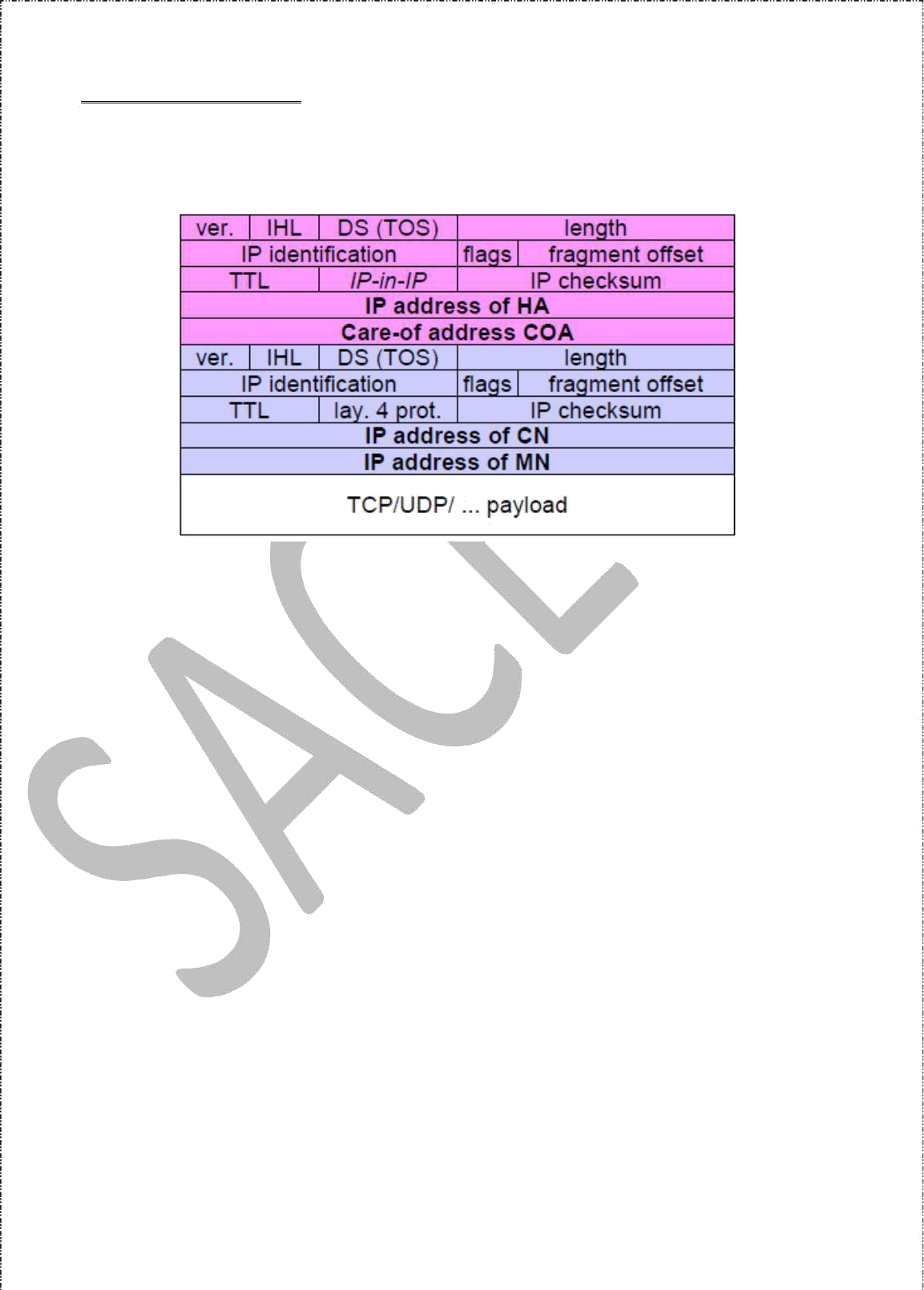
**UNIT 4**

**Tunnelling and encapsulation**

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.

****

11

**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 partof a new packet and sets the new IP header so that the packet is routed to the COA. The newheader 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.

Minimal encapsulation

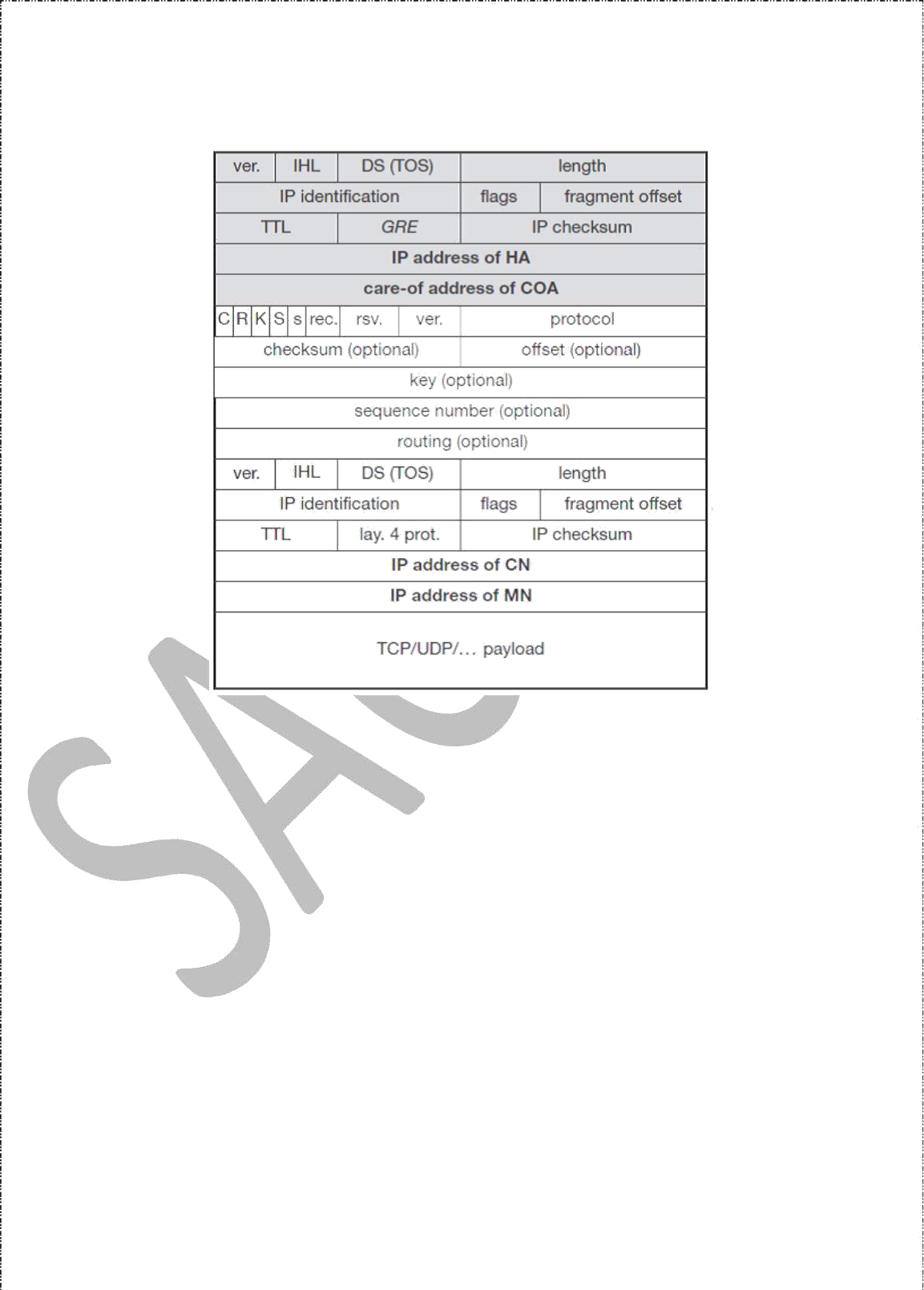
Minimal encapsulation (RFC 2004) as shown below is an optional encapsulation method for mobile IP which avoids repetitions of identical fields in IP-in-IP encapsulation. The tunnel entry point and endpoint are specified.

The field for the type of the following header contains the value 55 for the minimal encapsulation protocol. The inner header is different for minimal encapsulation. The type of the following protocol and the address of the MN are needed. If the **S** bit is set, the original sender address of the CN is included as omitting the source is quite often not an option. No field for fragmentation offset is left in the inner header and minimal encapsulation does not work with already fragmented packets.

Generic Routing Encapsulation

Unlike IP-in-IP and Minimal encapsulation which work only for IP packets, **Generic routing encapsulation** (GRE) allows the encapsulation of packets of one protocol suite into the payload portion of a packet of another protocol suite as shown below.

The packet of one protocol suite with the original packet header and data is taken and a new GRE header is prepended. Together this forms the new data part of the new packet. Finally, the header of the second protocol suite is put in front. The following figure shows the fields of a packet inside the tunnel between HA and COA using GRE as an encapsulation scheme according to RFC 1701. The outer header is the standard IP header with HA as source address and COA as destination address. The protocol type used in this outer IPheader is 47 for GRE.

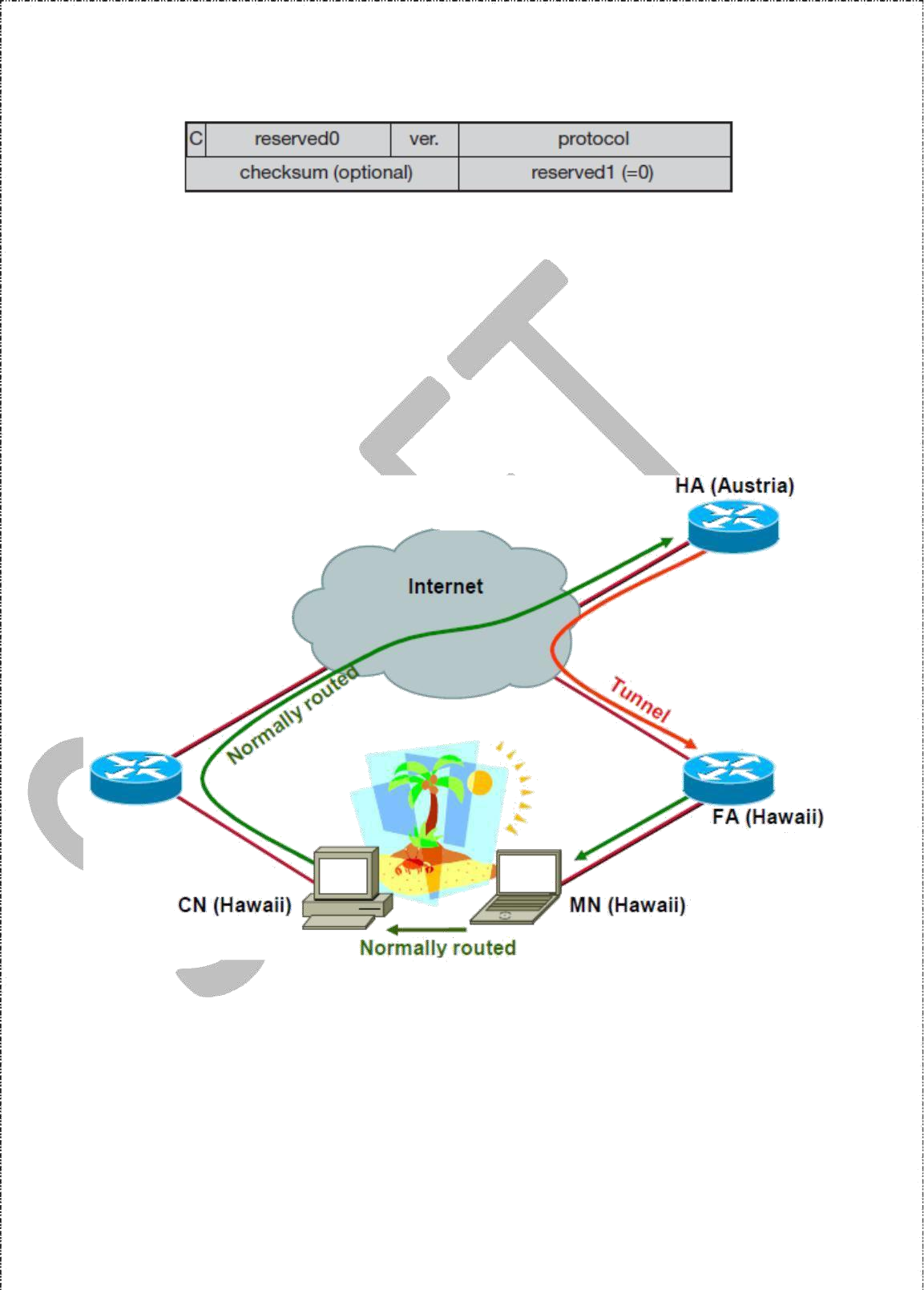


The GRE header starts with several flags indicating if certain fields are present or not. A minimal GRE header uses only 4 bytes. The **C** bit indicates if the checksum field is present and contains valid information. If **C** is set, the **checksum** field contains a valid IP checksum ofthe GRE header and the payload. The **R** bit indicates if the offset and routing fields are present and contain valid information. The **offset** represents the offset in bytes for the first source **routing** entry. The routing field, if present, has a variable length and contains fieldsfor source routing. GRE also offers a **key** field which may be used for authentication. If this field is present, the **K** bit is set. The sequence number bit **S** indicates if the **sequence** numberfield is present, if the s bit is set, strict source routing is used.

The **recursion control** field (rec.) is an important field that additionally distinguishes GRE from IP-in-IP and minimal encapsulation. This field represents a counter that shows the number of allowed recursive encapsulations. The default value of this field should be 0, thus allowing only one level of encapsulation. The following **reserved** fields must be zero and are ignored on reception. The **version** field contains 0 for the GRE version.

14

The following 2 byte **protocol** field represents the protocol of the packet following the GRE header. The standard header of the original packet follows with the source address of the correspondent node and the destination address of the mobile node.



A simplified header of GRE following RFC 2784 is shown below.

The field **C** indicates again if a checksum is present. The next 5 bits are set to zero, then 7 reserved bits follow. The **version** field contains the value zero. The **protocol** type, again, defines the protocol of the payload following RFC 3232. If the flag C is set, then **checksum** field and a field called reserved1 follows. The latter field is constant zero set to zero follow.

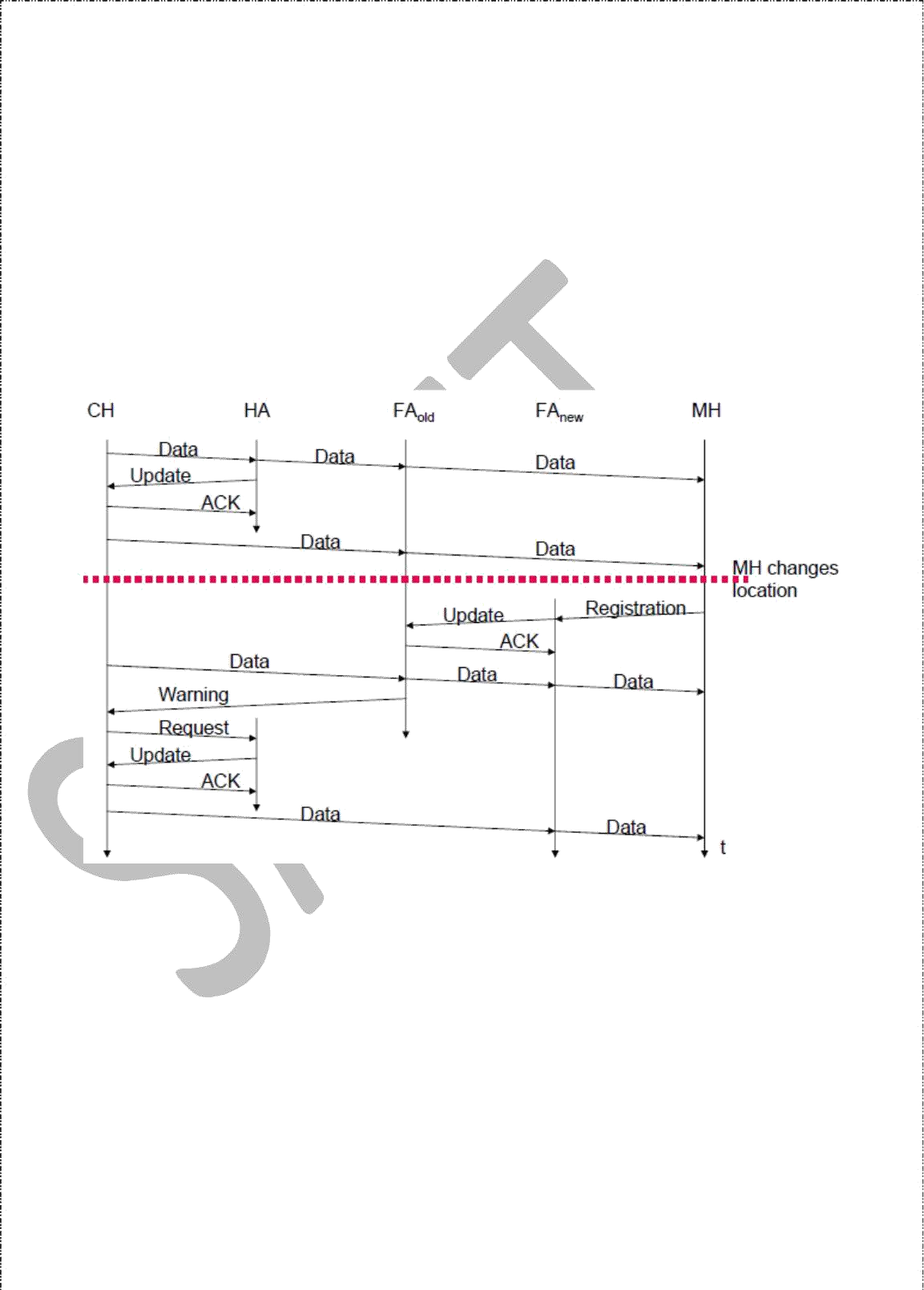
**Optimizations**

If a scenario occurs, where if the MN is in the same subnetwork as the node to which it is communicating and HA is on the other side of the world. It is called triangular routing problem as it causes unnecessary overheads for the network between CN and the HA.

A solution to this problem is to inform the CN of the current location of the MN. The CN can learn the location by caching it in a binding cache, which is a part of the routing table for the CN. HA informs the CN of the location. It needs four additional messages:

***Binding Request***: It is sent by the node that wants to know the current location of an MN to the HA. HA checks if it is allowed to reveal the location and then sends back a bindingupdate

***Binding update:*** It is sent by the HA to the CN revealing the current location of an MN. It contains the fixed IP address of the MN and the COA. This message can request an acknowledgement.



***Binding acknowledgement***: If requested, a node returns this acknowledgement after receiving a binding update message

***Binding warning:*** A node sends a binding warning if it decapsulates a packet for an MN,

but it is note the current FA of this MN. It contains MN’s home address and a target nodes address. The recipient can be the HA, so the HA now sends a binding update to thenode that obviously has a wrong COA for the MN.

The following figure shows how the four additional messages are used together if an MNchanges its FA.

The CN can request the current location from the HA. If allowed by the MN, the HA returns the COA of the MN via an update message. The CN acknowledges this update message and stores the mobility binding. Now the CN can send its data directly to the current foreign agent FAold. FAold forwards the packets to the MN. This scenario shows a COA located at an FA. Encapsulation of data for tunneling to the COA is now done by the CN, not the HA.

The MN might now change its location and register with a new foreign agent, FAnew. This registration is also forwarded to the HA to update its location database. Furthermore, FAnew informs FAold about the new registration of MN. MN’s registration message contains the address of FAold for this purpose. Passing this information is achieved via an update message, which is acknowledged by FAold.

Without the information provided by the new FA, the old FA would not get to know anything about the new location of MN. In this case, CN does not know anything about the new location, so it still tunnels its packets for MN to the old FA, FAold. This FA now notices packets with destination MN, but also knows that it is not the current FA of MN. FAold might now forward these packets to the new COA of MN which is FAnew in this example. This forwarding of packets is another optimization of the basic Mobile IP providing **smooth handovers**. Without this optimization, all packets in transit would be lost while the MN moves from one FA to another.

To tell CN that it has a stale binding cache, FAold sends, a binding warning message to CN. CN then requests a binding update. (The warning could also be directly sent to the HA triggering an update). The HA sends an update to inform the CN about the new location, which is acknowledged. Now CN can send its packets directly to FAnew, again avoiding triangular routing. Unfortunately, this optimization of mobile IP to avoid triangular routing causes several security problems

**Reverse Tunnelling**

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 1in 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 thetechnical 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 tunnelling is defined as an extension to mobile IP (per RFC 2344). It was designed backward compatible to mobile IP and defines topologically correct reverse tunnelling to handle the above stated problems.

**Reverse Tunnelling**

18

**Packet Forwarding Reverse Tunnel**

Reverse tunneling does not solve problems with *firewalls*, the reverse tunnel can be abused to circumvent securitymechanisms (tunnel hijacking) optimization of data paths, i.e. packets will be forwarded through the tunnel via the HAto a sender (double triangular routing)

**IPv6**

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"

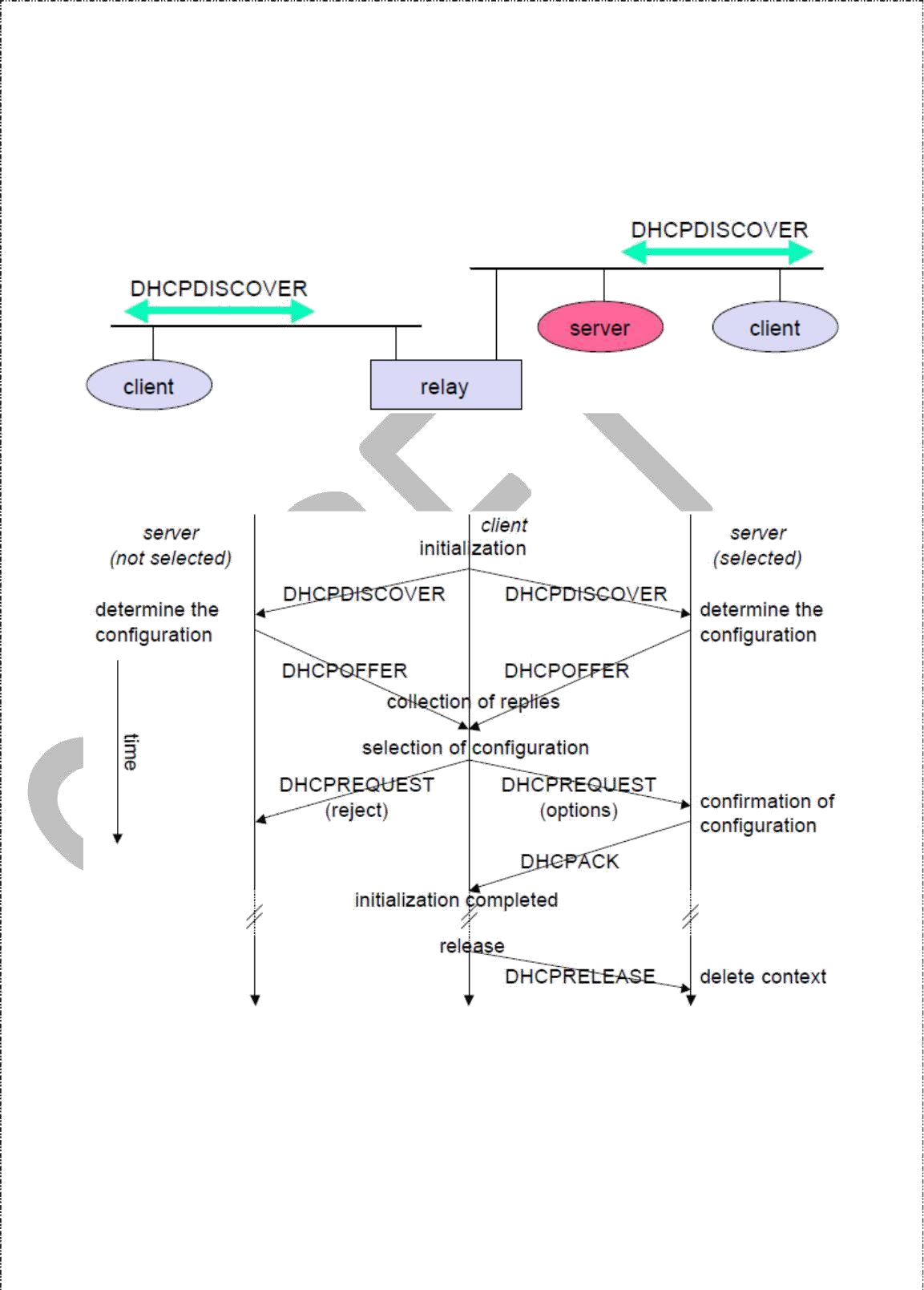
The IPv6 Neighbor Unreachability Detection assures symmetric reachability between the mobile node and its default router in the current location.

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 NeighborDiscovery 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 IPv6to 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)**

**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 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. 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…

#### Location Management

**Introduction**

Location management is an important problem in mobile computing since wireless mobile computers can change location while connected to the network. New strategies must be introduced to deal with the dynamic changes of a mobile computer`s network address. A few problems associated with mobility will be discussed in this article.

Mobility and Location Management

The ability to change locations while connected to the network creates a dynamic computing environment. This means that data which is static for stationary computing becomes dynamic for mobile computing. An example is that a stationary computer is permanently attached to the nearest server while mobile computers need a mechanism to determine which server to use.

As people move, their mobile computers will use different network addresses. The networking used today has to be changed to deal with dynamically changing addresses. If we, for example, look at how the Internet Protocol (IP) is designed for fixed computing, a host IP is bound with its network address so moving to a new location means that it needs a new IP name.

There are a few questions that must be answered when looking at a location management scheme. What happens when a mobile user changes location? Who should know about the change? How can you contact a mobile host? Should you search the whole network or does anyone know about the mobile users moves?

A few basic mechanisms to determine a mobile computer's current location has been discussed to modify the IP-based protocols. We will look at four of them in this article; broadcast, central services, home base and forwarding pointers.

**Selective Broadcast**

With this method a message is sent to all network cells asking the mobile computer to replywith its current address. This scheme may be too expensive in large networks.

However, if the mobile computer is known to be in one of a few cells a message is sent out to the selected cells. A disadvantage with selective broadcast is that it can only be used when we have enough information about current location.

**Central Services**

The current address for each mobile user is kept in a centralized database. When a mobile computer changes its address it also updates the central database by sending a message containing its new address.

**Home Bases**

With this method the location of a given mobile computer is known by a single server (MSS), often called the *Home Location Server*. The user is permanently registered under this server and it keeps track of where the mobile computer is. To send a message to a mobile user , thehome location server has to be contacted first to obtain the users' current address.

The main disadvantage with this scheme is that the way a message must travel may be much longer than the real distance. For example, two mobile computers, A and B, which are registered under two different home location servers in two different areas, may be currently in the same area. For A to contact B it has to first contact B's home location server which then contacts B. If A and B are likely to be in the same area, this scheme could be modified to first broadcast a message to all MSSs in that local area. If B is not currently located there a message is then sent to B's home location server. This scheme can also lead to low availability of information. The home location server maybe down or inaccessible which makes it impossible to track the requested mobile user.

**Forwarding Pointers**

This method is probably one of the fastest. Each time a mobile computer changes its address, a copy of the new address is added at the old location. The message sent is then forwarded along the chain of pointers until the mobile computer is reached. The pointer chain will be made longer every time the mobile computer changes location and this may lead to inefficient routing. To solve these pointers at the message forwarders can be updated to contain more recent addresses.

Even though this method is among the fastest it suffers from failure anywhere along the chain of pointers. Another problem is associated with deleting pointers which cannot be done before all message sources have been updated. The forwarding pointer method can be hard to implement. It does not fit standard networking models since it must have an active entity at the old address to receive and forward messages. Today’s network address is usually a passive entity.

There has not yet been done much work on comparing different locating and addressing schemes. The problem is difficult because it involves several dimensions. An issue introduced by these locating and addressing schemes is the cost of search. The less information the sender has about the mobile computer the more it will cost to search. This must also be considered when choosing for a location management scheme.

###### 

Mobile Transport Layer

**Traditional TCP**

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.

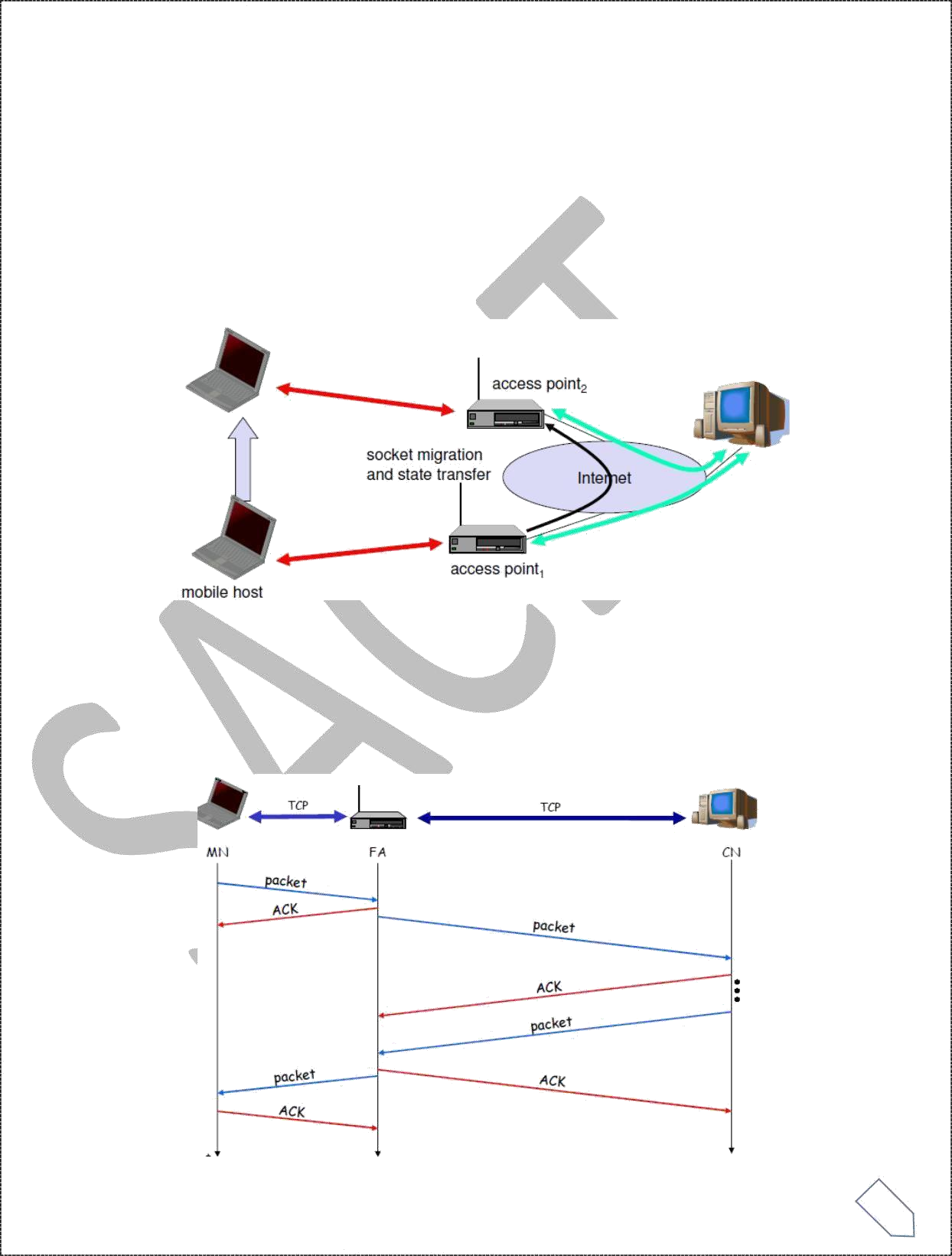
TCP uses a number of mechanisms to achieve high performance and avoid 'congestion collapse', where network performance can fall by several orders of magnitude. These mechanisms control the rate of data entering the network, keeping the data flow below a rate that would trigger collapse. There are several 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.

24

|  |
| --- |
| Congestion Control |
| 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 |

C

|  |  |
| --- | --- |
|  | |
| 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 effects 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

*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.

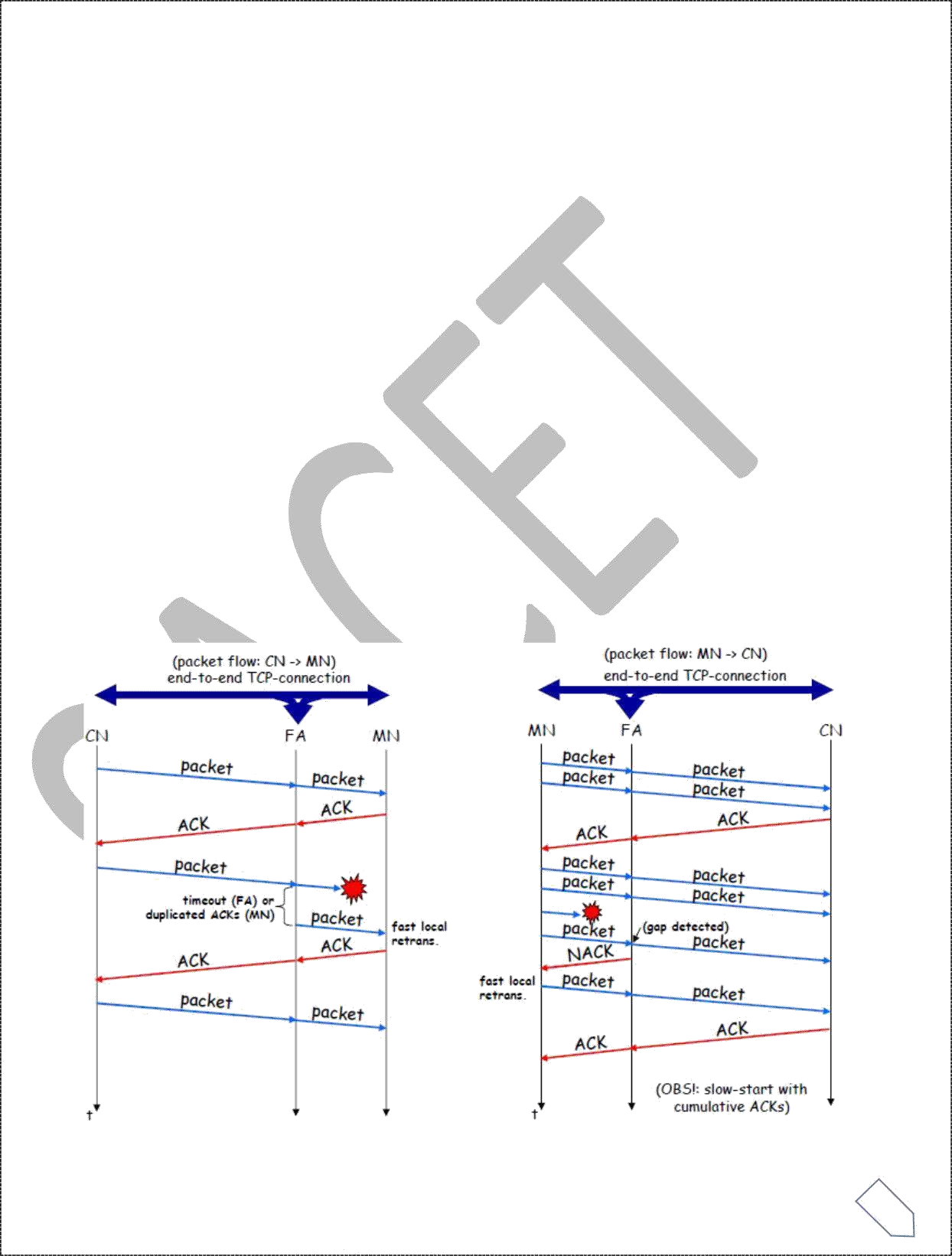
The foreign agent acts as a proxy and relays all data in both directions. If CH (correspondent host) sends a packet to the MH, the FA acknowledges it and forwards it to the MH. MH acknowledges on successful reception, but this is only used by the FA. If a packet is lost on the wireless link, CH doesn’t 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.

|  |
| --- |
| ***Socket and state migration after handover of a mobile host***  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:  Advantages of I-TCP |
|   No changes in the fixed network necessary, no changes for the hosts (TCP protocol) necessary, all current optimizations to TCP still work    Simple to control, mobile TCP is used only for one hop between, e.g., a foreign agent and mobile host   1. transmission errors on the wireless link do not propagate into the fixed network 2. therefore, a very fast retransmission of packets is possible, the short delay on the mobile hop s known     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.    It is easy to use different protocols for wired and wireless networks.  isadvantages of I-TCP |
|   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.    Higher latency possible:- due to buffering of data within the foreign agent and forwarding to a new foreign agent    Security issue:- The foreign agent must be a trusted entity  *Snooping TCP*  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. |

A

D

***Snooping TCP as a transparent TCP extension***

******

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.

|  |
| --- |
| ***Snooping TCP: Packet delivery***  Advantages of snooping TCP: |
|   The end-to-end TCP semantic is preserved.    Most of the enhancements are done in the foreign agent itself which keeps correspondent host unchanged.    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.    No problem arises if the new foreign agent uses the enhancement or not. If not, the approach automatically falls back to the standard solution.  isadvantages of snooping TCP |
|   Snooping TCP does not isolate the behavior of the wireless link as well as I -TCP. Transmission errors may propagate till CH.    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.    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.  *Mobile TCP*  Both I-TCP and Snooping TCP does not help much, if a mobile host gets disconnected. The **M-TCP (mobile TCP)** approach has the same goals as I-TCP and snooping TCP: to prevent the sender window from shrinking if bit errors or disconnection but not congestion cause current problems. M-TCP wants to improve overall throughput, to lower the delay, to maintain end-to- end semantics of TCP, and to provide a more efficient handover. Additionally, M-TCP is especially adapted to the problems arising from lengthy or frequent disconnections. M-TCP splits the TCP connection into two parts as I-TCP does. An unmodified TCP is used on the standard host-**supervisory host (SH)** connection, while an optimized TCP is used on the SH- MH connection.  The SH monitors all packets sent to the MH and ACKs returned from the MH. If the SH does not receive an ACK for some time, it assumes that the MH is disconnected. It then chokes the sender by setting the sender’s window size to 0. Setting the window size to 0 forces the sender to go into **persistent mode**, i.e., the state of the sender will not change no matter how long the receiver is disconnected. This means that the sender will not try to retransmit data. As soon as the SH (either the old SH or a new SH) detects connectivity again, it reopens the window of the sender to the old value. The sender can continue sending at full speed. |

A

D

|  |
| --- |
| This mechanism does not require changes to the sender’s TCP. The wireless side uses an adapted 8  Mobile Transport Layer Mobile Computing  TCP that can recover from packet loss much faster. This modified TCP does not use slow start, thus, M-TCP needs a **bandwidth manager** to implement fair sharing over the wireless link.  dvantages of M-TCP: |
|   It maintains the TCP end-to-end semantics. The SH does not send any ACK itself but forwards the ACKs from the MH.    If the MH is disconnected, it avoids useless retransmissions, slow starts or breaking connections by simply shrinking the sender’s window to 0.    As no buffering is done as in I-TCP, there is no need to forward buffers to a new SH. Lost packets will be automatically retransmitted to the SH.  isadvantages of M-TCP: |
|   As the SH does not act as proxy as in I-TCP, packet loss on the wireless link due to bit errors is propagated to the sender. M-TCP assumes low bit error rates, which is not always a valid assumption.    A modified TCP on the wireless link not only requires modifications to the MH protocol software but also new network elements like the bandwidth manager.  *Transmission/time-out freezing*  Often, MAC layer notices connection problems even before the connection is actually interrupted from a TCP point of view and also knows the real reason for the interruption. The MAC layer can inform the TCP layer of an upcoming loss of connection or that the current interruption is not caused by congestion. TCP can now stop sending and ‘freezes’ the current state of its congestion window and further timers. If the MAC layer notices the upcoming interruption early enough, both the mobile and correspondent host can be informed. With a fast interruption of the wireless link, additional mechanisms in the access point are needed to inform the correspondent host of the reason for interruption. Otherwise, the correspondent host goes into slow start assuming congestion and finally breaks the connection.  As soon as the MAC layer detects connectivity again, it signals TCP that it can resume operation at exactly the same point where it had been forced to stop. For TCP time simply does not advance, so no timers expire.  dvantages: |
|   It offers a way to resume TCP connections even after long interruptions of the connection.    It can be used together with encrypted data as it is independent of other TCP mechanisms such as sequence no or acknowledgements  isadvantages: |
|   Lots of changes have to be made in software of MH, CH and FA. |

A

D

A

D

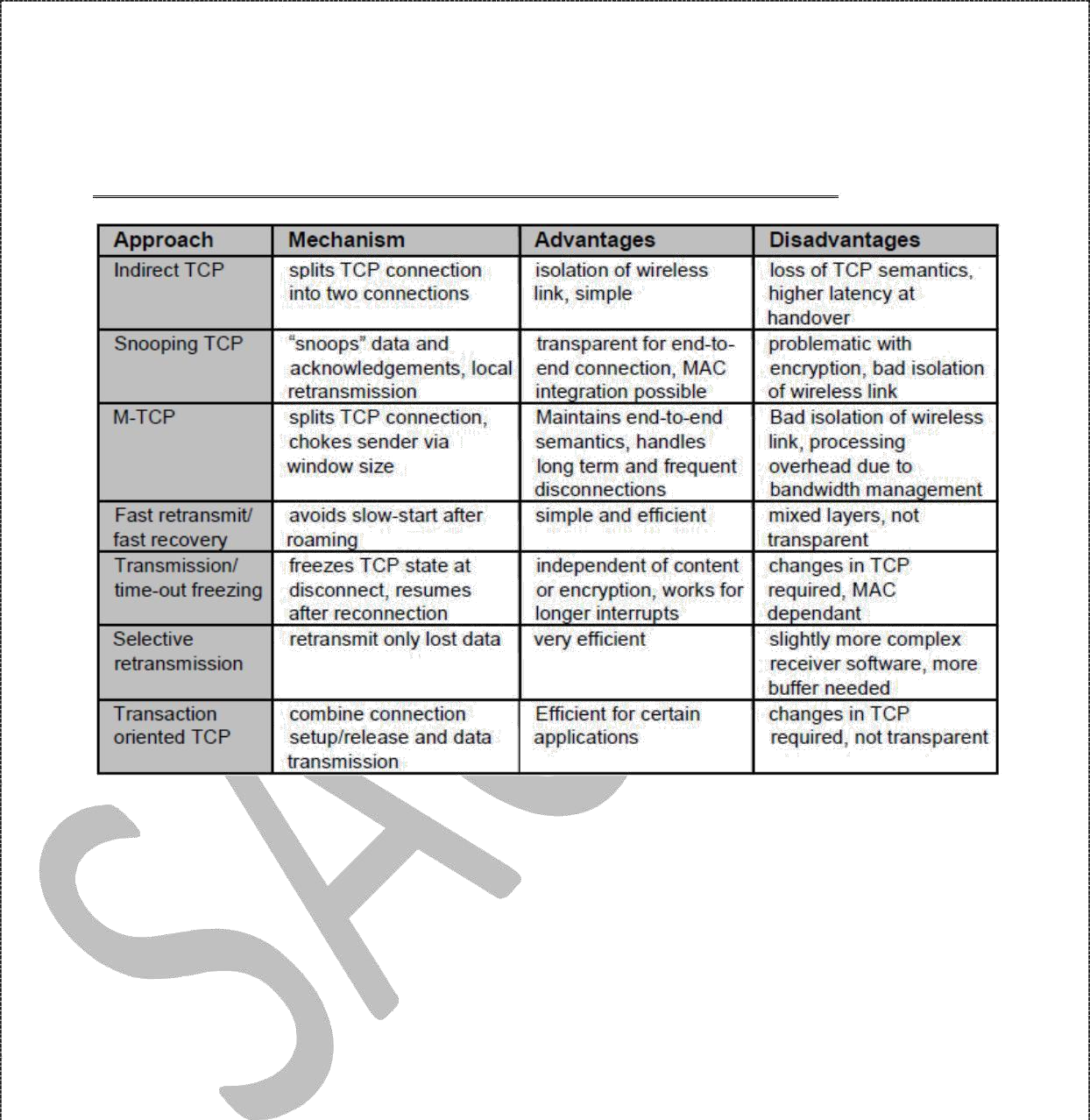
*Selective retransmission*

A very useful extension of TCP is the use of selective retransmission. TCP acknowledgements are cumulative, i.e., they acknowledge in-order receipt of packets up to a certain packet. A single acknowledgement confirms reception of all packets upto a certain packet. If a single packet is lost, the sender has to retransmit everything starting from the lost packet (go-back-n retransmission). This obviously wastes bandwidth, not just in the case of a mobile network, but for any network.

Using selective retransmission, TCP can indirectly request a selective retransmission of packets. The receiver can acknowledge single packets, not only trains of in-sequence packets. The sender can now determine precisely which packet is needed and can retransmit it. The **advantage** of this approach is obvious: a sender retransmits only the lost packets. This lowers bandwidth requirements and is extremely helpful in slow wireless links. The disadvantage is that a more complex software on the receiver side is needed. Also more buffer space is needed to resequence data and to wait for gaps to be filled.

*Transaction-oriented TCP*

Assume an application running on the mobile host that sends a short request to a server from time to time, which responds with a short message and it requires reliable TCP transport of the packets. For it to use normal TCP, it is inefficient because of the overhead involved. Standard TCP is made up of three phases: setup, data transfer and release. 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. So, for sending one data packet, TCP may need seven packets altogether. This kind of overhead is acceptable for long sessions in fixed networks, but is quite inefficient for short messages or sessions in wireless networks. This led to the development of transaction-oriented TCP (T/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. Disadvantage is that it requires changes in the software in mobile host** and all correspondent hosts. This solution does not hide mobility anymore. Also, T/TCP exhibits several security problems.