Lecture 3.c
Transport Layer
And DNS

Dr. Gabriele Pierantoni

02.03.2021

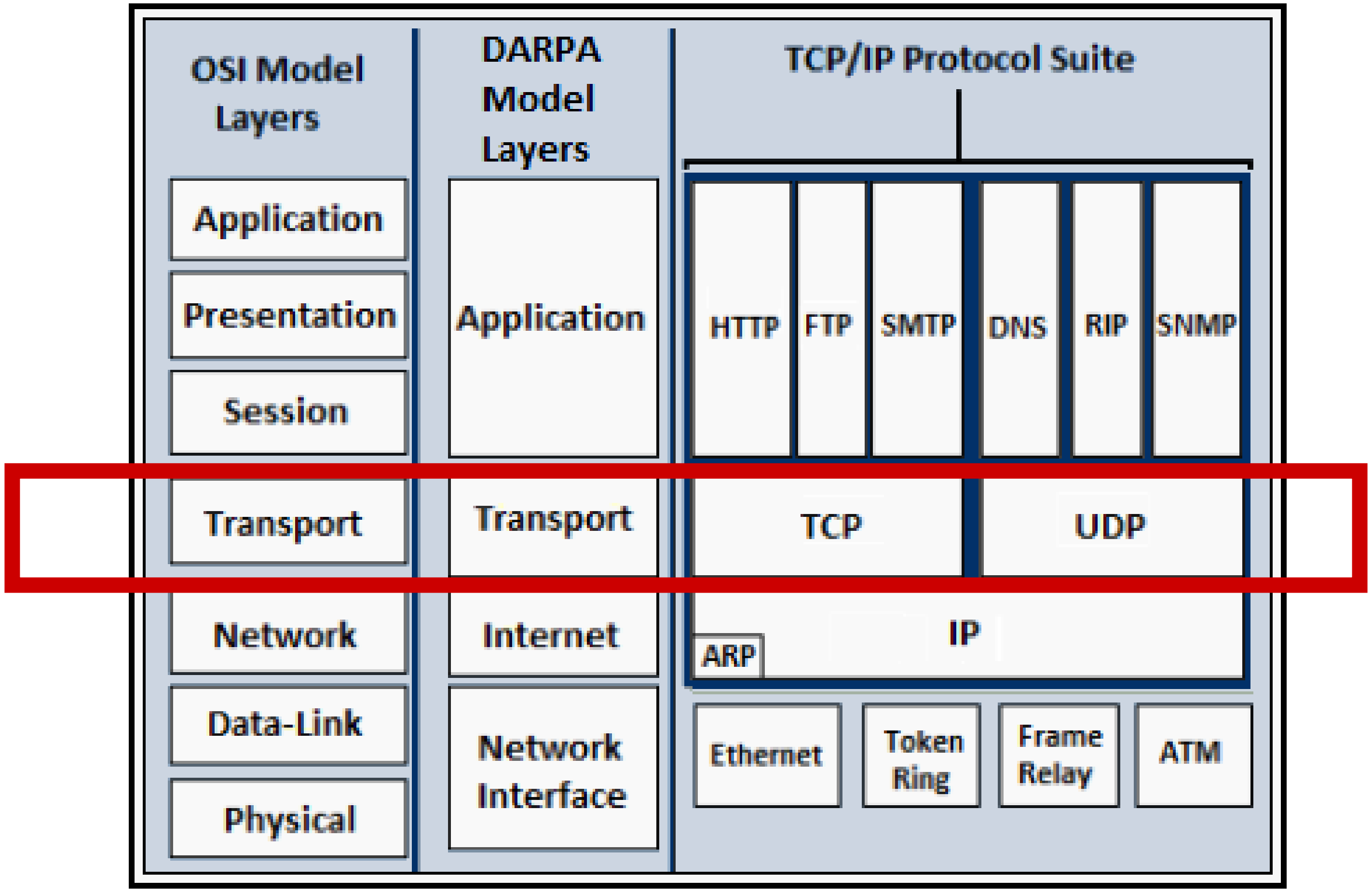
Today Contents

- Networking 2/2
  - Network Layer
  - **Transport Layer**

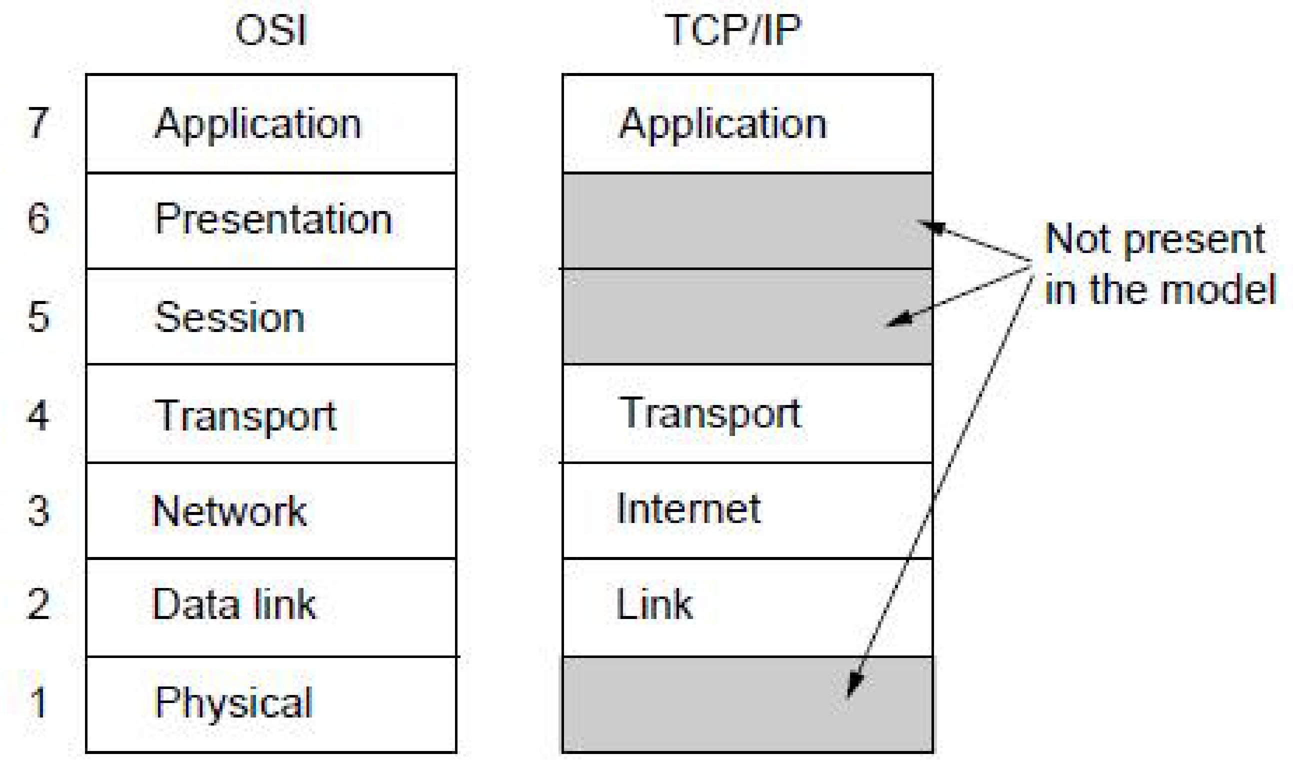
    and DNS

Networking Protocols

Where are we ?



OSI vs TCP/IP (Remember?)



(Today) Transport Layer/TCP

Application
Transport
Network (internet)
Link
Physical

- The TRANSPORT LAYER is the most important layer in the protocol stack
- Applications talk to the transport layer to get data sent across a network
- This conversation takes place by means of the Network Application Programming Interface (API)
- The Transport Layer is essentially the working interface on to the network
- **Whenever an application wants to send data or receive data, it calls upon the transport layer by opening a socket**
- **When developers write network applications they use sockets to allow the application access the network**

Goals of the layer

- The goals for the Transport Layer on the Internet for typical data are as follows:
  - Reliable data transfer: Error control etc…
  - Flow control:
    - Ensure packets are arranged in the correct order at the receiving end
    - Ensure that a fast sender does not swamp a slow receiver
- Congestion control: Throttle the sender when it overloads the network
- Its all of the things that we haven’t done at the other layers

Internet Transport Layer

- Application layer processes communicate with each other by sending and receiving messages through their sockets
- The socket can be thought of as a door that will transport the message to the door of another process on another host
- There are two transport mechanisms on the other side of the door (socket) that can take messages from the application
  - Transmission Control Protocol (TCP)
  - User Datagram Protocol (UDP)

Services offered

- Transmission Control Protocol (TCP) Service:
  - **connection-oriented**: setup required between client and server
  - **reliable** transport between sending and receiving process
  - **flow control**: sender wont overwhelm receiver
  - **congestion control**: throttle sender when network overloaded
  - does not provide: timing, minimum bandwidth guarantees
- We use it for getting web pages, email, file transfer basically where we need reliable transport of data

Services Offered

- User Datagram Protocol (UDP) Service:
  - **Unreliable data transfer** between sending and receiving process
  - Does not provide: connection setup, reliability, flow control, congestion control, timing, or bandwidth guarantee
  - **Best effort service**, UDP segments may be:
    - lost
    - delivered out of order to application

- Q: Why bother? Why is there a UDP?
- A: Basically because its very simple
  - No connection state at sender, receiver
  - No receive/send buffers
  - No Congestion control parameters
  - No Sequence, acknowledgement parameters
- Useful where real time delivery is required - multimedia

TCP

- First we will focus on TCP.
- Remember that from the point of view of the Application the TCP connection is a direct virtual pipe between the clients socket and the servers connection socket
- The application process sends an output stream to the socket and accepts an input stream from the socket

TCP

- An application process informs the TCP layer that it wishes to establish a connection
- Then the client TCP establishes a connection to Server TCP
  - First, three-way handshake
  - Once connection is established data can be sent
- Client and server can engage in full duplex transmission
- Client application process passes an output stream (of data) through the socket
  - TCP directs the data to a send buffer
  - TCP will grab chunks of data from send buffer from time to time

TCP

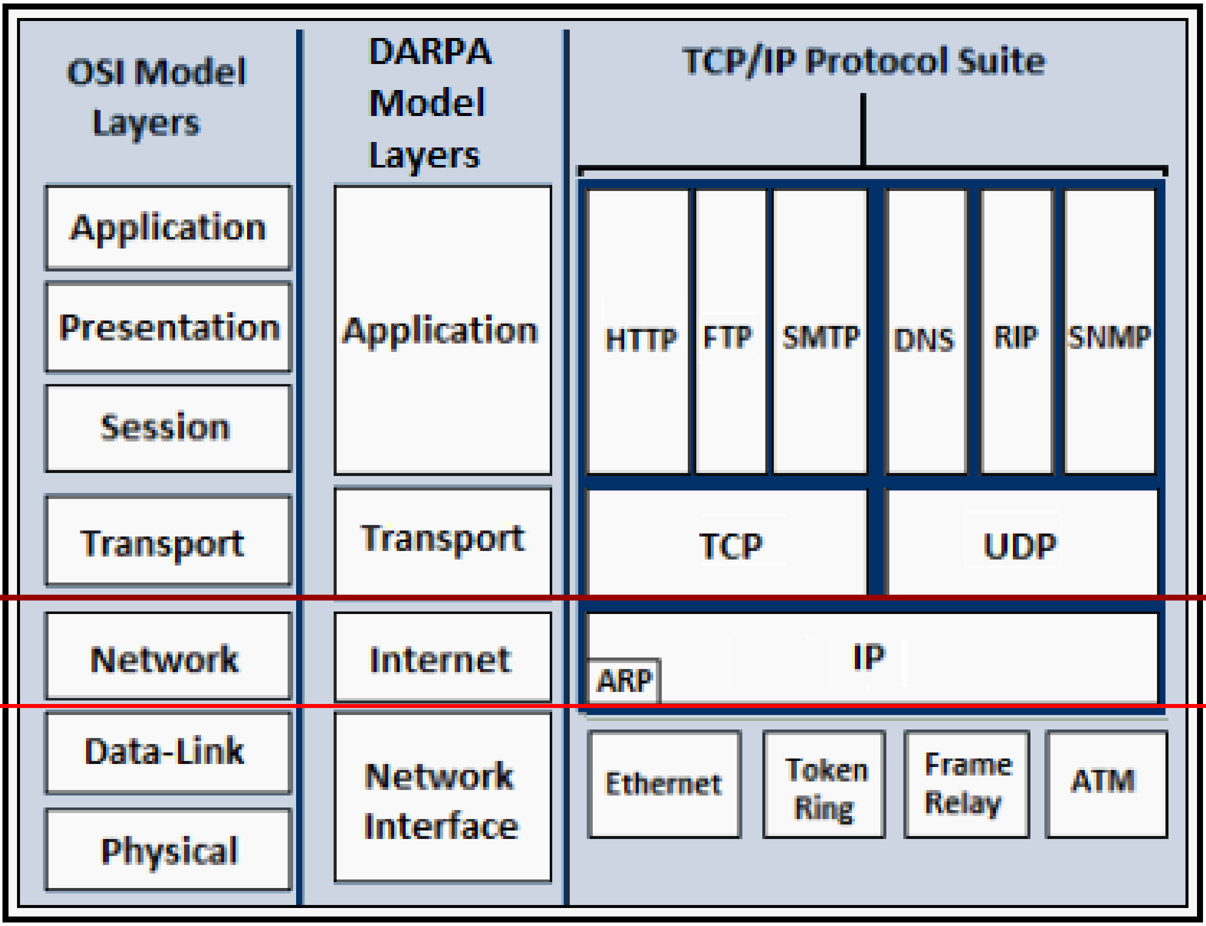
- First though: TCP - How does it provide a connection-oriented, reliable, flow controlled, congestion controlled service.

- Think about the service available to TCP from the network layer below it

- Using IPs service TCP offers the Application Layer connection-oriented, reliable, flow controlled, congestion controlled service. But IP offers TCP an unreliable, best effort service, if it loses packets it wont inform TCP.

- So, how does it do it?

Networking Protocols



Attempt to make Internet reliable =>

Unreliable =>

Internet Protocol

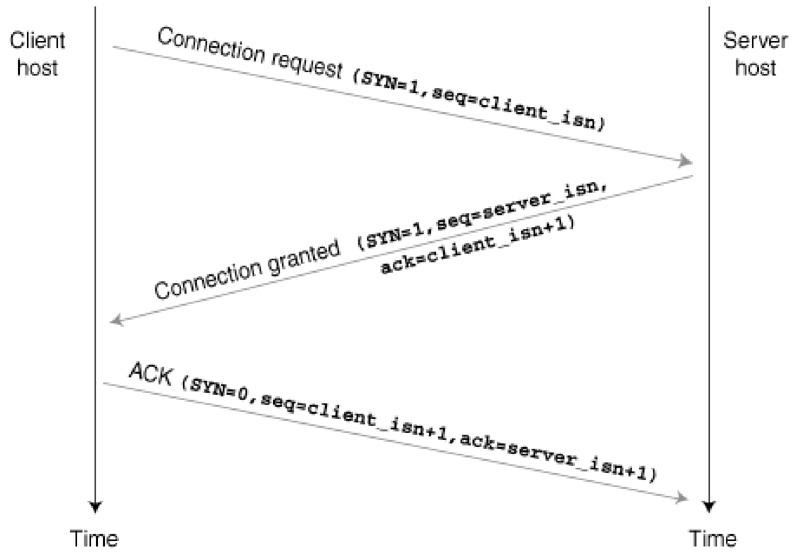
- **Internet Network Layer** offers a connectionless datagram service to TCP. The network layer encapsulates the segments received from the Transport layer in an IP datagram. The IP header includes the destination address.
- **IP service is best effort**
  - Does not guarantee packet will arrive within a certain time
  - Does not guarantee sequential delivery
  - Does not guarantee that delivery will occur at all

Reliability

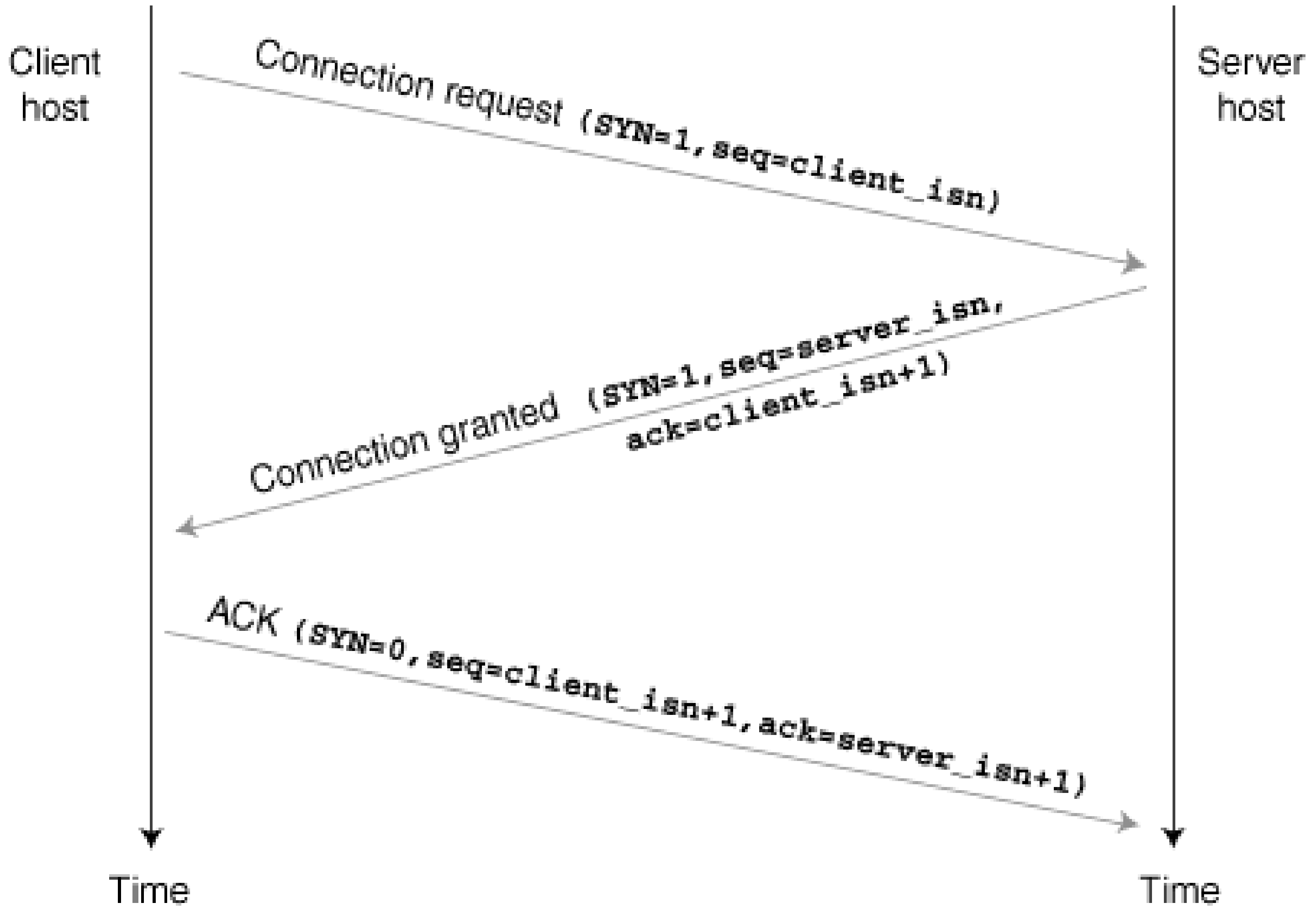
- TCP has to make sure that all the segments sent from our client to our server (and vice versa) arrive
  - Some will get lost on the way
  - So it needs some means of finding out from the receiver if a segment has arrived: Acknowledgements
  - If it hasn’t, it needs to be retransmitted
- It also has to make sure that all the segments are reassembled in the right order and are delivered to our application
  - Segments will arrive at the receiver out of order
  - So it has to know how to reassemble the original document (a web page for instance)
- It gives each segment a Sequence Number

Making a connection

- A TCP Connection is established using a **Three Way Handshake**
- **Step 1**: client TCP sends TCP control message to the server. Tells the server the sequence number of the first segment of data it is going to send
- **Step 2**: server TCP replies with an acknowledging control segment. Allocates buffers. Specifies initial sequence number.
- **Step 3**: client replies. Acknowledges servers message, allocates buffers and variables to the connection.


Client host
Server host
Connection request (SYN=1,seq=client_isn)
Connection granted (SYN=1,seq=server_isn, ack=client_isn+1)
ACK (SYN=0,seq=client_isn+1,ack=server_isn+1)
Time
Time

Making a connection


Client host
Server host
Connection request (SYN=1,seq=client_isn)
Connection granted (SYN=1,seq=server_isn, ack=client_isn+1)
ACK (SYN=0,seq=client_isn+1,ack=server_isn+1)
Time
Time

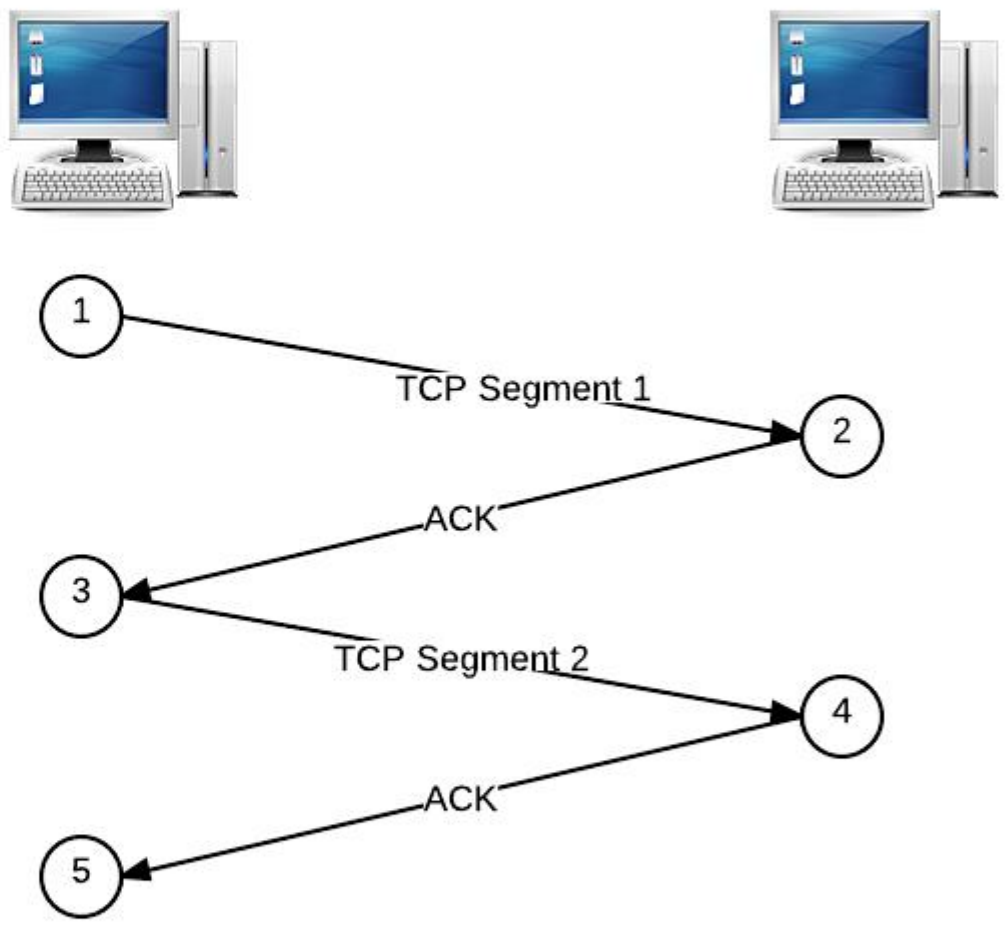
Issue

- The mechanism just described is called stop and wait and is not used in practice. It is Very inefficient.
- Consider two hosts communicating over a long distance:
  - Sender has to wait for an ACK after each segment transmission before transmitting again.
  - If Round Trip Time is long then the Sender will be idle waiting for an ACK most of the time
  - And overall transmission rate will be very slow
- Round Trip Time: The time it takes for a packet to get from one host to another host and back again.

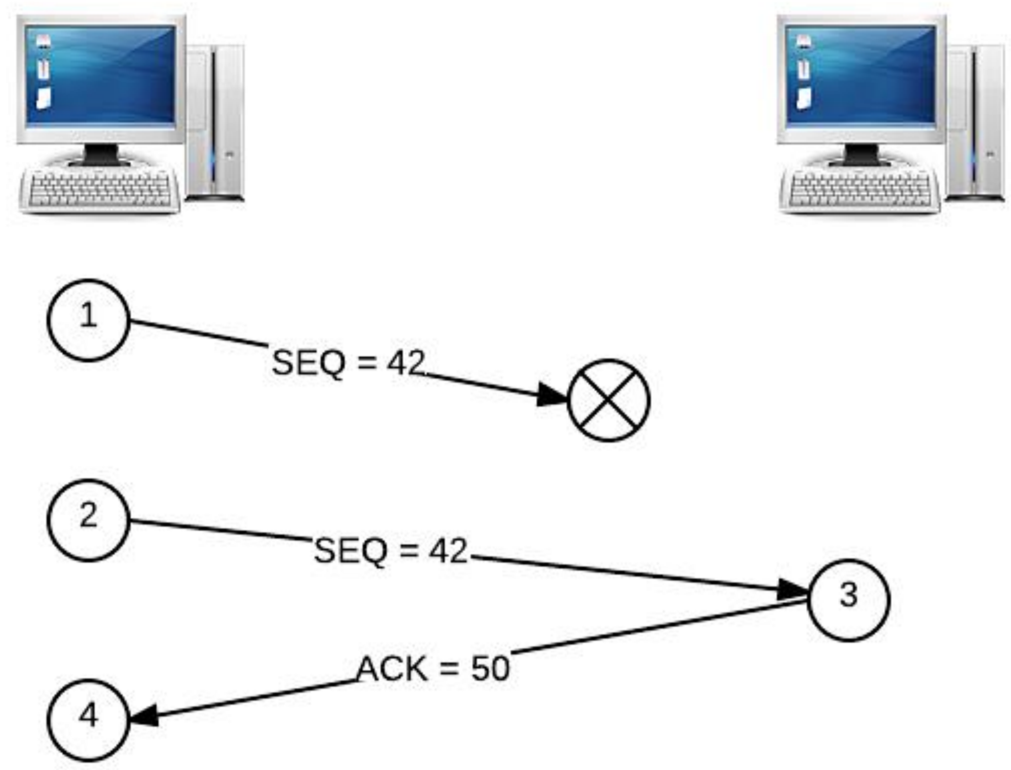
Reliable Data Transfer

- **TCP achieves reliable delivery using a mechanism called positive acknowledgement with retransmission**
  - For every segment sent the TCP sender starts a timer
  - When the receiver receives the segment it returns an acknowledgement (ACK) containing the sequence number of the correctly received segment
  - If the ACK arrives at the sender before the timeout on the timer, the sender cancels the timer for that segment
  - If the sender doesn’t receive an ACK before the timer expires it retransmits that segment

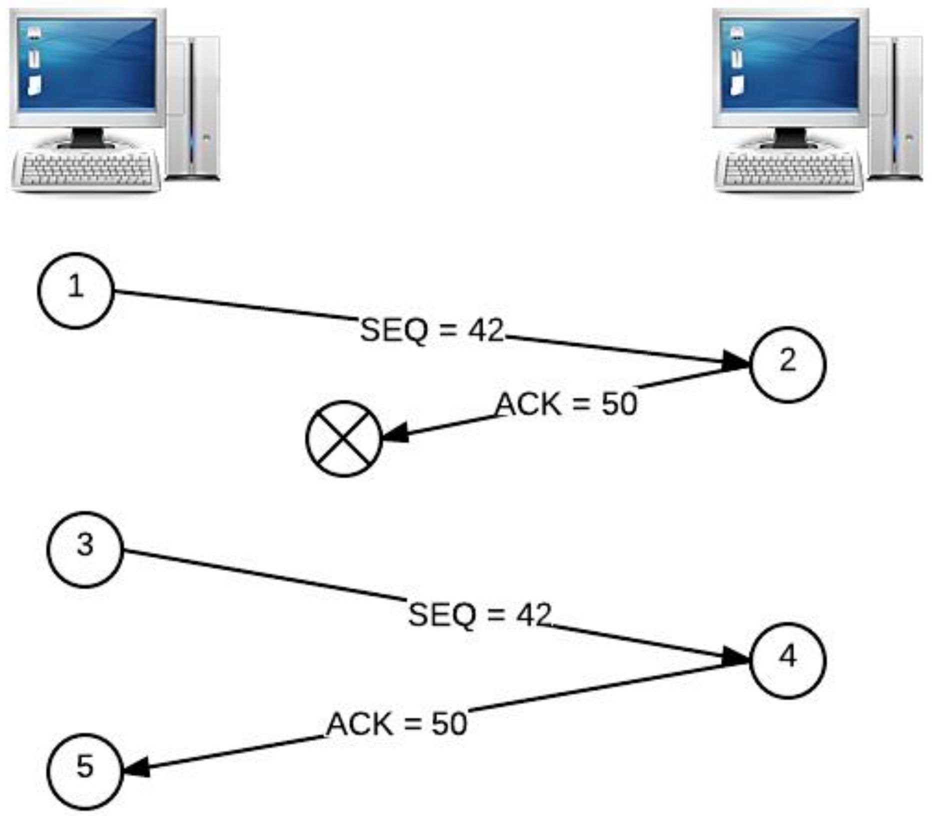
TCP Segments



TCP Segments



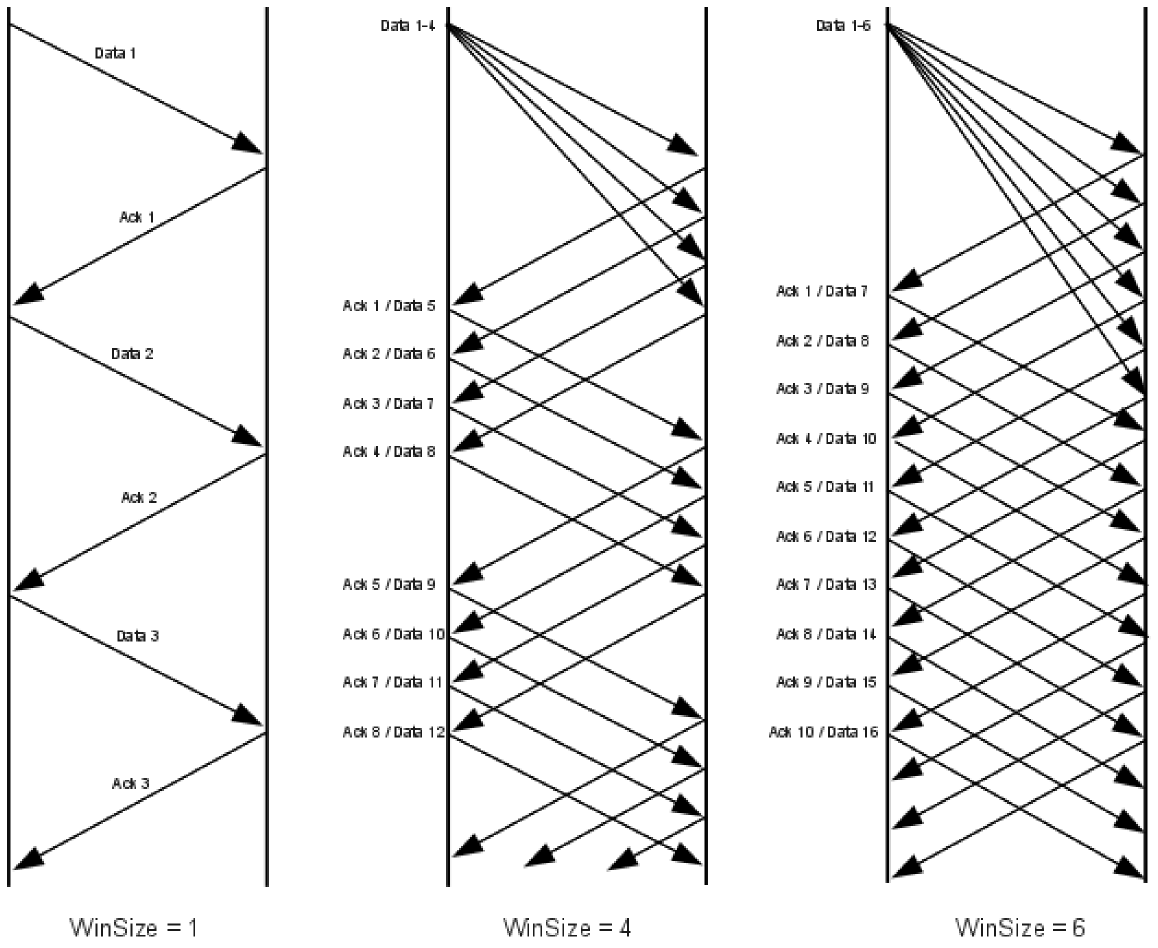
TCP Segments



Pipelining

- Pipelining: sender is allowed send multiple packets without waiting for acknowledgements
- Pipelining: filling the pipe
- Requires us to have a large range of sequence numbers
- We need buffering at sender and the receiver:
- Sender: packets sent not yet ACKed (may need to resend them)
- Receiver requires a receive buffer (hold out of order packets prior to reassembly)
- TCP uses the Sliding Window pipelining protocol

Sliding Windows



Sliding Windows, bandwidth 6 packets/RTT

Flow and Congestion Control

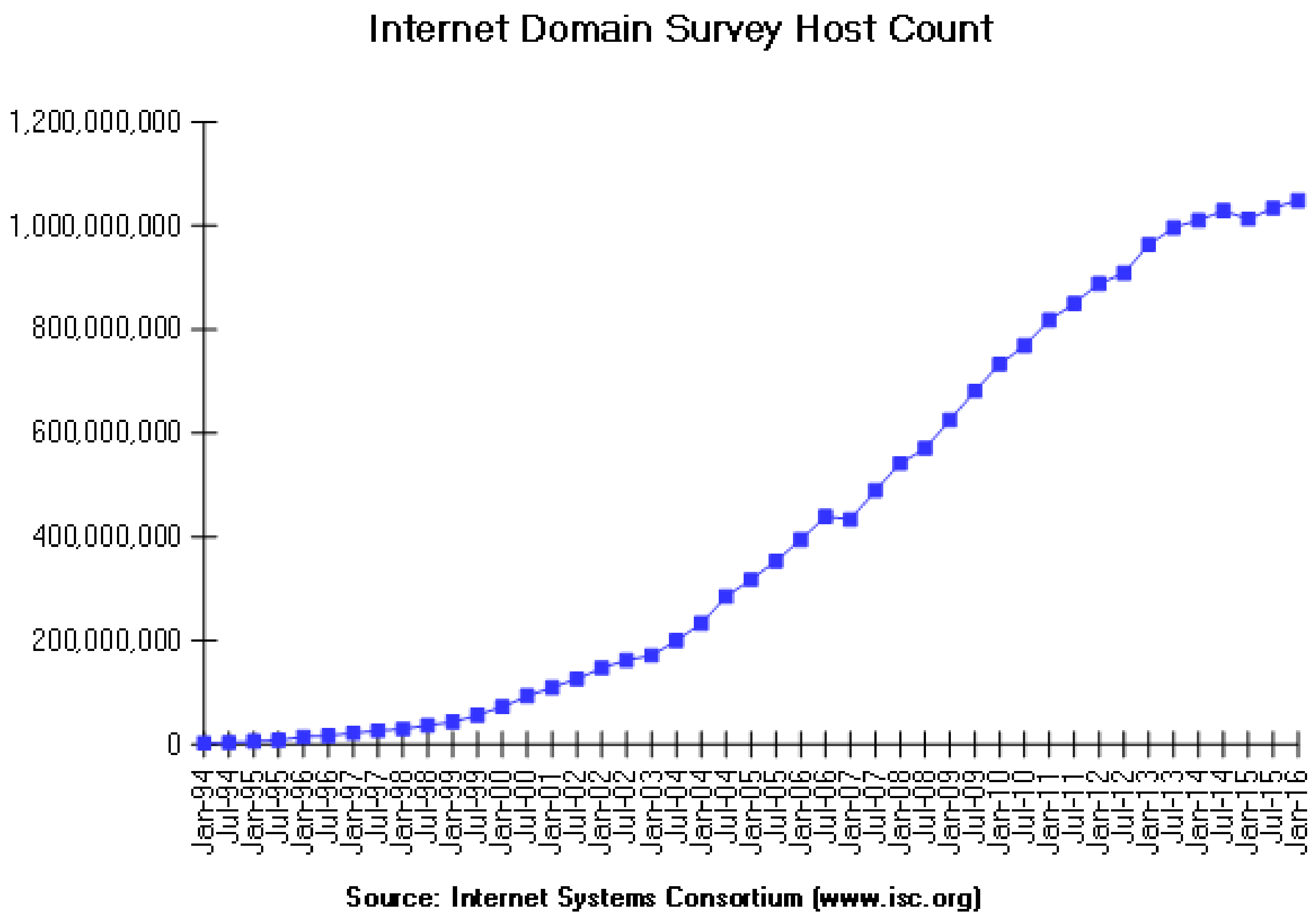
- **Flow Control**: if we send at will without consideration for the receivers rate of consumption, we could flood the receiver. So the sender has to have a sending window reflecting the amount of buffer space remaining in the receiver.

- **Congestion control**: if we send at a rate greater than can be handled by the intermediate links on the network, then we will have packet loss (which will get worse and worse as we actually increase traffic with retransmissions). So to avoid network collapse, the sender also has to have a sending window that is less than that which will cause packet loss

What does DNS do?

- Provides hostname – IP lookup services
- DNS defines
  - A hierarchical namespace for hosts and IP addresses
  - A distributed database of hostname and address info
  - A “resolver” – library routines that query this database
  - Improved routing for email
  - A mechanism for finding services on a network
  - A protocol for exchanging naming information
- DNS is essential for any org using the Internet

Addresses

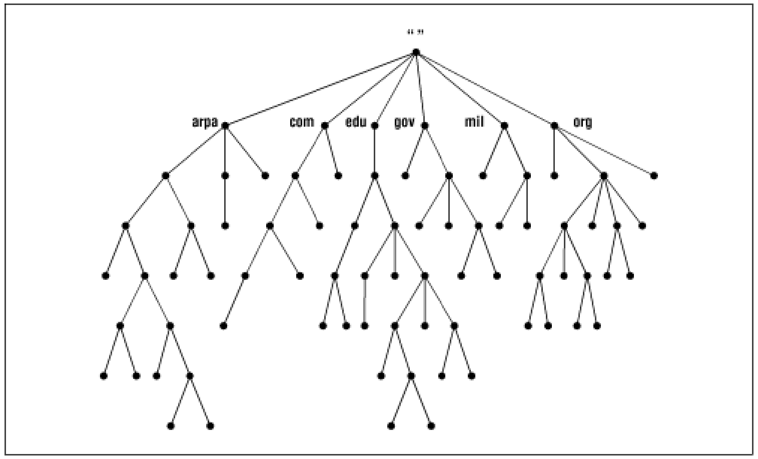

Internet Domain Survey Host Count
1,200,000,000
1,000,000,000
800,000,000
600,000,000
400,000,000
200,000,000
0
Jan-94
Jul-94
Jan-95
Jul-95
Jan-96
Jul-96
Jan-97
Jul-97
Jan-98
Jul-98
Jan-99
Jul-99
Jan-00
Jul-00
Jan-01
Jul-01
Jan-02
Jul-02
Jan-03
Jul-03
Jan-04
Jul-04
Jan-05
Jul-05
Jan-06
Jul-06
Jan-07
Jul-07
Jan-08
Jul-08
Jan-09
Jul-09
Jan-10
Jul-10
Jan-11
Jul-11
Jan-12
Jul-12
Jan-13
Jul-13
Jan-14
Jul-14
Jan-15
Jul-15
Jan-16
Source: Internet Systems Consortium (www.isc.org)

What uses DNS?

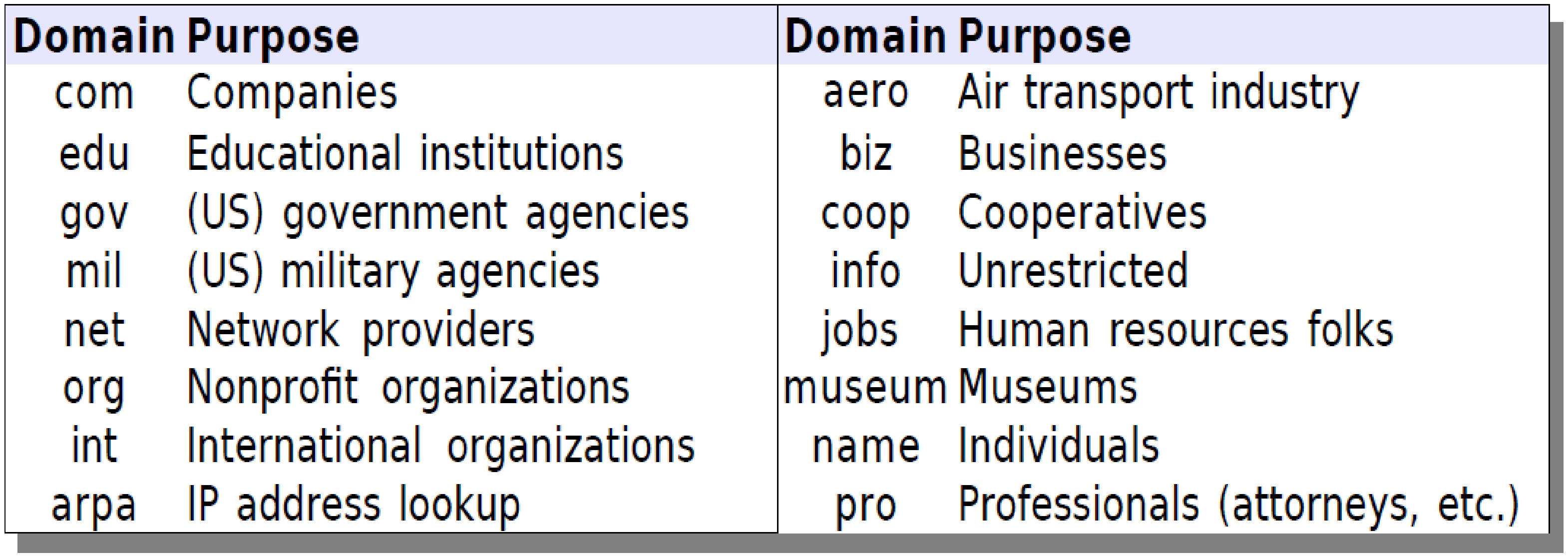
- Any application that operates over the Internet
  - email
    - Spam filters
  - WWW
  - FTP
  - IRC, IM
  - Windows update
  - telnet, ssh

The DNS namespace

- A tree of "domains"
- Root is "." (dot), followed by top-level (root-level) domains
- Two branches of tree
  - One maps hostnames to IP addresses
  - Other maps IP address back to hostnames
- Two types of top-level domain names used today
  - gTLDs: generic top-level domains
  - ccTLDs: country code top-level domains

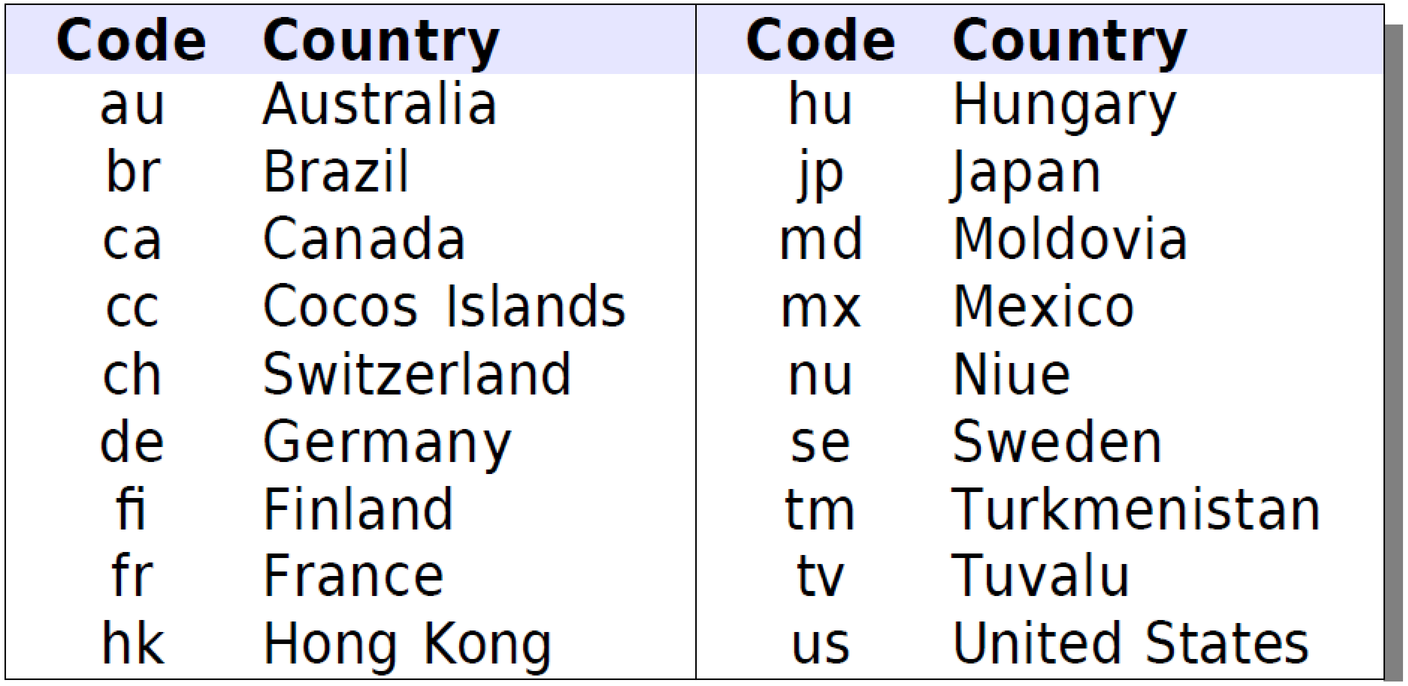


Generic top-level domains



Domain	Purpose
com	Companies
edu	Educational institutions
gov	(US) government agencies
mil	(US) military agencies
net	Network providers
org	Nonprofit organizations
int	International organizations
arpa	IP address lookup
aero	Air transport industry
biz	Businesses
coop	Cooperatives
info	Unrestricted
jobs	Human resources folks
museum	Museums
name	Individuals
pro	Professionals (attorneys, etc.)

Common country codes



Code	Country	Code	Country
au	Australia	hu	Hungary
br	Brazil	jp	Japan
ca	Canada	md	Moldovia
cc	Cocos Islands	mx	Mexico
ch	Switzerland	nu	Niue
de	Germany	se	Sweden
fi	Finland	tm	Turkmenistan
fr	France	tv	Tuvalu
hk	Hong Kong	us	United States

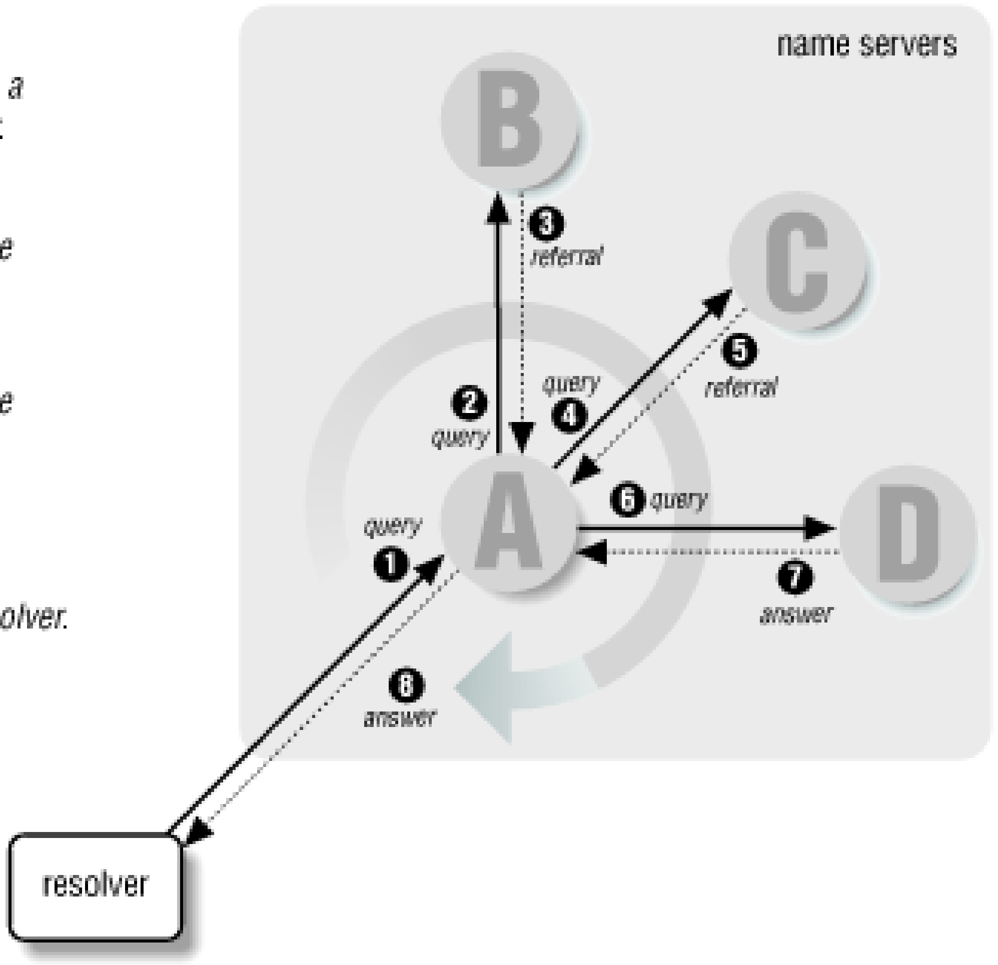
BIND software

- Berkeley Internet Name Domain system
    - By far, the most popular nameserver

- Three components
    - a daemon that answers queries
    - library routines that resolve host queries by contacting DNS servers
    - command-line utilities (nslookup, dig, host)

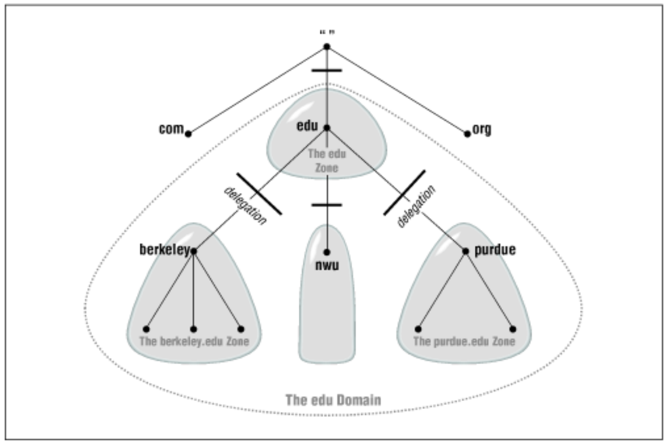
Resolving process

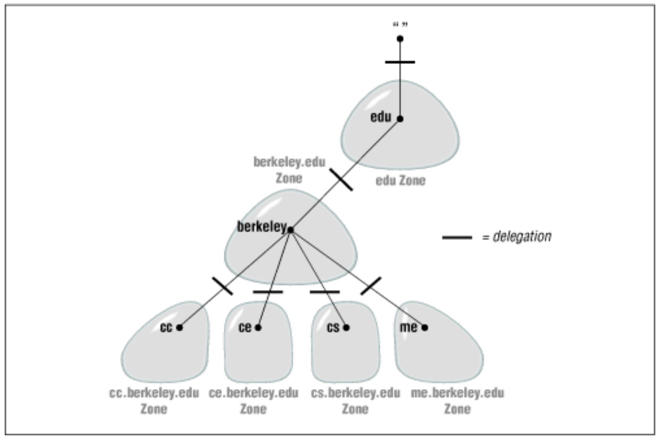
1. Name server **A** receives a query from the resolver.
2. **A** queries **B**.
3. **B** refers **A** to other name servers, including **C**.
4. **A** queries **C**.
5. **C** refers **A** to other name servers, including **D**.
6. **A** queries **D**.
7. **D** answers.
8. **A** returns answer to resolver.



Delegation

- Impractical for high-level servers to know about all hosts (or even subdomains) below
- Servers delegate specific zones to other servers
- Names and addresses of authoritative servers for the relevant zone are returned in referrals

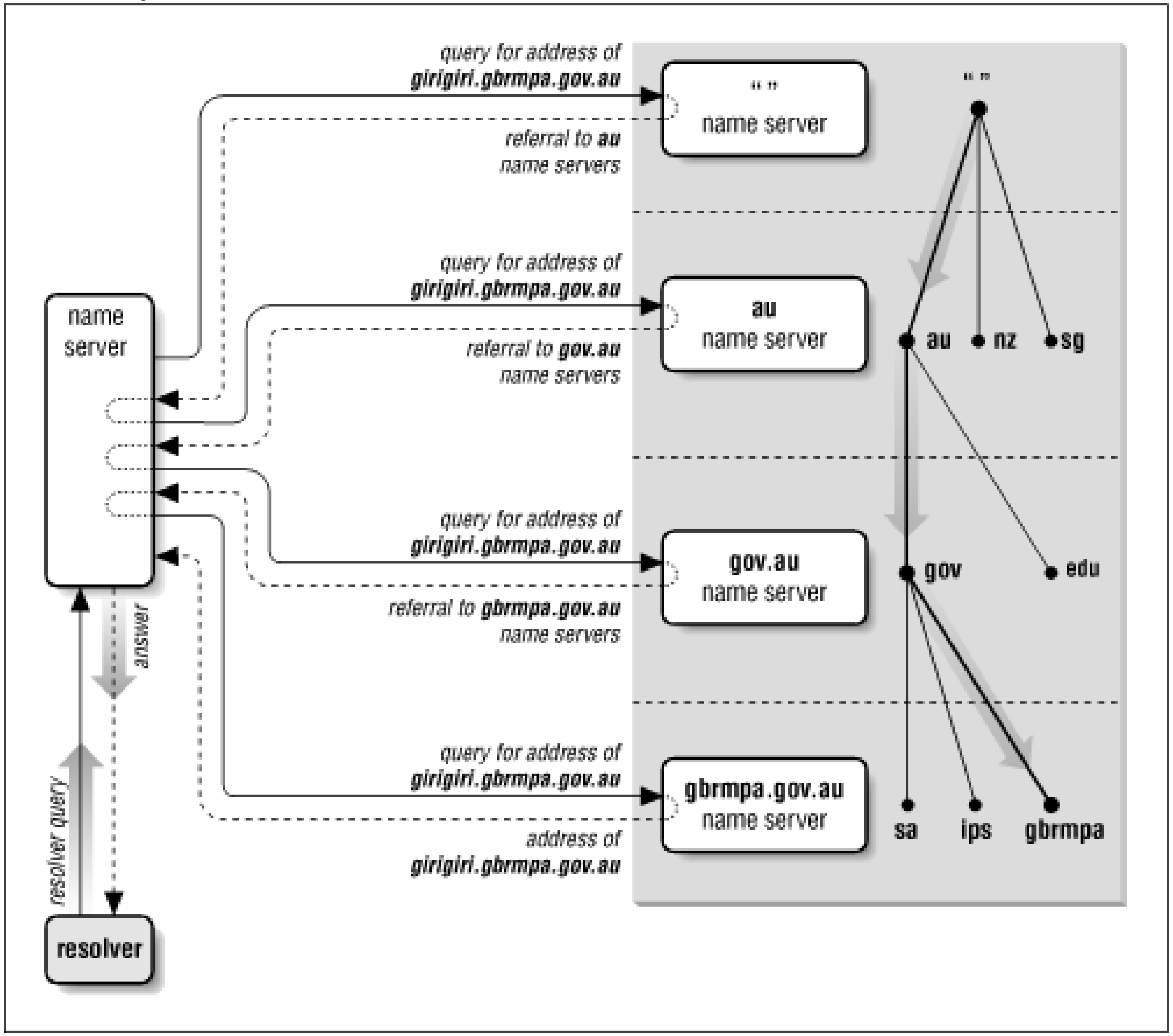




What servers know

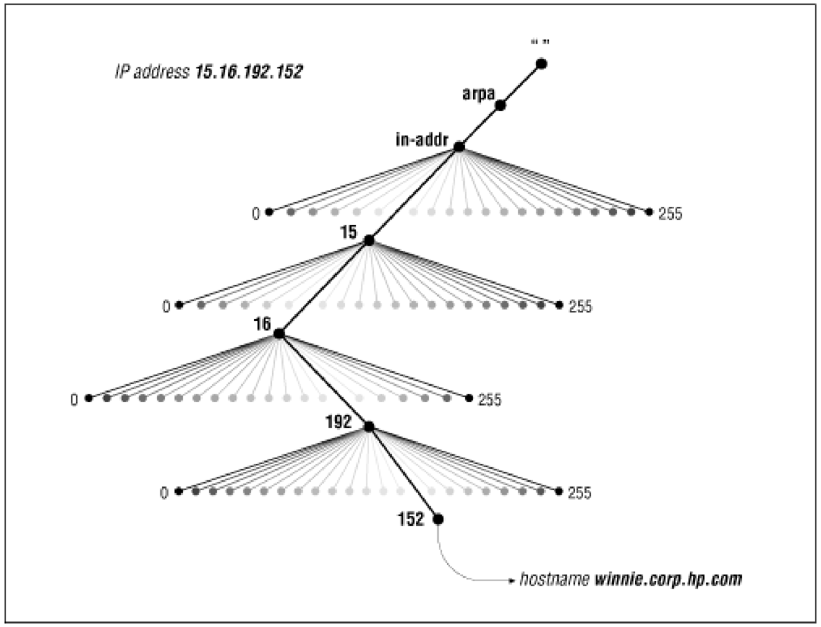
- All servers know about the 13 root servers
  - hardcoded (rarely changes!), or in hint file
  - a.root-servers.net ... m.root-servers.net
- Each root server knows about servers for every top-level domain (.com, .net, .uk, etc.)
- Each top-level domain knows the servers for each second-level domain within the toplevel domain
- Authoritative servers know about their hosts

Example resolution


query for address of girigiri.gbrmpa.gov.au
" " name server
referral to au name servers
query for address of girigiri.gbrmpa.gov.au
au name server
referral to gov.au name servers
query for address of girigiri.gbrmpa.gov.au
gov.au name server
referral to gbrmpa.gov.au name servers
query for address of girigiri.gbrmpa.gov.au
gbrmpa.gov.au name server
address of girigiri.gbrmpa.gov.au
name server
answer
resolver query
resolver
" "
au
nz
sg
gov
edu
sa
ips
gbrmpa

IP-to-hostname resolution

- IP resolution works essentially the same as hostname resolution
- Query for 15.16.192.152
  - Rendered as query for 152.192.16.15.in-addr.arpa
- Each layer can delegate to the next



IP address 15.16.192.152
" "
arpa
in-addr
0
255
15
0
255
16
0
255
192
0
255
152
hostname winnie.corp.hp.com