DHCP: Dynamic Host Configuration Protocol RFC 2131

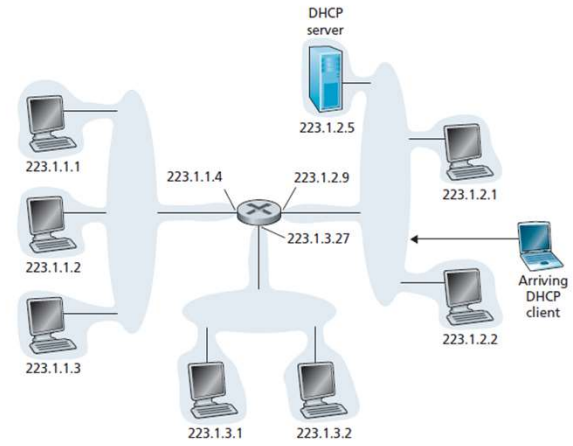


Figure 4.20 ♦ DHCP client-server scenario

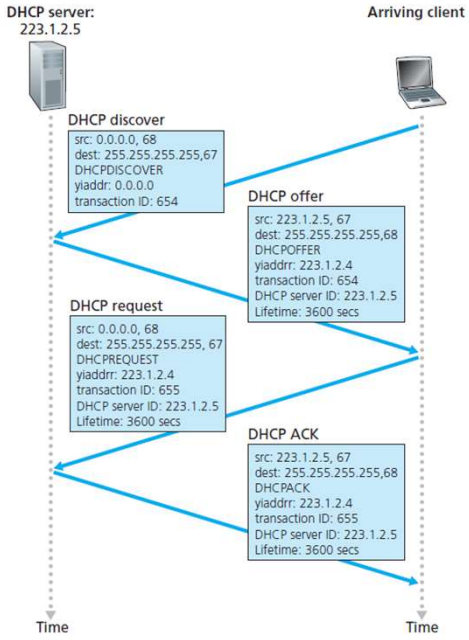
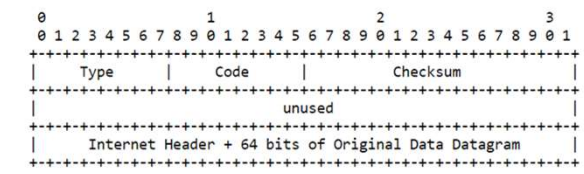


Figure 4.21 ♦ DHCP client-server interaction

https://www.serverbrain.org/network-services-2003/how-the-dhcp-lease-renewal-process-works-1.html

Internet Control Message Protocol [ICMP]

Time Exceeded Message



IP Fields:

Destination Address

The source network and address from the original datagram's data.

ICMP Fields:

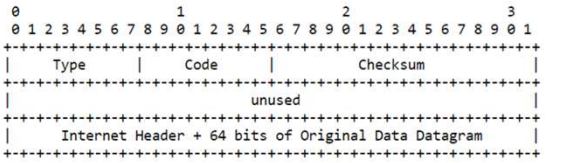
Type

11

Code

0 = time to live exceeded in transit;

1 = fragment reassembly time exceeded.



IP Fields:

Destination Address

The source network and address from the original datagram's data.

ICMP Fields:

Type

3

Code

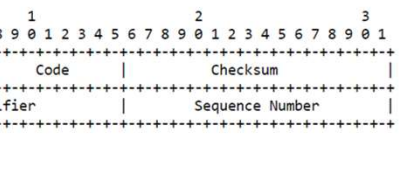
0 = net unreachable;

1 = host unreachable;

2 = protocol unreachable;

3 = port unreachable;

Echo or Echo Reply Message



IP Fields:

Addresses

The address of the source in an echo message will be the destination of the echo reply message. To form an echo reply message, the source and destination addresses are simply reversed, the type code changed to 0, and the checksum recomputed.

IP Fields:

Type

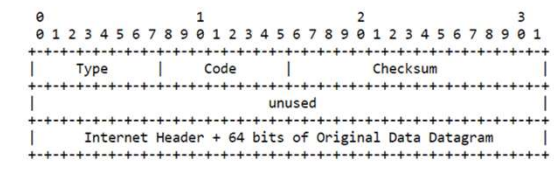
8 for echo message;

0 for echo reply message.

Code

0

Source Quench Message



IP Fields:

Destination Address

The source network and address of the original datagram's data.

ICMP Fields:

Type

4

Code

0

Dr. Gaurav Varshney IIT Jammu

ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

Figure 4.23 ♦ ICMP message types

Dr. Gaurav Varshney IIT Jammu

```
Internet Protocol Version 4, Src: 192.168.252.125, Dst: 142.250.194.238
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4d2c [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 47 (0x002f)
    Sequence Number (LE): 12032 (0x2f00)
    [Response frame: 148]
  Data (32 bytes)
      Data: 6162636465666768696a6b6c6d6e6f707172737475767761626364656667 6
      [Length: 32]
```

```
Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x552c [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 47 (0x002f)
    Sequence Number (LE): 12032 (0x2f00)
    [Request frame: 147]
    [Response time: 19.659 ms]
  Data (32 bytes)
      Data: 6162636465666768696a6b6c6d6e6f7071727374757677616263646566767 6869
      [Length: 32]
```

```
Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (Time to live exceeded in transit)
    Checksum: 0xf4ff [correct]
    [Checksum Status: Good]
    Unused: 00000000
  Internet Protocol Version 4, Src: 192.168.252.125, Dst: 142.250.194.238
      0100 .... = Version: 4
      .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 92
      Identification: 0x85c8 (34248)
    > Flags: 0x00
      Fragment Offset: 0
    > Time to Live: 1
      Protocol: ICMP (1)
      Header Checksum: 0x24ca [validation disabled]
      [Header checksum status: Unverified]
      Source Address: 192.168.252.125
      Destination Address: 142.250.194.238
  Internet Control Message Protocol
      Type: 8 (Echo (ping) request)
      Code: 0
      Checksum: 0xf7cd [unverified] [in ICMP error packet]
      [Checksum Status: Unverified]
```


Network Address Translation

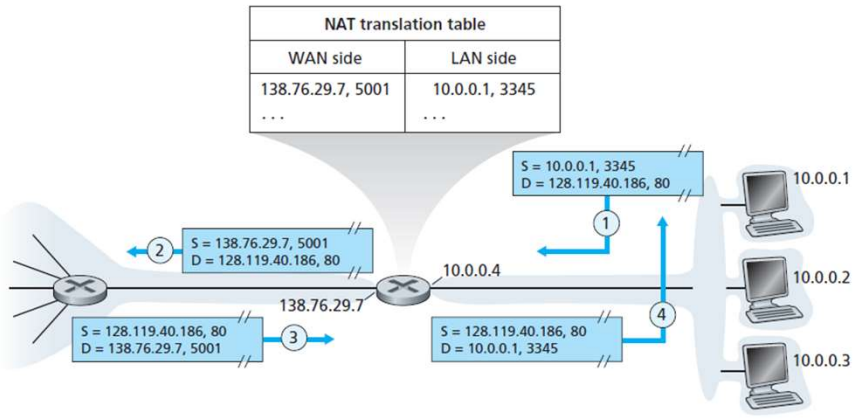


Figure 4.22 ♦ Network address translation

Link Layer Services and Implementation

- Services
  - Framing: Almost all link-layer protocols encapsulate each network-layer datagram within a link-layer frame before transmission over the link
  - Medium Access Control: A medium access control (MAC) protocol specifies the rules by which a frame is transmitted onto the link
  - Reliable Delivery
  - Error Detection and Correction

The link layer is implemented in a **network adapter**, also sometimes known as a **network interface card (NIC)**
Intel 8254x controller [Intel 2012] implements the Ethernet protocols the Atheros AR5006 [Atheros 2012] controller implements the 802.11 Wi-Fi protocols

The software components of the link layer implement higher-level link layer functionality such as assembling link-layer addressing information and activating the controller hardware.

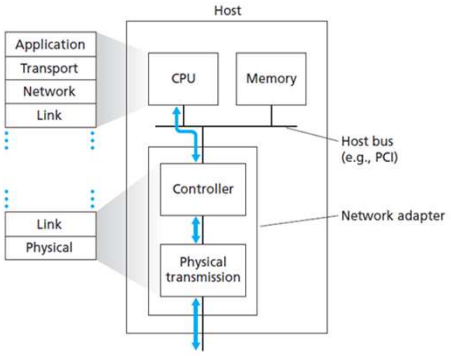


Figure 5.2 ♦ Network adapter: its relationship to other host components and to protocol stack functionality

On the receiving side, link-layer software responds to controller interrupts (e.g., due to the receipt of one or more frames), handling error conditions and passing a datagram up to the network layer.

Dr. Gaurav Varshney IIT Jammu

Error Detection and Correction

- At the sending node, data, *D,* to be protected against bit errors is augmented with error-detection and -correction bits (*EDC*).
- Parity Checks
  - The simplest form of error detection is the use of a single **parity bit**.
  - In an even parity scheme, the sender simply includes one additional bit and chooses its value such that the total number of 1s in the d + 1 bits is even.
  - The receiver need only count the number of 1s in the received d + 1 bits
  - But what happens if an even number of bit errors occur?
  - With this two-dimensional parity scheme, the parity of both the column and the row containing the flipped bit will be in error.

The ability of the receiver to both detect and correct errors is known as **forward error correction (FEC)**.

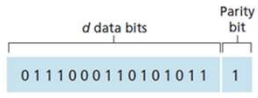
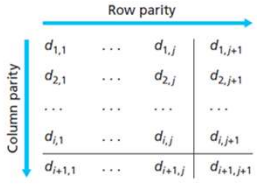


Figure 5.4 ♦ One-bit even parity



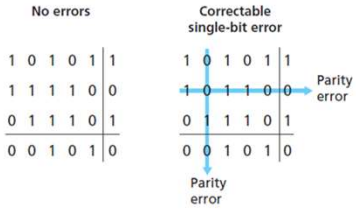


Figure 5.5 ♦ Two-dimensional even parity

Dr. Gaurav Varshney IIT Jammu

Error Detection and Correction

- Another method is Checksum the 16 bit Internet Checksum that we calculated for TCP and UDP.
- CRC [Cyclic Redundancy Check]
  - Consider the d-bit piece of data, D, that the sending node wants to send to the receiving node.
  - The sender and receiver must first agree on an r + 1 bit pattern, known as a generator, which we will denote as G.
  - We will require that the most significant (leftmost) bit of G be a 1.
  - For a given piece of data, D, the sender will choose r additional bits, R, and append them to D
  - such that the resulting d + r bit pattern (interpreted as a binary number) is exactly divisible by G (i.e., has no remainder) using modulo-2 arithmetic.
  - All CRC calculations are done in modulo-2 arithmetic without carries in addition or borrows in subtraction [bitwise exclusive-or (XOR) operation]

Dr. Gaurav Varshney IIT Jammu

```
1011 XOR 0101 = 1110
1001 XOR 1101 = 0100
```

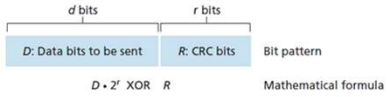


Figure 5.6 ♦ CRC

Error Detection and Correction

- CRC [Cyclic Redundancy Check]

$$D \cdot 2^r \text{ XOR } R = nG$$

That is, we want to choose R such that G divides into $D \cdot 2^r$ XOR R without remainder.

International standards have been defined for 8-, 12-, 16-, and 32-bit generators, G. The CRC-32 32-bit standard, which has been adopted in a number of link-level IEEE protocols, uses a generator of

$$G_{\text{CRC-32}} = 10000010011000001001110110110111$$

CRC can detect consecutive bit errors of r bits or fewer

Dr. Gaurav Varshney IIT Jammu

1011 XOR 0101 = 1110
1001 XOR 1101 = 0100

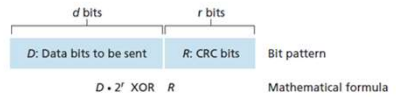
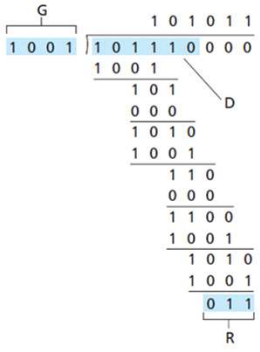


Figure 5.6 ♦ CRC



Dr. Gaurav Varshney IIT Jammu Figure 5.7 ♦ A sample CRC calculation

Multiple Access Links and Protocols

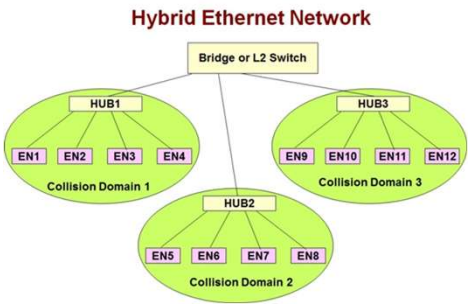
- Two types of network links
  - Point to point links
  - Broadcast links
- A **point-to-point link** consists of a single sender at one end of the link and a single receiver at the other end of the link
- The second type of link, a **broadcast link**, can have multiple sending and receiving nodes all connected to the same, single, shared broadcast channel.
- The term *broadcast* is used here because when any one node transmits a frame, the channel broadcasts the frame and each of the other nodes receives a copy. Ethernet and wireless LANs are examples of broadcast link-layer technologies.
- How to coordinate the access of multiple sending and receiving nodes to a shared broadcast channel—the **multiple access problem**

Dr. Gaurav Varshney IIT Jammu

Multiple Access Links and Protocols

- Computer networks similarly have protocols—so-called **multiple access protocols**—by which nodes regulate their transmission into the shared broadcast channel.
- In order to ensure that the broadcast channel performs useful work when multiple nodes are active, it is necessary to somehow coordinate the transmissions of the active nodes.
- Three categories of Multiple Access Protocols
  - **Channel partitioning protocols**
  - **Random access protocols**
  - **Taking-turns protocols**.

https://computernetworkingsimplified.wordpress.com/tag/collision-domain/



Dr. Gaurav Varshney IIT Jammu

Multiple Access Protocol: Desirable Characteristics

- A multiple access protocol for a broadcast channel of rate, R bits per second should have the following desirable characteristics:

1. When only one node has data to send, that node has a throughput of R bps.
2. When M nodes have data to send, each of these nodes has a throughput of R/M bps. This need not necessarily imply that each of the M nodes always has an instantaneous rate of R/M, but rather that each node should have an average transmission rate of R/M over some suitably defined interval of time.
3. The protocol is decentralized; that is, there is no master node that represents a single point of failure for the network.
4. The protocol is simple, so that it is inexpensive to implement.

Multiple Access Protocol: Channel Partitioning

- Time-division multiplexing (TDM) and frequency-division multiplexing (FDM) are two techniques that can be used to partition a broadcast channel bandwidth among all nodes sharing that channel.
- As an example, suppose the channel supports N nodes and that the transmission rate of the channel is R bps. TDM divides time into **time frames** and further divides each time frame into N **time slots.**
- Whenever a node has a packet to send, it transmits the packet's bits during its assigned time slot in the revolving TDM frame. Typically, slot sizes are chosen so that a single packet can be transmitted during a slot time

Multiple Access Protocol: Channel Partitioning

- TDM is appealing because it eliminates collisions and is perfectly fair: Each node gets a dedicated transmission rate of R/N bps during each frame time.
- However, it has two major drawbacks.
  - First, a node is limited to an average rate of R/N bps even when it is the only node with packets to send.
  - A second drawback is that a node must always wait for its turn in the transmission sequence—again, even when it is the only node with a frame to send.
- While TDM shares the broadcast channel in time, FDM divides the R bps channel into different frequencies (each with a bandwidth of R/N) and assigns each frequency to one of the N nodes.

Dr. Gaurav Varshney IIT Jammu

Multiple Access Protocol: Channel Partitioning

- FDM thus creates N smaller channels of R/N bps out of the single, larger R bps channel. FDM shares both the advantages and drawbacks of TDM. It avoids collisions and divides the bandwidth fairly among the N nodes.
- However, FDM also shares a principal disadvantage with TDM—a node is limited to a bandwidth of R/N, even when it is the only node with packets to send.

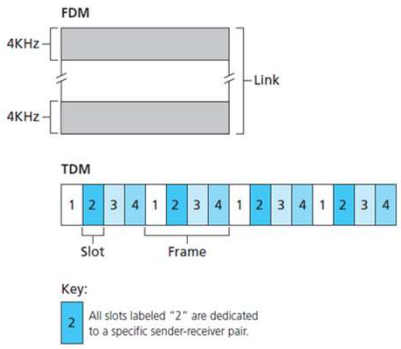


Figure 5.9 ♦ A four-node TDM and FDM example

Dr. Gaurav Varshney IIT Jammu

Dr. Gaurav Varshney IIT Jammu

Random Access Protocols: Slotted Aloha

- In a random access protocol, a transmitting node always transmits at the full rate of the channel, namely, R bps.
- When there is a collision, each node involved in the collision repeatedly retransmits its frame.
- But when a node experiences a collision, it doesn't necessarily retransmit the frame right away. *Instead it waits a random delay before retransmitting the frame.*
- **Slotted ALOHA**
  - All frames consist of exactly L bits.
  - Time is divided into slots of size L/R seconds (that is, a slot equals the time to transmit one frame).
  - Nodes start to transmit frames only at the beginnings of slots.
  - The nodes are synchronized so that each node knows when the slots begin.
  - If two or more frames collide in a slot, then all the nodes detect the collision event before the slot ends.

https://www.seas.upenn.edu/~kassam/tcom370/n99_12.pdf

Random Access Protocols

Let p be a probability, that is, a number between 0 and 1. The operation of slotted ALOHA in each node is simple:

- When the node has a fresh frame to send, it waits until the beginning of the next slot and transmits the entire frame in the slot.
- If there isn't a collision, the node has successfully transmitted its frame and thus need not consider retransmitting the frame. (The node can prepare a new frame for transmission, if it has one.)
- If there is a collision, the node detects the collision before the end of the slot. The node retransmits its frame in each subsequent slot with probability p until the frame is transmitted without a collision.

- When there are multiple active nodes, a certain fraction of the slots will have collisions and will therefore be "wasted."
- The second concern is that another fraction of the slots will be *empty* because all active nodes refrain from transmitting as a result of the probabilistic transmission policy.
- The only "unwasted" slots will be those in which exactly one node transmits.

Random Access Protocols: Slotted Aloha

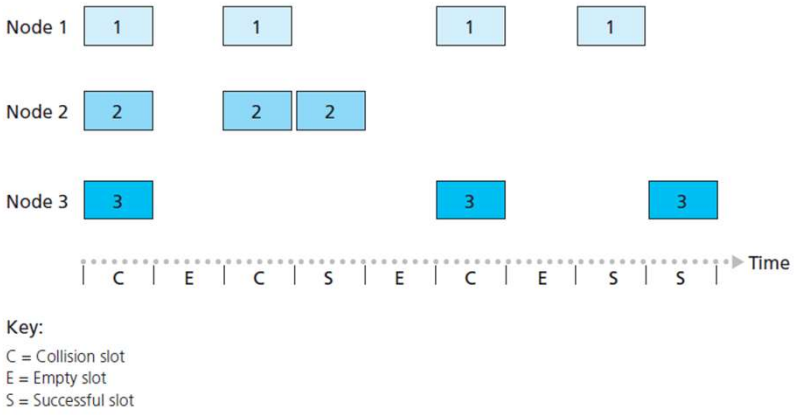


Figure 5.10 ♦ Nodes 1, 2, and 3 collide in the first slot. Node 2 finally succeeds in the fourth slot, node 1 in the eighth slot, and node 3 in the ninth slot

Random Access Protocols: Slotted Aloha

- We assume that each node always has a frame to send and that the node transmits with probability p for a fresh frame as well as for a frame that has already suffered a collision.
- Suppose there are N nodes. Then the probability that a given slot is a successful slot is the probability that one of the nodes transmits and that the remaining $N - 1$ nodes do not transmit.
- The probability that a given node transmits is p; the probability that the remaining nodes do not transmit is $(1 - p)^{N-1}$.
- Therefore the probability a given node has a success is $p(1 - p)^{N-1}$.
- Because there are N nodes, the probability that any one of the N nodes has a success is $Np(1 - p)^{N-1}$. The max efficiency is 0.37.

Pure Aloha and probability of success in slotted Aloha? Home Work

Random Access Protocol: Pure Aloha

The slotted ALOHA protocol required that all nodes synchronize their transmissions to start at the beginning of a slot. The first ALOHA protocol [1970] was actually an unslotted, fully decentralized protocol

the probability that a given node has a successful transmission is $p(1-p)^{2(N-1)}$.

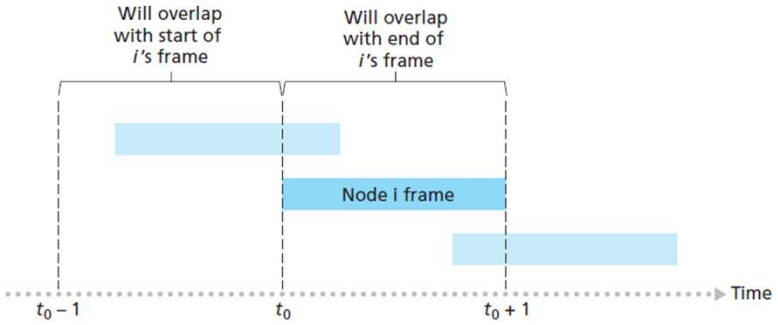


Figure 5.11 ♦ Interfering transmissions in pure ALOHA

Random Access : Carrier Sense Multiple Access

- In both slotted and pure ALOHA, a node's decision to transmit is made independently of the activity of the other nodes attached to the broadcast channel.
- Two important things that are required in a Random Access Protocol
  - Sense the channel before sending the data on the wire
  - If some other node begin the use of the broadcast medium stop sending
- These two rules are embodied in the family of carrier sense multiple access (CSMA) and CSMA with collision detection (CSMA/CD) protocols.
- If every node sense the channel why collision will ever happen at the very first place ?

Random Access : Carrier Sense Multiple Access

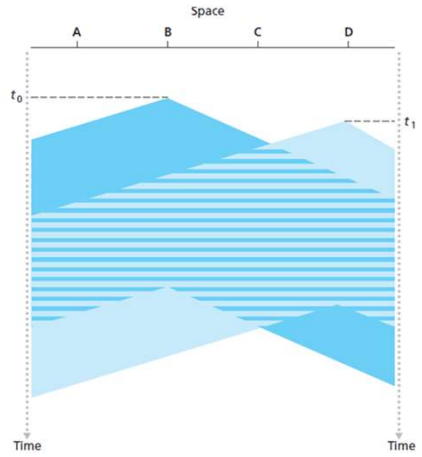


Figure 5.12 ♦ Space-time diagram of two CSMA nodes with colliding transmissions

it is evident that the end-to-end **channel propagation delay** of a broadcast channel—the time it takes for a signal to propagate from one of the nodes to another—will play a crucial role in determining its performance.

The longer this propagation delay, the larger the chance that a carrier-sensing node is not yet able to sense a transmission that has already begun at another node in the network

Dr. Gaurav Varshney IIT Jammu

Random Access : CSMA with Collision Detection (CD)

In the previous figure nodes do not perform collision detection; both B and D continue to transmit their frames in their entirety even though a collision has occurred.

When a node performs collision detection, it ceases transmission as soon as it detects a collision.

Before analyzing the CSMA/CD protocol, let us now summarize its operation from the perspective of an adapter (in a node) attached to a broadcast channel:

1. The adapter obtains a datagram from the network layer, prepares a link-layer frame, and puts the frame adapter buffer.
2. If the adapter senses that the channel is idle (that is, there is no signal energy entering the adapter from the channel), it starts to transmit the frame. If, on the other hand, the adapter senses that the channel is busy, it waits until it senses no signal energy and then starts to transmit the frame.
3. While transmitting, the adapter monitors for the presence of signal energy coming from other adapters using the broadcast channel.
4. If the adapter transmits the entire frame without detecting signal energy from other adapters, the adapter is finished with the frame. If, on the other hand, the adapter detects signal energy from other adapters while transmitting, it aborts the transmission (that is, it stops transmitting its frame).
5. After aborting, the adapter waits a random amount of time and then returns to step 2.

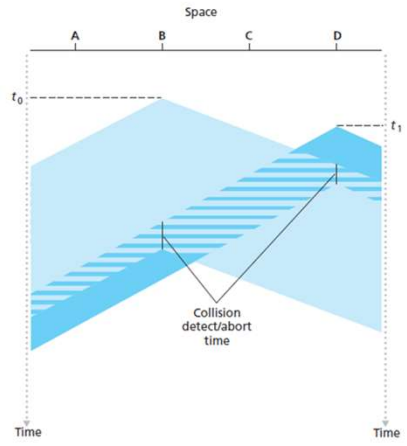


Figure 5.13 ♦ CSMA with collision detection

Dr. Gaurav Varshney IIT Jammu

Random Access : CSMA/CD

The need to wait a random (rather than fixed) amount of time is hopefully clear—if two nodes transmitted frames at the same time and then both waited the same fixed amount of time, they'd continue colliding forever.

What we'd like is an interval that is short when the number of colliding nodes is small, and long when the number of colliding nodes is large.

The **binary exponential backoff** algorithm, used in Ethernet as well as in DOCSIS cable network multiple access protocols [DOCSIS 2011], elegantly solves this problem.

Specifically, when transmitting a frame that has already experienced n collisions, a node chooses the value of K at random from $\{0, 1, 2, \ldots . 2^n -1)\}$.

Thus, the more collisions experienced by a frame, the larger the interval from which K is chosen.

For Ethernet, the actual amount of time a node waits is K*512 bit times (i.e., K times the amount of time needed to send 512 bits into the Ethernet) and the maximum value that n can take is capped at 10.

Random Access : CSMA/CD

The node then chooses $K = 0$ with probability 0.5 or chooses $K = 1$ with probability 0.5. If the node chooses $K = 0$, then it immediately begins sensing the channel.

If the node chooses $K = 1$, it waits 512 bit times (e.g., 0.01 microseconds for a 100 Mbps Ethernet) before beginning the sense-and-transmit-when-idle cycle.

After a second collision, K is chosen with equal probability from $\{0,1,2,3\}$. After three collisions, K is chosen with equal probability from $\{0,1,2,3,4,5,6,7\}$.

After 10 or more collisions, K is chosen with equal probability from $\{0,1,2, \ldots , 1023\}$. Thus, the size of the sets from which K is chosen grows exponentially with the number of collisions; for this reason this algorithm is referred to as **binary exponential backoff.**

Ethernet is an example.