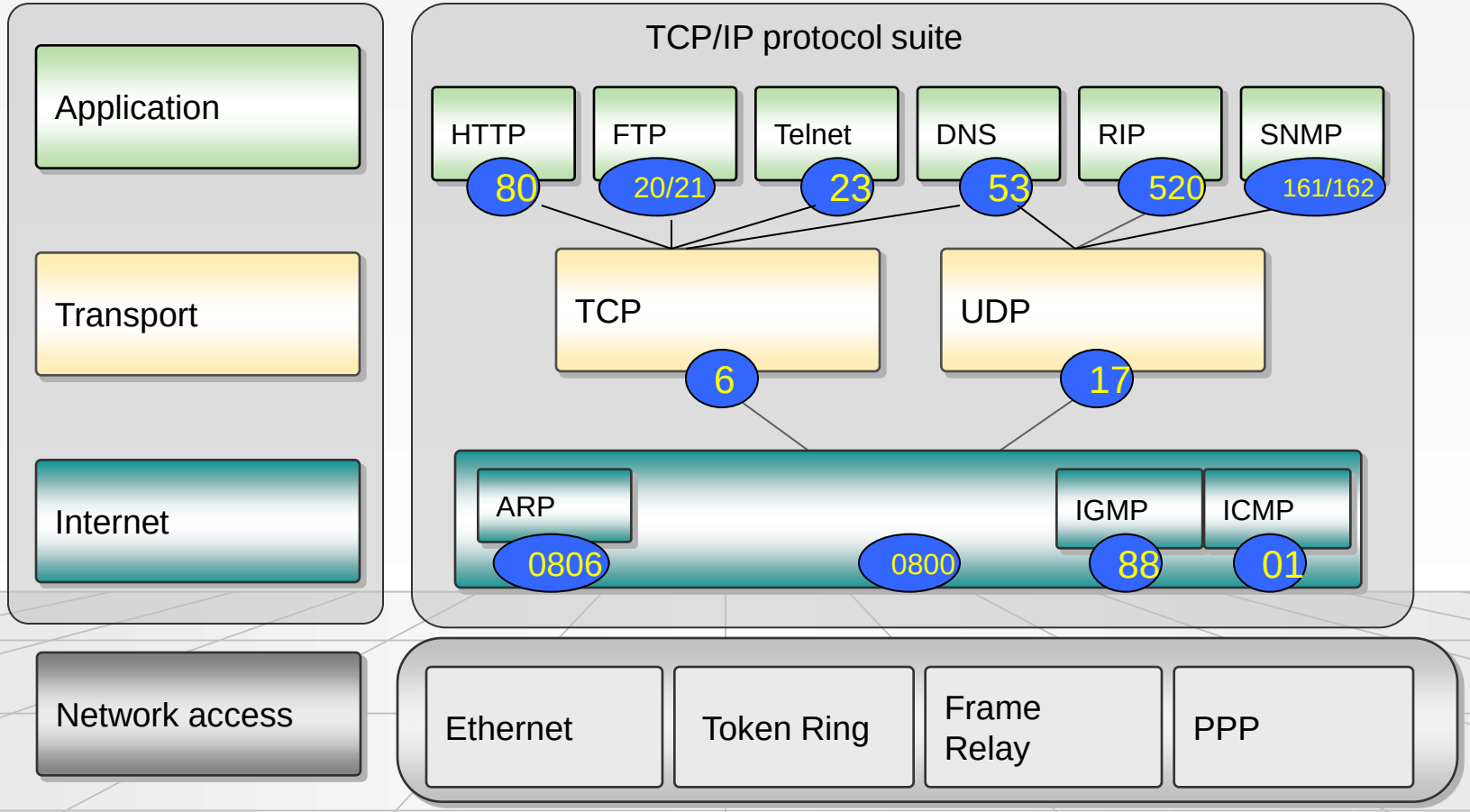
Data Networks

ISOTDAQ 2012

Enrico.Bonaccorsi@cern.ch

TCP/IP Protocols



Networks

- Networks connect s two or more computers
  - Computers can be located anywhere
- Networks can be categorized depending on the size:
  - Local Area Network
  - Metropolitan Area Network
  - Wide Area Network

Protocols

- A protocol defines the syntax, semantics, and synchronization of communication
- the specified behavior is typically independent of how it is to be implemented
- A protocol can therefore be implemented as hardware or software or both

History and Standards

- 1969: ARPANET is commisioned by the USA defense department for reasearch into networking
- 1972: First e-mail program written
  - Telnet is specified
- 1973: Ethernet is outlined
  - FTP is specified
- 1974: TCP is specified
- 1976: First email sent by Queen Elizabeth
- 1977: Number of hosts breaks 100
- 1981: IP Standard is published in RFC 791
- 1983: TCP/IP becomes the core Internet Protocol
- 1984: DNS is specified

RFCs

- http://www.ietf.org/
  - Large, open, international community of network designers, operators, vendors and researchers concerned with the evolution of the Internet Architecture
  - Open to any interested individual
- Network managers will readily agree that networks need documentation.
- RFCs (Request for Comments) document the functions of the Internet and the protocols that support it
- The documentation process start with the Internet Draft

ISO/OSI

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical

Network Components

- Hosts
- Hubs
- Switches
- Routers

Local Signaling

- Data formatting
- Session Handling
- Routing
- **Local Signaling**

- Local Signialing is not part of the TCP/IP family of protocols, but an IP datagram requires a physical interface to get to the target station
- Most popular LAN protocol in use today is Ethernet

Ethernet

Preamble	Physical Dest. Address	Physical Source Address	Type	Data/Payload	CRC
8Bytes	6B	6B	6B	46-1500B	4B

- Destination address:
  - UNICAST: *00 10 A4* BA 87 5B
  - MULTICAST: **01 00 5E** 00 00 09
  - BROADCAST: FF FF FF FF FF FF

Internet Protocol

- The IP network moves datagrams with the same functionality that the Postal Service delivers letters.
- An IP datagram is placed on the network by the source host.
  - Letters are deposited in the mailbox by the mailer
- The IP network tries to deliver the datagrams, if the necessary physical and logical connections exists
  - The postal service tries to deliver the letter if the right trucks, planes, buses, and mail personal exists
- IP is connectionless and not reliable
  - Just as the postal service, IP make no guarantee of delivery

IP Addressing

IP Address:

- Are 32-bits long
- Uniquely identify a particular network interface
- Contain two parts
  - **Network ID or prefix**
  - *Locally administered bits*
  - **137.138**.*111.12*

Reserved Address

- 0.0.0.0
  - The “unknown” address
- 127.0.0.1
  - The loopback address
- 255.255.255.255
  - The local broadcast address
- (all local bits off)
  - Our local network
- (all local bits on)
  - The broadcast address for our local network

Private Addressing

- Can be used by anyone, anywhere
- Not routable on the global Internet
- Three blocks allocated by RFC 1918
  - 10.0.0.0/8: About 16 million addresses
  - 172.16.0.0/12: About 1 million addresses
  - 192.168.0.0/16: 65.536 addresses

Address Resolution Protocol

IP address: 10.10.10.10 →

→ MAC address 00 00 0c 00 23 23

When any system in an IP network begins the process of communicating with another system, a key part of the process is to identify the MAC address that matches the target IP address

ARP Cache

```
[lxplus427] ~ $ /sbin/arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
137.138.210.193          ether   0A:00:30:89:D2:C1   C                     eth0
137.138.210.211          ether   00:30:48:F0:DF:BC   C                     eth0
137.138.210.238          ether   00:30:48:F0:E7:CA   C                     eth0
[lxplus427] ~ $
```

ARP Restrictions

- ARP uses the network broadcast address to find the hardware address of the target host, which is only a concern when the two hosts share the same network and subnet.
- Since routers block broadcasts, the ARP requests never leaves the subnet

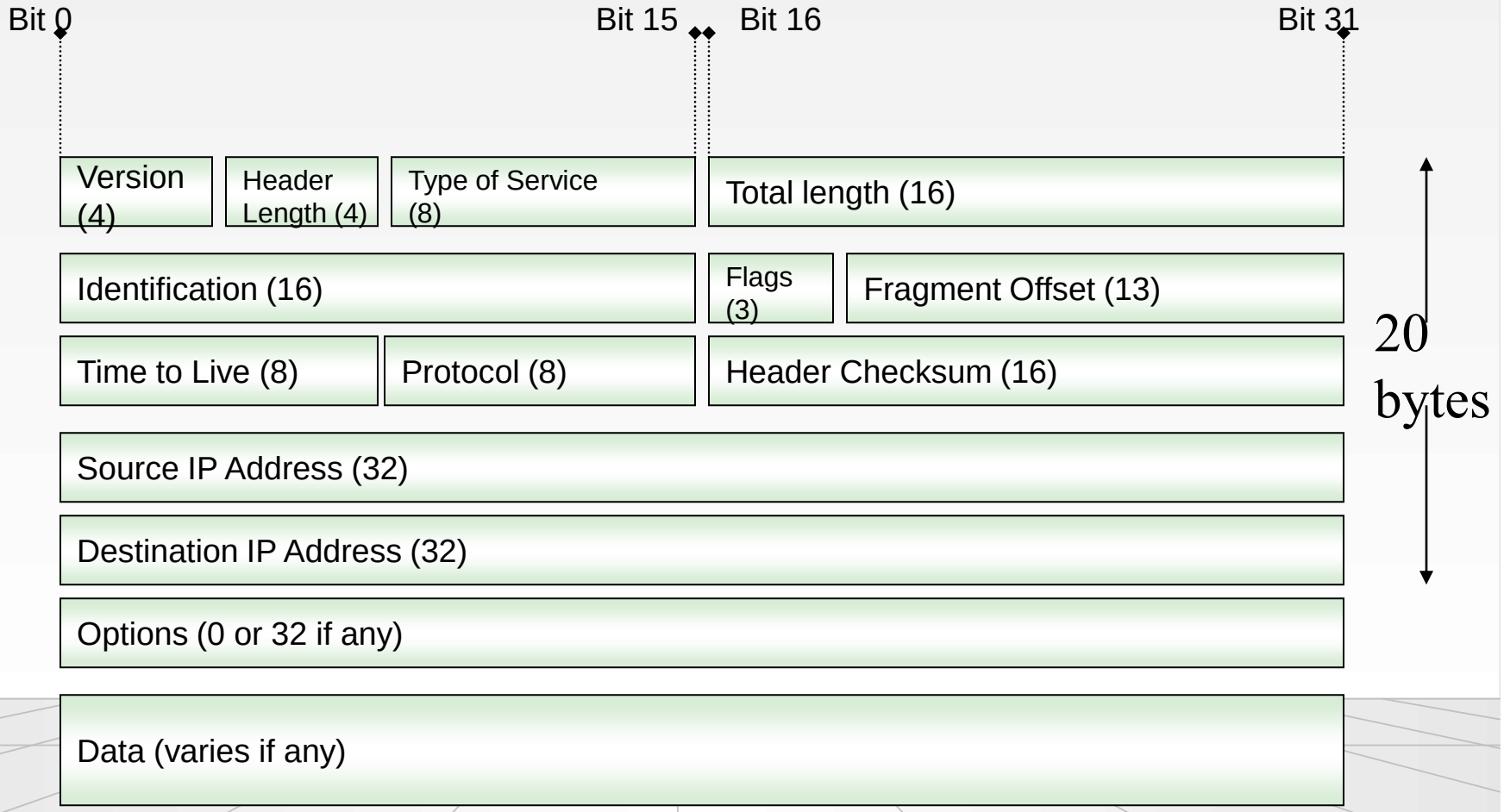
ARP Message Fields

0	15	16	31
Hardware Type (16)		Protocol Type (16)	
Hardware Lenght (8)	Protocol Lenght (8)	Operation (16)	
Source Hardware Address – Sender MAC address			
Source Protocol Address – Sender IP address			
Target Hardware Address – Receiver MAC address			
Target Protocol Address – Receiver IP address			

Prefix Notation

Prefix	Mask	Prefix	Mask	Prefix	Mask
/0	0.0.0.0	/11	255.224.0.0	/22	255.255.252.0
/1	128.0.0.0	/12	255.240.0.0	/23	255.255.254.0
/2	192.0.0.0	/13	255.248.0.0	/24	255.255.255.255
/3	224.0.0.0	/14	255.252.0.0	/25	255.255.255.128
/4	240.0.0.0	/15	255.254.0.0	/26	255.255.255.192
/5	248.0.0.0	/16	255.255.0.0	/27	255.255.255.224
/6	252.0.0.0	/17	255.255.128.0	/28	255.255.255.240
/7	254.0.0.0	/18	255.255.192.0	/29	255.255.255.248
/8	255.0.0.0	/19	255.255.224.0	/30	255.255.255.252
/9	255.128.0.0	/20	255.255.240.0	/31	255.255.255.254
/10	255.192.0.0	/21	255.255.248.0	/32	255.255.255.255

IP datagrams



- Header must be at least 20 bytes
- Header can increase in multiples of 4 bytes

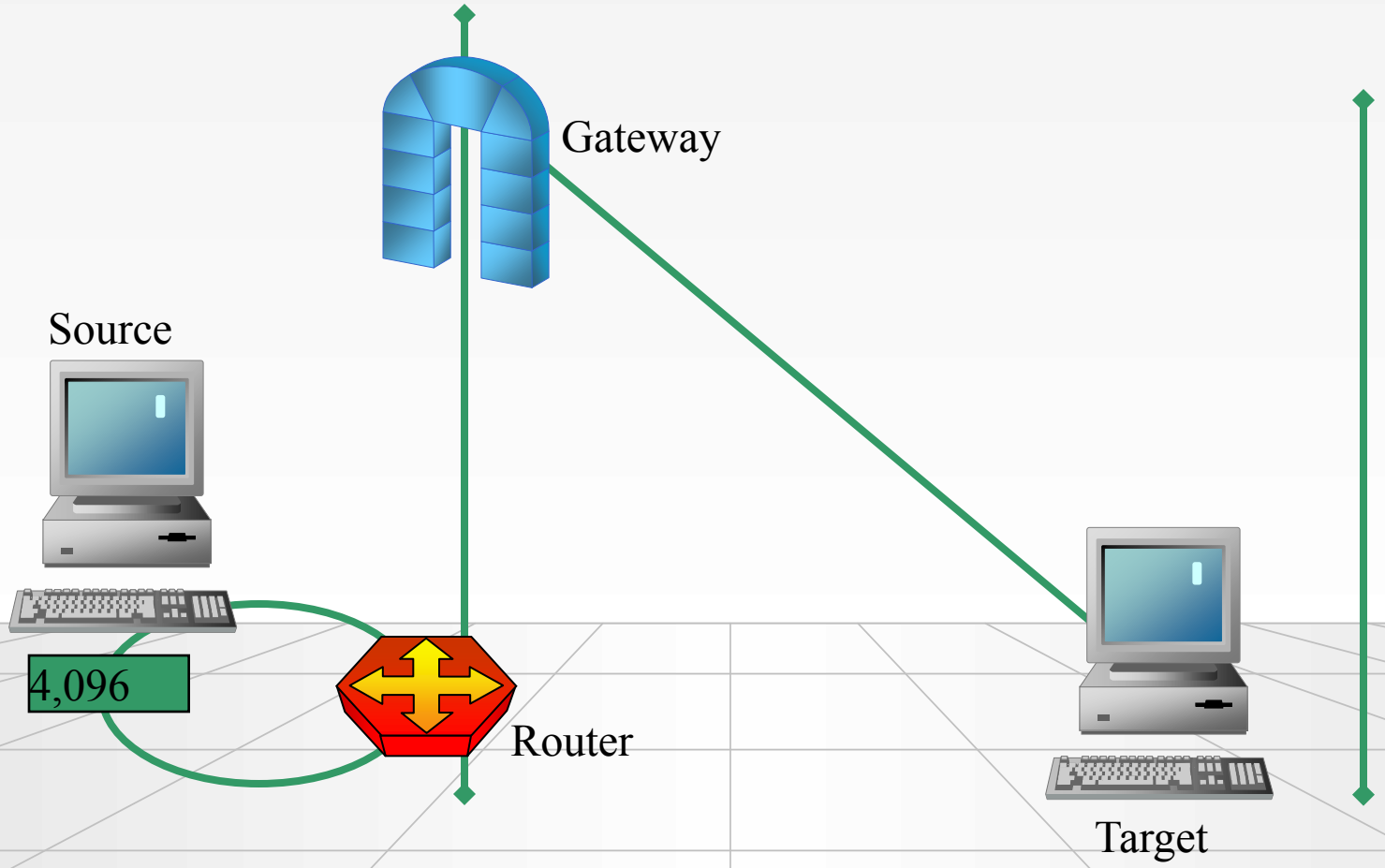
Fragment Bytes Layout

Fragmentation Field Layout (16 Bits)

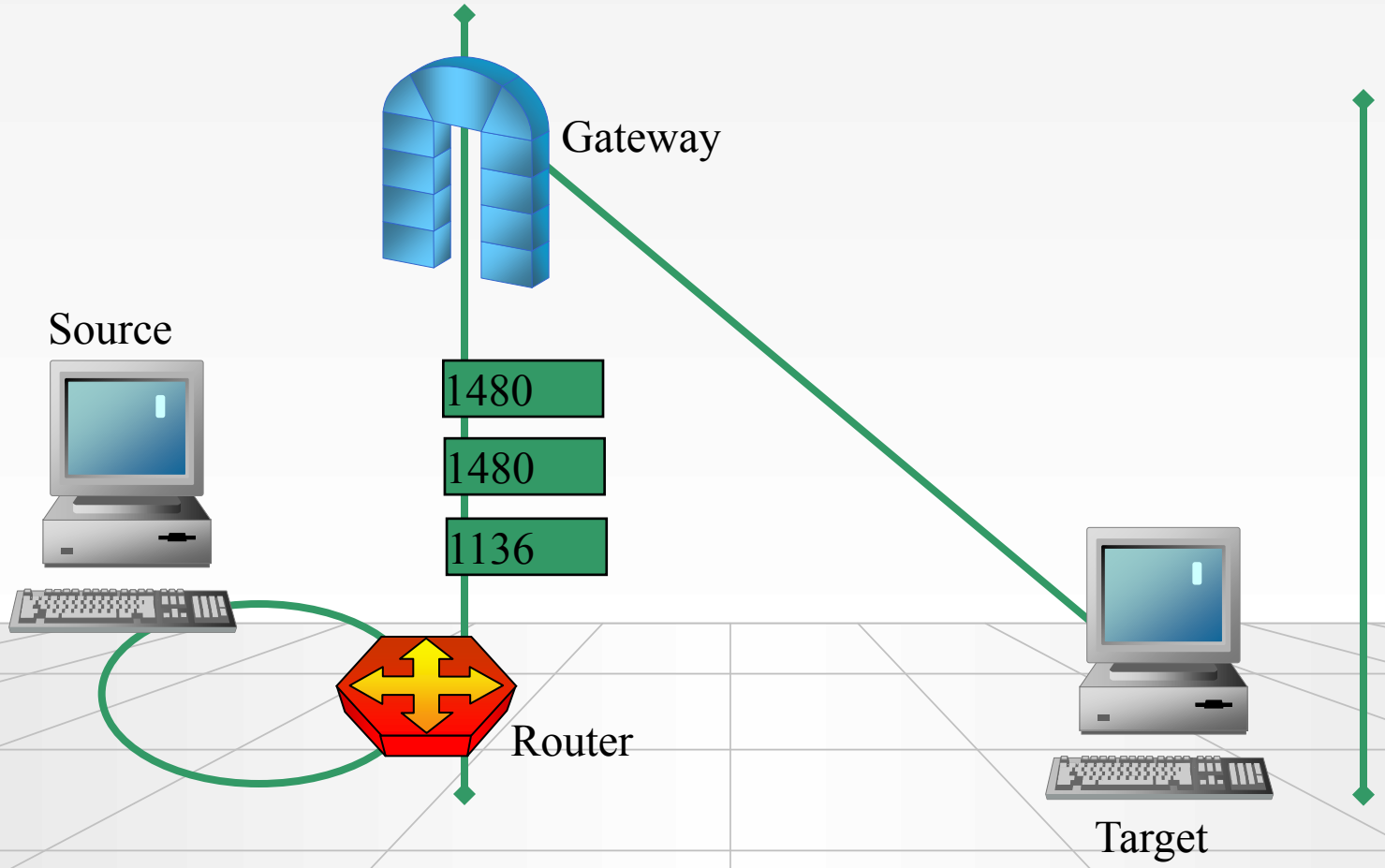
Reserved	Don't Fragment	More Fragments	Fragment Offset (13 bits)

If the "More" bit is	And the Offset is	Then the datagram is
0	0	Not fragmented
1	0	The first fragment
1	> 0	A middle fragment
0	> 0	The last fragment

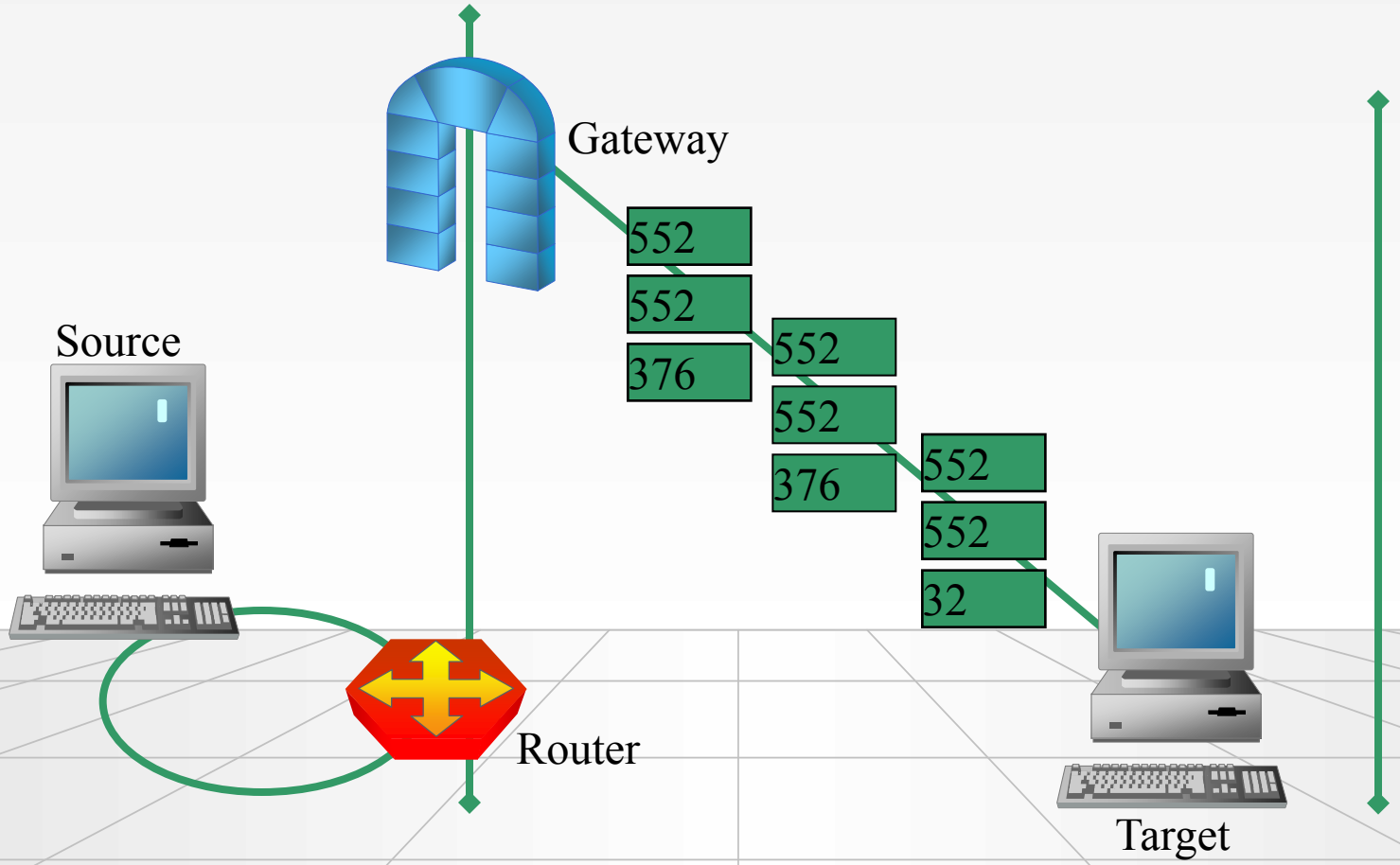
Fragmenting Fragments



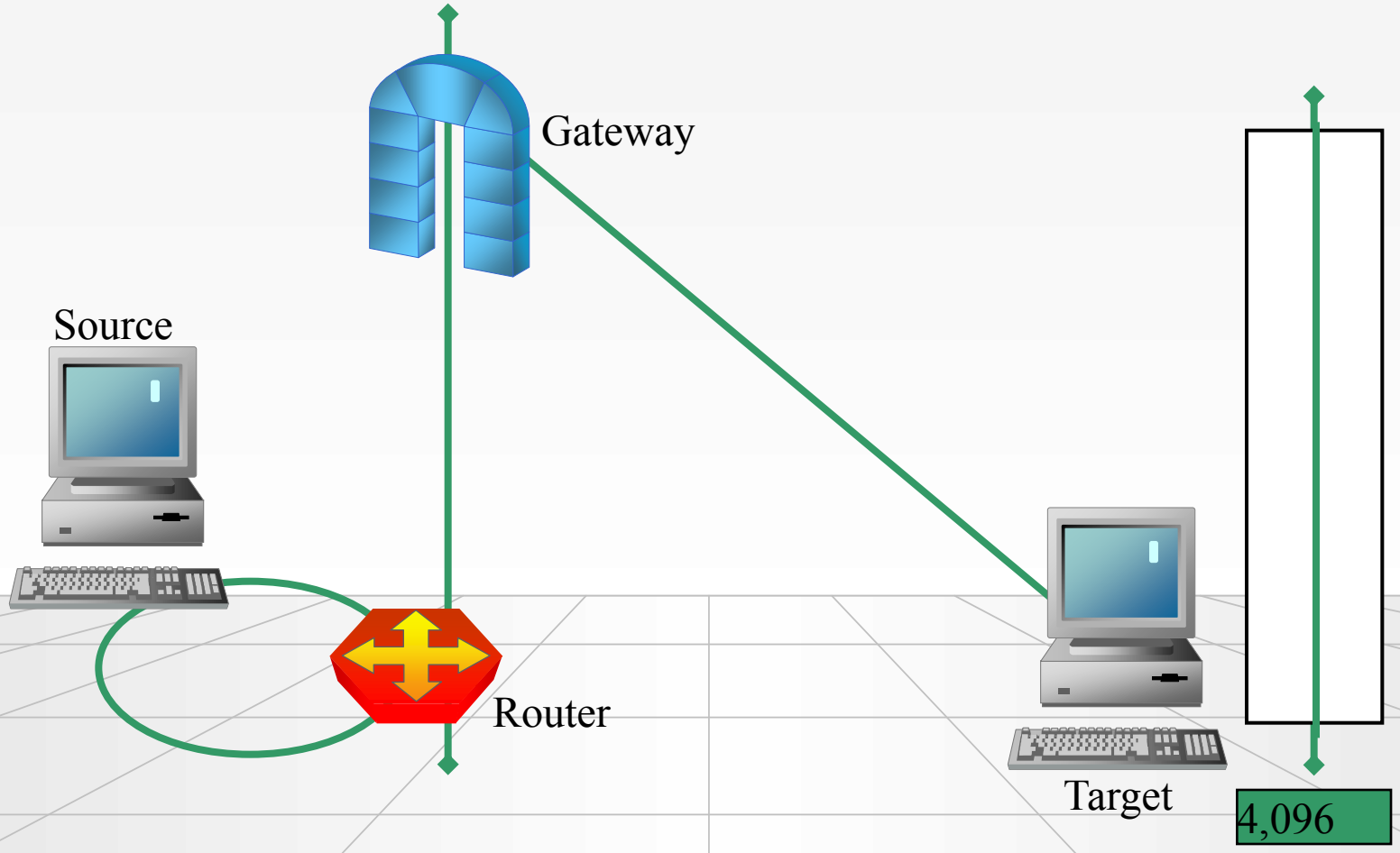
Fragmenting Fragments



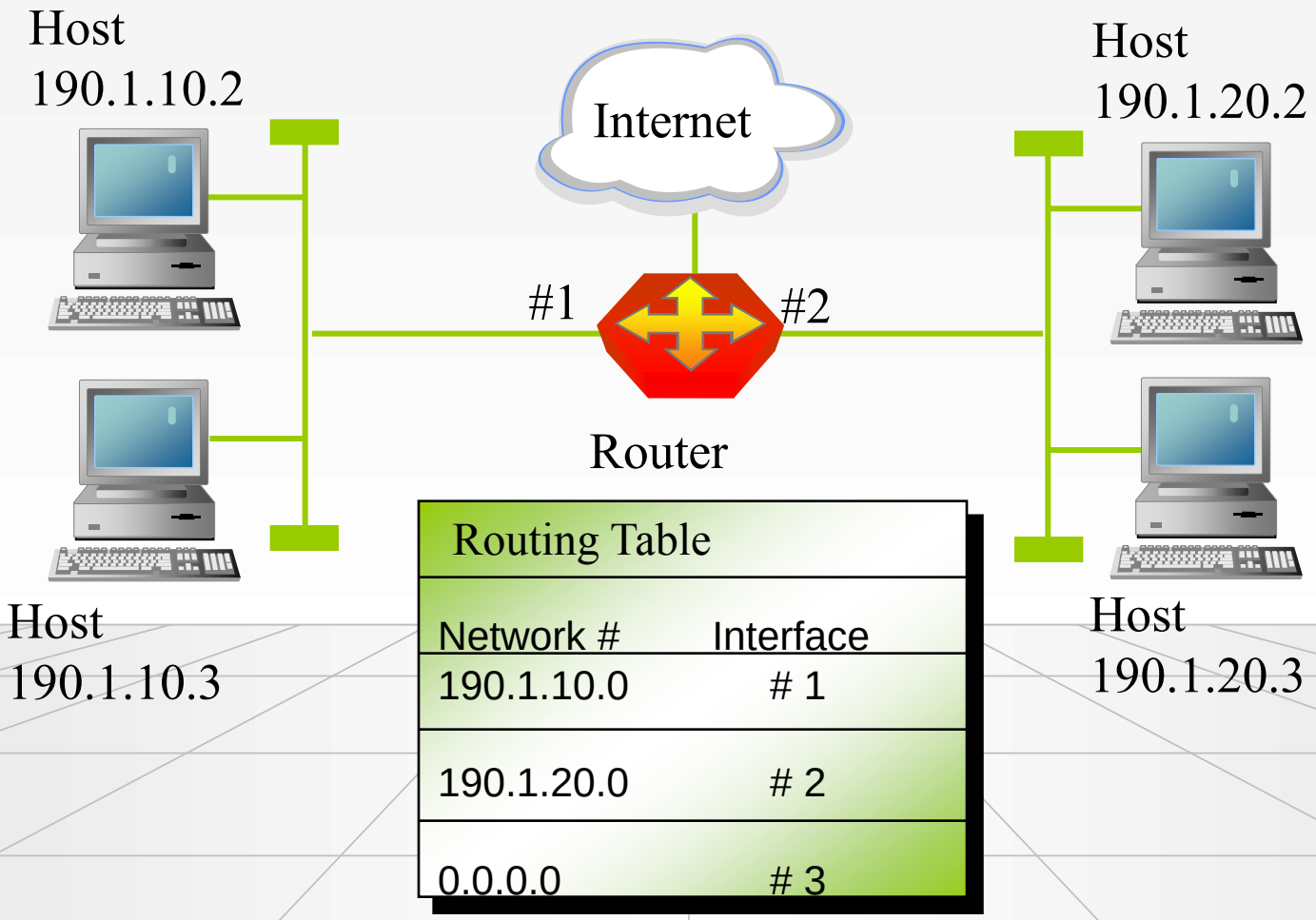
Fragmenting Fragments



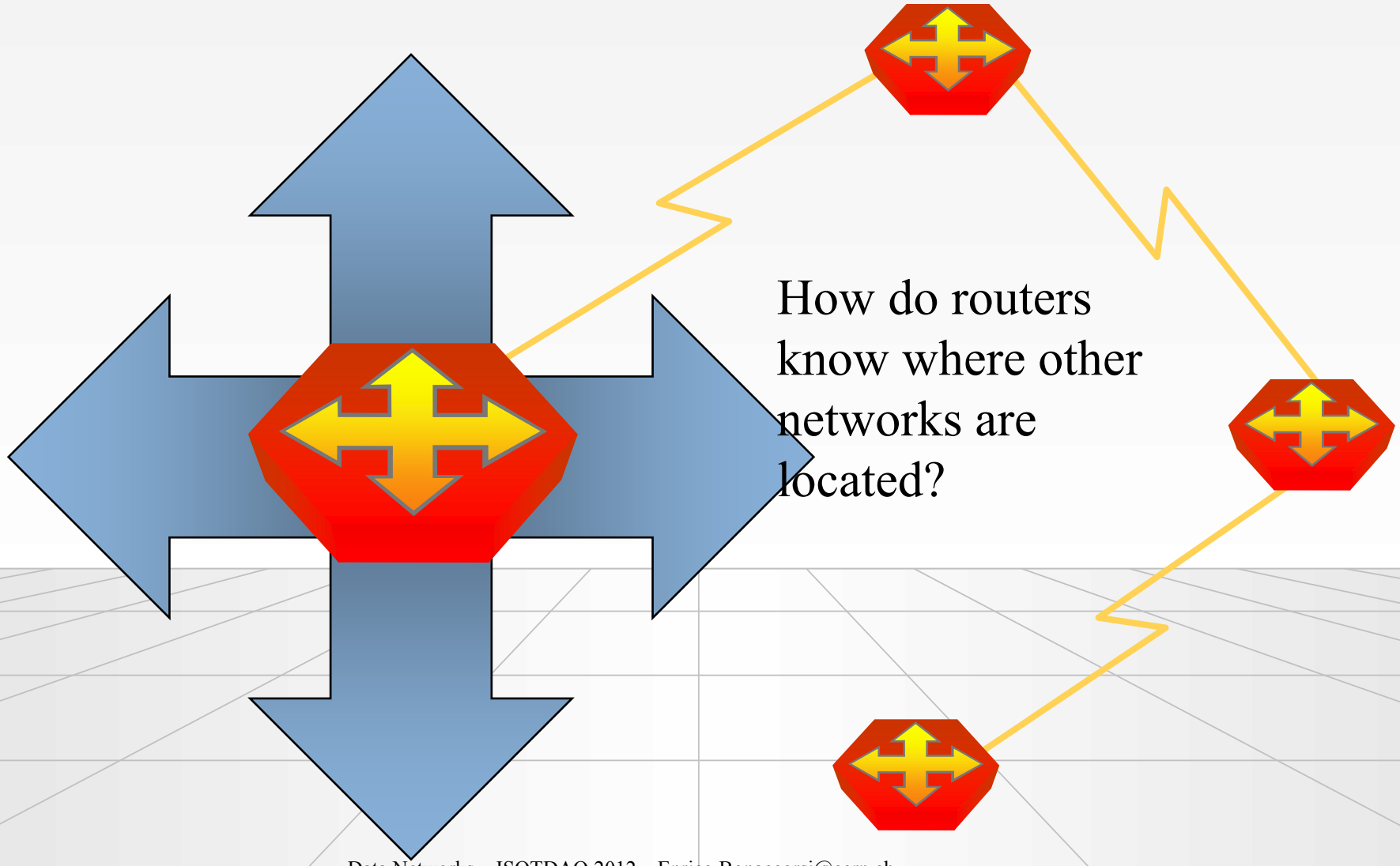
Fragmenting Fragments



What Is IP Routing?



Routing Function



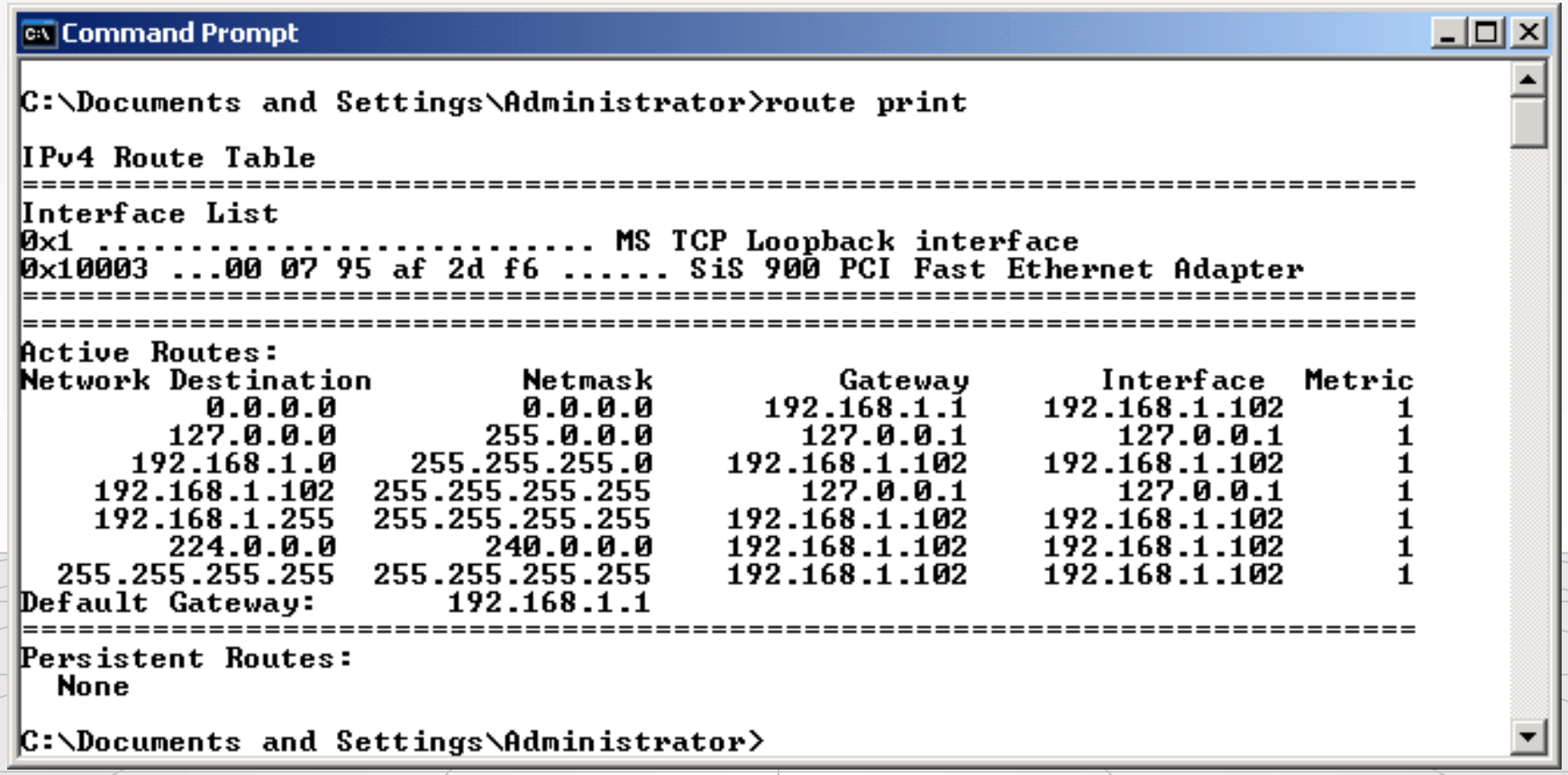
How do routers know where other networks are located?

Routing Table Basics

Most routing tables contain:

- Destination network address
- Subnet mask
- Cost or metric
- Next hop address or gateway address
- Exit interface

Routing Table Contents



Router Routing Table

Destination	Mask	Protocol	Age	Cost	Next Hop	Interface
150.7.0.0	255.255.0.0	RIP	22	4	200.1.2.3	1
150.8.0.0	255.255.0.0	RIP	13	2	200.1.5.3	2
191.153.66.0	255.255.255.0	LOCAL	3086	0	191.153.66.10	3
191.153.77.0	255.255.255.0	LOCAL	2246	0	191.153.77.10	4
191.153.88.0	255.255.255.0	LOCAL	1136	0	191.153.88.10	5
200.1.1.0	255.255.255.0	RIP	22	3	200.1.2.3	1
200.1.2.0	255.255.255.0	LOCAL	5002	0	200.1.2.1	1
200.1.5.0	255.255.255.0	LOCAL	5016	0	200.1.5.1	2

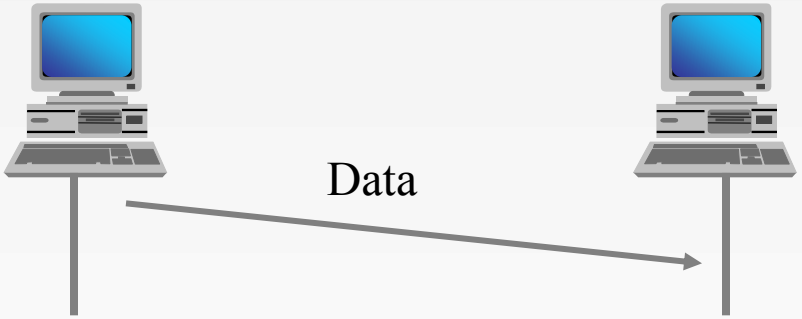
Reliable Transport Services

TCP is connection-oriented and reliable.

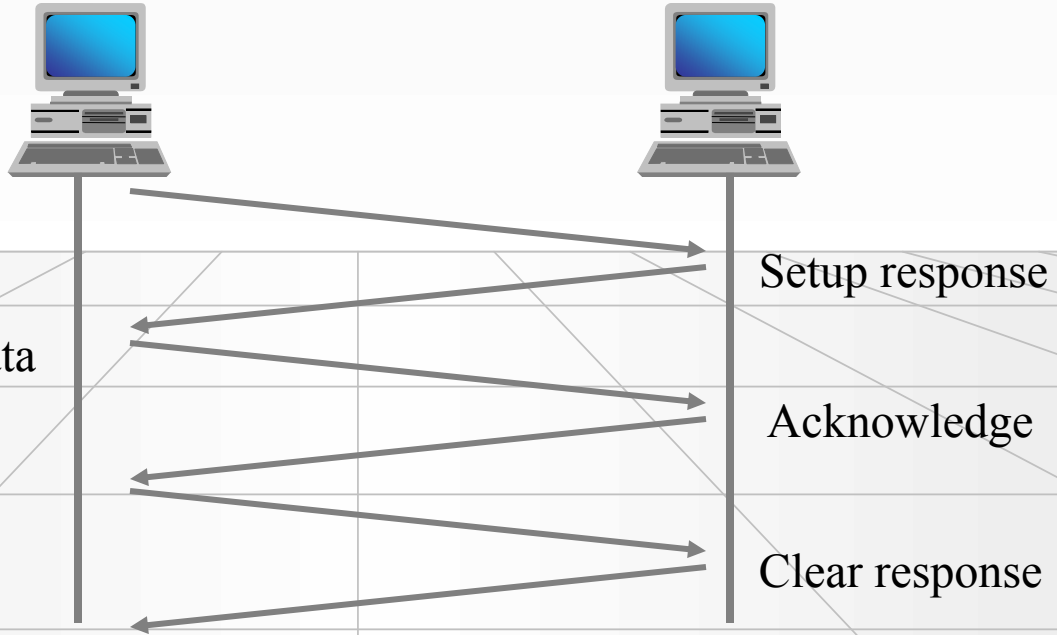
UDP is connectionless and unreliable.

Transport (Host-to-Host) Layer Protocols

- Connectionless Protocol



- Connection-Oriented Protocol



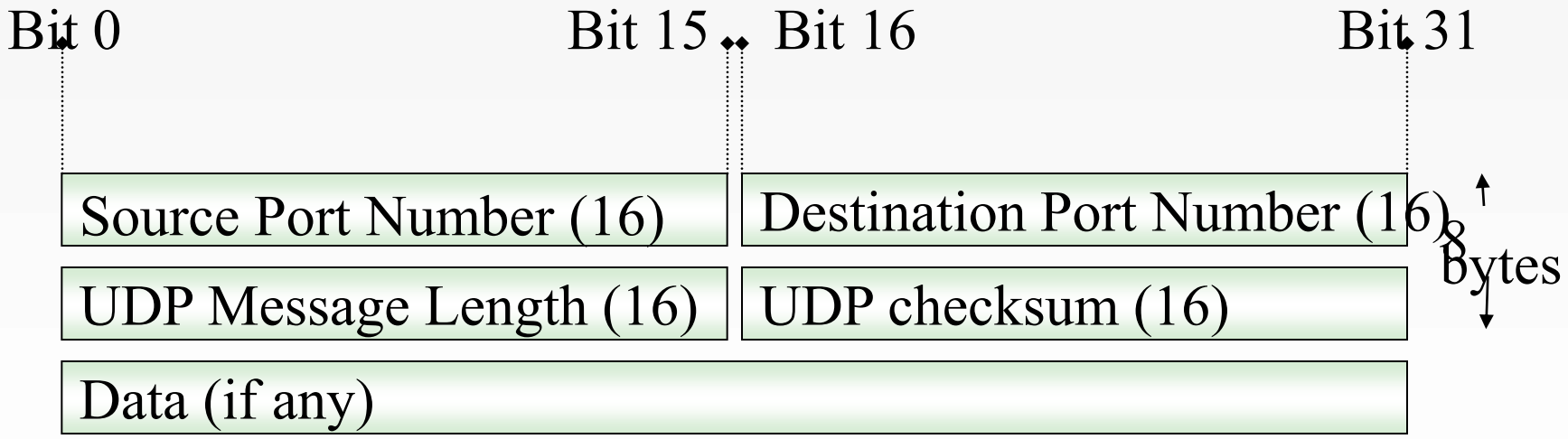
UDP

- Not reliable
- Connectionless
- Provides ports
  - Ports identifies the application that send or recieve the data
- Checksum is optional
  - Reliability reduced even further if it is not used
  - Checksum cover also the data field

UDP Header

Ethernet	IP	UDP	UPD Payload	CRC
14	20-60	8	Variable	4

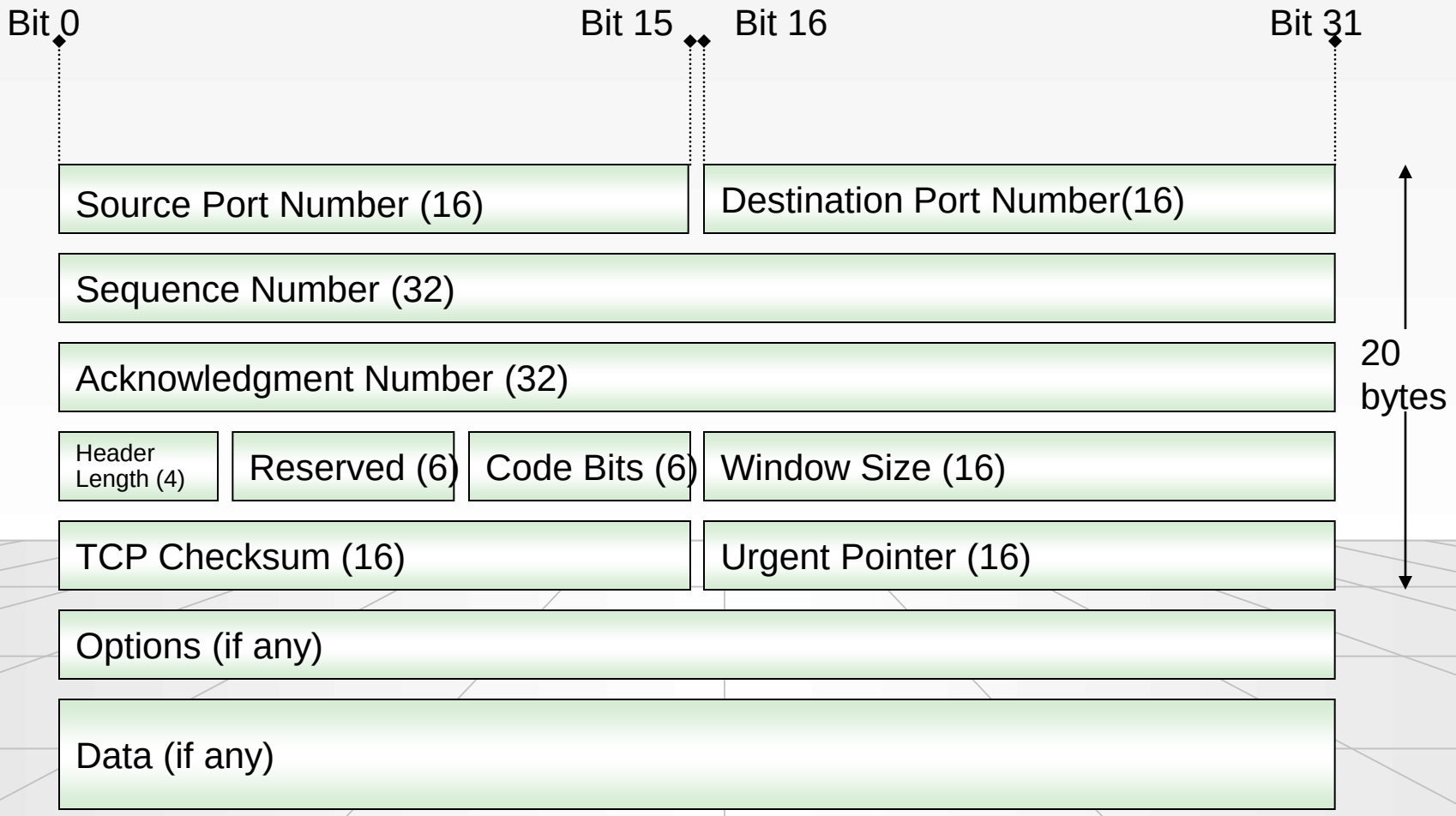
UDP Header Layout



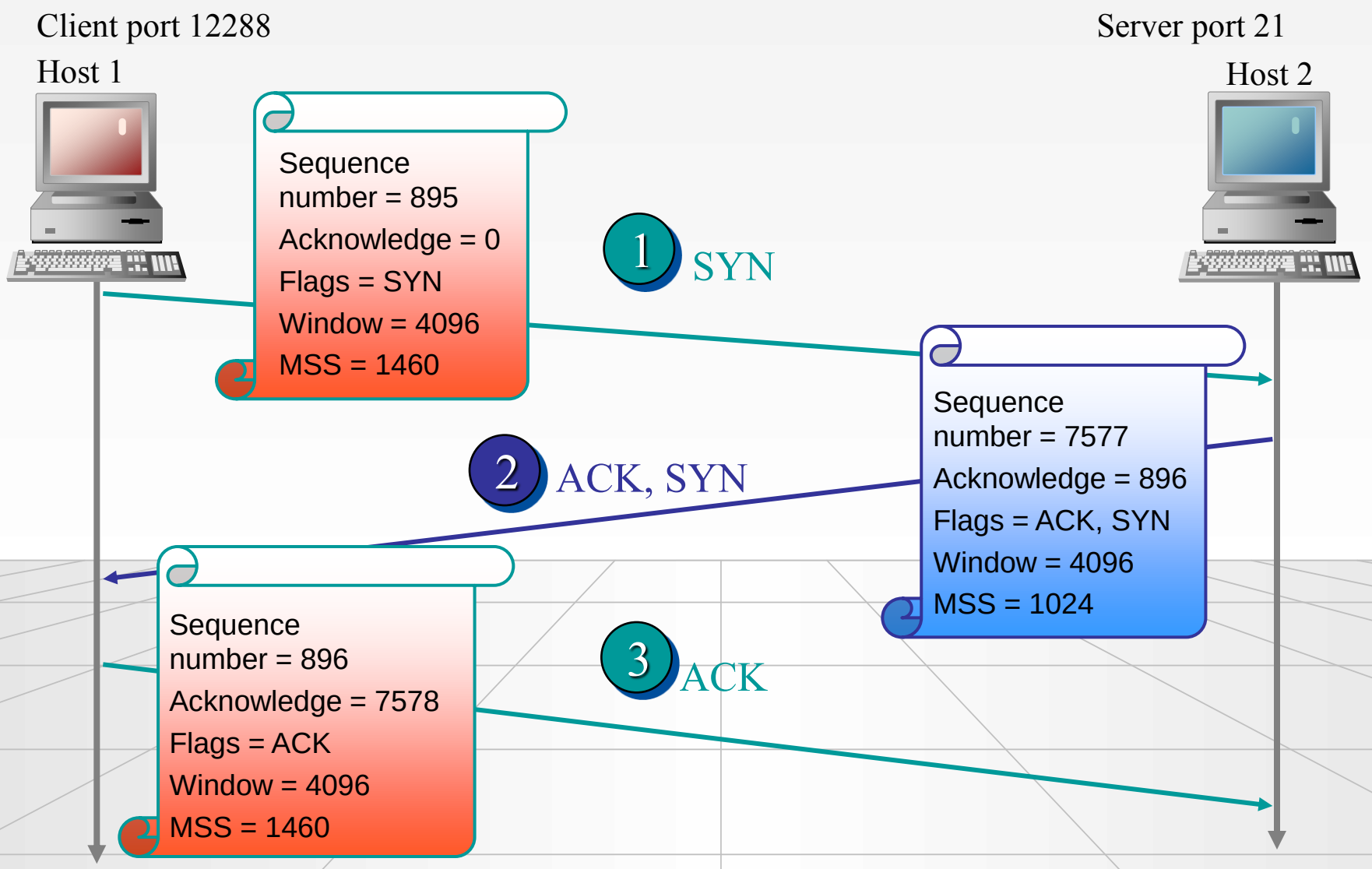
TCP Header

Ethernet Header	IP	TCP	TCP Payload	Checksum
14	20 to 60	20 or 24	Varies (may not exist)	4

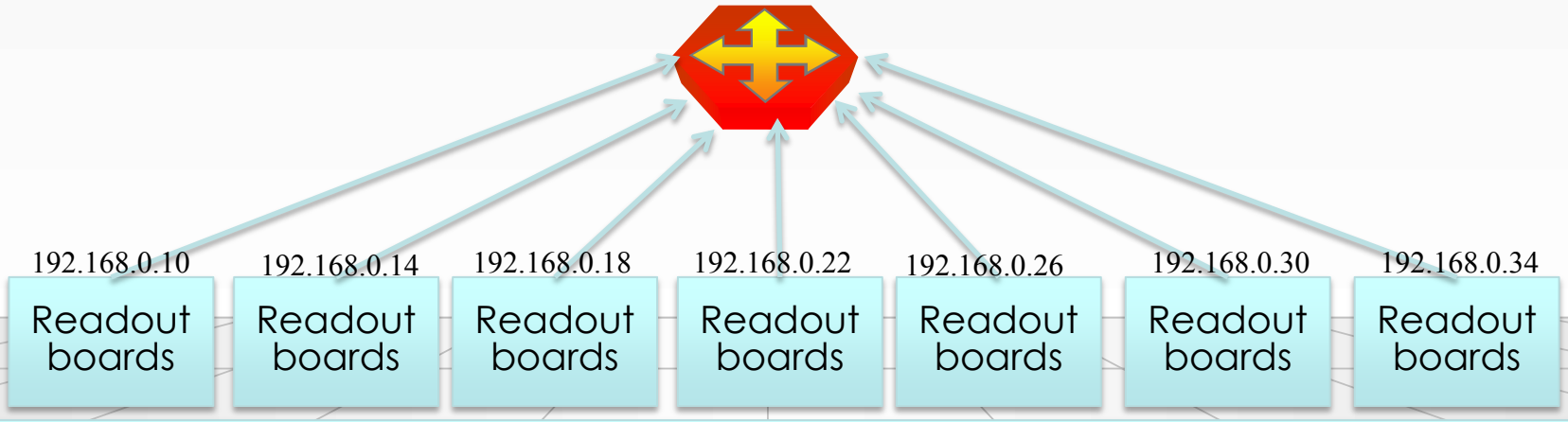
TCP Header Layout



TCP Three-Way Handshake



Example of DAQ Network



Experiment, composed by several sub detectors:
Our data is produced here

Backup slides

Step 1: Request for Synchronization

```
+Frame: Base frame properties
+ETHERNET: ETYPE = 0x0800 : Protocol = IP:  DOD Internet Protocol
+IP: ID = 0x2019; Proto = TCP; Len: 48
-TCP: ....S., len:    0, seq:3936932349-3936932349, ack:         0, win:16384, src: 1456  dst:   80
   TCP: Source Port = 0x05B0
   TCP: Destination Port = Hypertext Transfer Protocol
   TCP: Sequence Number = 3936932349 (0xEAA8D1FD)
   TCP: Acknowledgement Number = 0 (0x0)
   TCP: Data Offset = 28 (0x1C)
   TCP: Reserved = 0 (0x0000)
  -TCP: Flags = 0x02 : ....S.
    TCP: ..0..... = No urgent data
    TCP: ...0.... = Acknowledgement field not significant
    TCP: ....0... = No Push function
    TCP: .....0.. = No Reset
    TCP: ......1. = Synchronize sequence numbers
    TCP: .......0 = No Fin
   TCP: Window = 16384 (0x4000)
   TCP: Checksum = 0x4333
   TCP: Urgent Pointer = 0 (0x0)
  -TCP: Options
   -TCP: Maximum Segment Size Option
      TCP: Option Type = Maximum Segment Size
      TCP: Option Length = 4 (0x4)
      TCP: Maximum Segment Size = 1460 (0x5B4)
    TCP: Option Nop = 1 (0x1)
    TCP: Option Nop = 1 (0x1)
   -TCP: SACK Permitted Option
      TCP: Option Type = Sack Permitted
      TCP: Option Length = 2 (0x2)
```

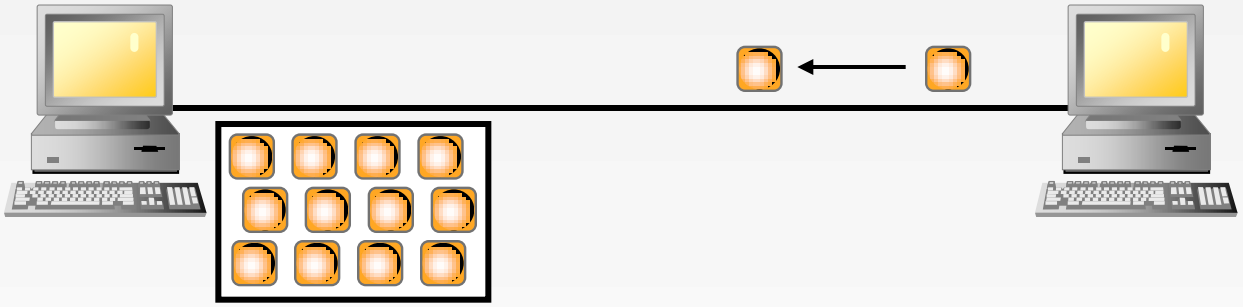
Step 2: Acknowledgment of the Client Request

```
+Frame: Base frame properties
+ETHERNET: ETYPE = 0x0800 : Protocol = IP:  DOD Internet Protocol
+IP: ID = 0x6742; Proto = TCP; Len: 48
-TCP:  .A..S.  len:    0  seq: 791458558-791458558  ack:3936932350  win:17424  src:   80  dst: 1456
  TCP: Source Port = Hypertext Transfer Protocol
  TCP: Destination Port = 0x05B0
  TCP: Sequence Number = 791458558 (0x2F2CB2FE)
  TCP: Acknowledgement Number = 3936932350 (0xEAA8D1FE)
  TCP: Data Offset = 28 (0x1C)
  TCP: Reserved = 0 (0x0000)
 -TCP: Flags = 0x12 : .A..S.
   TCP: ..0..... = No urgent data
   TCP: ...1.... = Acknowledgement field significant
   TCP: ....0... = No Push function
   TCP: .....0.. = No Reset
   TCP: ......1. = Synchronize sequence numbers
   TCP: .......0 = No Fin
  TCP: Window = 17424 (0x4410)
  TCP: Checksum = 0x5CEF
  TCP: Urgent Pointer = 0 (0x0)
 -TCP: Options
  -TCP: Maximum Segment Size Option
     TCP: Option Type = Maximum Segment Size
     TCP: Option Length = 4 (0x4)
     TCP: Maximum Segment Size = 1452 (0x5AC)
   TCP: Option Nop = 1 (0x1)
   TCP: Option Nop = 1 (0x1)
  -TCP: SACK Permitted Option
     TCP: Option Type = Sack Permitted
     TCP: Option Length = 2 (0x2)
```

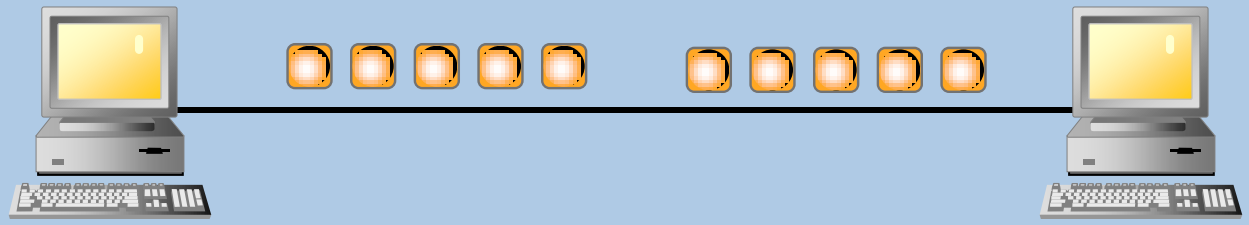
Step 3: Acknowledgment of the Server Request

```
+Frame: Base frame properties
+ETHERNET: ETYPE = 0x0800 : Protocol = IP:  DOD Internet Protocol
+IP: ID = 0x201B; Proto = TCP; Len: 40
-TCP:  A      len:    0  seq:3936932350-3936932350  ack: 791458559  win:17424  src: 1456  dst:   80
   TCP: Source Port = 0x05B0
   TCP: Destination Port = Hypertext Transfer Protocol
   TCP: Sequence Number = 3936932350 (0xEAA8D1FE)
   TCP: Acknowledgement Number = 791458559 (0x2F2CB2FF)
   TCP: Data Offset = 20 (0x14)
   TCP: Reserved = 0 (0x0000)
  -TCP: Flags = 0x10 : .A....
     TCP: ..0..... = No urgent data
     TCP: ...1.... = Acknowledgement field significant
     TCP: ....0... = No Push function
     TCP: .....0.. = No Reset
     TCP: ......0. = No Synchronize
     TCP: .......0 = No Fin
   TCP: Window = 17424 (0x4410)
   TCP: Checksum = 0x89AB
   TCP: Urgent Pointer = 0 (0x0)
```

Congestion and TCP

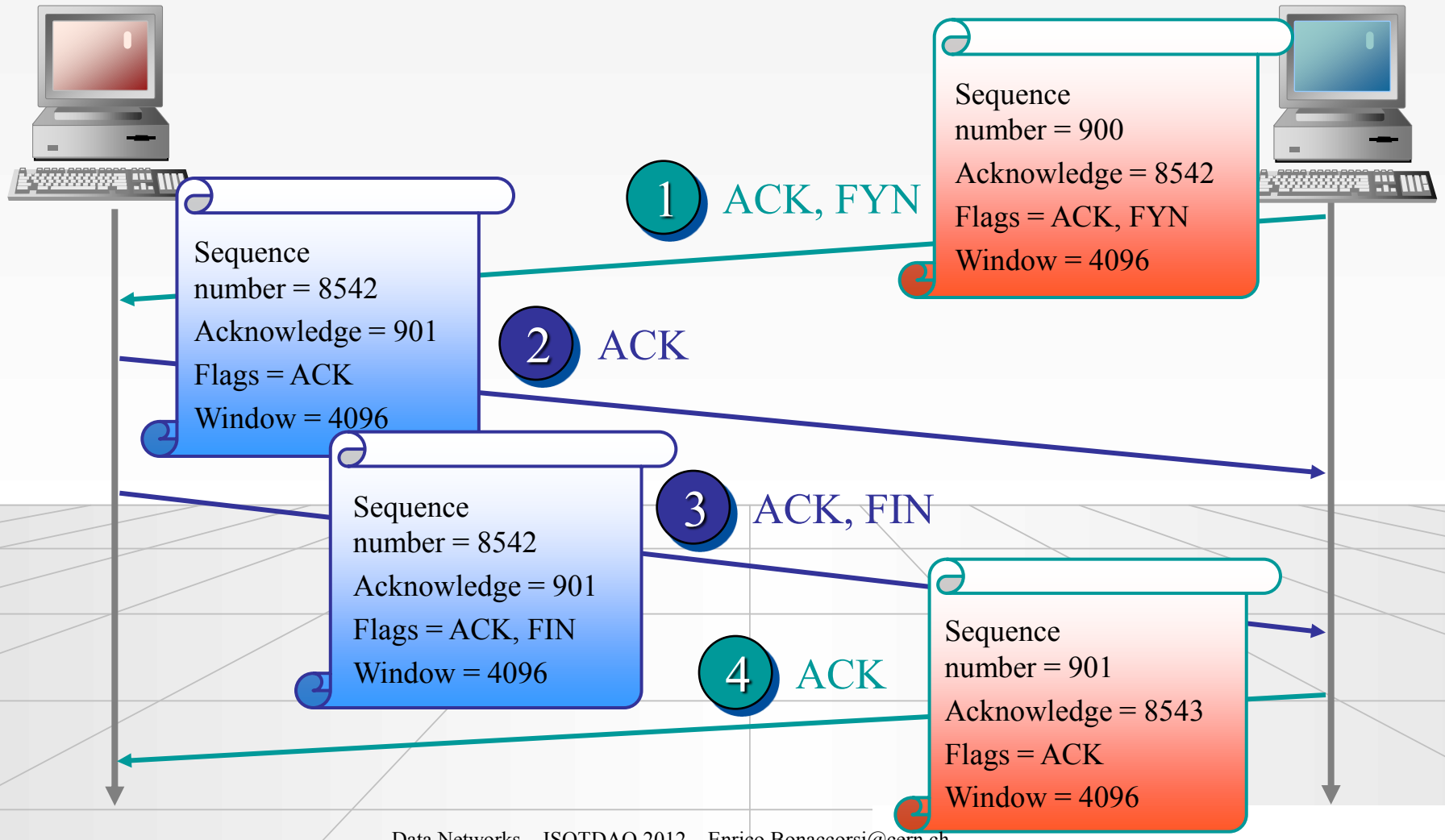


When a host is congested it sets its window size to 0.



An indication of a network congestion is a large window size and no change in the returned ACK numbers.

Normal End of the Session



Step 1: Sending a FIN

```
+Frame: Base frame properties
+ETHERNET: ETYPE = 0x0800 : Protocol = IP:  DOD Internet Protocol
+IP: ID = 0x3227; Proto = TCP; Len: 40
-TCP: .A...F, len:    0, seq:1955119560-1955119560, ack:2911408263, win: 8267, src:   21 (FTP)  dst: 2244
   TCP: Source Port = FTP [control]
   TCP: Destination Port = 0x08C4
   TCP: Sequence Number = 1955119560 (0x7488C1C8)
   TCP: Acknowledgement Number = 2911408263 (0xAD889087)
   TCP: Data Offset = 20 (0x14)
   TCP: Reserved = 0 (0x0000)
  -TCP: Flags = 0x11 : .A...F
     TCP: ..0..... = No urgent data
     TCP: ...1.... = Acknowledgement field significant
     TCP: ....0... = No Push function
     TCP: .....0.. = No Reset
     TCP: ......0. = No Synchronize
     TCP: .......1 = No more data from sender
   TCP: Window = 8267 (0x204B)
   TCP: Checksum = 0xFE6C
   TCP: Urgent Pointer = 0 (0x0)
```

Step 2: Acknowledging the FIN

```
+Frame: Base frame properties
+ETHERNET: ETYPE = 0x0800 : Protocol = IP:  DOD Internet Protocol
+IP: ID = 0x61A7; Proto = TCP; Len: 40
-TCP: A    , len:    0, seq:2911408263-2911408263, ack:1955119561, win:16454, src: 2244  dst:   21 (FTP)
  TCP: Source Port = 0x08C4
  TCP: Destination Port = FTP [control]
  TCP: Sequence Number = 2911408263 (0xAD889087)
  TCP: Acknowledgement Number = 1955119561 (0x7488C1C9)
  TCP: Data Offset = 20 (0x14)
  TCP: Reserved = 0 (0x0000)
 -TCP: Flags = 0x10 : .A....
   TCP: ..0..... = No urgent data
   TCP: ...1.... = Acknowledgement field significant
   TCP: ....0... = No Push function
   TCP: .....0.. = No Reset
   TCP: ......0. = No Synchronize
   TCP: .......0 = No Fin
  TCP: Window = 16454 (0x4046)
  TCP: Checksum = 0xDE71
  TCP: Urgent Pointer = 0 (0x0)
```

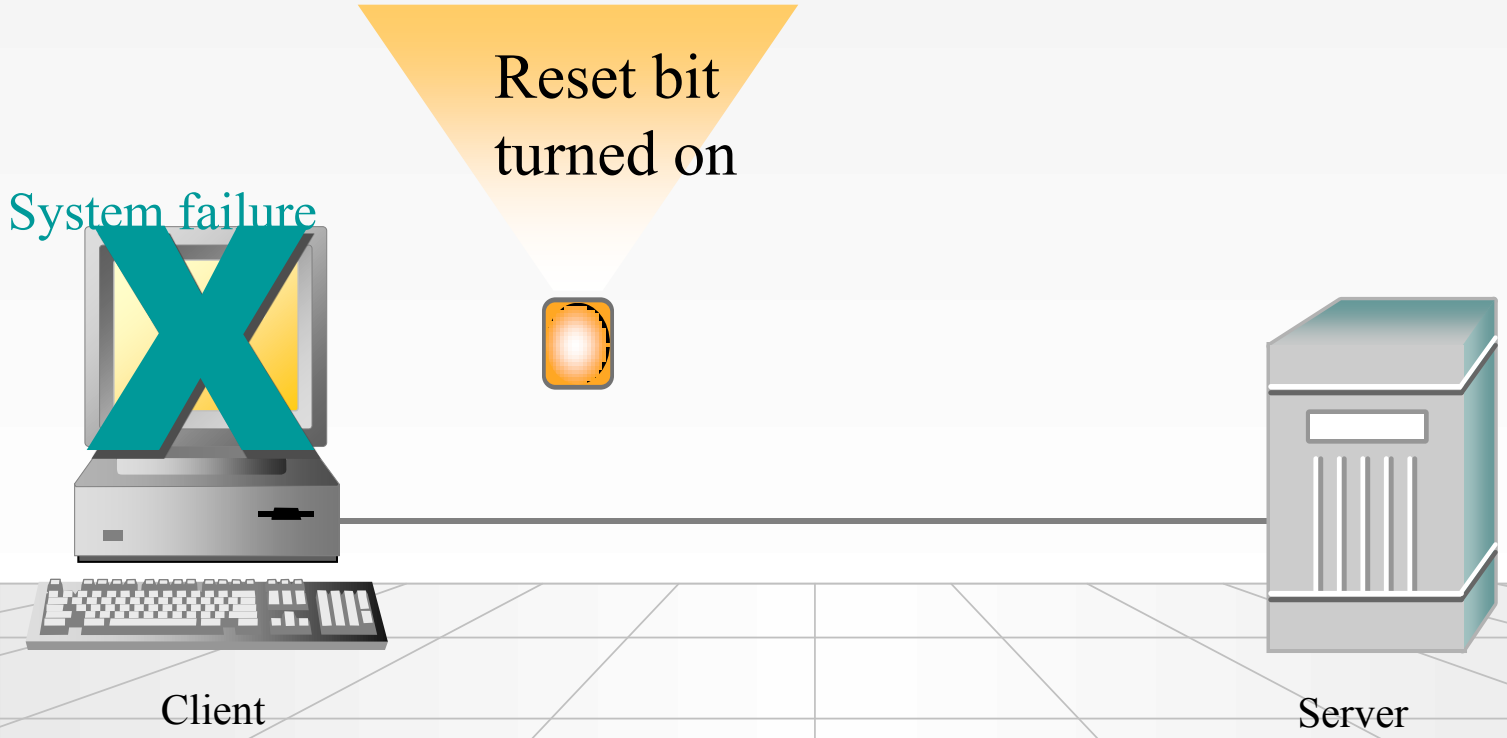
Step 3: Client Sends a FIN

```
+Frame: Base frame properties
+ETHERNET: ETYPE = 0x0800 : Protocol = IP:  DOD Internet Protocol
+IP: ID = 0x61A8; Proto = TCP; Len: 40
-TCP:  A  F  len:    0  seq:2911408263-2911408263  ack:1955119561  win:16454  src: 2244  dst:   21 (FTP)
  TCP: Source Port = 0x08C4
  TCP: Destination Port = FTP [control]
  TCP: Sequence Number = 2911408263 (0xAD889087)
  TCP: Acknowledgement Number = 1955119561 (0x7488C1C9)
  TCP: Data Offset = 20 (0x14)
  TCP: Reserved = 0 (0x0000)
 -TCP: Flags = 0x11 : .A...F
   TCP: ..0..... = No urgent data
   TCP: ...1.... = Acknowledgement field significant
   TCP: ....0... = No Push function
   TCP: .....0.. = No Reset
   TCP: ......0. = No Synchronize
   TCP: .......1 = No more data from sender
  TCP: Window = 16454 (0x4046)
  TCP: Checksum = 0xDE70
  TCP: Urgent Pointer = 0 (0x0)
```

Step 4: The Server Acknowledges the FIN

```
+Frame: Base frame properties
+ETHERNET: ETYPE = 0x0800 : Protocol = IP:  DOD Internet Protocol
+IP: ID = 0x3327; Proto = TCP; Len: 40
-TCP: .A....  len:    0  seq:1955119561-1955119561  ack:2911408264  win: 8267  src:   21 (FTP)  dst: 2244
  TCP: Source Port = FTP [control]
  TCP: Destination Port = 0x08C4
  TCP: Sequence Number = 1955119561 (0x7488C1C9)
  TCP: Acknowledgement Number = 2911408264 (0xAD889088)
  TCP: Data Offset = 20 (0x14)
  TCP: Reserved = 0 (0x0000)
 -TCP: Flags = 0x10 : .A....
   TCP: ..0..... = No urgent data
   TCP: ...1.... = Acknowledgement field significant
   TCP: ....0... = No Push function
   TCP: .....0.. = No Reset
   TCP: ......0. = No Synchronize
   TCP: .......0 = No Fin
  TCP: Window = 8267 (0x204B)
  TCP: Checksum = 0xFE6B
  TCP: Urgent Pointer = 0 (0x0)
```

Reset Session



Network Address Translation

- Nat includes:
  - Static NAT
    - Permanent one-to-one mapping
    - Allows outbound and inbound sessions
  - Dynamic NAT
    - Mappings dynamically assigned from pool
    - Allow outbound sessions only
  - PAT
    - One public address serves many internal sessions
    - Allows oubound sessions only