**Mobile IP**

**Introduction:** The IP addresses are designed to work with stationary hosts because part of the address defines the network to which the host is attached*.*A host cannot change its IP address without terminating on-going sessions and restarting them after it acquires a new address. Other link layer mobility solutions exist but are not sufficient enough for the global Internet.

***Mobility***is the abilityof a node to change its point-of-attachment while maintaining all existing communications and using the same IP address.

***Nomadicity*** allows a node to move but it must terminate all existing communications and then can initiate new connections with a new address.

Mobile IP is a network layer solution for homogenous and heterogeneous mobility on the global Internet which is scalable, robust, and secure and which allows nodes to maintain all ongoing communications while moving.

**Design Goals:** Mobile IP was developed as a means for transparently dealing with problems of mobile users. Mobile IP was designed to make the size and the frequency of required routing updates as small as possible. It was designed to make it simple to implement mobile node software. It was designed to avoid solutions that require mobile nodes to use multiple addresses.

Requirements: There are several requirements for Mobile IP to make it as a standard. Some of them are:

1.  *Compatibility*: The whole architecture of internet is very huge and a new standard cannot introduce changes to the applications or network protocols already in use. Mobile IP is to be integrated into the existing operating systems. Also, for routers also it may be possible to enhance its capabilities to support mobility instead of changing the routers which is highly impossible. Mobile IP must not require special media or MAC/LLC protocols, so it must use the same interfaces and mechanisms to access the lower layers as IP does. Finally, end-systems enhanced with a mobile IP implementation should still be able to communicate with fixed systems without mobile IP.

2. *Transparency*: Mobility remains invisible for many higher layer protocols and applications. Higher layers continue to work even if the mobile computer has changed its point of attachment to the network and even notice a lower bandwidth and some interruption in the service. As many of today’s applications have not been designed to use in mobile environments, the effects of mobility will be higher delay and lower bandwidth.

3. *Scalability and efficiency*: The efficiency of the network should not be affected even if a new mechanism is introduced into the internet. Enhancing IP for mobility must not generate many new messages flooding the whole network. Special care is necessary to be taken considering the lower bandwidth of wireless links. Many mobile systems have a wireless link to an attachment point. Therefore, only some additional packets must be necessary between a mobile system and a node in the network. It is indispensable for a mobile IP to be scalable over a large number of participants in the whole internet, throughout the world.

4. *Security:* Mobility possesses many security problems. A minimum requirement is the authentication of all messages related to the management of mobile IP. It must be sure for the IP layer if it forwards a packet to a mobile host that this host really is the receiver of the packet. The IP layer can only guarantee that the IP address of the receiver is correct. There is no way to prevent faked IP addresses and other attacks.

The goal of a mobile IP can be summarized as: ‘supporting end-system mobility while maintaining scalability, efficiency, and compatibility in all respects with existing applications and Internet protocols’.

**Entities and terminology for Mobile IP**

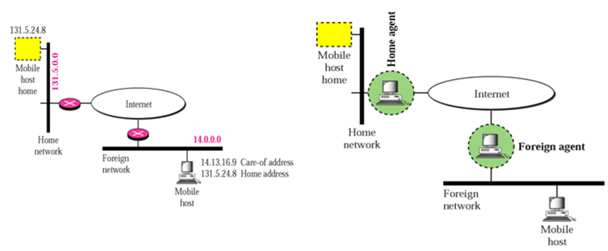
**Introduction:** The following defines several entities and terms needed to understand mobile IP as defined in RFC 3344.

* **Mobile Node (MN):**A mobile node is an end-system or router that can change its point of attachment to the internet using mobile IP. The MN keeps its IP address and can continuously communicate with any other system in the internet as long as link-layer connectivity is given. Examples are laptop, mobile phone, router on an aircraft etc.
* **Correspondent node (CN):**At least one partner is needed for communication. In the following the CN represents this partner for the MN. The CN can be a fixed or mobile node.
* **Home network:**The home network is the subnet the MN belongs to with respect to its IP address. No mobile IP support is needed within the home network.
* **Foreign network:**The foreign network is the current subnet the MN visits and which is not the home network.

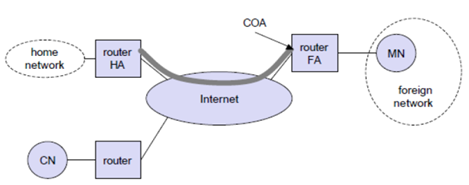
**Foreign agent (FA):**The FA can provide several services to the MN during its visit to the foreign network. The FA can have the COA, acting as tunnel endpoint and forwarding packets to the MN. The FA can be the default router for the MN. FAs can also provide security services because they belong to the foreign network as opposed to the MN which is only visiting. FA is implemented on a router for the subnet the MN attaches to.

**Care-of address (COA):**The COA defines the current location of the MN from an IP point of view. All IP packets sent to the MN are delivered to the COA, not directly to the IP address of the MN. Packet delivery toward the MN is done using a tunnel, i.e., the COA marks the tunnel endpoint, i.e., the address where packets exit the tunnel. There are two different possibilities for the location of the COA:

* **Foreign agent COA:**The COA could be located at the FA, i.e., the COA is an IP address of the FA. The FA is the tunnel end-point and forwards packets to the MN. Many MN using the FA can share this COA as common COA.
* **Co-located COA:**The COA is co-located if the MN temporarily acquired an additional IP address which acts as COA. This address is now topologically correct, and the tunnel endpoint is at the MN. Co-located addresses can be acquired using services such as DHCP.

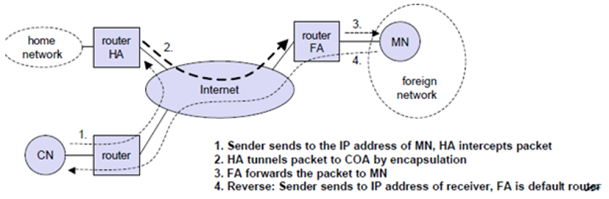
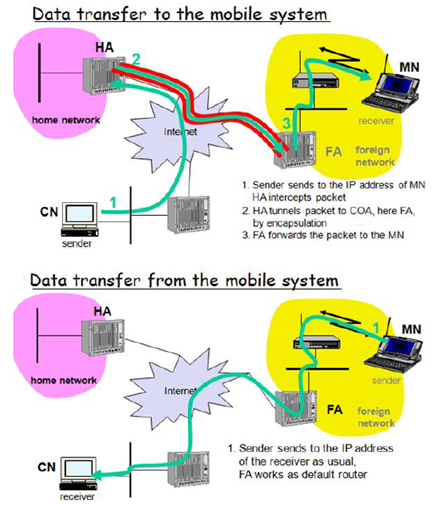
**Home agent (HA):**The HA provides several services for the MN and is located in the home network. The tunnel for packets toward the MN starts at the HA. The HA maintains a location registry, i.e., it is informed of the MN’s location by the current COA. Three alternatives for the implementation of an HA exist.

* The HA can be implemented on a router that is responsible for the home network. This is obviously the best position, because without optimizations to mobile IP, all packets for the MN have to go through the router anyway.
* If changing the router’s software is not possible, the HA could also be implemented on an arbitrary node in the subnet. One disadvantage of this solution is the double crossing of the router by the packet if the MN is in a foreign network. A packet for the MN comes in via the router; the HA sends it through the tunnel which again crosses the router.
* Finally, a home network is not necessary at all. The HA could be again on the ‘router’ but this time only acting as a manager for MNs belonging to a virtual home network. All MNs are always in a foreign network with this solution.

A CN is connected via a router to the internet, as are the home network and the foreign network. The HA is implemented on the route connecting the home network with the internet, an FA is implemented on the router to the foreign network. The MN is currently in the foreign network. The tunnel for packets toward the MN starts at the HA and ends at the FA, for the FA has the COA in the above example.

**IP packet delivery**

**Introduction:** Consider the above example in which a correspondent node (CN) wants to send an IP packet to the MN. One of the requirements of mobile IP was to support hiding the mobility of the MN. CN does not need to know anything about the MN’s current location and sends the packet as usual to the IP address of MN as shown below.

CN sends an IP packet with MN as a destination address and CN as a source address. The internet, not having information on the current location of MN, routes the packet to the router responsible for the home network of MN. This is done using the standard routingmechanisms of the internet. The HA now intercepts the packet, knowing that MN is currently not in its home network. The packet is not forwarded into the subnet as usual, but encapsulated and tunneled to the COA. A new header is put in front of the old IP header showing the COA as new destination and HA as source of the encapsulated packet (step 2).

The foreign agent now decapsulates the packet, i.e., removes the additional header, and forwards the original packet with CN as source and MN as destination to the MN (step 3). Again, for the MN mobility is not visible. It receives the packet with the same sender and receiver address as it would have done in the home network.

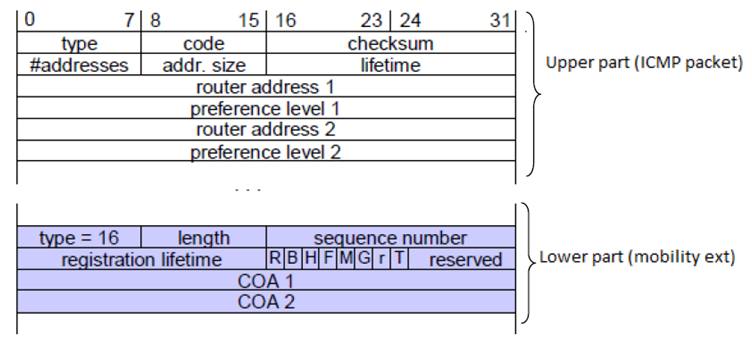
Sending packets from the mobile node (MN) to the CN is comparatively simple. The MN sends the packet as usual with its own fixed IP address as source and CN’s address as destination (step 4). The router with the FA acts as default router and forwards the packet in the same way as it would do for any other node in the foreign network. As long as CN is a fixed node the remainder is in the fixed internet as usual. If CN were also a mobile node residing in a foreign network, the same mechanisms as described in steps 1 through 3 would apply now in the other direction.

**Working of Mobile IP**:- Mobile IP has two addresses for a mobile host: one home address and one care-of address. The home address is permanent; the care-of addresses changes as the mobile host moves from one network to another. To make the change of address transparent to the rest of the Internet requires a home agent and a foreign agent. The specific function of an agent is performed in the application layer. When the mobile host and the foreign agent are the same, the care-of address is called a co-located care-of address. To communicate with a remote host, a mobile host goes through three phases: agent discovery, registration, and data transfer.

**Agent Discovery**

**Introduction:** A mobile node has to find a foreign agent when it moves away from its home network. To solve this problem, mobile IP describes two methods: agent advertisement and agent solicitation.

**Agent advertisement:**

For this method, foreign agents and home agents advertise their presence periodically using special **agent advertisement**messages, which are broadcast into the subnet. Mobile IP does not use a new packet type for agent advertisement; it uses the router advertisement packet of ICMP, and appends an agent advertisement message. The agent advertisement packet according to RFC 1256 with the extension for mobility is shown below:

The TTL field of the IP packet is set to 1 for all advertisements to avoid forwarding them. The **type**is set to 9, the **code**can be 0, if the agent also routes traffic from non-mobile nodes, or 16, if it does not route anything other than mobile traffic. The number of addresses advertised with this packet is in **#addresses**while the **addresses**themselves follow as shown. **Lifetime**denotes the length of time this advertisement is valid. **Preference**levels for each address help a node to choose the router that is the most eager one to get a new node.

The extension for mobility has the following fields defined: **type**is set to 16, **length**depends on the number of COAs provided with the message and equals 6 4\*(number of addresses). The **sequence number**shows the total number of advertisements sent since initialization by the agent. By the **registration lifetime**the agent can specify the maximum lifetime in seconds a node can request during registration. The following bits specify the characteristics of an agent in detail.

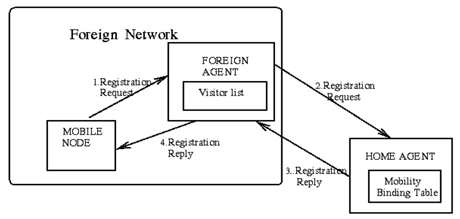
The **R**bit (registration) shows, if a registration with this agent is required even when using a collocated COA at the MN. If the agent is currently too busy to accept new registrations it can set the **B**bit. The following two bits denote if the agent offers services as a home agent (**H**) or foreign agent (**F**) on the link where the advertisement has been sent. Bits M and G specify the method of encapsulation used for the tunnel. While IP-in-IP encapsulation is the mandatory standard, **M**can specify minimal encapsulation and **G**generic routing encapsulation. In the first version of mobile IP (RFC 2002) the **V**bit specified the use of header compression according to RFC 1144. Now the field **r**at the same bit position is set to zero and must be ignored. The new field **T**indicates that reverse tunneling is supported by the FA. The following fields contain the **COAs**advertised. A foreign agent setting the F bit must advertise at least one COA. A mobile node in a subnet can now receive agent advertisements from either its home agent or a foreign agent. This is one way for the MN to discover its location.

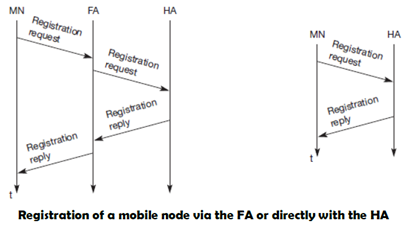
**Agent Solicitation**

If no agent advertisements are present or the inter-arrival time is too high, and an MN has not received a COA by other means, the mobile node must send **agent solicitations**. Care must be taken to ensure that these solicitation messages do not flood the network, but basically an MN can search for an FA endlessly sending out solicitation messages. If a node does not receive an answer to its solicitations it must decrease the rate of solicitations exponentially to avoid flooding the network until it reaches a maximum interval between solicitations (typically one minute). Discovering a new agent can be done anytime, not just if the MN is not connected to one. After these steps of advertisements or solicitations the MN can now receive a COA, either one for an FA or a co-located COA.

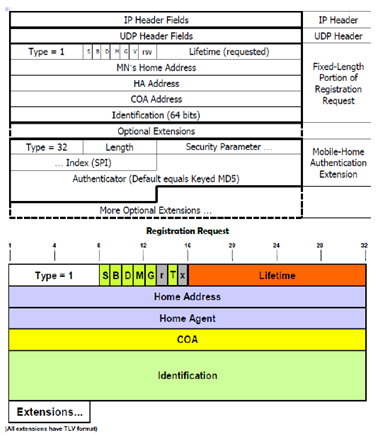
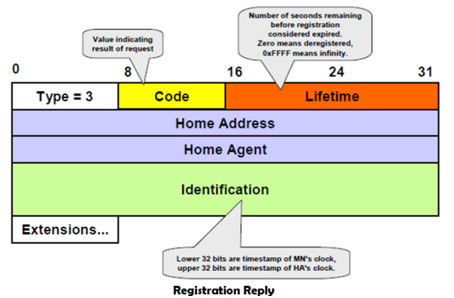
**Agent Registration**

**Introduction:** Having received a COA, the MN has to register with the HA. The main purpose of the registration is to inform the HA of the current location for correct forwarding of packets.

Registration can be done in two different ways depending on the location of the COA.



* If the COA is at the FA, the MN sends its registration request containing the COA to the FA which forwards the request to the HA. The HA now sets up a **mobility binding,**containing the mobile node’s home IP address and the current COA. It also contains the lifetime of the registration which is negotiated during the registration process. Registration expires automatically after the lifetime and is deleted; so, an MN should reregister before expiration. This mechanism is necessary to avoid mobility bindings which are no longer used. After setting up the mobility binding, the HA sends a reply message back to the FA which forwards it to the MN.
* If the COA is co-located, registration can be simpler, the MN sends the request directly to the HA and vice versa. This is also the registration procedure for MNs returning to their home network to register directly with the HA.
* UDP packets are used for the registration requests using the port no 434. The IP source address of the packet is set to the interface address of the MN, the IP destination address is that of the FA or HA.



The first field **type**is set to 1 for a registration request. With the **S**bit an MN can specify if it wants the HA to retain prior mobility bindings. This allows for simultaneous bindings. Setting the **B**bit generally indicates that an MN also wants to receive the broadcast packets which have been received by the HA in the home network. If an MN uses a co-located COA, it also takes care of the decapsulation at the tunnel endpoint. The **D**bit indicates this behavior. As already defined for agent advertisements, the bits **M**and **G**denote the use of minimal encapsulation or generic routing encapsulation, respectively. **T**indicates reverse tunneling;**r**and **x**are set to zero.

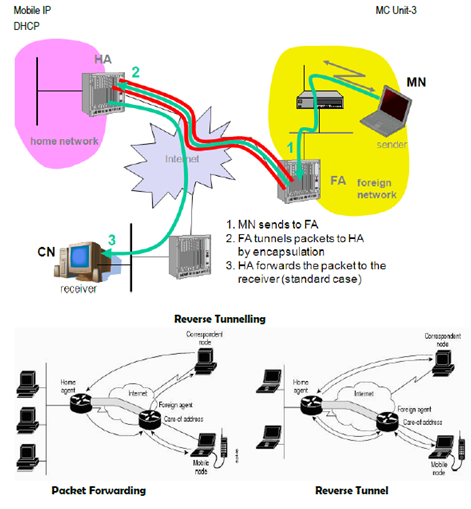
**Lifetime**denotes the validity of the registration in seconds. A value of zero indicates deregistration; all bits set indicates infinity. The **home address**is the fixed IP address of the MN, **home agent**is the IP address of the HA, and **COA**represents the tunnel endpoint. The 64 bit **identification**is generated by the MN to identify a request and match it with registration replies. This field is used for protection against replay attacks of registrations. The **extensions**must at least contain parameters for authentication

A **registration reply**, which is conveyed in a UDP packet, contains a **type**field set to 3 and a **code**indicating the result of the registration request.

The **lifetime**field indicates how many seconds the registration is valid if it was successful. **Home address**and **home agent**are the addresses of the MN and the HA, respectively. The 64-bit **identification**is used to match registration requests with replies. The value is based on the identification field from the registration and the authentication method. Again, the **extensions**must

**Reverse Tunneling**

**Introduction:** The reverse path from MS to the CN looks quite simple as the MN can directly send its packets to the CN as in any other standard IP situation. The destination address in the packets is that of CN.



**But it has some problems explained below:-**

Quite often firewalls are designed to only allow packets with topologically correct addresses to pass to provide simple protection against misconfigured systems of unknown addresses. However, MN still sends packets with its fixed IP address as source which is not topologically correct in a foreign network. Firewalls often filter packets coming from outside containing a source address from computers of the internal network. This also implies that an MN cannot send a packet to a computer residing in its home network.

* While the nodes in the home network might participate in a multi-cast group, an MN in a foreign network cannot transmit multi-cast packets in a way that they emanate from its home network without a reverse tunnel. The foreign network might not even provide the technical infrastructure for multi-cast communication (multi-cast backbone, Mbone).
* If the MN moves to a new foreign network, the older TTL might be too low for the packets to reach the same destination nodes as before. Mobile IP is no longer transparent if a user has to adjust the TTL while moving. A reverse tunnel is needed that represents only one hop, no matter how many hops are really needed from the foreign to the home network

Based on the above considerations, reverse tunneling is defined as an extension to mobile IP (per RFC 2344). It was designed backward compatible to mobile IP and defines topologically correct reverse tunneling to handle the above stated problems

Reverse tunneling does not solve

* problems with *firewalls*, the reverse tunnel can be abused to circumvent security mechanisms (tunnel hijacking)
* optimization of data paths, i.e. packets will be forwarded through the tunnel via the HA to a sender (double triangular routing)

## Working of Mobile IP

The working of Mobile IP can be described in 3 phases:

**Agent Discovery**

In the Agent Discovery phase, the mobile nodes discover their Foreign and Home Agents. The Home Agent and Foreign Agent advertise their services on the network using the ICMP Router Discovery Protocol (IRDP).

**Registration**

The registration phase is responsible for informing the current location of the home agent and foreign agent for the correct forwarding of packets.

**Tunneling**

This phase is used to establish a virtual connection as a pipe for moving the data packets between a tunnel entry and a tunnel endpoint.

## Applications of Mobile IP

The mobile IP technology is used in many applications where the sudden changes in network connectivity and IP address can cause problems. It was designed to support seamless and continuous Internet connectivity.

It is used in many wired and wireless environments where users have to carry their mobile devices across multiple LAN subnets.

Although Mobile IP is not required within cellular systems such as 3G, it is often used in 3G systems to provide seamless IP mobility between different packet data serving node (PDSN) domains.

**IPv6**

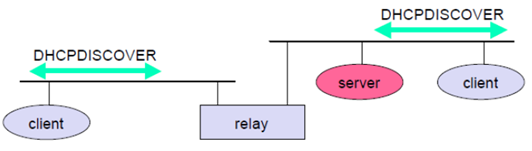
**Introduction:** The design of Mobile IP support in IPv6 (Mobile IPv6) benefits both from the experiences gained from the development of Mobile IP support in IPv4, and from the opportunities provided by IPv6. Mobile IPv6 thus shares many features with Mobile IPv4, but is integrated into IPv6 and offers many other improvements.

**This section summarizes the major differences between Mobile IPv4 and Mobile IPv6:**

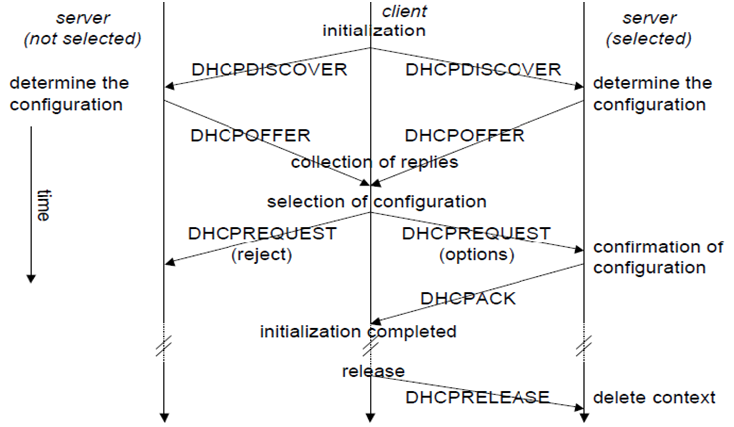
* There is no need to deploy special routers as "foreign agents", as in Mobile IPv4. Mobile IPv6 operates in any location without any special support required from the local router.
* Support for route optimization is a fundamental part of the protocol, rather than a nonstandard set of extensions.
* Mobile IPv6 route optimization can operate securely even without pre-arranged security associations. It is expected that route optimization can be deployed on a global scale between all mobile nodes and correspondent nodes.
* Support is also integrated into Mobile IPv6 for allowing route optimization to coexist efficiently with routers that perform "ingress filtering"
* The IPv6 Neighbor Unreachability Detection assures symmetric reachability between the mobile node and its default router in the current location.
* Most packets sent to a mobile node while away from home in Mobile IPv6 are sent using an IPv6 routing header rather than IP encapsulation, reducing the amount of resulting overhead compared to Mobile IPv4.
* Mobile IPv6 is decoupled from any particular link layer, as it uses IPv6 Neighbor Discovery instead of ARP. This also improves the robustness of the protocol.
* The use of IPv6 encapsulation (and the routing header) removes the need in Mobile IPv6 to manage "tunnel soft state".
* The dynamic home agent address discovery mechanism in Mobile IPv6 returns a single reply to the mobile node. The directed broadcast approach used in IPv4 returns separate replies from each home agent.

**Dynamic Host Configuration Protocol (DHCP)**

**Introduction: DHCP i**s an automatic configuration protocol used on IP networks. **DHCP**allows a computer to join an IP-based network without having a pre-configured IP address. DHCP is a protocol that assigns unique IP addresses to devices, then releases and renews these addresses as devices leave and re-join the network. 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, e.g., addresses of a DNS server and the default router, the subnet mask, the domain name, and an IP address. 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 as shown below. DHCP clients send a request to a server (DHCPDISCOVER in the example) to which the server responds. A client sends requests using MAC broadcasts to reach all devices in the LAN. A DHCP relay might be needed to forward requests across inter-working units to a DHCP server.

Consider the scenario where there is one client and two servers are present. A typical initialization of a DHCP client is shown below:

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. 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 possibleclients. 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. Now the server can free the context stored for the client and offer the configuration again. The configuration a client gets from a server is only leased for a certain amount of time; it has to be reconfirmed from time to time. Otherwise the server will free the configuration. This timeout of configuration helps in the case of crashed nodes or nodes moved away without releasing the context.

DHCP is a good candidate for supporting the acquisition of care-of addresses for mobile nodes. The same holds for all other parameters needed, such as addresses of the default router, DNS servers, the timeserver etc. A DHCP server should be located in the subnet of the access point of the mobile node, or at least a DHCP relay should provide forwarding of the messages. RFC 3118 specifies authentication for DHCP messages so as to provide protection from malicious DHCP servers. Without authentication, a DHCP server cannot trust the mobile node and vice versa…

## Benefits of DHCP

There are following benefits of DHCP:

**Centralized administration of IP configuration:** DHCP IP configuration information can be stored in a single location and enables that administrator to centrally manage all IP address configuration information.

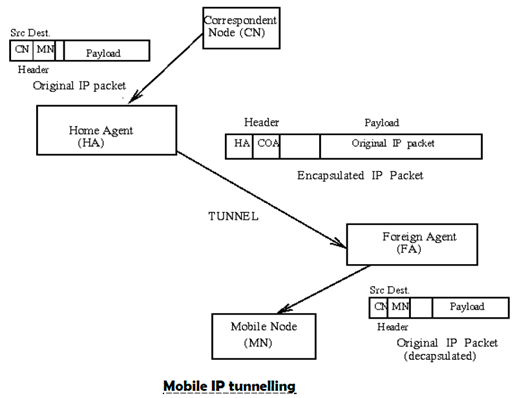
**Dynamic host configuration:** DHCP automates the host configuration process and eliminates the need to manually configure individual host. When TCP/IP (Transmission control protocol/Internet protocol) is first deployed or when IP infrastructure changes are required.

**Seamless IP host configuration:** The use of DHCP ensures that DHCP clients get accurate and timely IP configuration IP configuration parameter such as IP address, subnet mask, default gateway, IP address of DND server and so on without user intervention.

**Flexibility and scalability:** Using DHCP gives the administrator increased flexibility, allowing the administrator to move easily change IP configuration when the infrastructure changes.

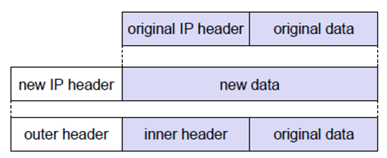
**Tunneling and encapsulation**

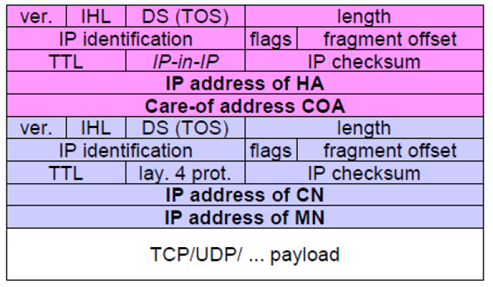
**Introduction:** A **tunnel**establishes a virtual pipe for data packets between a tunnel entry and a tunnel endpoint. Packets entering a tunnel are forwarded inside the tunnel and leave the tunnel unchanged. Tunneling, i.e., sending a packet through a tunnel is achieved by using encapsulation.

**Encapsulation**is the mechanism of taking a packet consisting of packet header and data and putting it into the data part of a new packet. The reverse operation, taking a packet out of the data part of another packet, is called **decapsulation**. Encapsulation and decapsulation are the operations typically performed when a packet is transferred from a higher protocol layer to a lower layer or from a lower to a higher layer respectively.

The HA takes the original packet with the MN as destination, puts it into the data part of a new packet and sets the new IP header so that the packet is routed to the COA. The new header is called outer header.

**IP-in-IP encapsulation**

There are different ways of performing the encapsulation needed for the tunnel between HA and COA. Mandatory for mobile IP is **IP-in-IP encapsulation**as specified in RFC 2003. The following fig shows a packet inside the tunnel.

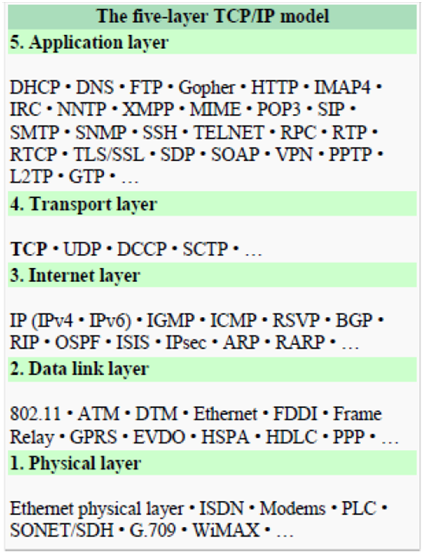
The version field **ver**is 4 for IP version 4, the internet header length (**IHL**) denotes the length of the outer header in 32 bit words.

**DS(TOS)**is just copied from the inner header, the **length**field covers the complete encapsulated packet. The fields up to TTL have no special meaning for mobile IP and are set according to RFC 791. **TTL**must be high enough so the packet can reach the tunnel endpoint. The next field, here denoted with **IP-in-IP**, is the type of the protocol used in the IP payload. This field is set to 4, the protocol type for IPv4 because again an IPv4 packet follows after this outer header. IP **checksum**is calculated as usual. The next fields are the tunnel entry as source address (the **IP address of the HA**) and the tunnel exit point as destination address (the **COA**).

If no options follow the outer header, the inner header starts with the same fields as above. This header remains almost unchanged during encapsulation, thus showing the original sender CN and the receiver MN of the packet. The only change is TTL which is decremented by 1. This means that the whole tunnel is considered a single hop from the original packet’s point of view. This is a very important feature of tunneling as it allows the MN to behave as if it were attached to the home network. No matter how many real hops the packet has to take in the tunnel, it is just one (logical) hop away for the MN. Finally, the payload follows the two headers.

**Traditional TCP(Transmission Control Protocol)**

**Introduction:** The **Transmission Control Protocol (TCP)** is one of the core protocols of the Internet protocol suite, often simply referred to as TCP/IP. TCP is reliable, guarantees in-order delivery of data and incorporates congestion control and flow control mechanisms.

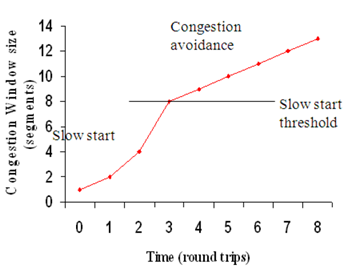
TCP supports many of the Internet's most popular application protocols and resulting applications, including the World Wide Web, e-mail, File Transfer Protocol and Secure Shell. In the Internet protocol suite, TCP is the intermediate layer between the Internet layer and application layer.

**The major responsibilities of TCP in an active session are to:**

* **Provide reliable in-order transport of data**: to not allow losses of data.
* **Control congestions in the networks**: to not allow degradation of the network performance,
* **Control a packet flow between the transmitter and the receiver**: to not exceed the receiver's capacity.

TCP uses a number of mechanisms to achieve high performance and avoid 'congestion collapse', where network performance can fall by several orders of magnitude. These mechanisms control the rate of data entering the network, keeping the data flow below a rate that would trigger collapse. There are several mechanisms of TCP that influence the efficiency of TCP in a mobile environment. Acknowledgments for data sent, or lack of acknowledgments, are used by senders to implicitly interpret network conditions between the TCP sender and receiver.

**Congestion Control**

**Introduction:** A transport layer protocol such as TCP has been designed for fixed networks with fixed end- systems. Congestion may appear from time to time even in carefully designed networks. The packet buffers of a router are filled and the router cannot forward the packets fast enough because the sum of the input rates of packets destined for one output link is higher than the capacity of the output link. The only thing a router can do in this situation is to drop packets. A dropped packet is lost for the transmission, and the receiver notices a gap in the packet stream. Now the receiver does not directly tell the sender which packet is missing, but continues to acknowledge all in-sequence packets up to the missing one.

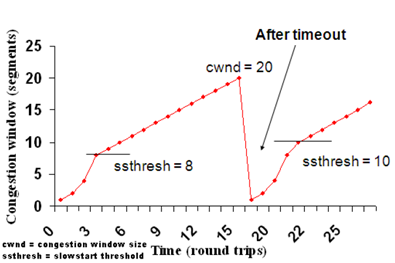
The sender notices the missing acknowledgement for the lost packet and assumes a packet loss due to congestion. Retransmitting the missing packet and continuing at full sending rate would now be unwise, as this might only increase the congestion. To mitigate congestion, TCP slows down the transmission rate dramatically. All other TCP connections experiencing the same congestion do exactly the same so the congestion is soon resolved.

**Slow start:**

TCP’s reaction to a missing acknowledgement is quite drastic, but it is necessary to get rid of congestion quickly. The behavior TCP shows after the detection of congestion is called **slow start.**The sender always calculates a **congestion window**for a receiver. The start size of the congestion window is one segment (TCP packet). The sender sends one packet and waits for acknowledgement. If this acknowledgement arrives, the sender increases the congestion window by one, now sending two packets (congestion window = 2). This scheme doubles the congestion window every time the acknowledgements come back, which takes one round trip time (RTT). This is called the exponential growth of the congestion window in the slow start mechanism.

But doubling the congestion window is too dangerous. The exponential growth stops at the **congestion threshold**. As soon as the congestion window reaches the congestion threshold, further increase of the transmission rate is only linear by adding 1 to the congestion window each time the acknowledgements come back.

Linear increase continues until a time-out at the sender occurs due to a missing acknowledgement, or until the sender detects a gap in transmitted data because of continuous acknowledgements for the same packet. In either case the sender sets the congestion threshold to half of the current congestion window. The congestion window itself is set to one segment and the sender starts sending a single segment. The exponential growth starts once more up to the new congestion threshold, then the window grows in linear fashion.

**Fast retransmit/fast recovery:**

The congestion threshold can be reduced because of two reasons. First one is if the sender receives continuous acknowledgements for the same packet. It informs the sender that the receiver has got all the packets upto the acknowledged packet in the sequence and also the receiver is receiving something continuously from the sender. The gap in the packet stream is not due to congestion, but a simple packet loss due to a transmission error. The sender can now retransmit the missing packet(s) before the timer expires. This behavior is called **fast retransmit**. It is an early enhancement for preventing slow-start to trigger on losses not caused by congestion. The receipt of acknowledgements shows that there is no congestion to justify a slow start. The sender can continue with the current congestion window. The sender performs a **fast recovery**from the packet loss. This mechanism can improve the efficiency of TCP dramatically. The other reason for activating slow start is a time-out due to a missing acknowledgement. TCP using fast retransmit/fast recovery interprets this congestion in the network and activates the slow start mechanism.

The advantage of this method is its simplicity. Minor changes in the MH’s software results in performance increase. No changes are required in FA or CH.

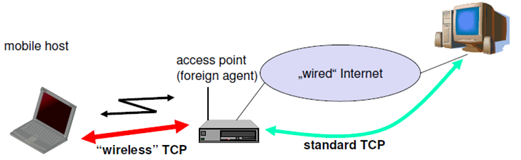
The **disadvantage** of this scheme is insufficient isolation of packet losses. It mainly focuses on problems regarding Handover. Also it affects the efficiency when a CH transmits already delivered packets.

**Problems with Traditional TCP in wireless environments:**

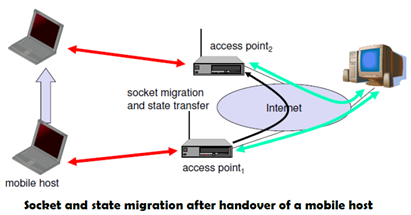
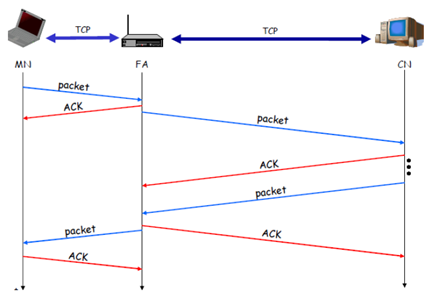
* Slow Start mechanism in fixed networks decreases the efficiency of TCP if used with mobile receivers or senders.
* Error rates on wireless links are orders of magnitude higher compared to fixed fiber or copper links. This makes compensation for packet loss by TCP quite difficult.
* Mobility itself can cause packet loss. There are many situations where a soft handover from one access point to another is not possible for a mobile end-system.
* Standard TCP reacts with slow start if acknowledgements are missing, which does not help in the case of transmission errors over wireless links and which does not really help during handover. This behavior results in a severe performance degradation of an unchanged TCP if used together with wireless links or mobile nodes

**Classical TCP Improvements**

**Introduction: Indirect TCP (I-TCP):**Indirect TCP segments a TCP connection into a fixed part and a wireless part. The following figure shows an example with a mobile host connected via a wireless link and an access point to the ‘wired’ internet where the correspondent host resides.

Standard TCP is used between the fixed computer and the access point. No computer in the internet recognizes any changes to TCP. Instead of the mobile host, the access point now terminates the standard TCP connection, acting as a proxy. This means that the access point is now seen as the mobile host for the fixed host and as the fixed host for the mobile host. Between the access point and the mobile host, a special TCP, adapted to wireless links, is used. However, changing TCP for the wireless link is not a requirement. A suitable place for segmenting the connection is at the foreign agent as it not only controls the mobility of the mobile host anyway and can also hand over the connection to the next foreign agent when the mobile host moves on.

The foreign agent acts as a proxy and relays all data in both directions. If CH (correspondent host) sends a packet to the MH, the FA acknowledges it and forwards it to the MH. MH acknowledges on successful reception, but this is only used by the FA. If a packet is lost on the wireless link, CH doesn’t observe it and FA tries to retransmit it locally to maintain reliable data transport. If the MH sends a packet, the FA acknowledges it and forwards it to CH. If the packet is lost on the wireless link, the mobile hosts notice this much faster due to the lower round trip time and can directly retransmit the packet. Packet loss in the wired network is now handled by the foreign agent.

During handover, the buffered packets, as well as the system state (packet sequence number, acknowledgements, ports, etc), must migrate to the new agent. No new connection may be established for the mobile host, and the correspondent host must not see any changes in connection state. Packet delivery in I-TCP is shown below:

1. No changes in the fixed network necessary, no changes for the hosts (TCP protocol) necessary, all current optimizations to TCP still work
2. Simple to control, mobile TCP is used only for one hop between, e.g., a foreign agent and mobile host

* transmission errors on the wireless link do not propagate into the fixed network
* therefore, a very fast retransmission of packets is possible, the short delay on the mobile hop s known

1. It is always dangerous to introduce new mechanisms in a huge network without knowing exactly how they behave.

* New optimizations can be tested at the last hop, without jeopardizing the stability of the Internet.

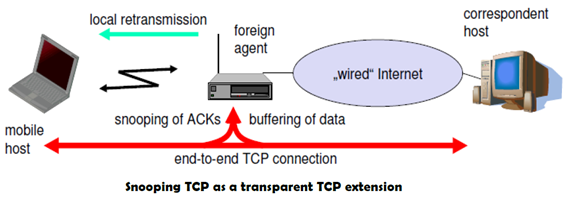
1. It is easy to use different protocols for wired and wireless networks.

**Disadvantages of I-TCP**

1. Loss of end-to-end semantics:- an acknowledgement to a sender no longer means that a receiver really has received a packet, foreign agents might crash.
2. Higher latency possible:- due to buffering of data within the foreign agent and forwarding to a new foreign agent
3. Security issue:- The foreign agent must be a trusted entity

**Snooping TCP**

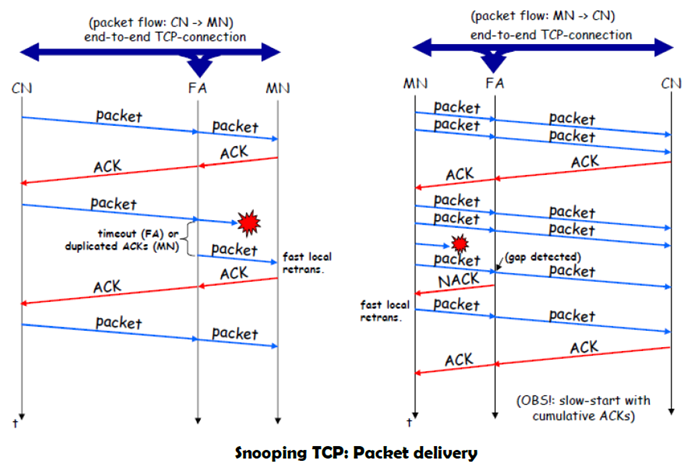
**Introduction:** The main drawback of I-TCP is the segmentation of the single TCP connection into two TCP connections, which loses the original end-to-end TCP semantic. A new enhancement, which leaves the TCP connection intact and is completely transparent, is Snooping TCP. The main function is to buffer data close to the mobile host to perform fast local retransmission in case of packet loss.

Here, the foreign agent buffers all packets with **destination mobile host**and additionally ‘snoops’ the packet flow in both directions to recognize acknowledgements. The foreign agent buffers every packet until it receives an acknowledgement from the mobile host. If the FA does not receive an acknowledgement from the mobile host within a certain amount of time, either the packet or the acknowledgement has been lost. Alternatively, the foreign agent could receive a duplicate ACK which also shows the loss of a packet. Now, the FA retransmits the packet directly from the buffer thus performing a faster retransmission compared to the CH. For transparency, the FA does not acknowledge data to the CH, which would violate end-to-end semantic in case of a FA failure. The foreign agent can filter the duplicate acknowledgements to avoid unnecessary retransmissions of data from the correspondent host. If the foreign agent now crashes, the time-out of the correspondent host still works and triggers a retransmission. The foreign agent may discard duplicates of packets already retransmitted locally and acknowledged by the mobile host. This avoids unnecessary traffic on the wireless link.

For data transfer from the mobile host with **destination correspondent host**, the FA snoops into the packet stream to detect gaps in the sequence numbers of TCP. As soon as the foreign agent detects a missing packet, it returns a negative acknowledgement (NACK) to the mobile host. The mobile host can now retransmit the missing packet immediately. Reordering of packets is done automatically at the correspondent host by TCP.

**Advantages of snooping TCP:**

1. The end-to-end TCP semantic is preserved.
2. Most of the enhancements are done in the foreign agent itself which keeps correspondent host unchanged.
3. Handover of state is not required as soon as the mobile host moves to another foreign agent. Even though packets are present in the buffer, time out at the CH occurs and the packets are transmitted to the new COA.
4. No problem arises if the new foreign agent uses the enhancement or not. If not, the approach automatically falls back to the standard solution.



**Disadvantages of snooping TCP**

1. Snooping TCP does not isolate the behavior of the wireless link as well as I-TCP. Transmission errors may propagate till CH.
2. Using negative acknowledgements between the foreign agent and the mobile host assumes additional mechanisms on the mobile host. This approach is no longer transparent for arbitrary mobile hosts.
3. Snooping and buffering data may be useless if certain encryption schemes are applied end-to-end between the correspondent host and mobile host. If encryption is used above the transport layer, (eg. SSL/TLS), snooping TCP can be used.

**Wireless Application Protocol-WAP**

**Introduction:** The Wireless Application Protocol (WAP) is an open, global specification that empowers mobile users with wireless devices to easily access and interact with information and services instantly.

WAP is a global standard and is not controlled by any single company. Ericsson, Nokia, Motorola, and Unwired Planet founded the WAP Forum in the summer of 1997 with the initial purpose of defining an industry-wide specification for developing applications over wireless communications networks. The WAP specifications define a set of protocols in application, session, transaction, security, and transport layers, which enable operators, manufacturers, and applications providers to meet the challenges in advanced wireless service differentiation and fast/flexible service creation.

**All solutions must be:**

1.       **interoperable**, i.e., allowing terminals and software from different vendors to communicate with networks from different providers

2.       **scalable**, i.e., protocols and services should scale with customer needs and number of customers

3.       **efficient,**i.e., provision of QoS suited to the characteristics of the wireless and mobile networks

4.       **reliable,**i.e., provision of a consistent and predictable platform for deploying services; and

5.       **Secure**, i.e., preservation of the integrity of user data, protection of devices and services from security problems.

**Uses of  WAP:**

In the past, wireless Internet access has been limited by the capabilities of handheld devices and wireless networks.

WAP utilizes Internet standards such as XML, user datagram protocol (UDP), and Internet protocol (IP). Many of the protocols are based on Internet standards such as hypertext transfer protocol (HTTP) and TLS but have been optimized for the unique constraints of the wireless environment: low bandwidth, high latency, and less connection stability.

Internet standards such as hypertext markup language (HTML), HTTP, TLS and transmission control protocol (TCP) are inefficient over mobile networks, requiring large amounts of mainly text-based data to be sent. Standard HTML content cannot be effectively displayed on the small-size screens of pocket-sized mobile phones and pagers.

WAP utilizes binary transmission for greater compression of data and is optimized for long latency and low bandwidth. WAP sessions cope with intermittent coverage and can operate over a wide variety of wireless transports.

WML and wireless markup language script (WML Script) are used to produce WAP content. They make optimum use of small displays, and navigation may be performed with one hand. WAP content is scalable from a two-line text display on a basic device to a full graphic screen on the latest smart phones and communicators.

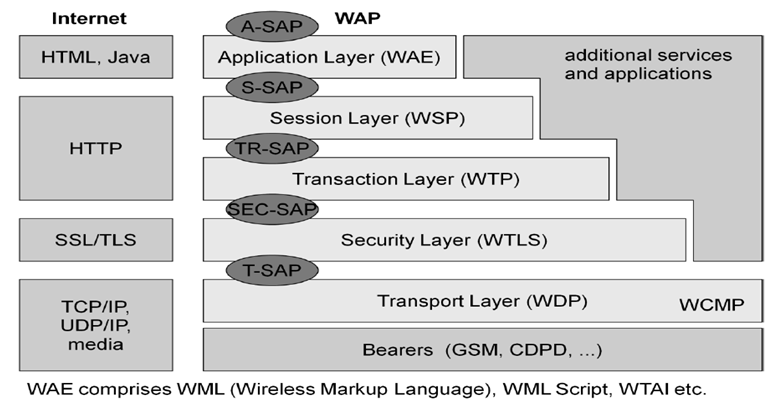
The lightweight WAP protocol stack is designed to minimize the required bandwidth and maximize the number of wireless network types that can deliver WAP content. Multiple networks will be targeted, with the additional aim of targeting multiple networks. These include global system for mobile communications (GSM) 900, 1,800, and 1,900 MHz; interim standard (IS)–136; digital European cordless communication (DECT); time-division multiple access (TDMA), personal communications service (PCS), FLEX, and code division multiple access (CDMA). All network technologies and bearers will also be supported, including short message service (SMS), USSD, circuit-switched cellular data (CSD), cellular digital packet data (CDPD), and general packet radio service (GPRS).

As WAP is based on a scalable layered architecture, each layer can develop independently of the others. This makes it possible to introduce new bearers or to use new transport protocols without major changes in the other layers.

WAP will provide multiple applications, for business and customer markets such as banking, corporate database access, and a messaging interface.

**WAP Architecture**

**Introduction:** The following figure gives an overview of the WAP architecture, its protocols and components, and compares this architecture with the typical internet architecture when using the World Wide Web. The basis for transmission of data is formed by different **bearer services**. WAP does not specify bearer services, but uses existing data services and will integrate further services. Examples are message services, such as short message service (SMS) of GSM, circuit-switched data, such as high-speed circuit switched data (HSCSD) in GSM, or packet switched data, such as general packet radio service (GPRS) in GSM. Many other bearers are supported, such as CDPD, IS-136, PHS.

**WDP**: The WAP datagram protocol (WDP) and the additional Wireless control message protocol (WCMP) is the transport layer that sends and receives messages via any available bearer network, including SMS, USSD, CSD, CDPD, IS–136 packet data, and GPRS. The **transport layer access point (T-SAP)**is the common interface to be used by higher layers independent of the underlying network.

**WTLS:**The next higher layer, the security layer with its wireless transport layer security protocol WTLS offers its service at the **security SAP (SEC-SAP)**. WTLS is based on transport layer security (TLS, formerly SSL, secure sockets layer). WTLS has been optimized for use in wireless networks with narrow-band channels. It can offer data integrity, privacy, authentication, and (some) denial-of-service protection.

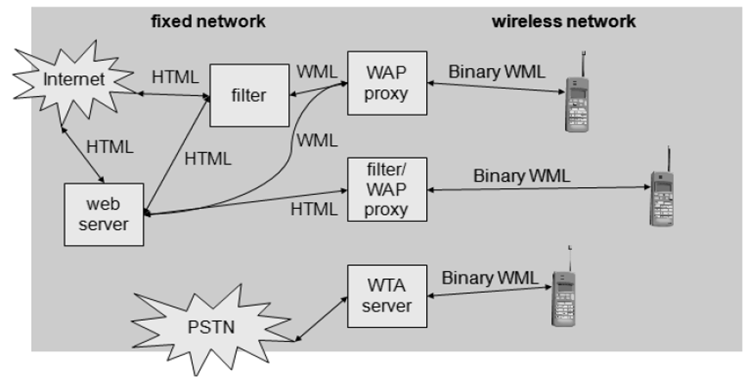
**WTP:**The WAP transaction protocol (WTP) layer provides transaction support, adding reliability to the datagram service provided by WDP at the **transaction SAP (TR-SAP)**.

**WSP:**The session layer with the wireless session protocol (WSP) currently offers two services at the session-SAP (S-SAP), one connection-oriented and one connectionless if used directly on top of WDP. A special service for browsing the web (WSP/B) has been defined that offers HTTP/1.1 functionality, long-lived session state, session suspend and resume, session migration and other features needed for wireless mobile access to the web.

**WAE:**The application layer with the wireless application environment (WAE) offers a framework for the integration of different www and mobile telephony applications

**Working of WAP**

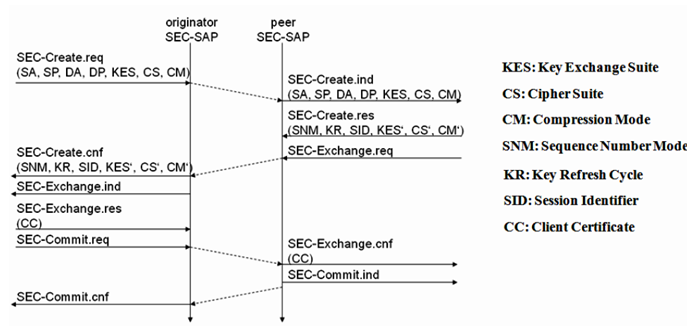
**Introduction:** WAP does not always force all applications to use the whole protocol architecture. Applications can use only a part of the architecture. For example, if an application does not require security but needs the reliable transport of data, it can **directly**use a service of the transaction layer. Simple applications can directly use WDP.

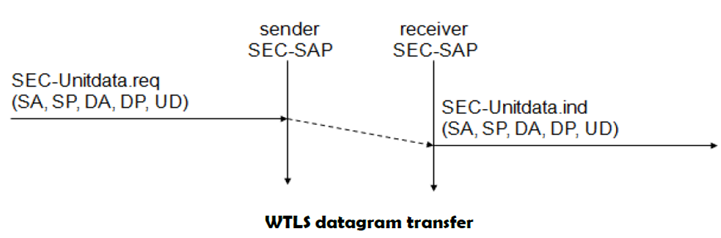
Different scenarios are possible for the integration of WAP components into existing wireless and fixed networks. On the left side, different fixed networks, such as the traditional internet and the public switched telephone network (PSTN), are shown. One cannot change protocols and services of these existing networks so several new elements will be implemented between these networks and the WAP-enabled wireless, mobile devices in a wireless network on the right-hand side.

**Wireless Transport Layer Security (WTLS)**

**Introduction:** WTLS can provide different levels of security (for privacy, data integrity, and authentication) and has been optimized for low bandwidth, high-delay bearer networks. WTLS takes into account the low processing power and very limited memory capacity of the mobile devices for cryptographic algorithms. WTLS supports datagram and connection-oriented transport layer protocols. WTLS took over many features and mechanisms from TLS, but it has an optimized handshaking between the peers.

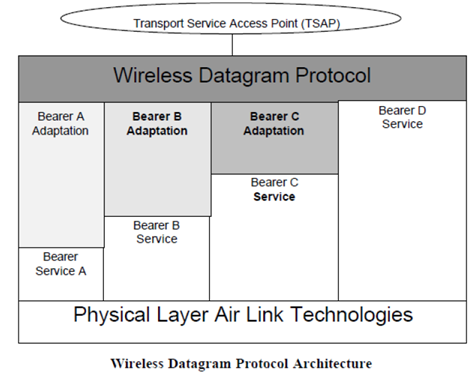
Before data can be exchanged via WTLS, a secure session has to be established. This session establishment consists of several steps: The following figure illustrates the sequence of service primitives needed for a so-called ‘full handshake’.

The first step is to initiate the session with the **SEC-Create**primitive. Parameters are **source address (SA),** **source port (SP)**of the originator, **destinationaddress (DA), destination port (DP)**of the peer. The originator proposes **a key exchange suite (KES)**(e.g., RSA, DH, ECC), a **cipher suite (CS)**(e.g., DES, IDEA ), and **a compression method (CM)**. The peer answers with parameters for the **sequence number mode (SNM)**, the **key refresh**cycle (**KR**) (i.e., how often keys are refreshed within this secure session), the **session identifier (SID)**(which is unique with each peer), and the selected **key exchange suite (KES’), cipher suite (CS’), compression method (CM’)**. The peer also issues a **SEC-Exchange primitive**. This indicates that the peer wishes to perform public-key authentication with the client, i.e., the peerrequests a **client certificate (CC)** from the originator. The first step of the secure session creation, the negotiation of the security parameters and suites, is indicated on the originator’s side, followed by the request for a certificate. The originator answers with its certificate and issues a **SEC-Commit.req** primitive. This primitive indicates that the handshake is completed for the originator’s side and that the originator now wants to switch into the newly negotiated connection state. The certificate is delivered to the peer side and the SEC-Commit is indicated. The WTLS layer of the peer sends back a confirmation to the originator. This concludes the full handshake for secure session setup.

After setting up a secure connection between two peers, user data can be exchanged. This is done using the simple **SEC-Unitdata** primitive as shown in above figure. **SEC-Unitdata** has exactly the same function as **T-DUnitdata** on the WDP layer, namely it transfers a datagram between a sender and a receiver. This data transfer is still unreliable, but is now secure. This shows that WTLS can be easily plugged into the protocol stack on top of WDP.

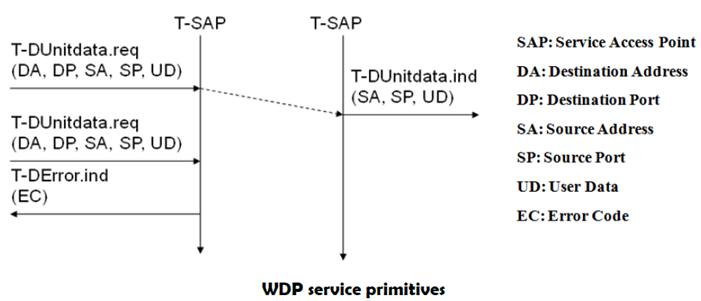
**Wireless Datagram Protocol (WDP)**

**Introduction: Wireless Datagram Protocol**defines the movement of information from receiver to the sender and resembles the User Datagram Protocol in the Internet protocol suite.

WDP offers a consistent service at the Transport Service Access Point to the upper layer protocol of WAP. This consistency of service allows for applications to operate transparently over different available bearer services. WDP can be mapped onto different bearers, with different characteristics. In order to optimize the protocol with respect to memory usage and radio transmission efficiency, the protocol performance over each bearer may vary.

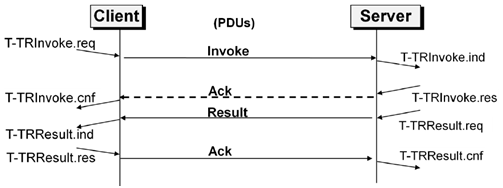
MukeshChintaAsst Prof, CSE, VNRVJIET **6**WDP offers **source**and **destination port numbers**used for multiplexing and demultiplexing of data respectively. The service primitive to send a datagram is **TDUnitdata. req**with the **destination address (DA), destination port (DP), Source address (SA), source port (SP)**, and **user data (UD)**as mandatory parameters. Destination and source address are unique addresses for the receiver and sender of the user data. These could be MSISDNs (i.e., a telephone number), IP addresses, or any other unique identifiers. The **T-DUnitdata.ind**service primitive indicates the reception of data. Here destination address and port are only optional parameters.

If a higher layer requests a service the WDP cannot fulfill, this error is indicated with the **T-DError.ind**service primitive. An **error code (EC)**is returned indicating the reason for the error to the higher layer. WDP is not allowed to use this primitive to indicate problems with the bearer service. It is only allowed to use the primitive to indicate local problems, such as a user data size that is too large. If any errors happen when WDP datagrams are sent from one WDP entity to another, the **wireless control message protocol (WCMP)**provides error handling mechanisms for WDP and should therefore be implemented. WCMP contains control messages that resemble the internet control message protocol messages and can also be used for diagnostic and informational purposes. WCMP can be used by WDP nodes and gateways to report errors.

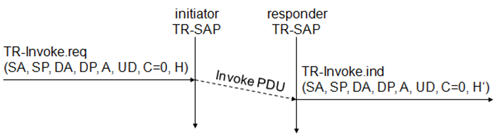
Typical WCMP messages are **destination unreachable**(route, port, address unreachable), **parameter problem**(errors in the packet header), **message too big, reassembly failure**, or **echo request/reply**. An additional **WDP management entity**supports WDP and provides information about changes in the environment, which may influence the correct operation of WDP.

**Wireless Transaction Protocol (WTP)**

**Introduction:** 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.**Class 0 provides unreliable message transfer without any result message. Classes 1 and 2 provide reliable message transfer, class 1 without, class 2 with, exactly one reliable result message (the typical request/response case). WTP achieves reliability using **duplicate removal, retransmission, acknowledgement**s and unique **transaction identifiers.**

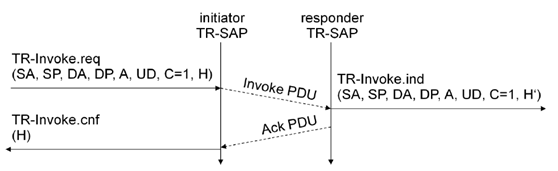
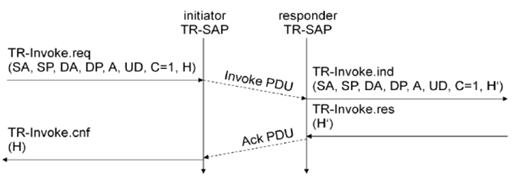
WTP allows for **asynchronous transactions, abort of transactions, concatenation of messages**, and can **report success or failure**of reliable messages (e.g., a server cannot handle the request). The three service primitives offered by WTP are **TR-Invoke**to initiate a new transaction, **TR-Result**to 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 special feature of WTP is its ability to provide a **user acknowledgement**or, alternatively, an **automatic acknowledgement**by the WTP entity.

***WTP Class 0 :***Class 0 offers an unreliable transaction service without a result message. The transaction is stateless and cannot be aborted. The service is requested with the **TR-Invoke.req**primitive as shown below. Parameters are same as in WDP.

Additionally, with the **A**flag, the user of this service can determine, if the responder WTP entity should generate an **acknowledgement**or if a user acknowledgement should be used. 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 uniquelyidentify the transaction and is an alias for the tuple (SA, SP, DA, DP), i.e., a socket pair, with only local significance. 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 initiator’s side, except for H’ which is now the local handle for the transaction on the responder’s side. WTP class 0 augments the transaction service with a simple datagram like service for occasional use by higher layers.

**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 below.

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

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.

**Wireless Session Protocol (WSP)**

**Introduction:** The **wireless session protocol (WSP)**has been designed to operate on top of the datagram service WDP or the transaction service WTP. WSP provides a shared state between a client and a server to optimize content transfer.

WSP offers the following general features needed for content exchange between cooperating clients and servers:

**Session management:**WSP introduces sessions that can be **established**from a client to a server and may be long lived. Sessions can also be **released**in an orderly manner. The capabilities of **suspending**and **resuming**a session are important to mobile applications.

**Capability negotiation:**Clients and servers can agree upon a common level of protocol functionality during session establishment. Example parameters to negotiate are maximum client SDU size, maximum outstanding requests, protocol options, and server SDU size.

**Content encoding:**WSP also defines the efficient binary encoding for the content it transfers. WSP offers content typing and composite objects, as explained for web browsing.

While WSP is a general-purpose session protocol, WAP has specified the **wireless session protocol/browsing (WSP/B)**which comprises protocols and services most suited for browsing-type applications, which offers the following features adapted to web browsing.

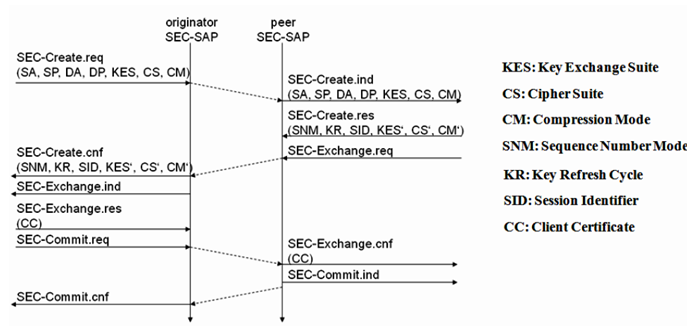
**HTTP/1.1 functionality:**WSP/B supports the functions HTTP/1.1 offers, such as extensible request/reply methods, composite objects, and content type negotiation.

**Exchange of session headers:**Client and server can exchange request/reply headers that remain constant over the lifetime of the session

**pull data transfer:**Pulling data from a server is the traditional mechanism of the web. This is also supported by WSP/B using the request/response mechanism from HTTP/1.1. Additionally, WSP/B supports three push mechanisms for data transfer: a confirmed data push within an existing session context, a non-confirmed data push within an existing session context, and a non-confirmed data push without an existing session context.

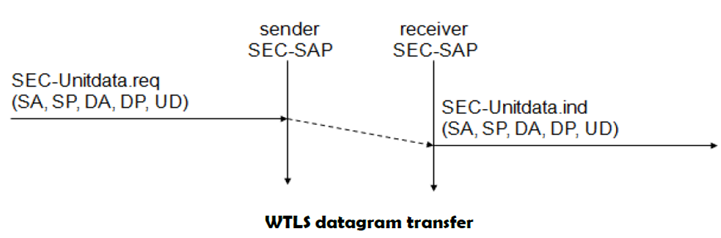
**Asynchronous requests:**Optionally, WSP/B supports a client that can send multiple requests to a server simultaneously. This improves efficiency for the requests and replies can now be coalesced into fewer messages.

**Wireless Transport Layer Security (WTLS)**

**Introduction:** WTLS can provide different levels of security (for privacy, data integrity, and authentication) and has been optimized for low bandwidth, high-delay bearer networks. WTLS takes into account the low processing power and very limited memory capacity of the mobile devices for cryptographic algorithms. WTLS supports datagram and connection-oriented transport layer protocols. WTLS took over many features and mechanisms from TLS, but it has an optimized handshaking between the peers.

Before data can be exchanged via WTLS, a secure session has to be established. This session establishment consists of several steps: The following figure illustrates the sequence of service primitives needed for a so-called ‘full handshake’.

The first step is to initiate the session with the **SEC-Create**primitive. Parameters are **source address (SA), source port (SP)**of the originator, **destinationaddress (DA), destination port (DP)**of the peer. The originator proposes **a key exchange suite (KES)**(e.g., RSA, DH, ECC), a **cipher suite (CS)**(e.g., DES, IDEA ), and **a compression method (CM)**. The peer answers with parameters for the **sequence number mode (SNM)**, the **key refresh**cycle (**KR**) (i.e., how often keys are refreshed within this secure session), the **session identifier (SID)**(which is unique with each peer), and the selected **key exchange suite (KES’), cipher suite (CS’), compression method (CM’)**. The peer also issues a **SEC-Exchange primitive**. This indicates that the peer wishes to perform public-key authentication with the client, i.e., the peerrequests a **client certificate (CC)** from the originator. The first step of the secure session creation, the negotiation of the security parameters and suites, is indicated on the originator’s side, followed by the request for a certificate. The originator answers with its certificate and issues a **SEC-Commit.req** primitive. This primitive indicates that the handshake is completed for the originator’s side and that the originator now wants to switch into the newly negotiated connection state. The certificate is delivered to the peer side and the SEC-Commit is indicated. The WTLS layer of the peer sends back a confirmation to the originator. This concludes the full handshake for secure session setup.

After setting up a secure connection between two peers, user data can be exchanged. This is done using the simple **SEC-Unitdata** primitive as shown in above figure. **SEC-Unitdata** has exactly the same function as **T-DUnitdata** on the WDP layer, namely it transfers a datagram between a sender and a receiver. This data transfer is still unreliable, but is now secure. This shows that WTLS can be easily plugged into the protocol stack on top of WDP.

**Wireless Markup Language (WML)**

**Introduction:** The **wireless markup language (WML)**is based on the standard HTML known from the www and on HDML. WML is specified as an XML document type. Several constraints of wireless handheld devices had to be taken into account, when designing WML.

WML follows a deck and card metaphor. A WML document is made up of multiple **cards**. Cards can be grouped together into a **deck**. A WML deck is similar to an HTML page, in that it is identified by a URL and is the unit of content transmission. A user navigates with the WML browser through a series of WML cards, reviews the contents, enters requested data, makes choices etc. The WML browser fetches decks as required from origin servers. Either these decks can be static files on the server or they can be dynamically generated.WML describes the intent of interaction in an abstract manner. The user agent on a handheld device has to decide how tobest present all elements of a card. This presentation depends much on the capabilities of the device.

**WML includes several basic features:**

**Text and images:**WML gives, as do other mark-up languages, hints how text and images can be presented to a user

**User interaction:**WML supports different elements for user input. Examples are: text entry controls for text or password entry, option selections or controls for task invocation. Navigation: As with HTML browsers, WML offers a history mechanism with navigation through the browsing history, hyperlinks and other intercard navigation elements.

**Context management:**WML allows for saving the state between different decks without server interaction, i.e., variable state can last longer than a single deck, and so state can be shared across different decks.

**WML script**

* WMLScript complements to WML and provides a general scripting capability in the WAP architecture. While all WML content is static (after loading on the client), WMLScript offers several capabilities not supported by WML:
* Validity check of user input: before user input is sent to a server, WMLScript can check the validity and save bandwidth and latency in case of an error.
* Access to device facilities: WMLScript offers functions to access hardware components and software functions of the device.
* Local user interaction: Without introducing round-trip delays, WMLScript can directly and locally interact with a user, show messages or prompt for input.
* Extensions to the device software: With the help of WMLScript a device can be configured and new functionality can be added even after deployment.

WMLScript is based on JavaScript, but adapted to the wireless environment. WMLScript is event-based, i.e., a script may be invoked in response to certain user or environment events. WMLScript also has full access to the state model of WML, i.e., WMLScript can set and read WML variables. WMLScript provides many features known from standard programming languages such as functions, expressions, or while, if, for, return etc. The WAP Forum has specified several standard libraries for WMLScript (WAP Forum, 2000i). These libraries provide access to the core functionality of a WAP client so they must be available in the client’s scripting environment. The six libraries defined are Lang, Float, String, URL, WML browser and Dialogs.