one API providing communication between the application layer and the operating system

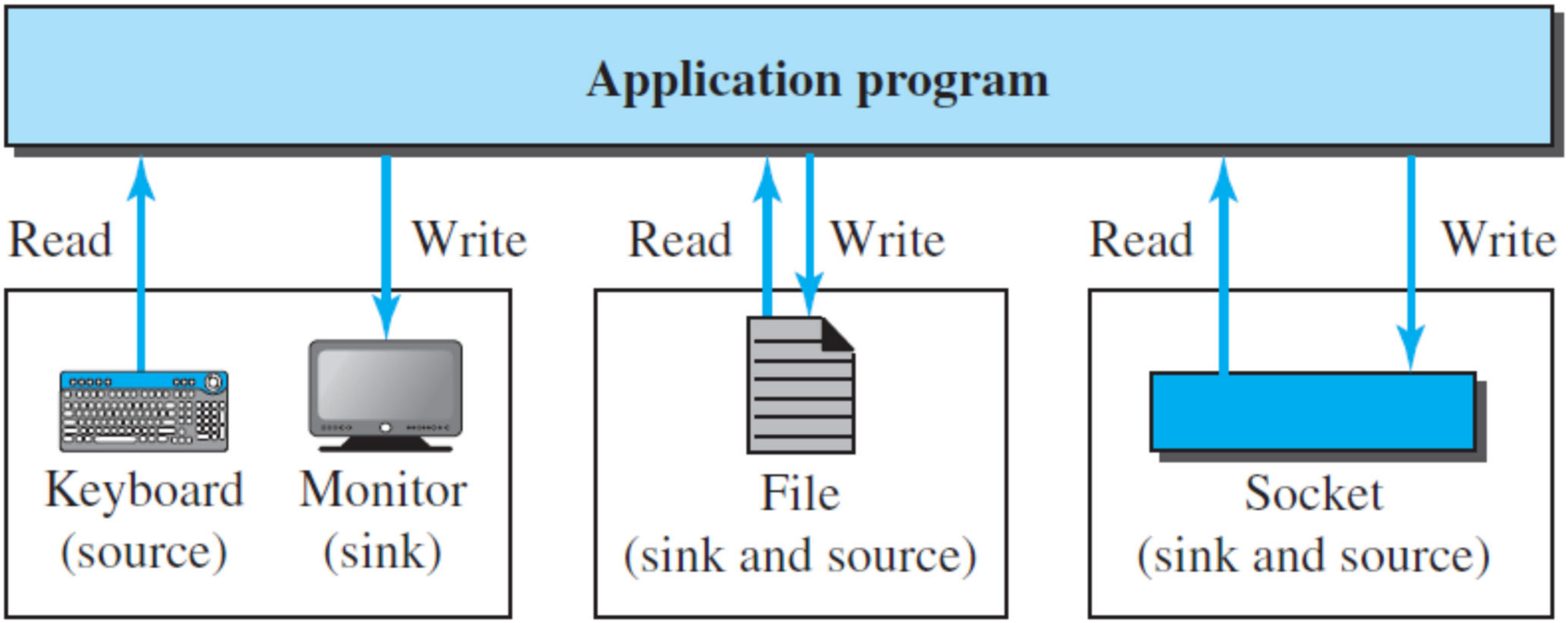
Socket Interface



- Socket is not a physical entity; it is an abstraction
- Socket is created and used by the application program

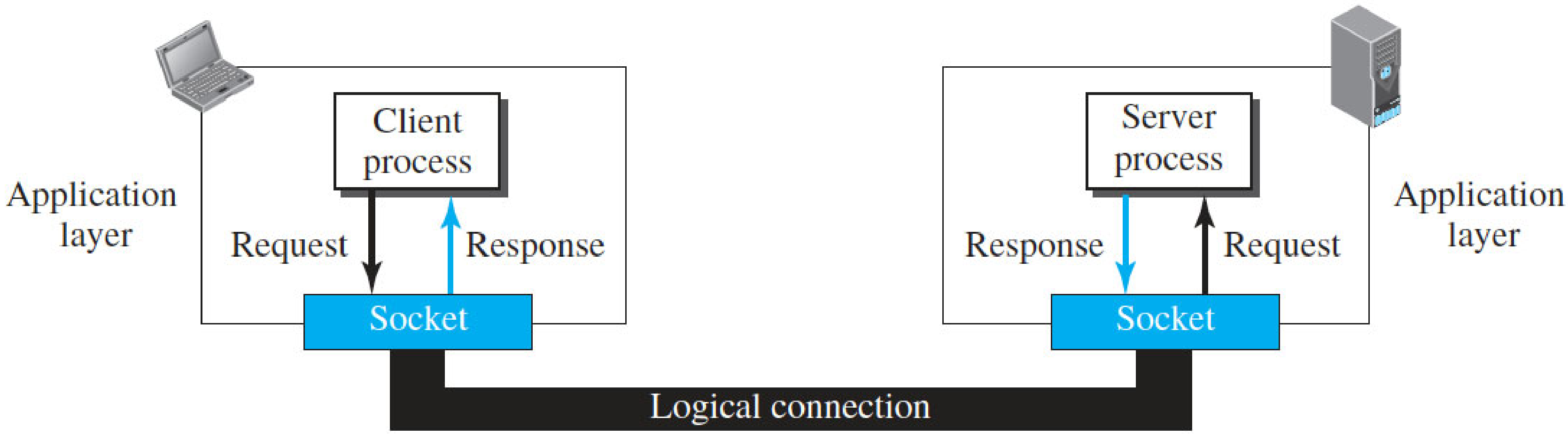
THE UNIVERSITY OF SYDNEY

Socket as Data Source/Sink



- Socket can be used the same way as other sources and sinks

Use Sockets in Process-to-Process Communication



- As far as the application layer is concerned, process-to-process communication is between two sockets
- The client thinks that the socket is the one that gives the response
- The server thinks that the socket is the one that sends the request
- A pair of socket addresses for communication

Using Services of the Transport Layer

- When writing a new application, can decide which transport layer protocol to use
- The choice of protocol seriously affects the capability of the application processes
  - TCP Protocol:
    - Connection-oriented
    - Reliable
    - Flow control
    - Error control
    - Congestion control
  - UDP Protocol:
    - Connectionless
    - Unreliable
    - Simple
    - Small delay
    - Low overhead

Internet Applications: Application and Transport Protocols Standards

Application	Application Protocol	Transport Protocol	IETF Standard
Email Transfer	SMTP	TCP	RFC 821 / 5321 / 6409
Email Delivery	POP3 / IMAP4	TCP	RFC 1939 / RFC 3501
Remote Access	Telnet	TCP	RFC854
	SSH	TCP	RFC4251
Web	HTTP1.1/2.0	TCP	RFC2068/7230/7235 RFC 7540
File Transfer	FTP	TCP	RFC959
	SFTP	TCP	Tunneled in SSH
Instant Messaging	XMPP	TCP	RFC6120 / 6121
VoIP	SIP	TCP/UDP	RFC3261
Video Streaming	RTSP	RTP/UDP	RFC2326 / 3550

Transport Service Requirements for Common Apps

Application	Data Loss	Bandwidth	Time Sensitive
Email	No Loss	Elastic	No
Remote Access	No Loss	Elastic	< 150 ms
Web	No Loss	Elastic	No
File Transfer	No Loss	Elastic	No
Instant Messaging	No Loss	Elastic	Yes & No
VOIP	Loss Tolerant	Audio: 5Kb - 1Mb Video: 1Kb - 5Mb	< 150 ms [ITU-T]
Video Streaming	Loss Tolerant	Elastic / 1 Mb – 10 Mb	Yes (Depend on the type of video)
Interactive Games	Loss Tolerant	>~1 Kbps	~ 100 ms
Financial Apps	No Loss	Elastic	Depends

Standard Client-Sever Protocols

- Hyper Text Transfer Protocol (HTTP)
- File Transfer Protocol (FTP)
- E-Mail: SMTP, POP, IMAP
- Domain Name System (DNS)
- Simple Network Management Protocol (SNMP)

WWW and HTTP

- World Wide Web (WWW): an information space where documents and other web resources are:
  - identified by Uniform Resource Locators (URLs)
  - interlinked by hypertext links, and
  - accessible via the Internet
- WWW today is a distributed client-server service
- **Web client** (browser): Internet Explorer, Firefox, Chrome, Safari
- **Web server**: where the web page is stored. Apache and Microsoft Internet Information Server
- HyperText Transfer Protocol (HTTP): Defines how the client-server programs can be written to retrieve web pages from the web

Uniform Resource Locator (URL)

- URL: The address of a resource on the Internet.
- It indicates the location of a resource and a protocol to access it.
- It usually contains four parts:
  - Protocol: the program needs to access the web page, e.g., http, ftp
  - Host: IP address of the server, or the domain name
  - Port: 16-bit port number, can be omitted if well-known
  - Path: the location and name of the file in the underlying operating system
- Different separators are used between the four parts

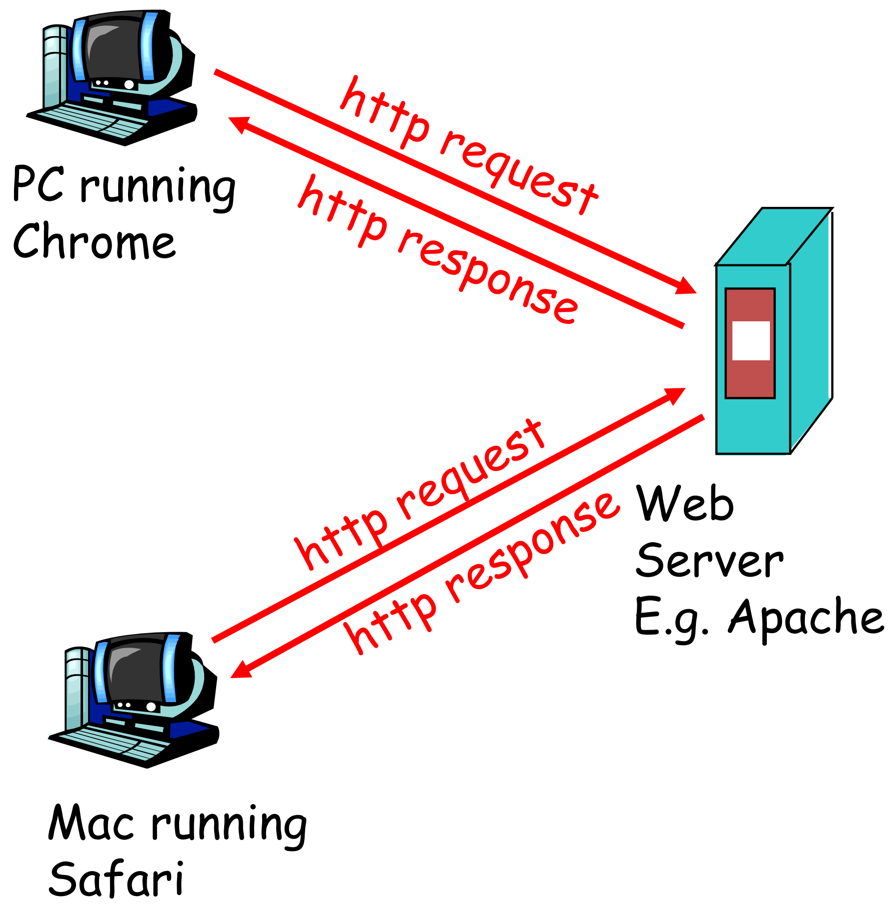
protocol://host/path	Used most of the time
protocol://host:port/path	Used when port number is needed

https://sydney.edu.au/engineering/about/school-of-electrical-and-information-engineering.html

The Web: HTTP protocol

HTTP: Hypertext Transfer Protocol

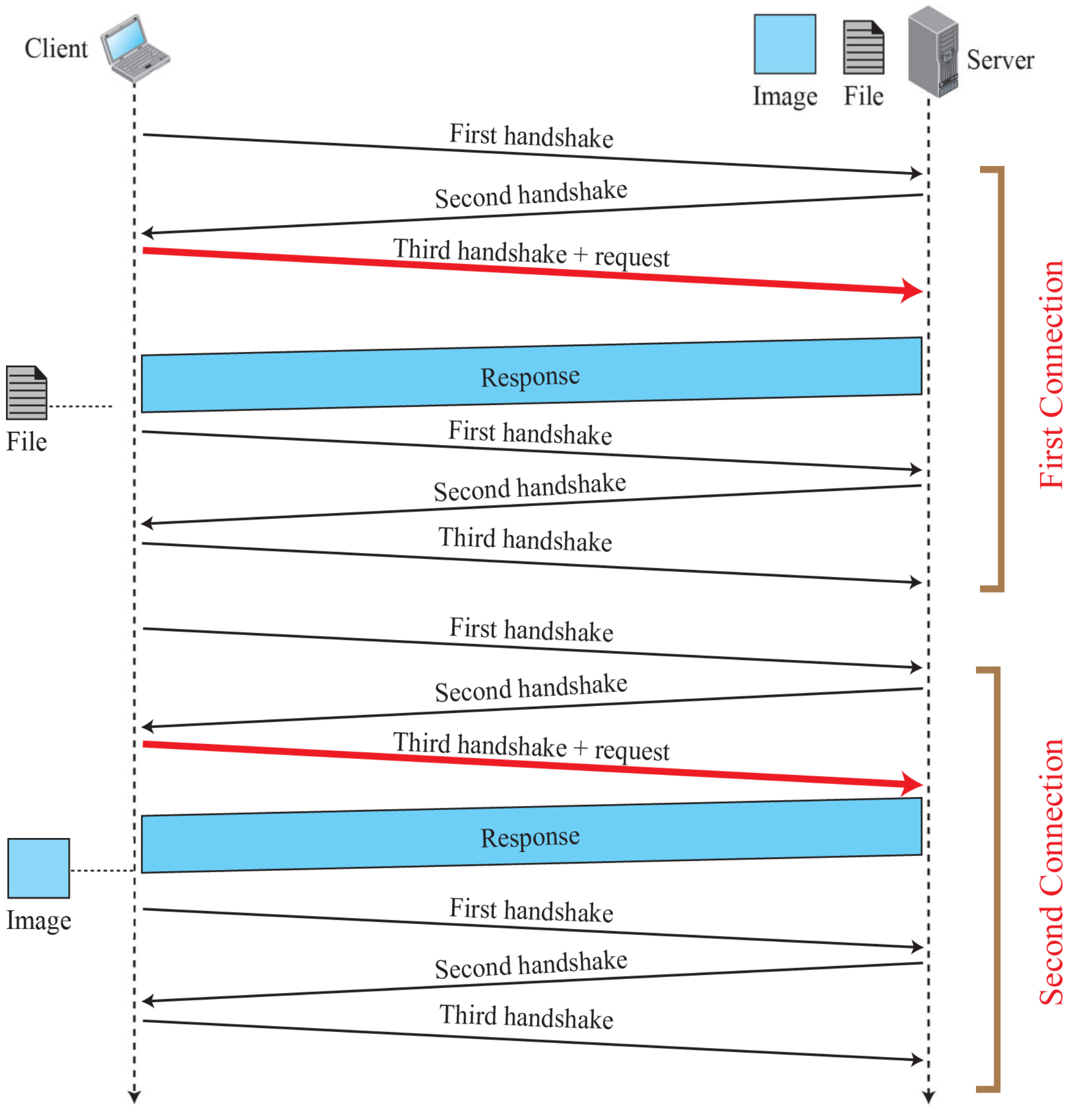
- Web's application layer protocol
- The server uses the port number 80
- Uses the services of TCP



Non-persistent vs Persistent Connections

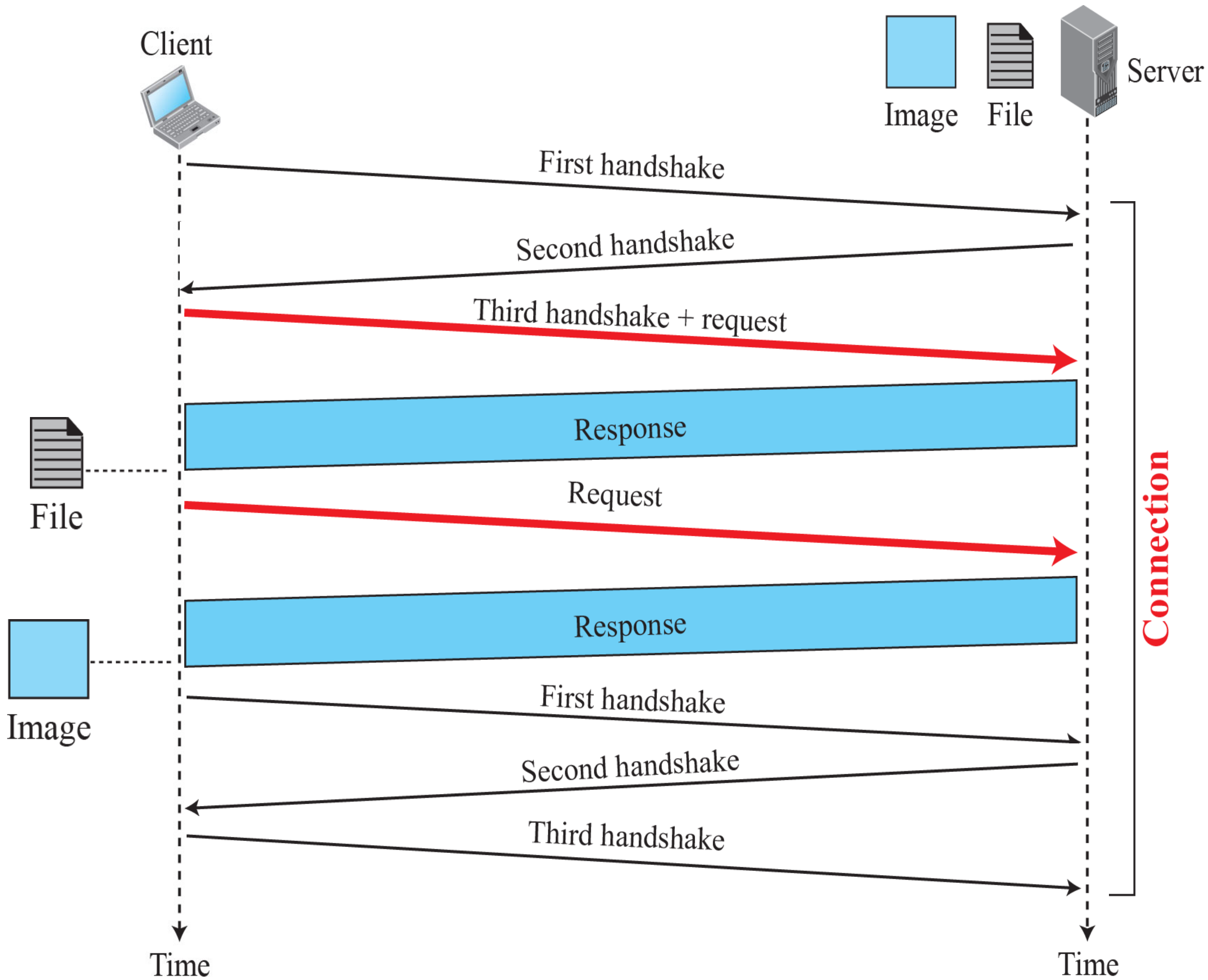
- Non-persistent connections
  - Retrieve each object (e.g., pictures etc) using a new TCP connection
  - One TCP connection is made for each request/response
  - Specified by HTTP prior to version 1.1
  - High overhead as different requests require different connections
- Persistent connections
  - Make a TCP connection to retrieve all objects
  - The server leaves the connection open for more request after sending a response
  - The server can close the connection at the request of a client or if a time-out has reached
  - HTTP version 1.1 specifies a persistent connection by default

HTTP example (Non-Persistent)



Connections are opened, closed, opened and closed

HTTP example (Persistent)

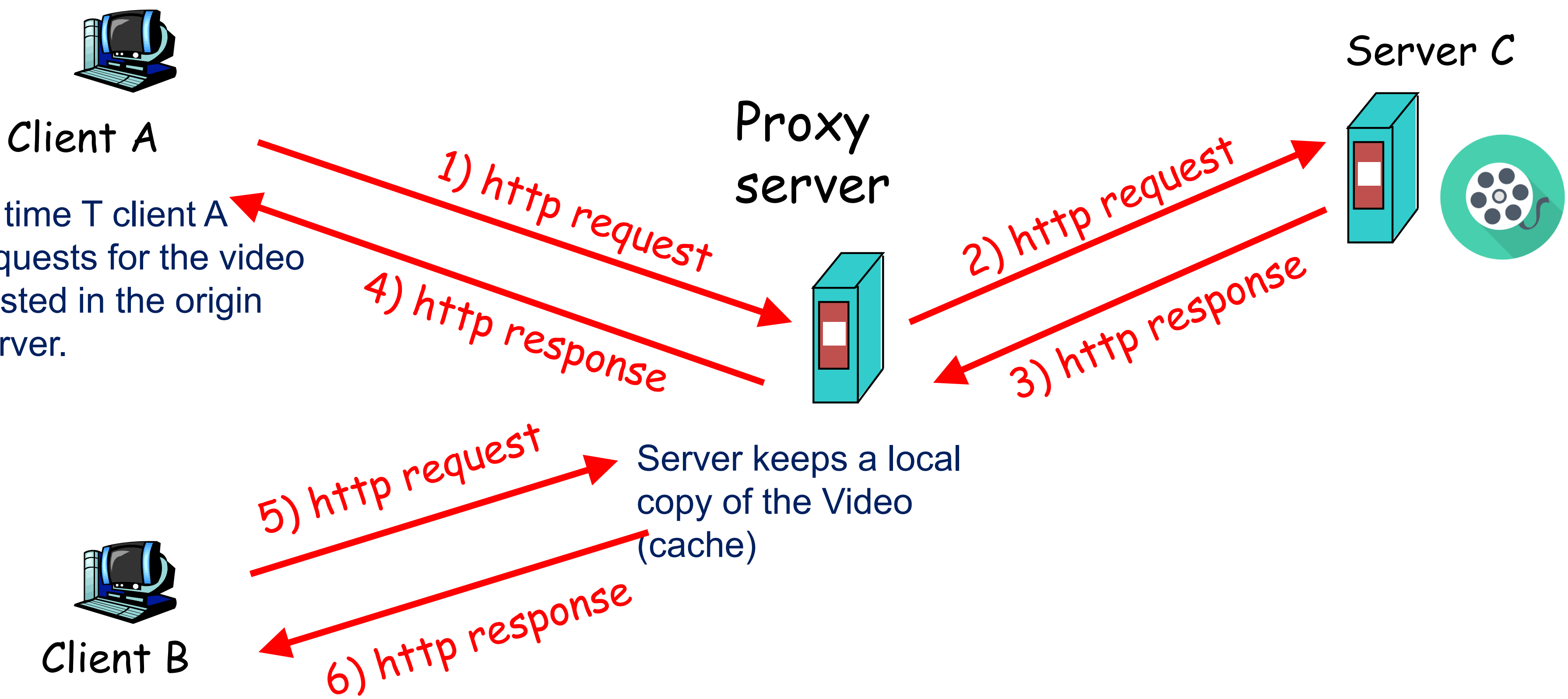


Connections are always open

Web Caching: Proxy Servers

- Proxy server: a computer that keeps copies of response to recent requests
- The HTTP client sends a request to proxy server
  - If object is in web cache: web cache returns object
  - Else: web cache requests object from origin server, then returns object to client
- Advantages:
  - Reduces the load on the original server
  - Decreases traffic
  - Improve latency

Proxy Server Operation



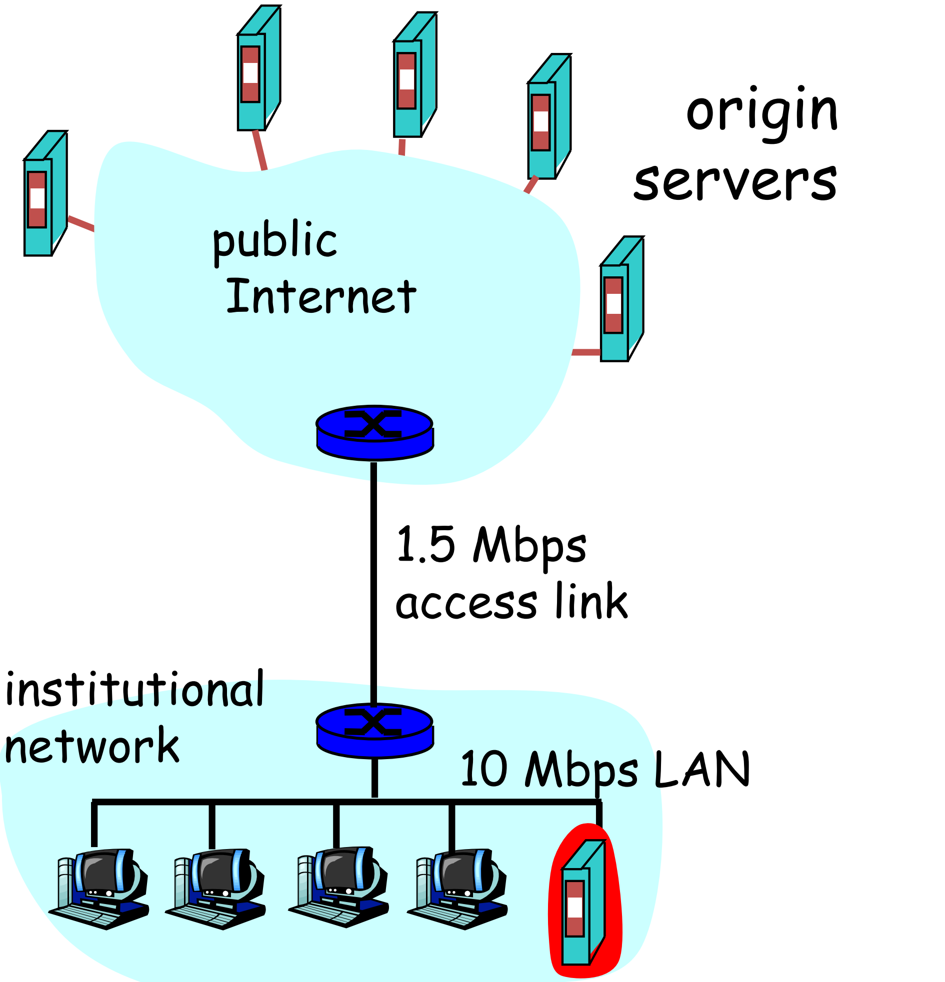
At time T client A requests for the video hosted in the origin server.

Server keeps a local copy of the Video (cache)

At time T+t client B request for the same video

Why Web Caching?

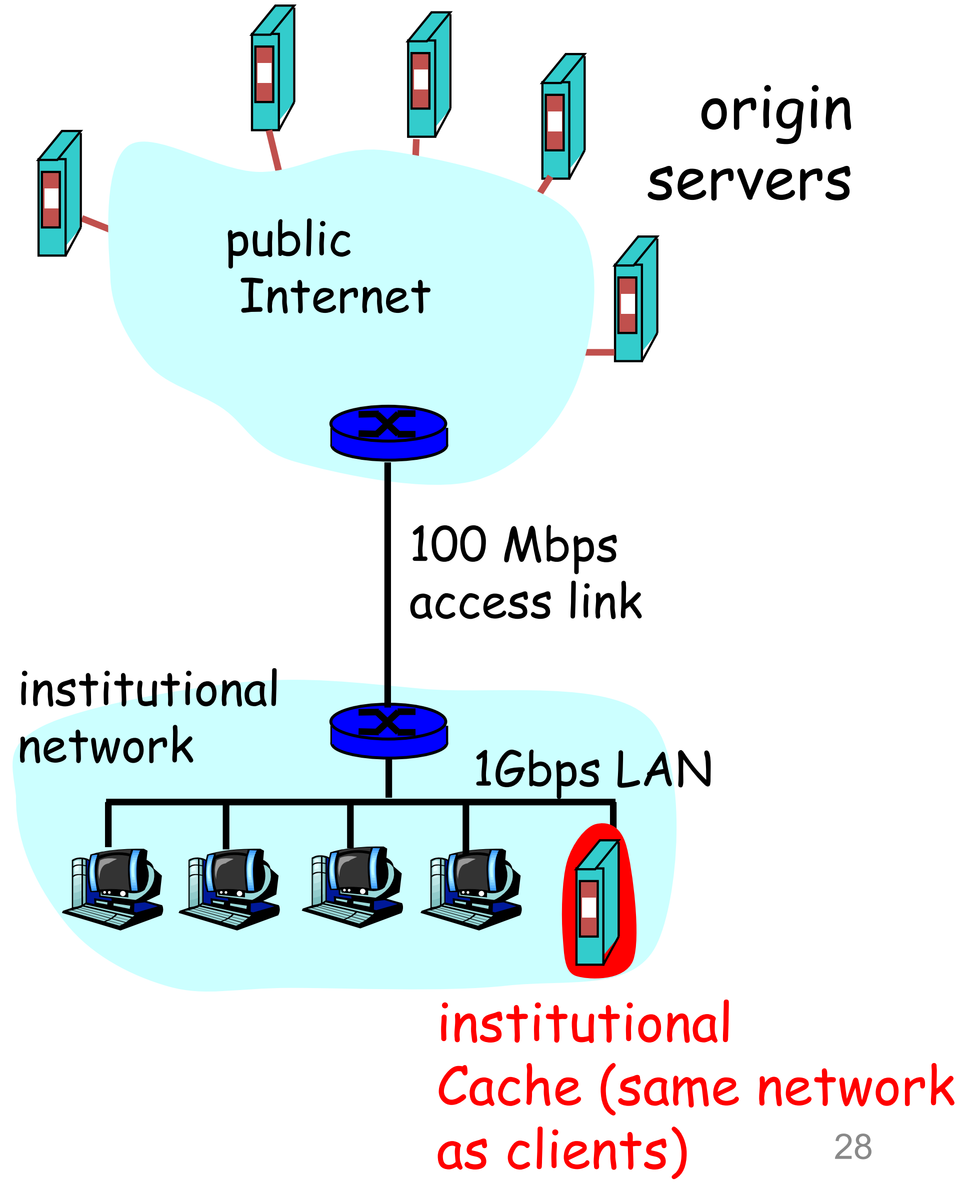
- Assume: cache is "close" to client (e.g., in same network)
- Smaller response time: cache "closer" to client
- Decrease traffic to distant servers
  - link out of institutional/local ISP network often bottleneck



institutional Cache (same network as clients)

Why Web Caching?

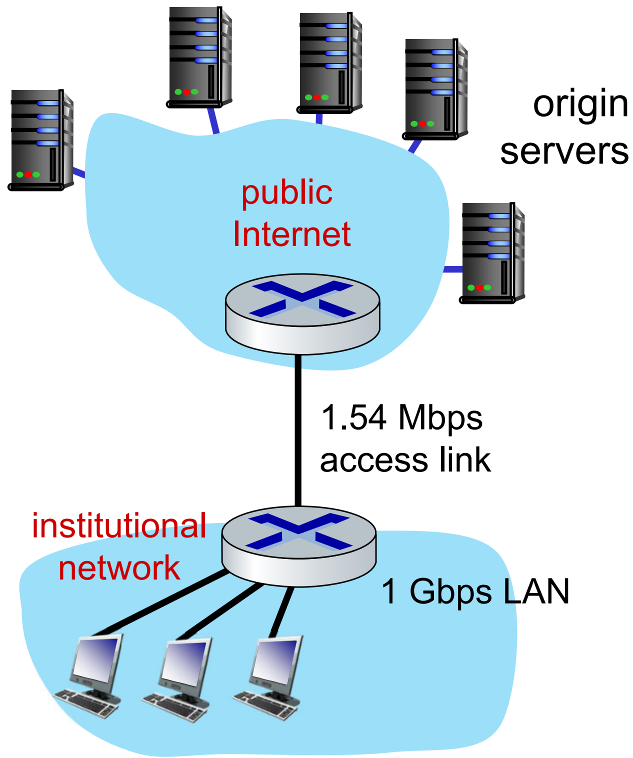
- Assume: cache is "close" to client (e.g., in same network)
- Smaller response time: cache "closer" to client
- Decrease traffic to distant servers
  - link out of institutional/local ISP network often bottleneck



Caching example

Scenario:

- access link rate: 1.54 Mbps
- RTT from public internet (PI) router to server: 2 sec
- Web object size: 100K bits
- Average request rate from browsers to origin servers: 15/sec
  - average data rate to browsers: 1.50 Mbps



Performance:

- Data rate=15 request/sec*100 kbits/request=1.5 Mbps
- LAN utilization: 1.5 Mbps /1Gbps=0.0015
- access link utilization = 1.5 Mbps/1.54 Mbps=0.97
- end-end delay = Internet delay + access link delay + LAN delay
  = 2 sec + ? + ??

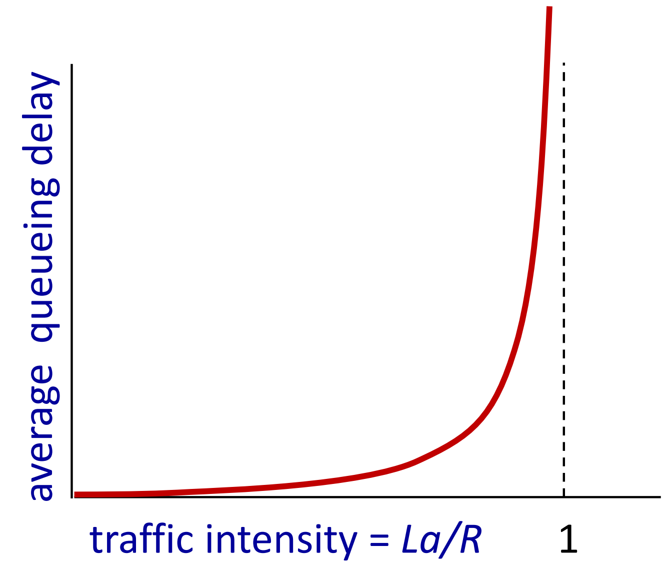
problem: large delays at high utilization!

- ? is in next slides

Packet queueing delay (M/M/1 model)

- *R:* link bandwidth (bps)
- *L:* packet length (bits)
- *a:* average packet arrival rate
- p: channel occupancy (e.g., busyness)

- *p=La/R* ~ 0: avg. queueing delay small
- *p=La/R* -> 1: avg. queueing delay large
- *p=La/R* > 1: more "work" arriving is more than can be serviced - average delay infinite!



- Prev slide a=15 request/sec, L=100 kbits/request and R=1.54 Mbps that give utilization denoted by p
- Effective transmission rate $R(1-p)$ bps
- Delay $=\frac{L}{R(1-p)} = \frac{0.065}{(1-p)} = 2.165\ sec$ for "?"
- When R=1Gbps p→0, delay ~0.0001 sec for "??"

La/R ~ 0

La/R -> 1

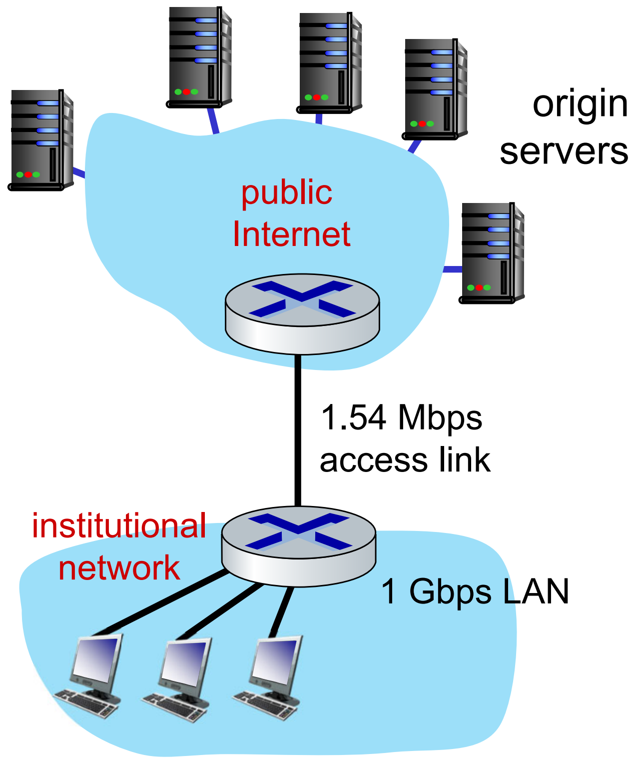
Caching example

Scenario:

- access link rate: 1.54 Mbps
- RTT from PI router to server: 2 sec
- Web object size: 100K bits
- Average request rate from browsers to origin servers: 15/sec
  - average data rate to browsers: 1.50 Mbps

Performance:

- Data rate=15 request/sec*100 kbits/request=1.5 Mbps
- LAN utilization: 1.5 Mbps /1Gbps=0.0015
- access link utilization = 1.5 Mbps/1.54 Mbps=0.97
- LAN delay=(100K bits/1Gbps)/(1-0.0015)=0.1msec
- end-end delay = Internet delay + access link delay + LAN delay
  = 2 sec + (2x2.165) sec + (2x0.0001) sec



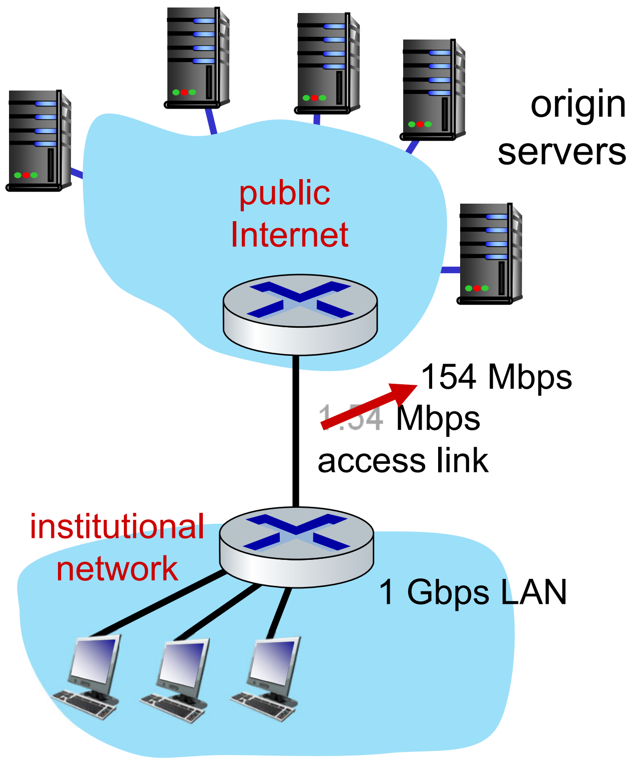
problem: large delays at high utilization!

- How about a faster link delay then?

Caching example: buy a faster access link

Scenario:

- access link rate: ~~1.54~~ 154 Mbps
- RTT from public internet router to server: 2 sec
- Web object size: 100K bits
- Avg request rate from browsers to origin servers: 15/sec
  - avg data rate to browsers: 1.50 Mbps



Performance:

- LAN utilization: 0.0015
- access link utilization = ~~.97~~ → 0.0097
- end-end delay = Internet delay + access link delay + LAN delay

  = 2 sec + ~~minutes~~ + 2x0.0001 sec

Reduction to msec delay but how much ?

Cost: faster access link (expensive!)

Caching example: install a web cache

Scenario:

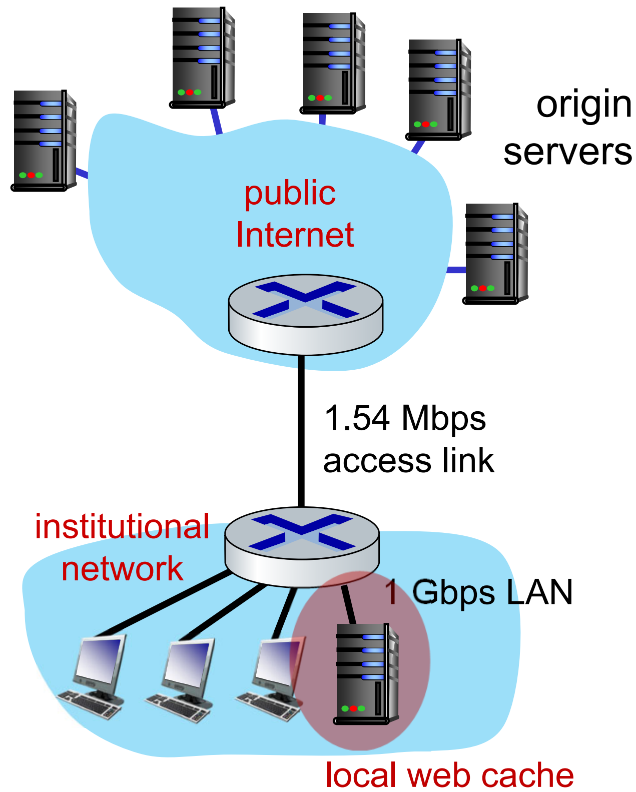
- access link rate: R=1.54 Mbps
- RTT from PI router to server: 2 sec
- Web object size: L=100K bits
- Avg request rate from browsers to origin servers: 15/sec
  - avg data rate to browsers: 1.50 Mbps

Performance:

- LAN utilization: .?
- access link utilization = ?
- average end-end delay = ?

How to compute link utilization, delay?

Cost: web cache (cheap!)

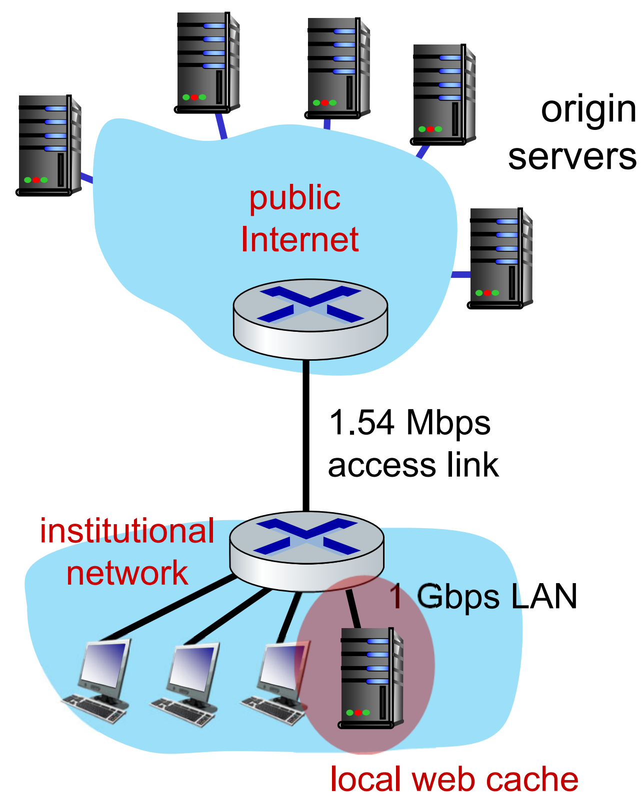


Caching example: install a web cache

Calculating access link utilization, end-end delay with cache:

- suppose cache hit rate is 0.4: 40% requests satisfied at the cache, 60% requests satisfied at the origin

- access link: 60% of requests use access link
- data rate to browsers over access link = 0.6 * 1.50 Mbps = .9 Mbps
- utilization = 0.9/1.54 = 0.58 $(2\text{x}\frac{0.065}{1-0.58})$ sec delay << RTT of 2 sec)
- average end-end delay
  = 0.6 * (delay from origin servers)
  + 0.4 * (delay when satisfied at cache)
  = 0.6 (RTT of 2 sec + 0.32 sec) + 0.4 (~ 2x0.0001 sec) = ~ 1.392 sec



lower average end-end delay than with 154 Mbps link (and cheaper too!)

Knowledge Test

Join at menti.com | use code 5606 3235

Consider a network with a utilisation of 1. Can the packets be transmitted successfully?

0 ×

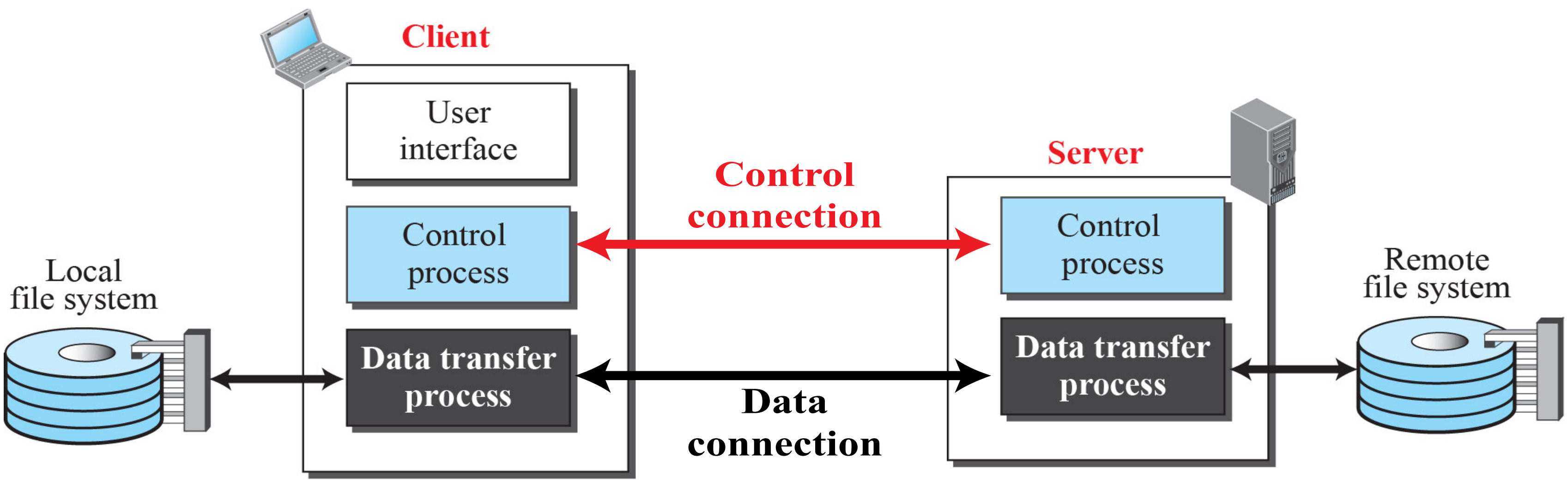
Yes

0 ✓

No

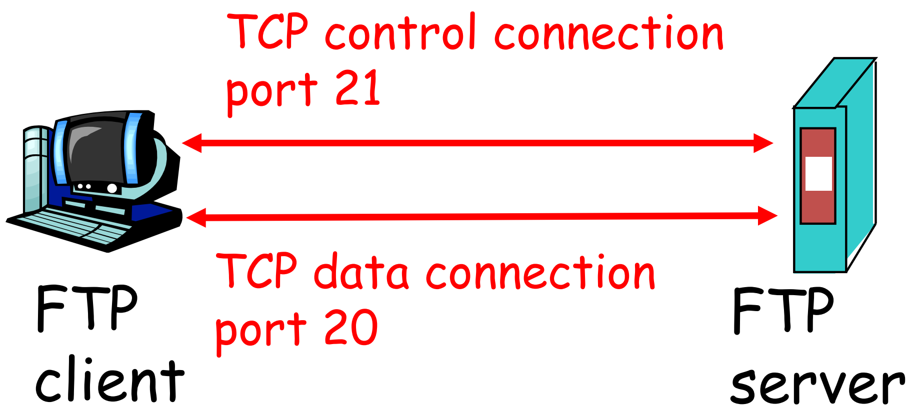
FTP: File Transfer Protocol

- FTP: standard protocol for transfer files from one host to another
- Client has three components:
  - User interface
  - Client control process
  - Data transfer process
- Server has two components:
  - Server control process
  - Server data transfer process



Separate Control And Data Connections

- Separation of commands and data transfer makes FTP more efficient
- Control connection: port 21
  - Uses very simple rules of communication
  - Needs to transfer only a line of command or a line of response at a time
  - Remains connected during the entire FTP session
- Data connection: port 20
  - Needs more complex rules due to the variety of data types transferred
  - Opened and then closed for each file transfer activity



FTP Commands, Responses

Sample FTP commands:

- Sent as ASCII text over control channel
  - USER *UserID*
  - PASS *password*
  - LIST: list subdirectories or files
  - RETR: *filename;* retrieve files from server
  - STOR: *filename;* store files to server

Sample FTP responses

- Status code and phrase (as in http)
  - 331 Username OK, password required
  - 125 data connection already open; transfer starting
  - 425 Can’t open data connection
  - 452 Error writing file

FTP File Transfer Example

- Connecting to a public FTP server using a FTP client.

- Most of the OSs come with a FTP client for the terminal (accessible via ftp command) or there are GUI based clients such as FileZilla.

```
Last login: Sun Sep 11 12:45:53 on ttys002
jolt-ev:~ sen040$ ftp ftp://speedtest.tele2.net
Trying 90.130.70.73...
Connected to speedtest.tele2.net.
220 (vsFTPd 2.3.5)
331 Please specify the password.
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
200 Switching to Binary mode.
ftp> ls
229 Entering Extended Passive Mode (|||28580|).
150 Here comes the directory listing.
-rw-r--r--    1 0        0        1073741824000 Feb 19  2016 1000GB.zip
-rw-r--r--    1 0        0        107374182400 Feb 19  2016 100GB.zip
-rw-r--r--    1 0        0          102400 Feb 19  2016 100KB.zip
-rw-r--r--    1 0        0        104857600 Feb 19  2016 100MB.zip
-rw-r--r--    1 0        0        10737418240 Feb 19  2016 10GB.zip
-rw-r--r--    1 0        0        10485760 Feb 19  2016 10MB.zip
-rw-r--r--    1 0        0        1073741824 Feb 19  2016 1GB.zip
-rw-r--r--    1 0        0            1024 Feb 19  2016 1KB.zip
-rw-r--r--    1 0        0         1048576 Feb 19  2016 1MB.zip
-rw-r--r--    1 0        0        209715200 Feb 19  2016 200MB.zip
-rw-r--r--    1 0        0        20971520 Feb 19  2016 20MB.zip
-rw-r--r--    1 0        0         2097152 Feb 19  2016 2MB.zip
-rw-r--r--    1 0        0         3145728 Feb 19  2016 3MB.zip
-rw-r--r--    1 0        0        524288000 Feb 19  2016 500MB.zip
-rw-r--r--    1 0        0        52428800 Feb 19  2016 50MB.zip
-rw-r--r--    1 0        0          524288 Feb 19  2016 512KB.zip
-rw-r--r--    1 0        0         5242880 Feb 19  2016 5MB.zip
drwxr-xr-x    2 105      108          4096 Sep 11 04:17 upload
226 Directory send OK.
ftp> get 5MB.zip
local: 5MB.zip remote: 5MB.zip
229 Entering Extended Passive Mode (|||23788|).
150 Opening BINARY mode data connection for 5MB.zip (5242880 bytes).
100% |***********************************|  5120 KiB   57.01 KiB/s    00:00 ETA
226 Transfer complete.
5242880 bytes received in 01:30 (56.69 KiB/s)
```

Connecting to the server

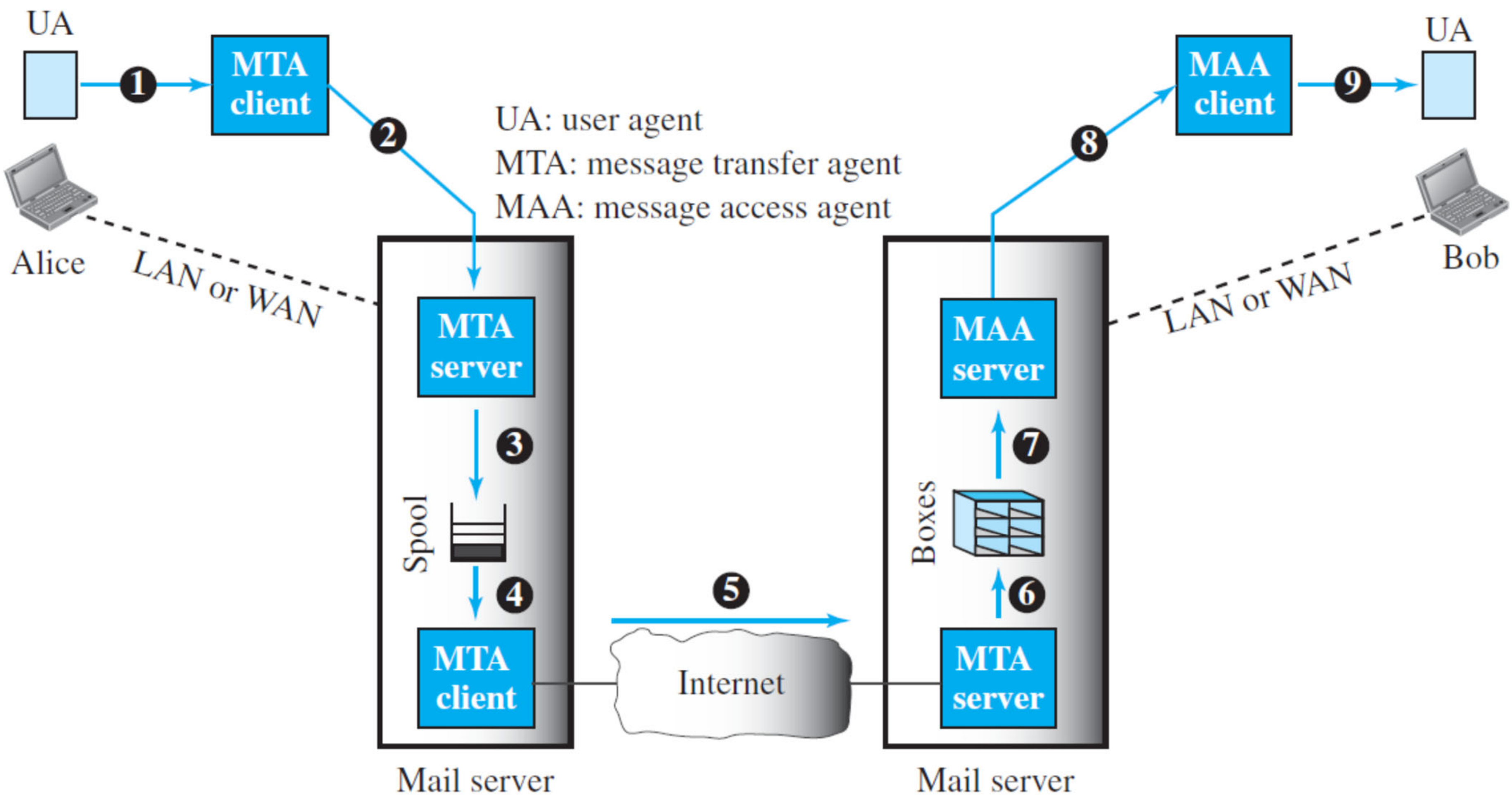
Listing the files

Accessing one file

Electronic Mail

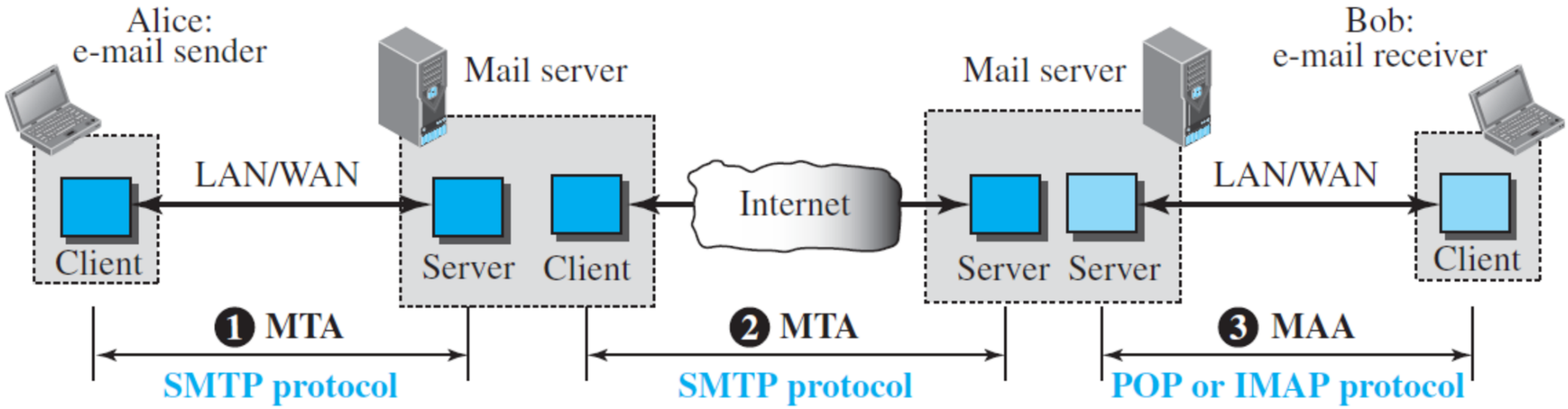
- Unique characteristics of E-mail application (compared with HTTP, FTP)
  - No immediate response is required
  - Infeasible for email recipients to run a server program and wait until someone sends an e-mail
  - The idea of client/server programming should be implemented using some intermediate servers
- Components of E-mail system
  - **User Agent** (UA): Software that composes, reads, replies to and forwards messages
  - **Message transfer agent** (MTA): Software that transfers messages from one computer to another
  - **Message access agent** (MAA): Software to access messages
  - Require two UAs, two pairs of MTAs (client and server), and a pair of MAAs (client and server)

Electronic Mail System Architecture



THE UNIVERSITY OF SYDNEY

Protocols Used in Electronic Mail



- E-mail application needs three uses of client-server paradigms
- MTA protocol: **Simple Mail Transfer Protocol (SMTP)**
  - Push protocol: message is pushed from client to server
  - SMTP is used two times
- MAA protocol: Requires pull protocols
  - Message is pulled by client from server
  - **Post Office Protocol, version 3 (POP3)** and **Internet Mail Access Protocol, version 4 (IMAP4)**

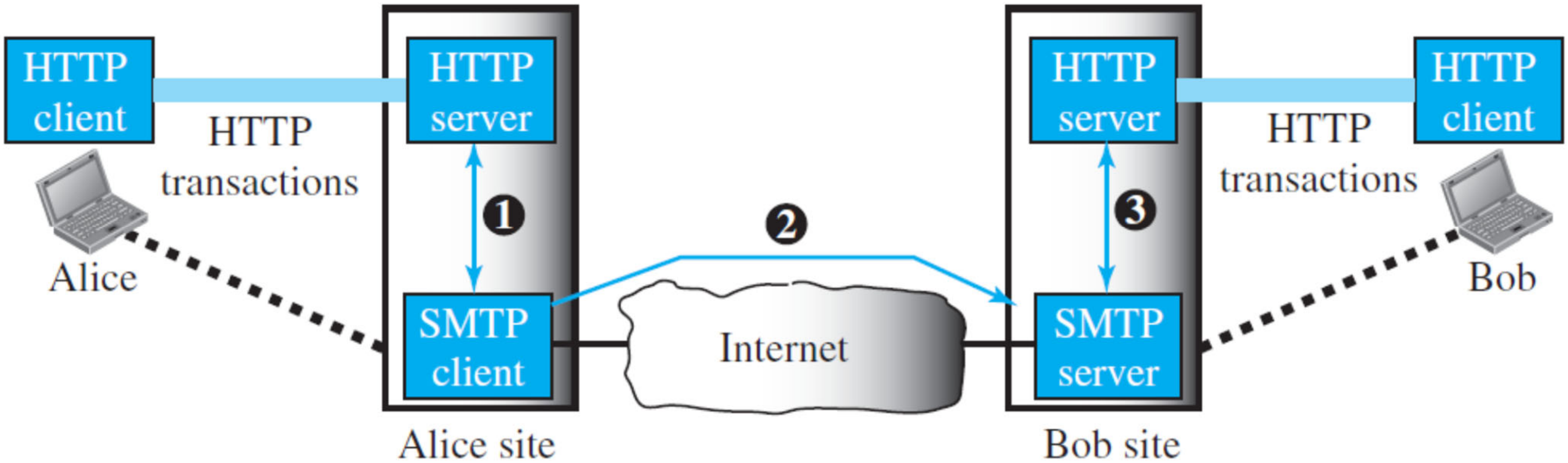
Message Transfer Agent: SMTP

- SMTP defines how commands and responses must be sent between MTA client and server
- Uses TCP connection with the well-known port 25
- Three phases of transfer
  - Connection establishment (Handshaking)
  - Transfer of messages
  - Connection termination
- Command/response interaction
  - Commands (from client to server): ASCII text
  - Response (from server to client): status code and phrase
- Messages must be in 7-bit ASCII

Message Access Agent: POP and IMAP

- Post Office Protocol, version 3 (POP3)
  - Simple but limited functionality
  - User cannot organize mails on the server
  - User cannot have different folders on the server
  - User cannot partially check the contents of the mail before downloading
- Internet Mail Access Protocol, version 4 (IMAP4)
  - More powerful and more complex with extra functions
  - User can check e-mail header prior to downloading
  - User can partially download e-mail
  - User can create, delete, or rename mailboxes on the mail server
  - User can create a hierarchy of mailboxes in a folder for e-mail storage

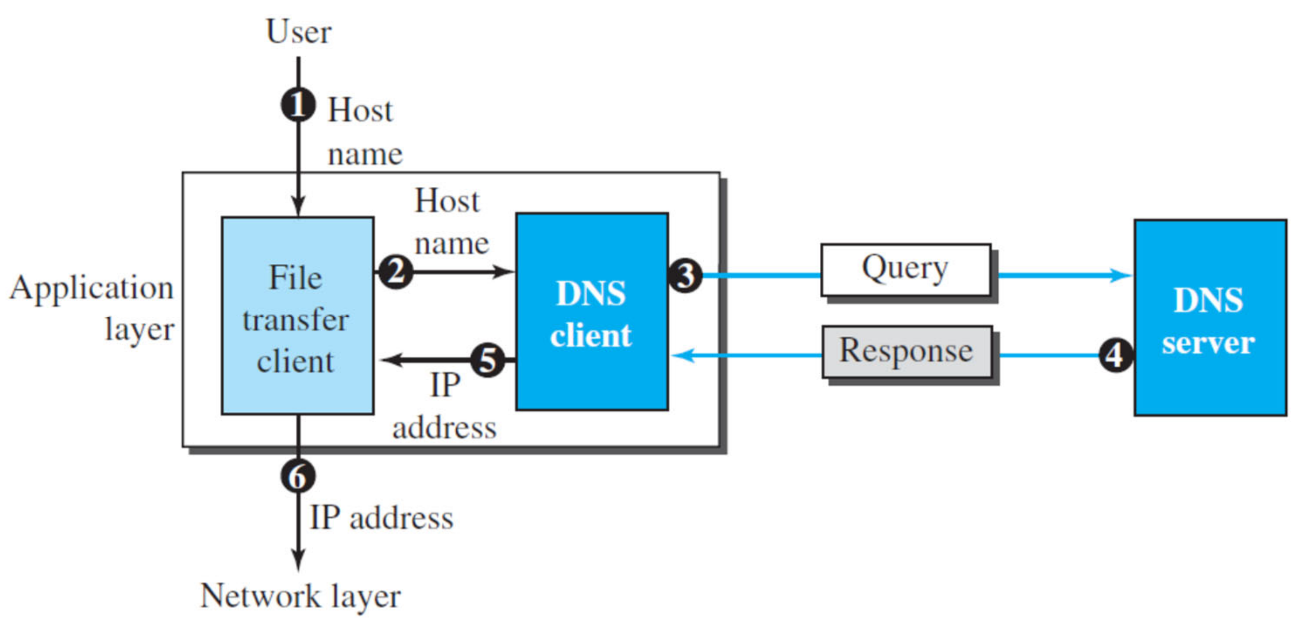
Web-Based Mail



- Alice sends the message to the web server using HTTP transactions
- Bob receives message using HTTP transactions
- However, the message from the server of Alice to the server of Bob still uses SMTP protocol

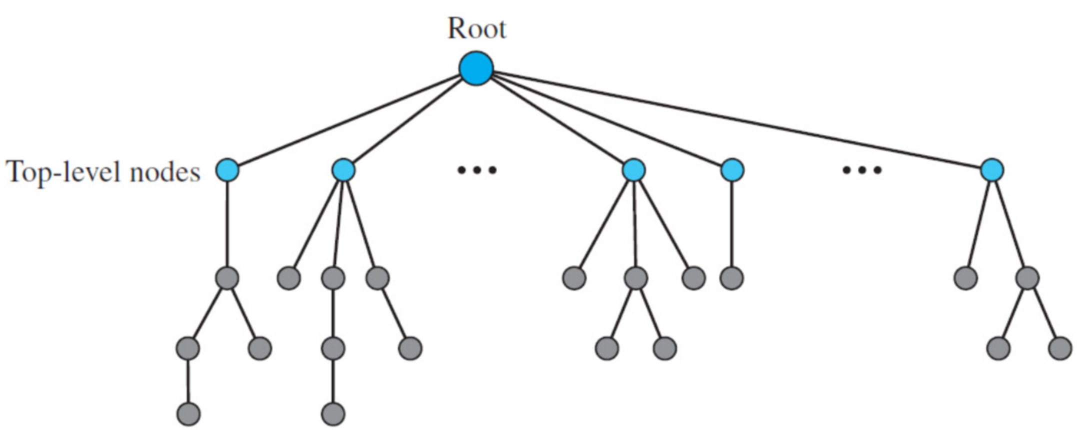
DNS: Domain Name System

- TCP/IP protocols use IP address to uniquely identify the connection of a host
- People prefer to use names, e.g., yahoo.com
- How to map a name to an address?
- Domain Name System (DNS):
  - An application-layer protocol allows host and name servers to communicate to resolve address/name translation

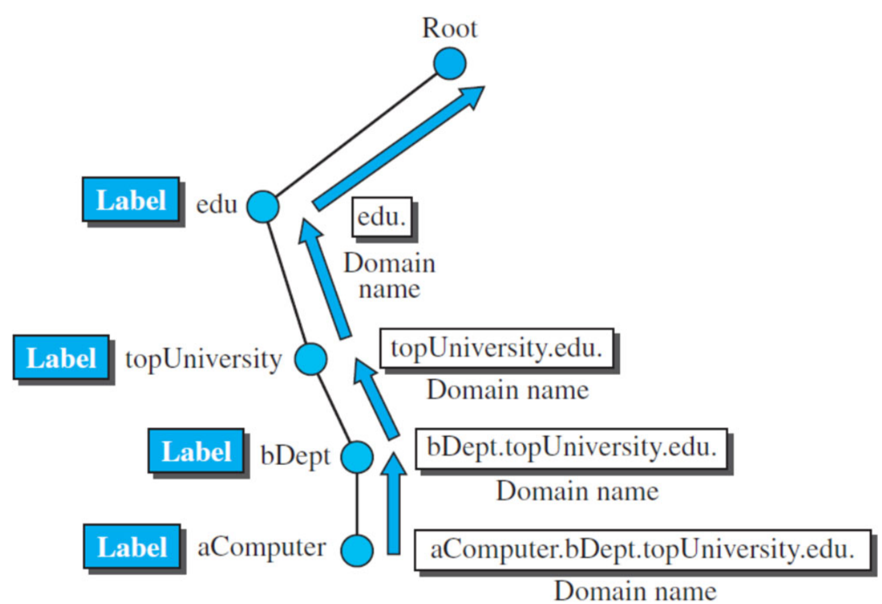


Domain Name Space

- **Hierarchical** domain name space: each name is made of several parts
- The names are defined in an inverted-tree structure with the root at the top
- Each node has a label and domain name
  - Label: a string with a maximum of 63 characters for each node
  - Domain name: a sequence of labels separated by dots, read from the node up to the root



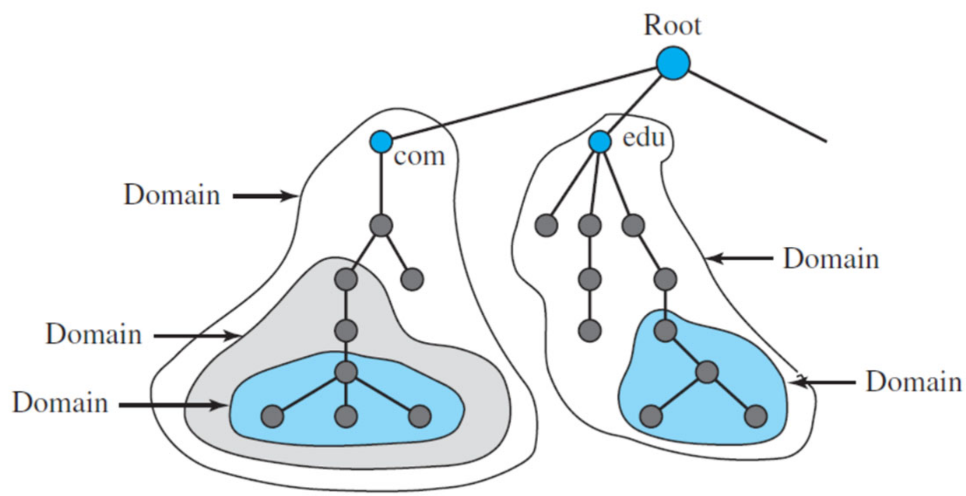
Domain name space



Domain names and labels

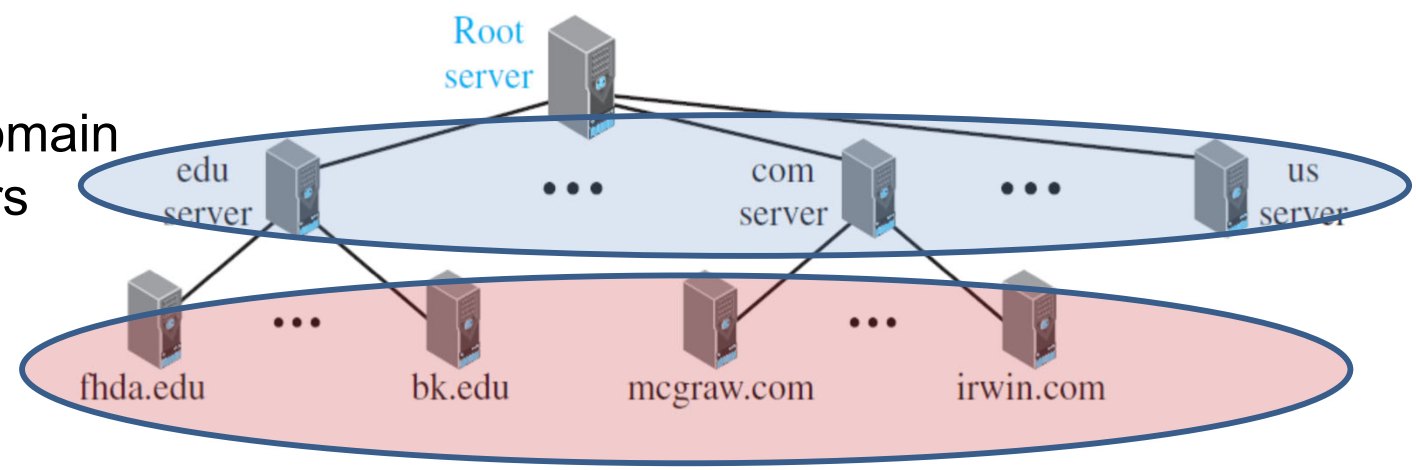
Domain and Hierarchy of Name Servers

- Domain: a subtree of the domain name space
- The information contained in the domain name space is distributed among many computers called DNS servers
- Hierarchy of servers in the same way as hierarchy of names



Top Level Domain (TLD) Servers

Local Servers



DNS Name Servers

Why not centralize DNS?

- Single point of failure
- Traffic volume
- Distant centralized database
- Maintenance
- Non-scalable

- No server has all name-to-IP address mappings
- Local name servers:
  - Each ISP, Organization has *local (default) name server*
  - Host DNS query first goes to local name server
- Authoritative name server:
  - Configured by an administrator with the hostname information for a particular domain
  - Information about these servers is added to the root servers when the domain is registered

Name-Address Resolution

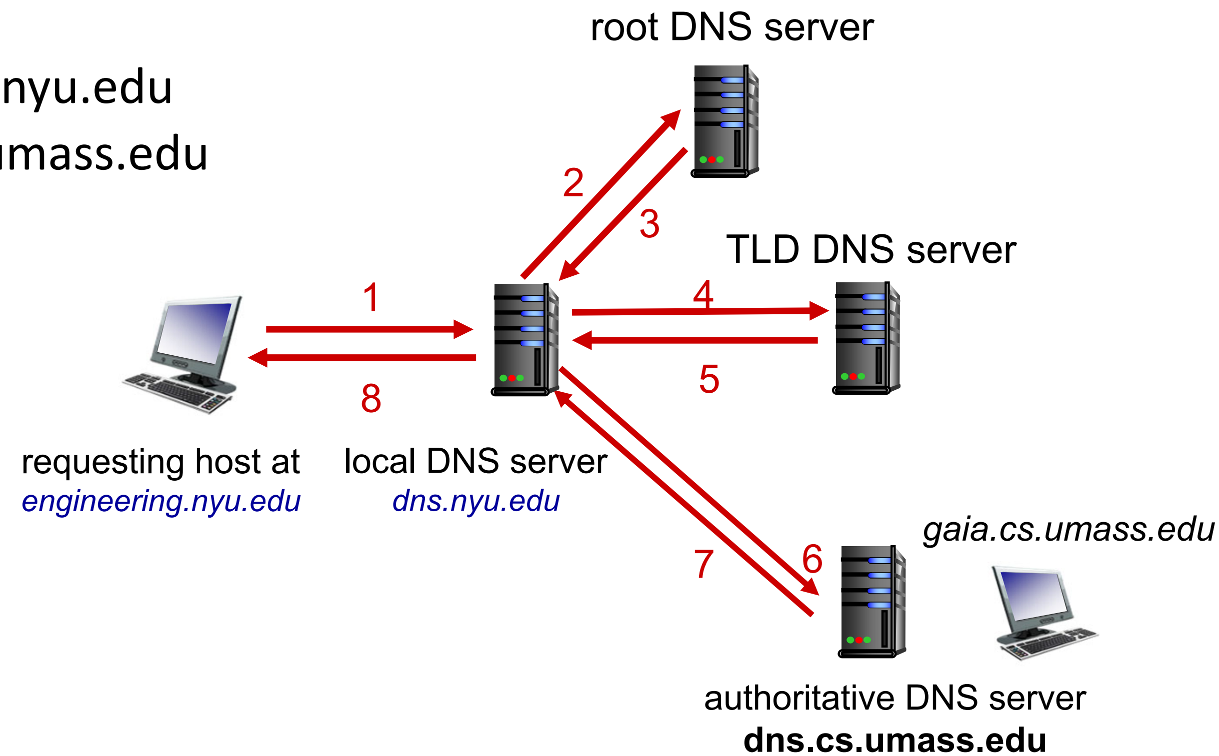
- Resolution: mapping a name to an address
- Recursive resolution:
  - The DNS server that does not know the mapping make queries to other DNS servers on behalf of the client
  - ***"I don't know this name, but I will find it out for you"***
- Iterative resolution:
  - The DNS server that does not known the mapping sends the IP address of the next server back to the one that requested it
  - ***"I don't know this name, but you may ask this server"***

DNS name resolution: iterated query

Example: host at engineering.nyu.edu wants IP address for gaia.cs.umass.edu

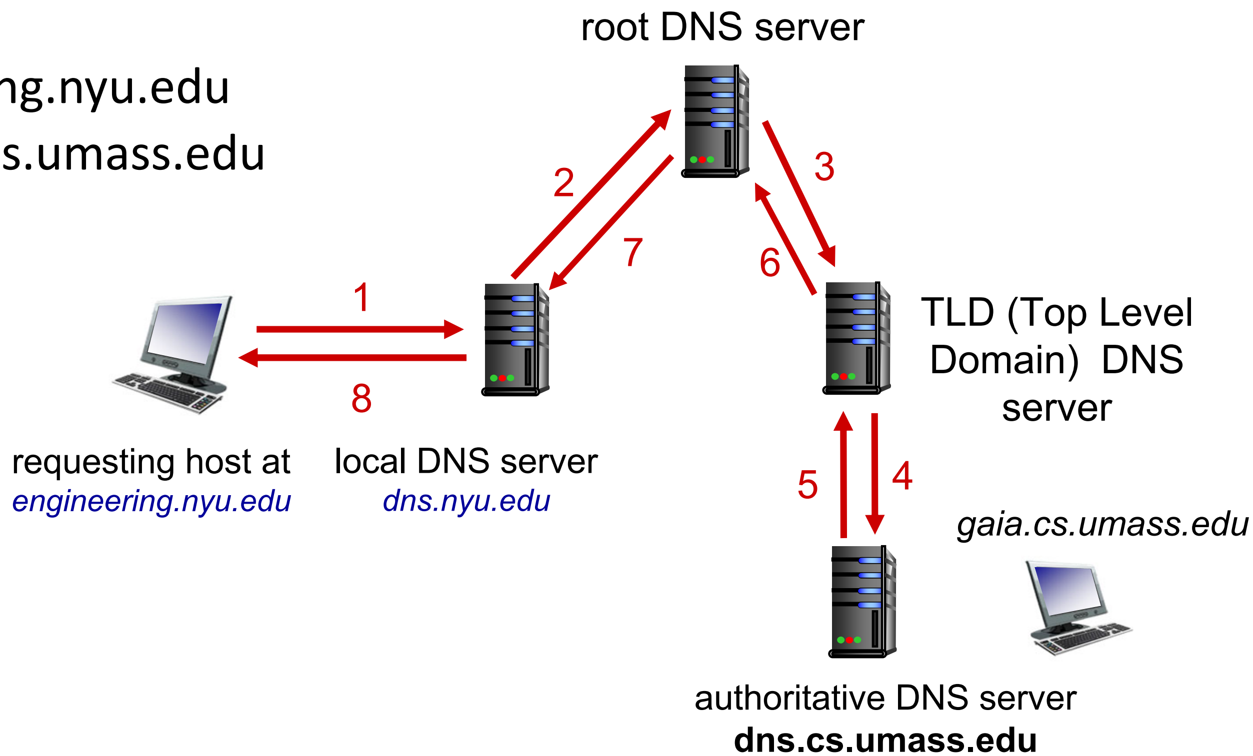
Iterated query:

- contacted server replies with name of server to contact
- “I don’t know this name, but ask this server”



DNS name resolution: recursive query

Example: host at engineering.nyu.edu wants IP address for gaia.cs.umass.edu

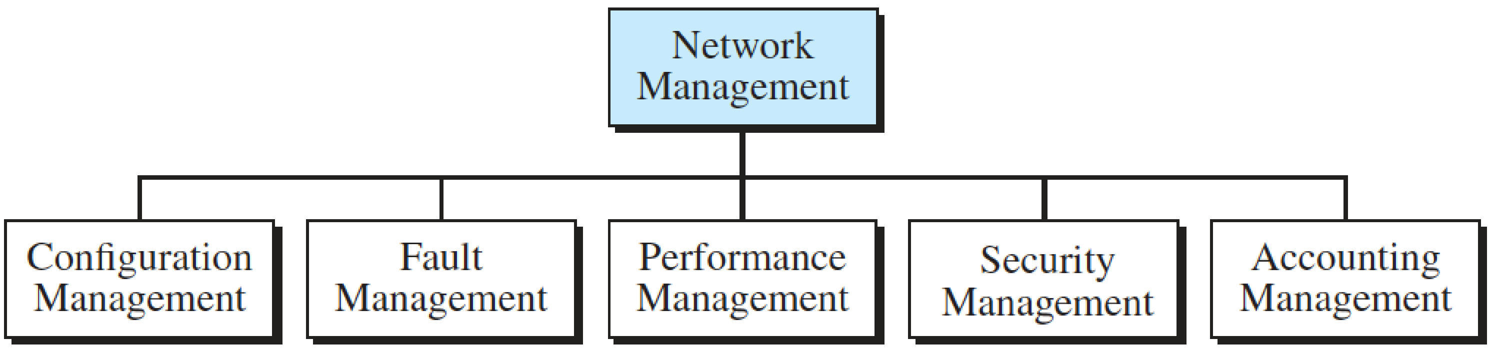


Recursive query:

- puts burden of name resolution on contacted name server
- heavy load at upper levels of hierarchy?

Network Management

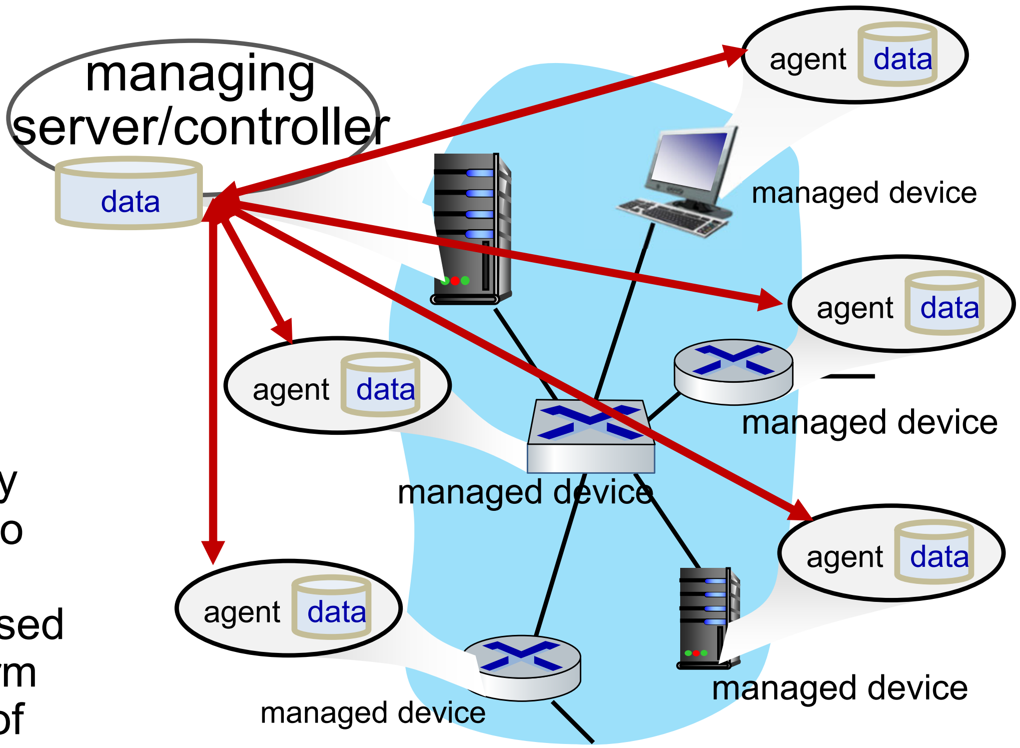
- Network management: monitoring, testing, configuring, and troubleshooting network components to meet a set of requirement



Components of network management

Managing server: application, typically with network managers (humans) in the loop

Network management protocol: used by managing server to query, configure, manage device; used by devices to inform managing server of data, events.



Managed device: equipment with manageable, configurable hardware, software components

Data: device "state" configuration data, operational data, device statistics

Network operator approaches to management

CLI (Command Line Interface)

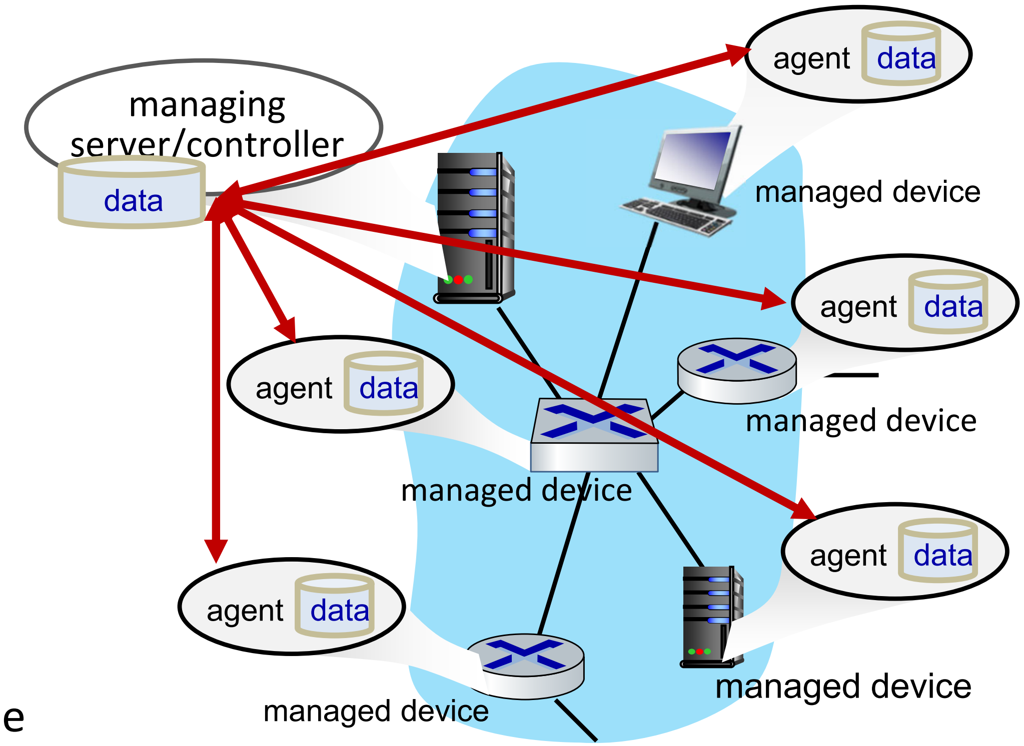
- operator issues (types, scripts) direct to individual devices (e.g., vis ssh)

SNMP/MIB

- operator queries/sets devices data (MIB) using Simple Network Management Protocol (SNMP)

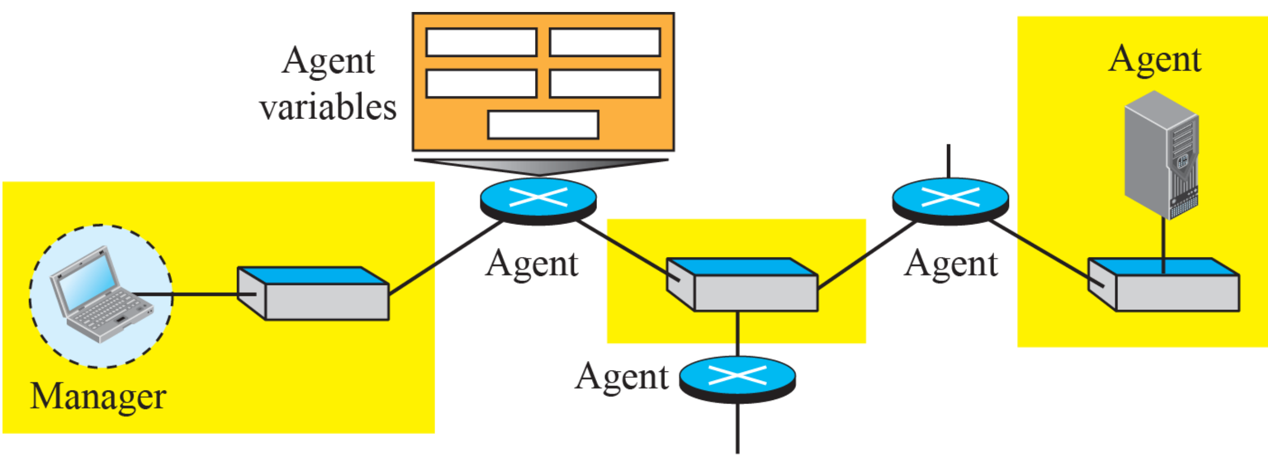
NETCONF/YANG

- more abstract, network-wide, holistic
- emphasis on multi-device configuration management.
- YANG: data modeling language
- NETCONF: communicate YANG-compatible actions/data to/from/among remote devices



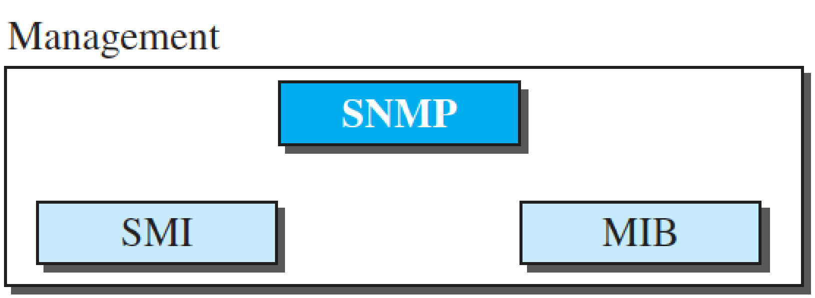
Simple Network Management Protocol (SNMP)

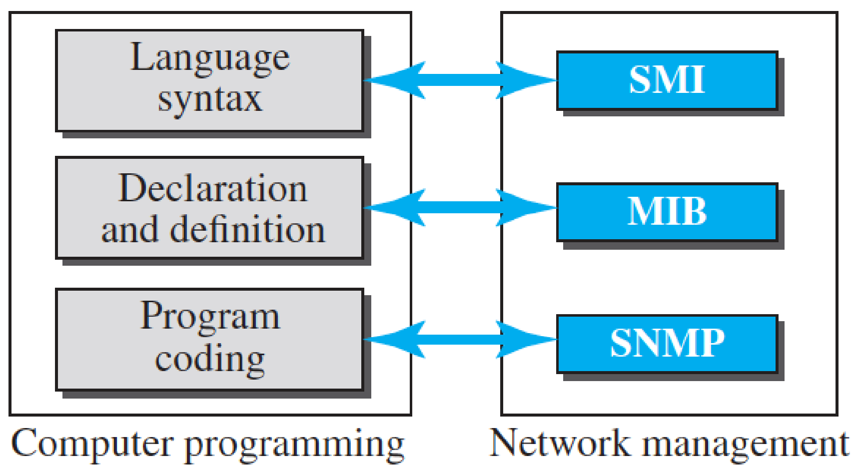
- An Application Layer protocol for managing devices in Internet using TCP/IP protocol suite
- Manager: a host that runs the SNMP client program
- Agent: a router or host that runs the SNMP server program
- Three basic ideas for management with SNMP
  - Manager checks an agent by requesting information of the agent
  - Manager forces agent to perform a task by resetting values in the agent database
  - Agent warns the manager of unusual situation



Management Protocols

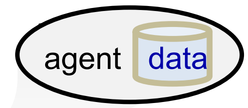
- Internet management needs the cooperation of the following three protocols:
- SNMP:
  - Defines the format of packets exchanged between a manager and an agent
  - Reads and changes the status of objects in SNMP packets
- Structure of Management Information (SMI)
  - Defines the general rules for naming objects
  - Defines object types and how to encode objects and values
- Management Information Base (MIB)
  - Creates a collection of named objects, their types and relationships





SNMP: Management Information Base (MIB)

- managed device's operational (and some configuration)
- gathered into device MIB module
  - 400 MIB modules defined in RFC's; many more vendor-specific MIBs

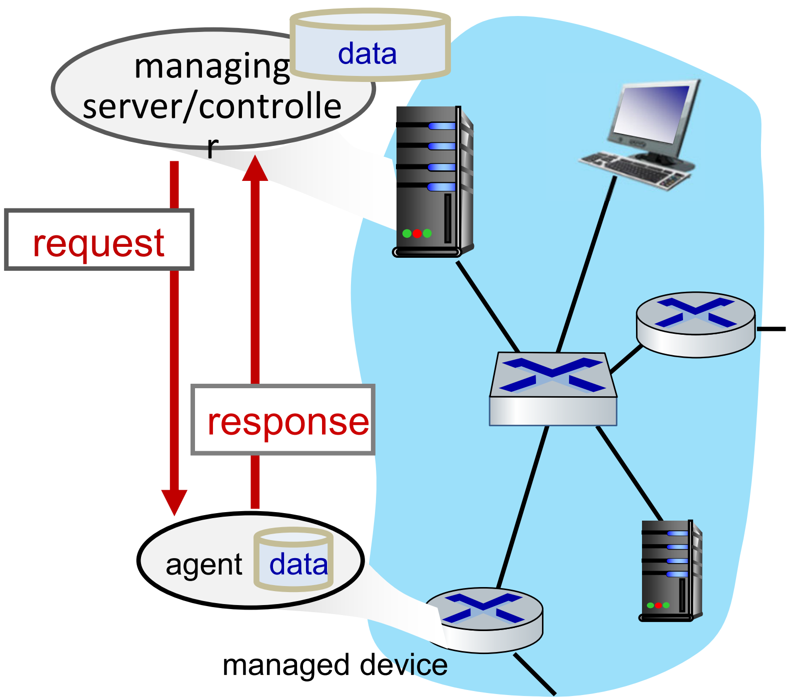


- Structure of Management Information (SMI): data definition language
- example MIB variables for UDP protocol:

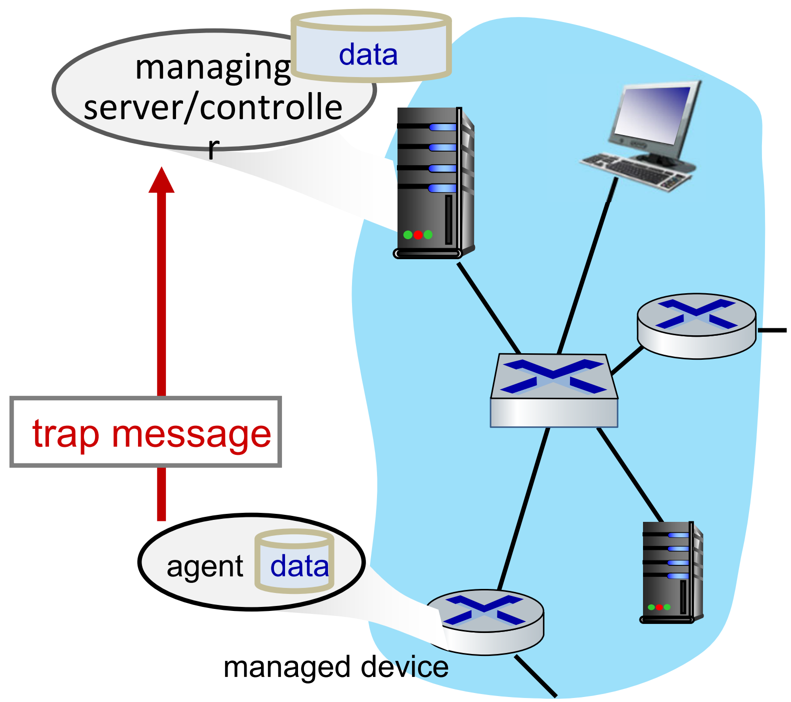
Object ID	Name	Type	Comments
1.3.6.1.2.1.7.1	UDPInDatagrams	32-bit counter	total # datagrams delivered
1.3.6.1.2.1.7.2	UDPNoPorts	32-bit counter	# undeliverable datagrams (no application at port)
1.3.6.1.2.1.7.3	UDInErrors	32-bit counter	# undeliverable datagrams (all other reasons)
1.3.6.1.2.1.7.4	UDPOutDatagrams	32-bit counter	total # datagrams sent
1.3.6.1.2.1.7.5	udpTable	SEQUENCE	one entry for each port currently in use

SNMP protocol

Two ways to convey MIB info, commands:



request/response mode

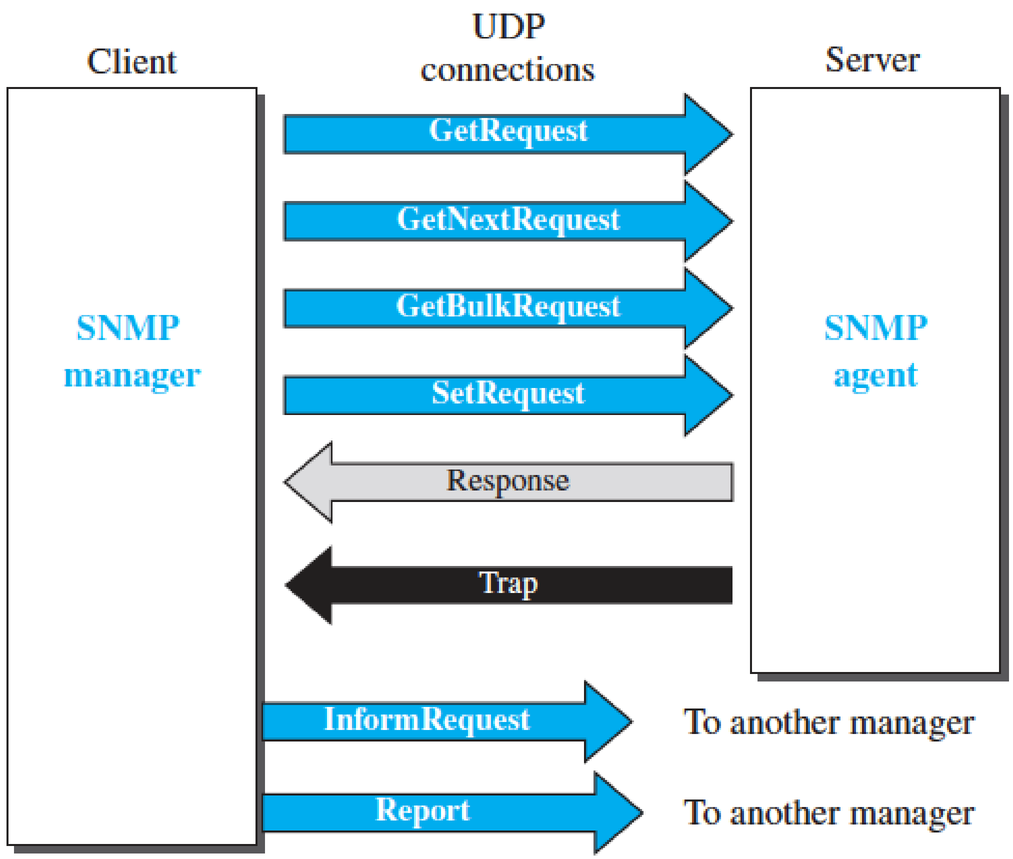


trap mode

Unsolicited massage by agent

SNMP Operations

- SNMPv3 defines eight types of packets (or protocol data units, PDUs): *GetRequest, GetNextRequest, GetBulkRequest, SetRequest, Response, Trap, InformRequest, and Report*



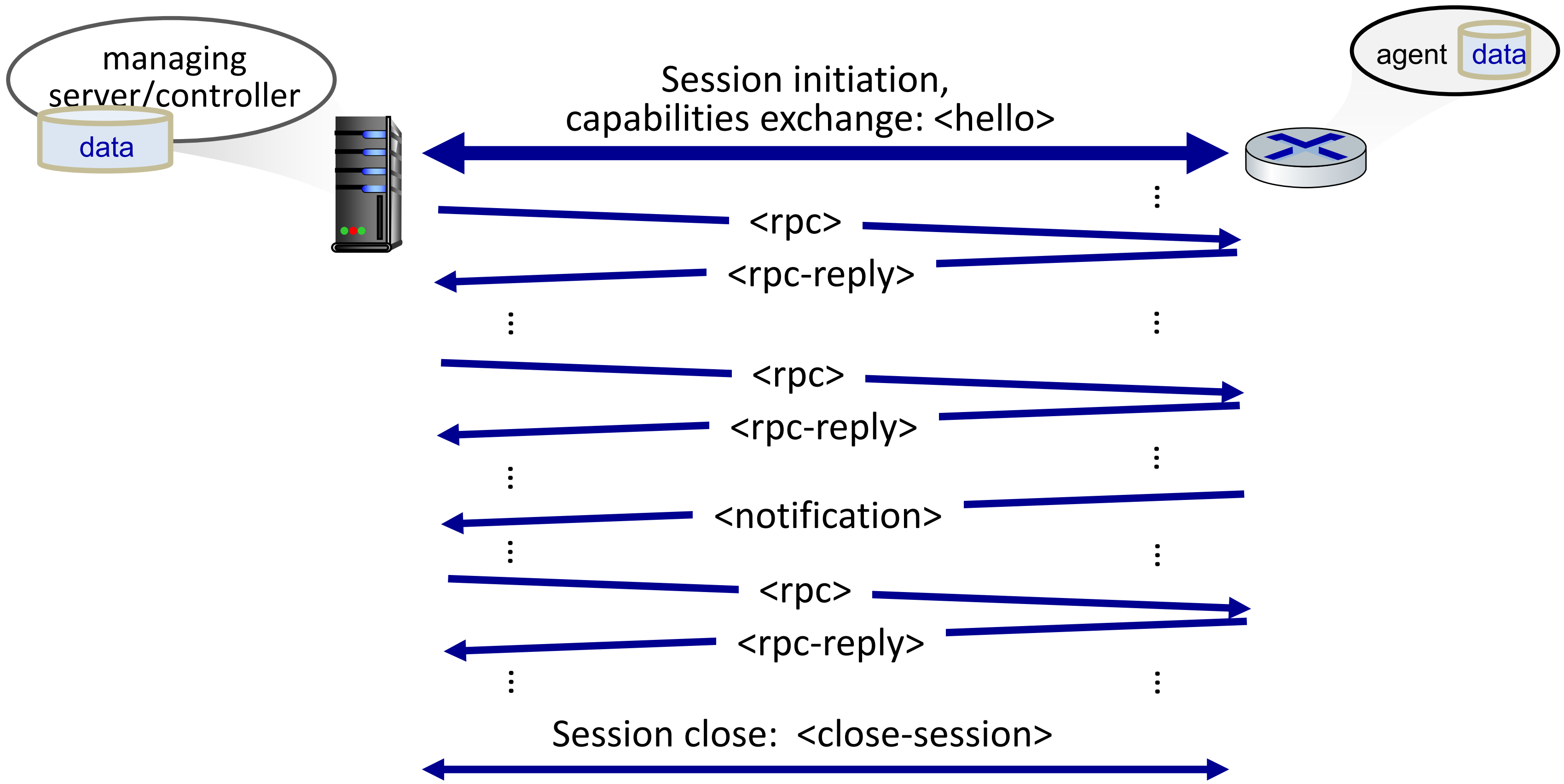
SNMP Operations

Message type	Direction	Function
GetRequest GetNextRequest GetBulkRequest	Manager to agent	Retrieve value of variables (instance, next in list, block)
SetRequest	Manager to agent	Set value in a variable
Response	Agent to manager	Give the value of variables requested by manger
Trap	Agent to manager	Inform manager of exceptional event
InformRequest	Manager to manager	Get value of variables from agent under the control of remote manager
Report	Manager to manager	Report errors between managers

NETCONF (Network Configuration Protocol)

- Goal: actively manage/configure devices network-wide
- operates between managing server and managed network devices
  - actions: retrieve, set, modify, activate configurations
  - atomic-commit actions over multiple devices
  - query operational data and statistics
  - subscribe to notifications from devices
- remote procedure call (RPC) paradigm
  - NETCONF protocol messages encoded in XML
  - exchanged over secure, reliable transport protocol

NETCONF initialization, exchange, close

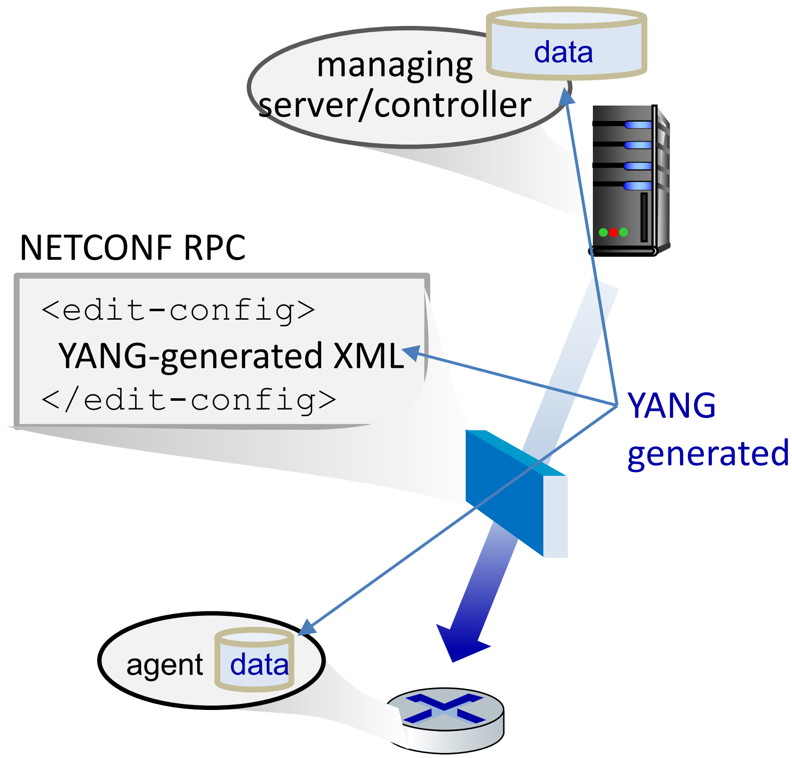


Selected NETCONF Operations

NETCONF	Operation Description
<get-config>	Retrieve all or part of a given configuration. A device may have multiple configurations.
<get>	Retrieve all or part of both configuration state and operational state data.
<edit-config>	Change specified (possibly running) configuration at managed device. Managed device <rpc-reply> contains <ok> or <rpcerror> with rollback.
<lock>, <unlock>	Lock (unlock) configuration datastore at managed device (to lock out NETCONF, SNMP, or CLIs commands from other sources).
<create-subscription>, <notification>	Enable event notification subscription from managed device

YANG

- data modeling language used to specify structure, syntax, semantics of NETCONF network management data
  - built-in data types, like SMI
- XML document describing device, capabilities can be generated from YANG description
- can express constraints among data that must be satisfied by a valid NETCONF configuration
  - ensure NETCONF configurations satisfy correctness, consistency constraints



Recommended Reading

- Behrouz A. Forouzan, Data Communications and Networking with TCP/IP Protocol Suite, 6th ed., 2022, Chapters 8 and 10

- J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach, 8th ed., 2022, Chapters 1, 2 and 5