

## 1) DIFFERENCES BETWEEN 802.11A & 802.11B?

List of Ratified Wireless Standards			
Standards	802.11a	802.11b	802.11g
Release Date	October 1999	October 1999	June 2003
Frequency	5 GHz	2.4 GHz	2.4 GHz
Throughput (Typical)	23 Mbps	4.3 Mbps	19 Mbps
Max. Data Rate	54 Mbps	11 Mbps	54 Mbps
Modulation Technique	OFDM	DSSS	OFDM
Range (Indoor)*	~ 35 meters	~ 38 meters	~ 38 meters
Range (Outdoor)**	~ 120 meters	~ 140 meters	~ 140 meters
* Range depends on number of walls and the type.			
** Range depends on total loss (include any obstacles)			

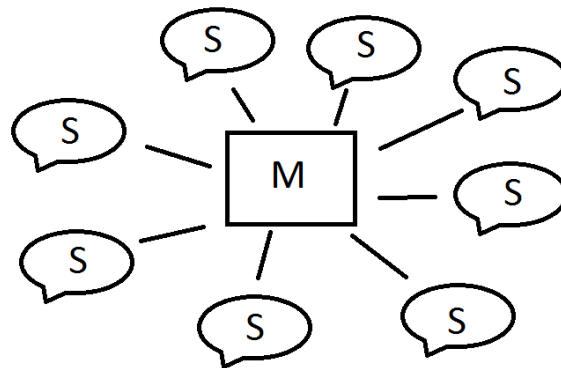
Comparison of 802.11a/b/g Wireless Standards			
	802.11a	802.11b	802.11g
Speed	Fast speed	Slow speed	Fast speed
Compatibility with other 802.11 standards	Not compatible with 802.11b/g standards	Compatible with 802.11g standard	Compatible with 802.11b standard
Vulnerable to interference	No	No	No
Distance coverage	Short distance	Long distance	Long distance
Signal strength due to penetrating obstacles	Poor	Lower	Lower
Suitable application	Nearby building-to-building connection	(rarely implemented today)	Hotspot area, office, hospital, etc

## 2) EXPLAIN THE ARCHITECTURE & LAYERS OF BLUETOOTH?

### I. BLUETOOTH ARCHITECTURE:

Bluetooth architecture defines two types of networks: **Piconet & Scatternets**.

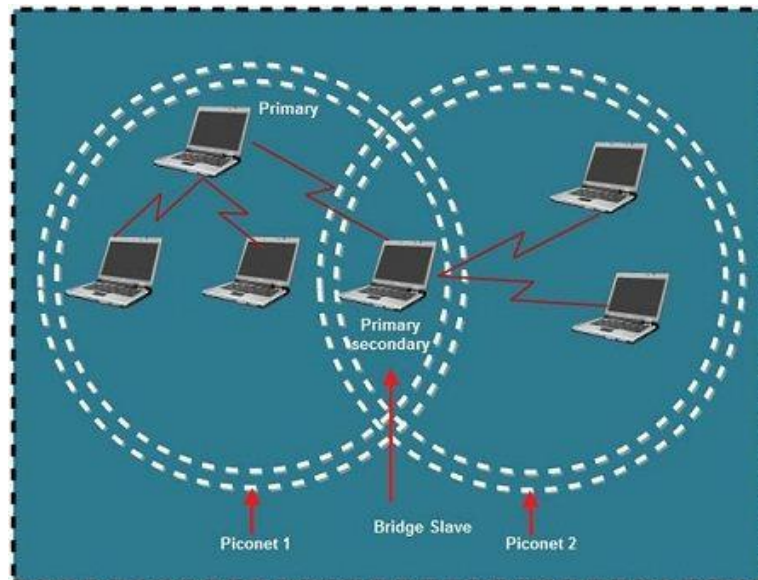
### Piconets:



- The first type of Bluetooth network is called as a **piconet** or a **small net**. It can have at the most eight stations.
- One of them is called as a **master** and all others are called as **Slaves**. A master can also be called as a primary station and slaves are the secondary station.
- There can be only one master station in each piconet.
- All communication is between master and a slave. Slave-slave communication is not possible. The communication between the master and the slave stations can be one-to-one or one-to-many.
- The formation of a piconet is governed by two factors:
  1. Address of each Bluetooth device.
  2. Clock associated with each device.
- A piconet can have up to **seven active** slaves at any given instant of time. In order to identify a slave, each one is assigned a locally unique active member address (AM-ADDR).

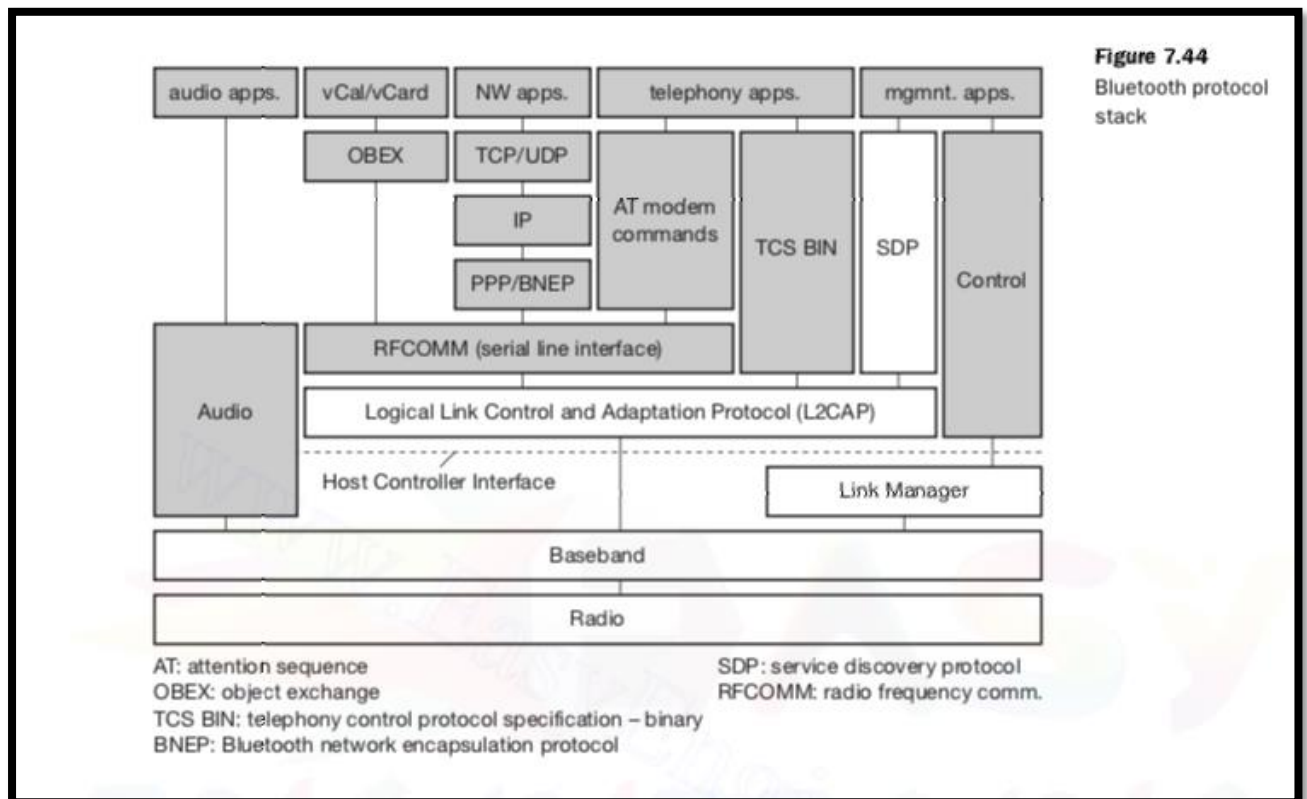
- If a Bluetooth device is not associated with any piconet, then it is said to be in **standby mode**.

### Scatternets:



- Many piconets may exist simultaneously in a given area and they may even overlap each other.
- A scatternet is obtained by combining piconets as shown in figure.
- A slave in one piconet can act as a master or primary in other piconet.
- Such a station or node can receive messages from the master in the first piconet and deliver the message to its slaves in other piconet where it is acting as master. This node is also called bridge slave.
- Thus a station can be a member of two piconets.
- A station cannot be a master in two piconets.

## II. BLUETOOTH PROTOCOLS:



### 1. Radio Layer:

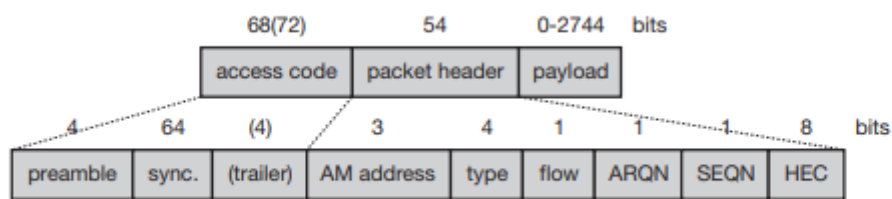
- The Bluetooth radio layer corresponds to the physical layer of OSI model.
- It deals with radio transmission and modulation.
- The radio layer moves data from master to slave or vice versa.
- It is a low power system that uses 2.4 GHz ISM band in a range of 10 meters.

- This band is divided into 79 channels of 1MHz each. Bluetooth uses the Frequency Hopping Spread Spectrum (FHSS) method in the physical layer to avoid interference from other devices or networks.
- Bluetooth hops 1600 times per second, *i.e.* each device changes its modulation frequency 1600 times per second.
- In order to change bits into a signal, it uses a version of FSK called GFSK *i.e.* FSK with Gaussian bandwidth filtering.

## 2. Baseband Layer:

- Baseband layer is equivalent to the MAC sublayer in LANs.
- Bluetooth uses a form of TDMA called TDD-TDMA (time division duplex TDMA).
- Master and slave stations communicate with each other using time slots.
- The master in each piconet defines the time slot of 625  $\mu$ sec.
- In TDD- TDMA, communication is half duplex in which receiver can send and receive data but not at the same time.
- If the piconet has only no slave; the master uses even numbered slots (0, 2, 4, ...) and the slave uses odd-numbered slots (1, 3, 5, .... ). Both master and slave communicate in half duplex mode. In slot 0, master sends & secondary receives; in slot 1, secondary sends and primary receives.
- If piconet has more than one slave, the master uses even numbered slots. The slave sends in the next odd-numbered slot if the packet in the previous slot was addressed to it.

- Figure shows the components of a Bluetooth packet at baseband layer. The packet typically consists of following three fields:



**Figure 7.46**  
Baseband packet  
format

**Access code:** This first field of a packet is needed for timing synchronization and piconet identification. It may represent special codes during paging and inquiry.

**Packet header:** This field contains typical layer 2 features: address, packet type, flow and error control, and checksum.

**Payload:** Up to 343 bytes payload can be transferred. The structure of the payload field depends on the type of link.

- Bluetooth offers two different types of links:

### 1. **Asynchronous Connection-less (ACL):**

- It is used for packet switched data that is available at irregular intervals.
- ACL delivers traffic on a best effort basis. Frames can be lost & may have to be re-transmitted.
- A slave can have only one ACL link to its master.
- Thus ACL link is used where correct delivery is preferred over fast delivery.

- The ACL can achieve a maximum data rate of 721 kbps by using one, three or more slots.

## 2. Synchronous Connection Oriented (SCO):

- sco is used for real time data such as sound. It is used where fast delivery is preferred over accurate delivery.
- In an sco link, a physical link is created between the master and slave by reserving specific slots at regular intervals.
- Damaged packet; are not re-transmitted over sco links.
- A slave can have three sco links with the master and can send data at 64 Kbps.

## 3. Link Manager Protocol (LMP):

- The Link Manager protocol manages various aspects of the radio link between master and slave.
- The following groups of functions are covered by the LMP:
  - (a) **Authentication, pairing, and encryption:** Although basic authentication is handled in the baseband, LMP has to control the exchange of random numbers and signed responses. The pairing service is needed to establish an initial trust relationship between two devices that have never communicated before. The result of pairing is a link key. This may be changed, accepted or rejected. LMP is not directly involved in the encryption process, but sets the encryption mode (no encryption, point-to-point, or broadcast), key size, and random speed.

- (b) **Synchronization:** Precise synchronization is of major importance within a Bluetooth network. The clock offset is updated each time a packet is received from the master. Additionally, special synchronization packets can be received. Devices can also exchange timing information related to the time differences (slot boundaries) between two adjacent piconets.
- (b) **Capability negotiation:** Not only the version of the LMP can be exchanged but also information about the supported features. Not all Bluetooth devices will support all features that are described in the standard, so devices have to agree the usage of, e.g., multi-slot packets, encryption, SCO links, voice encoding, park/sniff/hold, HV2/HV3 packets etc.
- (c) **Quality of service negotiation:** Different parameters control the QoS of a Bluetooth device at these lower layers. The poll interval, i.e., the maximum time between transmissions from a master to a particular slave, controls the latency and transfer capacity. Depending on the quality of the channel, DM or DH packets may be used (i.e., 2/3 FEC protection or no protection). The number of repetitions for broadcast packets can be controlled. A master can also limit the number of slots available for slaves' answers to increase its own bandwidth.
- (d) **Power control:** A Bluetooth device can measure the received signal strength. Depending on this signal level the device can direct the sender of the measured signal to increase or decrease its transmit power.
- (e) **Link supervision:** LMP has to control the activity of a link, it may set up new SCO links, or it may declare the failure of a link.
- (f) **State and transmission mode change:** Devices might switch the master/slave role, detach themselves from a connection, or change the operating mode.



#### 4. Logical Link Control & Adaptation Protocol Layer (L2CAP):

- The logical link control and adaptation protocol (L2CAP) is a data link control protocol on top of the baseband layer offering logical channels between Bluetooth devices with QoS properties. L2CAP is available for ACLs only.
- L2CAP provides three different types of logical channels that are transported via the ACL between master and slave:

(a) **Connectionless:** These unidirectional channels are typically used for broadcasts from a master to its slave(s).

(b) **Connection-oriented:** Each channel of this type is bi-directional and supports QoS flow specifications for each direction. These flow specs follow RFC 1363 (Partridge, 1992) and define average/peak data rate, maximum burst size, latency, and jitter.

(c) **Signaling:** This third type of logical channel is used to exchanging signaling messages between L2CAP entities.

- Each channel can be identified by its **channel identifier (CID)**.
- Signaling channels always use a CID value of 1, a CID value of 2 is reserved for connectionless channels. For connection-oriented channels a unique CID ( $\geq 64$ ) is dynamically assigned at each end of the channel to identify the connection (CIDs 3 to 63 are reserved).

- The various function of L2CAP is:

### **1. Segmentation and reassembly:**

- L2CAP receives the packets of up to 64 KB from upper layers and divides them into frames for transmission.
- It adds extra [information](#) to define the location of frame in the original packet.
- The L2CAP reassembles the frame into packets again at the destination.

### **2. Multiplexing:**

- L2CAP performs multiplexing at sender side and de-multiplexing at receiver side.
- At the sender site, it accepts data from one of the upper layer protocols frames them and deliver them to the Base-band layer.
- At the receiver site, it accepts a frame from the base-band layer, extracts the data, and delivers them to the appropriate protocol layer.

### **3. Quality of Service (QOS):**

- L2CAP handles quality of service requirements, both when links are established and during normal operation.
- It also enables the devices to negotiate the maximum payload size during connection establishment.

## 5. Service Discovery Protocol Layer (SDP):

- The service discovery protocol helps the applications to discover which services are available and to determine the characteristics of those available services.
- SDP defines only the discovery of services, not about their usage.
- New service is discovered as follows:

(a) Client sends a request to search for an interested service.

(b) Then the server responds to the client with the list of available services that match to the client's criteria.

(c) The client uses the list to retrieve additional service attribute for the service of interest.

## 6. Profiles:

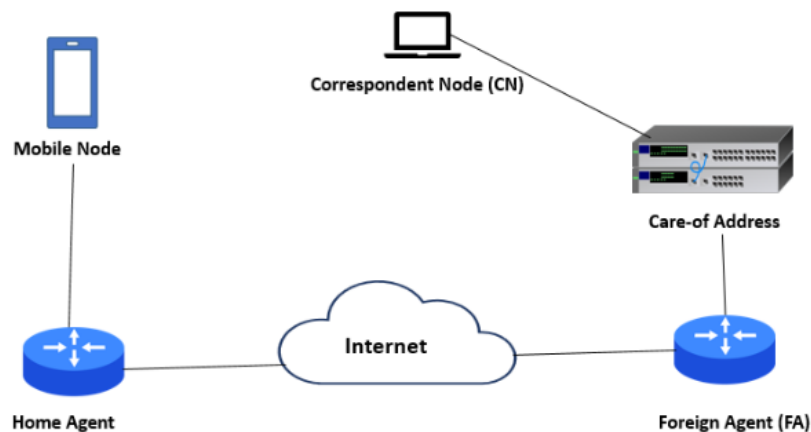
- Profiles are specifications which describe how Bluetooth should be used in a specific application and thus ensures that all devices from different manufacturers can seamlessly work with one another.
- There are about a dozen profiles: Generic Access, Serial Port, Dial up Networking, FAX, Headset, LAN, Access Point, Generic Object Exchange (OBEX), File Transfer, Object Push, Synchronization, Cordless Telephony, and Intercom.
- The profile concept is used to decrease the risk of interoperability problems between different manufacturers' products.

## 3) WHAT IS MOBILE IP & DHCP?

## I. MOBILE IP:

**Mobile IP** is a communication protocol (created by extending Internet Protocol, IP) that allows the users to move from one network to another with the same IP address. It ensures that the communication will continue without the user's sessions or connections being dropped.

Components of Mobile IP: Here are some important components that are included in the following mobile IP:



**1. Mobile Node:** Your devices like a laptop, cell phone, i-pad, etc.

**2. Home Agent:** It is the router to which your mobile device is connected located on your home network

**3. Foreign Agent (FA):**

- Router to which your mobile device is connected.
- When you are away from your home network, your device uses this agent to send/receive data to and from the HA (Home agent).

- Registers you as a foreigner in the foreign network similar to when you enter a new country, you are registered by a foreign country (as a newcomer) as well as your resident country (as a visitor to a foreign country with all details saved where you are going & why).

#### **4. Care-of Address:**

- When you take your mobile device outside out of its home network (MyHN) & enter a new network zone, your device is assigned a CoA, i.e. care-of address, which tells your home network your current location and that you belong to the family “MyHN”.
- It is usually the IP of the foreign agent.
- For example, when you want to go to a foreign country, you need to tell all your care-of details, i.e., details of your home country and everything, to the VISA office and embassy. After you reach a foreign country, they send all the details to your home country that you have arrived, and your ID is, for example, ID-001, so this ID is your care-of address.

#### **5. Correspondent Node (CN):**

- It is the node to which your device is corresponding (a web server).
- It communicates with your device.

**Working of Mobile IP:** It comprises 3 phases which are explained below in detail:

### **1. Agent Discovery:**

- Whenever you enter a network, your device needs to know in which network it is, home network or foreign network?
- For this, the home agent (HA) & foreign agent (FA) advertise their services so that your device can see those advertisements and discover that it is in which network zone.
- Your mobile node, i.e. your device determines its current point of attachment; if it fails to do so, it itself attempts to send messages to the agents (HA / FA) to send their advertisements.

### **2. Registration:**

- This happens when your device enters a foreign network.
- Your device registers itself by sending its care-of address to its home router/agent via the foreign agent or directly via a collocated care-of address (CCoA), which is a temporary IP address assigned to it.
- The response is then sent by the home router to your device via FA or directly to CCoA so that the data packets can be delivered to and from.
- The request is authenticated and has a registration time expiry before which the device has to register its current location.

- Great! You are added to their visitor lists.
- After your registration is successful & authenticated, the foreign and home agents add your device to their visitor lists.

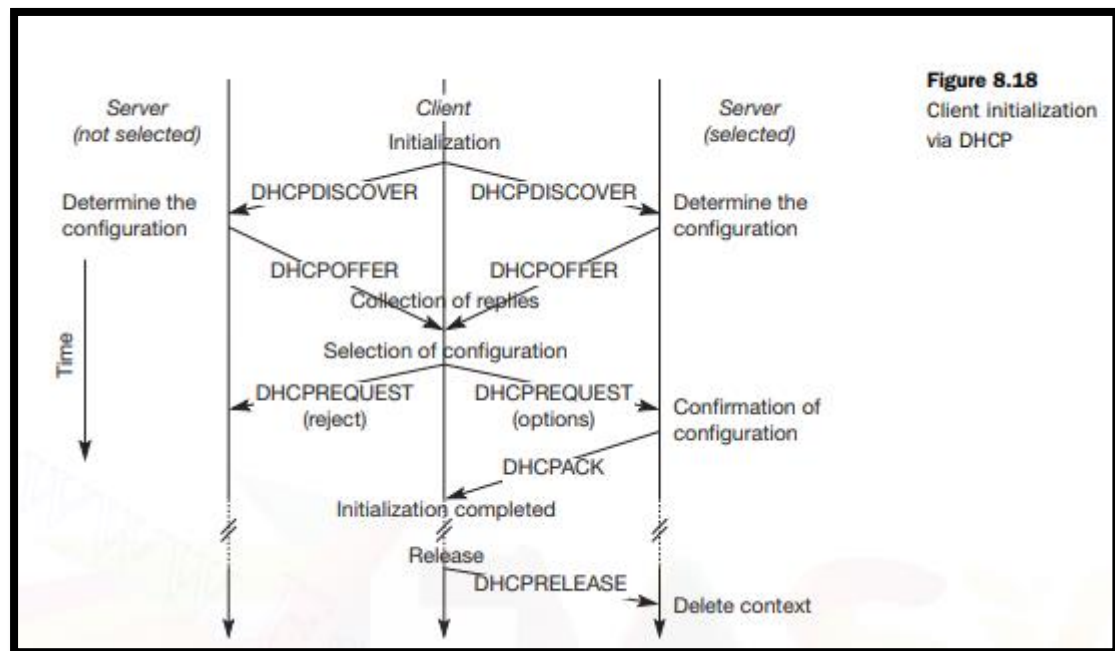
### **3. Tunneling:**

- A tunnel is a secure path from HA to FA that ensures the successful delivery of packets to your device.
- Your device sends packets using its home IP address but shows that it is always on its home network.
- The drop of packets is solved by reverse tunnelling by having the Foreign Agent send packets back to the Home Agent when it receives them from the device.

## **II. DHCP:**

- Stands for "Dynamic Host Configuration Protocol."
- The dynamic host configuration protocol is mainly used to simplify the installation and maintenance of networked computers. If a new computer is connected to a network, DHCP can provide it with all the necessary information for full system integration into the network.
- Providing an IP address, makes DHCP very attractive for mobile IP as a source of care-of-addresses.
- DHCP is based on a client/server model.

### Example:



- A typical initialization of a DHCP client is shown in Figure 8.18.
- The figure shows one client and two servers. As described above, the client broadcasts a DHCPDISCOVER into the subnet.
- There might be a relay to forward this broadcast. In the case shown, two servers receive this broadcast and determine the configuration they can offer to the client. One example for this could be the checking of available IP addresses and choosing one for the client.
- Servers reply to the client's request with DHCPOFFER and offer a list of configuration parameters.
- The client can now choose one of the configurations offered. The client in turn replies to the servers, accepting one of the configurations and rejecting the others using DHCPREQUEST. If a server receives a DHCPREQUEST with a rejection, it can free the reserved configuration for other possible clients.



- The server with the configuration accepted by the client now confirms the configuration with DHCPACK. This completes the initialization phase.
- If a client leaves a subnet, it should release the configuration received by the server using DHCPRELEASE.

#### 4) WRITE A SHORT NOTE ON DSR & AODV ROUTING ALGORITHMS?

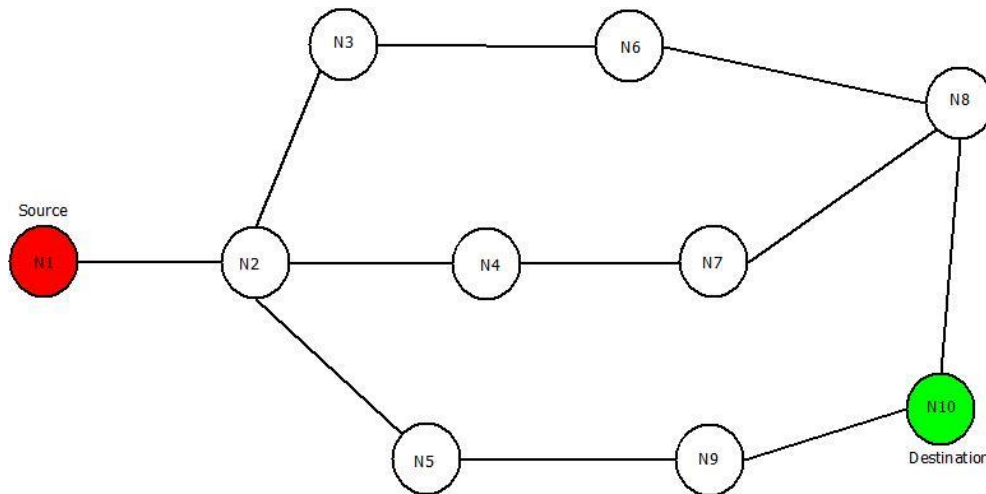
##### I. DYNAMIC SOURCE ROUTING (DSR):

- Dynamic Source Routing (DSR) comes under the reactive routing protocol category, as it is capable of discovering the route from source to destination only when required and needed.
- Dynamic Source Routing protocol uses a process called “Route Discovery Mechanism” that is capable of discovering the route for data packets from source node to destination nodes using intermediate nodes.
- The major change in DSR as compare to GSR and DSDV is, in DSDV after asking a requirement of route from source to destination, path via intermediate nodes is checked for its length. Then a “Re-Request” packet is sent back from destination to source via the smallest route possible in the whole network. The “Re-Request” packet contains its unique ID also.
- This process of separately sending a “Re-Request” packet from destination to source makes it easier for the sender to send the data packets on fixed path rather than sending it on multiple paths to check for total distance.

##### Dynamic Source Routing Protocol : Working

- **Consider a network containing 10 nodes where node N1 being the source and node N10 being the destination nodes. Below mentioned steps will let you know how**

**DSR protocol works and how Re-Request packet is transmitted through the network.**

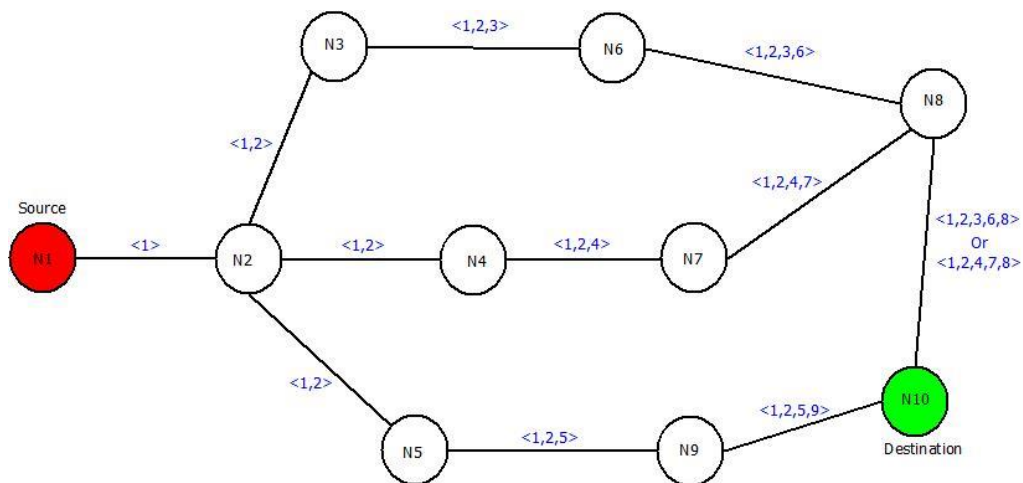


**Dynamic Source Routing : Network**

- **Step 1:** Start from source node N1 and broadcast the route information to its neighbors i.e. in this case the route information is "<1>", because of its one-to-one link between node N1 and N2.
- **Step 2:** Broadcast previous route information to neighbors of node N2 i.e. to node N3, N4, N5. The new route will remain same "<1,2>" in all the cases.
- **Step 3:** Take node N3 and broadcast previous route(<1,2>) to next neighboring nodes i.e. node N6. New route till node N6 will be "<1,2,3>" and same process can be done for other nodes i.e. Node N4 and N5.
- **Step 4:** Further, broadcast the new routes i.e. <1,2,3,6> , <1,2,4> , <1,2,5> to nodes N8, N7 & N9 respectively.

- **Step 5:** Repeat the above steps until destination node is reached via all the routes.

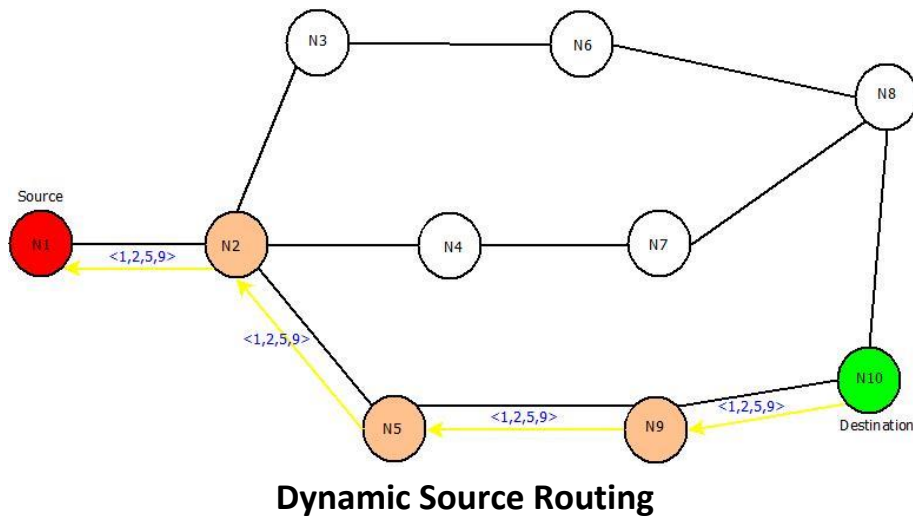
- The updated routes will be as:



**Dynamic Source Routing : Updated Network**

- After this, “Re-Request” packet will be sent in backward direction i.e. from destination node “N10” to source node “N1”. It will trace the shortest route by counting the number of nodes from route discovered in previous steps.
- The three possible routes are :
  - Route 1: <1,2,3,6,8>
  - Route 2: <1,2,4,7,8>
  - Route 3: <1,2,5,9>
- Route 3 i.e. "<1,2,5,9>" will be chosen as it contains the least number of nodes and hence it will definitely be the shortest path and then data can be transferred accordingly.

- The Re-Request Packet route can be located as:



## II. Ad-Hoc On Demand Distance Vector Routing Protocol (AODV):

- Another type of reactive routing protocol which does not maintain routes but build the routes as per requirements is Ad-Hoc On Demand Distance Vector Routing Protocol.
- AODV is used to overcome the drawbacks of Dynamic Source Routing Protocol and Distance Vector Routing Protocol i.e. Dynamic Source Routing is capable of maintaining information of the routes between source and destination which makes it slow. If the network is very large containing a number of routes from source to destination, it is difficult for the data packets header to hold whole information of the routes.

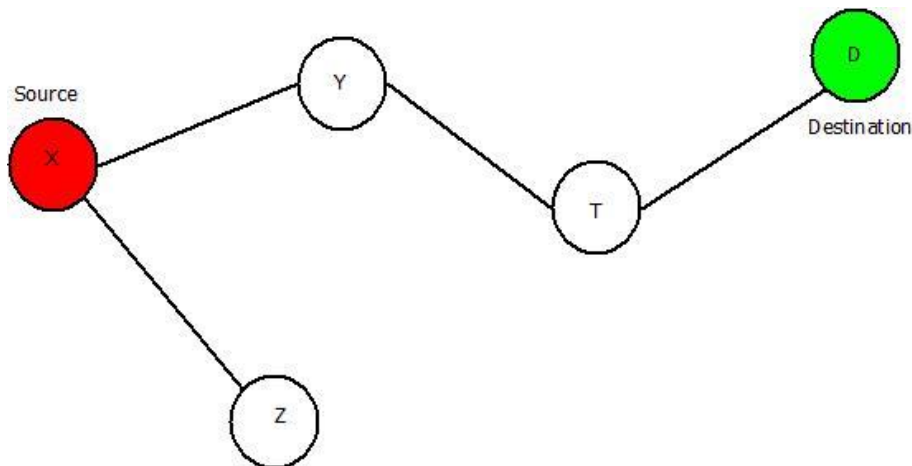
- In case of Dynamic Source Routing, multiple routes are present for sending a packet from source to destination but AODV overcomes this disadvantage too.
- In AODV, along with routing tables of every node, two counters including Sequence Number(SEQ NO) and broadcast ID are maintained also.
- The destination IP is already known to which data is to be transferred from source. Thus, the destination Sequence Number(SEQ NO) helps to determine an updated path from source to destination.
- Along with these counters, Route Request(RREQ) and Route Response(RRESP) packets are used in which RREQ is responsible for discovering of route from source to destination and RRESP sends back the route information response to its source.

### **Ad-Hoc On Demand Distance Vector Routing Protocol : Working**

- In Ad-Hoc On Demand Distance Vector Routing, the source node and destination nodes IP addresses are already known.
- The goal is to identify, discover and maintain the optimal route between source and destination node in order to send/receive data packets and informative.
- Each node comprises of a routing table along with below mentioned format of Route Request(RREQ) packet.

**RREQ { Destination IP, Destination Sequence Number, Source IP, Source Sequence Number, Hop Count }**

Consider a network containing 5 nodes that are “X”, “Y”, “Z”, “T”, “D” present at unit distance from each other, where “X” being the source node and “D” being the destination node.



**Ad-Hoc On Demand Distance Vector Routing : Sample Network**

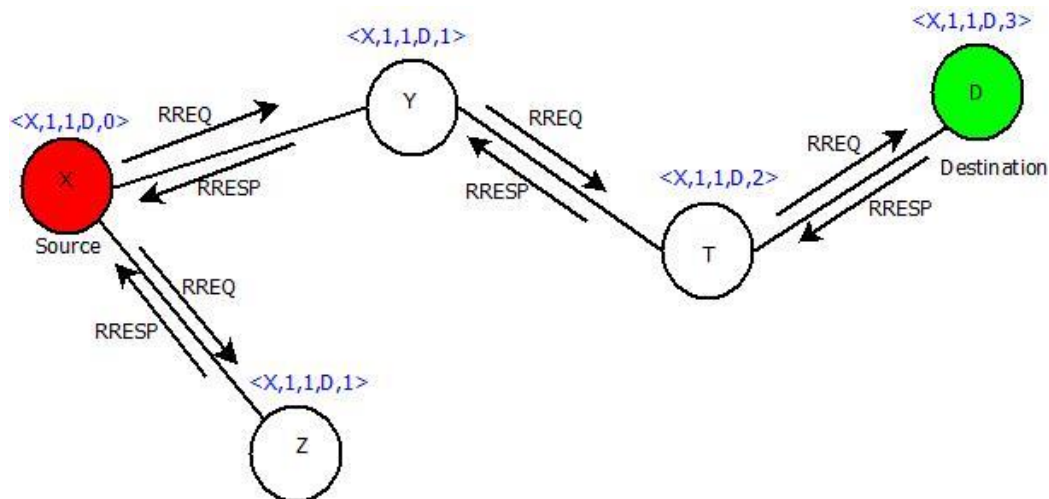
- The IP addresses of source node “X” and destination node “D” is already known. Below mentioned steps will let you know how AODV works and concept of Route Request(RREQ) and Route Response(RRESP) is used.

**Step 1:** Source node “X” will send Route Request i.e. RREQ packet to its neighbours “Y” and “Z”.

**Step 2:** Node “Y” & “Z” will check for route and will respond using RRESP packet back to source “X”. Here in this case “Z” is the last node but not the destination. It will send the RRESP packet to “X” stating “Route Not Found”. But node “Y” will send RRESP packet stating “Route Found” and it will further broadcast the RRESP to node “T”.

**Step 3:** Now the field of net hop in the RREQ format will be updated, Node “T” will send back the “Route Found” message to Node “Y” and will update the next hop field further.

**Step 4:** Then Node “T” will broadcast and RREQ packet to Node “D”, which is the destination and the next hop field is further updated. Then it will send RRES packet to “T” which will further be sent back to the source node “X” via node “Y” and Node “T” resulting in generation of an optimal path. The updated network would be:

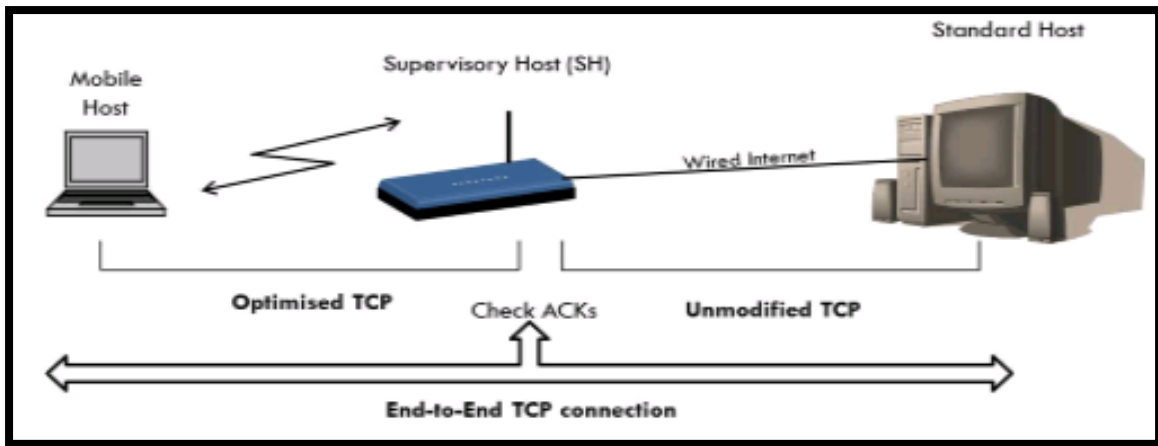


**Ad-Hoc On Demand Distance Vector Routing : Network**

## 5) WRITE ABOUT MOBILE TCP & TRANSACTION ORIENTED TCP?

### I. MOBILE TCP:

- M-TCP (mobile TCP) has the same goals as similar to its variants i.e. I-TCP and Snoop-TCP. It too wants to improve overall throughput, to lower the delay, to main end-to-end semantics of TCP.



- The M-TCP splits up the connection into two parts:
  - An unmodified TCP is used on the Standard host-Supervisory Host section
  - An optimised TCP is used on the Supervisory Host- Mobile Host section.
- The **Supervisory Host (SH)** adorns the same role as the proxy (Foreign Agent) in I-TCP.
- The SH is responsible for exchanging data to both the Standard host and the Mobile host.
- Here in this approach, we **assume** that the error bit rate is less as compared to other wireless links.
- So if any packet is lost, the retransmission has to occur from the original sender and not by the SH. (This also maintains the end-to-end TCP semantic)
- The SH monitors the ACKs (ACK means acknowledgement) being sent by the MH. If for a long period ACKs have not been received, then the SH assumes that the MH has been disconnected (maybe due to failure or moved out of range, etc...).



- If so the SH **chokes** the sender by setting its window size to 0.
- Because of this the sender goes into persistent mode i.e. the sender's state will not change no matter how long the receiver is disconnected.
- This means that the sender will not try to retransmit the data.
- Now when the SH detects a connectivity established again with the MH (the old SH or new SH if handover), the window of the sender is restored to original value.

### **Advantages:**

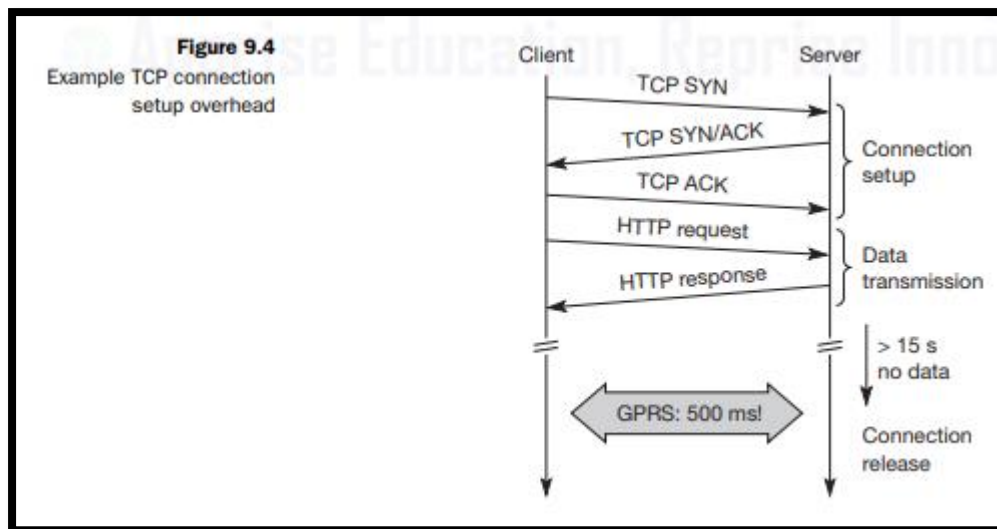
- Maintains the TCP end-to-end semantics. (No failed packet retransmission is done by the SH .All job handled by original sender)
- Does not require the change in the sender's TCP.
- If MH disconnected, it doesn't waste time in useless transmissions and shrinks the window size to 0.
- No need to send old buffer data to new SH in case of handover (as in I-TCP).

### **Disadvantages:**

- M-TCP assumes low bit error which is not always true. So, any packet loss due to bit-errors occurring, then its propagated to the sender.
- Modifications are required for the MH protocol software.

## II. TRANSACTION - ORIENTED TCP:

- Using TCP now requires several packets over the wireless link. First, TCP uses a three-way handshake to establish the connection. At least one additional packet is usually needed for transmission of the request, and requires three more packets to close the connection via a three-way handshake.
- Assuming connections with a lot of traffic or with a long duration, this overhead is minimal. But in an example of only one data packet, TCP may need seven packets altogether. Figure 9.4 shows an example for the overhead introduced by using TCP over GPRS in a web scenario.
- Web services are based on HTTP which requires a reliable transport system. In the internet, TCP is used for this purpose. Before a HTTP request can be transmitted the TCP connection has to be established. This already requires three messages. If GPRS is used as wide area transport system, one-way delays of 500 ms and more are quite common.
- The setup of a TCP connection already takes far more than a second. This led to the development of a transaction-oriented TCP. T/TCP can combine packets for connection establishment and connection release with user data packets. This can reduce the number of packets down to two instead of seven.
- The obvious advantage for certain applications is the reduction in the overhead which standard TCP has for connection setup and connection release. However, T/TCP is not the original TCP anymore, so it requires changes in the mobile host and all correspondent hosts, which is a major disadvantage. This solution no longer hides mobility. Furthermore, T/TCP exhibits several security problems (de Vivo, 1999).



- An additional scheme that can be used to reduce TCP overhead is header compression. Using tunneling schemes as in mobile IP together with TCP, results in protocol headers of 60 byte in case of IPv4 and 100 byte for IPv6 due to the larger addresses.

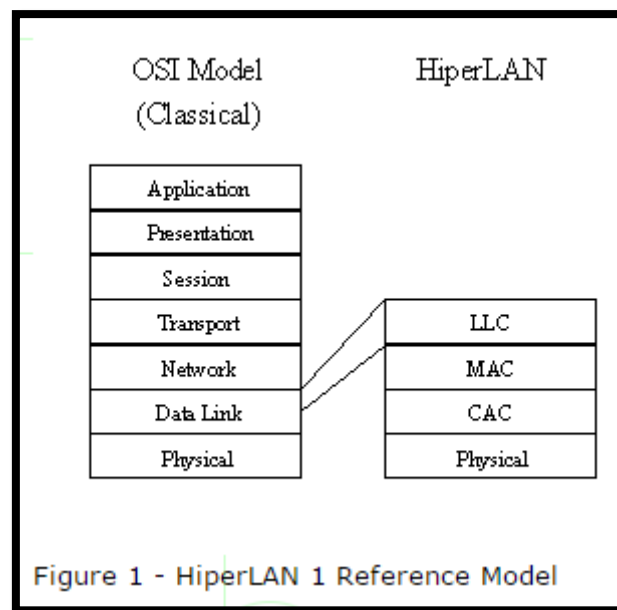
## 6) WRITE A SHORT NOTE ON HIPERLAN?

- HIPERLAN stands for **high performance local area network**. It is a wireless standard derived from traditional LAN environments and can support multimedia and asynchronous data effectively at high data rates of 23.5 Mbps.
- It is primarily a European standard alternative for the IEEE 802.11 standards and was published in 1996.
- HIPERLAN uses cellular-based data networks to connect to an ATM backbone. The main idea behind HIPERLAN is to provide an infrastructure or ad-hoc wireless with low mobility and a small radius. HIPERLAN supports isochronous traffic with low latency. The HiperLAN standard family has four different versions.
- The key feature of all four networks is their integration of time-sensitive data transfer services. Over time, names have changed and the former

HIPERLANs 2,3, and 4 are now called HiperLAN2, HIPERACCESS, and HIPERLINK.

### **HiperLAN 1 Reference Model:**

HiperLAN 1 defines Data Link Layer and Physical Layer. For Local Area Networks, Data Link Layer is further divided into two sublayers: the Logical Link Control (LLC) and the Medium Access Control (MAC). HiperLAN 1 only deals with MAC and PHY.



An intermediate layer, the Channel Access and Control (CAC) sublayer, is introduced in the HiperLAN 1 architecture to deal with the channel access signaling and protocol operation required supporting packet priority. A pseudo-hierarchically independent access mechanism is achieved via active signaling in a listen-before-talk access protocol.

The Elimination-Yield Non-Preemptive Multiple Access (EY-NPMA) mechanism codes priority level selection and contention resolution into a single, variable length radio pulse preceding packet data.

EY-NPMA provides good residual collision rate performance for even large numbers of simultaneous channel contenders.

With EYNPMA, each station may attempt to access the channel when a condition out of a group of three is met. The three conditions are:

- (a) Channel free condition
- (b) Synchronized channel condition
- (c) Hidden elimination condition

The channel free condition occurs when the channel remains idle for at least a predefined time interval. The synchronized channel access cycle consists of three distinct phases:

- (a) Prioritization
- (b) Contention (Elimination and Yield)
- (c) Transmission

In **prioritization**, EY-NPMA recognizes five distinct priorities from 0 to 4, with 0 being the highest priority. The cycle begins with each station having data to transmit sensing the channel for as many slots as the priority of the packet in its buffer. All stations that successfully sense the channel as idle for the whole interval proceed to the next phase, the elimination phase.

During the **elimination** phase, each station transmits an energy burst of random length. These bursts ensure that only the stations having the highest priority data at a time proceed to the elimination phase. The length of the energy burst is a multiple of slots up to a predefined maximum. As soon as a station finishes bursting, it immediately senses the channel. If the channel is sensed as idle, the station proceeds to the next phase. Otherwise, it leaves the cycle.

During the **yield** phase, the station that survived the two previous ones, back off for a random number of slots. The station that backs off for the shortest interval eventually gets access of the channel for data transmission. All other station senses the beginning of the **transmission** and refrain from transmitting.

## 7) EXPLAIN WIRELESS TRANSACTION PROTOCOL?

- The wireless transaction protocol (WTP) is on top of either WDP or, if security is required, WTLS. WTP has been designed to run on very thin clients, such as mobile phones.
- WTP offers several advantages to higher layers, including an improved reliability over datagram services, improved efficiency over connection-oriented services, and support for transaction-oriented services such as web browsing.
- WTP offers many features to the higher layers. The basis is formed from three classes of transaction service:

(A) Class 0 provides unreliable message transfer without any result message.

(B) Classes 1 and 2 provide reliable message transfer, class 1 without, class 2 with, exactly one reliable result message (the typical request/response case).

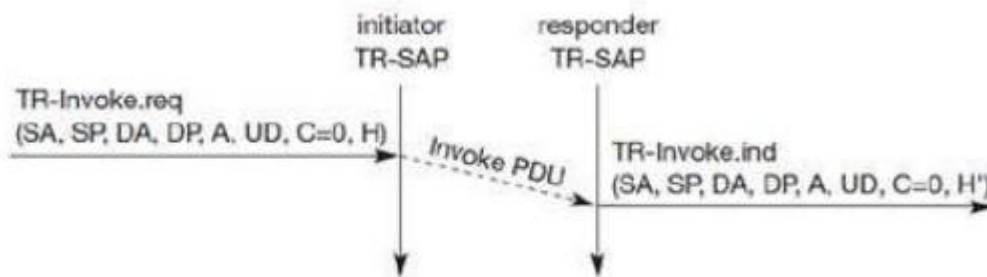
- The three service primitives offered by WTP are: **TR-Invoke** to initiate a new transaction, **TR-Result** to send back the result of a previously initiated transaction, and **TR-Abort** to abort an existing transaction. The PDUs exchanged between two WTP entities for normal transactions are the **invoke PDU, ack PDU, and result PDU**.

### (A) WTP class 0 :

Class 0 offers an unreliable transaction service without a result message.

The service is requested with the TR-Invoke.req primitive as shown in Figure. Parameters are the source address (SA), source port (SP), destination address (DA), destination port (DP) Additionally, with the A flag.

The WTP layer will transmit the user data (UD) transparently to its destination. The class type (C) indicates here class 0. Finally, the transaction handle (H) provides a simple index to uniquely identify the transaction.



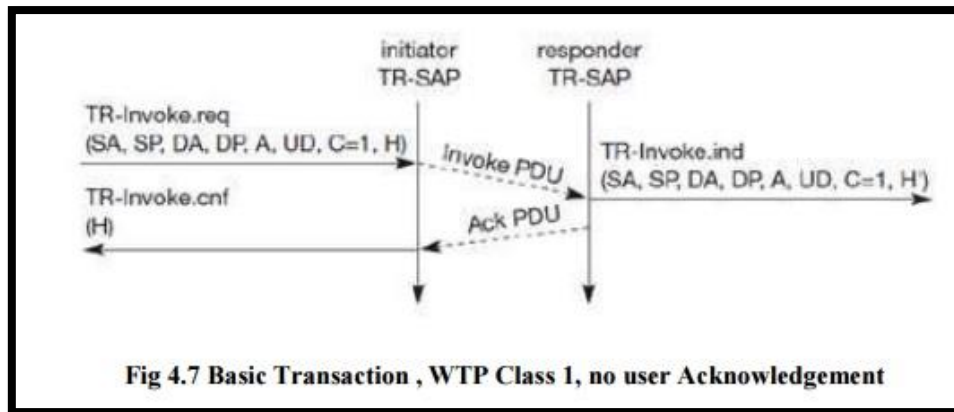
**Fig 4.6 Basic Transaction , WTP Class 0**

The WTP entity at the initiator sends an invoke PDU which the responder receives.

The WTP entity at the responder then generates a TR-Invoke.ind primitive with the same parameters as on the initiators side, except for which is now the local handle for the transaction on the responders side. In this class, the responder does not acknowledge the message and the initiator does not perform any retransmission.

## **(B) WTP class 1 :**

Class 1 offers a reliable transaction service but without a result message.



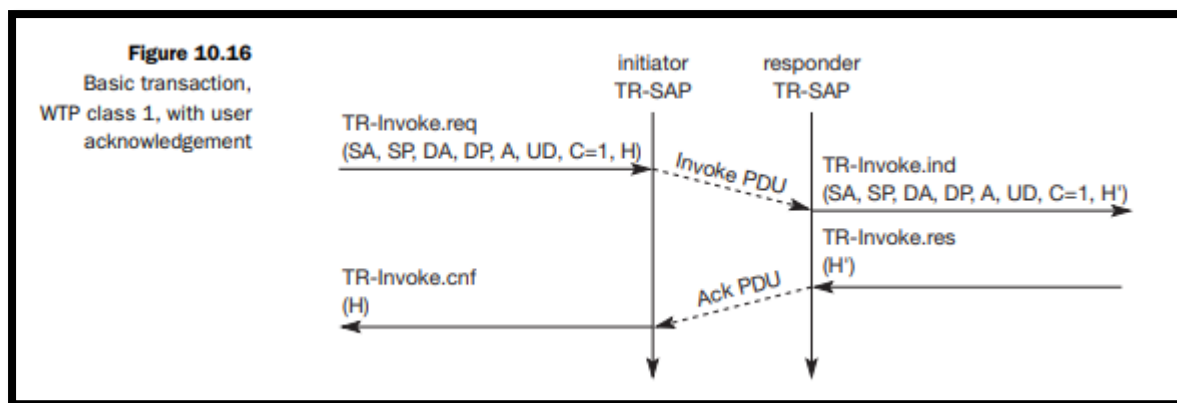
Again, the initiator sends an invoke PDU after a **TR-Invoke.req** from a higher layer. This time, class equals '1', and no user acknowledgement has been selected as shown in.

The responder signals the incoming invoke PDU via the **TR-Invoke.ind** primitive to the higher layer and acknowledges automatically without user interference.

The specification also allows the user on the responder's side to acknowledge, but this acknowledgement is not required. For the initiator the transaction ends with the reception of the acknowledgement. The responder keeps the transaction state for some time to be able to retransmit the acknowledgement if it receives the same invoke PDU again indicating a loss of the acknowledgement.

If a user of the WTP class 1 service on the initiator's side requests a user acknowledgement on the responder's side, the sequence diagram looks like **Figure shown below**. Now the WTP entity on the responder's side does not send an acknowledgement automatically, but waits for the **TR-Invoke.res** service primitive from the user. This service primitive must have the appropriate local handle H' for identification of the right transaction. The WTP entity can now send the ack PDU. Typical uses for this transaction class are reliable push services.





### (C) WTP class 2 :

Finally, class 2 transaction service provides the classic reliable request/response transaction known from many client/server scenarios.

Depending on user requirements, many different scenarios are possible for initiator/responder interaction. Two examples are presented below.

**Figure 10.17 shows the basic transaction of class 2 without-user acknowledgement.**

Here, a user on the initiator's side requests the service and the WTP entity sends the invoke PDU to the responder.

The WTP entity on the responder's side indicates the request with the **TR-Invoke.ind** primitive to a user. The responder now waits for the processing of the request, the user on the responder's side can finally give the **result UD\*** to the WTP entity on the responder side using **TR-Result.req**.

The result PDU can now be sent back to the initiator, which implicitly acknowledges the invoke PDU. The initiator can indicate the successful

transmission of the invoke message and the result with the two service primitives **TR-Invoke.cnf** and **TR-Result.ind**.

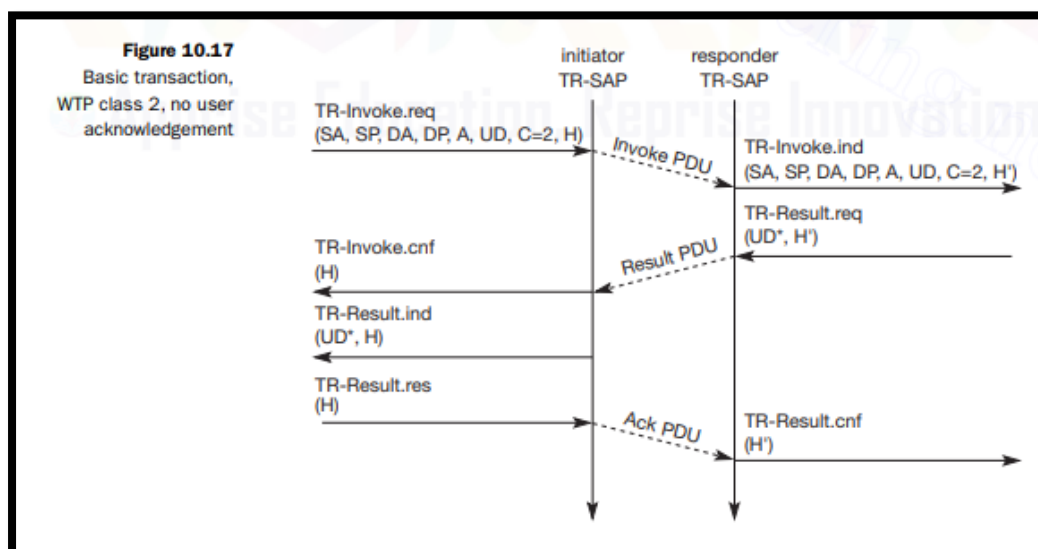
A user may respond to this result with **TR-Result.res**. An acknowledgement PDU is then generated which finally triggers the **TR-Result.cnf** primitive on the responder's side. This example clearly shows the combination of two reliable services (TR-Invoke and TR-Result) with an efficient data transmission/acknowledgement.

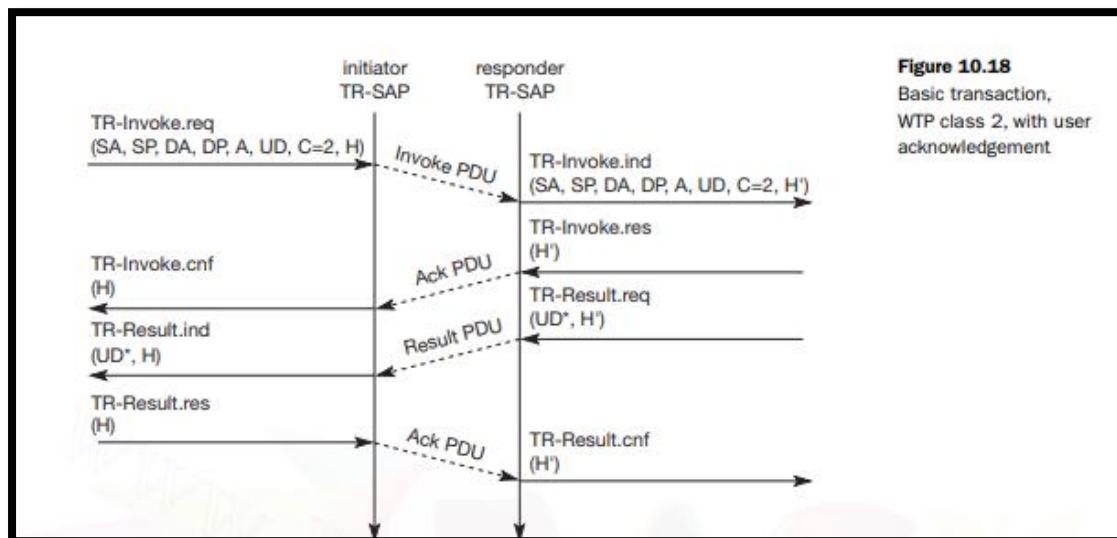
**Figure 10.18 shows the basic transaction of class 2 with -user acknowledgement.**

The time-sequence diagram looks different (see Figure 10.18).

The user on the responder's side now explicitly responds to the Invoke PDU using the TR-Invoke.res primitive, which triggers the **TR-Invoke.cnf** on the initiator's side via an ack PDU.

The transmission of the result is also a confirmed service, as indicated by the next four service primitives. This service will likely be the most common in standard request/response scenarios as, e.g., distributed computing.





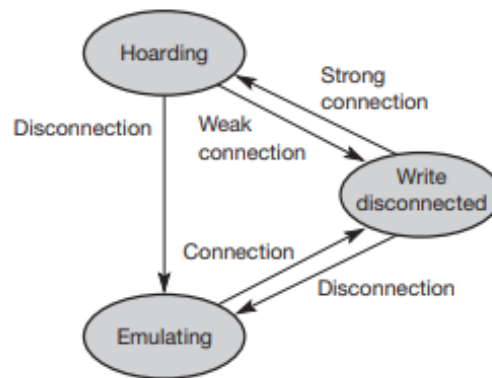
**Figure 10.18**  
Basic transaction,  
WTP class 2, with user  
acknowledgement

If the calculation of the result takes some time, the responder can put the initiator on “hold on” to prevent a retransmission of the invoke PDU as the initiator might assume packet loss if no result is sent back within a certain timeframe

## 8) EXPLAIN CODA ARCHITECTURE?

Coda is the successor of AFS (Andrew File System) and offers two different types of replication: server replication and caching on clients. Disconnected clients work only on the cache, i.e., applications use only cached replicated files.

**Figure 10.2**  
States of a client  
in Coda




- To provide all the necessary files for disconnected work, Coda offers extensive mechanisms for pre-fetching of files while still connected, called **hoarding**.
- If the client is connected to the server with a strong connection (see Figure 10.2), hoarding transparently pre-fetches files currently used. This automatic data collection is necessary for it is impossible for a standard user to know all the files currently used.
- While standard programs and application data may be familiar to a user, he or she typically does not know anything about the numerous small system files needed in addition (e.g., profiles, shared libraries, drivers, fonts). A user can pre-determine a list of files, which Coda should explicitly pre-fetch. Additionally, a user can assign priorities to certain programs. Coda now decides on the current cache content using the list and a least-recently-used (LRU) strategy.
- As soon as the client is disconnected, applications work on the replicates (see Figure 10.2, **emulating**). Coda follows an optimistic approach and allows read and write access to all files. The system keeps a record of changed files, but does not maintain a history of changes for each file. The cache always has only one replicate (possibly changed). After reconnection, Coda compares the replicates with the files on the server. If Coda notices

that two different users have changed a file, reintegration of this file fails and Coda saves the changed file as a copy on the server to allow for manual reintegration.

- Most files are just read, only some files are changed. Experiences with Coda showed that only 0.72 per cent of all file accesses resulted in write conflicts (Satyanarayanan, 1993). Considering only user files this is reduced to 0.3 per cent. However, this low conflict rate is not applicable to arbitrary shared files as used in, e.g., computer-supported cooperative work (CSCW). The tool application specific resolver (ASR) was developed to automate conflict resolution after failed reintegration (Kumar, 1993). A general problem with these tools is that they can only work after the fact. This means that the tools have to reconstruct a history of changes based on the replicate because Coda does not record every single change.
- Another problem of Coda is the definition of a conflict. Coda detects only write conflicts, i.e., if two or more users change a file. Now consider two files f1 and f2. One client uses values from files f1 and f2 to calculate something and stores the result in file f1. The other client uses values from files f1 and f2 to calculate something else and stores the result in file f2. Coda would not detect any problem during reintegration of the files. However, the results may not reflect the correct values based on the files. The order of execution plays an important role. To solve this problem, a simple transaction mechanism was introduced into Coda as an option, the so-called isolation-only transactions (IOT, (Lu, 1994)). IOT allows grouping certain operations and checks them for serial execution.
- While in the beginning Coda simply distinguished the two states “hoarding” while connected and “emulating” while disconnected, the loosely connected state **write disconnected** was later integrated, (see Figure 10.2). If a client is only weakly connected, Coda decides if it is worthwhile to fetch a file via this connection or to let the user wait until a better connection is available. In other words, Coda models the patience of a user and weighs it against the cost of fetching the file required by the user.

Figure 10.2 illustrates the three states of a client in Coda. The client only performs hoarding while a strong connection to the server exists. If the connection breaks completely, the client goes into emulating and uses only the cached replicates. If the client loses the strong connection and only a weak connection remains, it does not perform hoarding, but decides if it should fetch the file in case of a cache miss considering user patience and file type. The weak connection, however, is not used for reintegration of files.

## 9) DIFFERENTIATE WAP & WAP2.0?

 Edit		WPA	WPA2
<b>Stands For</b>		Wi-Fi Protected Access	Wi-Fi Protected Access 2
<b>What Is It?</b>		A security protocol developed by the Wi-Fi Alliance in 2003 for use in securing <a href="#">wireless networks</a> ; designed to replace the WEP protocol.	A security protocol developed by the Wi-Fi Alliance in 2004 for use in securing wireless networks; designed to replace the <a href="#">WEP and WPA</a> protocols.
<b>Methods</b>		As a temporary solution to WEP's problems, WPA still uses WEP's insecure RC4 stream cipher but provides extra security through TKIP.	Unlike WEP and WPA, WPA2 uses the AES standard instead of the RC4 stream cipher. CCMP replaces WPA's TKIP.
<b>Secure and Recommended?</b>		Somewhat. Superior to <a href="#">WEP</a> , inferior to WPA2.	WPA2 is recommended over WEP and WPA, and is more secure when Wi-Fi Protected Setup (WPS) is disabled. It is not recommended over <a href="#">WPA3</a> .

	WPA	WPA2
Year it became available	2003	2004
Encryption method	Temporal Key Integrity Protocol (TKIP)	Advanced Encryption Standard (AES)
Security strength	Stronger than WEP, offers basic security	Stronger than WPA, offers increased security
Device support	Can support older software	Only compatible with newer software
Password length	Shorter password required	Longer password required
Business usage	No enterprise solutions	Has enterprise option
Processing power required	Minimal needed	Significant amount needed

	<del>WEP</del>	✓ WPA	✓ WPA2
<b>Year introduced</b>	1999	2003	2004
<b>Encryption protocol</b>	Fixed-key	TKIP	CCMP
<b>Session key size</b>	64-bit/128-bit	256-bit	256-bit
<b>Cipher type</b>	RC4 stream cipher	TKIP (RC4-based)	AES
<b>Data integrity</b>	Cyclic Redundancy Check	Message Integrity Check	CCMP
<b>Authentication method</b>	Open system/Shared key	PSK	PSK + PMK
<b>Key management</b>	Symmetric key encryption	WPA + WPA-PSK	PMK + PSK