TWO ARMY PROBLEM

Let's imagine two armies, led by two generals, planning an attack on a common enemy. The enemy's city is in a valley and has a strong defence that can easily fight off a single army. The two generals have to communicate with each other to plan a synchronised attack as this is their only chance to win. The only problem is that to communicate with each other they have to send a messenger across the enemy's territory. If a messenger is captured the message he's carrying is lost. Also, each general wants to know that the other general knows when to attack. Otherwise, a general wouldn't be sure if he's attacking alone and as we know attacking alone is rather pointless.

Now, let's go through a simple scenario. Let's call our generals A and B and let's assume everything goes perfectly fine. General A, who is the leader, sends a message – "Attack tomorrow at dawn". General B receives a message and sends back an acknowledgement – "I confirm, attack tomorrow at dawn". A receives B's confirmation. Is this enough to form a consensus between the generals? Unfortunately not, as General B still doesn't know if his confirmation was received by General A. Ok, so what if General A confirms General's B confirmation? Then, of course, that confirmation has to be also confirmed and we end up with an infinite exchange of confirmations.

In the second scenario, let's also assume that General A sends a message to General B. Some time has passed and General A starts wondering what happened to his message as there is no confirmation coming back from General B. There are two possibilities here. Either the messenger sent by General A has been captured and hasn't delivered a message or maybe B's messenger carrying B's confirmation has been captured. In both scenarios, they cannot come to a consensus again as A is not able to tell if his message was lost or if it was B's confirmation that didn't get through. Again, we ended up in an inconsistent state which would result in either General A or B attacking by himself.

We can quickly realise that no matter how many different scenarios we try and how many messages we send we cannot guarantee that consensus is reached and each general is certain that his ally will attack at the same time. To make it even worse, there is no solution to the Two Generals' Problem, so the problem remains unsolvable.

Hidden Terminal Problem:

The hidden terminal problem occurs when a terminal is visible from a wireless access point (APs), but not from other nodes communicating with that AP. This situation leads the difficulties in medium access control sublayer over wireless networking.

hidden terminals are nodes in a wireless network that are out of range of other node or a collection of nodes. Consider a wireless networking, each node at the far edge of the access point's range, which is known as A, can see the access point, but it is unlikely that the same node can see a node on the opposite end of the access point's range, C. These nodes are known as hidden. The problem is when nodes A and C start to send packets simultaneously to the access point B. Because the nodes A and C are out of range of each other and so cannot detect a collision while transmitting, Carrier sense multiple access with collision detection (CSMA/CD) does not work, and collisions occur, which then corrupt the data received by the access point. To overcome the hidden node problem, RTS/CTS handshaking (IEEE 802.11 RTS/CTS) is implemented in conjunction with the Carrier sense multiple accesses with collision avoidance (CSMA/CA) scheme. The same problem exists in a MANET.

Consider the scenario of wireless networking with three wireless devices (e.g. mobile phones) as shown below.

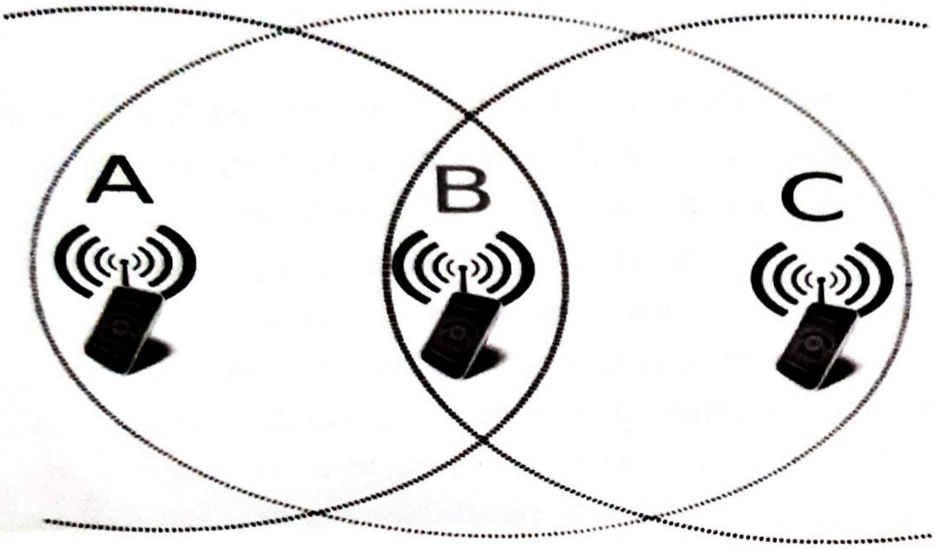
The transmission range of access point A reaches at B, but not at access point C, similarly transmission range of access point C reaches B, but not at A. These nodes are known as hidden terminals. The problem occurs when nodes A and C start to send data packets simultaneously to the access point B. Because the access points A and C are out of range of each other and resultant they cannot detect a collision while transmitting, Carrier sense multiple access with collision detection (CSMA/CD) does not work, and collisions occur, which then corrupt the data received by the access point B due to the hidden terminal problem.

The hidden terminal analogy is described as follows:

- Terminal A sends data to B, terminal C cannot hear A
- Terminal C wants to send data to B, terminal C senses a "free" medium (CS fails) and starts transmitting
- Collision at B occurs, A cannot detect this collision (CD fails) and continues with its transmission to B

- Terminal A is "hidden" from C and vice versa.

IEEE 802.11 uses 802.11 RTS/CTS acknowledgment and handshake techniques over wireless networks to transferring packets that partly overcome the hidden node problem. RTS/CTS is not a proper and permanent solution and may decrease throughput even further, but adaptive acknowledgments from the base station can help too.



to solve hidden node problem

Increase Transmitting Power from the Nodes: With the enhancement of the transmission power of access point can solve the hidden terminal problem by allowing the cell around each node to increase in size, encompassing all of the other nodes.

Use Omni directional antennas: Since nodes using directional antennas are nearly invisible to nodes that are not positioned in the direction the antenna is aimed at, directional antennas should be used only for very small networks.

Remove obstacles: Keep away the obstacles that affect the performance of access point accessibility.

Move the node: Provide the mobility features to the nodes.

Use protocol enhancement software: Pooling and token passing strategy should be used before start data transformation.

Exposed Terminal Problem:

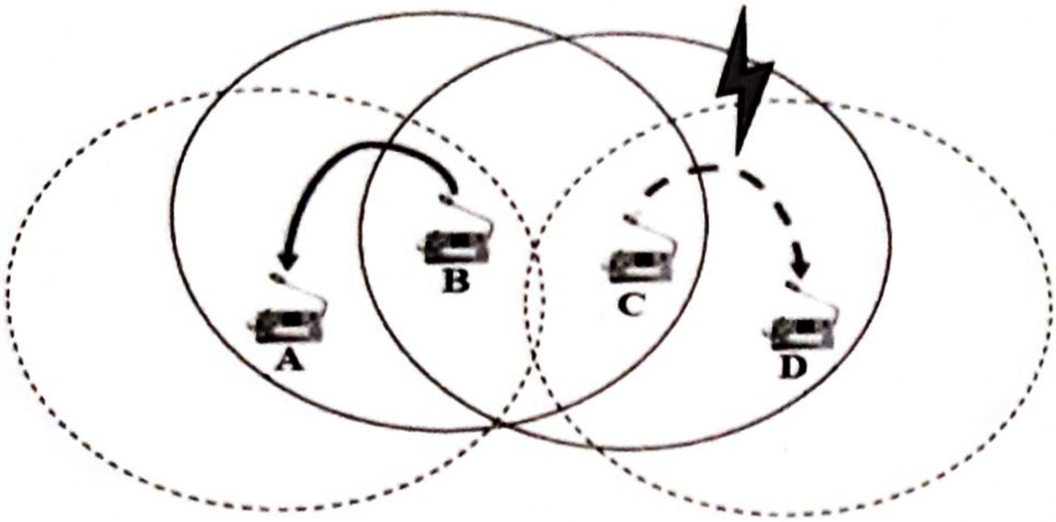
In wireless networks, when a node is prevented from sending packets to other nodes because of a neighboring transmitter is known as the exposed node problem.

Consider the below wireless network having four nodes labeled A, B, C, and D, where the two receivers are out of range of each other, yet the two transmitters (B, C) in the middle are in range of each other. Here, if a transmission between A and B is taking place, node C is prevented from transmitting to D as it concludes after carrier sense that it will interfere with the transmission by its neighbor node B. However note that node D could still receive the transmission of C without interference because it is out of range from B. Therefore, implementing directional antenna at a physical layer in each node could reduce the probability of signal interference, because the signal is propagated in a narrow band.

The exposed terminal analogy is described as follows:

- B sends to A, C wants to send to another terminal D not A or B
- C senses the carrier and detects that the carrier is busy.
- C postpones its transmission until it detects the medium as being idle again
- But A is outside radio range of C, waiting is not necessary
- C is "exposed" to B

Hidden terminals cause collisions, where as Exposed terminals causes' unnecessary delay.


A
B
C
D

Definition of Internal Fragmentation

Internal fragmentation occurs when the memory is divided into **fixed sized blocks**. Whenever a process request for the memory, the fixed sized block is allocated to the process. In case the memory assigned to the process is somewhat larger than the memory requested, then the difference between assigned and requested memory is the **Internal fragmentation**.

This leftover space inside the fixed sized block can not be allocated to any process as it would not be sufficient to satisfy the request of memory by the process. Let us understand Internal fragmentation with the help of an example. The memory space is partitioned into the fixed-sized blocks of 18,464 bytes. Let us say a process request for 18,460 bytes and partitioned fixed-sized block of 18,464 bytes is allocated to the process. The result is 4 bytes of 18,464 bytes remained empty which is the internal fragmentation.

The overhead of keeping track of the internals hole created due to internal fragmentation is substantially more than the number of internal holes. The problem of internal fragmentation can be solved by **partitioning the memory into the variable sized block** and assign the best-sized block to a process requesting for the memory. Still, it will not totally eliminate the problem of internal fragmentation but will reduce it to some extent.

Definition of External Fragmentation

External fragmentation occurs when there is a sufficient amount of space in the memory to satisfy the memory request of a process. But the process's memory request can not be satisfied as the memory available is in a non-contiguous manner. Either you apply first-fit or best-fit memory allocation strategy it will cause external fragmentation.

When a process is loaded and removed from the memory the free space creates the hole in the memory space, and there are many such holes in the memory space, this is called External fragmentation. Although the first fit and best fit can affect the amount of external fragmentation, it can not be totally eliminated. **Compaction** may be the solution for external fragmentation.

Key Differences Between Internal and External fragmentation

1. The basic reason behind the occurrences of internal and external fragmentation is that internal fragmentation occurs when memory is partitioned into **fixed-sized blocks** whereas external fragmentation occurs when memory is partitioned into **variable size blocks**.
2. When the memory block allotted to the process comes out to be slightly larger than requested memory, then the free space left in the allotted memory block causes internal fragmentation. On the other hands, when the process is removed from the memory it creates free space causing a hole in the memory which is called external fragmentation.
3. The problem of internal fragmentation can be solved by partitioning the memory into variable sized blocks and assign the best fit block to the requesting process. However, the solution for external fragmentation is compaction, but it is expensive to implement, so the processes must be allowed to acquire physical memory in a non-contiguous manner, to achieve this the technique of paging and segmentation is introduced.