CA169

Networks & Internet

Lecturer & Co-ordinator

Brian.Stone@DCU.IE

The Course (indicitive)

- Layered Network Architectures.
- History of the Internet
- Physical Issues, Data Encoding, etc..
- The Data-Link Layer
  - The 802 protocols
  - Wired & Wireless Networks
- The Web
- Network Devices
- Communications Protocols
- Local area networks (data link layer)
- Internet Protocols
- Tying together MAC, IP and TCP addressing and introducing URLs
- Network tools

Indicative Assignments and Lab work

- Generating HTML for the Web
- Networking tools – Using Wireshark
- Wireshark traffic capture and analysis exercises (several)
  - Web browsers
- Moodle quizzes
- Network survey

Course Structure

- This course
  - Small timetabling differences later in semester
  - Exam worth 50%
  - Assignments worth 50%
  - Repeats possible for assessments and exams.
- 2 lectures per week (for now)
- Labs on occasions (some online)
- **Moodle** for distributing notes and collecting assignments.

Lecture Notes

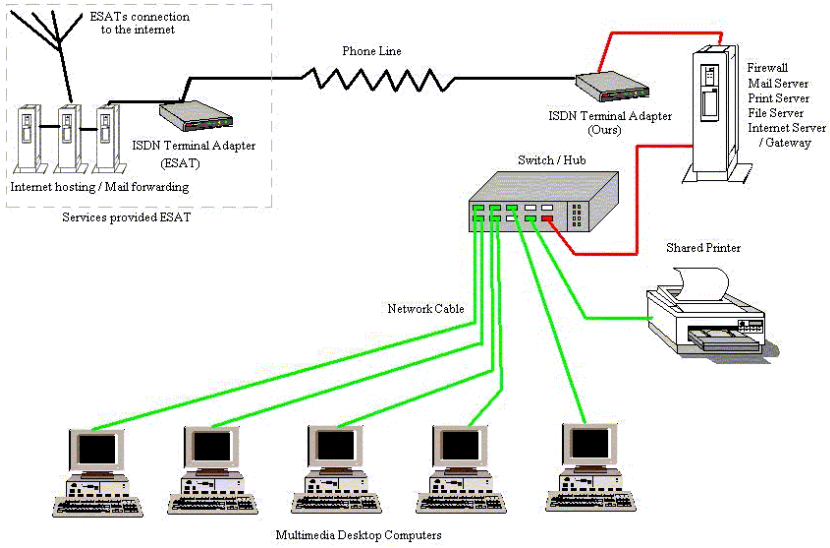
CA169

What is a network?

- Two or more computers linked together to share resources
  - Share files, printers, electronic communication etc.
  - Linked through cables, telephone lines, radio waves, satellites or infrared beams.
- Four basic types of networks
  - Local Area Network (LAN)
  - Personal Area Networks (PAN)
  - Metropolitan Area Network (MAN)
  - Wide Area Network (WAN)

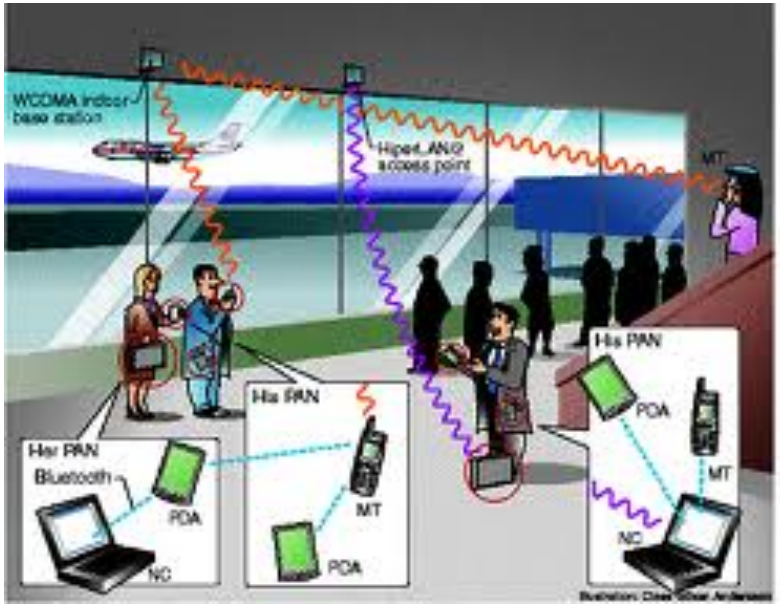
Local Area Network

- Network confined to a relatively small geographic area. Typically lab in CA, office etc.
- Typical LAN has a central server which controls the network. Can share resources ….printers, files. Should be more powerful than the clients.



PAN

- Tends to be wireless WPAN
- Phone talking to organiser or Internet
- May be carried over
  - Ethernet
  - WiFi
  - NFC
  - IR
  - Bluetooth
  - ZigBee



Metropolitan Area Network

- Covers larger geographic areas, cities, schools, local libraries, government offices
- Typically uses dedicated phone lines, coaxial cabling, fibre optic cable and wireless communication

Wide Area Network

- Connects larger geographic areas, such as global companies. Local and global networks are connected to form larger network.
- Typically uses transoceanic or satellite links
- Protocols used can be ATM networks or MPLS (carrying Ethernet) or others.
- Typically use special hardware and special fibres.
- Physical layer can be DWDM

The Internet

- All of the previous network types can connect to the internet
- Infrastructure software needed – TCP\IP
- Global services available through Internet
- Internet aware applications
- Network aware devices

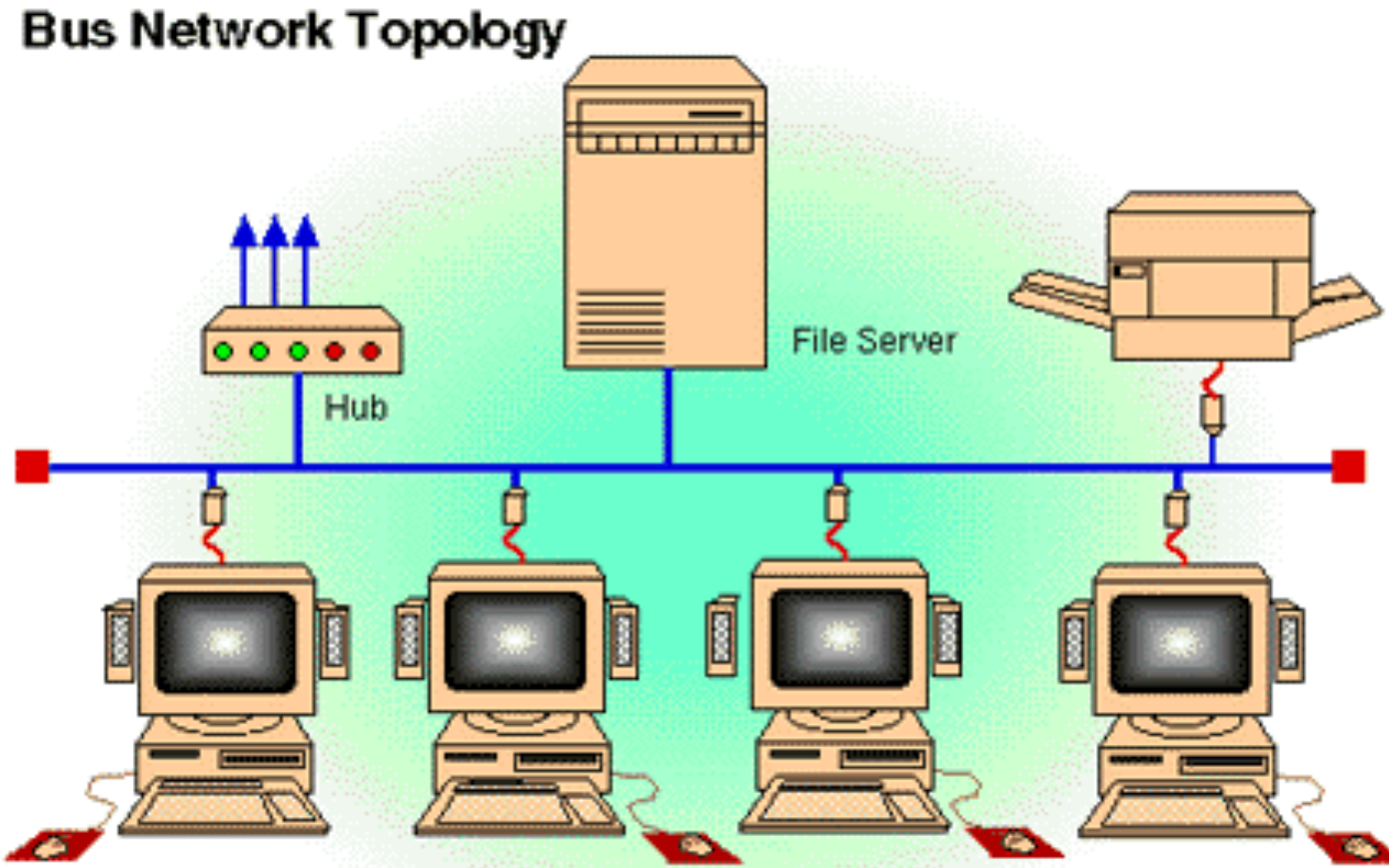
- The Internet has transformed the way we do things!

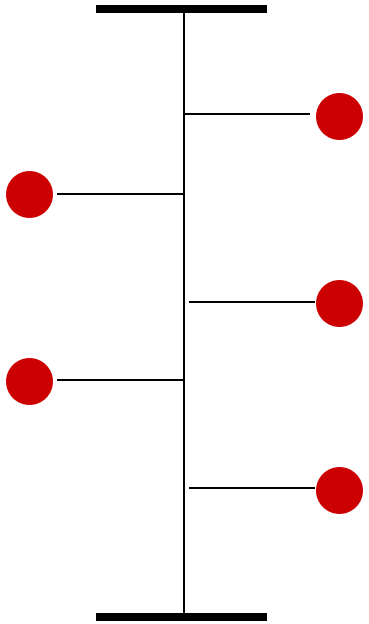
Topology

Network Topologies

- Topology is how the cables, computers and other peripherals are connected
- Different types of topologies
  - Star
  - Ring
  - Bus
  - Tree
  - Complete
  - Irregular

Bus Topology

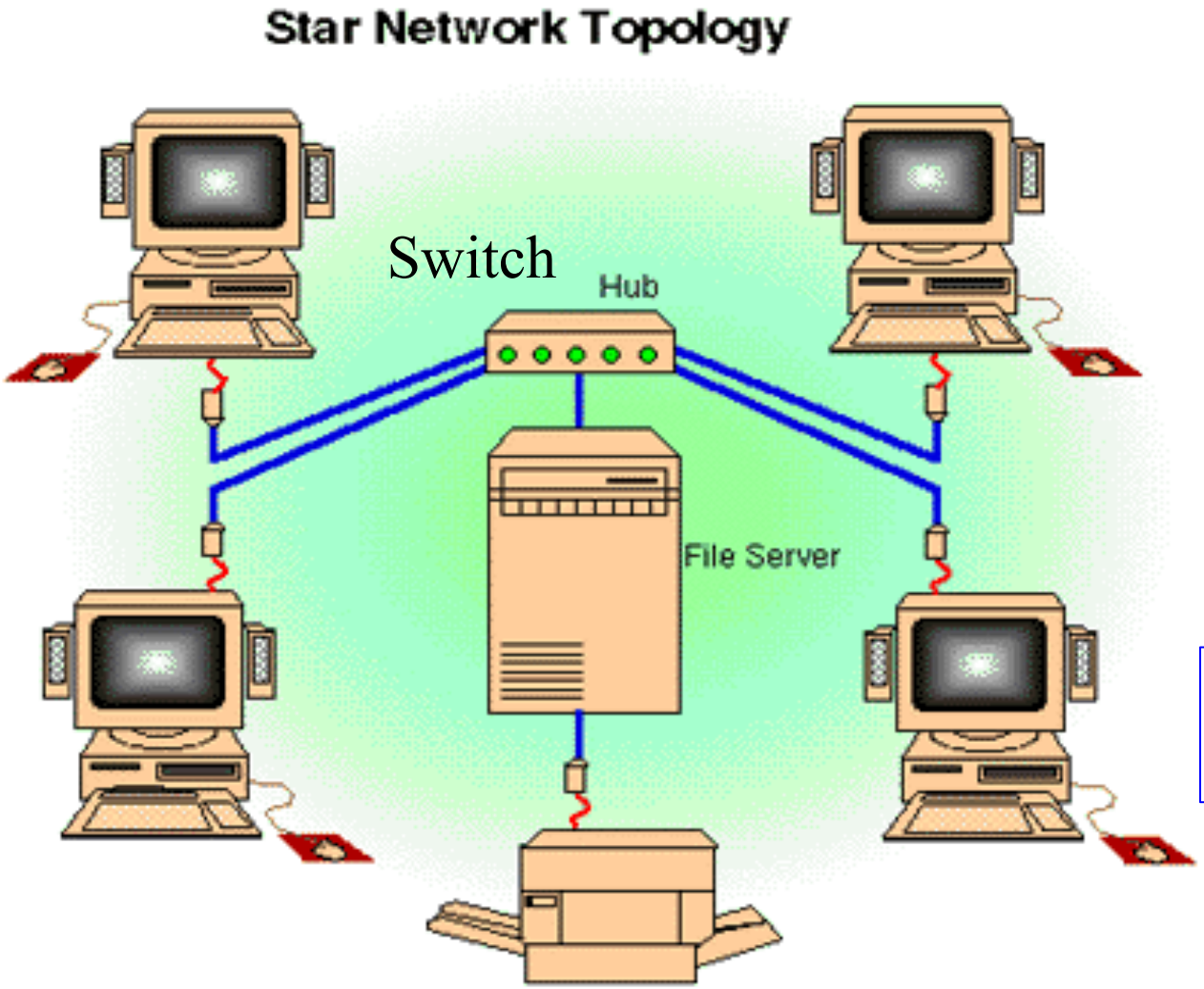


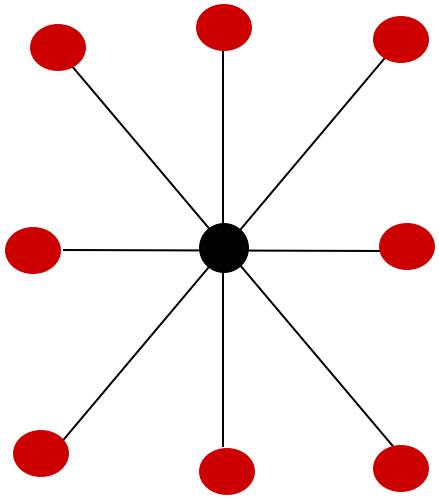


Bus Topology (2)

- Computers share the same bus (cable) with a terminator at each end. Each client is connected to the bus.
  - Old Ethernet on coaxial cable utilises a bus
- Simple and reliable, not much hardware needed.
- Inexpensive cable and easy to expand
  - Uses the least cable
  - Management more problematic
- Heavy traffic slows overall throughput
- A break in network brings whole thing down. Can be difficult to detect.

Star Topology



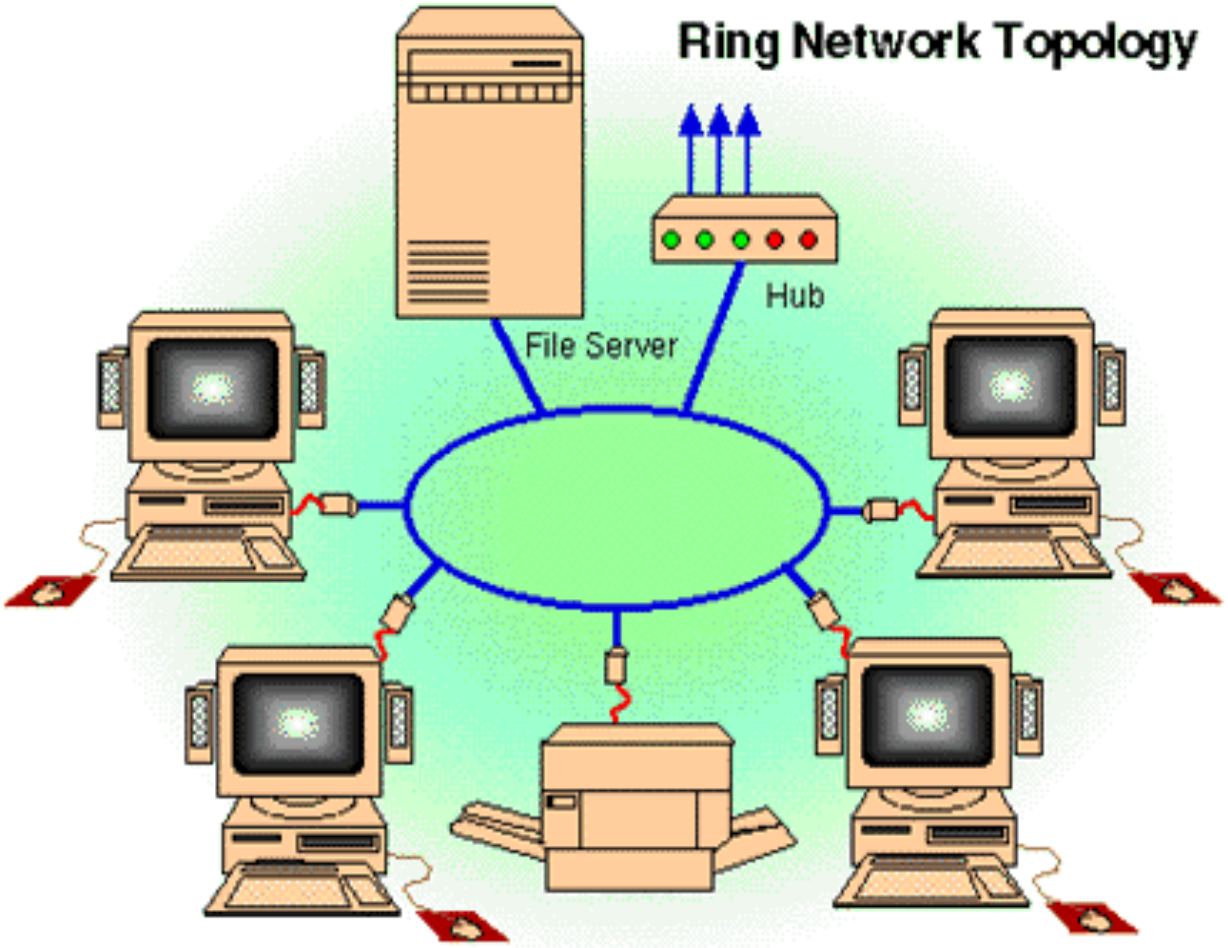


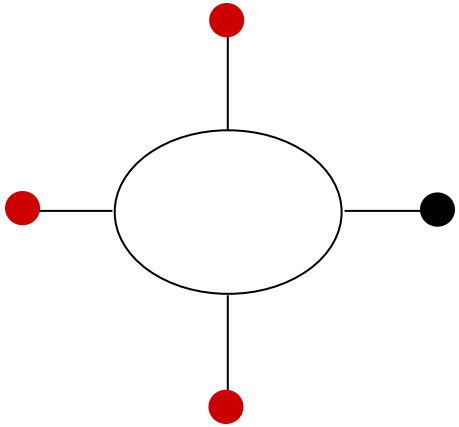
Check out the cabinets in our labs.

Star Topology (2)

- Each node connected directly to central computer. All data must go through central node (hub/switch)
- Relies heavily on central computer
- Each device has a separate wire. Easy to install new devices. Disconnecting / Adding devices does not interrupt network. Easy to detect breaks/faults
- More cable is required (look at our cabinets)
- If central node fails, network falls over
- Commonly uses twisted pair, also uses co-axial (rare now) cable and fibre optics

Ring Topology



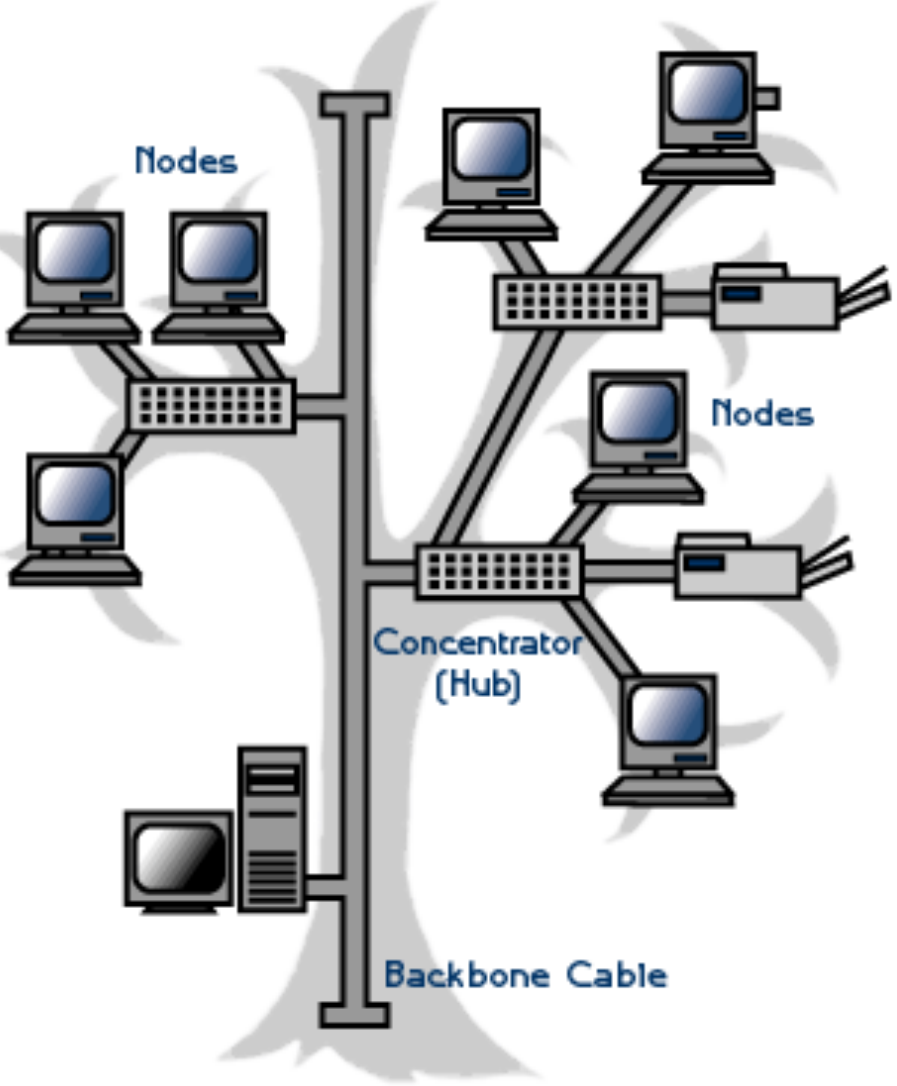


Ring Topology (2)

- Computers tied together in a ring
- Each device is connected to the next one in line
- Circle of cable
- Signal travels in one direction
- When a device receives control (token)
  - It acts on it
  - or passes it on
- Not common these days for LAN.

Tree Topology

- Modern LANs utilise Switches to build a tree topology, even when the network looks like a mesh.
- Need to ensure that loops are not introduced, special protocols built into switches (STP)



Topology Considerations

- Money
  - Bus cheapest, no need for central node
  - But what about management cost?
- Length of cable
  - Bus uses shortest cable, but how expensive is cable anyhow?
- Efficiency
  - Star topology easies to add new nodes
  - Manage existing infrastructure
- Cable type
  - Most common cable is twisted pair, most often used with star topologies

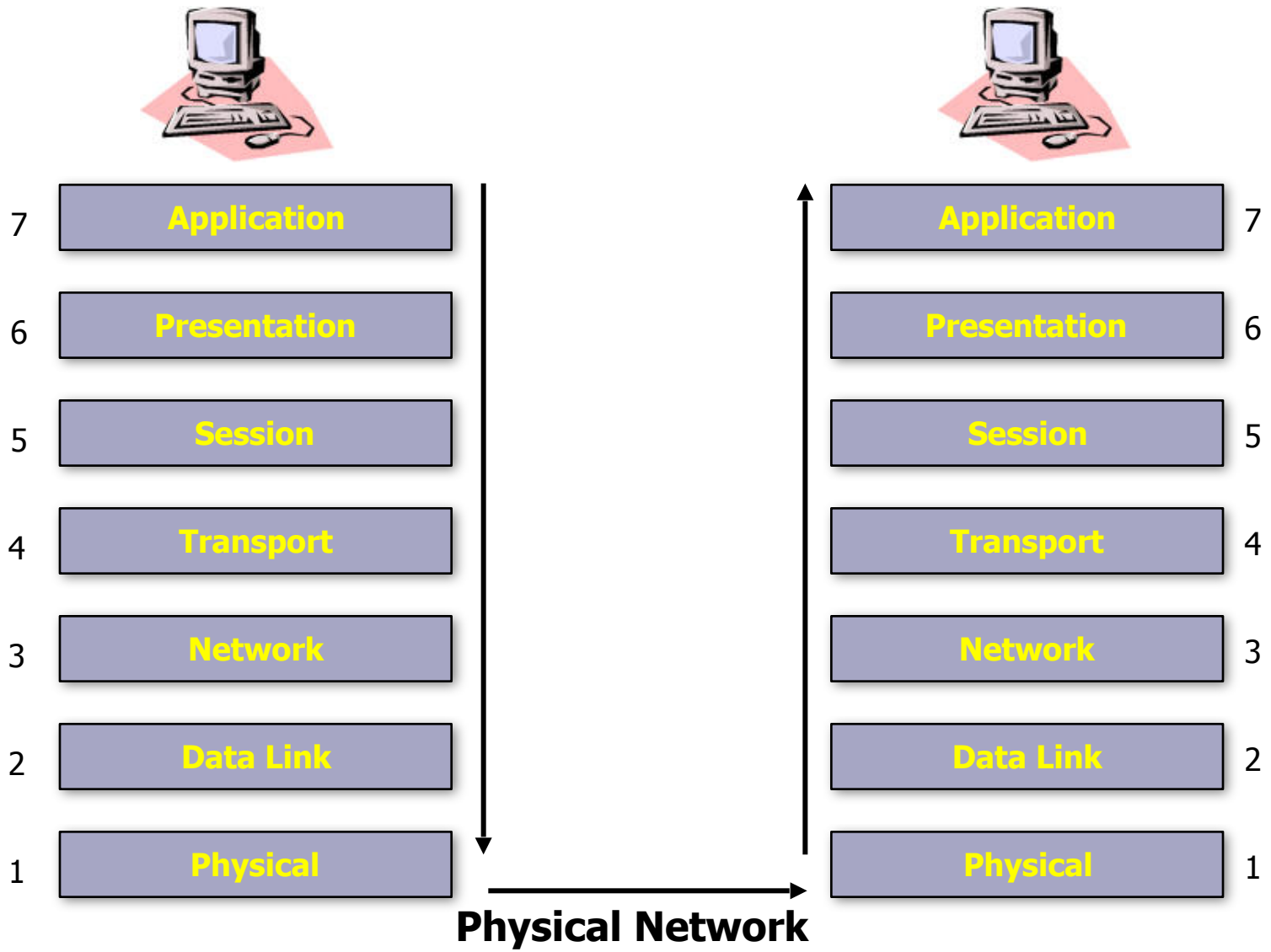
Network Architecture

Need to break down idea of a network into easily understood abstractions – use layered architecture

Layered Architectures

- The ISO defines a 7 layer Architecture.
- TCP\IP is defined as a 4 layer Architecture.
- ISO is a prescriptive Architecture
- TCP\IP is descriptive.

OSI Model


7
Application
6
Presentation
5
Session
4
Transport
3
Network
2
Data Link
1
Physical
Physical Network
Application
7
Presentation
6
Session
5
Transport
4
Network
3
Data Link
2
Physical
1

OSI (2)

7. Application Layer (Top Layer)
6. Presentation Layer
5. Session Layer
4. Transport Layer
3. Network Layer
2. Data Link Layer
1. Physical Layer (Bottom Layer)

OSI (3)

- The **Application layer** represents the level at which applications access network services. This layer represents the services that directly support applications such as software for file transfers, database access, and electronic mail.
- The **Presentation layer** translates data from the Application layer into an intermediary format. This layer also manages security issues by providing services such as data encryption, and compresses data so that fewer bits need to be transferred on the network.
- The **Session layer** allows two applications on different computers to establish, use, and end a session. This layer establishes dialog control between the two computers in a session, regulating which side transmits, plus when and how long it transmits.

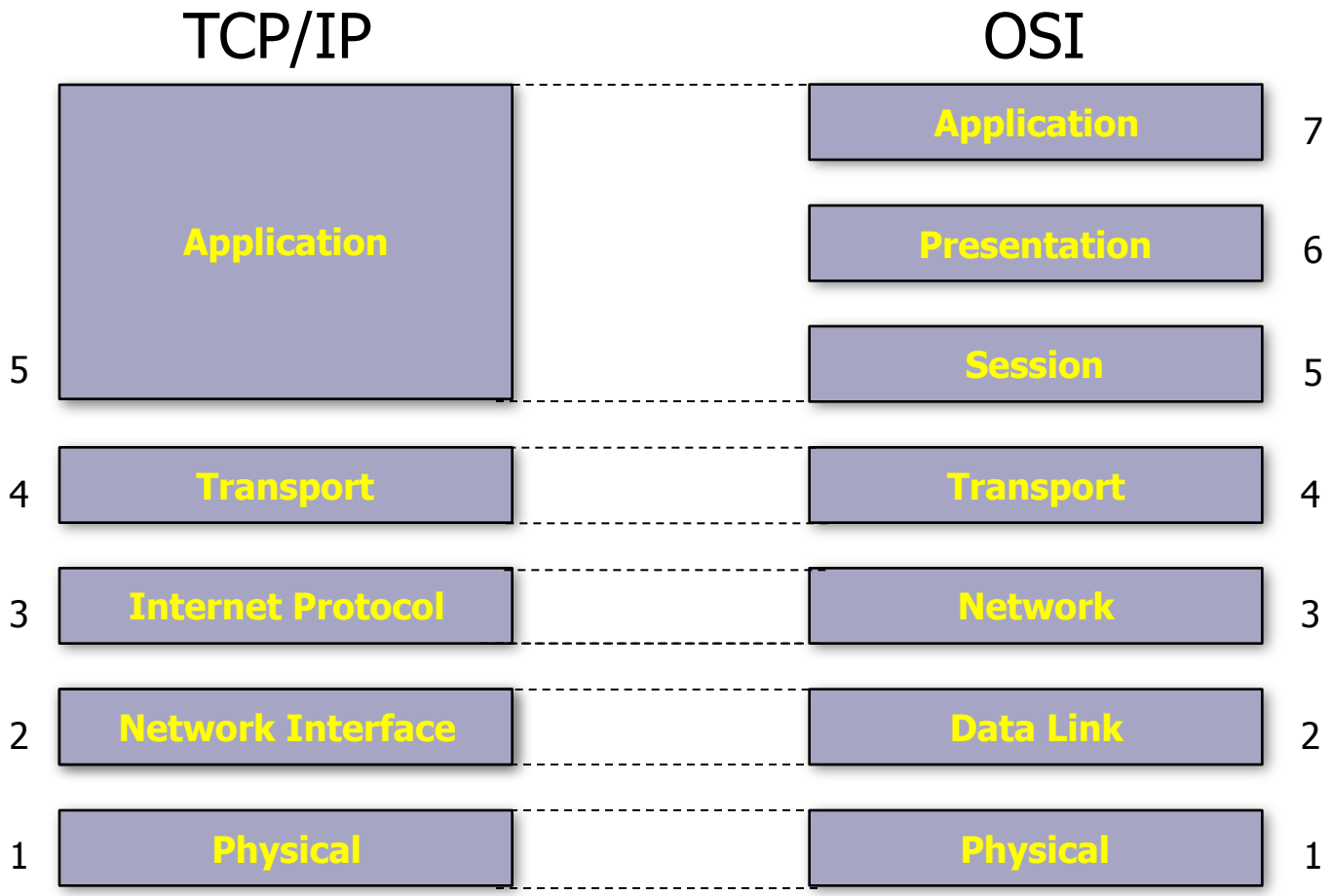
OSI (4)

- The **Transport layer** handles error recognition and recovery. It also repackages long messages when necessary into small packets for transmission and, at the receiving end, rebuilds packets into the original message. The receiving Transport layer also sends receipt acknowledgments.
- The **Network layer** addresses messages and translates logical addresses and names into physical addresses. It also determines the route from the source to the destination computer and manages traffic problems, such as switching, routing, and controlling the congestion of data packets.
- The **Data Link layer** packages raw bits from the Physical layer into frames (logical, structured packets for data). This layer is responsible for transferring frames from one computer to another, without errors. After sending a frame, it waits for an acknowledgment from the receiving computer.

OSI (5)

- The **Physical layer** transmits bits from one computer to another and regulates the transmission of a stream of bits over a physical medium. This layer defines how the cable is attached to the network adapter and what transmission technique is used to send data over the cable.

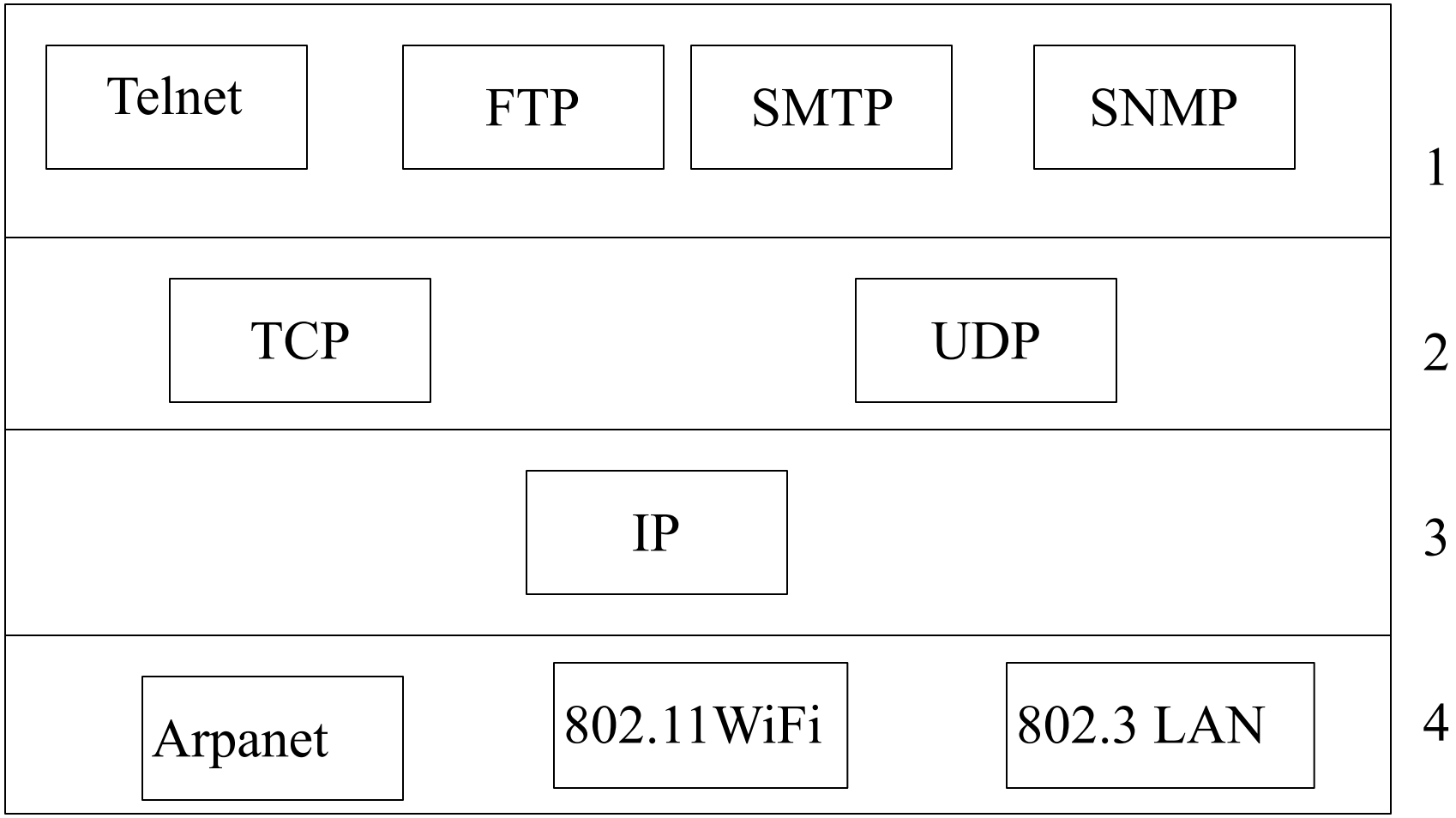
TCP/IP & OSI


TCP/IP
OSI
Application
Application
Presentation
Session
Transport
Transport
Internet Protocol
Network
Network Interface
Data Link
Physical
Physical
7
6
5
5
4
4
3
3
2
2
1
1

TCP/IP & OSI (2)

- The OSI model is the seven layer Open Systems Interconnection model. It was developed to serve as a standard model for network architectures. Transmission Control Protocol / Internet Protocol (TCP/IP) is the protocol on which the Internet is based upon. It has five layers and they are related to the OSI model as above.
- Information is transmitted around the Internet in packets. These packets contain among other things the destination and source addresses of the packet and the data. The protocol used is TCP/IP. Internet Protocol is protocol, which sends packets around the Internet. TCP sits on top of IP and it guarantees reliable delivery of packets for applications such as FTP and Telnet. An end-to-end connection is open for the delivery session between two applications.

Meet Some of the TCP/IP Family



Why Two Models

- OSI concepts
  - Services (definition)
  - Interfaces (how to access)
  - Protocols (peer protocols, private)
- Kind of OO approach, encapsulation.
- Prescriptive & Descriptive origins
  - Simple services, interfaces, protocols

Why Two Models (cont.)

- Bad Timing
  - Apocalypse of Two elephants (research & investment)

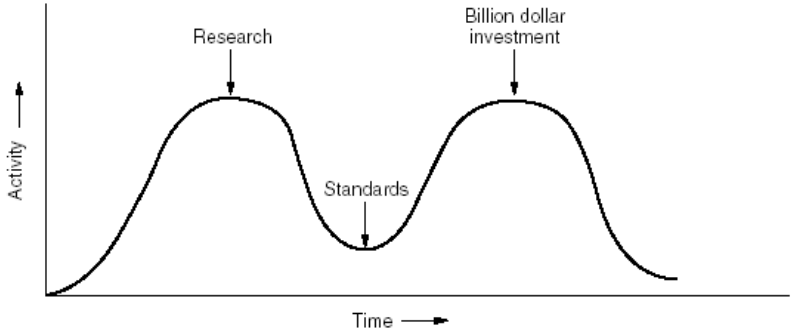


Figure 1-23. The apocalypse of the two elephants.

- Bad technology.
  - Copying proprietary SNA.
  - Empty layers.
  - Cross a mobster with a standard…
    - you get made an offer that you cannot understand.

Physical Layer

Host-to-Network Layer of TCP/IP

Physical Layer Issues

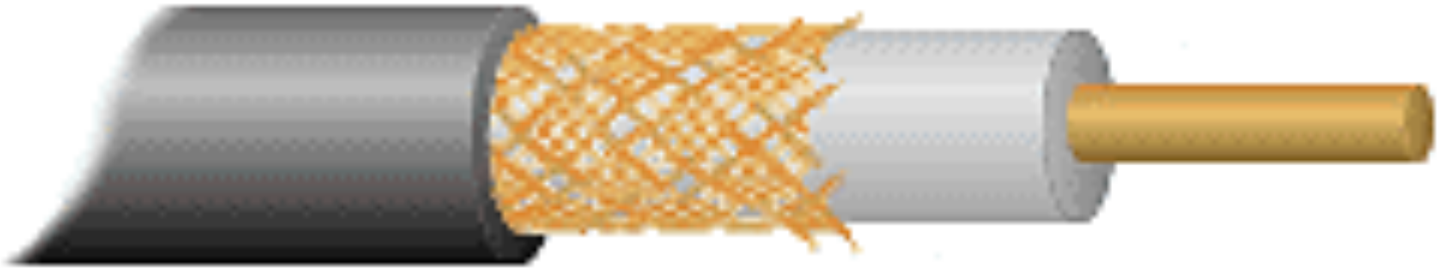
- Attenuation & Data Rates.
- Nyquest's Theorem
  - Max rate = 2H bps (H is bandwidth in Hz)
- Shannon's Theorem
  - Max rate = H $\log_2$ (1 + S/N) bps
  - Shannon takes noise into account.

Media

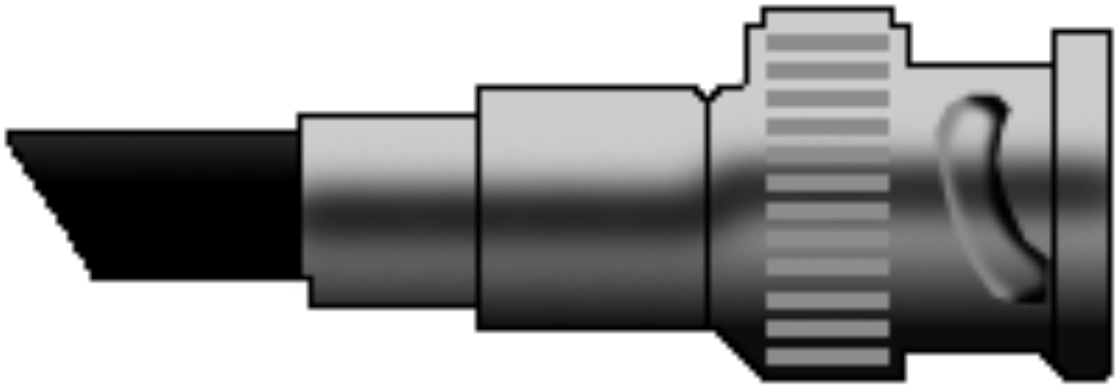
Coaxial Cable

- Like your TV cable at home
- Single copper conductor at centre with plastic layer providing insulation between conductor and braided metal shield. Shield prevents interference
- Supports longer cable length the UTP
- Thin coaxial (10Base2)
  - Max segment length 185M
- Thick coaxial (10Base5)
  - Max segment length 500M

Coaxial Cable (2)



Coaxial Cable

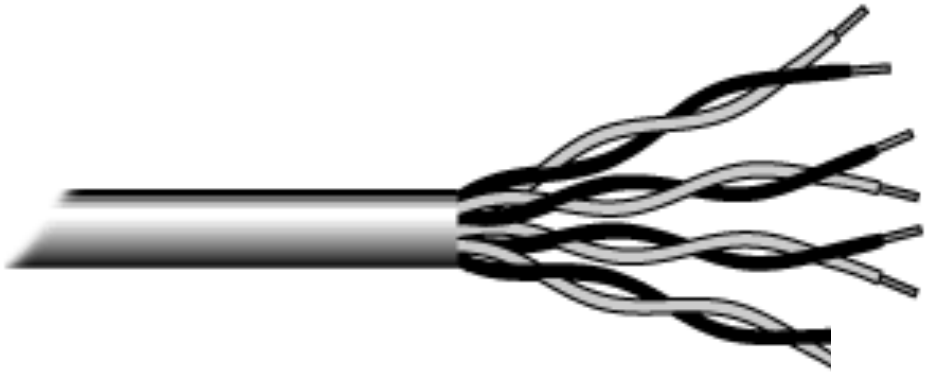


BNC Connector

Unshielded Twisted Pair

- Cable has 4 pairs of wires, twisted in pairs
- UTP can be telephone grade to high-speed cable
- 5 categories
  - 1 voice only
  - 2 Data up to 4 Mbps
  - 3 Data up to 10 Mbps
  - 4 Data up to 20 Mbps
  - 5 Data up to 100 Mbps
- Can be susceptible to radio and electrical interference. Shielded Twisted Pair exists, but extra shielding makes it bulky

Unshielded Twisted Pair (2)

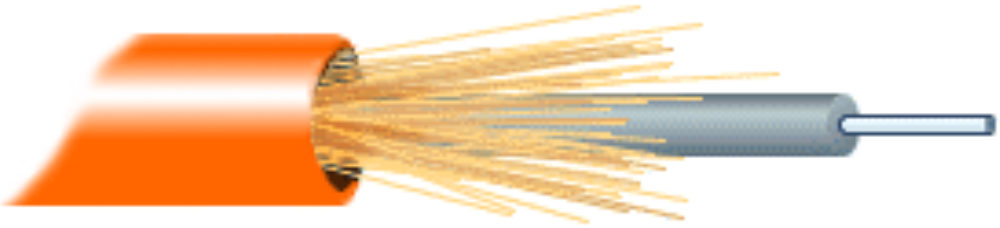


Unshielded twisted pair

RJ-45 Connector

Fibre Optic

- Centre glass core surrounded by layers of protection
- Transmits light rather than electrical signals Not susceptible to electrical interference
- Capable of transmitting data over longer distances and at higher speeds than coaxial and TP
- 10BaseF
- Outer coating is made from Teflon or PVC
- Plastic helps cushion glass core
- Kevlar around plastic strengthens cable and prevents breakage

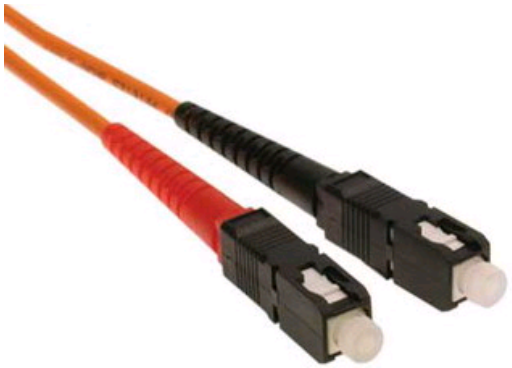


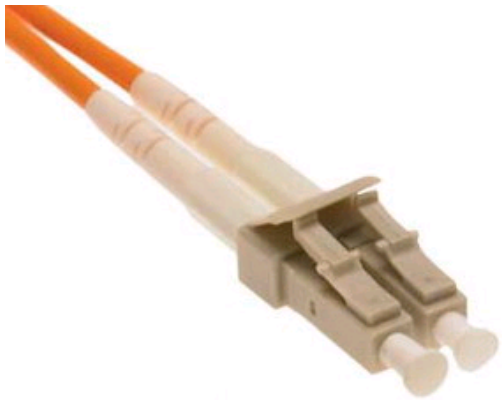
Fibre Optic Cable

Security and Fibre Networks

- From a security perspective, one of the great advantages of fibre networks is that they do not radiate any electromagnetic signals
- There is a prevalent myth that fibre networks cannot be tapped: with physical access to the cable, they can
- However, it is considered impossible to tap an optical cable without introducing a detectable increase in attenuation. A secure system should continuously monitor received optical signal strength and should alert on any abrupt change

Connectors





- SC – In widespread use. Used on the original Gigabit Ethernet GBICs
- LC – Used in newer cabling installations. Used on new small form factor (SFP) GBICs
- ST – “Bayonet” mount, often used on older fibre installations

Connectors

- FC – Screw mount. Only ever found on carrier-grade equipment (and usually with higher-powered lasers…don’t look into these)

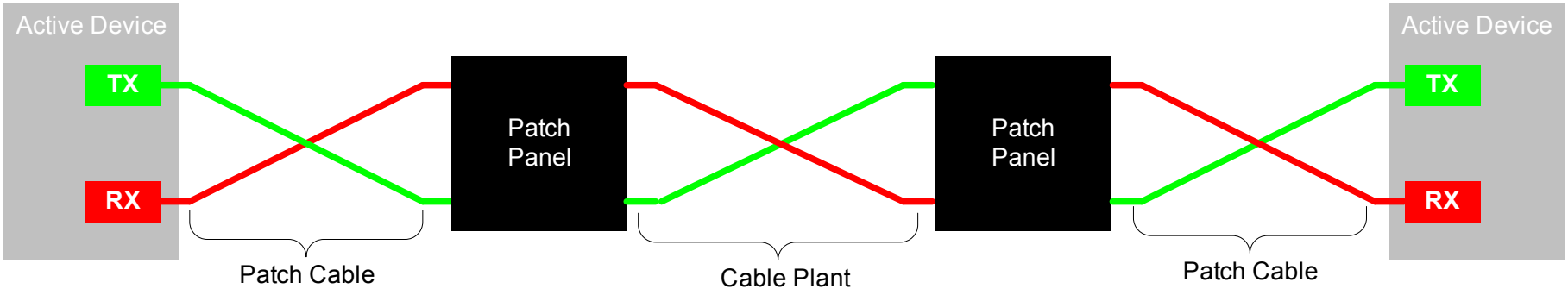
Patch Panels

- Fibre within or between buildings are typically terminated on patch panels like these

- Fibre patch cables are used to link active equipment to the patch panels

Patch Panels

- In order to ensure that Transmit is always connected to Receive in each direction, patch leads and internal cable plant are always (supposed to be) crossed over



- This means that you can use a patch cable on its own to link two physically adjacent devices

Laser Safety

- Lasers are categorised into various classes according to the amount of optical power they emit. It is important to know what these mean:-

Class 1	The output power is below the level at which it is believed eye damage will occur. Exposure to the beam of a Class 1 laser will not result in eye injury and may therefore be considered safe
Class 2	A person receiving an eye exposure from a Class 2 laser beam, either accidentally or as a result of someone else's deliberate action (misuse) will be protected from injury by their own natural aversion response

Laser Safety

Class 3R	The laser beams from these products exceed the maximum permissible exposure for accidental viewing and can potentially cause eye injuries, but the actual risk of injury following a short, accidental exposure, is still small.
Class 3	Class 3B lasers may have sufficient power to cause an eye injury, both from the direct beam and from reflections.
Class 4	Have an output power greater than 500 mW (half a watt). There is no upper restriction on output power. Capable of causing injury to both the eye and skin and will also present a fire hazard if sufficiently high output powers are used.

Laser Safety

- In “enterprise” communications equipment, lasers more powerful than Class1 are rarely encountered (but always check !).
- Class 3 lasers are sometimes encountered in long-haul, DWDM carrier networks.

Radio

- Wireless LAN
- No cables
- High frequency radio signals
- Each workstation has a transceiver / antenna
- Also includes mobile phone technology, microwave transmission, satellite for longer distances
- Expensive, history of poor security (now down to ignorance, strong encryption available now),
- Susceptible to interference
  - More on this later

Transmission Errors

- Errors due to following factors
  - Thermal noise.
  - Impulse noise.
  - Signal distortion.
- Impulse is worst offender.
- Must provide methods of error detection and correction.

Cyclic Redundancy Code CRC

- Error Detecting Code (not correcting!)
- Nomenclature
  - A string of N bits may be represented as an N-1 degree polynomial with co-efficient of 0 or 1.
  - F = 110001
  - $F(x) = x^5 + x^4 + x^0$
- Modulo 2 arithmetic.
  - Addition and Subtraction identical… XOR

CRC Algorithm

- Sender & receiver agree on G(x) generator polynomial.
- Append R 0 bits to M(x), the message, where R is the degree of G(x), this yields
  - $x^R * M(x)$
- Divide G(x) into $x^R * M(x)$.
- Add remainder to $x^R * M(x)$, result T(x)

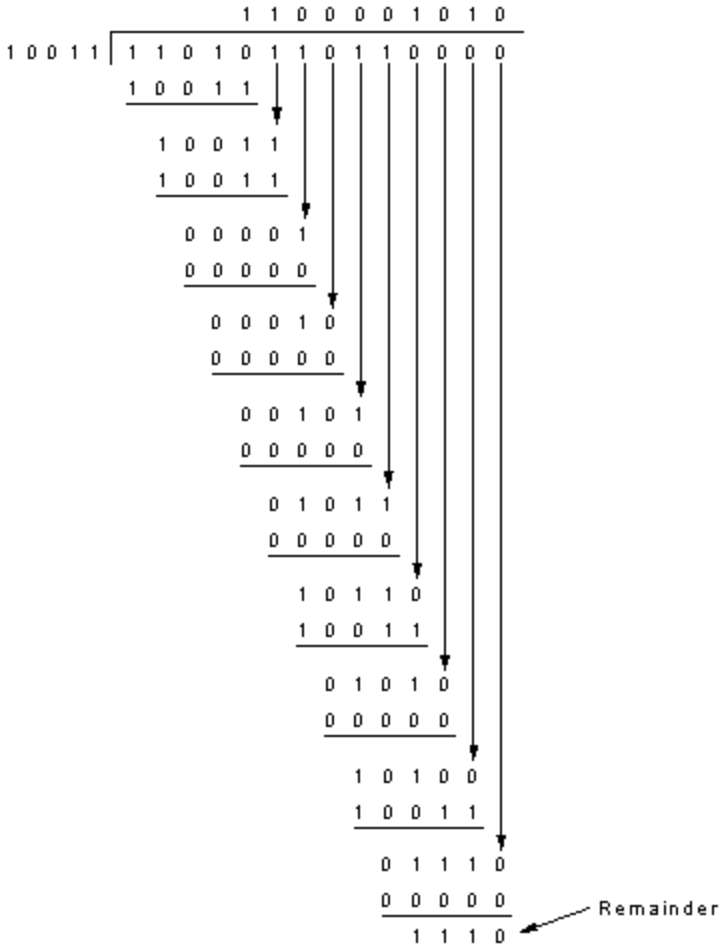
CRC Example

- M(x) = 1101 0110 11
- x^R * M(x) = 1101 0110 1100 00
- G(x) = 10011 *or* $x^4 + x^1 + x^0$
- Remainder should be 1110
- T(x) transmitted message should be
  - 1101 0110 11 1110
- This will be evenly divisible by G(x) !

Frame : 1 1 0 1 0 1 1 0 1 1

Generator: 1 0 0 1 1

Message after appending 4 zero bits: 1 1 0 1 0 1 1 0 0 0 0



Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 0

Standard CRCs

- CRC-12
  - $x^{12} + x^{11} + x^3 + x^2 + x + 1$
- CRC-16
  - $x^{16} + x^{15} + x^2 + 1$
- CRC-CCITT
  - $x^{16} + x^{12} + x^5 + 1$
- CRC-32
  - $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Signalling AM, FM, PM

- Modems use analogue signals down a telephone line
- Digital signals must be converted to analog signals for this
- Nyquest pointed out that it is not possible to merely keep increasing the sampling rate, so even with perfect 3000Hz line, no point in sampling faster than 6000Hz

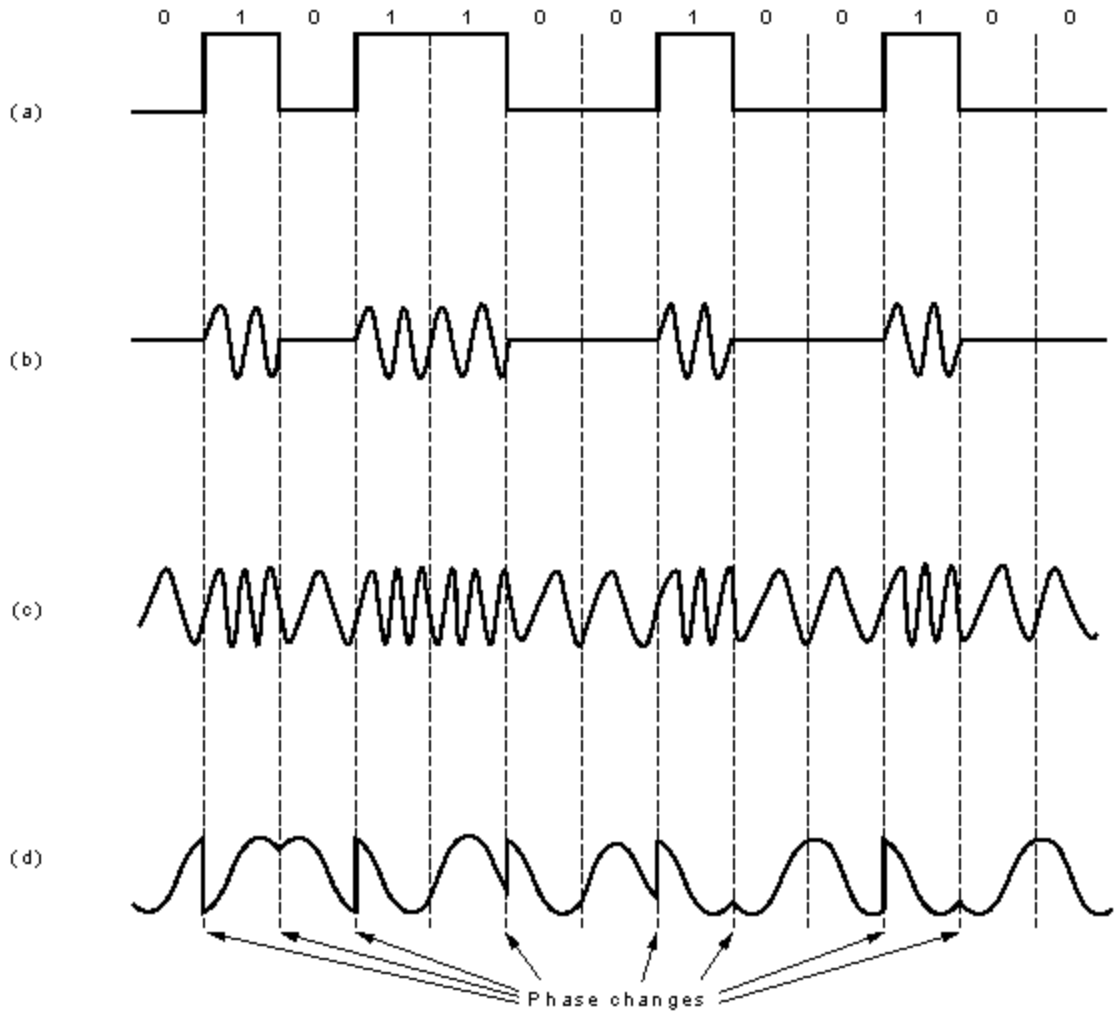


Fig. 2-18. (a) A binary signal. (b) Amplitude modulation. (c) Frequency modulation. (d) Phase modulation.

Digital Encoding

- NRZ- L and NRZ- I
- Manchester
- Differential Schemes
- Differential Manchester
- 4B\5B
- Issues of efficiency and clocking.

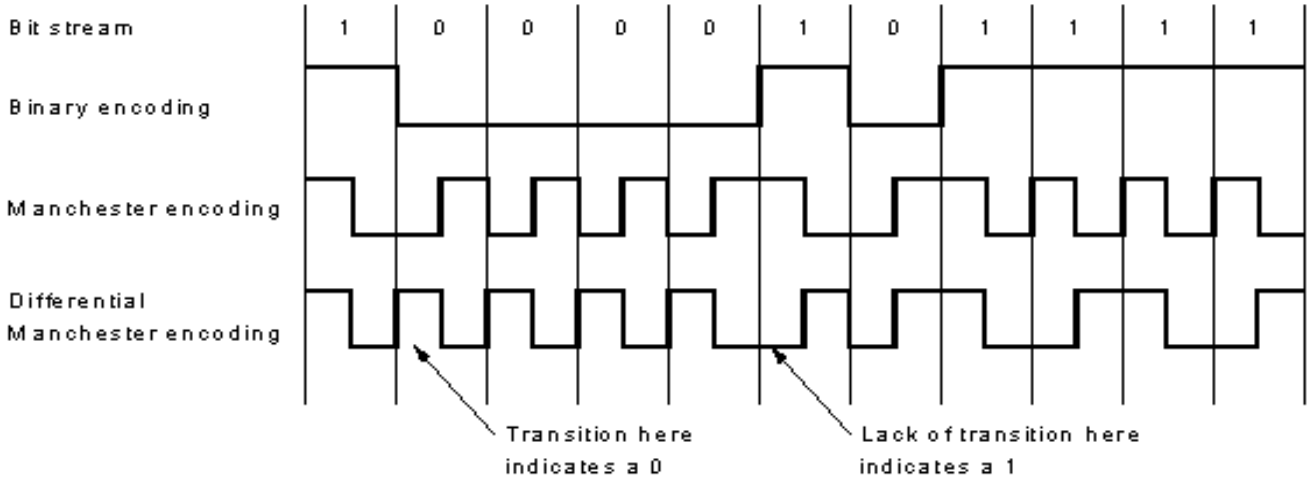


Fig. 4-20. (a) Binary encoding. (b) Manchester encoding. (c) Differential Manchester encoding.

Encoding (Notes)

- NRZ-L is used for short connections (RS232) but not for longer connections. Positive denotes a ‘0’ and negative denotes a ‘1’. (L refers to level).
- **NRZ-I (Inverted)** is a differential scheme where a transition denotes a ‘1’, and no transition denotes a ‘0’. With differential coding schemes a signal is decoded by comparrison of the polarity of adjacent signal elements, rather than determining the absolute value of a signal element.

- An advantage of this scheme is that it may be more reliable to detect a transition, in the presence of noise, rather than to compare a value to a threshold. Differential encoding on a twisted pair medium is also immune to the wires being crossed as the thresholds are not being examined, but the transitions are. After all a transition from +'ive to -'ive is just as much a transition as from -'ive to +'ive.
- There is a requirement for clocking information to be embedded in the data. One technique which does this is called *Manchester Encoding*, and a variation on it is called *Differential Manchester Encoding*. These schemes are called **biphase codes**.
- In the binary encoded signal there is no clock information, i.e. nothing to differentiate repeating digits.

- In **Manchester Encoding**, each bit period is divided into two equal intervals, thus the name **biphase**. There is always a transition between these intervals (thus clocking). A binary ‘0’ is represented as having the first interval set high and the second interval set low. A binary ‘1’ is the reverse -- the first interval is low and the second high.
- Advantage: always a transition in each bit, thus making synchronisation between sender and receiver possible.
- Disadvantage: requires twice as much bandwidth as plain binary coding.
- Differential Manchester Encoding scheme distinguishes 1’s and 0’s by using a transition at the start of a period to indicate a ‘0’ and no transition to represent a ‘1’. A transition in the middle of the period between the two intervals is still used to help provide clocking information, just as in Manchester Encoding.

- This differential scheme is more complex to operate but is more immune to noise. An error must invert the signal before and after expected transitions to cause undetected errors. It also requires twice the bandwidth of ordinary binary encoding.
- In differential encoding schemes, it is the *transition* from one voltage level to another that distinguishes bit values, not the *voltage levels*. This makes the coding scheme more immune to noise.
- If there are two wires carrying a signal from one device to another and the wires are accidentally confused, then this does not affect the interpretation of the data as the transitions in voltage levels will still be correctly interpreted. The signal levels are not important, only the transition from one state to another.

4b\5b Coding

- Greater efficiency may be achieved than that of Manchester Codes (50%). One such better scheme is 4B/5B.
- Each four bits are coded in a symbol with 5 *cells* where each cell contains a single signal element. So each group of four bits (nibble) is encoded as 5 bits. This raises the efficiency to 80% for this scheme, as opposed to Manchester encodings 50%.
- Synchronisation is achieved by including a second stage of encoding. Each cell of the 4B/5B stream is treated as a binary value itself and encoded using a NRZI scheme. This takes on the advantages of differential encoding.

- Only 16 of the possible 32 bit patterns generated from 5 bit code are required to represent the 4 bit data patterns.
- The 5B codes selected to represent the 16 4-bit data blocks are such that a 1 is present at least twice for each 5-cell (or 5-bit) code.
- This minimum transition rule aids synchronisation by including an element of clocking.
- 4b5b Coding is utilised in 100 Mb Ethernets

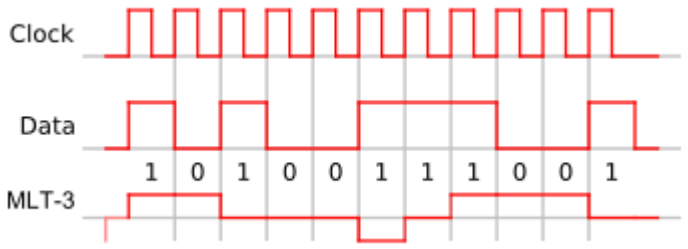
100 Mbps 4B5B Code

- 4-bit 5-bit
- 0 0000 11110
  1 0001 01001
  2 0010 10100
  3 0011 10101
  4 0100 01010
  5 0101 01011
  6 0110 01110
  7 0111 01111
  8 1000 10010
  9 1001 10011
  A 1010 10110
  B 1011 10111
  C 1100 11010
  D 1101 11011
  E 1110 11100
  F 1111 11101

MLT

- A 100 MHz data stream that uses a 4B/5B encoding results in a125 MHz signal.
- 125 MHz signal is itself encoded as a mult-level signal using three signal levels (instead of the two levels used in Manchester encoding).
- The three signal levels are -1, 0, +1.
- This reduces the bandwidth required on the physical cable to only 31.25 MHz, which is within the specification of the CAT5e cable used in UTP.

NRZ	0	0	0	I	0	0	I	0	I	I	I	0	I	0
MLT-3	0	0	0	+	+	+	0	0	-	0	+	+	0	0



8b/10b Coding in Gigabit Ethernet (fiber)

- The 10 bit code must contain either five ones and five zeros, or four ones and six zeros, or six ones and four zeros.
- The difference between the count of 1s and 0s in a string of at least 20 bits is no more than 2, and there are not more than five 1s or 0s in a row
- This prevents a sequence of too many consecutive ones and zeros, assisting clock synchronisation.

Framing Problem

- Communication protocols break data stream into frames or packets.
- Easier to multiplex several connections over one comms. channel.
- Problem…
  - Where does one frame begin and another end?

Solutions to Framing Problem

- Timed Delay: fixed delay between frames.
  - Delays may occur due to interrupts or breaks in transmission.
- Character count: include a header which contains length of frame, cover with a CRC.
  - If count becomes corrupted, may not find CRC and receiver loses synch.

Solutions to Framing Problem (cont.)

- Character stuffing: mark start of each frame with sequence DLE STX and end with DLE ETX (byte oriented).
  - What happens is DLE STX or DLE ETX occur naturally in the data? Solution - stuff it with another DLE! Receiver watches for double DLEs and destuffs them. Only true delimiters have a single DLE.

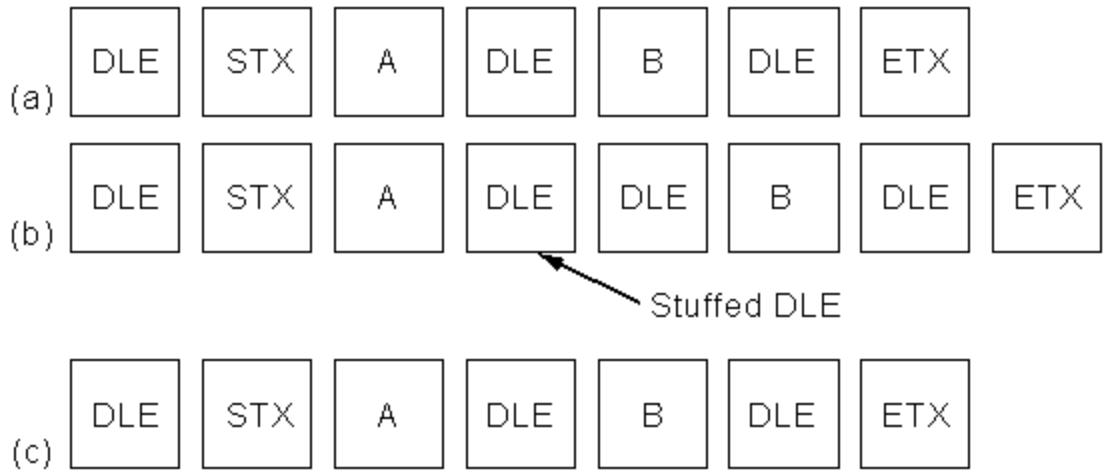


Fig. 3-4. (a) Data sent by the network layer. (b) Data after being character stuffed by the data link layer. (c) Data passed to the network layer on the receiving side.

Solutions to Framing Problem (cont.)

- Bit stuffing: Use unique bit sequence (bit oriented) such as 0111 1110 to mark start of frame.
  - If this occurs in data, stuff it with a 0 after 5 consecutive 1s!
  - Receiver de-stuffs five 1s followed by a 0, only 6 consecutive 1s is a real delimiter.

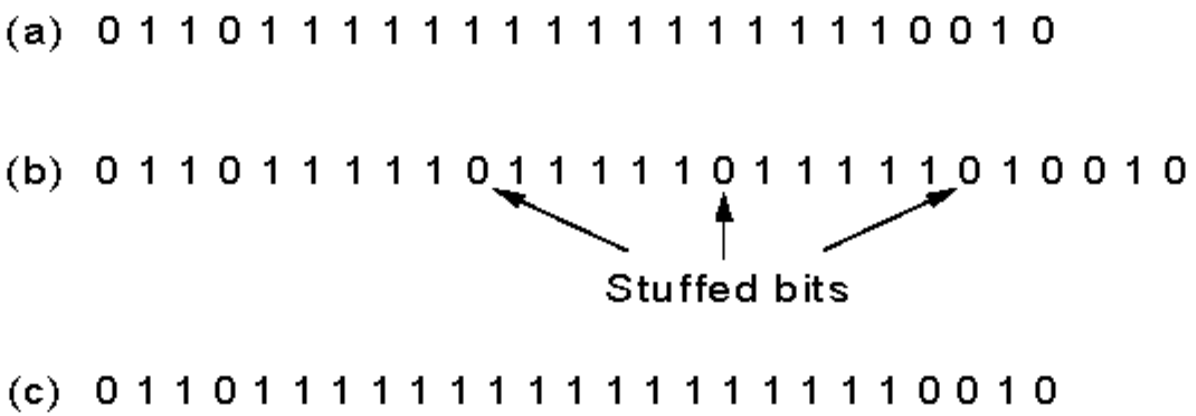


Fig. 3-5. Bit stuffing. (a) The original data. (b) The data as they appear on the line. (c) The data as they are stored in the receiver's memory after destuffing.

OSI Data Link Layer

Part of Network Access Layer of TCP/IP

Details Computer-to-Network Issues

Data Link Issues

- Currency of Data-Link is frames or packets.
- Issues to be addressed here are…
  - Framing
  - Error Control
  - Flow Control
- We have seen how frame and error detection are done, now we have to *do* something about lost or damaged frames.

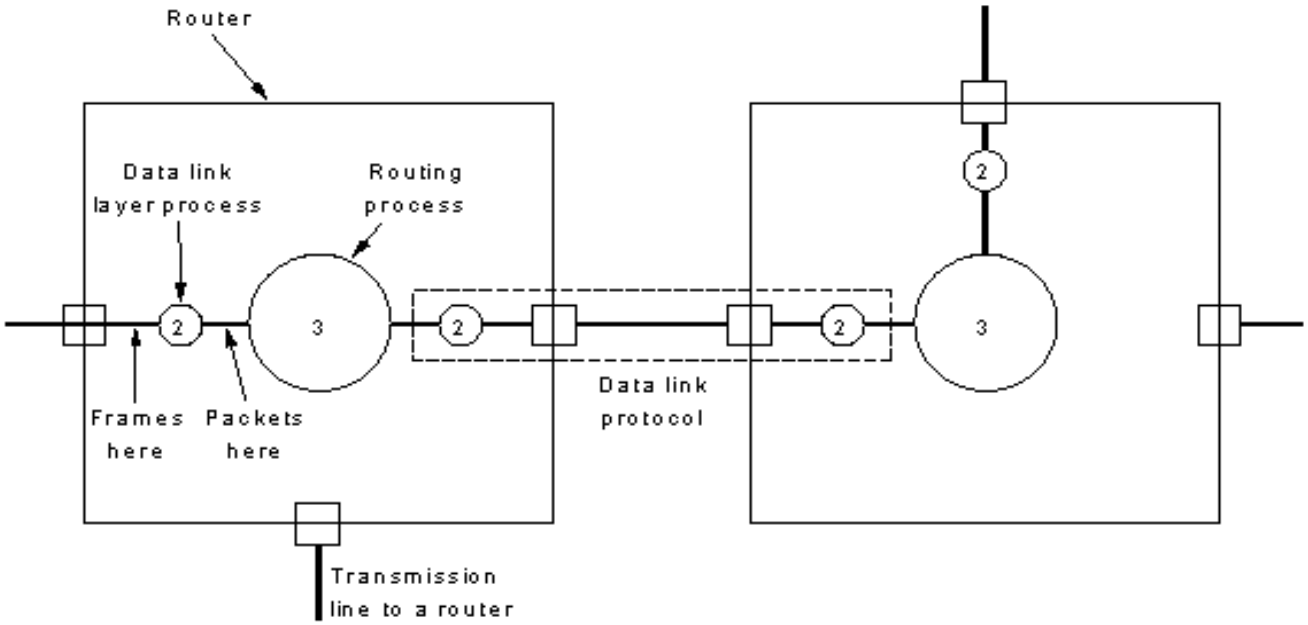
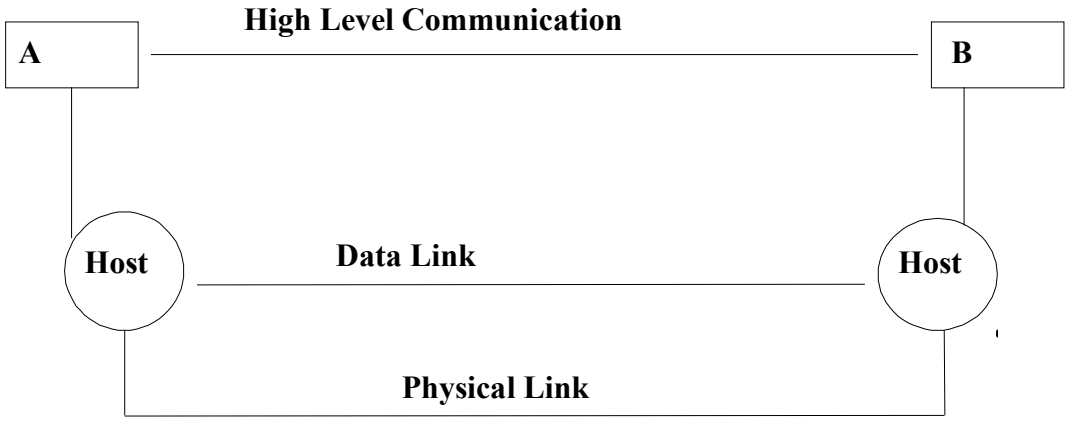


Fig. 3-2. Placement of the data link protocol.

The Need For Protocols

- Protocols are algorithms which will implement framing, flow control and error detection and correction.
- Makes error prone physical layer appear as error free to higher layers.
- Does so as efficiently as possible using valuable bandwidth.

Two Hosts Communicate



Idea for a Protocol

- To begin designing a protocol let us make the following *unrealistic assumptions* :
  - a) Simplex data transmission.
  - b) Transmitting and receiving hosts are always ready to transmit and receive data.
  - c) Processing time is negligible and infinite buffer space is available
  - d) Perfect error-free link

Utopia Protocol

- Make all assumptions, a, b, c, d.
- For this protocol the control header and checksum are unnecessary. The transmitting host simply takes packets from host *A* (which always has one ready) and pumps them as fast as it can onto the physical link.
- The receiver accepts the frames and passes them straight to host *B*.

Stop-and-Wait

- Drop assumption c); processing time ≠ 0; buffer ≠ ∞ ; not simplex
- In practice the receiving host needs time to process incoming frames, has only a finite amount of buffer space to queue processed frames i.e. the receiver needs to be able to prevent the sender from flooding it with data faster than it can handle it. Some form of handshaking is required.

Stop-and-Wait (cont.)

- In a simple stop-and-wait protocol, the receiver sends an acknowledgement frame back to the sender after delivering the packet to the host. Only after receiving the *control frame* will the sender fetch and transmit the next packet.

Positive Acknowledgement with Retransmission PAR

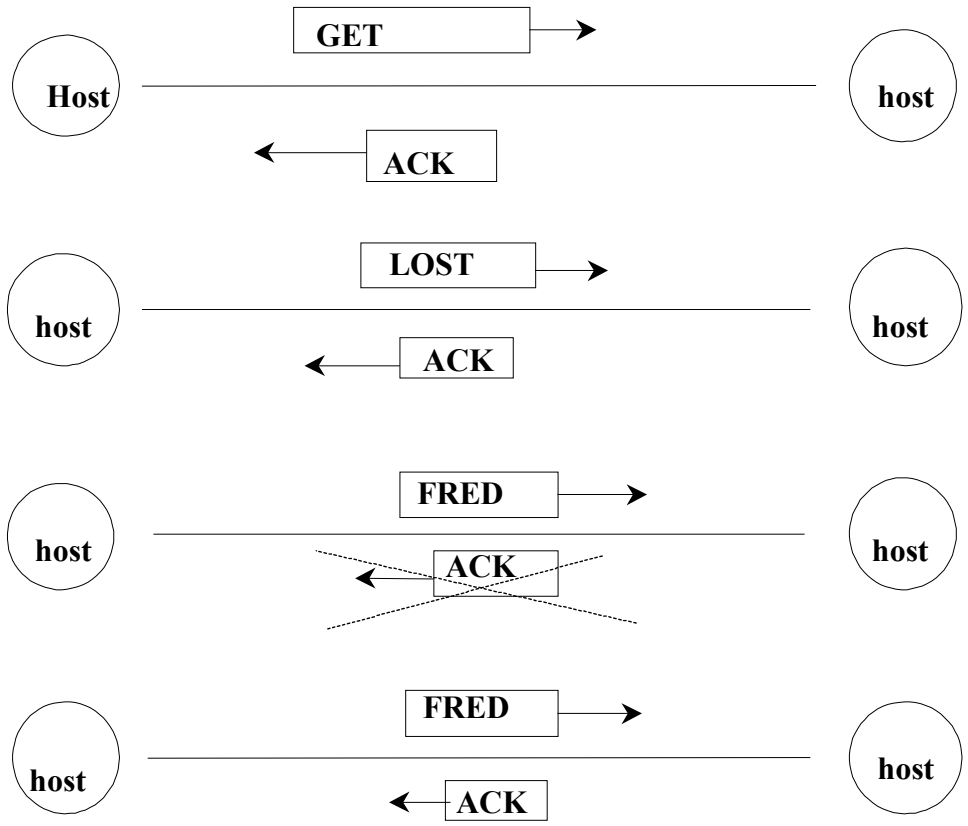
- Noisy channel simplex protocol: drop some assumptions
  - c) [processing time, buffers pace] and
  - d) [error free link]
- With error prone physical link, frames may be either damaged or lost completely.
- However,
  - Damaged frames can be detected by the checksum.
  - Lost frames will not be acknowledged. Eventually the sender will tire of waiting for an acknowledgement, timeout, and retransmit the frame.

Bright Idea for a Protocol

- Same as Stop-and-Wait, except that
  - damaged frames are not acknowledged,
  - causing a timeout and subsequent retransmission.

Lets Examine a Protocol

- Suppose the message “Get Lost Fred” is being sent from *A* to *B*, one word per packet


GET
Host
host
ACK
LOST
host
host
ACK
FRED
host
host
ACK
FRED
host
host
ACK

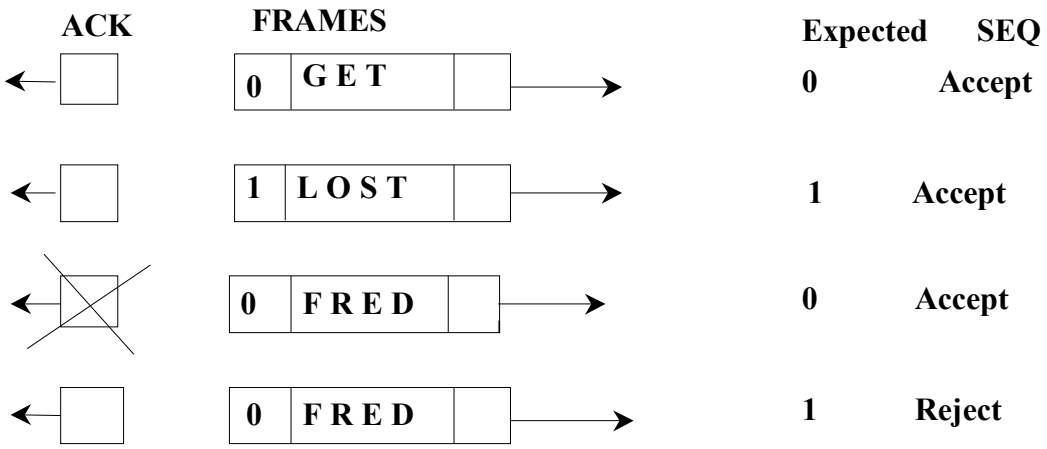
Get Lost Fred

- *B* receives “Get Lost Fred Fred” and the protocol has failed !
- Basically what has happened is that the receiver has accepted a duplicate frame.
- The solution is to use ***SEQ***, a sequence number in the control header to differentiate between frames and allow duplicates to be discarded.

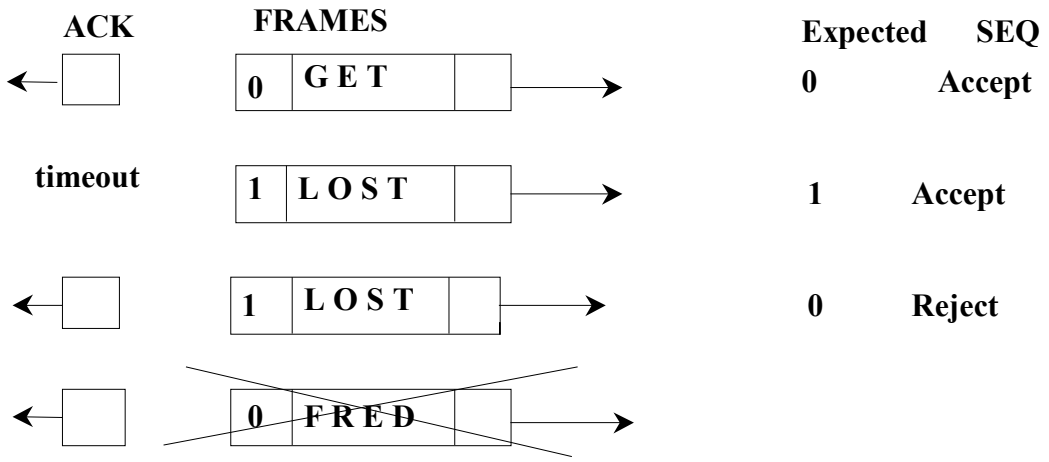
Get Lost Fred!

- As a frame must be acknowledged before the next one is sent, a one-bit sequence number [0,1] is sufficient.
- The receiver will expect alternatively numbered frames (0 1 0 1 0 1 ... etc.).
- Any frame with the wrong sequence number is rejected as a duplicate
  - (but still acknowledged).

Get Lost Fred



Premature timeouts

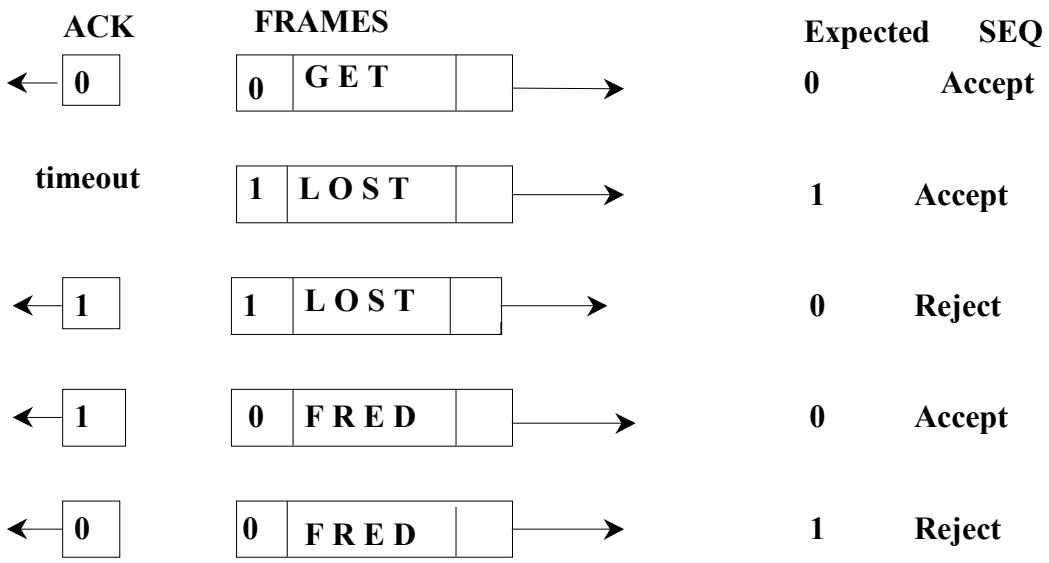


- The message “Get Lost” is received and the protocol has failed
- DOH!

Solution

- Include in the **ACK** field of the acknowledgement control frame the **SEQ** number of the last frame *received without error*.
  - Then if this number (0 or 1) differs from the transmitted frame, the sender transmits it again.
- The resulting PAR simplex protocol will now work in the face of any combination of
  - garbled frames,
  - lost frames and
  - premature timeouts.

Solution


ACK
FRAMES
Expected
SEQ
0
0
G E T
0
Accept
timeout
1
L O S T
1
Accept
1
1
L O S T
0
Reject
1
0
F R E D
0
Accept
0
0
F R E D
1
Reject

Summary

- After transmitting a frame and starting the timer, Host *A* waits for a response. There are three possibilities:
  1. an acknowledgement frame arrives undamaged
  2. a damaged acknowledgement arrives, or
  3. the timer goes off.

Summary (cont.)

- If a valid ACK comes in, *A* fetches the next packet and puts it in the buffer
  - overwriting the previous packet, and advancing the sequence number.
- If a damaged frame arrives or no frame at all arrives,
  - neither a buffer nor the sequence number are changed, so that a duplicate can be sent.

Bi-directional PAR

- The control header of all frames contains the three fields *KIND*, *SEQ* and *ACK*.
- Sending data packets/acknowledgements in both directions is no problem –
  - by looking at the *KIND* bit in the header, the receiver knows which it is dealing with.
- However, this would be inefficient.

Bi-directional PAR (2)

Consider an Host *B* which is about to acknowledge a data frame received from Host *A*, and also about to send off a data frame to *A*.

- Instead of sending two frames, the acknowledgement can hitch a lift on the data frame, using the ACK field in the header.
- This is called ***piggybacking***.

Bi-directional PAR (3)

- As we are still making assumption b)
  - [*Transmitting and receiving hosts always ready to transmit & receive data*]
- All acknowledgements can be piggybacked. Thus, data packets are bounced back and forth between *A* and *B*.
- Notes:
  - For the protocols considered so far, only one frame is *in the wire* at any one time.
  - The sender needs to keep a copy of each frame in a buffer for possible retransmission until the frame has been successfully acknowledged.

Bi-directional PAR (4)

- Assumption b)
  - [*Transmitting and receiving hosts always ready to transmit & receive data*]
- Easily dealt with.
  - If there is no outgoing data frame, the host will wait a short while to see if one comes along to provide a piggyback.
  - If not, a separate acknowledgement frame will be sent.
  - It must not wait too long to avoid unnecessary duplicates being sent due to the sender timing-out.

Bi-directional PAR (5)

- Up until now, lost and damaged frames have been dealt with in the same way.
  - No *ACK* is sent, leading to timeout and retransmission. The timeout period is usually set quite long in order to avoid complications caused by premature timeout.
  - This is inefficient, as while the timeout is expiring, the link is not being used.
- A partial solution is *NAK*, a negative acknowledgement.
  - This is sent immediately a damaged frame is received and elicits immediate retransmission.

Bi-directional PAR (6)

- The *NAK* may also be piggybacked.
- If the *NAK* is damaged, no harm is done as the sender will eventually timeout and retransmit as before.
- A damaged frame is *Nak*'d only once.

The Long Haul

Data Transfer Protocols in the Internet

Pipelining

- When propagation delay is not negligible, these previous methods are wasteful of bandwidth.
- The solution is to '*fill up the pipe*'. However, doing this entails sending off frames before ACKs for previous frames have arrived.

Sliding Window Protocol

- Each outbound frame is given a sequence number in the range of 2^n-1 using an n-bit field, e.g. if n=1, then range is 0.....1 as in ABP or PAR protocols.
- Both sender and receiver keep *windows* informing them of which frames can be validly sent and which validly received.

Rules

- **Sender** :- The upper edge of sender is advanced when a frame is sent (up to max. window size). The lower edge advanced when ACK received for lowest numbered frame in the window.
- **Receiver** :- Both edges are advanced when the lowest numbered frame in window is correctly received and ACK sent.

Notes

- Buffering requirements at both sender and receiver depend on the size of the sending and receiving windows respectively.
- Each transmitted frame has its own separate timeout clock.

Notes (cont.)

- In these protocols, an acknowledgement for frame *N* is accepted as acknowledging all transmitted frames numbered up to *N* (counting circularly).
- Thus, if ACK(0) and ACK(1) were both destroyed, but ACK(2) now arrives, it implicitly acknowledges 0 and 1 also.

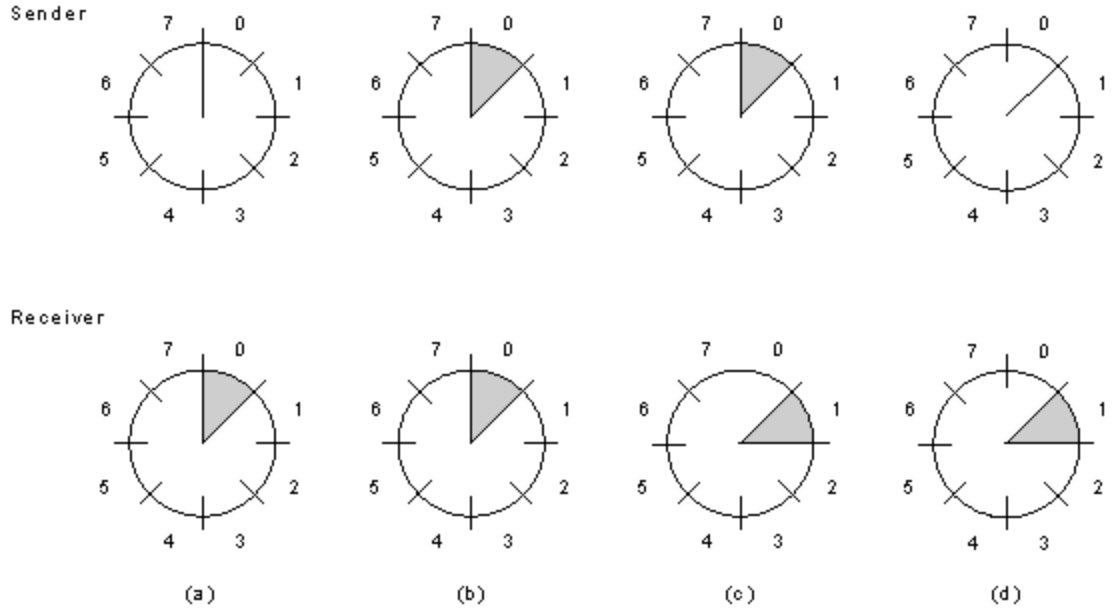
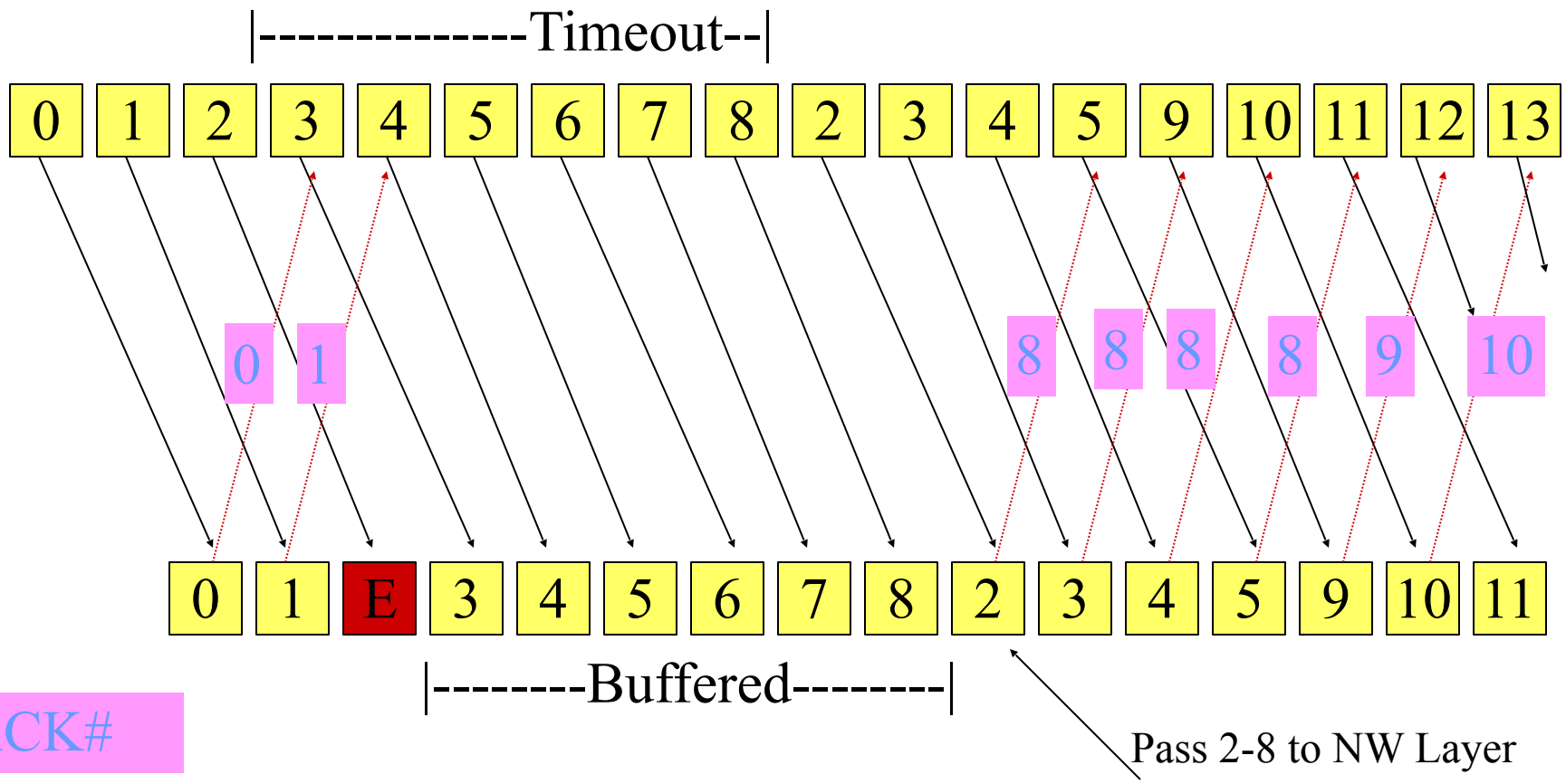


Fig. 3-12. A sliding window of size 1, with a 3-bit sequence number. (a) Initially. (b) After the first frame has been sent. (c) After the first frame has been received. (d) After the first acknowledgement has been received.

Example Session with Recovery



The Stopping Problem

- A Data-Link cannot be stopped.
- Consider a session termination.
- Neither terminal knows that other has sent *last* packet, the *last* packet must be ACK'd.
- In practice the data-link is dropped after the link is sensed as being dead for a prolonged period.

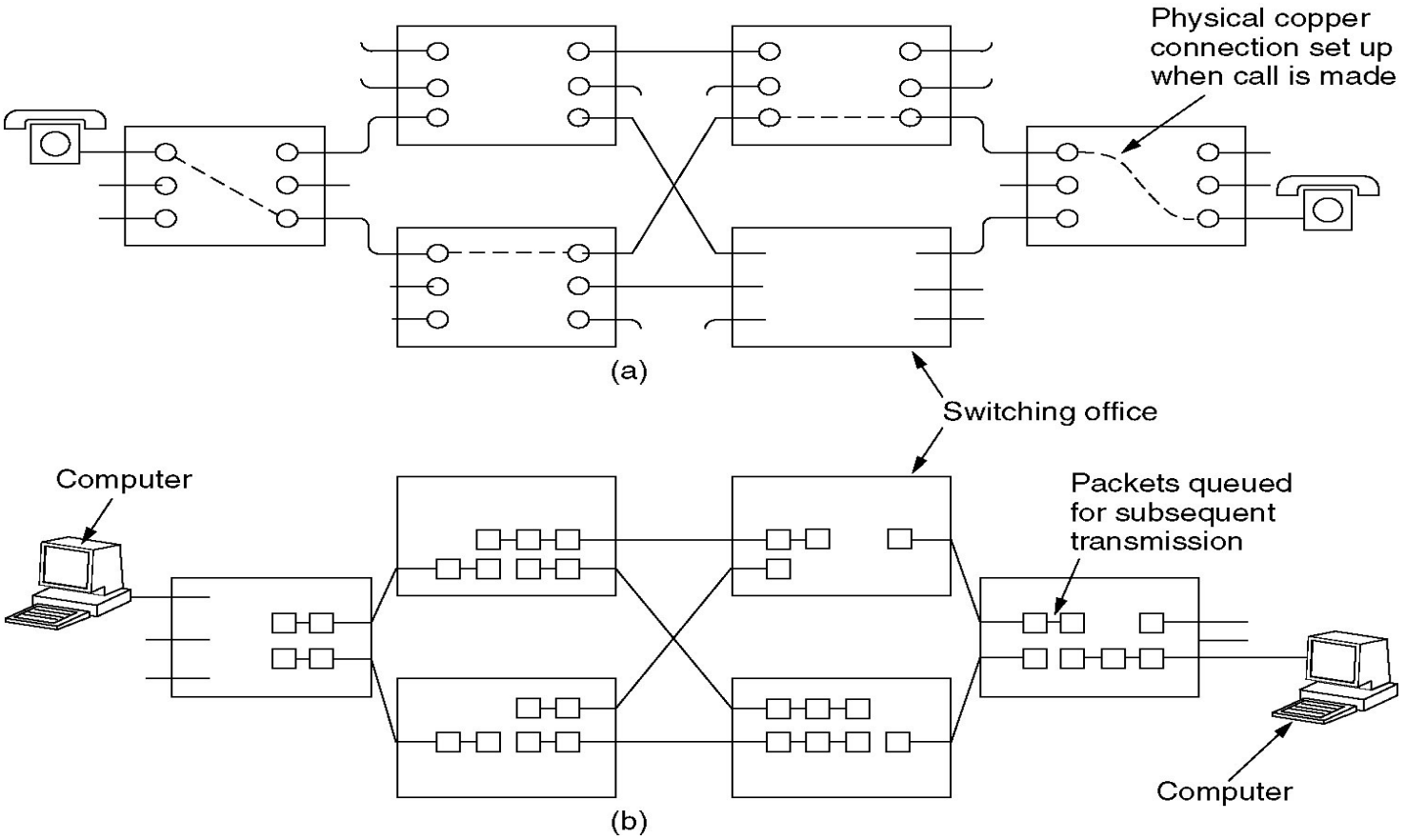
Switching

- Packets must be sent from host to host across a directed network.
- Three types of switching are employed.
  - Circuit Switching
  - Message Switching
  - Packet Switching

Circuit Switching

- Like old fashioned terrestrial telephone system.
- Try to form dedicated physical path from source to destination.
- Path remains dedicated until session is terminated.
- Not typical operation of bursty comms.

Circuit Switching



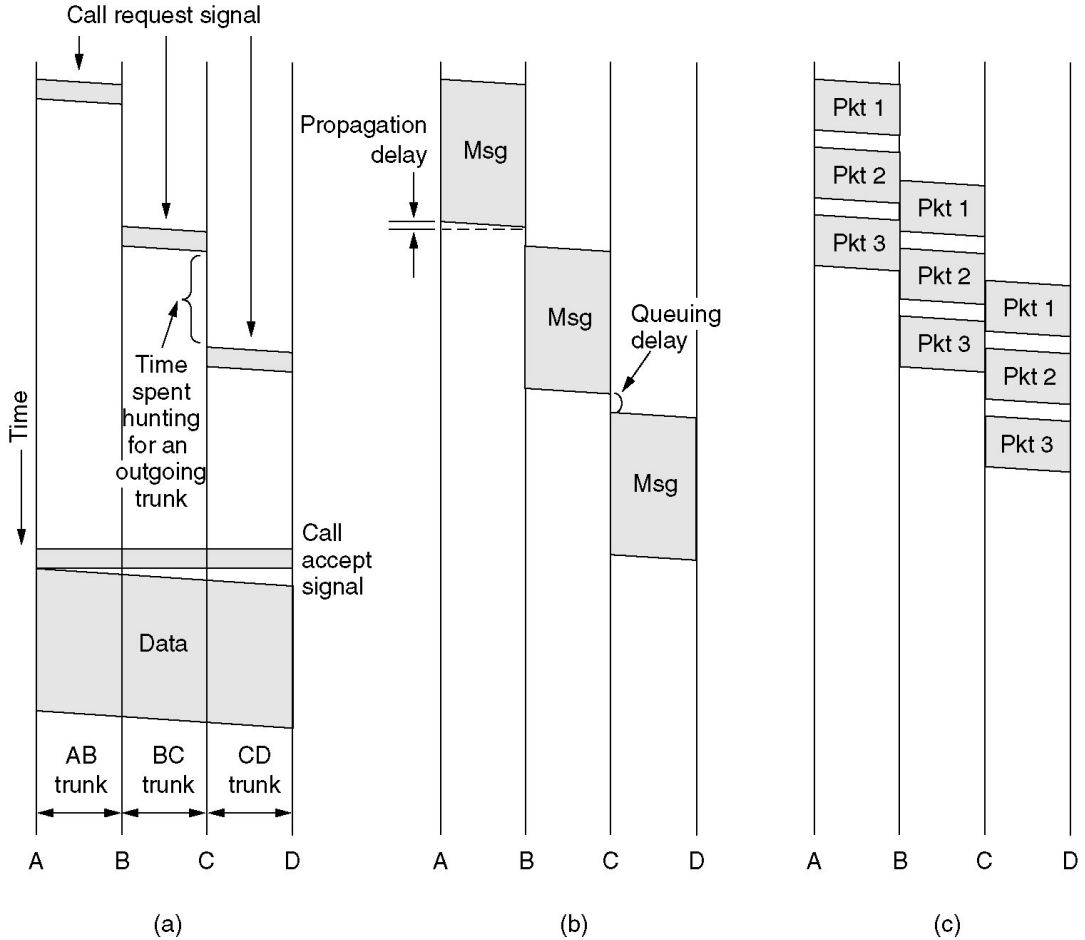
(a) Circuit switching.

(b) Packet switching.

Message Switching

- No physical path established.
- Large bursts of data transmitted from sender to receiver.
- Each burst stored and forwarded from host to host throughout network.
- No limit to burst size, may encounter memory\buffering and link availability problems.

Message Switching



(a) Circuit switching (b) Message switching (c) Packet switching

Packet Switching

- Upper limit set on size of blocks to be transmitted.
- Ideal for bursty computer communications.
- May utilise pipelining to improve throughput.
- Large packet size will emulate message switching, small emulates circuit switching.

Packet Switching

Item	Circuit-switched	Packet-switched
Call setup	Required	Not needed
Dedicated physical path	Yes	No
Each packet follows the same route	Yes	No
Packets arrive in order	Yes	No
Is a switch crash fatal	Yes	No
Bandwidth available	Fixed	Dynamic
When can congestion occur	At setup time	On every packet
Potentially wasted bandwidth	Yes	No
Store-and-forward transmission	No	Yes
Transparency	Yes	No
Charging	Per minute	Per packet

A comparison of circuit switched and packet-switched networks.

Local Area Networks 802.X

Local Area Networks and 802

- IEEE formulated 802 standard for LAN.
- ITU (CCITT) adopted 802 as 8802
- Common media types are UTP and Co-axial cable.
- Topologies may be Ring\ Bus\ Star or Wireless.

Organisation of 802

- Layered within the Data-link and Physical layers of OSI protocol stack.
- Composed of
  - Physical Medium Dependent (PMD) layer.
  - Medium Access Control (MAC) layer.
  - Logic Link Control (LLC) layer.

802 Standards

- **802.2 LLC (HDLC based)**
- **802.3 CSMA/CD Bus (Ethernet)**
- 802.4 Token Bus
- 802.5 Token Ring
- 802.6 DQDB
- 802.7 Broadband LAN using Coaxial Cable (disbanded)
- 802.8 Fiber Optic TAG (disbanded)
- 802.9 Integrated Services LAN (disbanded)
- 802.10 Interoperable LAN Security (disbanded)
- **802.11 WiFi**
- 802.12 demand priority (disbanded)
- 802.13 Not used (officially)
- 802.13ah Defines "Copper for the first mile" for Metro Area Networks (proposed)
- 802.14 Cable modems (disbanded)
- **802.15 Wireless PAN**
- **802.15.1 Bluetooth certification**
- **802.15.2 coexistance of 802.15 and 802.11**
- **802.15.1 (Bluetooth certification)**
- **802.15.4 (ZigBee certification)**
- 802.16 Broadband Wireless Access (WiMAX certification)
- 802.16e (Mobile) Broadband Wireless Access
- 802.16.1 Local Multipoint Distribution Service
- 802.17 Resilient packet ring
- 802.18 Radio Regulatory TAG
- 802.19 Coexistence TAG
- 802.20 Mobile Broadband Wireless Access
- 802.21 Media Independent Handoff
- 802.22 Wireless Regional Area Network

Ethernet Networks

Implementing 802.3

Ethernet

- May operate over several cable types.
  - **10 Base 2** Thin wire coax, bus topology.
  - 10 Base 5 Thick wire coax, bus topology.
  - 10 Base T Twisted pair, star topology.
  - 10 Base F Optical fibre, star topology.
  - **100BASE-TX** fast Ethernet over 100Mbps 802.3u
  - 1000BASE-T Gbit/s Ethernet over twisted pair
  - Today many types of Gbps versions over fiber depending on type of lasers used.

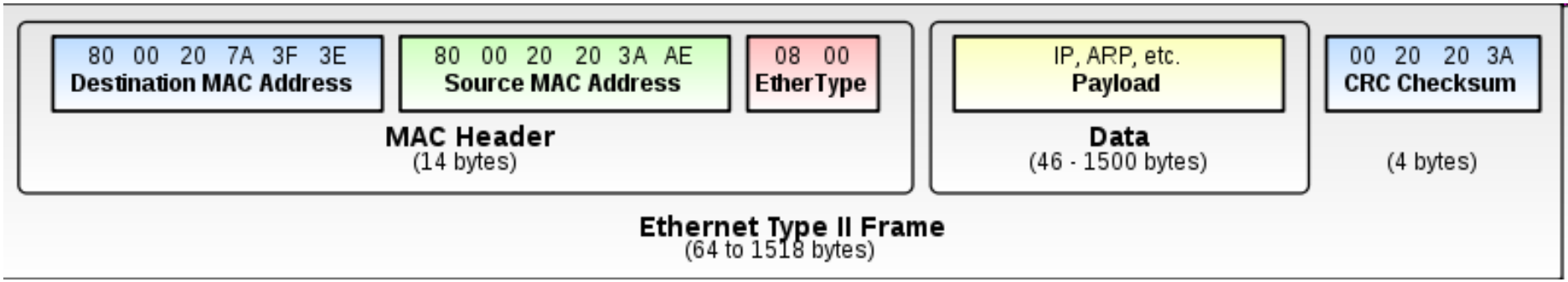
802.3 Frame Format

Preamble	SD	Dest Addr	Src Addr	LD	Data	Pad	CRC

- Preamble (7 bytes): Sine wave, clock synch.
- SFD (1 byte): 10101011 denoted.
- Dest Addr: 6 byte unique 802 address.
- Src Address: 6 byte address, 2^{48} possible.
- LD: Size of payload.
- Data: Payload max 1500 bytes.
- Pad: Ensures min size of 64 bytes.
- CRC: As discussed previously.

Ethernet II or DIX frames

- Defines the 2 octet Type field (LD previously), defining the upper layer protocol encapsulating the frame data
  - 0x0800 indicates IP V4
  - 0x0806 is ARP
  - 0x06DD is IP V6
  - Must be greater than 0x0600 (1,536 decimal, > 0x05DC or 1500_{10} the max payload of Ethernet)



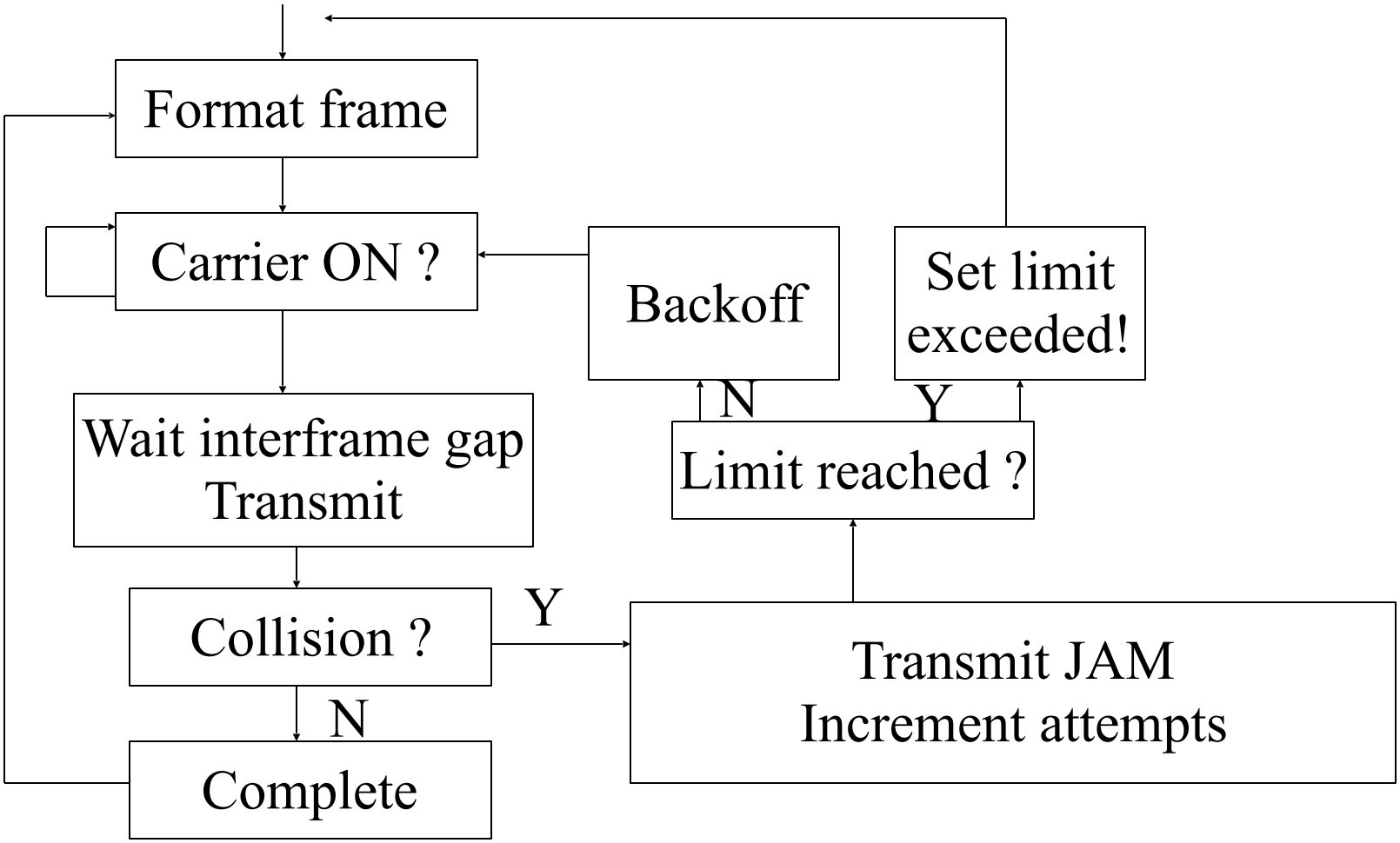
Coexistence of Ethernet & Ethernet II

- Both types can exist on the same Ethernet network.
- Distinguish V1 and V2 by value in type field
- For V2, value in type field must be >=$1{,}536_{10}$ or 0x600
  - Maximum payload for Ethernet is 0x05DC or 1500_{10}
- For V1, value must be <= 1500_{10} or 0x05DC

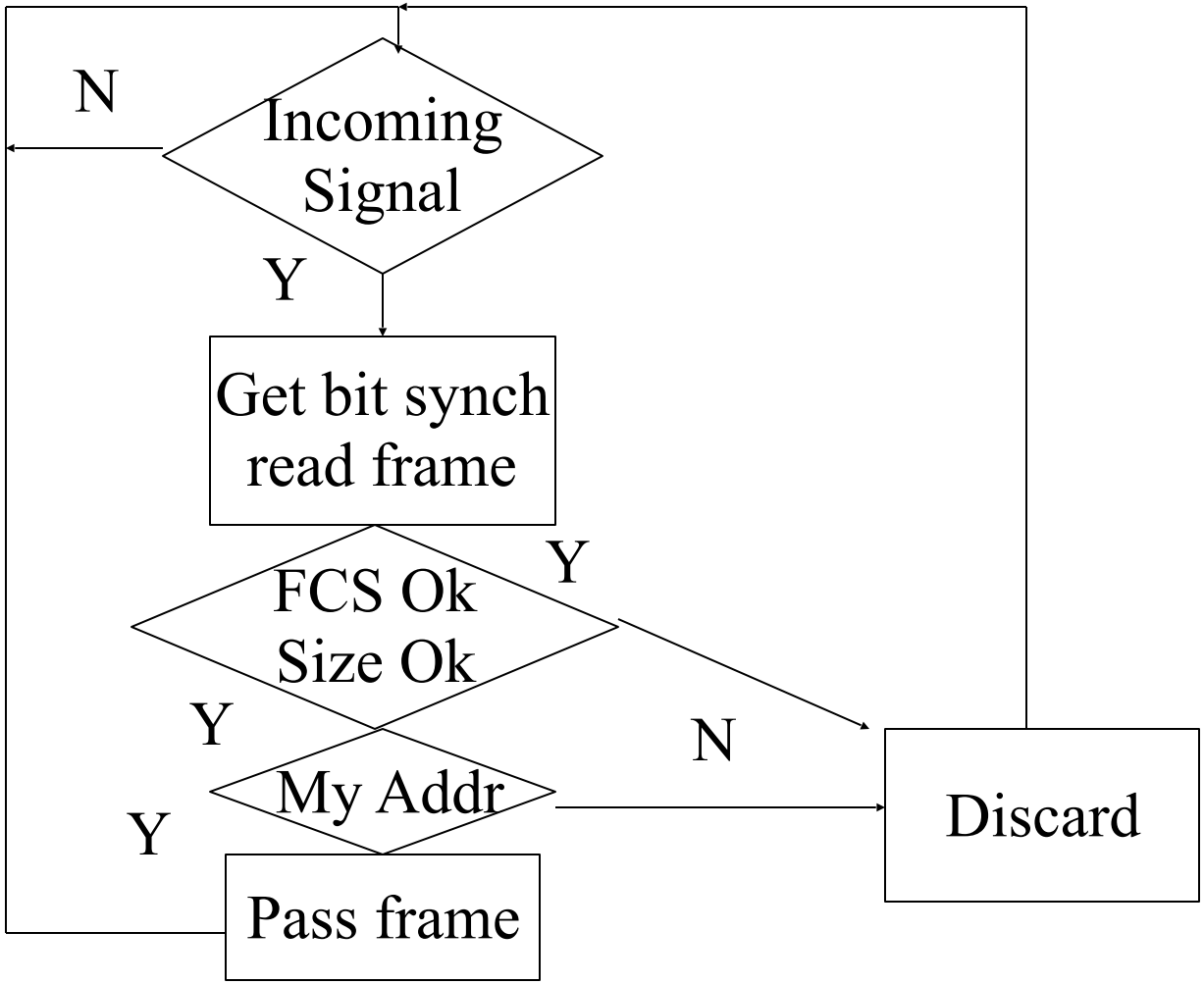
802.3 MAC

- Carrier Sense Multiple Access with Collision Detection CSMA\CD
- Allows multiple stations to share transmission medium.
- Senses carrier to see if medium is quiet.
- Be able to detect if another station is interfering by continuing to listen to carrier while transmitting.

802.3 MAC Sender Operation



802.3 MAC Receiver Operation



Truncated Binary Exponential Backoff

- When collision is detected, two stations wish to transmit simultaneously.
- Need to prevent continuous collisions between this pair.
- Better to have graceful degradation of throughput.

Algorithm

- The number of slot times before the N^{th} retransmission attempt is chosen as a uniformly distributed random integer in the range
  - $0 \leq R \leq 2^K$,
  - where K = min(N, backoff limit),
  - e.g. for a backoff limit of 20, possible ranges of K will be 0..2, 0..4, 0..8, 0..16, 0..20, 0..20, 0..20 for successive attempts at retransmission up to a maximum number of attempts.
- The backoff limit of 20 is imposed and prevents the series continuing 8, 16, 32, 64, etc, etc and thus the heuristic is called *truncated binary exponential backoff*.

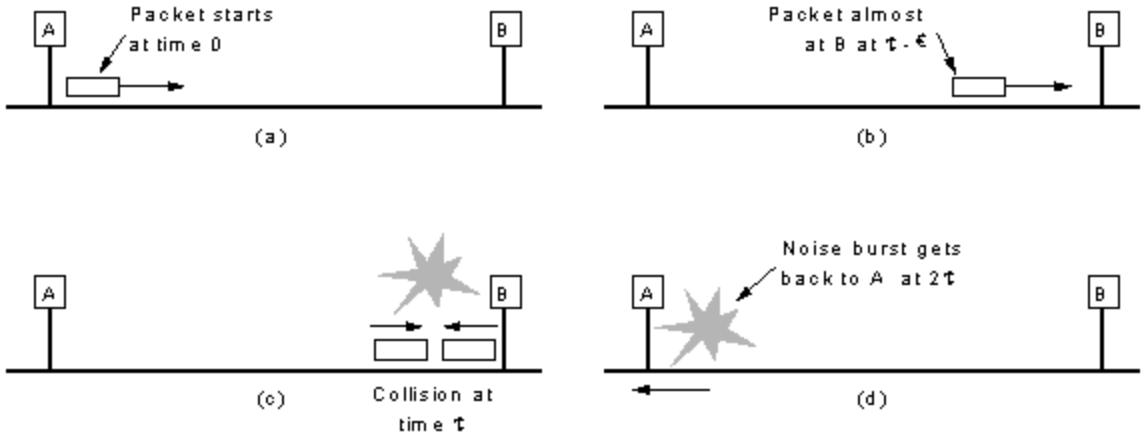


Fig. 4-22. Collision detection can take as long as 2τ.

802.3 Modern Implementations

- Most modern implementations of Ethernet use Switched Ethernet.
- There are almost no collisions
- Packet paths can cross over the switch without colliding, provided each “conversation” has no receivers in common
- Improved throughput and better utilisation.

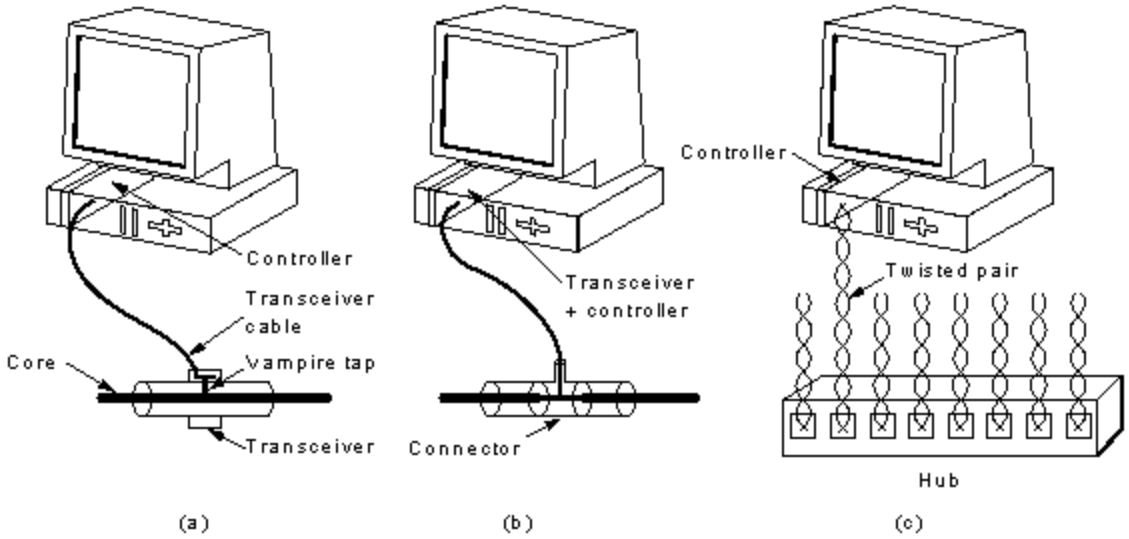
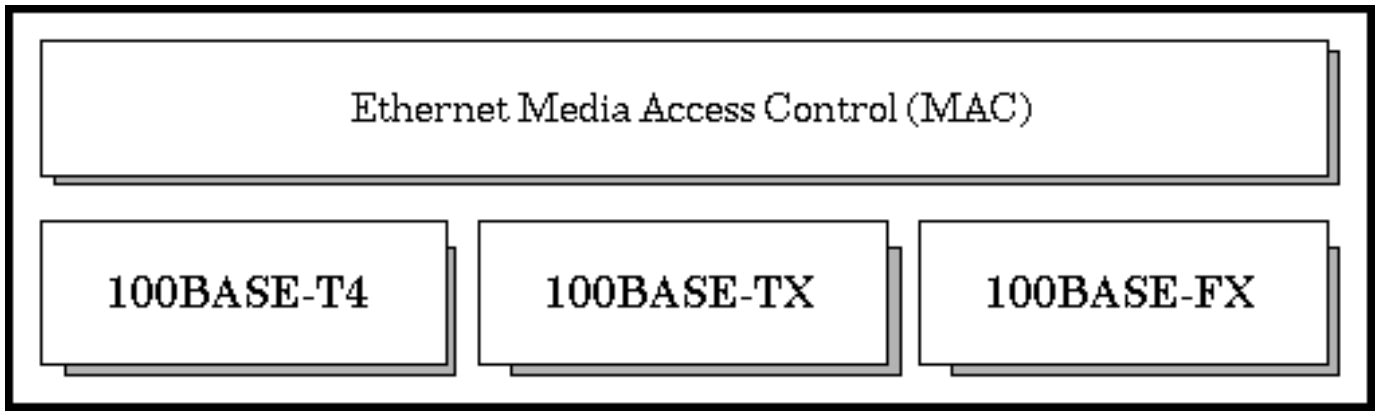


Fig. 4-18. Three kinds of 802.3 cabling. (a) 10Base5. (b) 10Base2. (c) 10Base-T.

Fast & Gigabit Ethernet

Fast Ethernet (100Mbps)

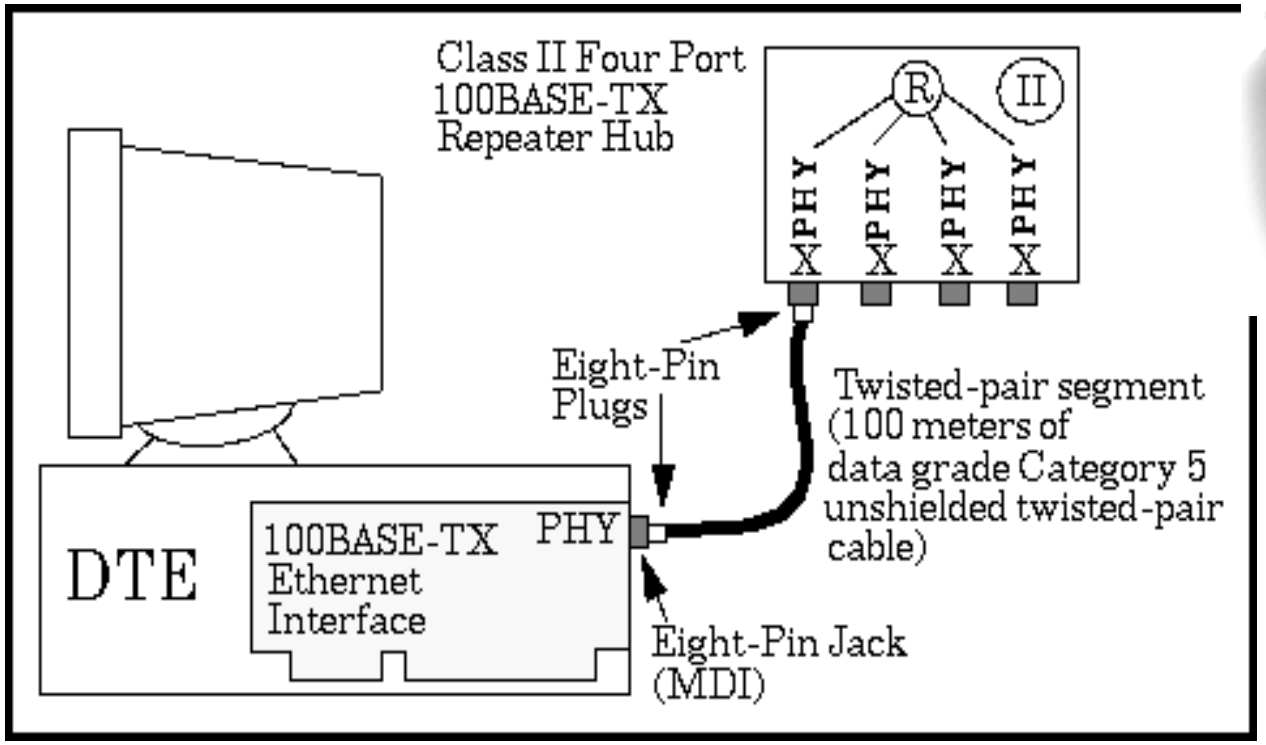
- Factor of 10 reduction in bit-time
- No changes to frame format, payload, MAC
- Ethernet card negotiates speed at interface

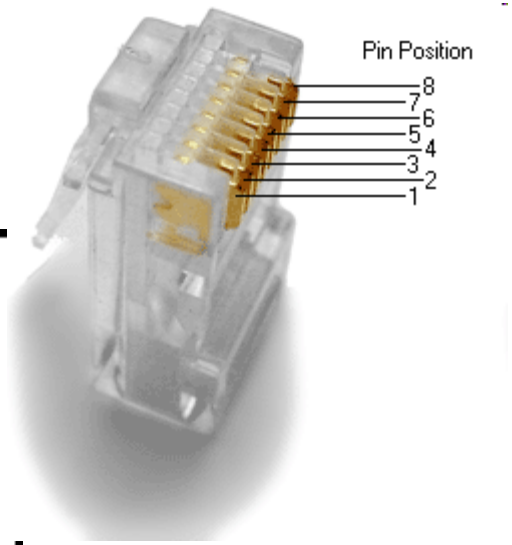


- 100Base-TX is the one used, all others obsolete
  - 100 is the speed,
  - Base is Baseband
  - T4 is twisted pair, 4 pair, TX is 2 twised pair, FX is fiber
- Hub reads in a packet and retransmits it on all outgoing lines, except one on which it arrived.
- Intelligent switches watch ports and know destination MAC addresses on each port.

- Physical Medium
  - 3 media types, fiber 2-pair UTP, 4 pair UTP
- Physical Layer device
  - PHY, transceiver, onboard or box with MII cable
- MII
  - Optional device to allow 10 or 100 Mbps
  - Provides flexibility
- Data Terminal Equipment DTE
  - The Network device itself, the card.

100-Mbps TX Media System





100-Mbps TX Media System

- 100BASE-TX system operates over two pairs of wires, one pair for receive data signals and the other pair for transmit data signals.
- Most popular wiring is unshielded twisted-pair.
  - The two wires in each pair of the cable must be twisted together for the entire length of the segment, and kept twisted to within approximately 1/2 inch of any connector or wire termination point

100BASE-TX Components

- - Network Medium
- - 100BASE-TX Repeaters
- - 100BASE-TX Crossover Wiring
- - 100BASE-TX Link Integrity Test

Network Medium

TABLE 0.1 100BASE-TX eight-pin connector

Pin Number	Signal
1	Transmit+
2	Transmit-
3	Receive+
4	Unused
5	Unused
6	Receive-
7	Unused
8	Unused

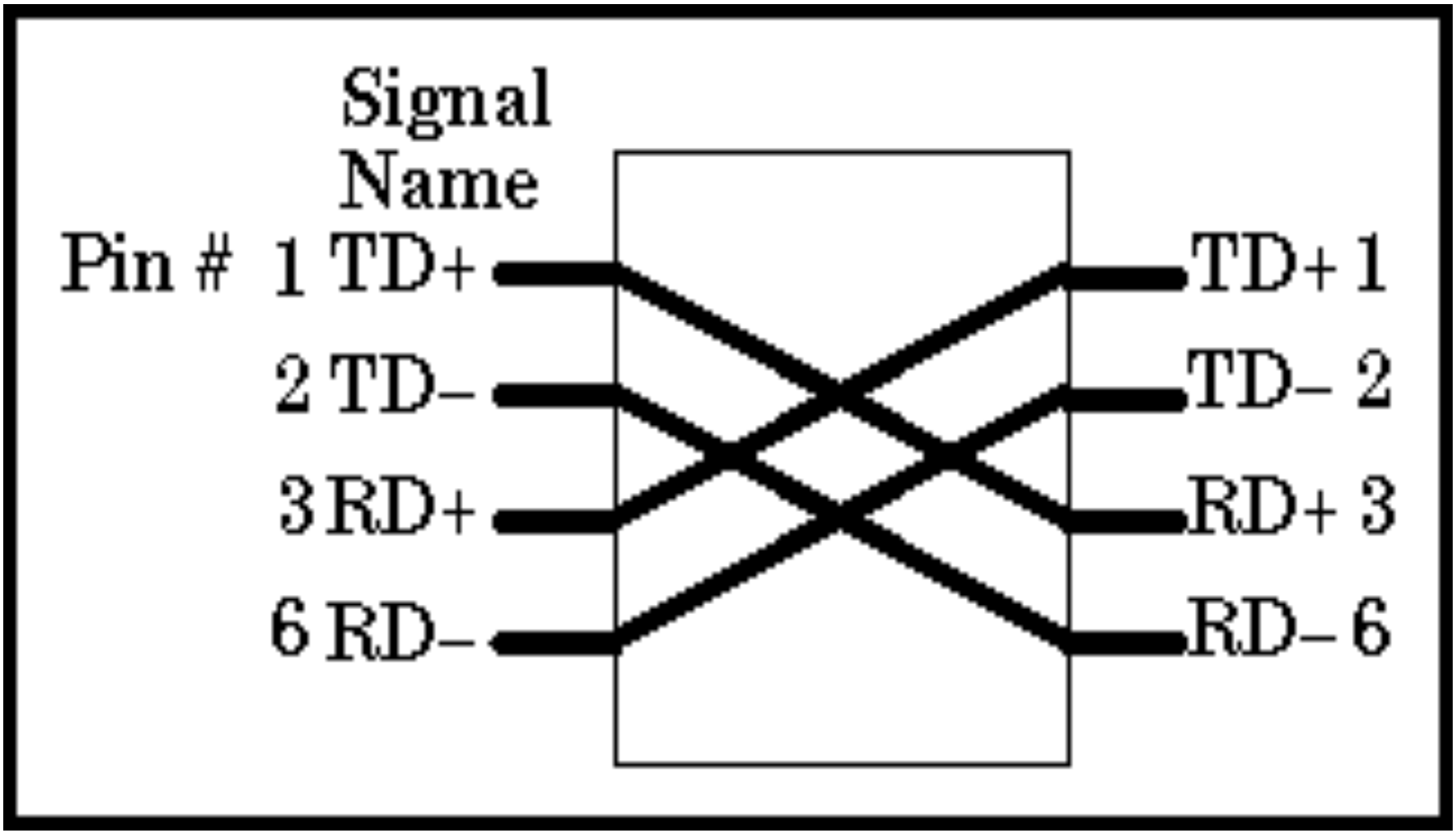
- Allows segments of up to 100 meters in length
- EIA/TIA standard recommends segment length 90 m between the wire termination equipment in the wiring closet, and the wall plate in the office
  - This provides 10 m of cable allowance to accommodate patch cables at each end of the link, signal losses in intermediate wire terminations on the link, etc.

100BASE-TX Repeaters

- Two types of repeater: Class I and Class II.
  - A Class I repeater allowed to have larger timing delays, and operates by translating line signals on an incoming port to digital form, and then retranslating them to line signals when sending them out on the other ports.
  - Possible to repeat signal between media segments that use different signaling techniques, such as 100BASE-TX/FX segments and 100BASE-T4 segments
  - Class II repeaters:- restricted to smaller timing delays, and immediately repeats the incoming signal to all other ports without a translation process ;connect only to segment types that use the same signaling technique

100BASE-TX Crossover Wiring

- Wiring multiple segments in a building.
  - Easier to wire cable connectors "straight through" do crossover wiring inside the repeater hub
- For single segment connecting 2 PCs, build special crossover cable
  - transmit pins on eight-pin plug at one end wired to receive data pins on eight-pin plug at other end of crossover cable.


Signal
Name
Pin # 1 TD+
2 TD–
3 RD+
6 RD–
TD+ 1
TD– 2
RD+ 3
RD– 6

100BASE-TX Configuration

- Connect the Ethernet interface in your computer to one end of the link segment, and the other end of the link segment is connected to the hub.
  - That way you can attach as many link segments with their associated computers as you have hub ports, and the computers all communicate via the hub.

100BASE-TX segment configuration guidelines

Maximum Segment Length		Maximum Number of MAUs	
100BASE-TX	100 m (328 ft.)[a]	Per Link Segment	2

a. 100BASE-TX segments are limited to a maximum of 100 m.

Gigabit Ethernet

- Easy migration to higher speed networks, as opposed to ATM or FDDI (no translation)
- Cost is always the issue
- Support for new applications and new data demands
- Flexibility in network design
- MIB (SNMP) management is the same as 802.3

Migration Issues

- Frame formats
  - Same variable length (64 to 1514 byte) frames
  - Allows seamless integration
  - No frame translation necessary
  - Where to install the upgrade (desktop to switch to backbone) ?

Physical Layer

- 1000 Base-X based on **F**iber **C**hannel **P**hysical **L**ayer (FCPL)
  - Proven technology
  - 1000 Base-SX :- 850 nm laser multimode
  - 1000 Base-LX :- 1300 nm laser single and multimode laser
  - 1000 Base-CX copper Shielded Twisted Pair
  - table 1
- 1000 Base-T:- long haul 4 pair category 5 UTP cable (802.3ab task force)

MAC Layer - *Carrier Extension*

- Carrier Extension
  - 10 times faster than Fast Ethernet, 10m would be max slot size…. Problem
  - Slot size of 1512 bytes employed, with pads.
  - Carrier Extension allows longer distances
  - Transparent to LLC

Carrier Extension Diagram

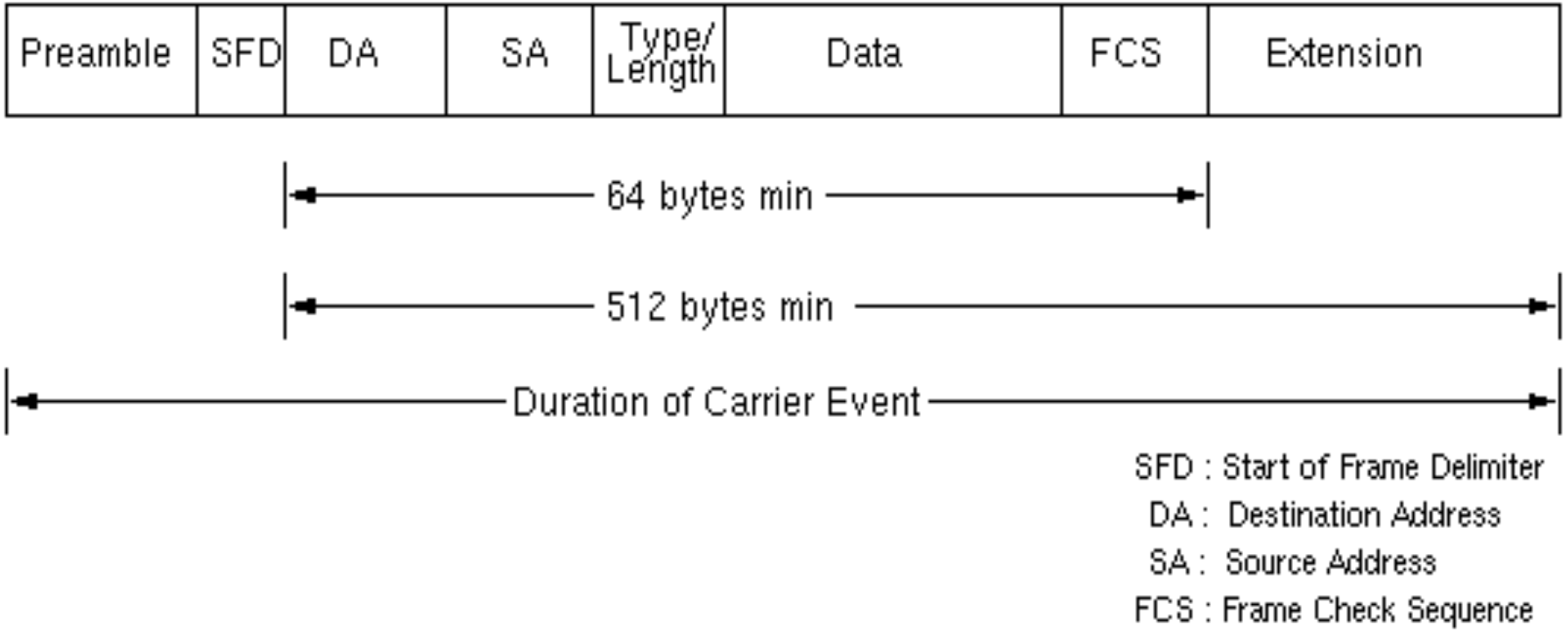


Fig 1. Ethernet Frame Format with Carrier Extension

MAC Layer - *Packet Bursting*

- Carrier extension wastes bandwidth, with 448 pad bytes in small packets.
- For small packets, throughput only marginally better than fast Ethernet, 802.3X…. Problem !
- Solution:- extend the Carrier Extension
  - Pad 1st packet to slot time (512 bytes), subsequent packets back to back with minimum inter-packet-gap until burst timer (1500 bytes) expires.

Packet Bursting Diagram

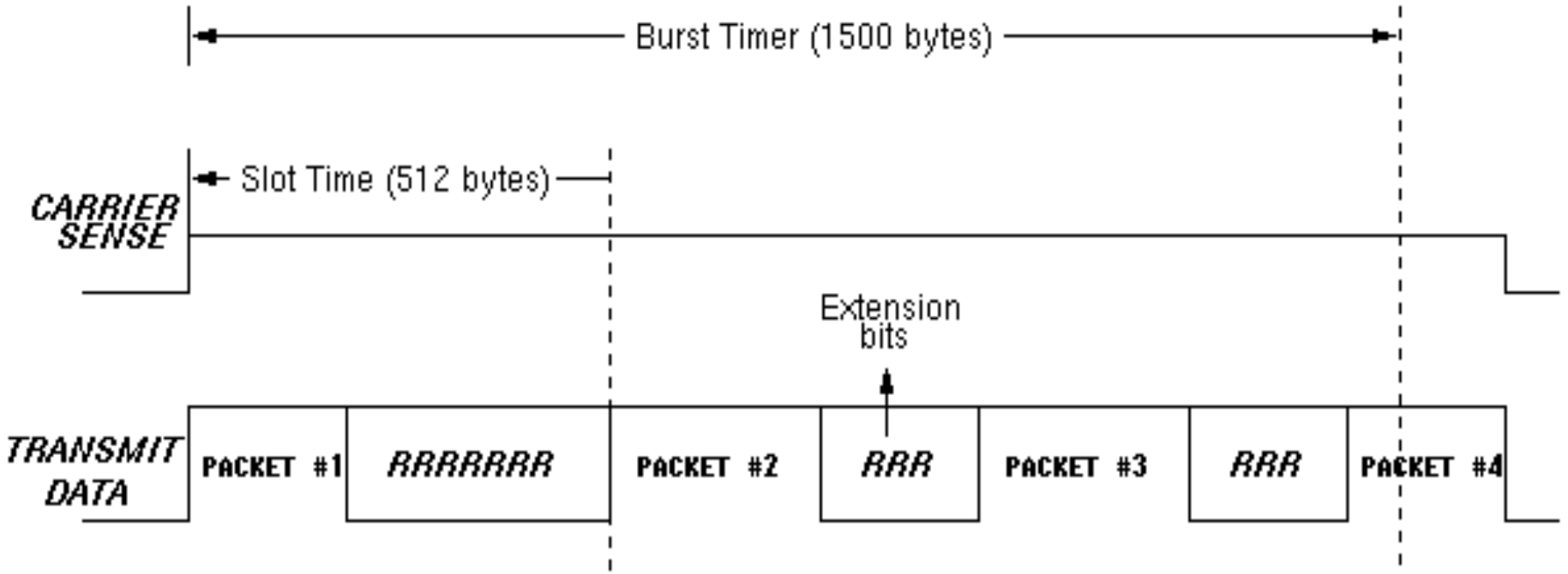


Fig. 2. Packet Bursting

Topologies

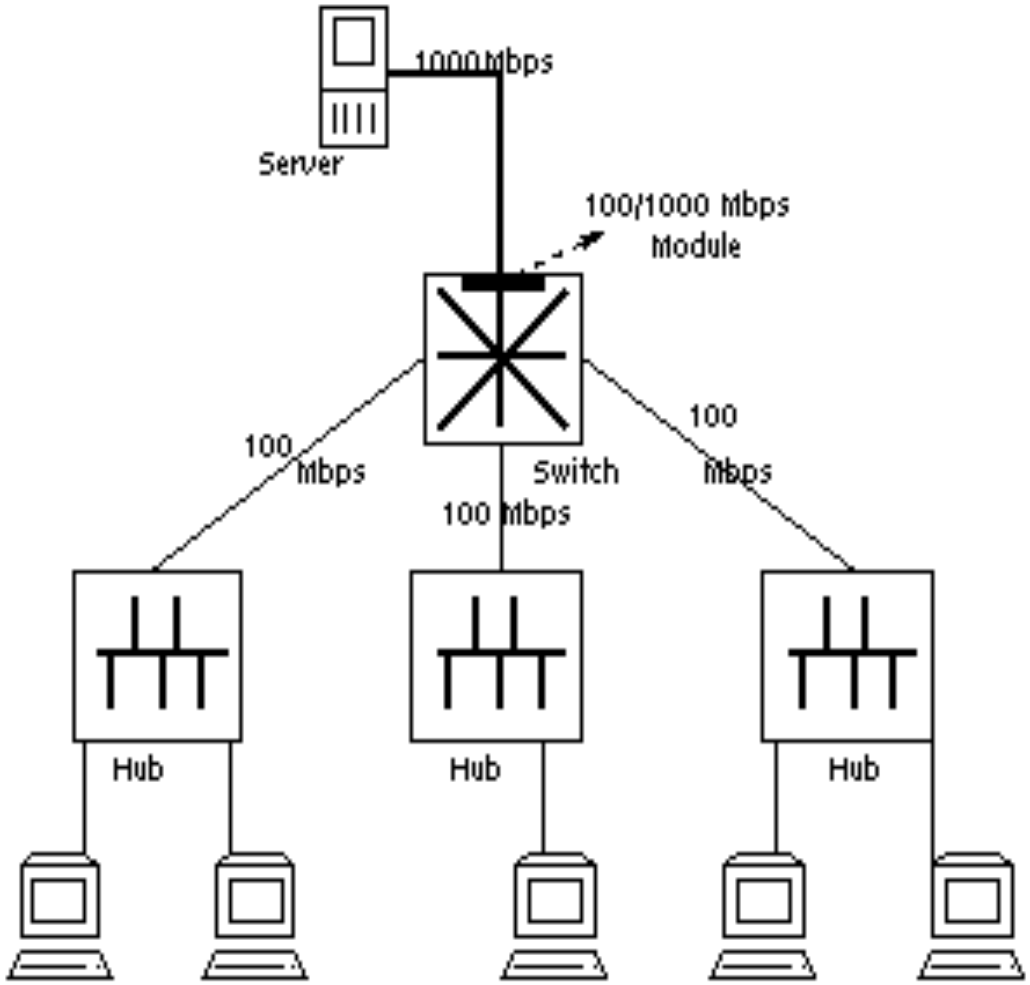
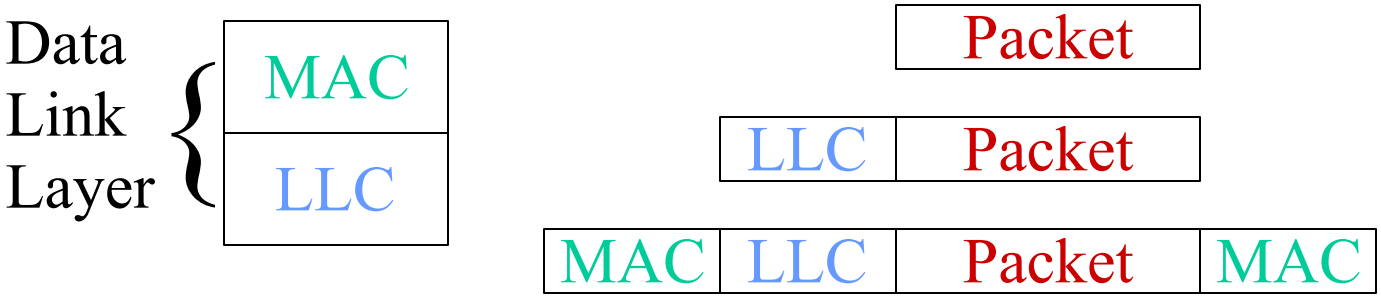


Fig. 5. Server-Switch Connection

Encapsulation and Protocol Hierarchies

- Higher layer entities build packets and provide these as a bit\byte stream to lower layer entities.
- Wrapping like Russian Dolls.

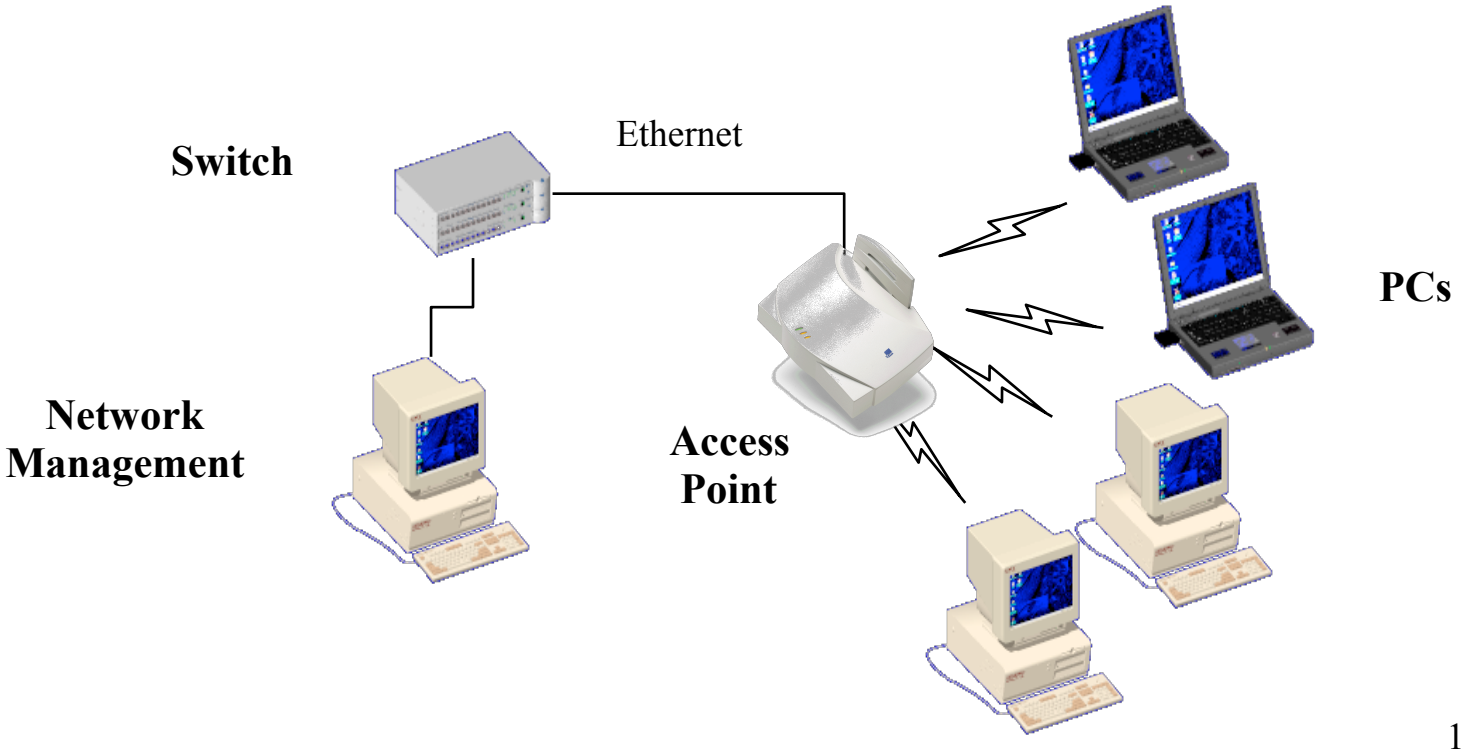


Wireless Technologies

PAN "Personal Area Network"	LAN "Local Area Network"	MAN "Metropolitan Area Network"	WAN "Wide Area Network"
Bluetooth	802.11b 802.11a HiperLAN2	802.11 MMDS LMDS	GSM GPRS CDMA 2.5-3 G
Low Data Rates Short Distances Notebook/PC to Devices/ Printer/Keyboard/Phone	Higher Data Rates Medium Distances Computer-Computer and to Internet	Higher Data Rates Med-longer Distances Fixed, last mile access	Lower Data Rates Longer Distances PDA Devices and Handhelds to Internet
< 1 Mbps	2 to 54+ Mbps	22+ Mbps	10 to 384 Kbps

802.11b

WiFi

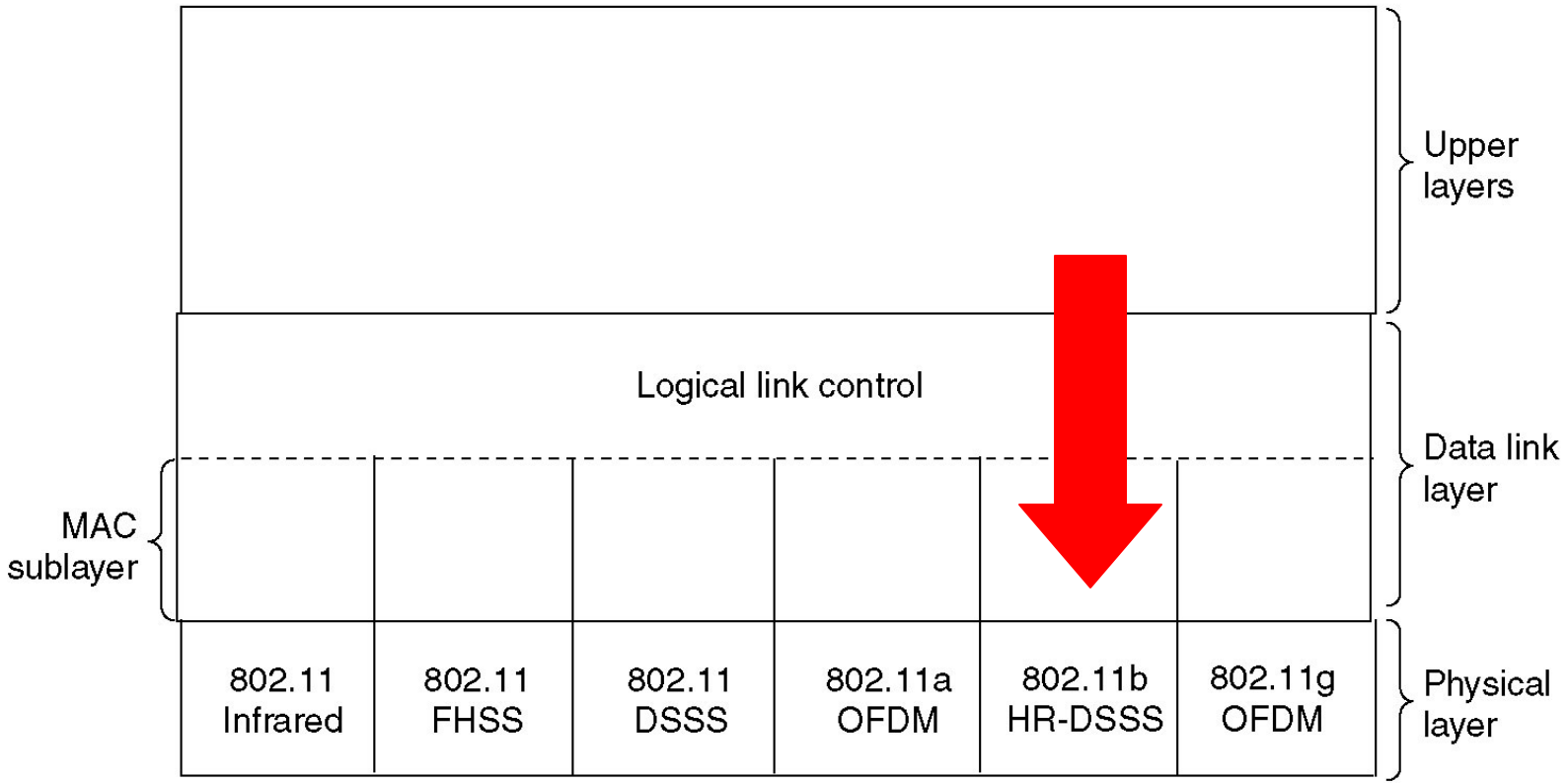


Wireless LANs

- The 802.11 Protocol Stack
- The 802.11 Physical Layer
- The 802.11 MAC Sublayer Protocol
- The 802.11 Frame Structure

The 802.11 Protocol Stack

Part of the 802.11 protocol stack.

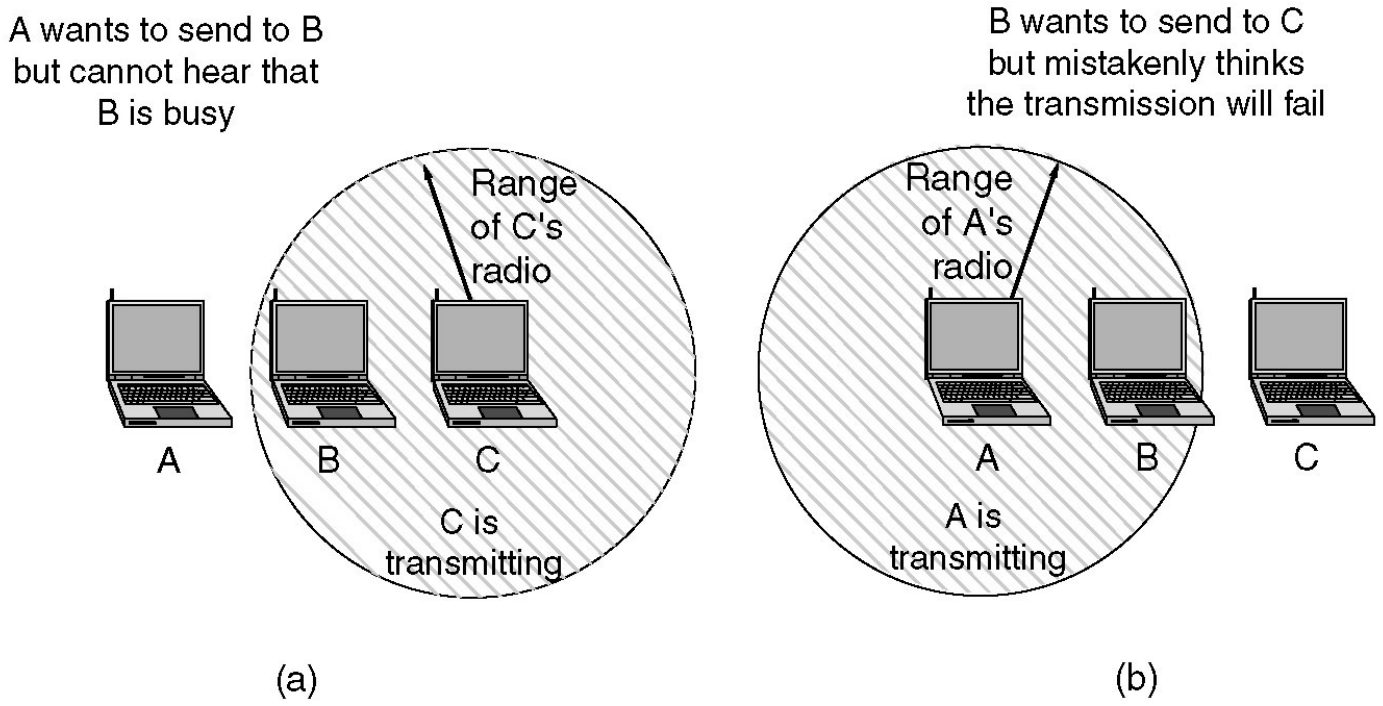


802.11 HR-DSS

- High Rate - Direct Sequence Spread Spectrum (HR-DSSS)
- Speeds
  - 1, 2, 5.5, 11 Mbps
- Dynamic speed adaptation
- Same bandwidth as cordless phones, Bluetooth and microwave ovens
- ISM Band

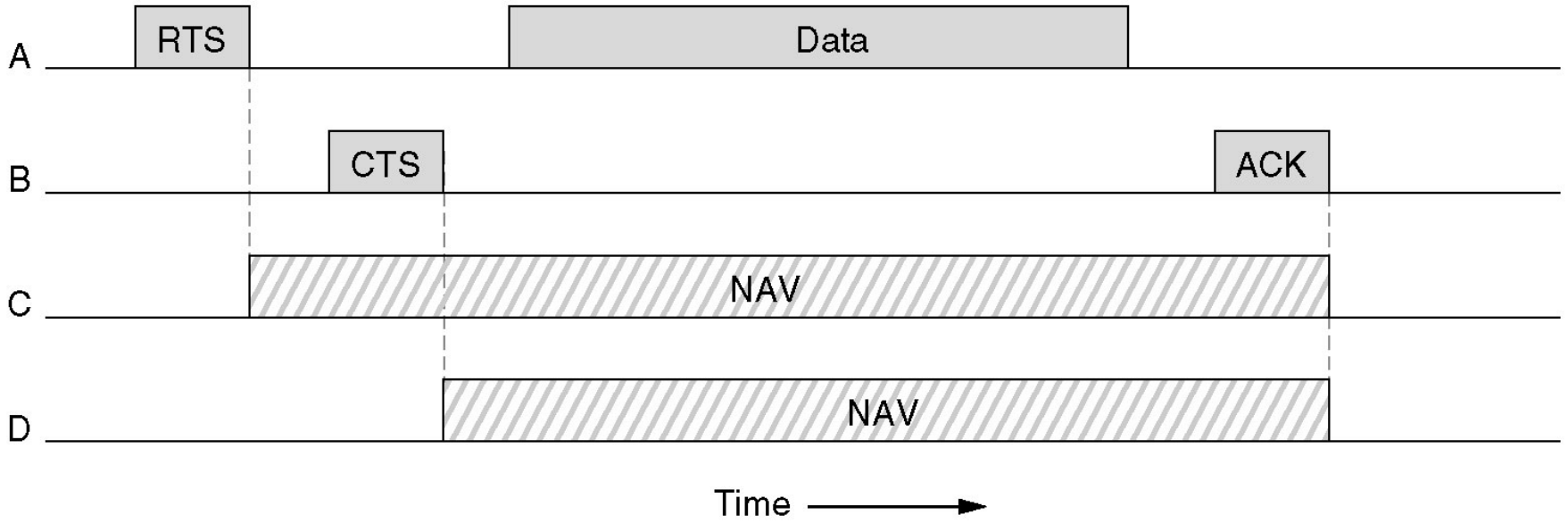
802.11 MAC

(a) The hidden station problem.
(b) The exposed station problem.

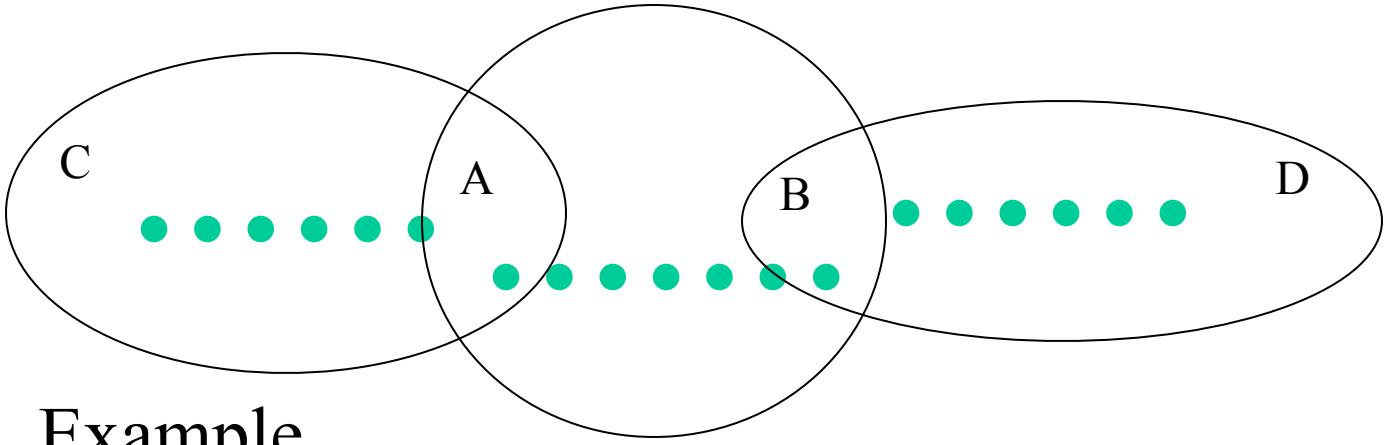


The 802.11 MAC Protocol

- CSMA/CA - Collision Avoidance

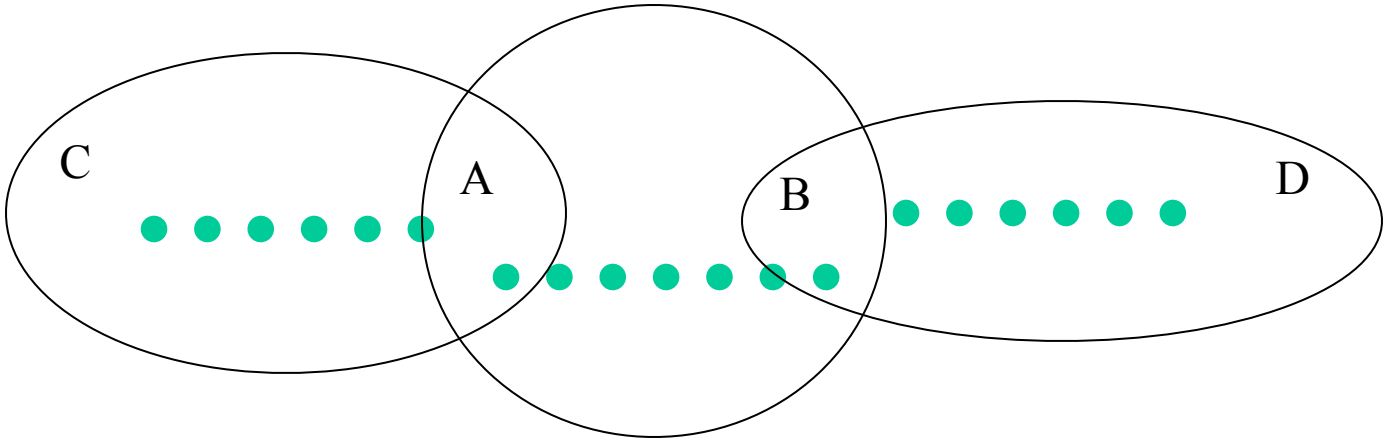


CSMA/CA



- Example…
  - A wants to send to B, sends RTS
  - B says Ok with a CTS frame
  - A sends its frame & starts ACK timer.
  - B gets frame Ok and sends ACK frame.
  - If A's ACK timer expires, start again

- Considering other stations…
  - C within range of A… may receive RTS, if so Hush. This is Network Allocation Vector **NAV**
  - D doesn't hear RTS but hears CTS… assert NAV
  - All fine & dandy!

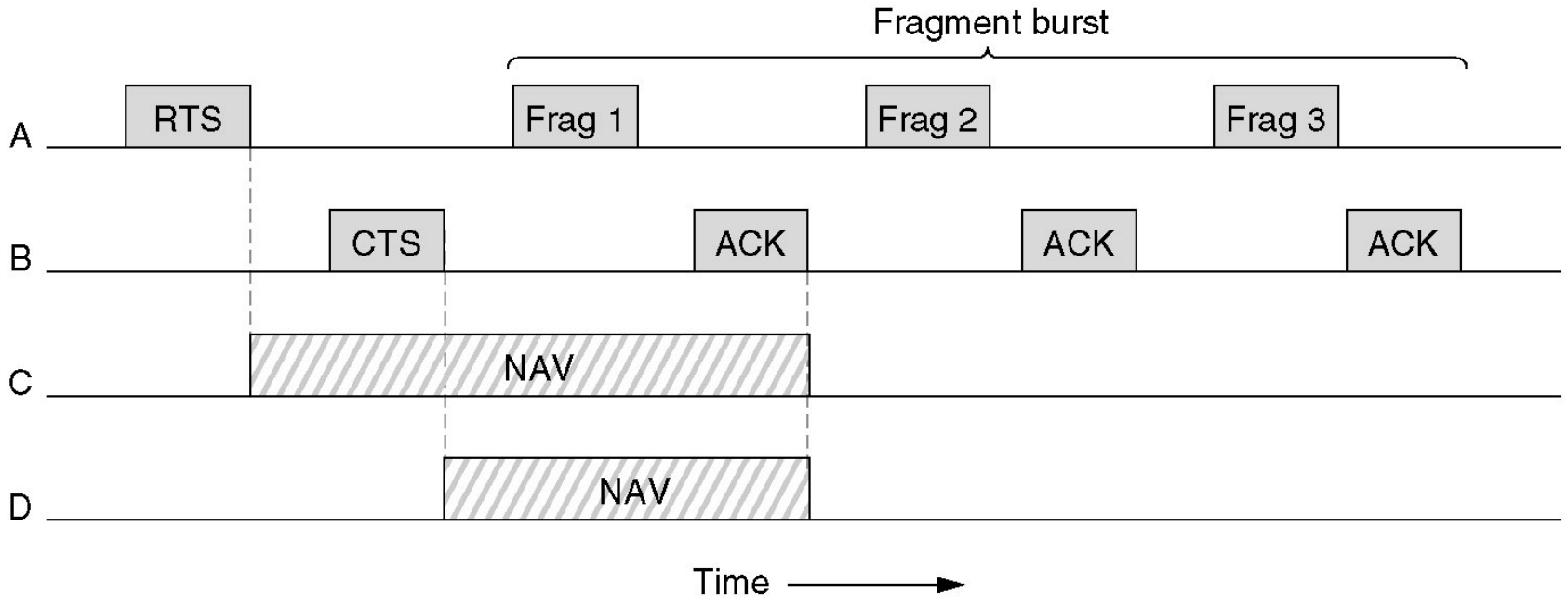


But ISM is Noisy!

- Probability of 1 bit error is *p*
- Probability of n bit frame arriving uncorrupted is $(1-p)^n$
- So, for $P = 10^{-4}$, 12144 bit frame has <30% probability of arriving correct.
- If 10^{-5}, roughly 1:9 will be damaged.
- If 10^{-6}, roughly >1:100 will be damaged.
- Bigger frames more susceptible to damage!

802.11 MAC & Noisy Channel

A fragment burst.



802.11 MAC & Noisy Channel

- Fragment frames, use checksums & number
- Acknowledge using Stop & Wait
- Once channel is acquired (RTS & CTS), send fragment burst, ACK each fragment.
- This is what is called Distributed Coordination Function (DCF) Mode

802.11 Point Coordination Function - PCF

- Base station polls... central control.
- Beacon frame transmitted periodically.
- There cannot be any collisions.
- Beacon frame contains system parameters.
- PCF and DCF may coexist, check Tanenbaum.

The 802.11 Frame Structure

The 802.11 data frame.

802.11 Frame Structure

- Data, Control & Management frames
- Control has 11 fields
  - protocol version [PCF | DCF]
  - Type - [Data | Control | Management]
  - Subtype [RTS | CTS]
  - To DS and From DS indicate to\from intercell distribution system (e.g. Ethernet)
  - MF More Fragments
  - Retry - this is a retransmission
  - Pwr - power management [go asleep | wake up]
  - W - encrypted with WEP
  - O - process this frame sequence in order

- Duration field says how long frame & acknowledgement will occupy channel.
- 4 addresses - Source & Dest, also Source & Dest. Base stations for intercell traffic.
- Sequence is for fragment numbering, 12 bits for frame, 4 for fragment
- Data contains payload, up to 2312 bytes
- Checksum is CRC
- Mgmt frames operate within single cell
- Control frames are RTS, CTS and ACK

802.11 Distribution Services

- Association
- Disassociation
- Reassociation (roaming)
- Distribution (wired or wireless)
- Integration (protocol translation)

802.11g High Speed Wireless LAN

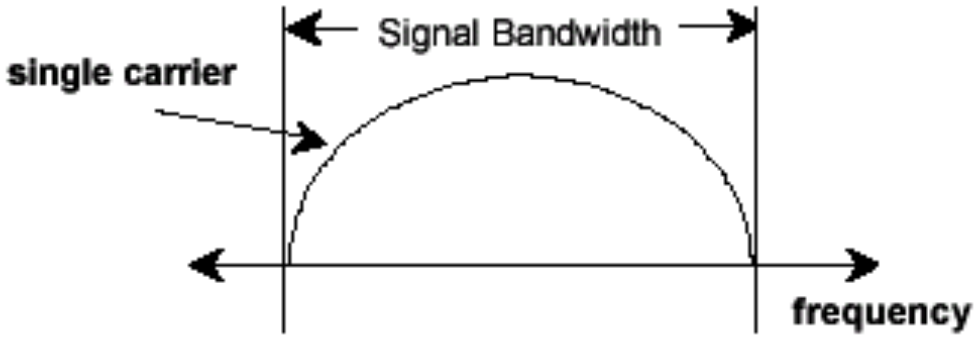
- 2.4GHz is still the frequency band with 54Mbps
- Compulsory…
  - Orthogonal Frequency Division Multiplexing (OFDM) used for rates > 20Mbps.
  - Complementary Code Keying (CCK) required for backward compatability.
- Optional
  - CCK\OFDM Hybrid Header\Payload
  - PBCC Hybrid Header\Payload (Texas Instruments)

802.11g Packet
Preambles & Payloads

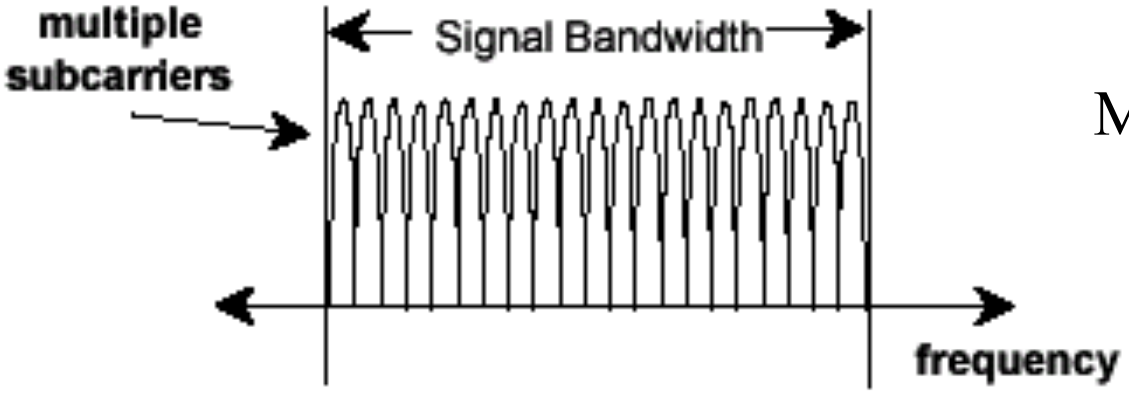
Preamble/Header	Payload

- Preamble warns of forthcoming packet
- Header contains length of packet.
- Payloads vary from 64Byte to 1500Byte.
- Generally CCK used to transmit header and payload, usually!

CCK & OFDM



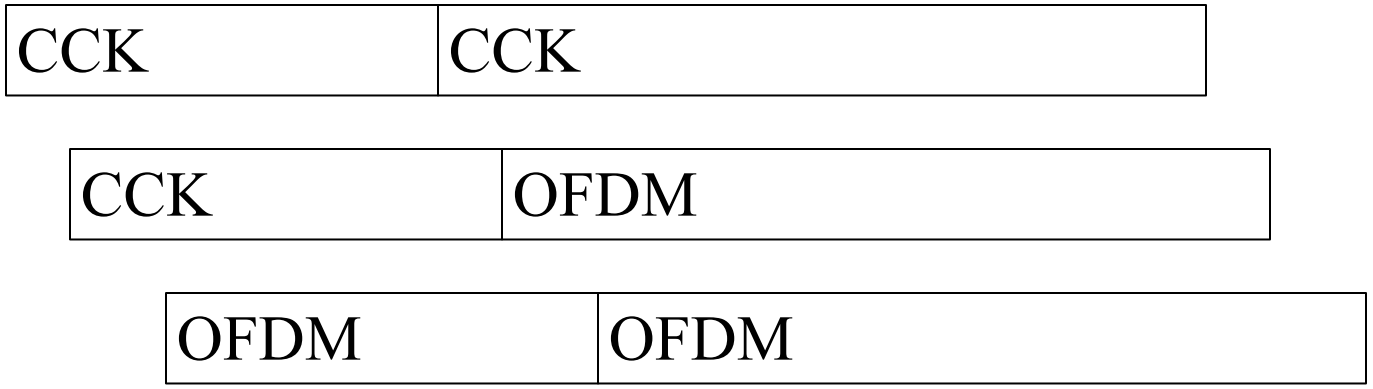
Max. 11Mbps



Max. 54Mbps

WiFi Interoperability

- CSMA\CA will be used again.
- RTS\CTS will be used
- Headers may be transmitter using CCK and payloads may use OFDM

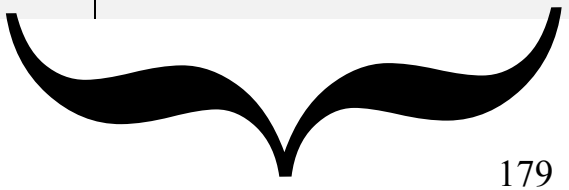


802.11g Security

- Wired Equivalent Privacy (WEP)
  - Garbage
- Service Set Identifier (SSID)
  - Disable broadcasts
- WiFi Protected Access (WPA)
  - Stronger than WEP
- MAC Address Authentication
- 802.1x Network Authentication
  - EAP

Comparing Wireless Technologies

	Infrared	Bluetooth	802.11b
Frequency	10^{13} - 10^{14} Hz	2.4 GHz	2.4 GHz
Transmission Method	Line-of-sight	Frequency Hopping	Direct Sequence Spread-Spectrum
Speed	4 Mbps	1 Mbps	11 Mbps
Range	1 meters	30 meters	100 meters
Network	PAN	PAN/LAN/WAN	LAN
Signal	Data or Voice	Data & Voice	Data
Security	None	Authentication, Encryption	Authentication, Encryption



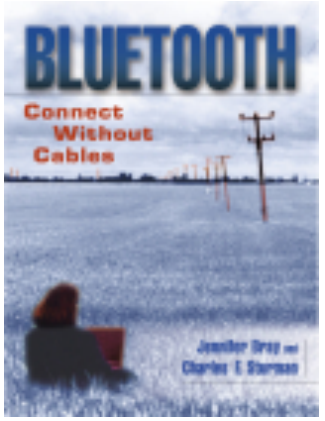
Security? - RUBBISH !

Links

- http://www.xilinx.com/esp/Bluetooth/tutorials/bt_overview.htm
- http://www.palowireless.com/bluearticles/intro.asp
- Bluetooth Application Developer's Guide, Edited by Jennifer Bray (Syngress, published 15/12/01)
- Bluetooth, Connect Without Cables, Jennifer Bray and Charles Sturman (Prentice Hall) [more hardware focused]
- Wireless ad hoc networking - the art of networking without a network. Magnus Frodigh, Per Johansson and Peter Larsson, Ericsson Review No. 4, 2000
- Infrastructure for Distributed Applications in Ad Hoc Networks of Small Mobile Wireless Devices (the Anhinga project). Alan Kaminski, Rochester Institute of Technology, May 2001.

Books

- Bluetooth: Connect without cables. Bray & Sturman ISBN 0-13-089840-6



The Internet

Issues to be Addressed

Internetworking

- TCP/IP is the de-facto internet standard.
- Major issues to be addressed in Internetworking are…
  - Service type.
  - Addressing
  - Routing
  - QOS
  - Max. packet size
  - Flow & congestion control
  - Error reporting

Service Type

- Connection oriented TCP
  - Provides reliable error free transport.
  - Utilises sliding window protocol.
- Connectionless UDP
  - Provides best effort datagram delivery.
  - Unreliable, packets may be discarded, not acknowledged.

Addressing

- How do we address processes running on hosts ?
- How do we ensure unique addresses ?
- How do we map LAN addresses to TCP/IP addresses ?
- How do we interpret addresses ?
- How do we know where to send packets, i.e. route packets ?

Routing

- Issues include …
  - How does host determine address of router attached to its network.
  - How does router determine the NPA addresses of hosts attached to its network.
  - How does host select a particular router when sending a packet.
  - How does router determine addresses of other routers attached to the same network
  - How does router select another router to which to send packets given destination host address.

Quality of Service

- Issues include…
  - Transit delay expected when delivering packets to destination.
  - Security and privacy required.
  - Cost of delivery.
  - Probability of error.
  - Priority of transfer.

Maximum Packet Size

- Prevailing conditions may determine size.
  - High bit error-rates: smaller packets better.
  - Large transit delay: large queuing delays at each intermediate router, reduces efficiency.
  - Buffer requirements at routers may dictate that it is easier to store smaller than larger packets..
  - Processing overheads used in processing large numbers of small packets are larger than processing smaller numbers of larger packets.

Introduction to TCP/IP

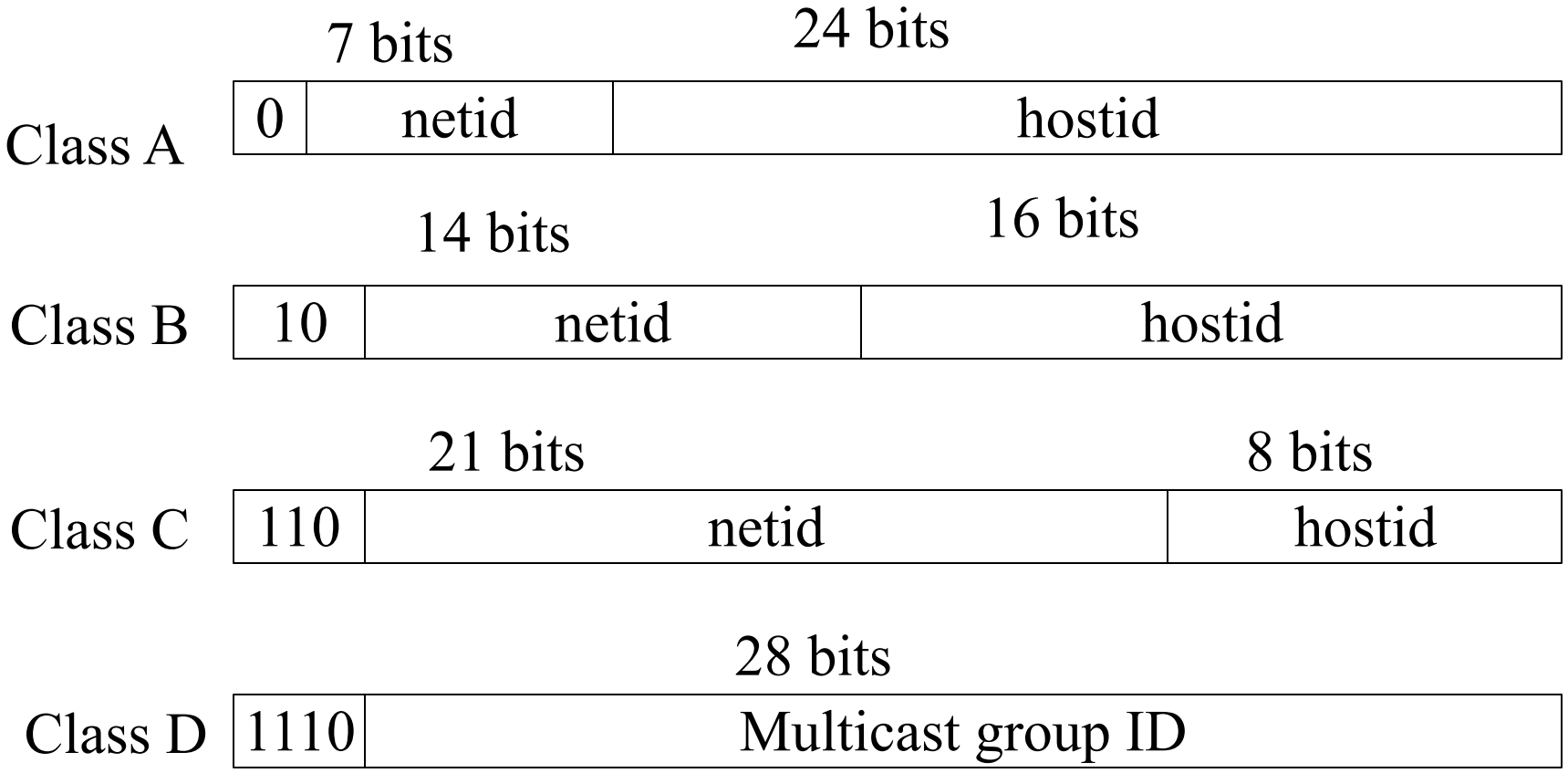
TCP/IP

- Four layer Architecture
- Developed in 1960's
- *Open* System
- Not just one protocol, whole family.
- Many programming interfaces available.
- Standardised protocol set.

IP Addressing Scheme

- Need capability of mapping addresses of one type onto another.
- LAN address, Network Point of Attachment NPA, must be mapped onto an IP address.
- NPA formats differ from one LAN standard to another.
- IP addresses are homogenous within single IP version.

IP Address Format



IP Address Format (cont.)

- Different size networks may use different address classes, defined by the first few bits in the address. 0 for Class A, 10 for Class B, 110 for Class C, etc. etc.
- Networks with large numbers of hosts may use Class A, while Class C may have many subnets with a small number of attached hosts.

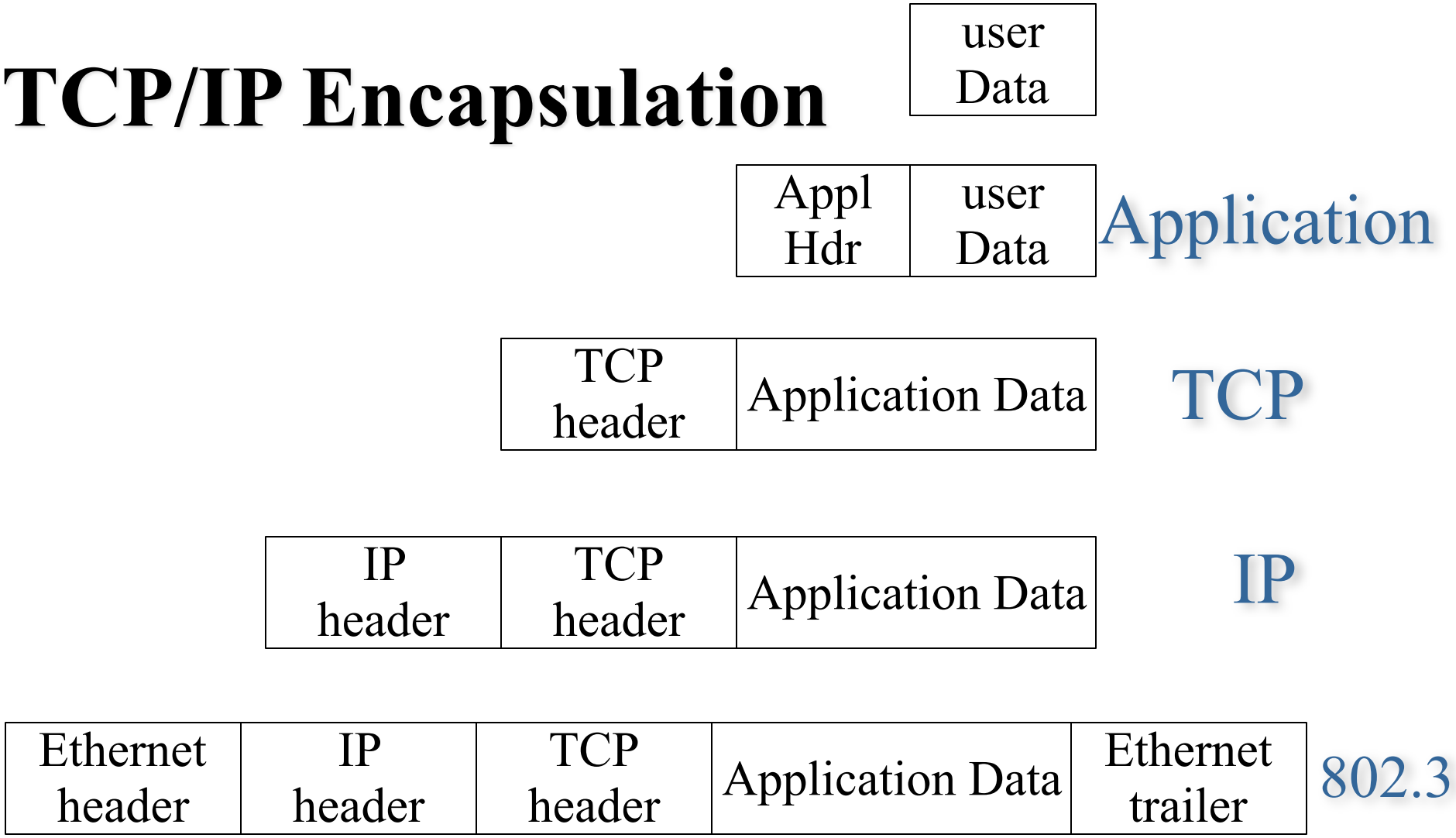
IP Address Notation

- A decimal dot notation is used to break down the IP address.
- Example
  - 10001000 11001110 00001011 00000110
  - gives the address 136.206.11.6 aka boole !
  - Note that this is a Class B address (first zero in second position) and the subnet is defined with 14 bits, the host address with 16 bits.

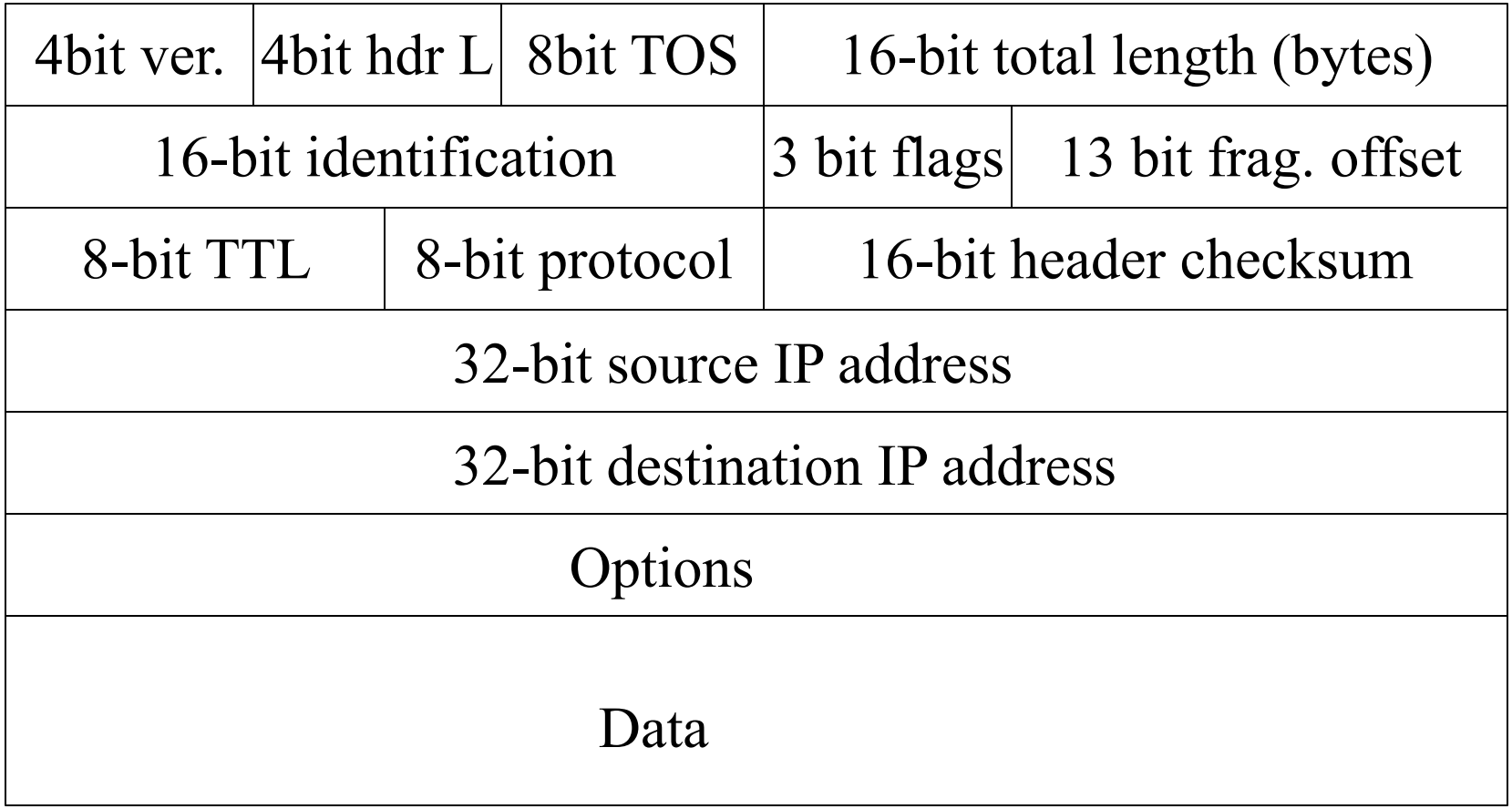
IP Allocations

- A central authority has responsibility for allocation of IP addresses. They are the network Information centre, or NIC.

TCP/IP Encapsulation



IP Packet Header



IP Header Description

- *Version*: Currently V 4.
- *Header Length*:Specifies length of header as some fields are optional.
- *Type of Service*:This is the same as the QOS mentioned previously.
- *Total length*:Specifies the length of the datagram.

- *Identification*: Used to identify a set of datagrams which were formed from a single user message, but which got fragmented while traversing possibly several networks.
- *D bit*:Indicates that routers should not fragment a datagram i.e. Don’t fragment bit.
- *M bit*:Indicates that there are more fragments to follow in later datagrams.
- *Fragment offset*:Where this fragments fits into the original fragmented datagram.

- *Time to live*: Datagram loses a life (or some time to live) on each hop across the internet. Datagram destroyed when time\lives run out. Prevents Datagrams from wandering endlessly.
- *Header Checksum*: Checks header only.
- IP addresses (Source, Destination): As described previously.

IP Routing

- Central function of IP is routing along with fragmentation and re-assembly of data across an internet.
- Routing information organised in a hierarchy. With hosts and gateways involved.
- ARP address resolution protocol maps IP to Ethernet addresses, an Interior Gateway Protocol (IGP)

- Exterior Gateway Protocol (EGP) knows about other routers on the internet and can route from network to network.
- Distance Vector and Link State routing are most popular, Link State is superior.
- Subnet addressing may be performed on a group of related networks (owned by one organisation).
- More on Routing later…

Special IP Addresses

- Some addresses are reserved for special use.
- IP address composed of all 0 means this host.
- Network part all 0, Host part not, host on this network.
- All 1s broadcast on LAN
- Host part 127.0.0.x is Loopback, useful for debugging.
- 192.168.0.0 and 10.0.0.0 are reserved by IANA and are private addresses
- 172.16.0.0 up to 173.31.255.255 are reserved /12 or 16 class B addresses also reserved.

Creating Subnets

- Address space
  - [network#, host#]
  - [network#, subnet#, host#]
- Subnet *mask* used to find the host part of IP address and distinguish it from the NW part.

Class	Format	Default subnet mask
A	nw.node.node.node	255.0.0.0
B	nw.nw.node.node	255.255.0.0
C	nw.nw.nw.node	255.255.255.0

Subnetting – Why?

- Reduces Network traffic
  - Routers create smaller broadcast domains, more smaller domains limits the span of a broadcast.
- Optimizes NW performance
  - Less traffic, things run faster.
- Simplifies management
  - Easier to do fault analysis on a smaller self-contained NW than with a single huge NW
- Facilitates spanning of large geographical distances
  - Single large NW over large distance incurs big overhead of resources. Smaller NWs which keep much traffic local will incur less overhead over the long haul.

CIDR

- Classless Inter Domain Routing -
- Give the IP address space some breathing room!
- Basic idea: allocate the remaining IP addresses in variable-size blocks without regard to classes
  - original name: Supernetting, the opposite of Subnetting (sortof)
- A site needing 2000 addresses receives a block of 2408 addresses i.e., 8 contiguous class C networks. If need 8000 hosts, then allocate a block of 8192 addresses, i.e., 32 contiguous class C networks.

Variable Length Subnet Masks

- Only works with routing protocols which support CIDR
- Different masks on each router interface. Small number of bits for routers so they have few hosts, few routers. Keep big numbers for LANs
- Match required number of hosts to appropriate mask on each interface.
- Requires careful design so that blocks do not overlap
- Routes may be summarised, providing a hierarchy.

Transmission Control Protocol

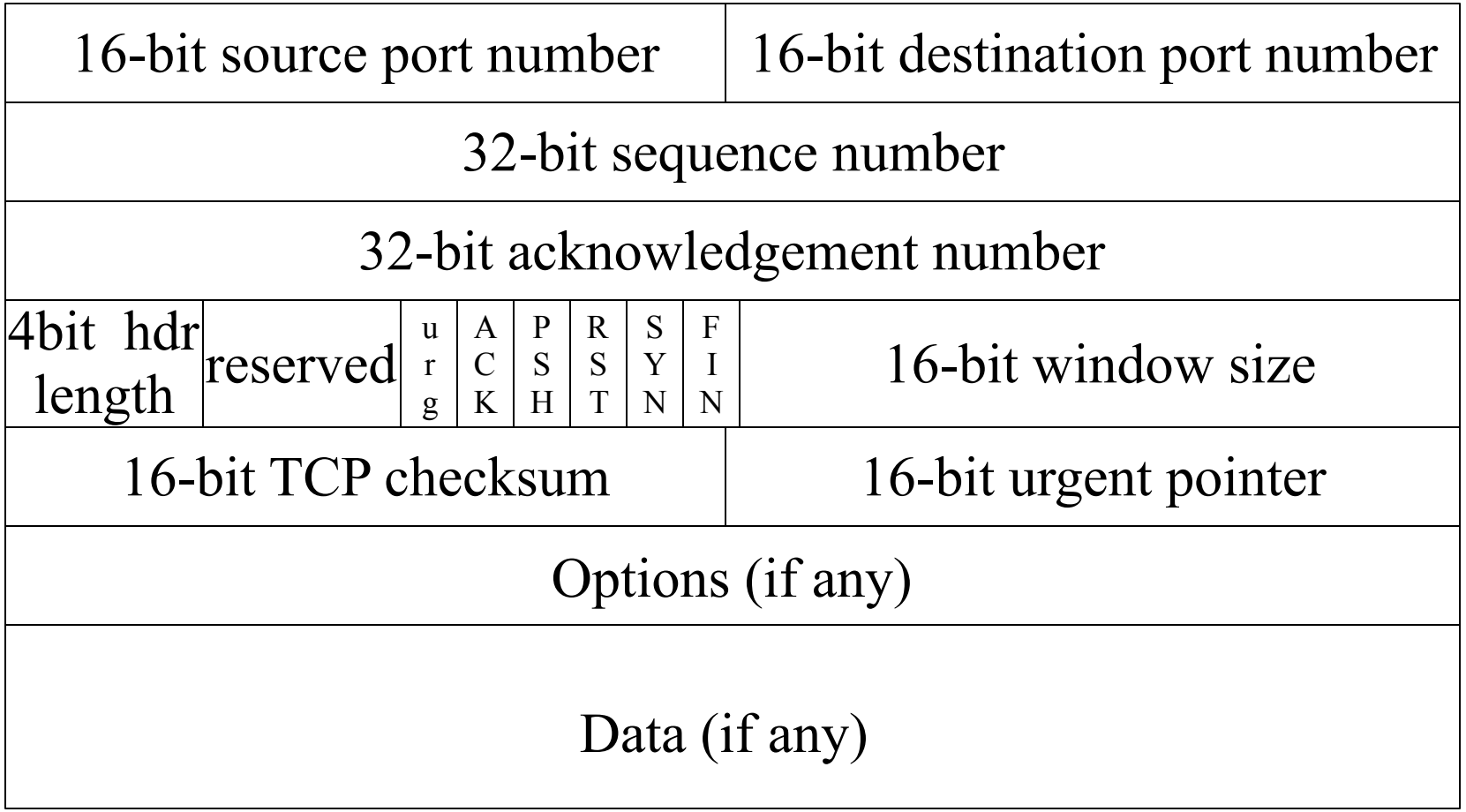
OSI Transport Layer

TCP Services

- Provides connection-oriented, reliable, byte stream service.
- Segments passed to IP for routing, timer attached for each segment.
- Sliding window protocol utilised with go-back-n or selective-repeat for retransmission.
- All TCP segments acknowledged.

- TCP segments may arrive out of order, sliding window will sort order.
- TCP segments may be duplicated, duplicated are discarded.
- TCP provides flow control, no process\host will be swamped, helps avoid congestion.
- TCP utilised by many internet applications such as Telnet, Rlogin, FTP, E-mail, WWW Browsers.

TCP Segment Header



TCP Header Description

- *Source Port* and *Destination Port* identify transport end-points of connection.
- *Sequence Number* and *Acknowledgement Number* perform usual functions, Ack numbers next byte expected.
- *TCP Header Length* indicates number of 32 bit words in header. Length varies because of options.
- Not used. No bug fixes required !

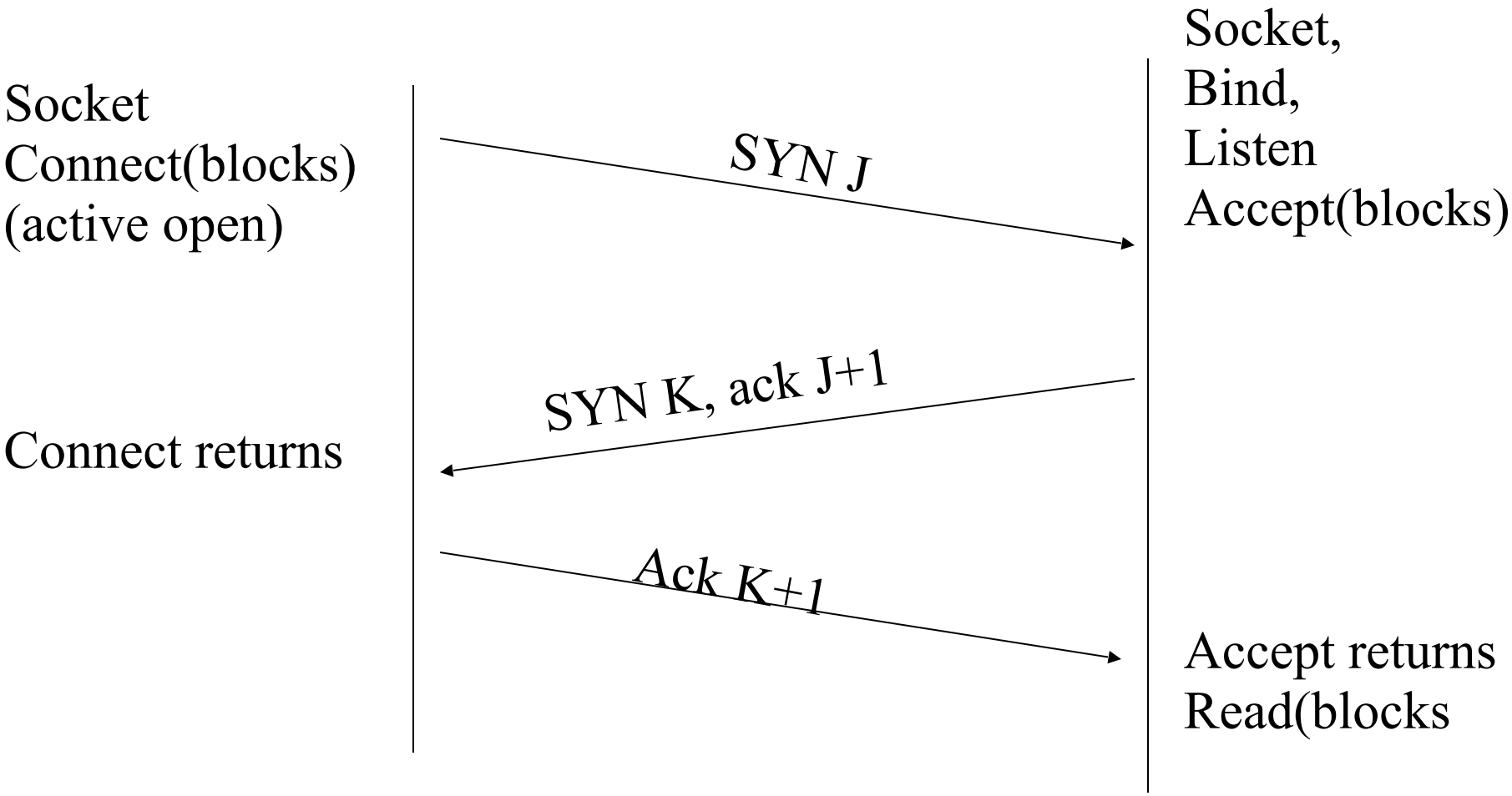
- Six one bit flags…
  - URGent pointer in use, used for indicating interrupts and offset from seq no. to urgent data.
  - ACK bit used to indicate piggybacked acknowledgement.
  - PSH requests that receiver does not buffer but to deliver.
  - RST is reset connection, means problems !
  - SYN used in conjunction with ACK to request connection.
  - FIN release connection

- *Window size* used for variable-sized sliding window. Size of zero indicates a choke packet.
- Checksum checks header.
- Options field for things like specification of maximum TCP payload. Negotiated at startup lowest bid wins.
- A *selective repeat* instead of *go-back-n* sliding window protocol may be specified as an option.

TCP Addressing

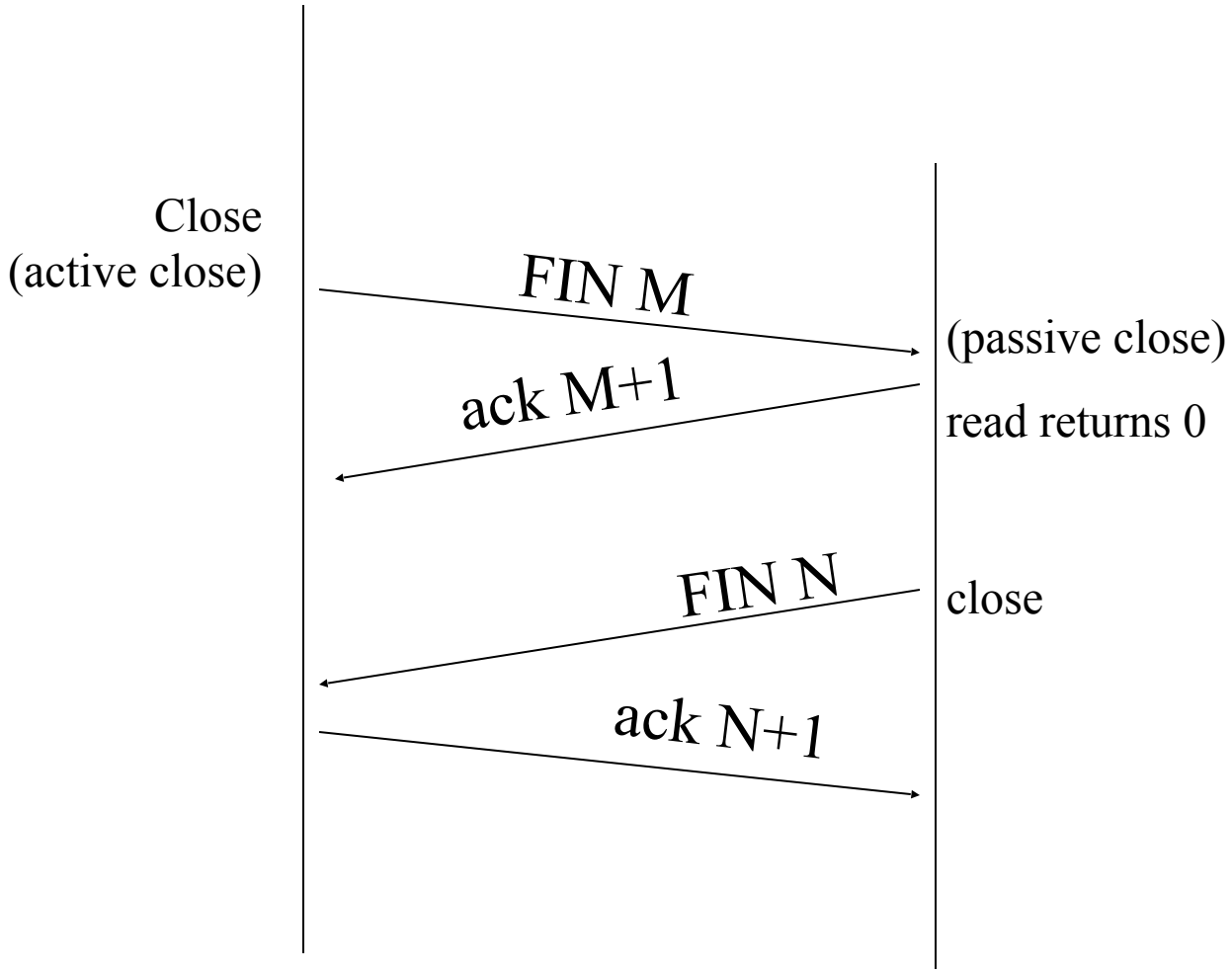
- TCP uses notion of Port Number to access transport endpoint on a single host. Many Ports may be in use simultaneously.
- Combination of IP address and port number uniquely identifies a port for process running on a particular machine.
- Process may even have several ports open.

Three Way Handshake



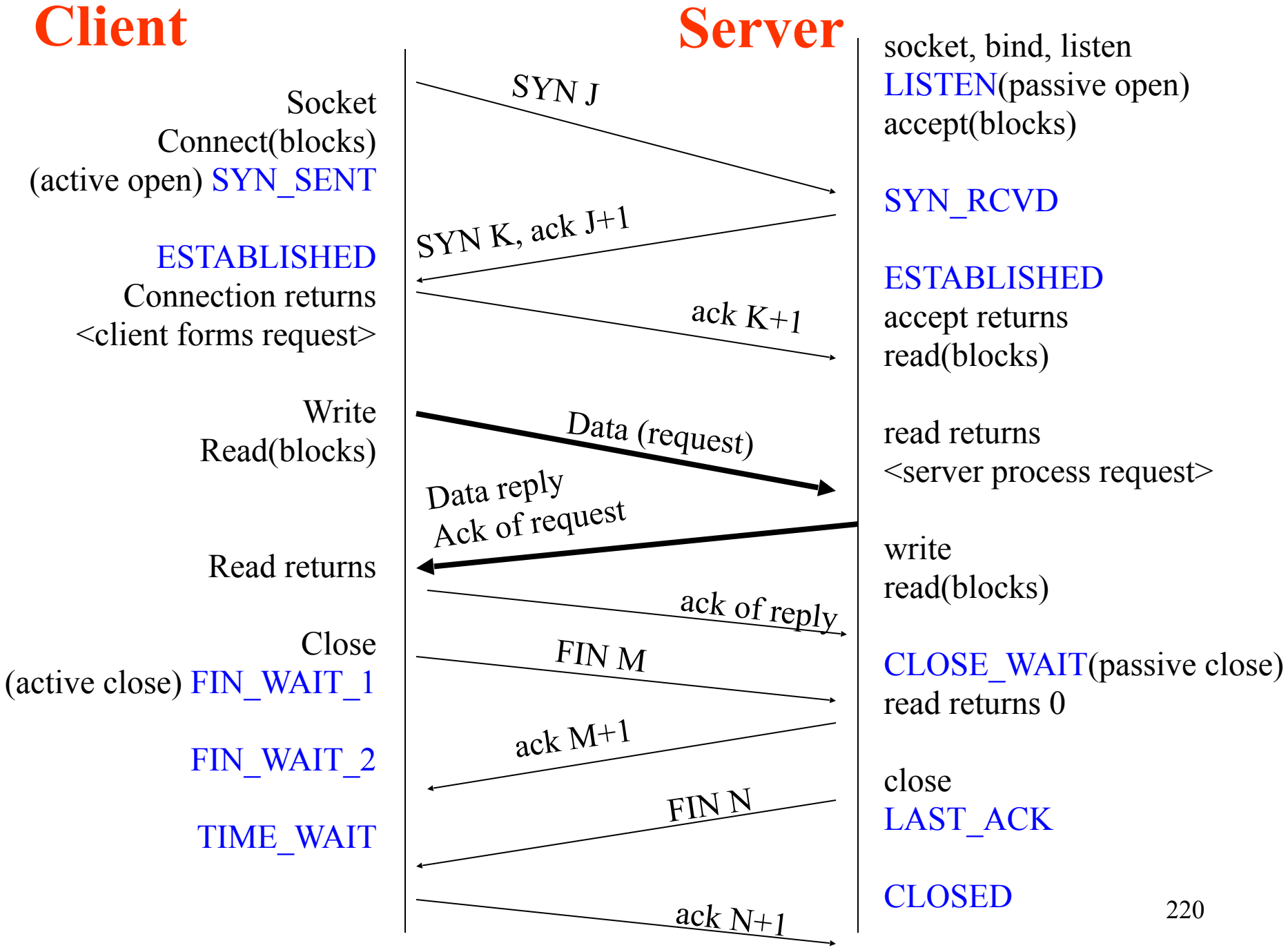
TCP Connection Termination

- If application calls close first, this is an *active close*.
- Sends FIN segment, meaning finished sending data.
- Server performs *passive close*.
- Clients FIN is ack'ed and sent to application as EOF, after any queued data to receive.
- When application receives its EOF, it will close its socket. TCP sends FIN.
- The server on receiving final FIN acks that FIN.


Close
(active close)
FIN M
(passive close)
ack M+1
read returns 0
FIN N
close
ack N+1

TCP Connection & The Packets

- A complete TCP connection involves many packet exchanges.
  - Connection establishment
  - Data transfer
  - Connection termination
  - TCP states are also shown as client and server enter them.


Client
Server
Socket
Connect(blocks)
(active open) SYN_SENT
ESTABLISHED
Connection returns
<client forms request>
Write
Read(blocks)
Read returns
Close
(active close) FIN_WAIT_1
FIN_WAIT_2
TIME_WAIT
SYN J
SYN K, ack J+1
ack K+1
Data (request)
Data reply
Ack of request
ack of reply
FIN M
ack M+1
FIN N
ack N+1
socket, bind, listen
LISTEN(passive open)
accept(blocks)
SYN_RCVD
ESTABLISHED
accept returns
read(blocks)
read returns
<server process request>
write
read(blocks)
CLOSE_WAIT(passive close)
read returns 0
close
LAST_ACK
CLOSED

The interactions

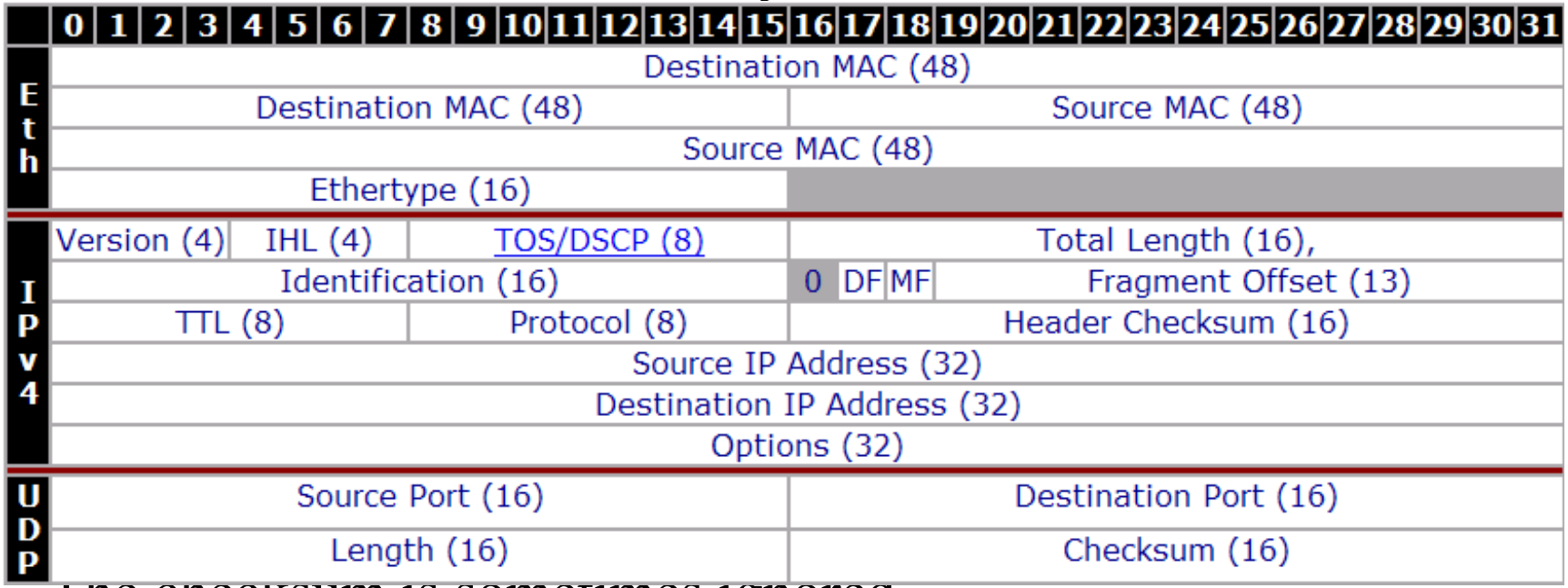
- Once connection established, clients forms request for server.
- Server processes request and replies with piggybacked ack.
- Termination by client (active close)
  - Waits 2MSL (Maximum Segment Lifetime) to deal with lost or wandering IP packets.

UDP

- The User Datagram Protocol. Its characteristics are:-
  - Packet-oriented
  - Connectionless
  - Unreliable
- UDP adds almost nothing to the IP network layer over which it is transported. It just introduces the concept of a **port** (a concept it shares with TCP as we will soon see).
- A port is an abstraction which can be regarded as a transport-layer address (remember the role of the transport layer) which uniquely identifies a particular process (or endpoint) on the destination node

UDP

- The UDP header is very brief...



- The checksum is sometimes ignored…
  - Most datalink layer protocols include some form of error-checking (e.g. Ethernet CRC)
  - For some data types (e.g. VoIP), timely but (slightly) corrupt data is better than late but accurate data

UDP

- Services **listen** on **well-known** ports.
  - DNS on UDP port 53
  - Syslog on 514
  - SIP on 5060
- These are administered by IANA (the Internet Assigned Numbers Authority) and the definitive list is maintained at http://www.iana.org/assignments/port-numbers
- Another good place to look these up is the /etc/services file on a Linux box or the %WinDir% \system32\drivers\etc\services file on Windows

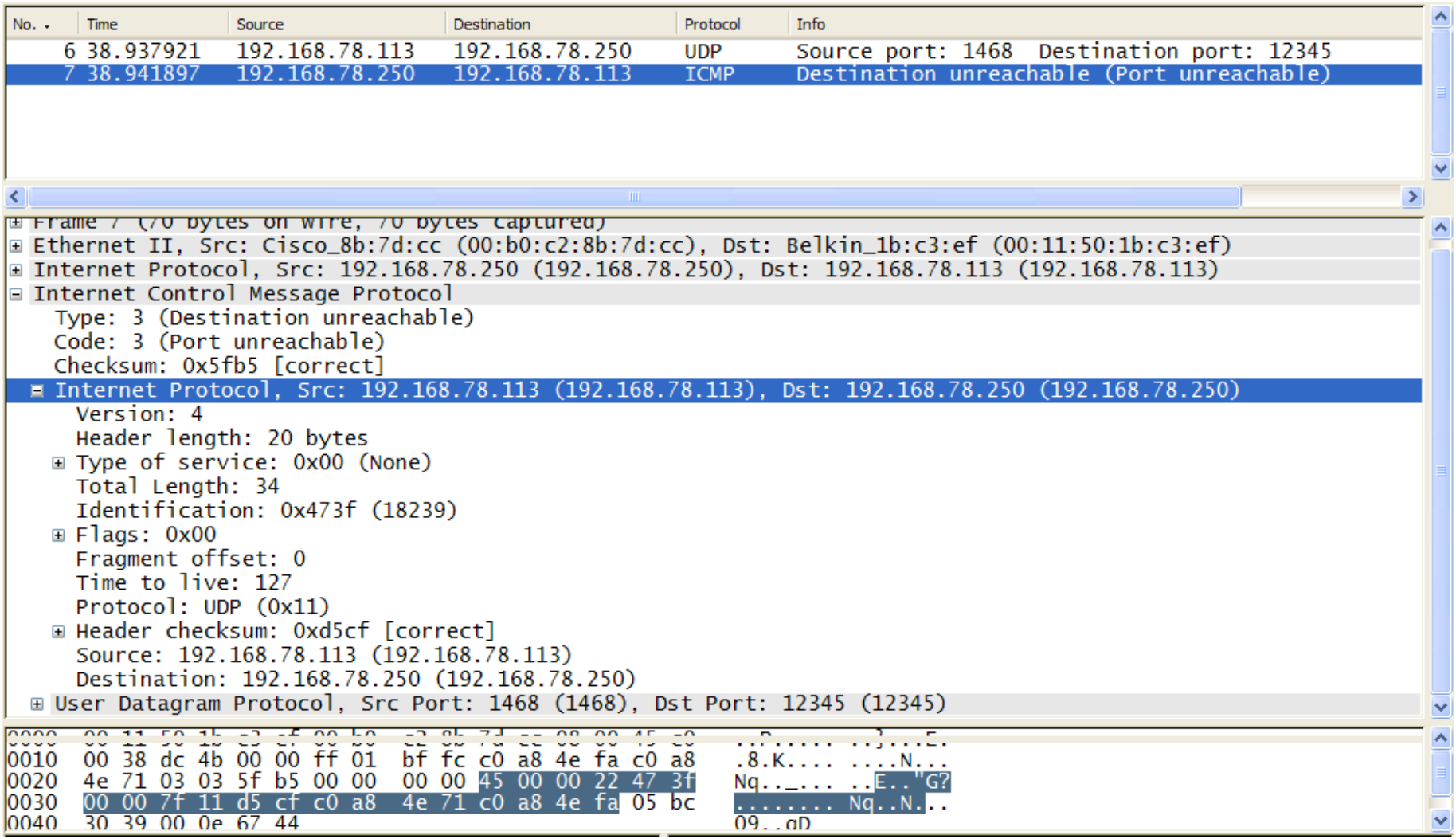
UDP

- When a client wants to communicate with a UDP server, it starts by allocating a randomly-chosen UDP port > 1024. This will be the source UDP port.
- It will then transmit to the server on the destination port (e.g. one of the well-known ports mentioned on the previous slide).
- The server will reply with a UDP packet from the well-known port back to the port the client transmitted the request from
- The combination of (source IP address, source UDP port, destination IP address, destination UDP port) uniquely identifies this “session” (although the concept of a session is artificial with the connectionless UDP protocol)

UDP

- When the client transmits its packet to the server, it has no way to know if there actually is a service (i.e. process) listening on this port at the destination. If not, the network (IP) layer on the server will return an ICMP "Port Unreachable" message

UDP


No. Time Source Destination Protocol Info
6 38.937921 192.168.78.113 192.168.78.250 UDP Source port: 1468 Destination port: 12345
7 38.941897 192.168.78.250 192.168.78.113 ICMP Destination unreachable (Port unreachable)
Ethernet II, Src: Cisco_8b:7d:cc (00:b0:c2:8b:7d:cc), Dst: Belkin_1b:c3:ef (00:11:50:1b:c3:ef)
Internet Protocol, Src: 192.168.78.250 (192.168.78.250), Dst: 192.168.78.113 (192.168.78.113)
Internet Control Message Protocol
Type: 3 (Destination unreachable)
Code: 3 (Port unreachable)
Checksum: 0x5fb5 [correct]
Internet Protocol, Src: 192.168.78.113 (192.168.78.113), Dst: 192.168.78.250 (192.168.78.250)
Version: 4
Header length: 20 bytes
Type of service: 0x00 (None)
Total Length: 34
Identification: 0x473f (18239)
Flags: 0x00
Fragment offset: 0
Time to live: 127
Protocol: UDP (0x11)
Header checksum: 0xd5cf [correct]
Source: 192.168.78.113 (192.168.78.113)
Destination: 192.168.78.250 (192.168.78.250)
User Datagram Protocol, Src Port: 1468 (1468), Dst Port: 12345 (12345)
0010 00 38 dc 4b 00 00 ff 01 bf fc c0 a8 4e fa c0 a8 .8.K.... ....N...
0020 4e 71 03 03 5f b5 00 00 00 00 45 00 00 22 47 3f Nq.._... ..E.."G?
0030 00 00 7f 11 d5 cf c0 a8 4e 71 c0 a8 4e fa 05 bc .... Nq..N...
0040 30 39 00 0e 67 44 09..gD

ICMP, ARP, DNS and DHCP

Keeping everything in working order
by using client protocols from the
TCP\IP family

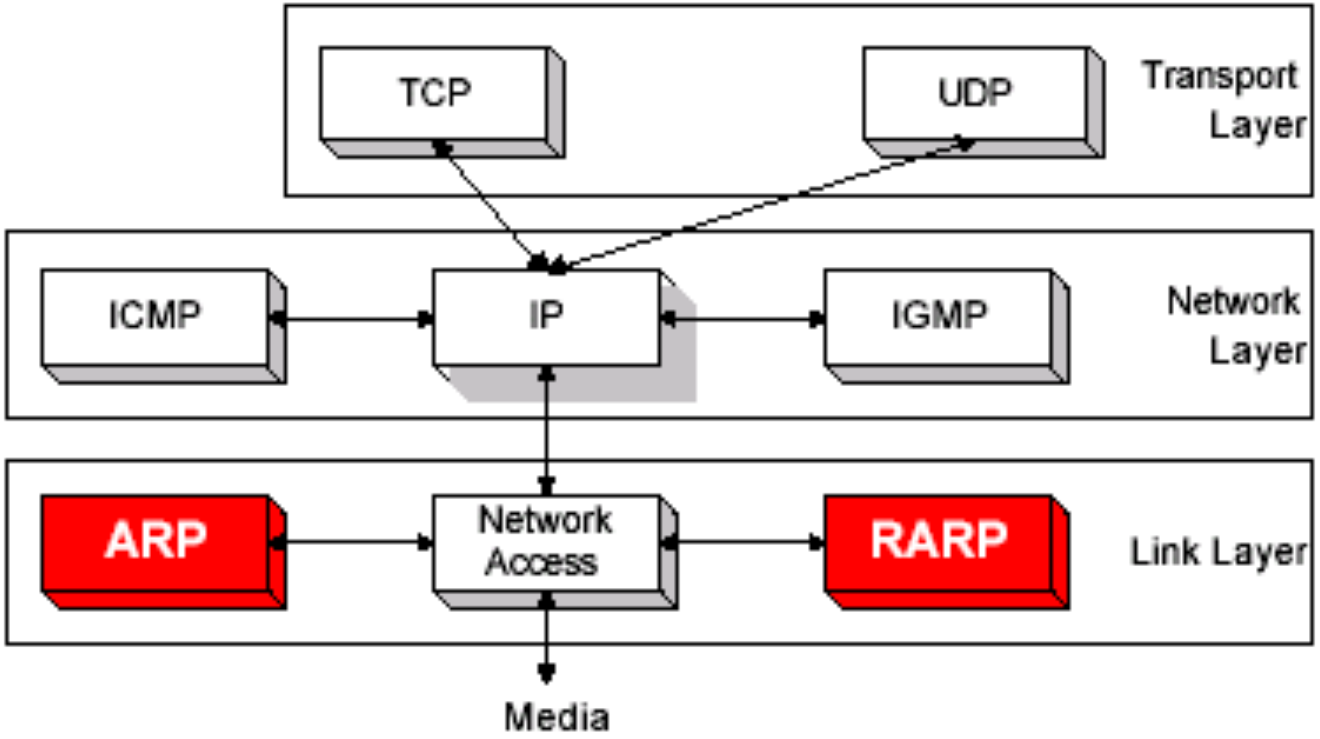
ICMP, ARP, DNS and DHCP

- ICMP is the Internet Control Message Protocol which is and "internal" protocol IP uses for sending control and status information. ICMP message types include *unreachable, source quench, time exceeded, redirect, echo request & reply*
- ARP is the address resolution protocol. It it a protocol by which IP addresses are mapped to corresponding MAC addresses in a LAN environment
- DNS or Domain Name Server translates addresses like **www.dcu.ie** into corresponding IP addresses
- DHCP Dynamic Host Configuration Protocol leases out IP addresses to machines temporarilly attached to a LAN, like your laptops in DCU

Address Resolution Protocol

ARP

ARP



- The Internet is based on IP addresses
- Data link protocols (Ethernet, FDDI, ATM) may have different (MAC) addresses
- The ARP and RARP protocols perform the translation between IP addresses and MAC layer addresses

Address Resolution Protocol

- ARP performs a lookup service that finds a MAC address for a given IP address.
- A system that needs a MAC address for a given IP address broadcasts a query which contains the IP address to all systems on the network.
- If a system receives the query and the IP address in the message matches its own IP address, it sends its MAC address to the sender of the query.
- IP and MAC addresses are the usual but not the only formats available to ARP

Operation of ARP

- Each host maintains a table, the *ARP cache*, temporarily stores the results from previous address resolutions.
- ARP Request is broadcast to all systems on the network.
- In Ethernet, a frame is broadcast when the destination MAC address is set to broadcast address ff:ff:ff:ff:ff:ff.
- A broadcast frame is received and processed by all hosts.
- If a system receives the ARP request and the IP address in the message matches its own IP address, it issues an ARP Reply message to the sender of the query.

Gratuitous ARP

- Every host that sees an ARP Request verifies its ARP cache checking for the sender IP of the ARP Request.
- If such an entry exists, it updates the MAC address with the address in the ARP Request.
- Since ARP Requests are broadcast message, these updates are made by all systems each time an ARP Request is transmitted on the network.
- This feature is exploited in a concept that is called *gratuitous ARP*.

ARP Vulnerabilities

- ARP may be used to redirect traffic intended for a certain IP address to another on the NW
- Broadcast ARP replies with invalid MAC addresses insert incorrect entries into ARP caches.
  - Poison ARP attack

ARP Packet Formats

Hardware type (2 bytes)		Protocol type (2 bytes)
Hardware address length (1 byte)	Protocol address length (1 byte)	Operation code (2 bytes)
Source hardware address*		
Source protocol address*		
Target hardware address*		
Target protocol address*		

* Depends on length of Datalink and Network layer addresses

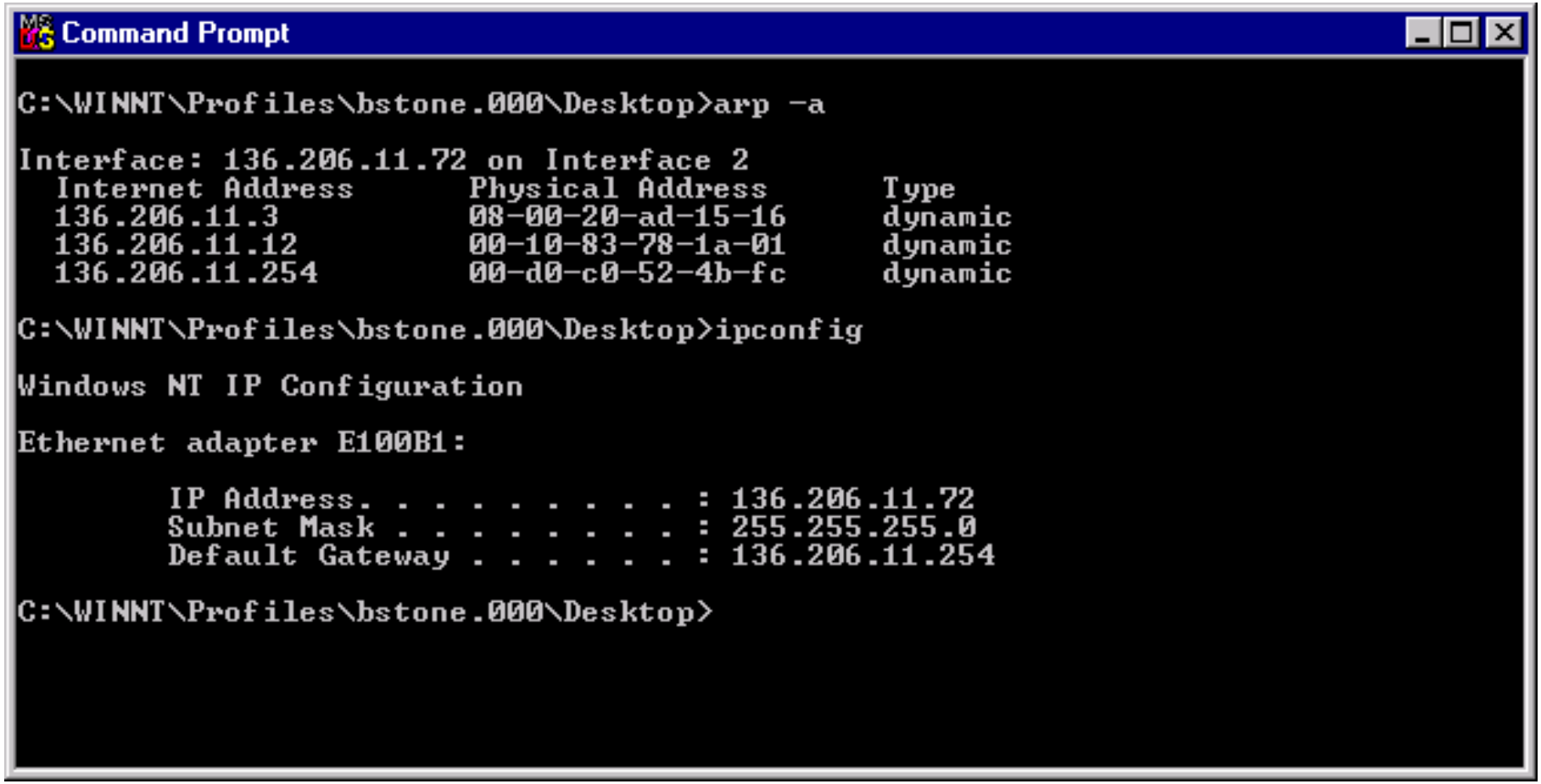
Note: ARP Packet Formats - I

- Ethernet carries ARP, with type set to 0x8060
- ARP message, in IP and Ethernet scenario is 28 bytes (48 bit MAC + 32 bit IP)
- Hardware type is datalink protocol
  - Ethernet = 0x0001, 802 = 0x0006
- Protocol type field is the network layer used
  - IP = 0x8000
- Operation code is 0x0001 for ARP requests and 0x0002 for ARP replies
- Hardware address length and Protocol address length specify length of addresses (MAC-6, IP-4).

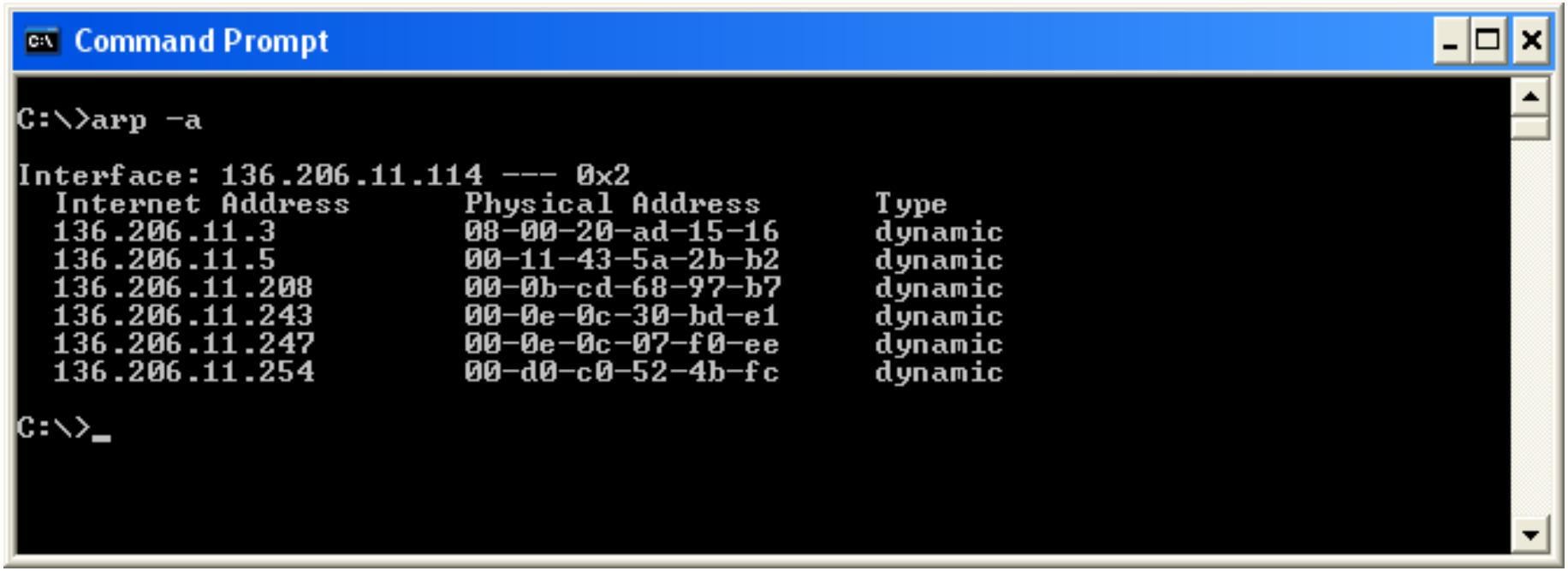
Note: ARP Packet Formats - II

- The next four fields contain the hardware address and the network address of the sender and the intended receiver of the ARP packet.
- The former is referred to as the source and the latter is referred to as the target.

An ARP, the Router, and My PC



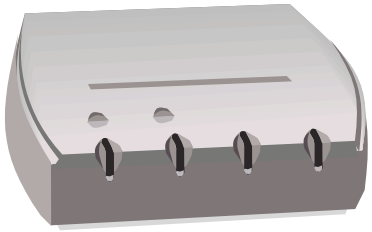
Arp Cache


Command Prompt
C:\>arp -a
Interface: 136.206.11.114 --- 0x2
Internet Address Physical Address Type
136.206.11.3 08-00-20-ad-15-16 dynamic
136.206.11.5 00-11-43-5a-2b-b2 dynamic
136.206.11.208 00-0b-cd-68-97-b7 dynamic
136.206.11.243 00-0e-0c-30-bd-e1 dynamic
136.206.11.247 00-0e-0c-07-f0-ee dynamic
136.206.11.254 00-d0-c0-52-4b-fc dynamic
C:\>

Sample NW and test

A
00:06:58:e3:4d:1d
192.168.0.105

B
00:07:e9:53:87:d9
192.168.0.100



Hub \ DHCP Server
00:06:25:8d:be:1d
192.168.0.1

`arp` Tool

- Issue `arp` command
- Tells you how to use it
- Issue `arp -a` it dumps its cache

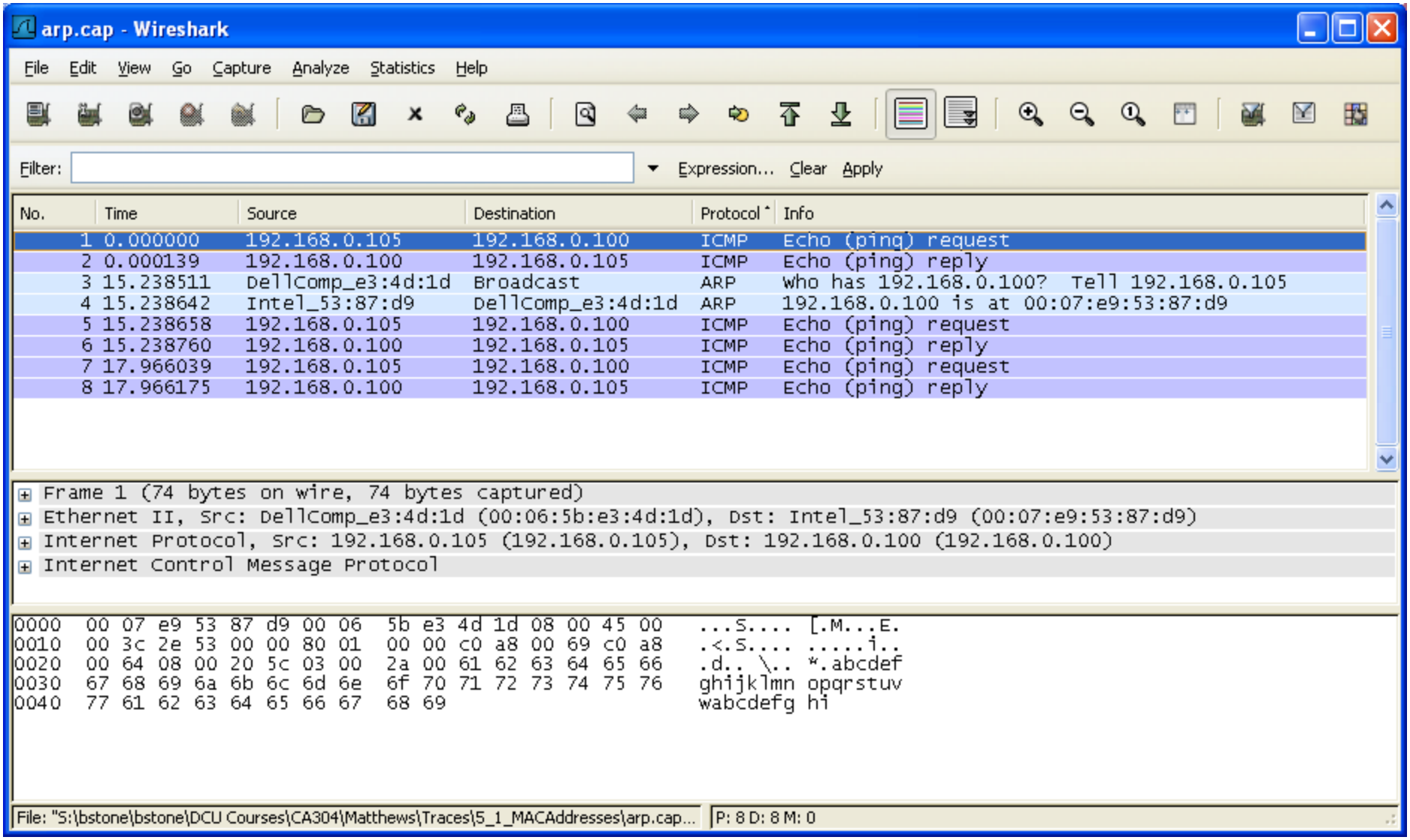
```
>arp –a
Interface 192.168.0.105 --- 0x10004
  Internet Address      Physical Address      Type
  192.168.0.1           00-06-25-8d-be-1d     dynamic
  192.168.0.100         00:07:e9:53:87:d9     dynamic
```

Make Data

- Issue `ping -n 1 192.168.0.100` from A
- Machine A sends a request message to B
- Check out **`arp.cap`** for results
- Note the source and dstination addreses used in this trace.

Make More data

- Now delete the arp cache with
  - `arp -d 192.168.0.100`
- Do second `ping -n 1 192.168.0.100`
- A issues arp request in packet 3 (broadcast addr)
- B replies in packet 4, replenishing arp cache of A and allowing it to issue a ping request in packet 5
- Finally issue third
  - `ping -n 1 192.168.0.100`
- Results are in packet 7 and 8


arp.cap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 192.168.0.105 192.168.0.100 ICMP Echo (ping) request
2 0.000139 192.168.0.100 192.168.0.105 ICMP Echo (ping) reply
3 15.238511 DellComp_e3:4d:1d Broadcast ARP Who has 192.168.0.100? Tell 192.168.0.105
4 15.238642 Intel_53:87:d9 DellComp_e3:4d:1d ARP 192.168.0.100 is at 00:07:e9:53:87:d9
5 15.238658 192.168.0.105 192.168.0.100 ICMP Echo (ping) request
6 15.238760 192.168.0.100 192.168.0.105 ICMP Echo (ping) reply
7 17.966039 192.168.0.105 192.168.0.100 ICMP Echo (ping) request
8 17.966175 192.168.0.100 192.168.0.105 ICMP Echo (ping) reply
Frame 1 (74 bytes on wire, 74 bytes captured)
Ethernet II, Src: DellComp_e3:4d:1d (00:06:5b:e3:4d:1d), Dst: Intel_53:87:d9 (00:07:e9:53:87:d9)
Internet Protocol, Src: 192.168.0.105 (192.168.0.105), Dst: 192.168.0.100 (192.168.0.100)
Internet Control Message Protocol
0000 00 07 e9 53 87 d9 00 06 5b e3 4d 1d 08 00 45 00 ...S.... [.M...E.
0010 00 3c 2e 53 00 00 80 01 00 00 c0 a8 00 69 c0 a8 .<.S.... .....i..
0020 00 64 08 00 20 5c 03 00 2a 00 61 62 63 64 65 66 .d.. \.. *.abcdef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv
0040 77 61 62 63 64 65 66 67 68 69 wabcdefg hi
File: "S:\bstone\bstone\DCU Courses\CA304\Matthews\Traces\5_1_MACAddresses\arp.cap... P: 8 D: 8 M: 0

ARP Exercise

- Replicate the previous experiments on machines in the lab (ground floor is equipped with Wireshark).
- Use `arp` and `ipconfig` to find out the IP and MAC addresses of the machines, clear the caches etc as done in the experiments. You will need 2 machines to do this.
- Something new:
  - Check out the CRC calculations in the frames and account for any discrepencies.
- Use appropriate filters in Wireshark to limit captured traffic to that of interest for the experiment.
- Save your traces in Wireshark to a file.

Proxy ARP

- Proxy ARP is a configuration option for IP routers, where an IP router responds to ARP
- Request that arrive from one of its connected networks for a host that is on another of its connected networks.
- Without Proxy ARP enabled, an ARP Request for a host on a different network is unsuccessful, since routers do not forward ARP packets to another network.

RARP

- Given an Ethernet address, what is the IP?
- RFC 903 – RARP sovves this problem – Broadcasts MAC gets back IP from RARP server.
- Broadcast address stays within 1 domain (router)
- Needs to get further or else have 1 RARP server in each MAC broadcast domain.
- Solution – use BOOTP

BOOTP

- RFCs 951, 1048, 1084
- Use UDP messages, broadcasts forwarded over routers!
- Also provides
  - info on IP of file server with disk image
  - IP address of default router
  - Subnet mask
- Problem: Manual config of IP – MAC, gives rise to errors.

Under the Hood: HTTP

Please view the podcast on Wireshark
in association with this lecture

Introduction

- HyperText Transfer Protocol
- Application Layer
  - Program to Program communications
  - Underlying networks are abstracted
- Client \ server paradigm
  - Typically Web browser like Firefox the client, Apache Webserver the server.

URL

- Universal Resource Locator
  - `http://moodle.dcu.ie/`
  - `http` is Hypertext Transfer protocol
  - `moodle` is machine name on a domain
  - `dcu.ie` is the domain name

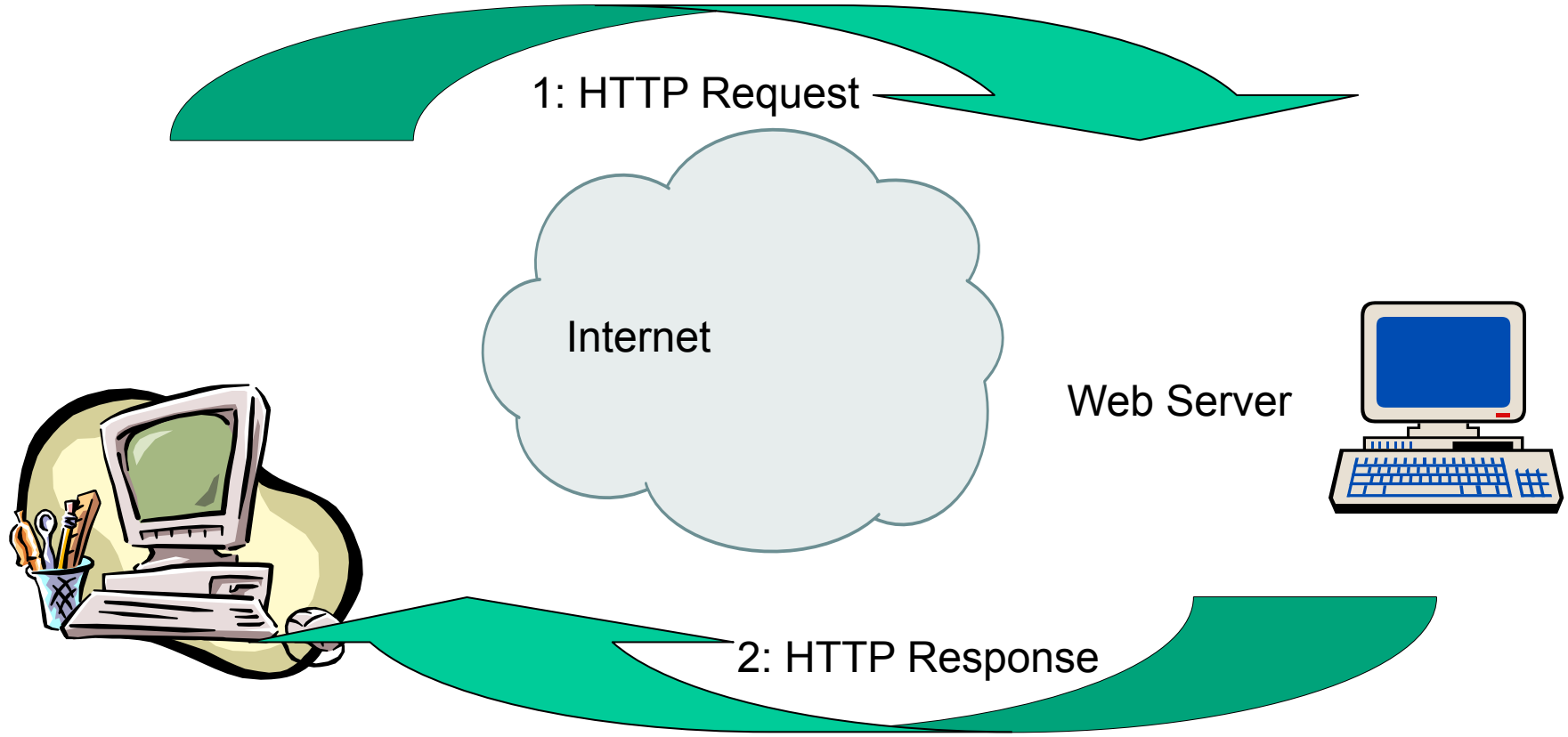
  - `www.computing.dcu.ie`
  - `www.computing` is the webserver for the school

HTML

- Hypertext Markup Language
- Transfers hypertext (text with URLs embedded)
- Also transfers images, music, emails output from programs etc.
- RFC 2616 is the standard for HTTP 1.1

Request\Response

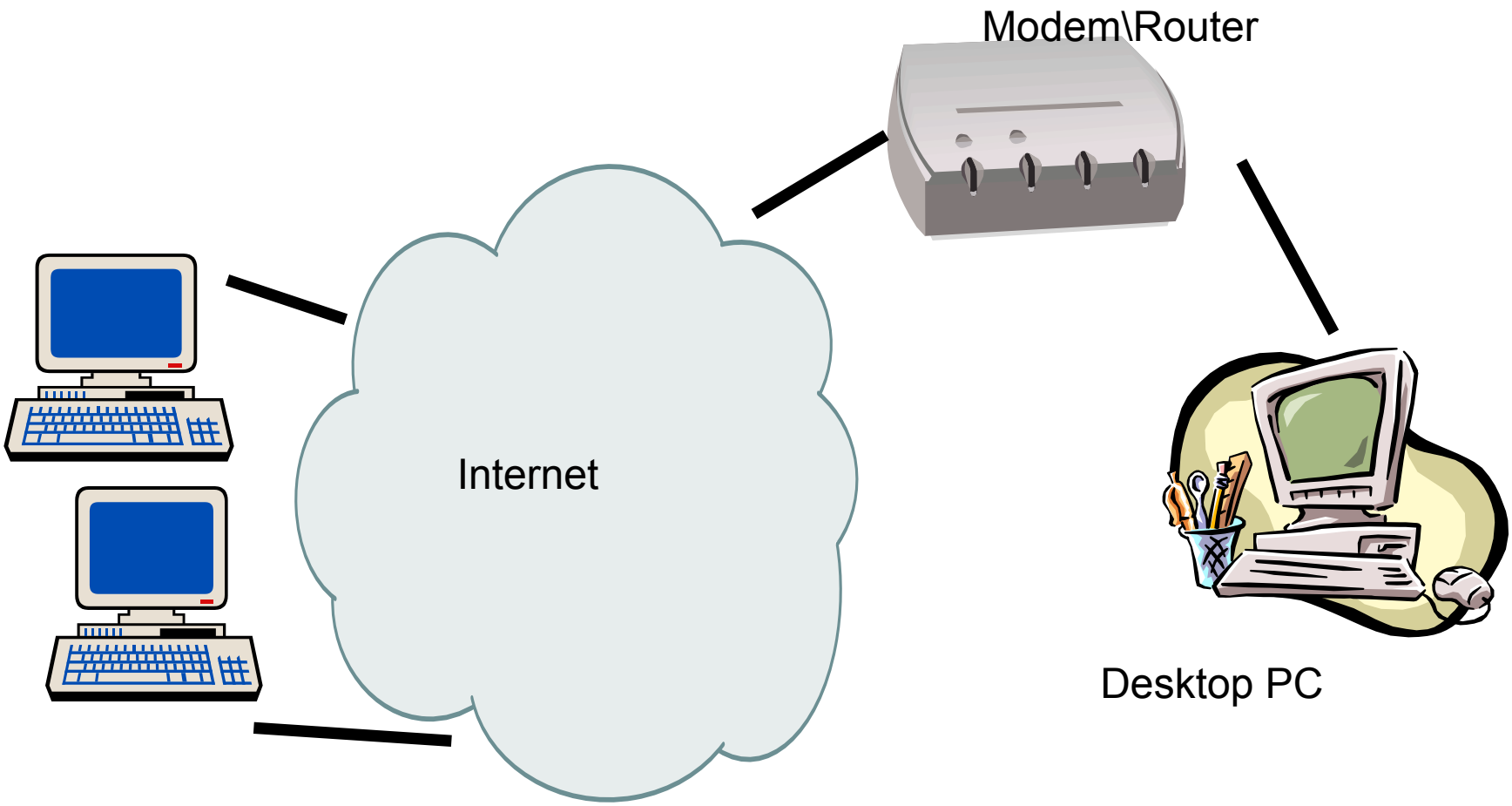
- HTTP works on a request\response mechanism



RFC 2616: HTTP 1.1

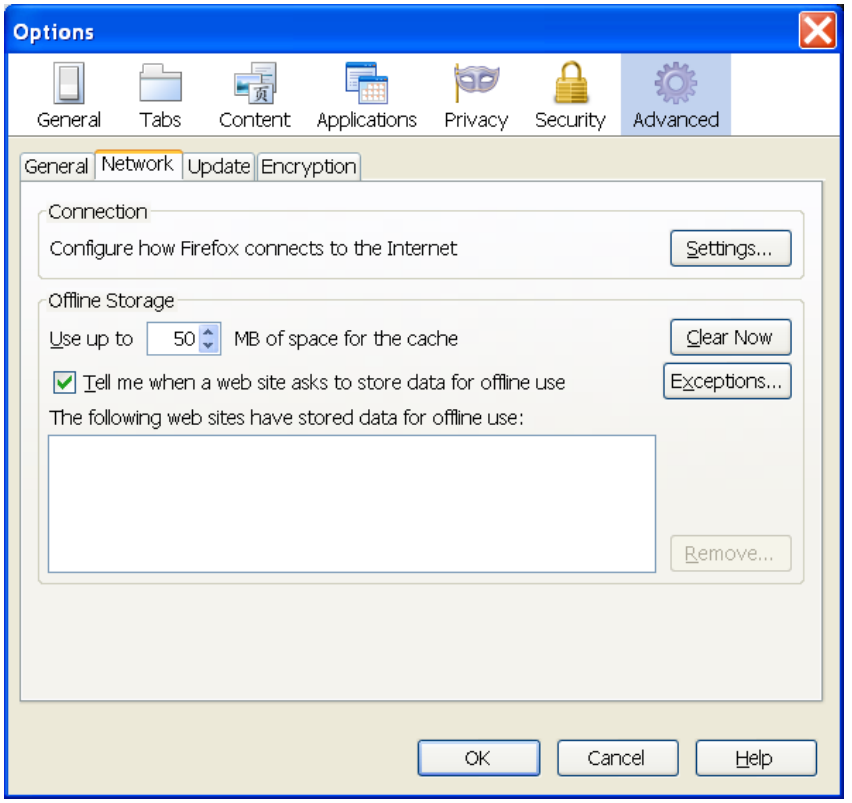
- Standards document for HTTP
- Describes
  - formats for all legal HTTP requests and responses
  - Formats of URLs
  - Controls for caching web pages
  - Persistence of connections
- Allows independent developers to develop their own web servers and clients and have them inter-operable

Example: Configuration



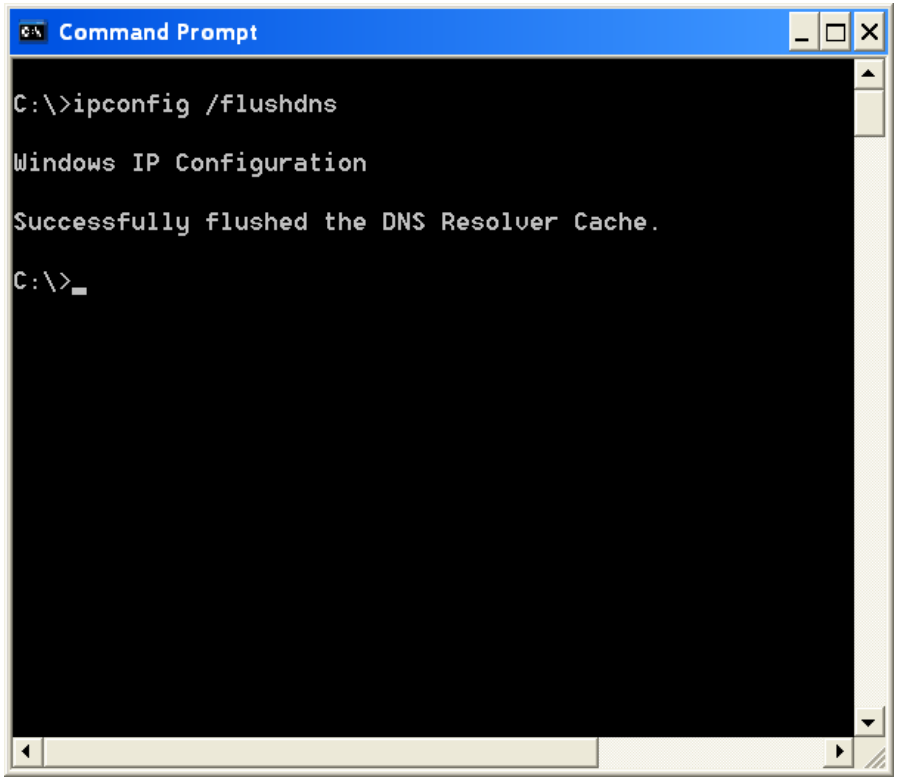
Setup I

- Clear web browser cache
  - In Firefox, under menu <Tools><Options>
    - Advanced tab
  - Click Clear Now
- Now when you ask for a website, you generate new traffic and do not just retrieve something saved earlier.



Setup II

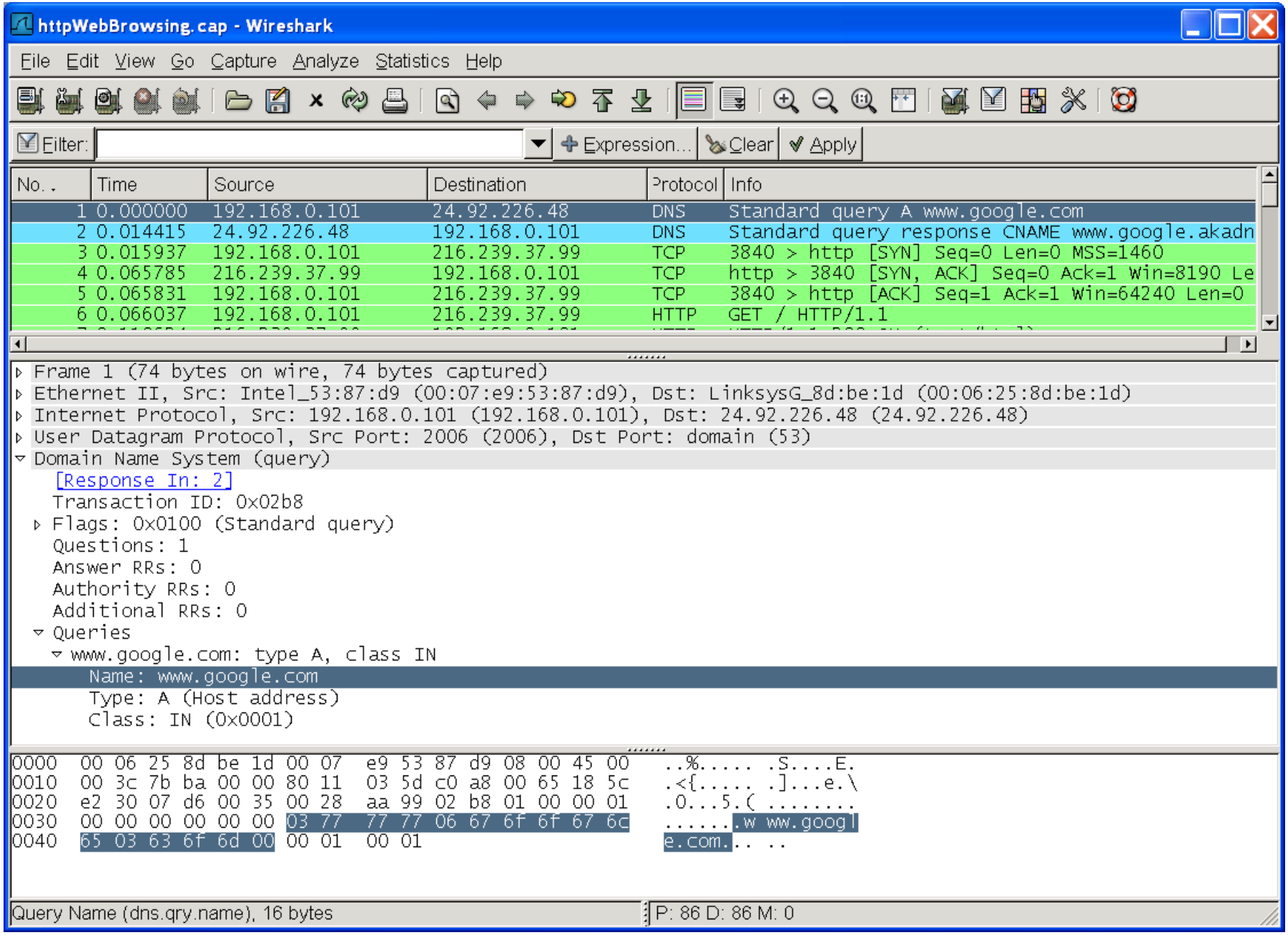
- Clear the DNS Cache
  - DNS remembers the IP address for a corresponding website already visited.
- In a CMD prompt type …
- `Ipconfig /flushdns`

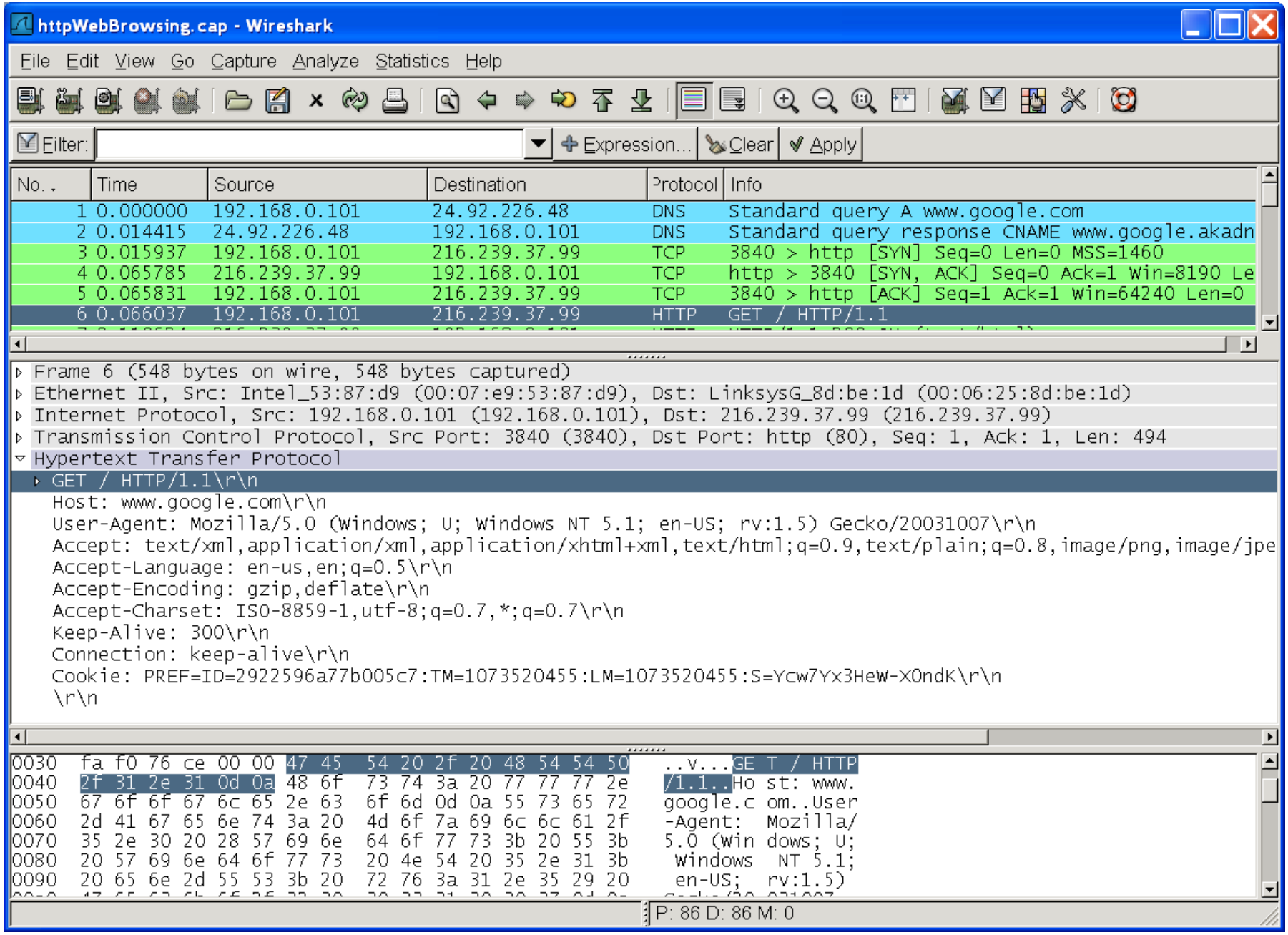


Now all buffers are clear and it is as though we have a brand new connection

Experiment

- Contact Google.com
- See the file on Moodle
- First 2 packets resolve google.com into an IP address using application layer protocol called Domain Name System, DNS
- Now a TCP connection is established, packets 3-5.
- Packet 6 sends "`GET / HTTP1.1\r\n`"


httpWebBrowsing.cap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 192.168.0.101 24.92.226.48 DNS Standard query A www.google.com
2 0.014415 24.92.226.48 192.168.0.101 DNS Standard query response CNAME www.google.akadn
3 0.015937 192.168.0.101 216.239.37.99 TCP 3840 > http [SYN] Seq=0 Len=0 MSS=1460
4 0.065785 216.239.37.99 192.168.0.101 TCP http > 3840 [SYN, ACK] Seq=0 Ack=1 Win=8190 Le
5 0.065831 192.168.0.101 216.239.37.99 TCP 3840 > http [ACK] Seq=1 Ack=1 Win=64240 Len=0
6 0.066037 192.168.0.101 216.239.37.99 HTTP GET / HTTP/1.1
Frame 1 (74 bytes on wire, 74 bytes captured)
Ethernet II, Src: Intel_53:87:d9 (00:07:e9:53:87:d9), Dst: LinksysG_8d:be:1d (00:06:25:8d:be:1d)
Internet Protocol, Src: 192.168.0.101 (192.168.0.101), Dst: 24.92.226.48 (24.92.226.48)
User Datagram Protocol, Src Port: 2006 (2006), Dst Port: domain (53)
Domain Name System (query)
[Response In: 2]
Transaction ID: 0x02b8
Flags: 0x0100 (Standard query)
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
www.google.com: type A, class IN
Name: www.google.com
Type: A (Host address)
Class: IN (0x0001)
0000 00 06 25 8d be 1d 00 07 e9 53 87 d9 08 00 45 00 ..%..... .S....E.
0010 00 3c 7b ba 00 00 80 11 03 5d c0 a8 00 65 18 5c .<{..... .]...e.\
0020 e2 30 07 d6 00 35 00 28 aa 99 02 b8 01 00 00 01 .0...5.(........
0030 00 00 00 00 00 00 03 77 77 77 06 67 6f 6f 67 6c ....w ww.googl
0040 65 03 63 6f 6d 00 00 01 00 01 e.com.. ..
Query Name (dns.qry.name), 16 bytes
P: 86 D: 86 M: 0


httpWebBrowsing.cap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 192.168.0.101 24.92.226.48 DNS Standard query A www.google.com
2 0.014415 24.92.226.48 192.168.0.101 DNS Standard query response CNAME www.google.akadn
3 0.015937 192.168.0.101 216.239.37.99 TCP 3840 > http [SYN] Seq=0 Len=0 MSS=1460
4 0.065785 216.239.37.99 192.168.0.101 TCP http > 3840 [SYN, ACK] Seq=0 Ack=1 Win=8190 Le
5 0.065831 192.168.0.101 216.239.37.99 TCP 3840 > http [ACK] Seq=1 Ack=1 Win=64240 Len=0
6 0.066037 192.168.0.101 216.239.37.99 HTTP GET / HTTP/1.1
Frame 6 (548 bytes on wire, 548 bytes captured)
Ethernet II, Src: Intel_53:87:d9 (00:07:e9:53:87:d9), Dst: LinksysG_8d:be:1d (00:06:25:8d:be:1d)
Internet Protocol, Src: 192.168.0.101 (192.168.0.101), Dst: 216.239.37.99 (216.239.37.99)
Transmission Control Protocol, Src Port: 3840 (3840), Dst Port: http (80), Seq: 1, Ack: 1, Len: 494
Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
Host: www.google.com\r\n
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.5) Gecko/20031007\r\n
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,image/jpe
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
Keep-Alive: 300\r\n
Connection: keep-alive\r\n
Cookie: PREF=ID=2922596a77b005c7:TM=1073520455:LM=1073520455:S=Ycw7Yx3HeW-XOndK\r\n
\r\n
0030 fa f0 76 ce 00 00 47 45 54 20 2f 20 48 54 54 50 ..v...GE T / HTTP
0040 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 77 77 77 2e /1.1..Ho st: www.
0050 67 6f 6f 67 6c 65 2e 63 6f 6d 0d 0a 55 73 65 72 google.c om..User
0060 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f -Agent: Mozilla/
0070 35 2e 30 20 28 57 69 6e 64 6f 77 73 3b 20 55 3b 5.0 (Win dows; U;
0080 20 57 69 6e 64 6f 77 73 20 4e 54 20 35 2e 31 3b Windows NT 5.1;
0090 20 65 6e 2d 55 53 3b 20 72 76 3a 31 2e 35 29 20 en-US; rv:1.5)
P: 86 D: 86 M: 0

HTTP Get Request

- `"GET / HTTP1.1\r\n"`
- GET something, a file called "/" (default)
- \r is carriage return and \n is a line feed
  - Old typewriter terminology, used to separate one header from the next, each successive header has one, check it out.

The Headers I

- **The name** www.google.com\r\n specifies which webserver is being contacted.
  - IP addressed machines may support several web-servers
- **User-agen**t describes web browser and client machine making the request (this setup is quite old, a Mozilla browswe, forerunner of Firefox and the Windows NT operating system)

The Headers II

- Accept headers
  - Server may support several languages, encodings, character sets, these tell server which is preferred.
  - Keep alive and connection headers tell about the TCP connection being used, whether connection should be kept open and for how long.
  - Most connections are *persistent*, allowing multiole requests from same client to server. This improves performance greatly of HTTP 1.1 over HTTP 1.0

HTTP Response

- Packet 7 the response…
  - First HTTP 1.1 is fine with the server
- Headers…
  - **Cache-control**: whether to store copies for future reference. Private here means that this is a specially generated and can be cashed by the user, but not by a group of users on a "shared proxy cache"
  - Lists the types of content and encodings it can accept, `text/html` and `gzip` (compressed)
  - GWS identifies itself as google's own webserver
  - Content length is 1216 long and we get the date.

Multiple GET requests per URL

- Only 1 GET request in packet 8
- Second request generated by the HTML source sent back for processing at the client.
  - `GET /images.logo.gif HTTP/1.1\r\n`
- It asks for the Google logo
- Several requests may be daisy-chained like this
- Multiple requests are very sophisticated now

GNU

- GNU is a famous open source and licensing organisation on the web
- From packet 21 a similar interaction can be seen sorting out where it is
- Packet 26 onwards contains the interactions with it.

Following TCP Stream

- Wireshark allows us to follow particular interactions.
- Looking at the full interaction for Google
  - Select <Analyze> menu, and <Follow TCP Stream> from the menu. You will see every interaction between your client and the server. Each are coloured differently.
- A filter has been automatically entered for you
  - (ip.addr eq 192.168.0.101 and ip.addr eq 216.239.37.99) and (tcp.port eq 3840 and tcp.port eq 80)
  - This is very useful

Multiple TCP Streams

- Packets 35, 41 and 42 open second TCP connection, same IP address as the first, same port (80), but local client port is different, 3842 instead of 3841.
  - This gives rise to a second, parallel connection which speeds up transfer.

Questions

- Isolate the requests sent by the browser to the server
- Visit Google.com
  - <View ><Page Source>
  - Copy it into a file called test.html using Notepad application
  - Open a web browser and drag the file into the browser
  - What website do you see and what is missing and why?
  - Write a colour filter to highlight all of the HTTP requests in the trace
  - Write a colour filter to highlight all of the HTTP responses
  - Combine the two above, HTTP requests and responses only
- Visit three websites, one in DCU, another in Ireland and one abroad
  - compute the average response time for each. Describe how you did the calculation
  - What is the IP address of each

Do Email & HTTP here

Network Layer

Routing Algorithms

Routing Protocols

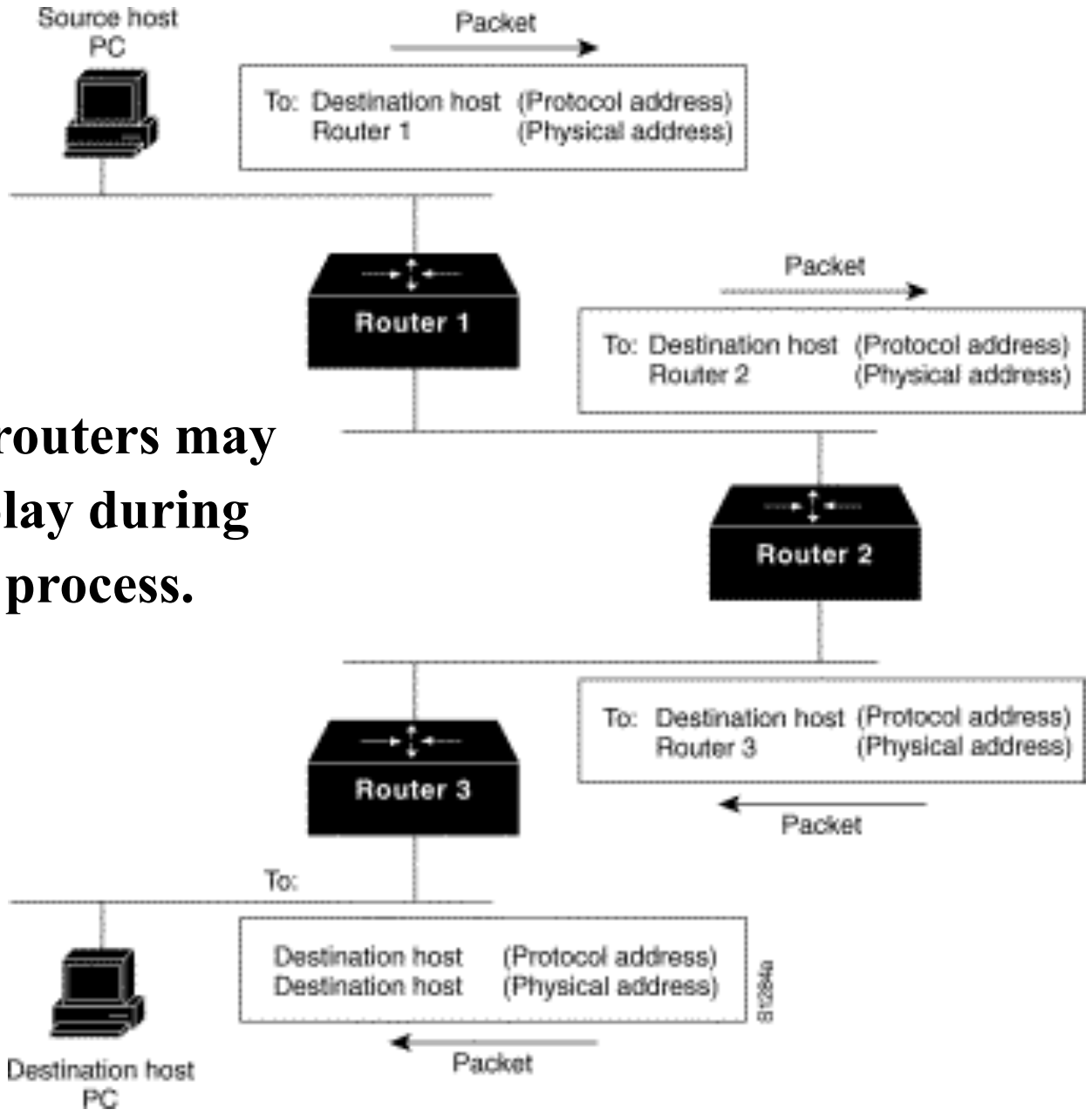
Routing in a Nutshell

- Direct delivery
- Indirect delivery
- Static routing
- Default routing
- Dynamic routing
  - Distance vector routing
  - Link state routing

What Routing Does

- Finds a path to a destination address
- Direct delivery: performed by a host when the destination network is the local network
- Indirect delivery: performed by router when the destination network is NOT the local network
  - Packet is forwarded to default gateway as seen on *ipconfig*

Numerous routers may come into play during the routing process.



Use ipconfig in DCU lab

- Paste in output

Indirect Delivery

- Default gateway delivers packet on behalf of host using a routing table
- Routing table…
  - Destination network (+subnet mask)
  - Next hop (+outgoing interface)
  - Metric (+administrative distance)

Routers

- Initially Unix workstations with several (typically now Ethernet) interface cards.
- More common today to use dedicated specialised hardware, computers with special operating systems and routing hardware for speed.

Routing Protocol Comparison

Protocol	**Complexity**	**Max Size**	**Convergence Time (**to learn about all other routers)	**Reliability**	**Protocol Traffic**
RIP	V simple	16 hops	<480 sec	Not loop safe	High
RIPv2	V simple	16 hops	<480 sec	Not loop safe	High
OSPF	V complex	100000+ NW	fast	High	Low\depends

Routing Metric

- Routing protocols typically find more than one route to destination
- Which to use? Need a metric!
  - Hop count
  - Cost (recipricol value of bandwidth)
  - Load, reliability, time, MTU

Classification

- Older protocols use classful IP addresses (no subnet masks)
- Newer protocols use VLSM and CIDR

THE END of Lectures

Material below here may be duplicated above

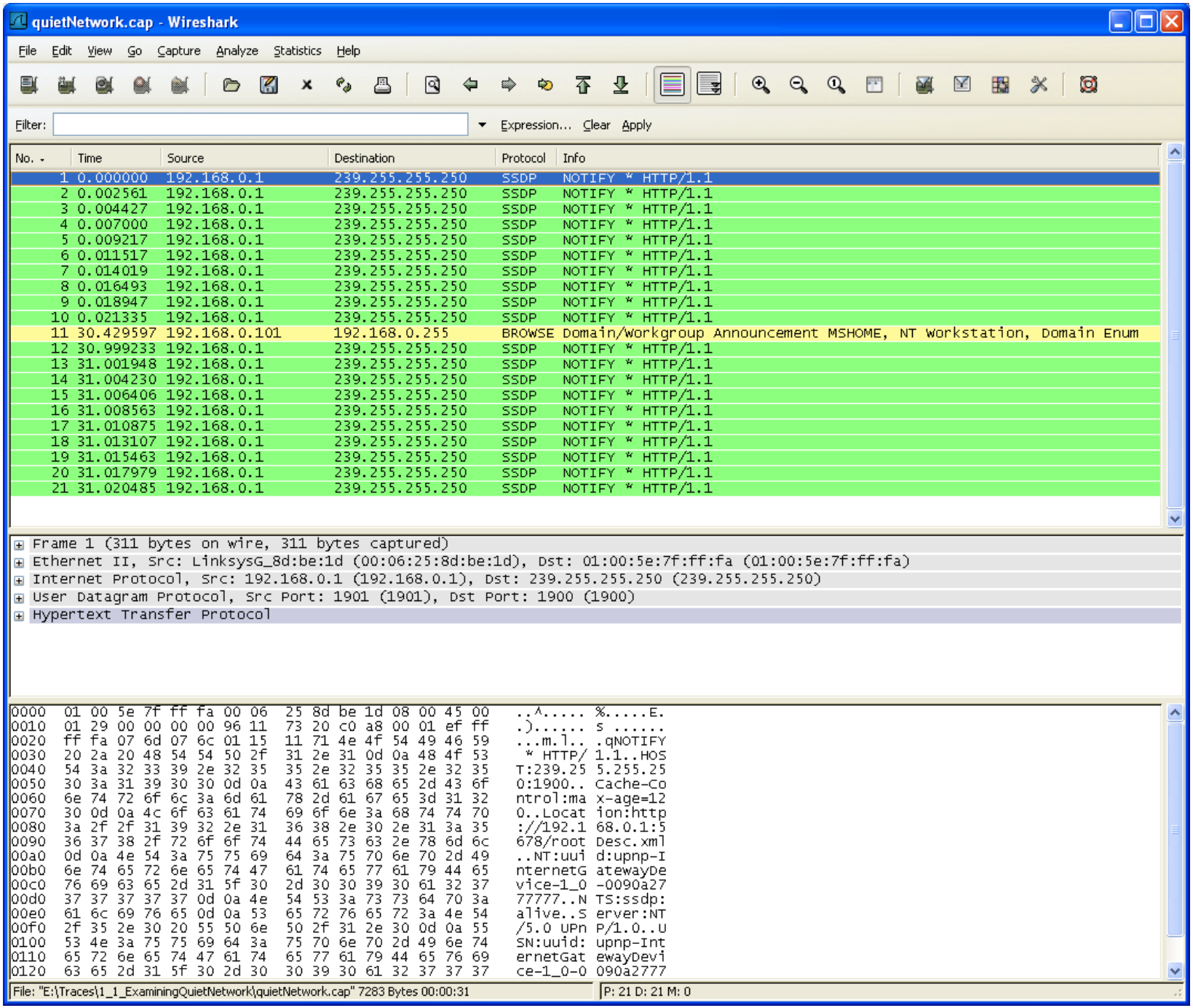
Programming Texts

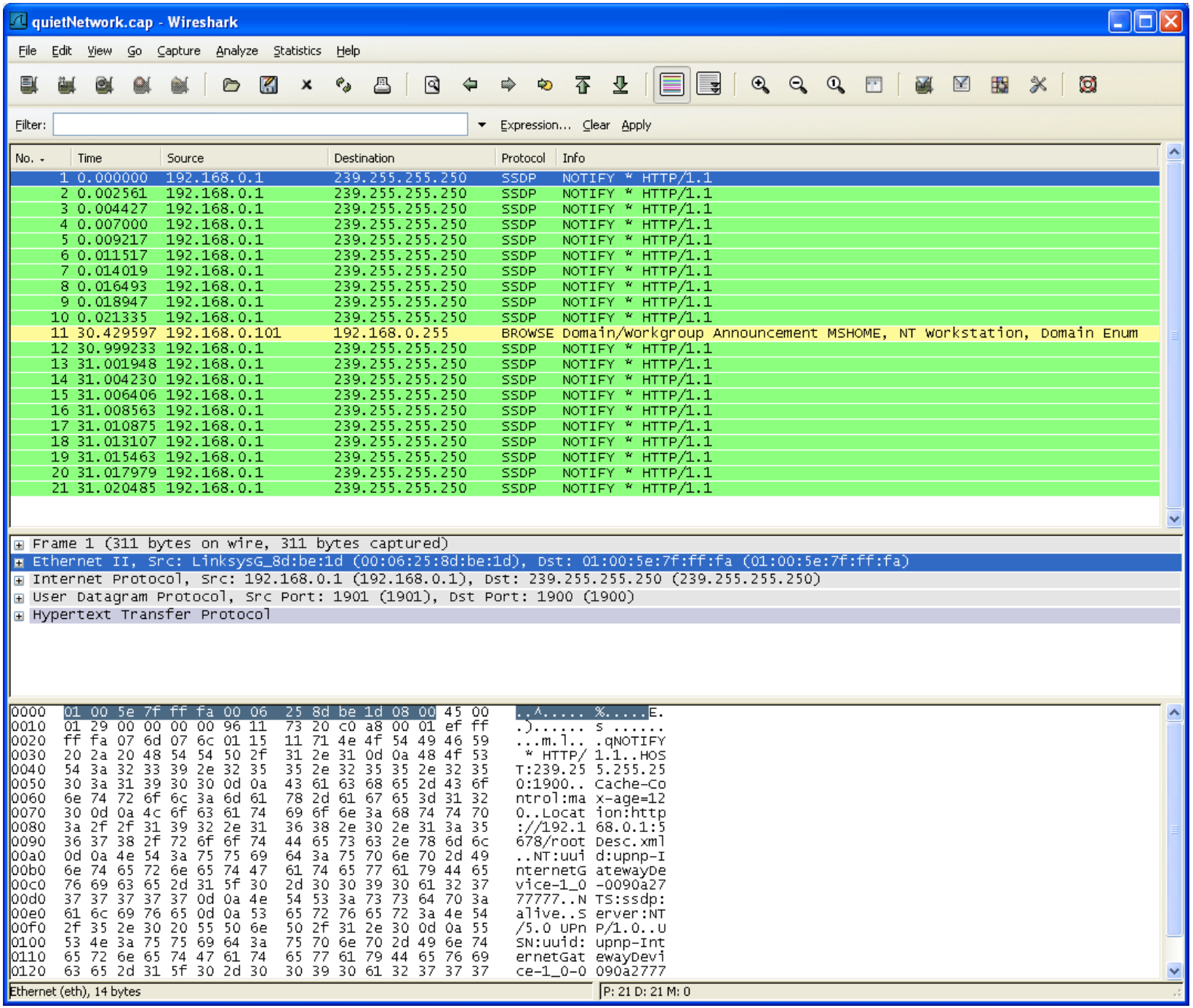
- *Java Network Programming,* Harold, O'Reilly, ISBN 1-56592-870-9
- *Network Programming in Windows NT*, Sinha, Addison Wesley, ISBN 0201590565
- *Windows Sockets Network Programming*, Quinn & Shute, Addison Wesley, ISBN 0201633728

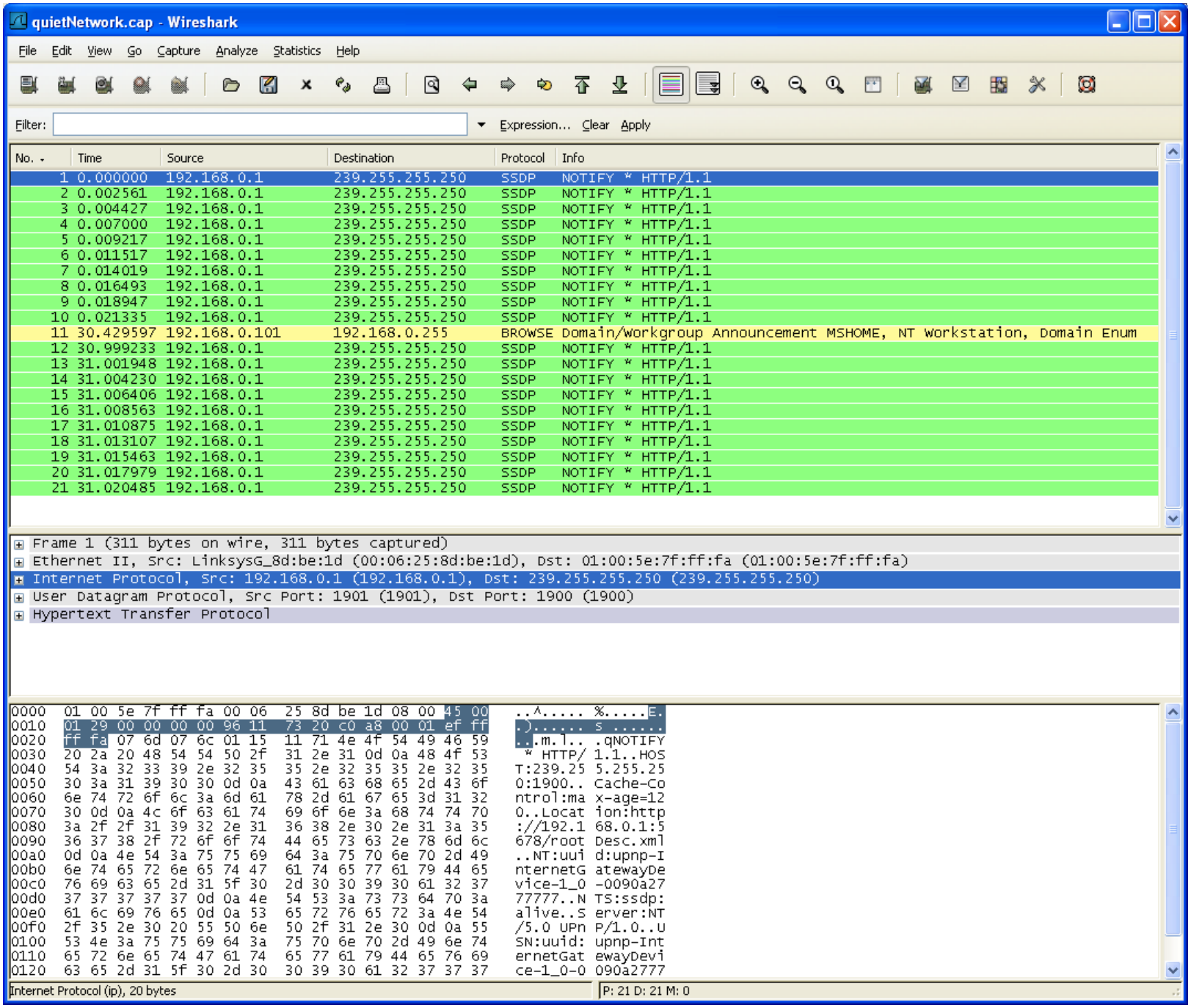
Web References

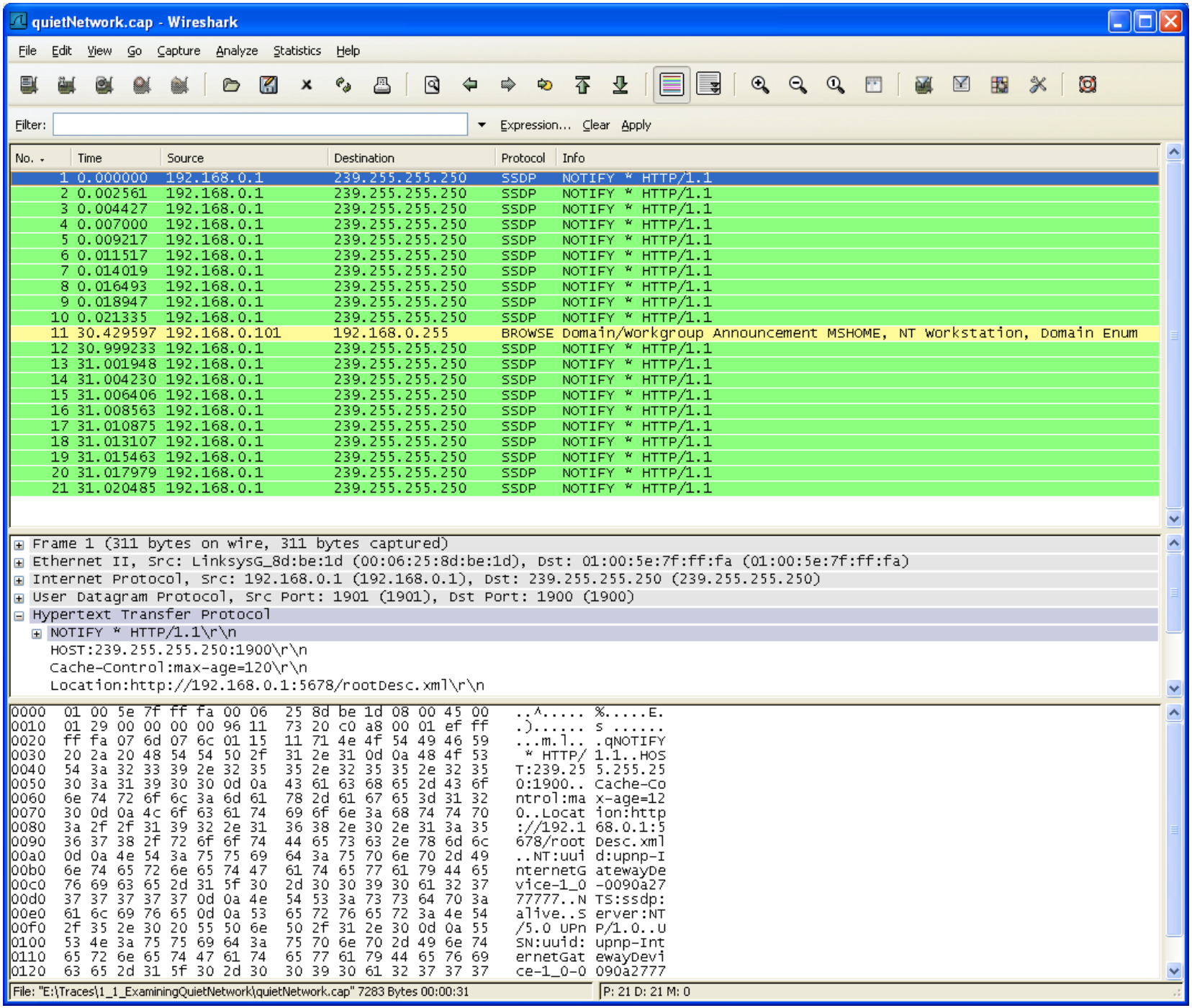
- Cisco Site - A Web book really...
  - http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/index.htm
- Cisco Glossary
  - http://www.cisco.com/univercd/cc/td/doc/cisintwk/ita/index.htm

Introduction


quietNetwork.cap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
2 0.002561 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
3 0.004427 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
4 0.007000 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
5 0.009217 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
6 0.011517 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
7 0.014019 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
8 0.016493 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
9 0.018947 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
10 0.021335 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
11 30.429597 192.168.0.101 192.168.0.255 BROWSE Domain/Workgroup Announcement MSHOME, NT Workstation, Domain Enum
12 30.999233 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
13 31.001948 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
14 31.004230 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
15 31.006406 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
16 31.008563 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
17 31.010875 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
18 31.013107 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
19 31.015463 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
20 31.017979 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
21 31.020485 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
Frame 1 (311 bytes on wire, 311 bytes captured)
Ethernet II, Src: LinksysG_8d:be:1d (00:06:25:8d:be:1d), Dst: 01:00:5e:7f:ff:fa (01:00:5e:7f:ff:fa)
Internet Protocol, Src: 192.168.0.1 (192.168.0.1), Dst: 239.255.255.250 (239.255.255.250)
User Datagram Protocol, Src Port: 1901 (1901), Dst Port: 1900 (1900)
Hypertext Transfer Protocol
0000 01 00 5e 7f ff fa 00 06 25 8d be 1d 08 00 45 00 ..^..... %.....E.
0010 01 29 00 00 00 00 96 11 73 20 c0 a8 00 01 ef ff .)...... s ......
0020 ff fa 07 6d 07 6c 01 15 11 71 4e 4f 54 49 46 59 ...m.l.. .qNOTIFY
0030 20 2a 20 48 54 54 50 2f 31 2e 31 0d 0a 48 4f 53 * HTTP/ 1.1..HOS
0040 54 3a 32 33 39 2e 32 35 35 2e 32 35 35 2e 32 35 T:239.25 5.255.25
0050 30 3a 31 39 30 30 0d 0a 43 61 63 68 65 2d 43 6f 0:1900.. Cache-Co
0060 6e 74 72 6f 6c 3a 6d 61 78 2d 61 67 65 3d 31 32 ntrol:ma x-age=12
0070 30 0d 0a 4c 6f 63 61 74 69 6f 6e 3a 68 74 74 70 0..Locat ion:http
0080 3a 2f 2f 31 39 32 2e 31 36 38 2e 30 2e 31 3a 35 ://192.1 68.0.1:5
0090 36 37 38 2f 72 6f 6f 74 44 65 73 63 2e 78 6d 6c 678/root Desc.xml
00a0 0d 0a 4e 54 3a 75 75 69 64 3a 75 70 6e 70 2d 49 ..NT:uui d:upnp-I
00b0 6e 74 65 72 6e 65 74 47 61 74 65 77 61 79 44 65 nternetG atewayDe
00c0 76 69 63 65 2d 31 5f 30 2d 30 30 39 30 61 32 37 vice-1_0 -0090a27
00d0 37 37 37 37 37 0d 0a 4e 54 53 3a 73 73 64 70 3a 77777..N TS:ssdp:
00e0 61 6c 69 76 65 0d 0a 53 65 72 76 65 72 3a 4e 54 alive..S erver:NT
00f0 2f 35 2e 30 20 55 50 6e 50 2f 31 2e 30 0d 0a 55 /5.0 UPn P/1.0..U
0100 53 4e 3a 75 75 69 64 3a 75 70 6e 70 2d 49 6e 74 SN:uuid: upnp-Int
0110 65 72 6e 65 74 47 61 74 65 77 61 79 44 65 76 69 ernetGat ewayDevi
0120 63 65 2d 31 5f 30 2d 30 30 39 30 61 32 37 37 37 ce-1_0-0 090a2777
File: "E:\Traces\1_1_ExaminingQuietNetwork\quietNetwork.cap" 7283 Bytes 00:00:31
P: 21 D: 21 M: 0


quietNetwork.cap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
2 0.002561 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
3 0.004427 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
4 0.007000 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
5 0.009217 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
6 0.011517 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
7 0.014019 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
8 0.016493 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
9 0.018947 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
10 0.021335 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
11 30.429597 192.168.0.101 192.168.0.255 BROWSE Domain/Workgroup Announcement MSHOME, NT Workstation, Domain Enum
12 30.999233 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
13 31.001948 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
14 31.004230 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
15 31.006406 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
16 31.008563 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
17 31.010875 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
18 31.013107 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
19 31.015463 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
20 31.017979 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
21 31.020485 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
Frame 1 (311 bytes on wire, 311 bytes captured)
Ethernet II, Src: LinksysG_8d:be:1d (00:06:25:8d:be:1d), Dst: 01:00:5e:7f:ff:fa (01:00:5e:7f:ff:fa)
Internet Protocol, Src: 192.168.0.1 (192.168.0.1), Dst: 239.255.255.250 (239.255.255.250)
User Datagram Protocol, Src Port: 1901 (1901), Dst Port: 1900 (1900)
Hypertext Transfer Protocol
0000 01 00 5e 7f ff fa 00 06 25 8d be 1d 08 00 45 00 ..^..... %.....E.
0010 01 29 00 00 00 00 96 11 73 20 c0 a8 00 01 ef ff .)...... s ......
0020 ff fa 07 6d 07 6c 01 15 11 71 4e 4f 54 49 46 59 ...m.l.. .qNOTIFY
0030 20 2a 20 48 54 54 50 2f 31 2e 31 0d 0a 48 4f 53 * HTTP/ 1.1..HOS
0040 54 3a 32 33 39 2e 32 35 35 2e 32 35 35 2e 32 35 T:239.25 5.255.25
0050 30 3a 31 39 30 30 0d 0a 43 61 63 68 65 2d 43 6f 0:1900.. Cache-Co
0060 6e 74 72 6f 6c 3a 6d 61 78 2d 61 67 65 3d 31 32 ntrol:ma x-age=12
0070 30 0d 0a 4c 6f 63 61 74 69 6f 6e 3a 68 74 74 70 0..Locat ion:http
0080 3a 2f 2f 31 39 32 2e 31 36 38 2e 30 2e 31 3a 35 ://192.1 68.0.1:5
0090 36 37 38 2f 72 6f 6f 74 44 65 73 63 2e 78 6d 6c 678/root Desc.xml
00a0 0d 0a 4e 54 3a 75 75 69 64 3a 75 70 6e 70 2d 49 ..NT:uui d:upnp-I
00b0 6e 74 65 72 6e 65 74 47 61 74 65 77 61 79 44 65 nternetG atewayDe
00c0 76 69 63 65 2d 31 5f 30 2d 30 30 39 30 61 32 37 vice-1_0 -0090a27
00d0 37 37 37 37 37 0d 0a 4e 54 53 3a 73 73 64 70 3a 77777..N TS:ssdp:
00e0 61 6c 69 76 65 0d 0a 53 65 72 76 65 72 3a 4e 54 alive..S erver:NT
00f0 2f 35 2e 30 20 55 50 6e 50 2f 31 2e 30 0d 0a 55 /5.0 UPn P/1.0..U
0100 53 4e 3a 75 75 69 64 3a 75 70 6e 70 2d 49 6e 74 SN:uuid: upnp-Int
0110 65 72 6e 65 74 47 61 74 65 77 61 79 44 65 76 69 ernetGat ewayDevi
0120 63 65 2d 31 5f 30 2d 30 30 39 30 61 32 37 37 37 ce-1_0-0 090a2777
Ethernet (eth), 14 bytes
P: 21 D: 21 M: 0


quietNetwork.cap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
2 0.002561 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
3 0.004427 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
4 0.007000 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
5 0.009217 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
6 0.011517 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
7 0.014019 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
8 0.016493 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
9 0.018947 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
10 0.021335 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
11 30.429597 192.168.0.101 192.168.0.255 BROWSE Domain/Workgroup Announcement MSHOME, NT Workstation, Domain Enum
12 30.999233 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
13 31.001948 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
14 31.004230 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
15 31.006406 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
16 31.008563 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
17 31.010875 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
18 31.013107 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
19 31.015463 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
20 31.017979 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
21 31.020485 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
Frame 1 (311 bytes on wire, 311 bytes captured)
Ethernet II, Src: LinksysG_8d:be:1d (00:06:25:8d:be:1d), Dst: 01:00:5e:7f:ff:fa (01:00:5e:7f:ff:fa)
Internet Protocol, Src: 192.168.0.1 (192.168.0.1), Dst: 239.255.255.250 (239.255.255.250)
User Datagram Protocol, Src Port: 1901 (1901), Dst Port: 1900 (1900)
Hypertext Transfer Protocol
0000 01 00 5e 7f ff fa 00 06 25 8d be 1d 08 00 45 00 ..^..... %.....E.
0010 01 29 00 00 00 00 96 11 73 20 c0 a8 00 01 ef ff .)...... s .....
0020 ff fa 07 6d 07 6c 01 15 11 71 4e 4f 54 49 46 59 ...m.l.. .qNOTIFY
0030 20 2a 20 48 54 54 50 2f 31 2e 31 0d 0a 48 4f 53 * HTTP/ 1.1..HOS
0040 54 3a 32 33 39 2e 32 35 35 2e 32 35 35 2e 32 35 T:239.25 5.255.25
0050 30 3a 31 39 30 30 0d 0a 43 61 63 68 65 2d 43 6f 0:1900.. Cache-Co
0060 6e 74 72 6f 6c 3a 6d 61 78 2d 61 67 65 3d 31 32 ntrol:ma x-age=12
0070 30 0d 0a 4c 6f 63 61 74 69 6f 6e 3a 68 74 74 70 0..Locat ion:http
0080 3a 2f 2f 31 39 32 2e 31 36 38 2e 30 2e 31 3a 35 ://192.1 68.0.1:5
0090 36 37 38 2f 72 6f 6f 74 44 65 73 63 2e 78 6d 6c 678/root Desc.xml
00a0 0d 0a 4e 54 3a 75 75 69 64 3a 75 70 6e 70 2d 49 ..NT:uui d:upnp-I
00b0 6e 74 65 72 6e 65 74 47 61 74 65 77 61 79 44 65 nternetG atewayDe
00c0 76 69 63 65 2d 31 5f 30 2d 30 30 39 30 61 32 37 vice-1_0 -0090a27
00d0 37 37 37 37 37 0d 0a 4e 54 53 3a 73 73 64 70 3a 77777..N TS:ssdp:
00e0 61 6c 69 76 65 0d 0a 53 65 72 76 65 72 3a 4e 54 alive..S erver:NT
00f0 2f 35 2e 30 20 55 50 6e 50 2f 31 2e 30 0d 0a 55 /5.0 UPn P/1.0..U
0100 53 4e 3a 75 75 69 64 3a 75 70 6e 70 2d 49 6e 74 SN:uuid: upnp-Int
0110 65 72 6e 65 74 47 61 74 65 77 61 79 44 65 76 69 erneGat ewayDevi
0120 63 65 2d 31 5f 30 2d 30 30 39 30 61 32 37 37 37 ce-1_0-0 090a2777
Internet Protocol (ip), 20 bytes
P: 21 D: 21 M: 0


quietNetwork.cap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
2 0.002561 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
3 0.004427 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
4 0.007000 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
5 0.009217 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
6 0.011517 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
7 0.014019 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
8 0.016493 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
9 0.018947 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
10 0.021335 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
11 30.429597 192.168.0.101 192.168.0.255 BROWSE Domain/Workgroup Announcement MSHOME, NT Workstation, Domain Enum
12 30.999233 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
13 31.001948 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
14 31.004230 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
15 31.006406 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
16 31.008563 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
17 31.010875 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
18 31.013107 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
19 31.015463 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
20 31.017979 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
21 31.020485 192.168.0.1 239.255.255.250 SSDP NOTIFY * HTTP/1.1
Frame 1 (311 bytes on wire, 311 bytes captured)
Ethernet II, Src: LinksysG_8d:be:1d (00:06:25:8d:be:1d), Dst: 01:00:5e:7f:ff:fa (01:00:5e:7f:ff:fa)
Internet Protocol, Src: 192.168.0.1 (192.168.0.1), Dst: 239.255.255.250 (239.255.255.250)
User Datagram Protocol, Src Port: 1901 (1901), Dst Port: 1900 (1900)
Hypertext Transfer Protocol
NOTIFY * HTTP/1.1\r\n
HOST:239.255.255.250:1900\r\n
Cache-Control:max-age=120\r\n
Location:http://192.168.0.1:5678/rootDesc.xml\r\n
0000 01 00 5e 7f ff fa 00 06 25 8d be 1d 08 00 45 00 ..^..... %.....E.
0010 01 29 00 00 00 00 96 11 73 20 c0 a8 00 01 ef ff .)...... s ......
0020 ff fa 07 6d 07 6c 01 15 11 71 4e 4f 54 49 46 59 ...m.l.. .qNOTIFY
0030 20 2a 20 48 54 54 50 2f 31 2e 31 0d 0a 48 4f 53 * HTTP/ 1.1..HOS
0040 54 3a 32 33 39 2e 32 35 35 2e 32 35 35 2e 32 35 T:239.25 5.255.25
0050 30 3a 31 39 30 30 0d 0a 43 61 63 68 65 2d 43 6f 0:1900.. Cache-Co
0060 6e 74 72 6f 6c 3a 6d 61 78 2d 61 67 65 3d 31 32 ntrol:ma x-age=12
0070 30 0d 0a 4c 6f 63 61 74 69 6f 6e 3a 68 74 74 70 0..Locat ion:http
0080 3a 2f 2f 31 39 32 2e 31 36 38 2e 30 2e 31 3a 35 ://192.1 68.0.1:5
0090 36 37 38 2f 72 6f 6f 74 44 65 73 63 2e 78 6d 6c 678/root Desc.xml
00a0 0d 0a 4e 54 3a 75 75 69 64 3a 75 70 6e 70 2d 49 ..NT:uui d:upnp-I
00b0 6e 74 65 72 6e 65 74 47 61 74 65 77 61 79 44 65 nternetG atewayDe
00c0 76 69 63 65 2d 31 5f 30 2d 30 30 39 30 61 32 37 vice-1_0 -0090a27
00d0 37 37 37 37 37 0d 0a 4e 54 53 3a 73 73 64 70 3a 77777..N TS:ssdp:
00e0 61 6c 69 76 65 0d 0a 53 65 72 76 65 72 3a 4e 54 alive..S erver:NT
00f0 2f 35 2e 30 20 55 50 6e 50 2f 31 2e 30 0d 0a 55 /5.0 UPn P/1.0..U
0100 53 4e 3a 75 75 69 64 3a 75 70 6e 70 2d 49 6e 74 SN:uuid: upnp-Int
0110 65 72 6e 65 74 47 61 74 65 77 61 79 44 65 76 69 ernetGat ewayDevi
0120 63 65 2d 31 5f 30 2d 30 30 39 30 61 32 37 37 37 ce-1_0-0 090a2777
File: "E:\Traces\1_1_ExaminingQuietNetwork\quietNetwork.cap" 7283 Bytes 00:00:31
P: 21 D: 21 M: 0

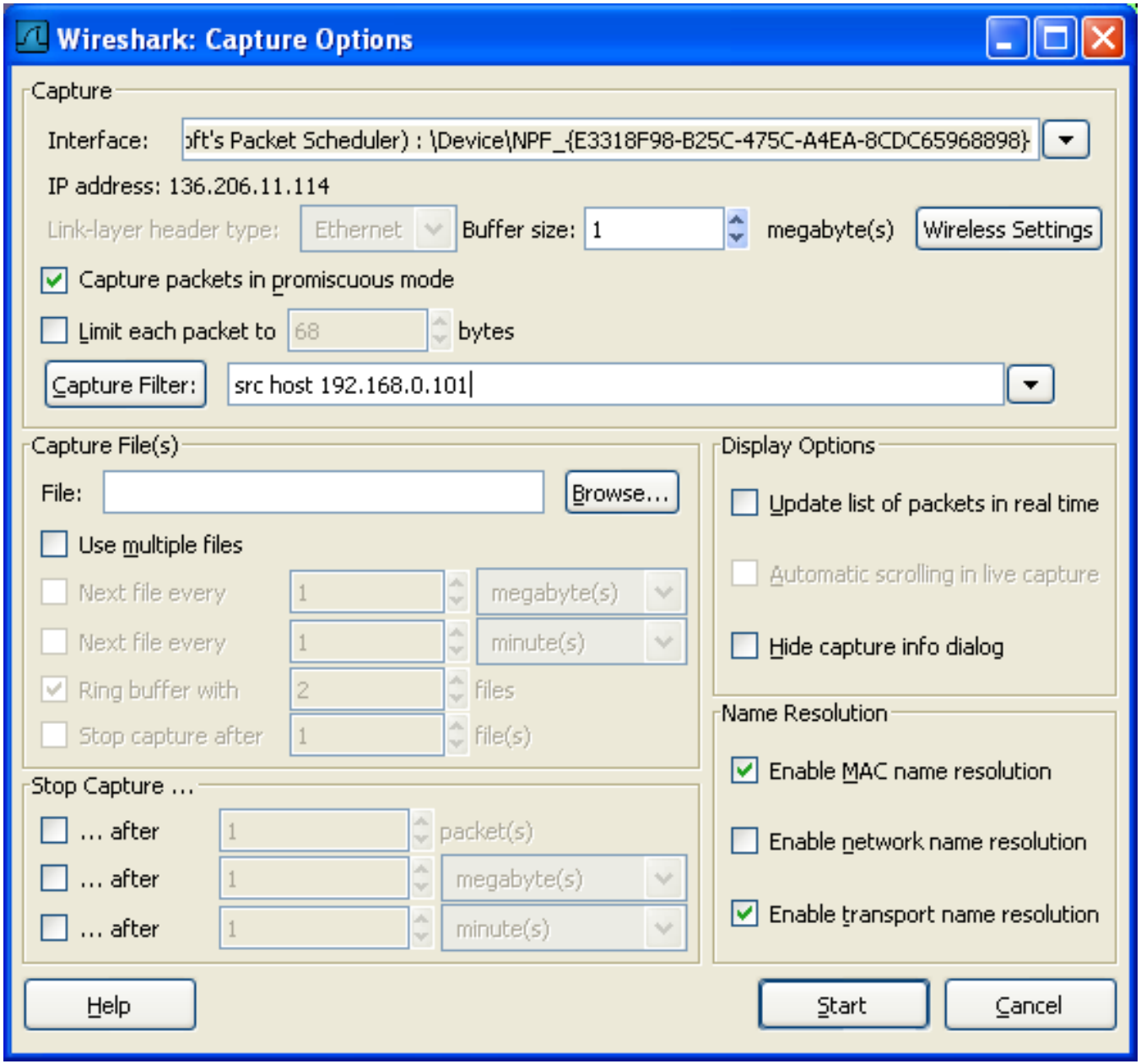
- Use the next slide include it into project spec.

quietNetwork_15 Questions

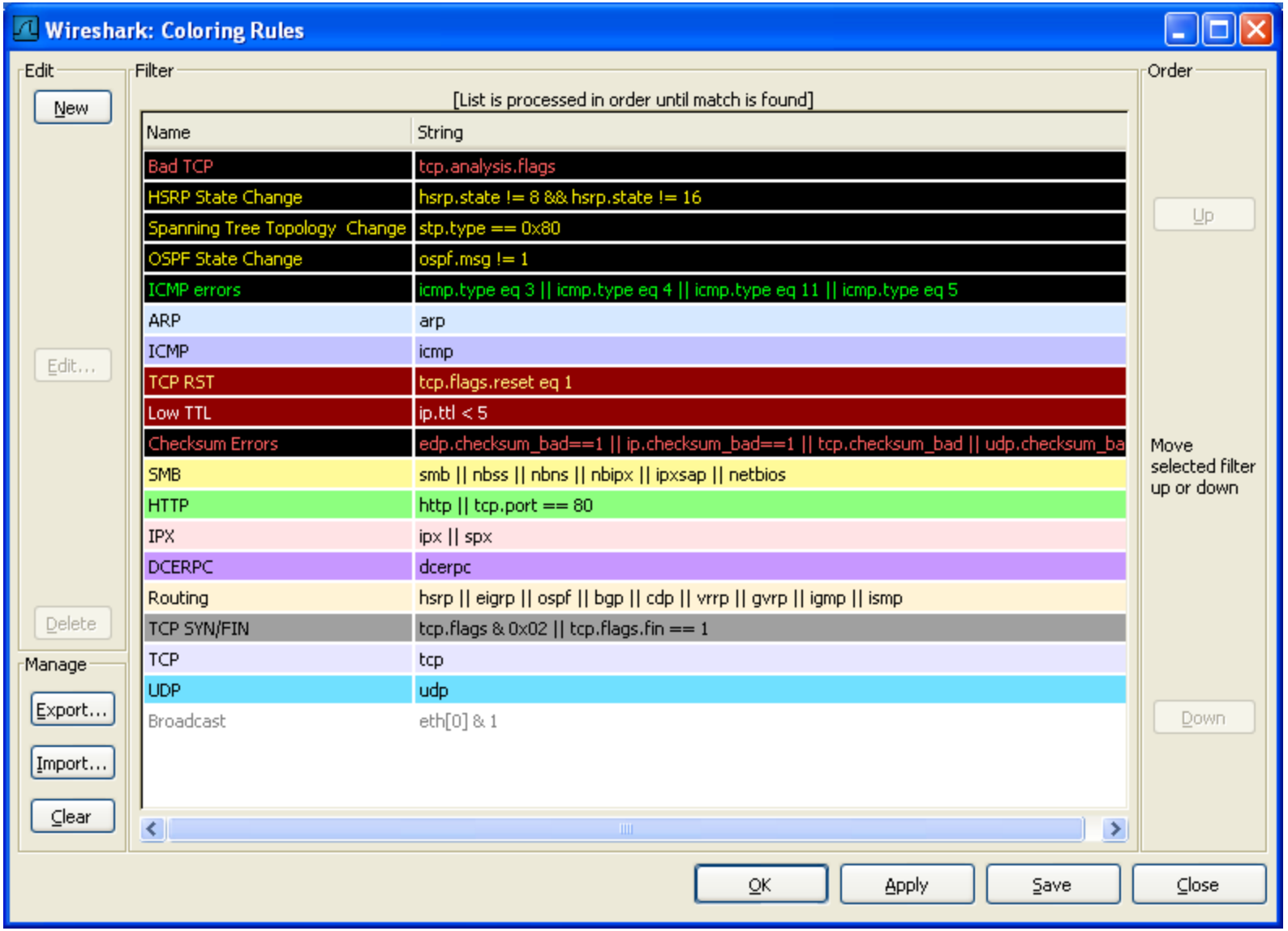
- What is the duration of this trace (2 methods to evaluate)
- What protocols do you see in this trace? (hint: sort by protocol)
- What computers are sending packets in this trace (source IP and Mac addresses)
- Take a trace of the local network, anything interesting?

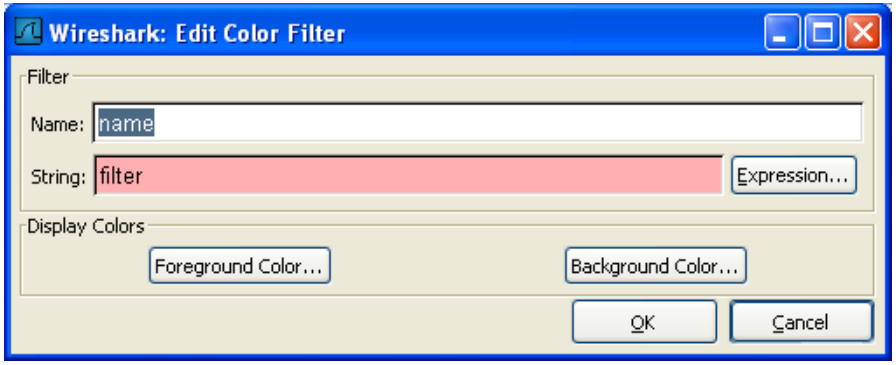
Filters

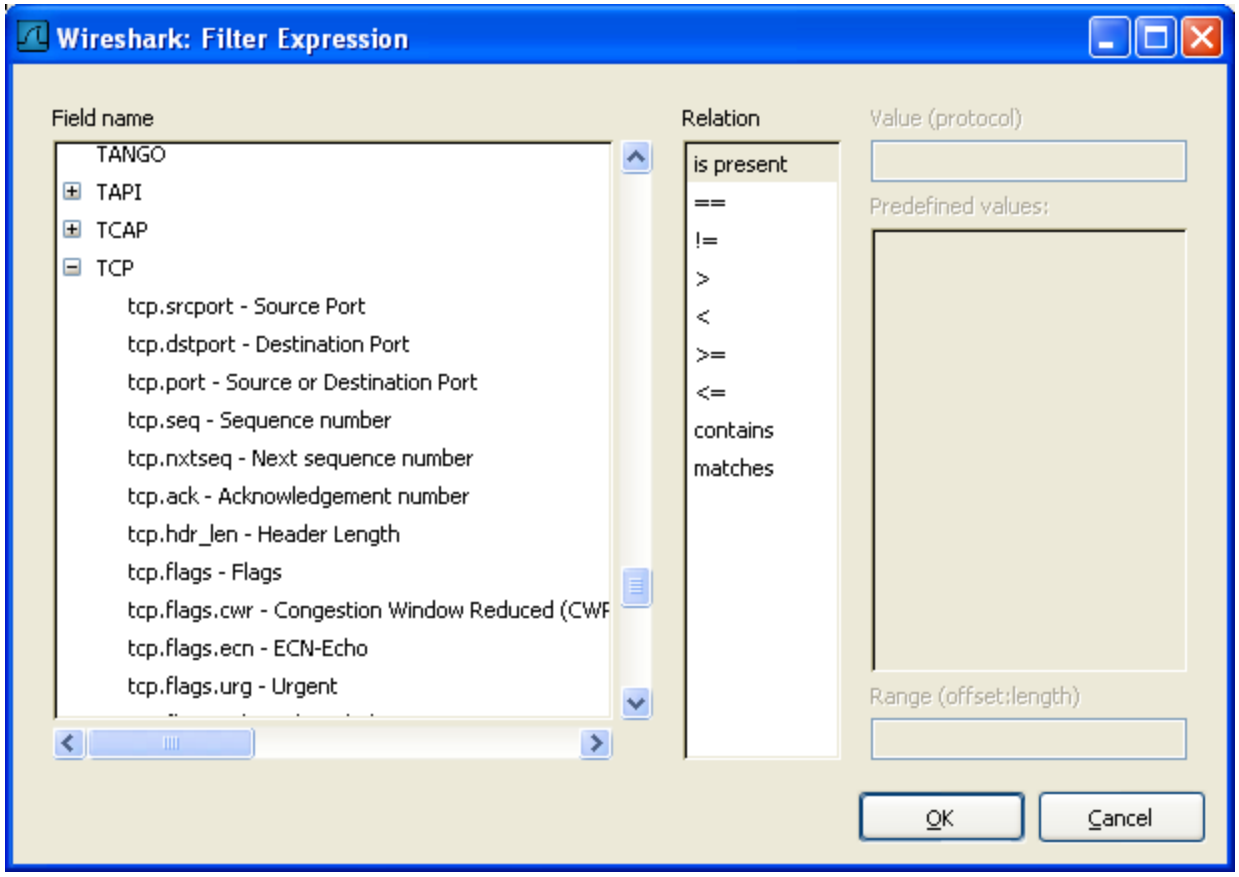
src host
dst
Net
src net
dst net
ether host
ether src

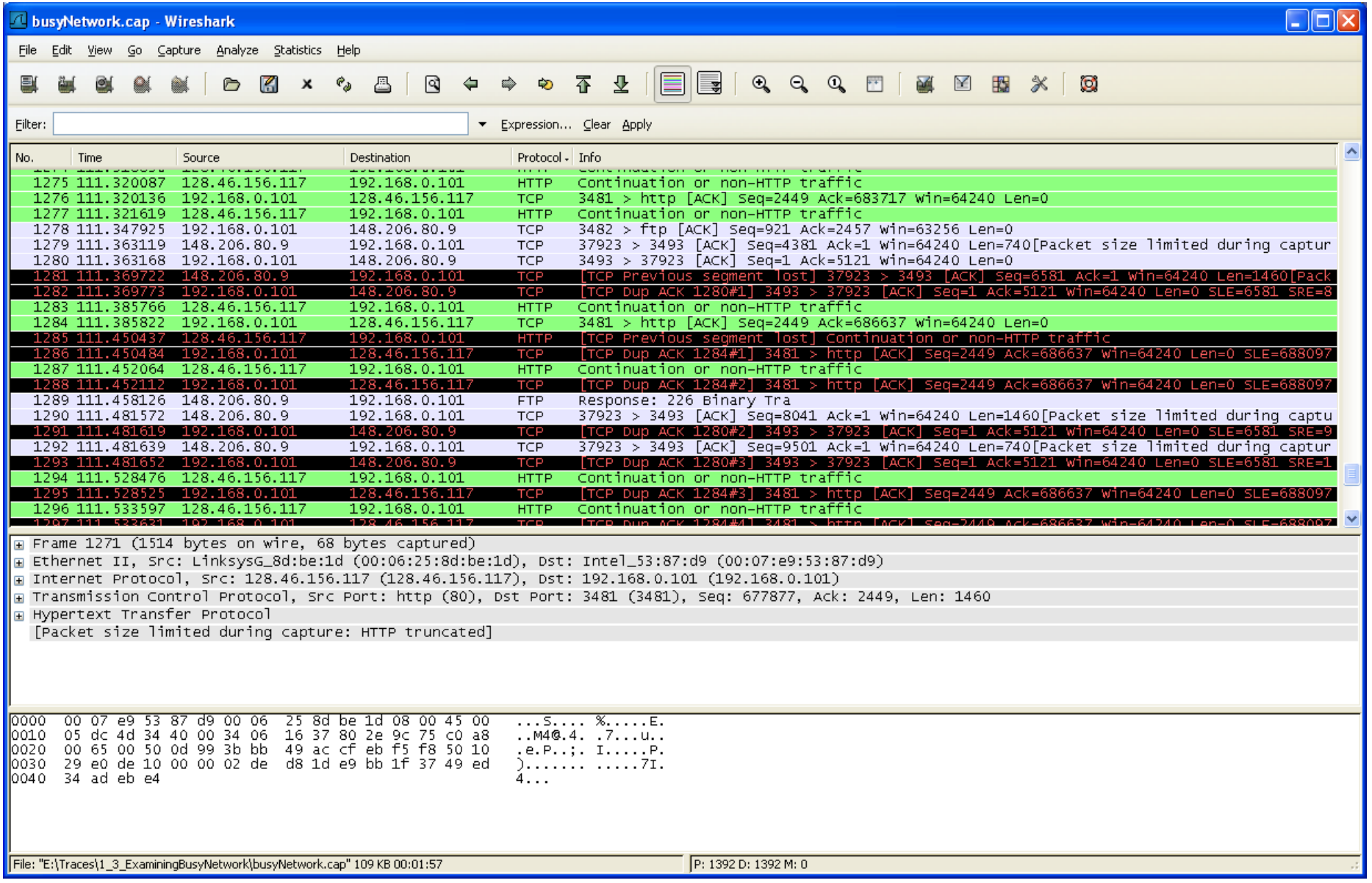


Colour Filters




Wireshark: Edit Color Filter
Filter
Name:
name
String:
filter
Expression...
Display Colors
Foreground Color...
Background Color...
OK
Cancel


Wireshark: Filter Expression
Field name
TANGO
TAPI
TCAP
TCP
tcp.srcport - Source Port
tcp.dstport - Destination Port
tcp.port - Source or Destination Port
tcp.seq - Sequence number
tcp.nxtseq - Next sequence number
tcp.ack - Acknowledgement number
tcp.hdr_len - Header Length
tcp.flags - Flags
tcp.flags.cwr - Congestion Window Reduced (CWF
tcp.flags.ecn - ECN-Echo
tcp.flags.urg - Urgent
Relation
is present
==
!=
>
<
>=
<=
contains
matches
Value (protocol)
Predefined values:
Range (offset:length)
OK
Cancel


busyNetwork.cap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1275 111.320087 128.46.156.117 192.168.0.101 HTTP Continuation or non-HTTP traffic
1276 111.320136 192.168.0.101 128.46.156.117 TCP 3481 > http [ACK] Seq=2449 Ack=683717 Win=64240 Len=0
1277 111.321619 128.46.156.117 192.168.0.101 HTTP Continuation or non-HTTP traffic
1278 111.347925 192.168.0.101 148.206.80.9 TCP 3482 > ftp [ACK] Seq=921 Ack=2457 Win=63256 Len=0
1279 111.363119 148.206.80.9 192.168.0.101 TCP 37923 > 3493 [ACK] Seq=4381 Ack=1 Win=64240 Len=740[Packet size limited during captur
1280 111.363168 192.168.0.101 148.206.80.9 TCP 3493 > 37923 [ACK] Seq=1 Ack=5121 Win=64240 Len=0
1281 111.369722 148.206.80.9 192.168.0.101 TCP [TCP Previous segment lost] 37923 > 3493 [ACK] Seq=6581 Ack=1 Win=64240 Len=1460[Pack
1282 111.369773 192.168.0.101 148.206.80.9 TCP [TCP Dup ACK 1280#1] 3493 > 37923 [ACK] Seq=1 Ack=5121 Win=64240 Len=0 SLE=6581 SRE=8
1283 111.385766 128.46.156.117 192.168.0.101 HTTP Continuation or non-HTTP traffic
1284 111.385822 192.168.0.101 128.46.156.117 TCP 3481 > http [ACK] Seq=2449 Ack=686637 Win=64240 Len=0
1285 111.450437 128.46.156.117 192.168.0.101 HTTP [TCP Previous segment lost] Continuation or non-HTTP traffic
1286 111.450484 192.168.0.101 128.46.156.117 TCP [TCP Dup ACK 1284#1] 3481 > http [ACK] Seq=2449 Ack=686637 Win=64240 Len=0 SLE=688097
1287 111.452064 128.46.156.117 192.168.0.101 HTTP Continuation or non-HTTP traffic
1288 111.452112 192.168.0.101 128.46.156.117 TCP [TCP Dup ACK 1284#2] 3481 > http [ACK] Seq=2449 Ack=686637 Win=64240 Len=0 SLE=688097
1289 111.458126 148.206.80.9 192.168.0.101 FTP Response: 226 Binary Tra
1290 111.481572 148.206.80.9 192.168.0.101 TCP 37923 > 3493 [ACK] Seq=8041 Ack=1 Win=64240 Len=1460[Packet size limited during captu
1291 111.481619 192.168.0.101 148.206.80.9 TCP [TCP Dup ACK 1280#2] 3493 > 37923 [ACK] Seq=1 Ack=5121 Win=64240 Len=0 SLE=6581 SRE=9
1292 111.481639 148.206.80.9 192.168.0.101 TCP 37923 > 3493 [ACK] Seq=9501 Ack=1 Win=64240 Len=740[Packet size limited during captur
1293 111.481652 192.168.0.101 148.206.80.9 TCP [TCP Dup ACK 1280#3] 3493 > 37923 [ACK] Seq=1 Ack=5121 Win=64240 Len=0 SLE=6581 SRE=1
1294 111.528476 128.46.156.117 192.168.0.101 HTTP Continuation or non-HTTP traffic
1295 111.528525 192.168.0.101 128.46.156.117 TCP [TCP Dup ACK 1284#3] 3481 > http [ACK] Seq=2449 Ack=686637 Win=64240 Len=0 SLE=688097
1296 111.533597 128.46.156.117 192.168.0.101 HTTP Continuation or non-HTTP traffic
Frame 1271 (1514 bytes on wire, 68 bytes captured)
Ethernet II, Src: LinksysG_8d:be:1d (00:06:25:8d:be:1d), Dst: Intel_53:87:d9 (00:07:e9:53:87:d9)
Internet Protocol, Src: 128.46.156.117 (128.46.156.117), Dst: 192.168.0.101 (192.168.0.101)
Transmission Control Protocol, Src Port: http (80), Dst Port: 3481 (3481), Seq: 677877, Ack: 2449, Len: 1460
Hypertext Transfer Protocol
[Packet size limited during capture: HTTP truncated]
0000 00 07 e9 53 87 d9 00 06 25 8d be 1d 08 00 45 00 ...S.... %.....E.
0010 05 dc 4d 34 40 00 34 06 16 37 80 2e 9c 75 c0 a8 ..M4@.4. .7...u..
0020 00 65 00 50 0d 99 3b bb 49 ac cf eb f5 f8 50 10 .e.P..;. I.....P.
0030 29 e0 de 10 00 00 02 de d8 1d e9 bb 1f 37 49 ed)....... .....7I.
0040 34 ad eb e4 4...
File: "E:\Traces\1_3_ExaminingBusyNetwork\busyNetwork.cap" 109 KB 00:01:57
P: 1392 D: 1392 M: 0

The End

- That’s it folks!
- Bye !

- Vroom !
- Vroom !