

**Multiple Access Protocol:**

If many users share a channel two or more users transmit data packets at the same time,

- There may be a collision.

To avoid conflicts, please use the appropriate access protocol

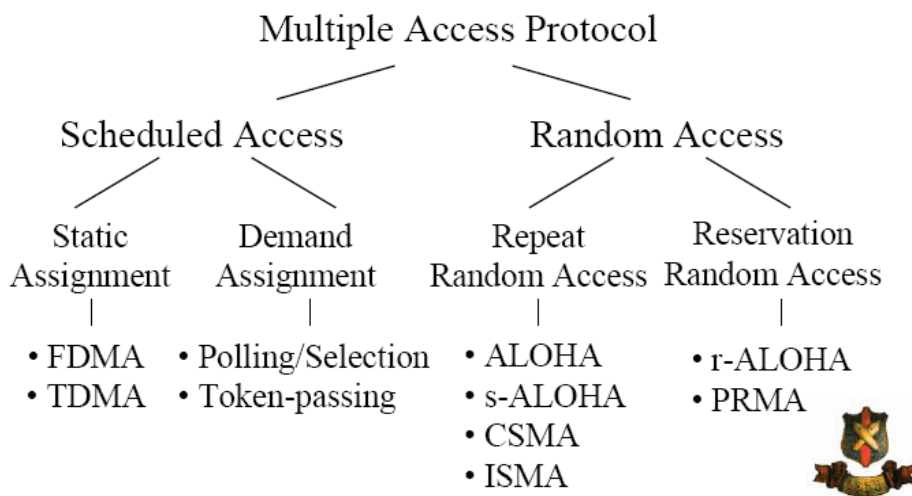
- Should choose.

**Channel:**

The signal space is divided and allocated to each user.

- Orthogonal (no interference between users)
- Allocation (e.g. frequency, time slot, code)-static allocation
- Simple but inefficient. -Demand allocation
- Efficient, but requires circuit control functions.

**Classification:**



**Scheduled Multiple Access Protocol:**


- **Static allocation**-whether each user is active, channel allocation is fixed to each user.  
TDMA: Time Division Multiple Access ,  
FDMA: Frequency Division Multiple Access.
- **Demand allocation**-channels allocated to each user on demand-central control:  
Polling/Selection-distributed control,  
Token passing protocol.

**Mobile Computing**  
**Mr. Saurabh Singh**

**Random Multiple Access Protocol:**

**Repeat Random Access-** The second broad type of access protocol is the so-called random access protocol. In the random access protocol, the sending node always sends at the full rate of the channel (i.e. R bps). When a conflict occurs, each node involved in the conflict will repeatedly send its frame until the frame passes without conflict. However, when a node encounters a conflict, it does not necessarily retransmit the frame immediately. Instead, it waits for a random delay before retransmitting the frame. Each node involved in the conflict chooses an independent random delay. Because the random delay is independently selected after the collision, one of the nodes may choose a delay sufficiently smaller than the delay of the other conflicting nodes, so that it can "eavesdrop" its frame until there is no collision in the channel. Dozens, if not hundreds, of random access protocols are described in the literature [Rom 1990, Bertsekas 1992]. In this section, we will introduce several of the most commonly used random access protocols-ALOHA [Abramson 1970, Abramson 1985] and Carrier Sense Multiple Access (CSMA) protocol [Kleinrock 1975]. Later, in Section 5.5, we will introduce the details of Ethernet [Metcalfe 1976], which is a popular and widely deployed CSMA protocol.

**Multiple Access Protocol Comparison:**

	Advantages	Disadvantages	Note
FDMA	Asynchronous between channels.	High frequency stability is required. Not flexible.	In the case of mobile communications, SCPC system is common.
TDMA	A demand of frequency stability is comparatively easy.	Burst transmission. Peak electric power is large. Synchronization among users is required.	As user bit rate and multiplex number increases, the performance degradation due to multipass fading becomes serious.
CDMA	Hand over is easy. The same frequency can be used also in adjacent cells. Anti-multipass fading and interference rejection properties are inherent.	Equipment is a little complicated. Highly precise transmitted electric power control is required.	Efficiency depends on various elements such as propagation characteristic, CN ratio, transmitted power control error, etc. 

**TCP:**

- Provides a connection between two applications.
- Within the connection, TCP can provide certain guarantees, such as in-order delivery or use of transmission technology for reliable data transmission.
- It has built-in mechanisms, such as operating in a "network friendly" manner. If TCP encounters packet loss, it will assume congestion within the network and reduce the transmission rate.

**Mobile Computing**  
**Mr. Saurabh Singh**

**Problems with TCP in Wireless Networks:**

- In wireless networks, the performance of TCP is generally lower than that of wired networks.
- TCP cannot distinguish the problems that usually occur in wireless networks from congestion. The congestion control algorithm in TCP is based on the following assumption: data is mainly lost due to congestion, and data loss caused by transmission errors is very small. Therefore, data loss is interpreted as a signal of congestion in the network.
- Even in wireless networks, data loss may have nothing to do with congestion, but data loss will still signal congestion to the sender.
- If the radio conditions are poor and the reliability provided by the link layer protocol is low, the TCP segment may be lost. After some retransmission attempts, the link layer protocol gave up and left further error recovery to TCP.
- Switching events can also cause data loss.
- TCP may also misunderstand the sudden increase in round-trip time as data loss. If the delay is long enough that the retransmission timer expires before the acknowledgement is received, TCP will misinterpret the delay as an indication of data loss due to congestion.
- The delayed data is retransmitted unnecessarily, and TCP enters a slow start. The highly variable round-trip time may also result in a larger RTO (retransmission timeout), because the RTO is based on both round-trip time estimates and changes in round-trip time. If the RTO is large, TCP's response to data loss will be slow.
- Changes in the round-trip time may also be caused by handover or competing traffic.
- The queuing of routers, base stations and other intermediate nodes may also cause long round trip times.
- Long round-trip times may result in low throughput and insufficient network utilization, because many round-trip times are required before the congestion window reaches the network capacity.
- TCP performance degradation, especially for short-lived streams (transmitting small amounts of data)
- Large bandwidth and delay changes make TCP incorrectly estimate the RTO value.
- False transmission due to incorrect RTO value will reduce TCP throughput.

**Factors Affecting TCP Performance:**

- Channel Losses:
- Low Bandwidth
- Signal Fading
- Movement Across Cells
- Channel Asymmetry

## Mobile Computing Mr. Saurabh Singh

### ■ Link Latency

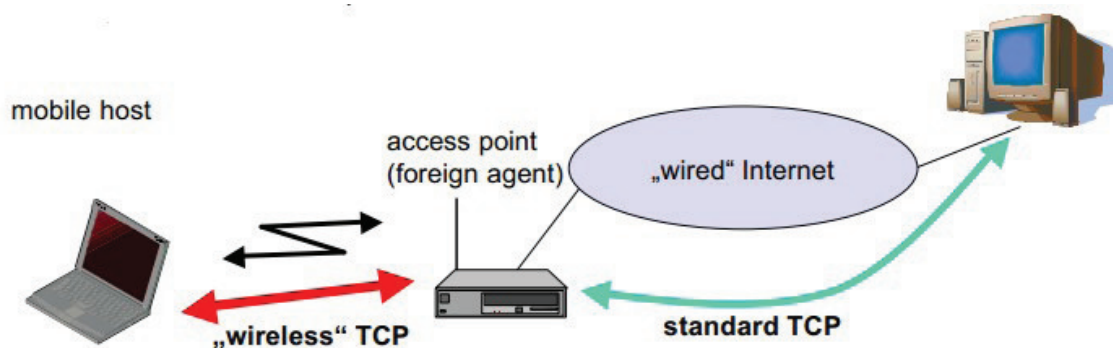
#### TCP Improvement:

- Indirect TCP (I-TCP)
- Listening TCP
- Mobile TCP
- Fast forwarding/fast recovery
- Transmission/timeout freeze
- Selective retransmission
- Transaction-oriented TCP (T / TCP)

#### Indirect TCP (I-TCP):

I-TCP divides the TCP connection into a fixed part and a wireless part. A mobile host connected via a wireless link is an access point to the wired Internet where the host is located. Wired computers and access points use standard TCP. Now, instead of the mobile host, the access point terminates the standard TCP connection and acts as a proxy. This means that the access point is now regarded as a mobile host of a fixed host and a fixed host of a mobile host. The black and white access point and mobile host are a special TCP suitable for wireless links. A handover mechanism is proposed to handle the situation when the wireless host moves between different cells.

- **Consequences**-The consequence of using I-TCP is that TCP Acks is not end-to-end, thus violating TCP's end-to-end semantics.
- The host in the fixed part of the network will not notice the characteristics of the wireless part.)



#### Snooping TCP:

Interception (Snooping) means secretly investigating things that have nothing to do with you. One of the disadvantages of I-TCP is that it splits a single TCP connection into two TCP connections. This loses the original end-to-end TCP semantics. The following TCP enhancements (i.e., snooping TCP) work completely transparently, so end-to-end semantics are not lost. The main function of the enhanced function is to buffer data packets close to the mobile host to perform fast local retransmission in case of packet loss.

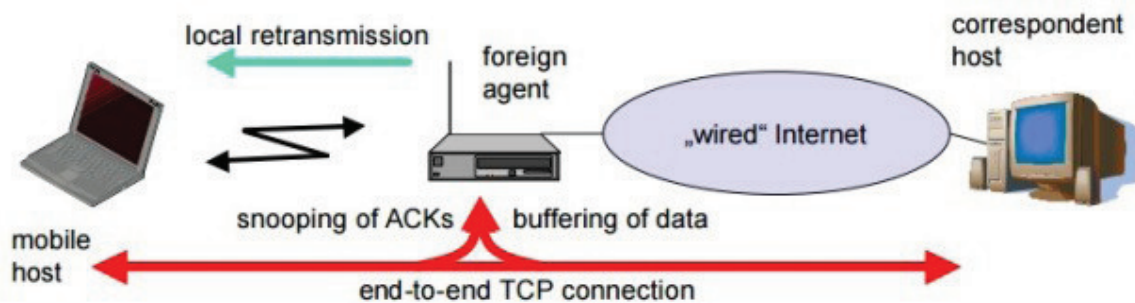
## Mobile Computing Mr. Saurabh Singh

In this method, the foreign agent buffers all packets with the target mobile host and "snoops" the packet flow in both directions to identify confirmations.

The process of transmitting data to the mobile host while listening to TCP is as follows:

- The external FA proxy buffers the data until it receives the confirmation from the mobile host. The FA detects the packet loss through repeated ACKs or timeouts, and then performs a faster retransmission compared to the corresponding host. This is for a fixed n/ w is transparent.

This will make the communication host believe that the mobile host has received the data, and the end-to-end semantics will be violated if the foreign agent fails the data transmission. Therefore, the working principle of the mobile host with the target communication host is as follows, that is, -FA listens to the flow Detect gaps in the TCP sequence number. FA detects the data packet loss on the wireless link through the serial number, and FA directly answers MH with Nack. MH can now retransmit data within a short delay.



### Advancement and disability:

- Detect lost packets
- Perform NACK
- The reordering of data packets is automatically completed by TCP on the corresponding host.

### TCP Improvement:

**Mobile TCP (M-TCP):**-It has the same goals as I-TCP and listening TCP, that is, it also uses a method based on split connections, but tries to preserve end-to-end semantics. M-TCP uses a three-level hierarchy-

At the lowest level, the mobile host communicates with the mobile support station in each cell; in turn, it is controlled by the "master host" (SH); the SH connects to the wired n/w and serves as the point of disconnection. There is one on the SH TCP client.

The TCP client receives the segment from the TCP sender and passes it to the M-TCP client to send it to the wireless device. Therefore, standard TCP is used for the sender and SH, and M-TCP is used for SH and wireless devices. M-TCP is designed to quickly recover from wireless loss caused by disconnection and eliminate serial timeouts. The TCP on SH does not acknowledge the received data packet until the wireless device confirms it.

**Mobile Computing**  
**Mr. Saurabh Singh**

**Fast forwarding/fast recovery:-**

Migrating to a new external agent may cause packet loss or timeout on the mobile host or corresponding host. Although there is no congestion, TCP will end the congestion and start to slow down. The idea is to artificially force the mobile host and the corresponding host to perform fast retransmissions. Once the mobile host uses mobile IP to register on the new foreign agent, it will start sending repeated confirmations to the corresponding host. The suggestion is to send three copies. This forces the corresponding host to enter the fast retransmission mode.

**Fast Retransmit:**

Fast retransmission is an enhancement to TCP, which reduces the time the sender waits before retransmitting the lost segment.

The TCP sender uses a timer to identify missing segments. If the confirmation of a specific segment is not received within a specified time (a function of the round-trip delay time), the sender will assume that the segment has been lost in the network and will retransmit the segment.

Repeated confirmation is the basis of the fast retransmission mechanism. The working principle of this mechanism is as follows: After receiving a data packet (for example, the sequence number is 1), the receiver sends the confirmation by adding 1 to the sequence number (i.e., sequence number 2). The data packet number received by the receiver is 1, and it expects the data packet number from the sender to be 2. Assume that three subsequent data packets have been lost. At the same time, the receiver receives data packet numbers 5 and 6. After receiving packet number 5, the receiver sends another confirmation, but still only for sequence number 2. When the receiver receives the data packet number 6, it sends another confirmation value of 2. In this way, the sender receives multiple acknowledgments with the same sequence number 2, which is called duplicate acknowledgment.

The fast retransmission enhancement works as follows: If the TCP sender receives a specified number of confirmations, it is usually set to three repeated confirmations with the same confirmation number (i.e., a total of four confirmations with the same confirmation number), then send The party can be confident enough that the segment with the next higher sequence number will be deleted and will not arrive out of order. Then, the sender will resend the supposedly dropped packet and wait for it to time out.

**Fast Recovery:**

After fast retransmission, it seems to be a lost segment to avoid congestion, but it cannot start slowly. This is a fast recovery algorithm. It is an improvement that can achieve high throughput under moderate congestion, especially for large windows.

The reason why slow start is not performed in this case is that receiving repeated ACKs can tell TCP, not just the packet loss. Since the receiver can only generate repeated ACKs when

## Mobile Computing

### Mr. Saurabh Singh

another segment is received, the segment has left the network and is in the receiver's buffer. In other words, there is still data flow between the two ends, and TCP does not want to suddenly reduce the flow by entering a slow start.

Fast retransmission and fast recovery algorithms are usually implemented in the following way.

1. When receiving the third consecutive repeated ACK, set  $ssthresh$  to half of the current congestion window  $cwnd$ , but not less than two segments. Retransmit the lost segment. Set  $cwnd$  to  $ssthresh$  plus 3 times the segment size. This will increase the congestion window by the number of segments that have left the network and have been cached on the other end (3).
2. every time another repeated ACK arrives, increase  $cwnd$  by the segment size. This will increase the congestion window of other segments leaving the network. If the new  $cwnd$  value allows, the packet is sent.

3. When the next ACK to confirm new data arrives, set  $cwnd$  to  $ssthresh$  (the value set in step 1). This ACK should be a confirmation of the retransmission from step 1, that is, a round trip time after the retransmission. In addition, this ACK should acknowledge all intermediate segments sent between the lost packet and the reception of the first duplicate ACK. This step is to avoid congestion, because TCP is reduced to half the rate at which packets are lost.

The fast retransmission algorithm first appeared in the 4.3BSD Tahoe release, followed by a slow start. The fast recovery algorithm appeared in the 4.3BSD Reno version.

**Transmission/timeout freeze:-**The MAC layer can notify the TCP layer that the connection is about to be lost, or that the current interruption is not caused by congestion. TCP can now stop sending and "freeze" the current state of its congestion window and other timers. If the MAC layer notices the upcoming interruption early enough, it can notify the mobile node and the communication node. Once the MAC layer detects connectivity again, it will signal to TCP that it can resume operation at the exact same point where it was forced to stop.

**Selective retransmission-(TCP SACK):-**A very useful extension of TCP is the use of selective retransmission. TCP ACKs are cumulative, that is, they acknowledge the orderly reception of packets until a certain packet. If a single packet is lost, the sender must retransmit everything starting from the lost packet (go-back-n retransmission). This is obviously a waste of bandwidth. Using RFC 2018, TCP can indirectly request selective retransmission of data packets. The receiver can not only confirm the packets in the sequence, but also a single packet. Now, the sender can determine exactly which packet is needed and can resend it.

Advanced sender only retransmits lost packets

**Transaction-oriented TCP (T / TCP):-**Using TCP requires multiple data packets to be sent over a wireless link. First, TCP uses a three-way handshake to establish a connection. Usually, at least one additional data packet is required to send a request, and three other data packets are needed to close the connection through the three-way handshake. In order

**Mobile Computing**  
**Mr. Saurabh Singh**

to reduce this overhead, it led to the development of T/TCP. T/TCP can combine data packets used to establish and release connections with user data packets. This can reduce the number. The number of packets is reduced to two instead of seven.

**Adv.:** Reduce the overhead of standard TCP for connection establishment and connection release.