

CMPUT 299 (B1) Winter 2008

Security in a Networked World

Network Protocol Vulnerabilities

yannis@cs.ualberta.ca

Authentication in a Networked Environment

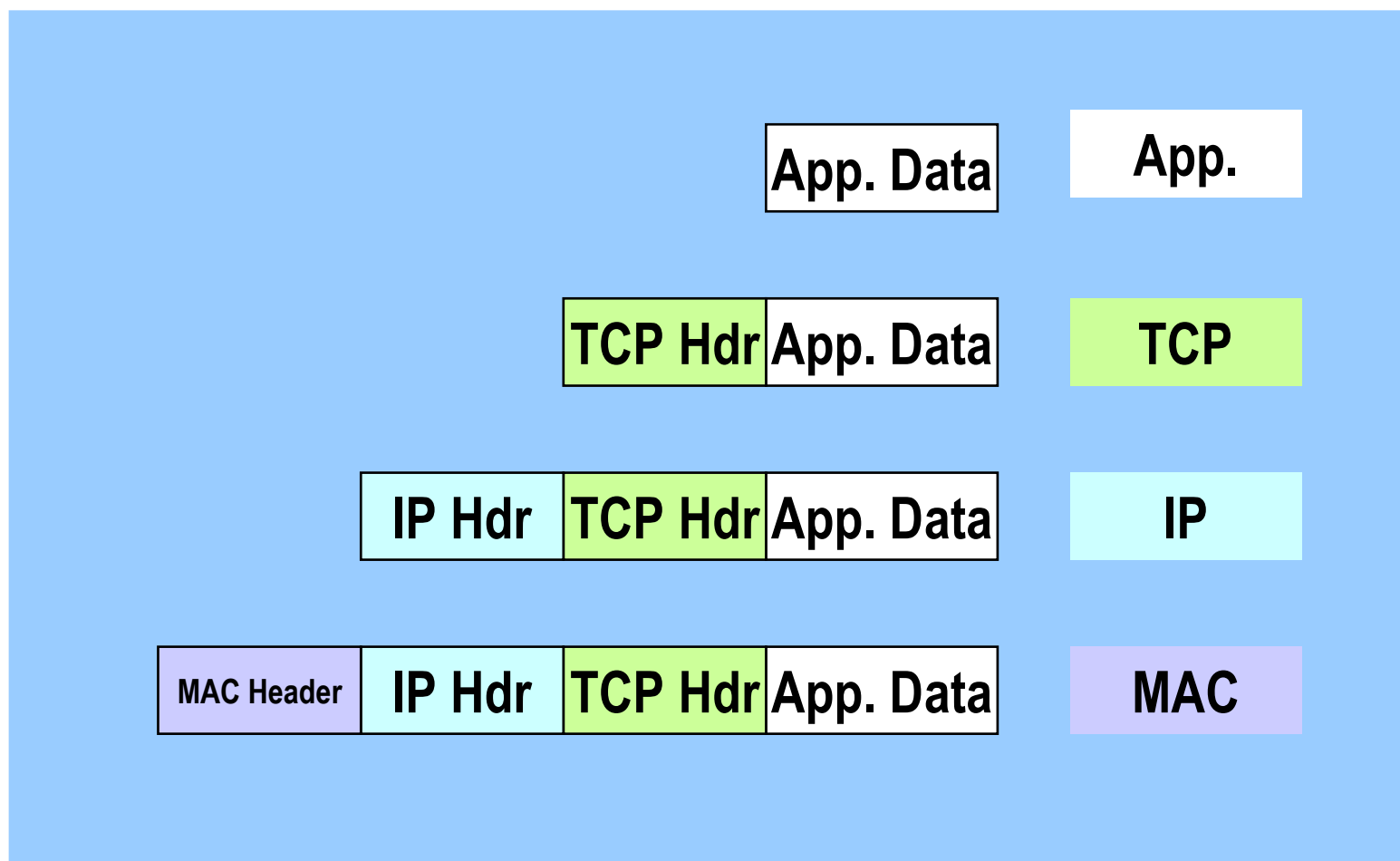
- ▶ Until now we have assumed that users trust the links connecting them with hosts.
 - ▶ Meaningful in the “old days” of terminals physically wired to a mainframe.
- ▶ Today, authentication protocols are executed over a networked infrastructure.
 - ▶ There are plenty of opportunities for attacks at various points in the protocol stack.
 - ▶ The lower in the protocol stack the attack, the potentially more powerful its impact (that is instead of interfering, e.g., with one application, it can subvert an entire host,).

Why are network protocols vulnerable? We need to see how they were designed.

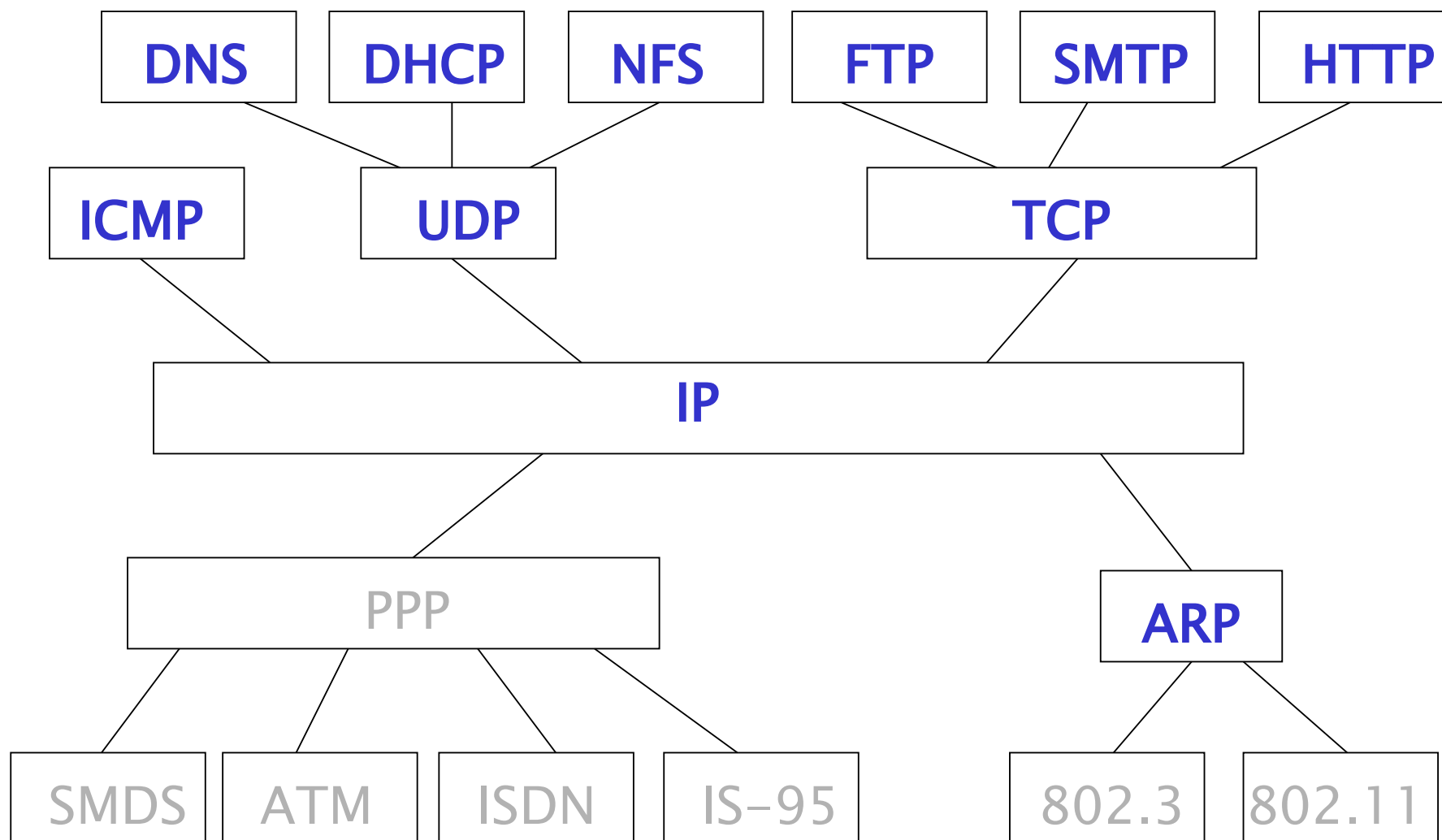
The Protocol Stack

Application	Application
Presentation	
Session	
Transport	TCP
Network	IP
Data Link	Link Access or Host-to-Host or "Network"
Physical	

Data Encapsulation



The TCP/IP Protocol Suite



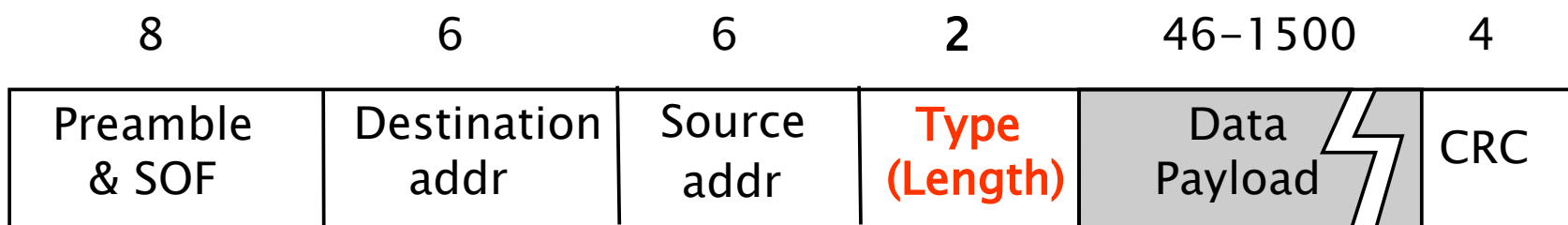
We will not cover attacks to IP yet. We will cover them along with routing vulnerabilities.

MAC Addresses

- ▶ Hyphened Hex Convention: aa-bb-cc-dd-ee-ff
- ▶ Locally Unique Addresses (administrator).
- ▶ Globally Unique Addresses (manufacturer).
- ▶ Multicast Addresses
 - ▶ Destination is all nodes within particular set (group).
- ▶ Broadcast Address
 - ▶ All ffs: ff-ff-ff-ff-ff-ff
- ▶ Multiple Interfaces → Multiple Addresses

MAC addresses are easily spoofed.

MAC Address Format

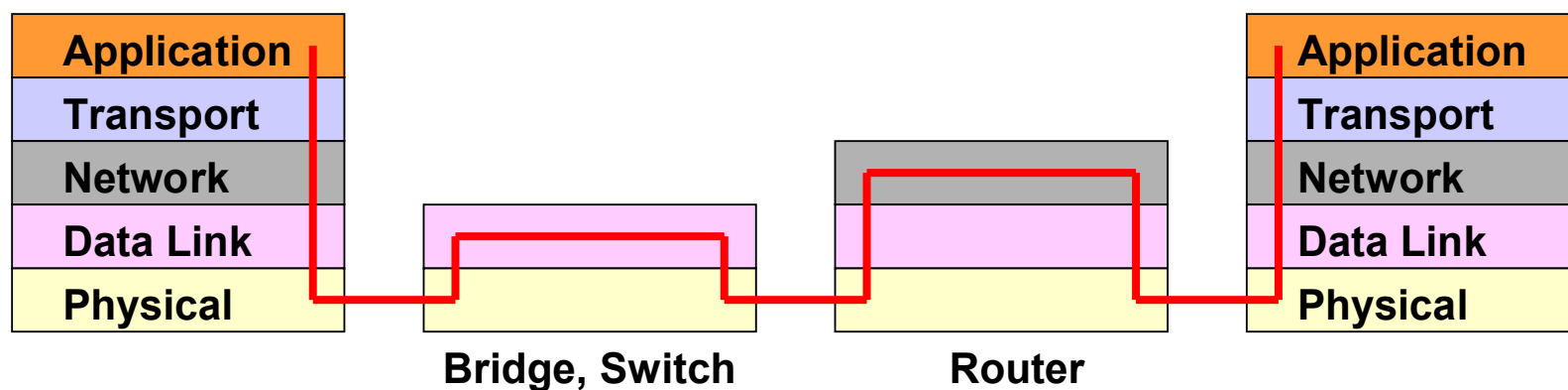


Dec .	Hex .	
0000	0000-05DC	IEEE802.3 Length Field
1536	0600	XEROX NS IDP
	0660	DLOG
	0661	DLOG
2048	0800	Internet IP (IPv4)
2049	0801	X.75 Internet
2050	0802	NBS Internet
2051	0803	ECMA Internet
2052	0804	Chaosnet
2053	0805	X.25 Level 3
2054	0806	ARP
2055	0807	XNS Compatibility

...

From <http://www.iana.org/assignments/ethernet-numbers>

Routers, Bridges, and Switches



From Bridging to Routing

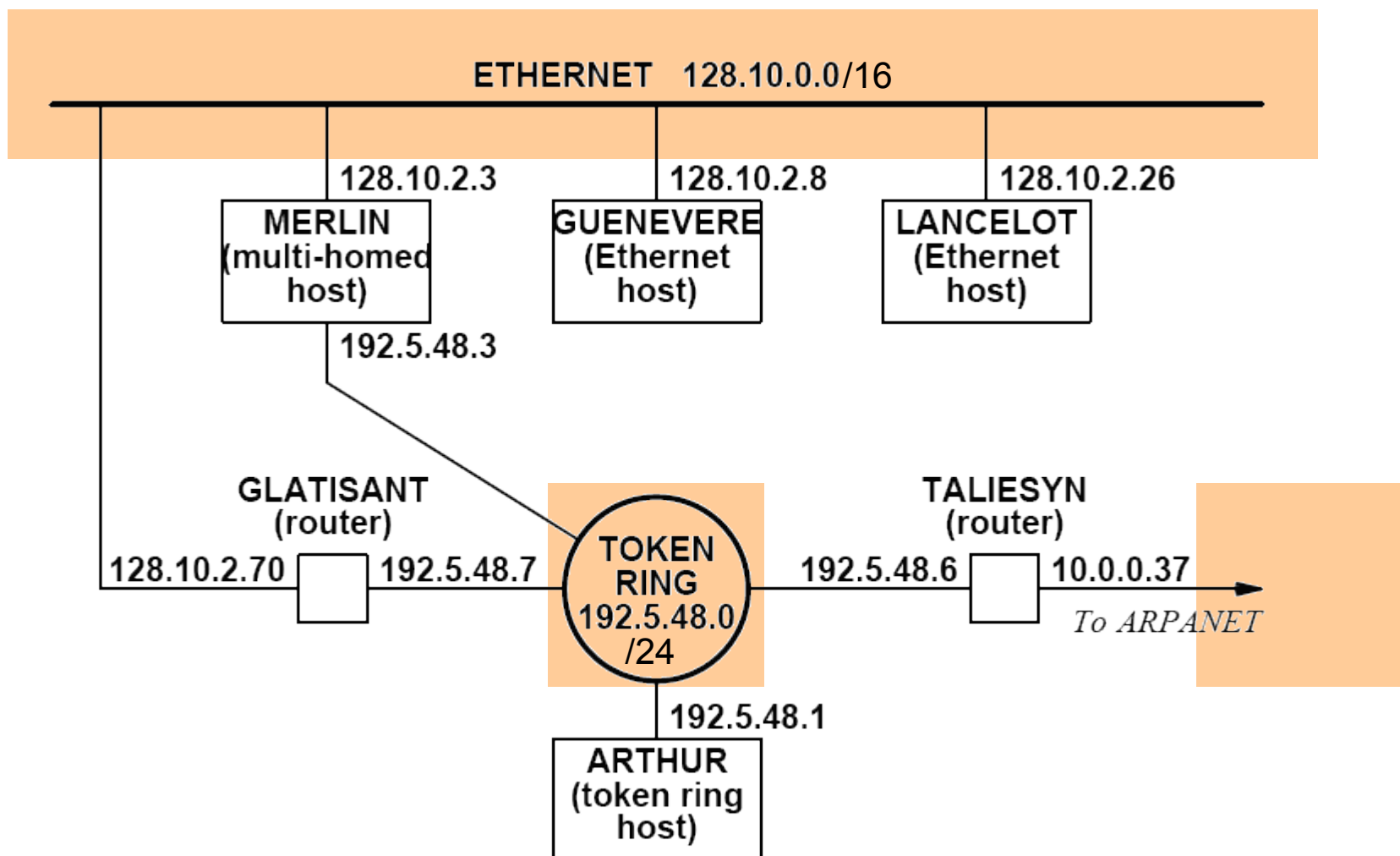
- ▶ Bridging does not scale well to large networks.
 - ▶ Large forwarding tables.
 - ▶ Lots of updates.
- ▶ We introduce routing as a separate, but complementary to, technique to bridging.
- ▶ A usual approach: route in large networks (e.g., the Internet, or medium to large corporate networks) and bridge small LANs.

Addresses for Routing

- ▶ The bit representation of an address must be short (to avoid wasting header space) but also long enough to accommodate the expected maximum number of hosts.
- ▶ In hierarchical addressing (e.g. IPv4) the prefix identifies the network (network id) and the suffix identifies the host (host id).
 - ▶ Short network id field, means a small number of networks can be supported, but each network can have many hosts.
 - ▶ Long prefix, the opposite problem, many networks, but few hosts per network.
 - ▶ Today the prefix is indicated by a variable length “netmask.” (delineates network from host part)
 - ▶ An all 1s host address is a broadcast address

IP Addresses

/16



based on D. Comer, "Internetworking with TCP/IP"

Subnets are identified by their common prefix.

ICMP Redirects

- ▶ Under normal circumstances ICMP redirects are sent to hosts that have forwarded a packet to the “wrong” router (e.g., not with the shortest path to the destination).
- ▶ ICMP redirects are meant to be honored by hosts (they are inconsequential to routers; routers run specific routing algorithms to determine the best path for a given destination).
- ▶ Injecting ICMP Redirects (for a particular destination host) forces a (source) host to send traffic destined for the (destination) host via the gateway indicated in the Redirect. Allowing, e.g., Man-in-the-Middle (MitM) attacks.

ICMP & Remote Broadcasts

- ▶ What if host 128.10.2.8 in the example sends a packet to 192.5.48.255?
- ▶ Suppose we send a large amount of ICMP echo pings to an IP broadcast addresses, while spoofing the source address to be that of a victim.
- ▶ If the router implements the IP broadcast as a layer 2 (Ethernet) broadcast, all hosts in the subnet 192.5.48.0/24 will receive the ICMP echo request and all reply, sending traffic to the victim.
- ▶ This is a so called *smurf* attack (for ICMP echo) or *fraggle* attack (for UDP echo).

Getting from A to B: Bridging

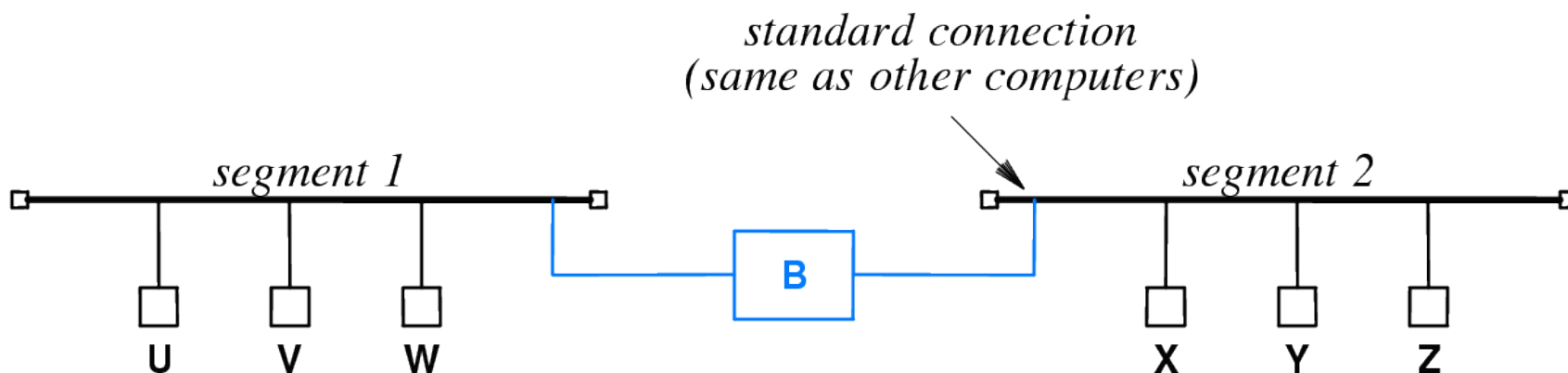
Algorithm:

Receive frame (Source S, Destination D) on port P.
(Check CRC. Drop frame if CRC incorrect.)
Lookup for the port (say L) associated with the address D.
If $P=L$, then drop the frame.
Else, transmit frame on port L.

Lookup:

Given the destination address, perform table lookup to determine the output port (and any other information associated with the destination address).
Options: binary search, sorted table entries, hash table (hash function can be based on CRC), content addressable memory (CAM): usually 48bit key + 16 bit data.

Backward Learning Bridges



Event	Segment 1 List	Segment 2 List
Bridge boots	–	–
U sends to V	U	–
V sends to U	U, V	–
Z broadcasts	U, V	Z
Y sends to V	U, V	Z, Y
Y sends to X	U, V	Z, Y
X sends to W	U, V	Z, Y, X
W sends to Z	U, V, W	Z, Y, X

Note that the forwarding tables are finite (and sometimes small).
Need to evict entries (if not timed out) to insert new ones.

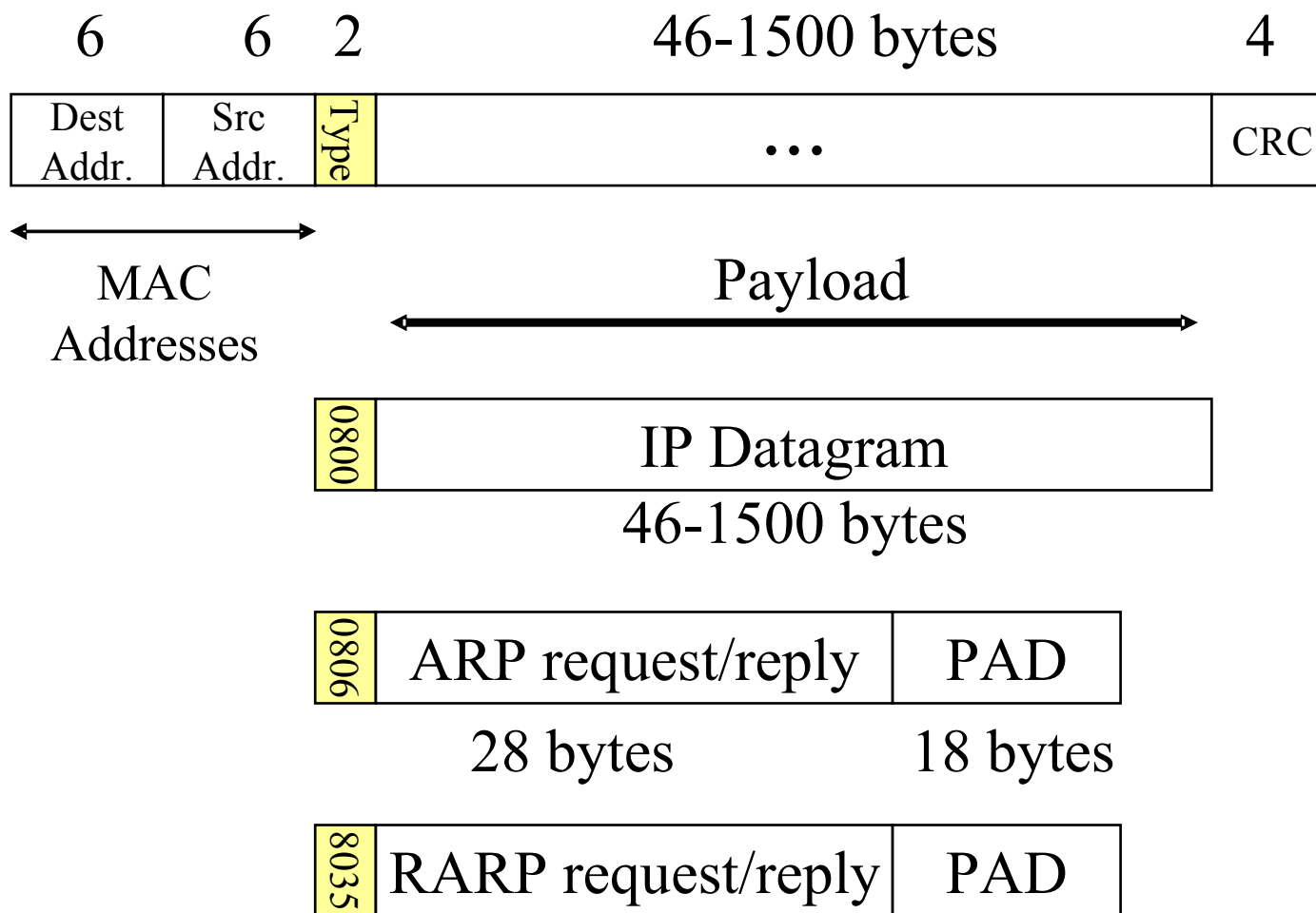
Attacks Against Bridges

- ▶ The objective of the attack is to force the bridge to broadcast traffic instead of sending it to a single segment. (It has to “forget” its table.)
- ▶ The attacker introduces bogus entries (at a high rate) into the forwarding table by sending out frames from spoofed MAC addresses.
- ▶ The table eviction policy will eventually drop entries to make space for the new ones.
- ▶ Some of the evicted entries are of legitimate nodes. The bridge will be forced to broadcast (due to lacking a corresponding forwarding entry).

Address Resolution

- ▶ Inventing a new address scheme comes with its own implications. How do the “new” addresses map to the existing address schemes.
 - ▶ Who/what decides this mapping?
 - ▶ Is the mapping fixed or dynamic?
 - ▶ Can this mapping be automated?
 - ▶ Do these mappings have lifetimes?
 - ▶ Can the mapping be used even if you move?
- ▶ Note: assume two hosts on the same Ethernet LAN wish to communicate with each other. Even if they know their corresponding IP addresses, they have to also know their Ethernet addresses. Anything they send to each other has to be encapsulated in Ethernet frames in order to be transmitted on the shared medium.

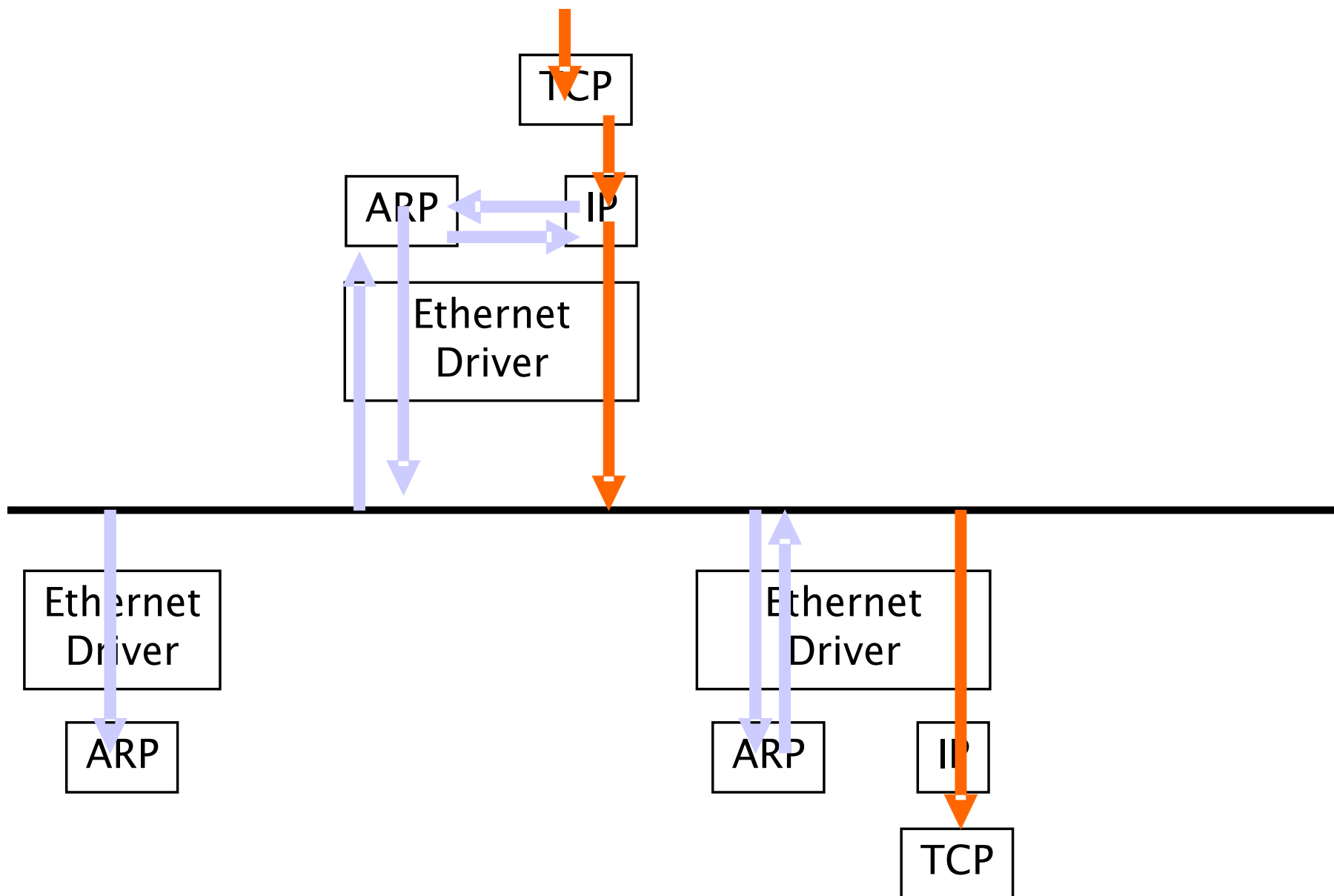
Ethernet Encapsulation (RFC 894)



Address Resolution Protocol (ARP)

- ▶ We do not know the recipients MAC address, although we know its IP address. We need a “binding” between MAC and IP address.
- ▶ **Broadcast** a special control frame, ARP Request. (“Who has X.Y.Z.W as its IP address?”)
- ▶ Only the host configured to “own” the IP address “X.Y.Z.W” is supposed to send an ARP Reply message.
- ▶ The requesting node (and anyone else listening) can now associate the “X.Y.Z.W” with the proper MAC address (e.g., making a table entry).
- ▶ [After a period of time has elapsed, the entry is erased. If needed, it can be re-discovered later in the same way.]

ARP Operation

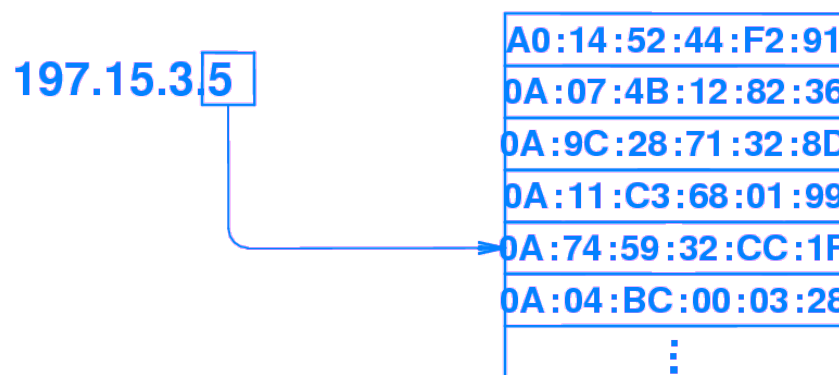


ARP Features

- ▶ The request is sent to a broadcast address. The response is sent as a unicast frame.
- ▶ Conceivably a host might be snooping to receive responses in order to populate its own tables but there is little value to doing so without having the need to communicate with the hosts whose bindings you know. *However a node is obliged to listen at least to the requests.*
- ▶ Note that the ARP messages are unreliable and in addition they are an open security liability (anyone can basically respond to the request – need not be the owner of the IP address being inquired).
- ▶ A responder to a request caches the sender's binding (it is in the payload of the request message) for future use. It is to be expected that if someone is going to send IP packets to us, we may (very soon) be sending packets back to them.

Implementing ARP

- ▶ If the network is small enough (a dozen or so hosts), then even sequential search of a table could be acceptable.
- ▶ In larger networks a hash table can be built to map IP to a hash bucket where the hardware address is stored.
- ▶ The structure of the IP addresses is sometimes exploited: all hosts in a network share the same prefix. Hence, use the suffix to index a table.



Proxy ARP

- ▶ The host replying to the ARP may be a different host from the one actually owning the IP address for which the ARP Request was issued.
 - ▶ But it is a security liability because it allows one to intercept all traffic intended for another host.
- ▶ Why allow it then? Routers routinely use Proxy ARP to convince nodes in the shared network that a host is present in the current broadcast segment of a large network, while in reality forwarding the traffic to a remote location.
 - ▶ Helpful in some corporate LANs where one subnet is split physically across two or more locations.

Gratuitous ARP

- ▶ A Gratuitous ARP is in essence an ARP **request** sent by a host where the address being inquired about is its own.
 - ▶ Other hosts treat this as a legitimate request because it was sent as an Ethernet broadcast.
 - ▶ Even though no host can reply to this request, the hosts are still going to observe the binding of the sender's Ethernet address and the sender's IP address. This is what the sender wanted to accomplish.
 - ▶ A host is obliged to update its ARP table entry (for a particular binding it already possesses) if it receives a new binding for a binding it currently has in its table.
- ▶ Why allow Gratuitous ARP: Failover
 - ▶ A host, say B, can, upon detecting the absence of another host, say A, (e.g. a failed server) send a “Gratuitous ARP” that presents B’s MAC address as the legitimate MAC associated with A’s IP. All the traffic intended for A will be intercepted by B. Conceivably B is a standby backup for A.

TCP

- ▶ A TCP “connection” is defined by the 4-tuple
<localhost, localport, remotehost, remoteport>
- ▶ A server “listens” to localport, i.e., a server waits on <localhost, localport, *, *> until a remote client sends a TCP SYN.
- ▶ It used to be the case that port numbers less than 1024 were considered “privileged” ports, and still survives as restrictions on what privileges are needed by a process to bind to low numbered ports.

TCP Open

Client States**Messages****Server States**

Active open

SYN,CSEQ

SYN,ACK(CSEQ+1),SSEQ

ACK(SSEQ+1),CSEQ+1

Half-opened

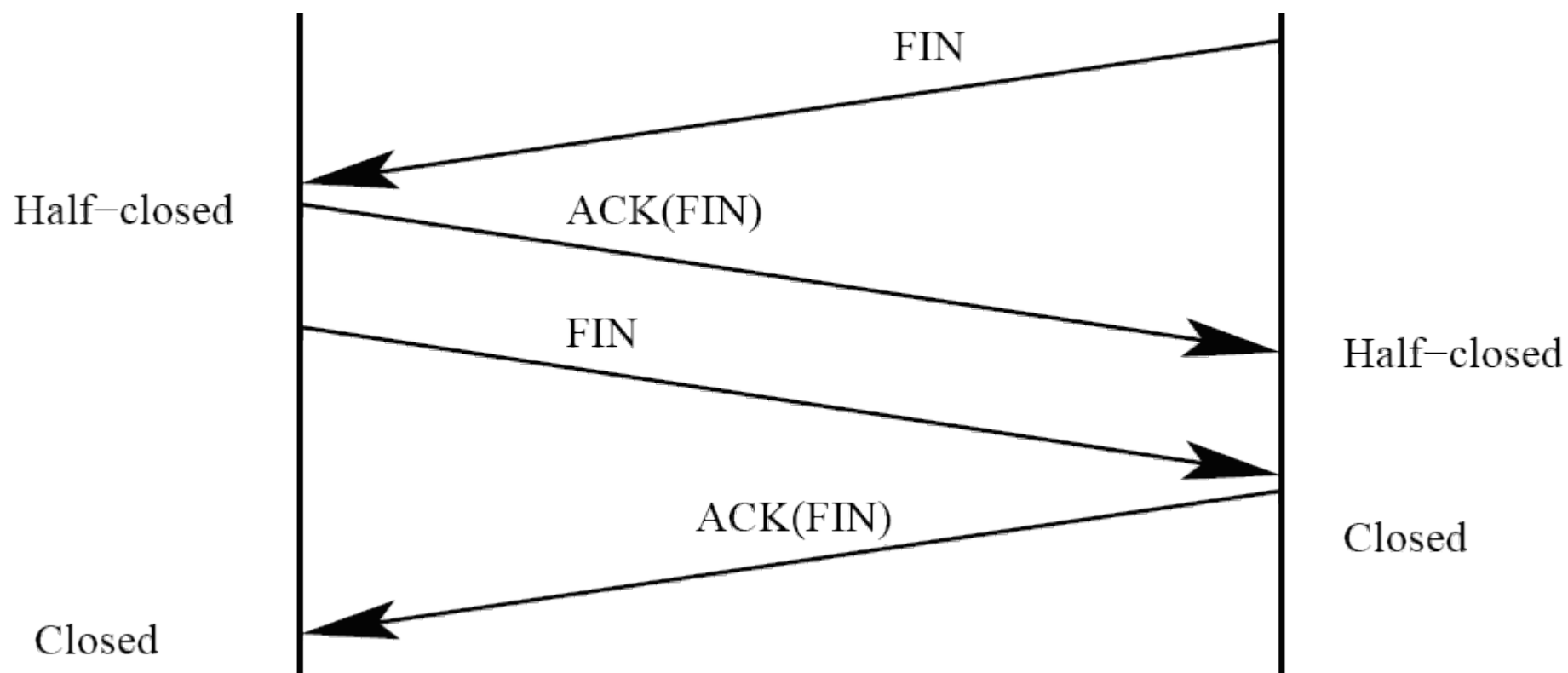
Connection
established

```
roc.3985 > coot.telnet: S 2131328000:2131328000(0) win 4096
coot.telnet > roc.3985: S 1925568000:1925568000(0) ack 2131328001 win 4096
roc.3985 > coot.telnet: . ack 1 win 4096
```

TCP Attacks

- ▶ Send SYN but do not follow up with the rest of the handshake (TCP SYN floods). A DoS attack.
- ▶ Send SYN to probe whether a service is available. (Then use a service-specific attack.)
- ▶ The Initial Sequence Number (ISN) is predictable in certain cases, allowing a host that impersonates another to send traffic to a target host (without necessarily getting traffic back from the target) exploiting other weaknesses through the traffic it gets to send.
 - ▶ A “truly random” ISN solves the problem.

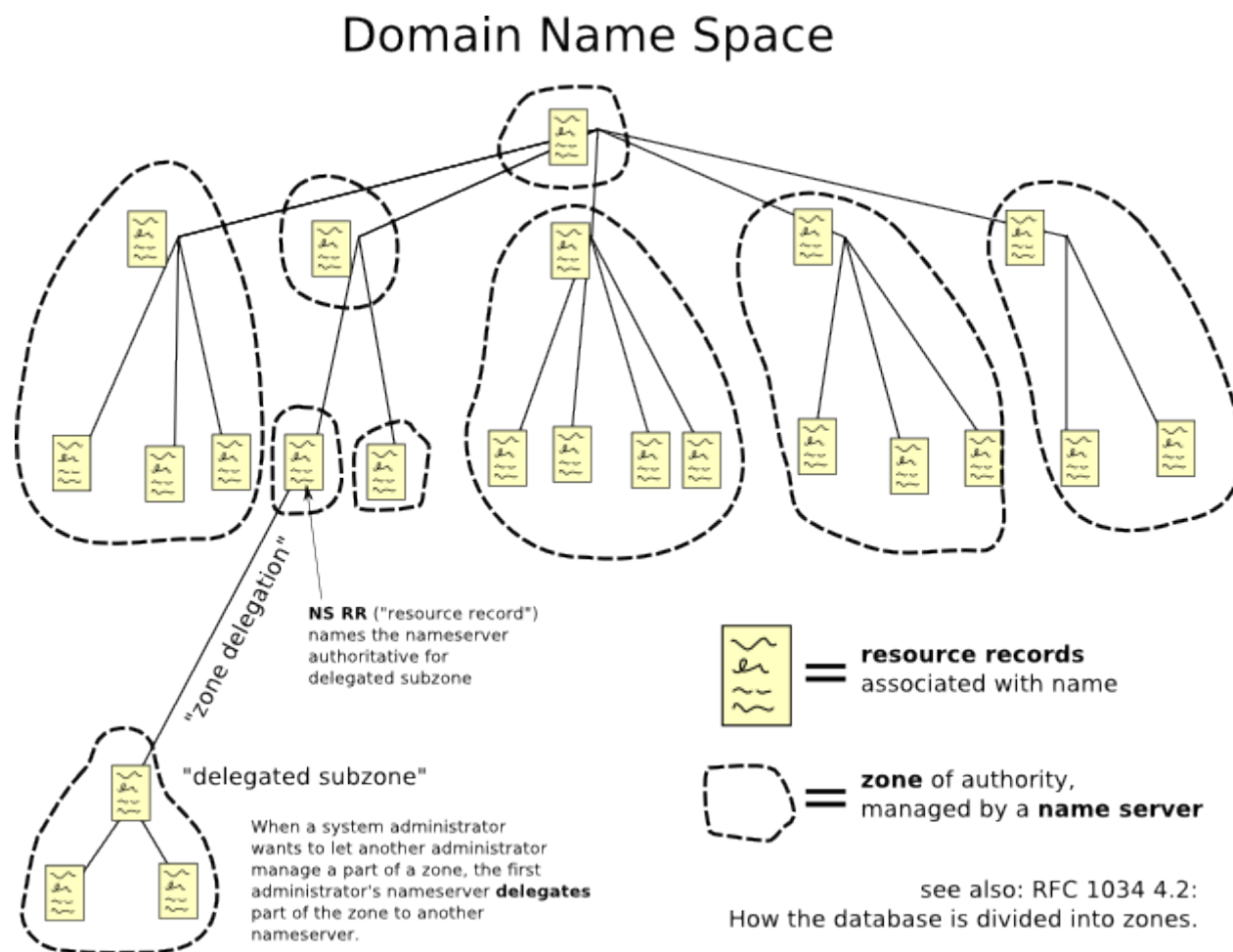
TCP's Full Duplex Nature



Note that the TCP connection termination is an asymmetric process. One endpoint can close while the other endpoints continues to be active.

Domain Name System (DNS)

- ▶ DNS is a distributed database for mapping human-readable hostnames to IP addresses. Responses are cached for efficiency.
- ▶ Hierarchical arrangement of (sub)domains, and likewise hierarchical nameserver arrangement.



DNS Vulnerabilities

- ▶ DNS cache poisoning, in which an attacker provides mapping information before the queried name server does (usually the first response to arrive is accepted). The attack involves prediction of PRNG (again!) values, i.e., predicting the sent “nonce” and preparation of many response messages (one being successful is enough).
- ▶ The lifetime record of an entry might have long-term effects (faulty resolutions can remain alive for a long period of time after implanted).
- ▶ (Not a DNS vulnerability per-se) but Homographs are easy to miss for human users (0 for an O, or S for 5), a technique exploited mainly by *phishing*.
- ▶ Attempts for authentication of DNS responses have started relatively recently. Scalability is an issue.

DNS Vulnerabilities (cont'd)

- ▶ Attacks are more potent when the attackers have taken control of a name server. (Can even be the result of fraud / social engineering of convincing a registration authority that the attacker is indeed in charge of sub-domain.)
- ▶ Request an attacked name server to resolve via a nameserver already in the control of the attacker. The name server receives not only the resolved name, but also additional (fake) resolution records that are “volunteered”. This is because name servers can “volunteer” resolution records for other domains (not just the one they were queried about).

DHCP (BOOTP)

- ▶ DHCP provides a means to obtain an IP address upon boot, as well as additional info, e.g., default gateway, DNS servers etc.
 - ▶ Queries are unauthenticated and responses can be subjected to MitM attacks as well as DoS attacks.
 - ▶ The attacker can set up a bogus DHCP server (however it does not result in appreciable advantage assuming one can already perform ARP-spoofing).
 - ▶ DHCP is a “weak spot,” and many networks will provide a DHCP response, but subsequently require authentication for any further access
 - ▶ It is the norm to find wireless access in public places where DHCP is provided but transport-layer sessions require further authentication (possibly payment, etc.)