Carleton University
Canada's Capital University

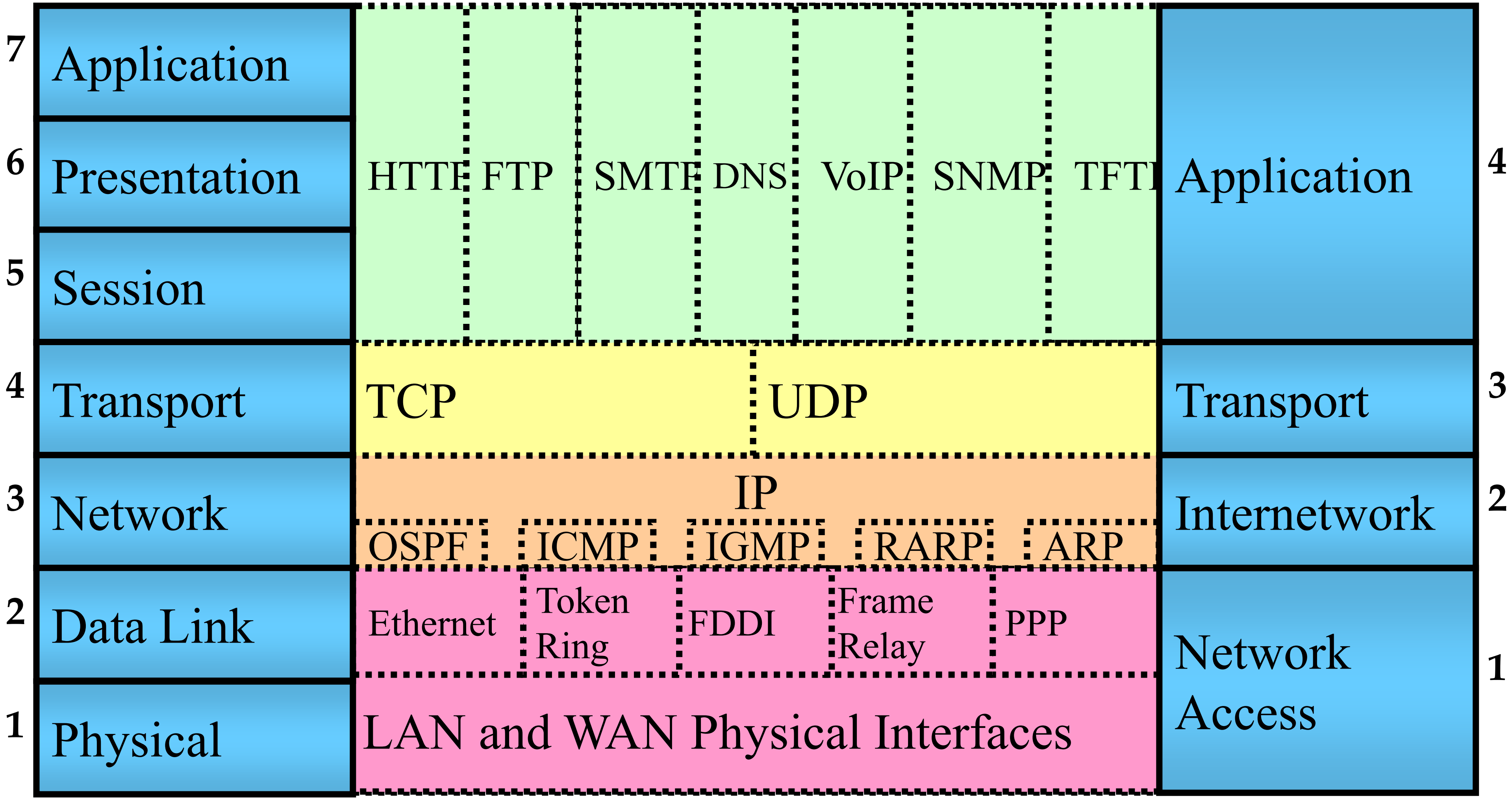
Seminar Presentations

- Idea: give students a chance to practice presentation skills
  - Perfection is not required
  - I already selected possible topics and papers
  - I will help/provide feedback on presentations, if desired
- Send me an e-mail with your topic selection to
  - thomas.kunz@sei.ecnu.edu.cn
  - tkunz@sce.carleton.ca

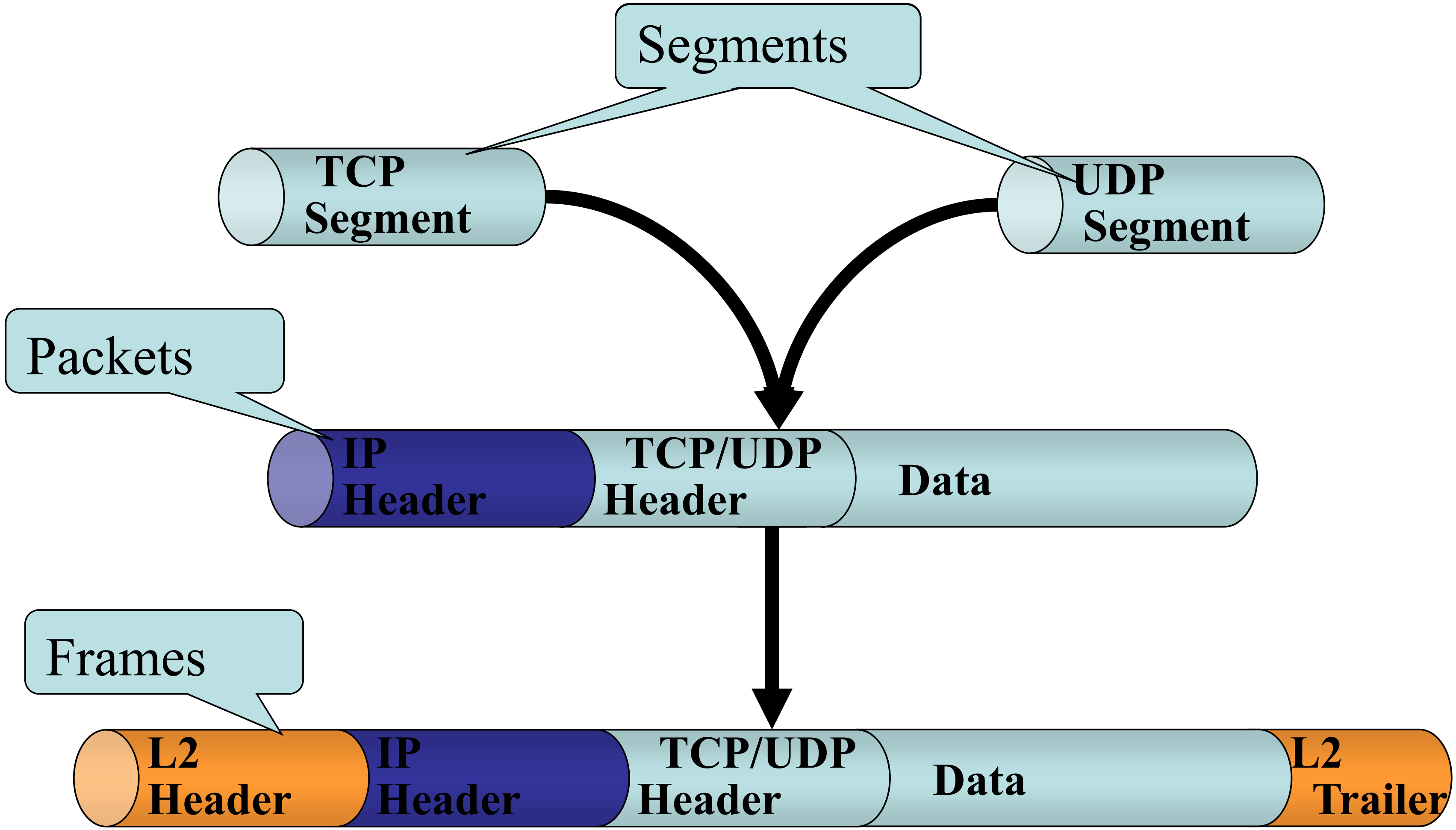
IP and TCP in Wireless Networks

Carleton
UNIVERSITY
Canada's Capital University

TCP/IP Protocol Suite: The Hourglas



Relationship Through the Layers



TCP Header

TCP Header Facts

8 bits = 1 octet

4 octets = 1 word

Maximum Option field size is 40 octets

Minimum header size 20 octets = 5 words

Maximum header size 60 octets = 15 words

IP Header

Header Facts

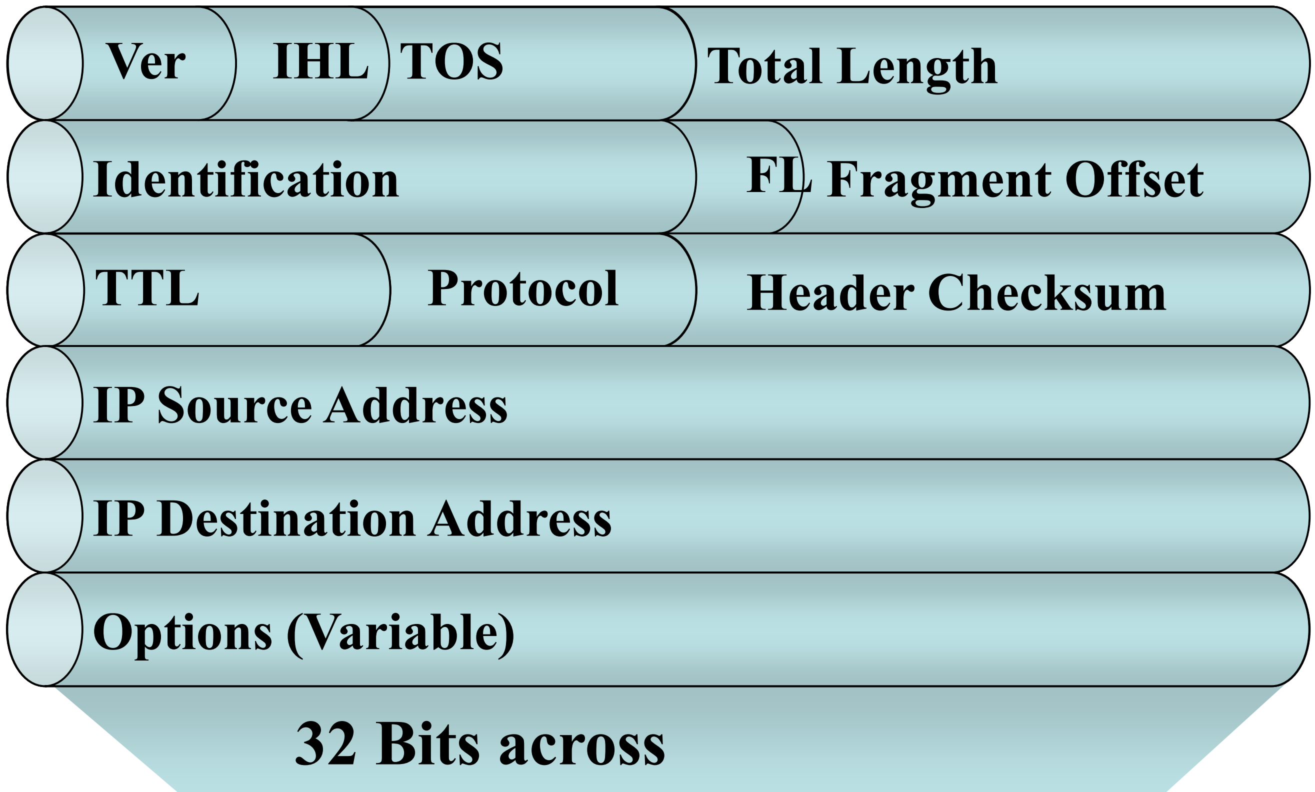
8 bits = 1 octet

4 octets = 1 word

Maximum Option field size is 40 octets

Minimum header size 20 octets = 5 words

Maximum header size 60 octets = 15 words

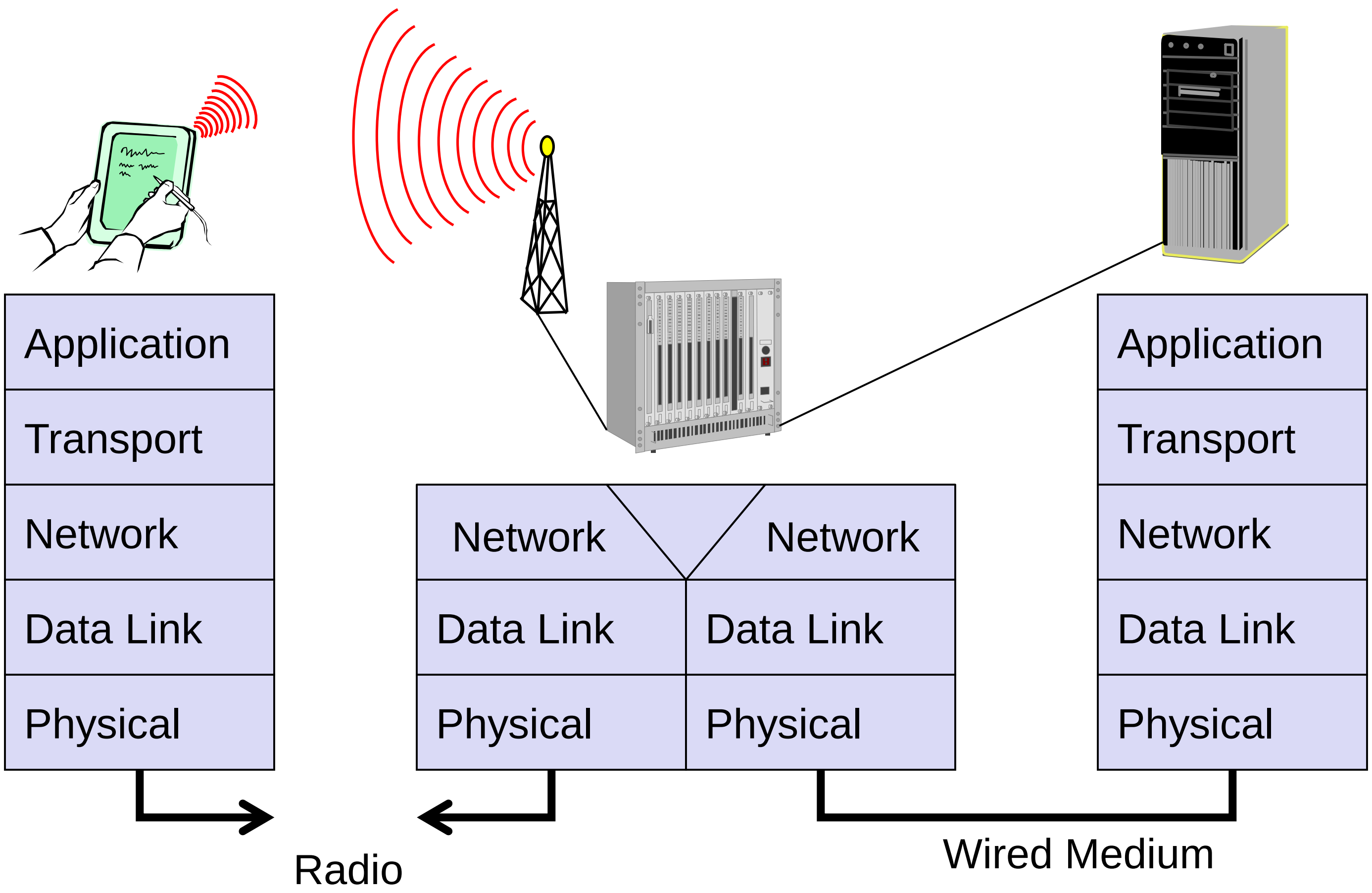


Wireless Access: Started "for real" in 1990s

- Originally: computers/communication endpoints are stationary, network links are wired (Ethernet, dial-up modems, DSL, etc.)
- Starting in the 1970s: cellular networks for voice (AMPS)
  - Could use for data, using modems, but expensive and low data rates
  - Some cellular data packet network technologies such as ARDIS or CDPD, but not widely used
  - GSM/2nd generation cellular networks started in 1980s, but no support for packet data service originals
- That all changed in 1990s
  - GSM was extended with GPRS (GSM Packet Radio Service)
  - Future cellular networks (3G, 4G, etc.) see packet data service as core service
  - Pure data networks: IEEE 802.11 was finally standardized and became popular really fast

➔ Running TCP/IP over such networks ran into problems

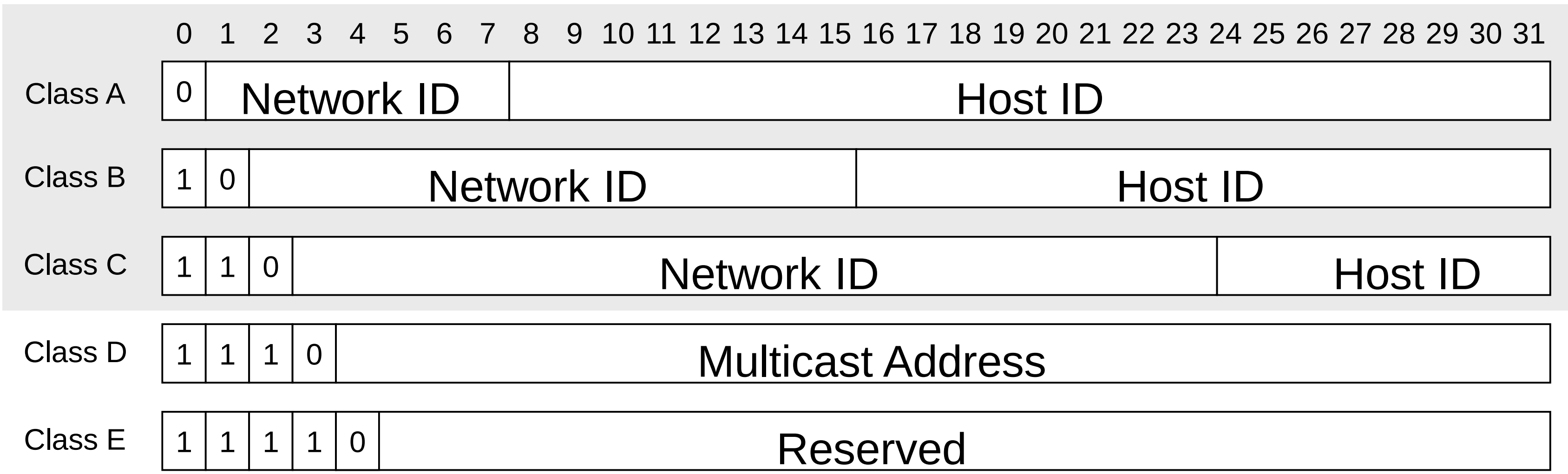
Model: Wireless Access (first or last hop)



Issues

- IP layer: mobility breaks implicit assumption about what an IP address means
  - Mobility Support in IP: A Survey of Related Protocols
- Transport layer: TCP's design assumes that packet loss is due to congestion and congestion only
  - Transmission Control Protocol (TCP) in Wireless Networks: Issues, Approaches, and Challenges
- Solutions: keep protocols and protocol stack, make specific changes
  - i.e., do NOT replace TCP with a different transport layer protocol

IP Addresses



class	# of Nets	# of hosts
A	127	16,777,214
B	16,384	66,534
C	1,097,152	254

Example

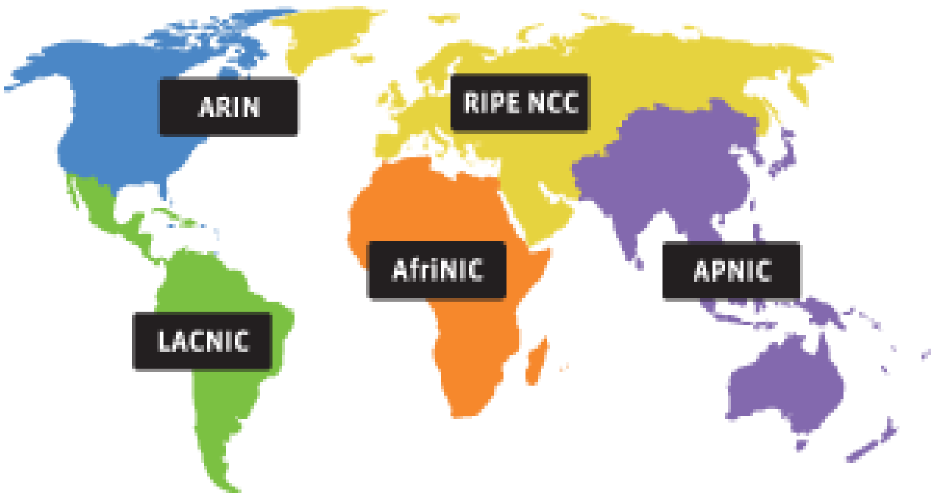
- Seminar website runs on PC `kunz-pc.sce.carleton.ca`
- Corresponding IP address (can be looked up using dnslookup): `134.117.63.134`
- What does that tell us:
  - PC exists in Carleton U's network (Carleton has a class B address allocation), which determines prefix `134.117`
  - Carleton uses subnetting with subnet mask `255.255.255.0`, so PC exists in subnet `63`
- Physical address: `00-25-64-8C-3E-04`
  - How to translate from IP address to PHY address?

IP Addresses and Physical Addresses

- Map IP addresses into physical addresses
  - destination host
  - next hop router
- Techniques
  - encode physical address in host part of IP address
  - table-based
- ARP
  - table of IP to physical address bindings
  - broadcast request if IP address not in table
  - target machine responds with its physical address
  - table entries are discarded if not refreshed

IPv4 Address Allocation State

- IANA handed out last unallocated address blocks to regional registrars February 1, 2011
- http://en.wikipedia.org/wiki/IPv4_address_exhaustion: Three of the five RIRs have exhausted allocation of all the blocks they have not reserved for IPv6 transition; this occurred for the Asia-Pacific on 15 April 2011, for Europe on 14 September 2012, and for Latin America and the Caribbean on 10 June 2014.
- Some global IP addresses will probably never be used



IPv6

- Extended addressing capabilities: 128-bit address field and other improvements.
- Simplified header format: Some fields of IPv4 are dropped or turned into options
- Improved support for extensions and options: flexibility and ability to introduce new options
- Flow labeling
- Authentication and privacy, mobility support part of core protocol, not added later (IPsec, MobileIP)
- What is not changing: IP addresses hierarchical, embed notion of "where" a device is in the network

IP and Mobility

- Routing
  - based on IP destination address, network prefix (e.g. 134.117) determines physical subnet
  - change of physical subnet implies change of IP address to have a topological correct address (standard IP) or needs special entries in the routing tables
- Specific routes to end-systems?
  - change of all routing table entries to forward packets to the right destination
  - does not scale with the number of mobile hosts and frequent changes in the location, security problems
- Changing the IP address?
  - adjust the host IP address depending on the current location
  - almost impossible to find a mobile system, DNS updates take long time
  - TCP connections break, security problems

IP and Mobility

- Partial solutions exist
  - WiFi: roaming handles mobility within an IP subnet
  - CDPD: has its own mobility management
  - GSM/GPRS: handles mobility within and among networks
- No truly global solution (on the Internet scale)
  - Support (seamless) movement from WiFi to GPRS to Office Ethernet

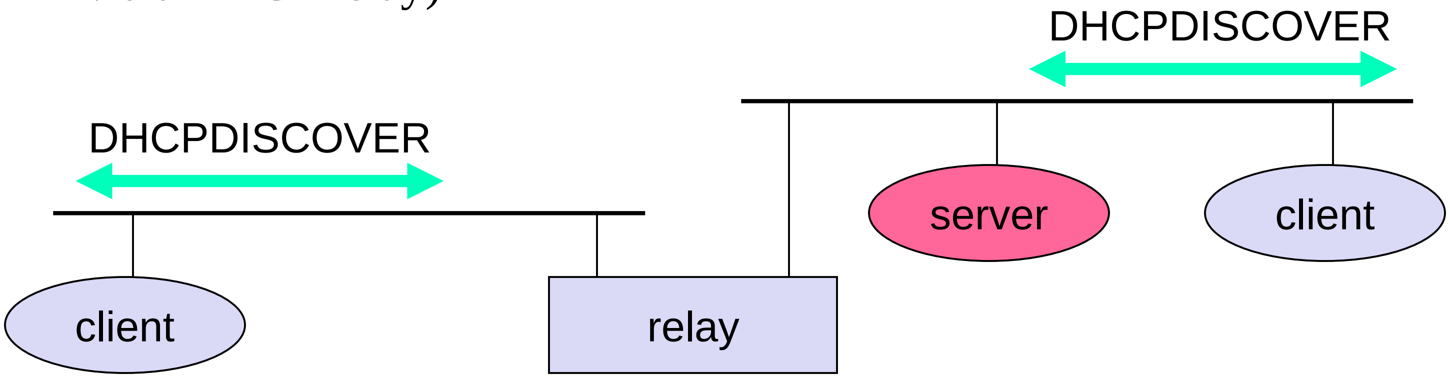
Where to Solve Mobility Problem

- What model of mobility
  - "nomadic clients": DHCP or similar solutions enough
  - Truly mobile: need to keep connections alive WHILE moving: Mobile IP
  - Offering services: need to be known under constant/well-known address (P2P systems, M2M communication, mobile servers): Mobile IP
- Where in the protocol stack
  - IP is common glue, solve it once and for all at IP layer
  - BUT: may be in contradiction to end-to-end argument
  - Other solutions/proposals exits, such as TCP connection migration, SIP, etc/

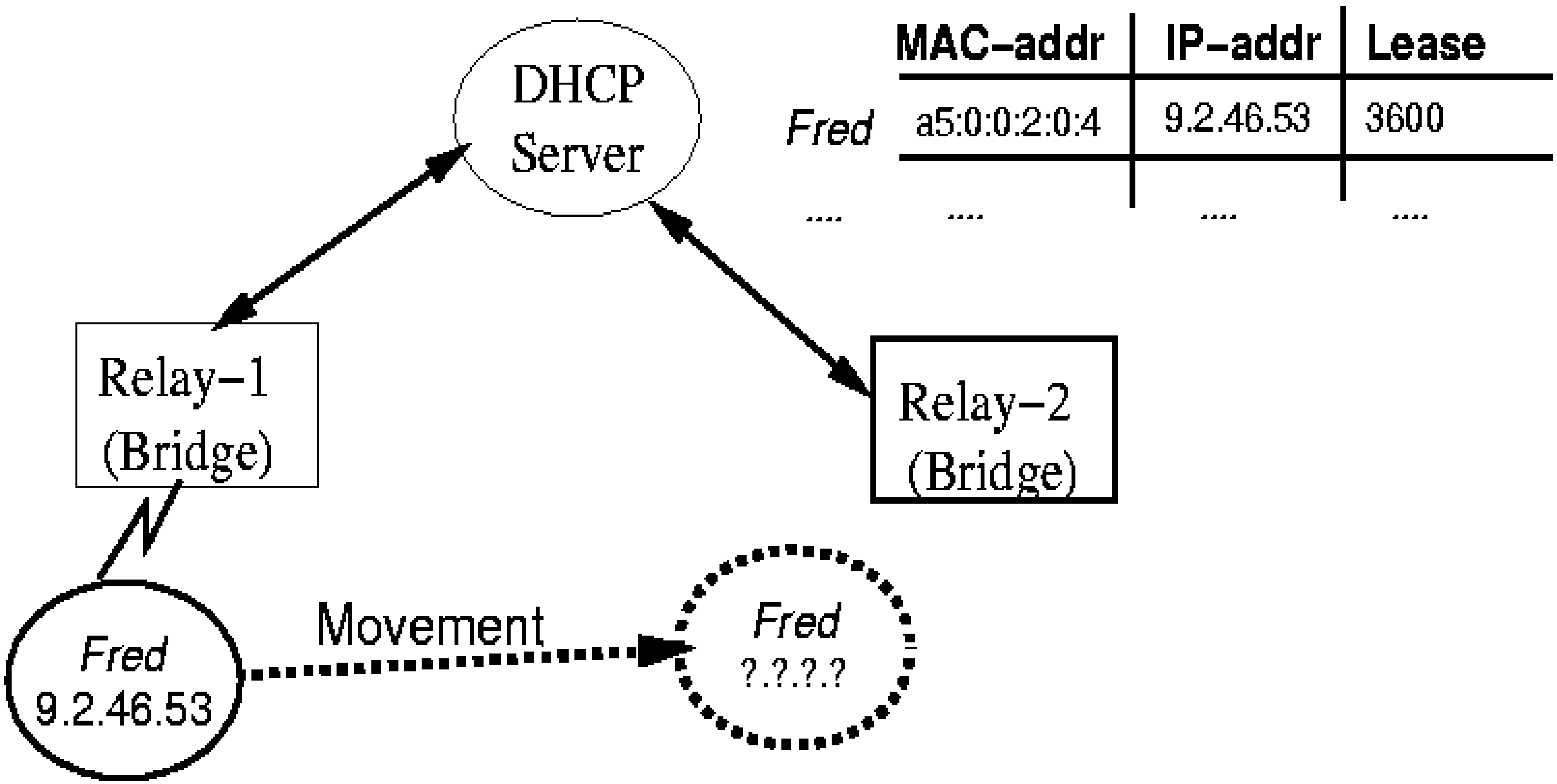
Carleton UNIVERSITY
Canada's Capital University

DHCP: Dynamic Host Configuration Protocol

- Application
  - simplification of installation and maintenance of networked computers
  - supplies systems with all necessary information, such as IP address, DNS server address, domain name, subnet mask, default router etc.
  - enables automatic integration of systems into an Intranet or the Internet, can be used to acquire a COA for Mobile IP
- Client/Server-Model
  - the client sends via a MAC broadcast a request to the DHCP server (might be via a DHCP relay)



DHCP: Portability



DHCP: Portability

- Initiate connectivity to Internet by DHCP request
- Once initial IP address has been obtained, start all servers/demons, etc.
- Suppose host detects movement: re-issue new DHCP request to validate current IP address
  - if okay, proceed
  - if new address needed, we have a problem
    - new IP address will not work with existing connections
      - shut down and reboot machine
    - since no other node knows new IP address, MH has to initiate all requests
      - alternative: allow DNS updates, which takes time and introduces new security problem

Requirements for Mobile IP
RFC 3344 (updated by RFC 4721), was: 3220, was: 2002

- Transparency
  - mobile end-systems keep their IP address
  - continuation of communication after interruption of link possible
  - point of connection to the fixed network can be changed
- Compatibility
  - support of the same layer 2 protocols as IP
  - no changes to current end-systems and routers required → no changes to core IP protocol
  - mobile end-systems can communicate with fixed systems
- Security
  - authentication of all registration messages
- Efficiency and scalability
  - only little additional messages to the mobile system required (connection typically via a low bandwidth radio link)
  - world-wide support of a large number of mobile systems in the whole Internet

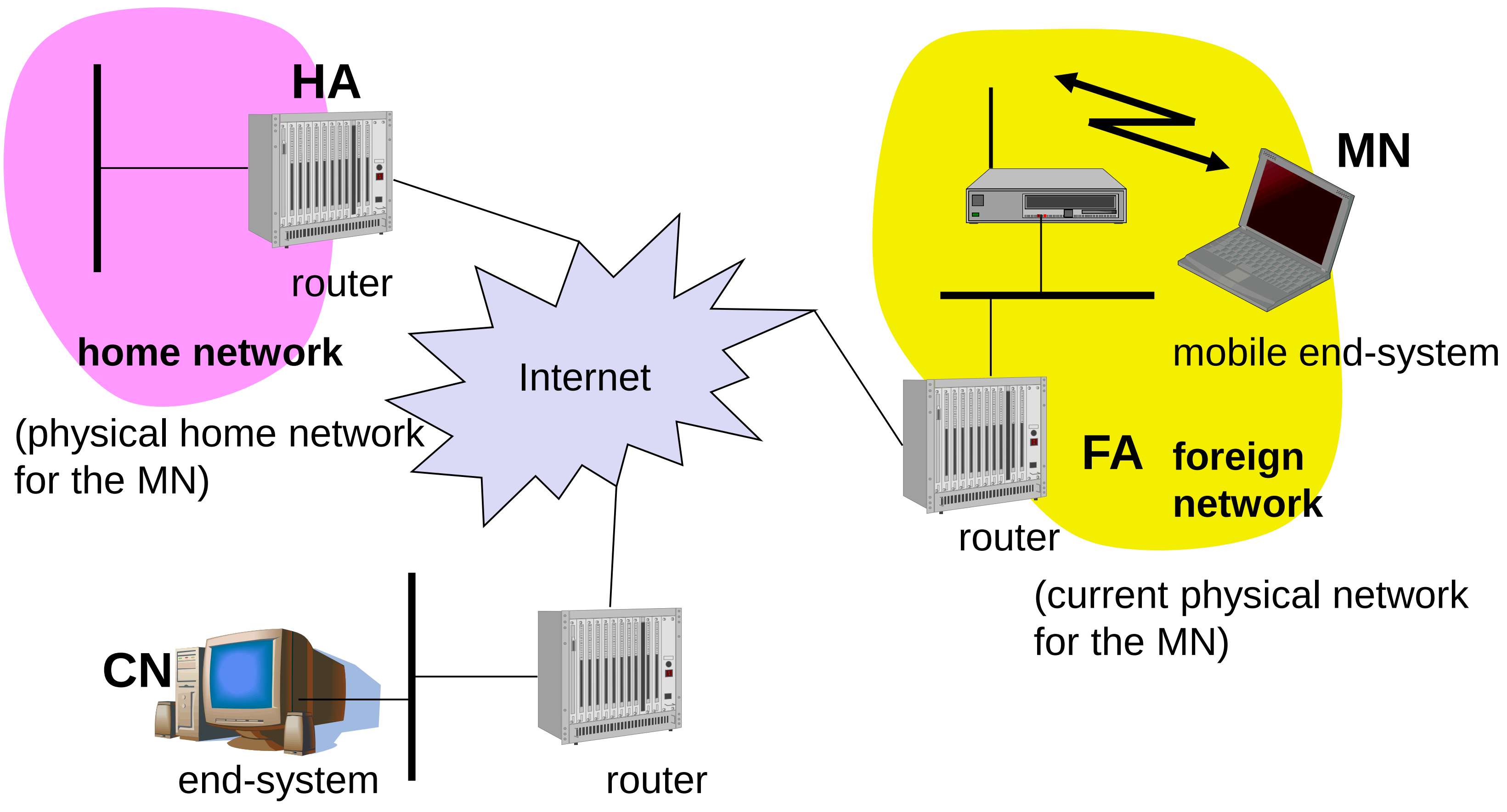
Terminology

- Mobile Node (MN)
  - system (node) that can change the point of connection to the network without changing its IP address
- Home Agent (HA)
  - system in the home network of the MN, typically a router
  - registers the location of the MN, tunnels IP datagrams to the COA
- Foreign Agent (FA)
  - system in the current foreign network of the MN, typically a router
  - forwards the tunneled datagrams to the MN, typically also the default router for the MN
- Care-of Address (COA)
  - address of the current tunnel end-point for the MN (at FA or MN)
  - actual location of the MN from an IP point of view
  - can be chosen, e.g., via DHCP
- Correspondent Node (CN)
  - communication partner

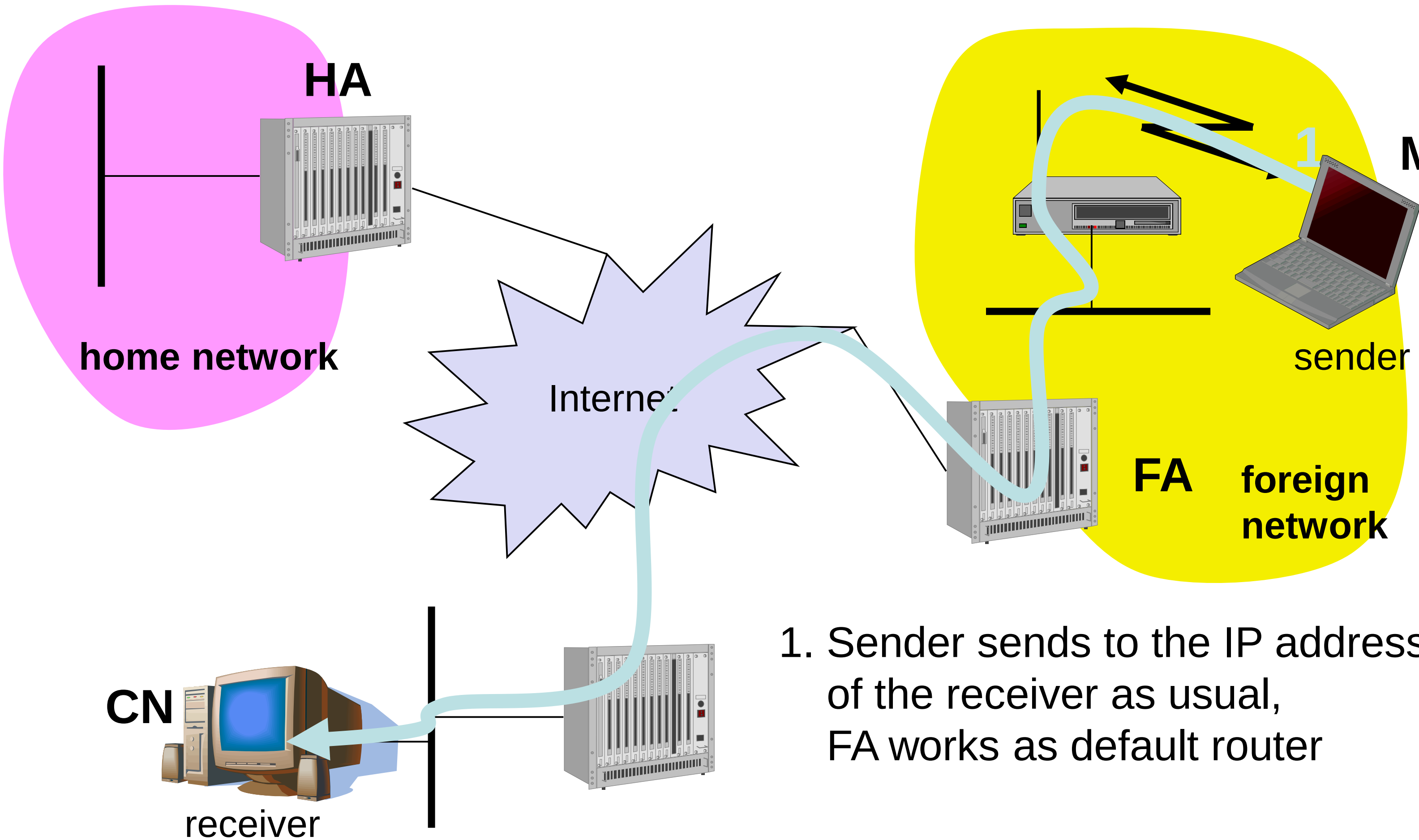
Properties of Care-of Address

- A care-of address is an IP address associated with mobile node that is visiting a foreign link:
  - A care-of address is specific to the foreign link currently being visited by a mobile node
  - Generally changes every time the mobile node moves from one foreign link to another
  - No Mobile IP-specific procedures are needed in order to deliver packets to a care-of address
  - Is used as the exit-point of a tunnel from the home agent toward the mobile node
  - Is never returned by DNS when another node looks up the mobile node's hostname

Example Network

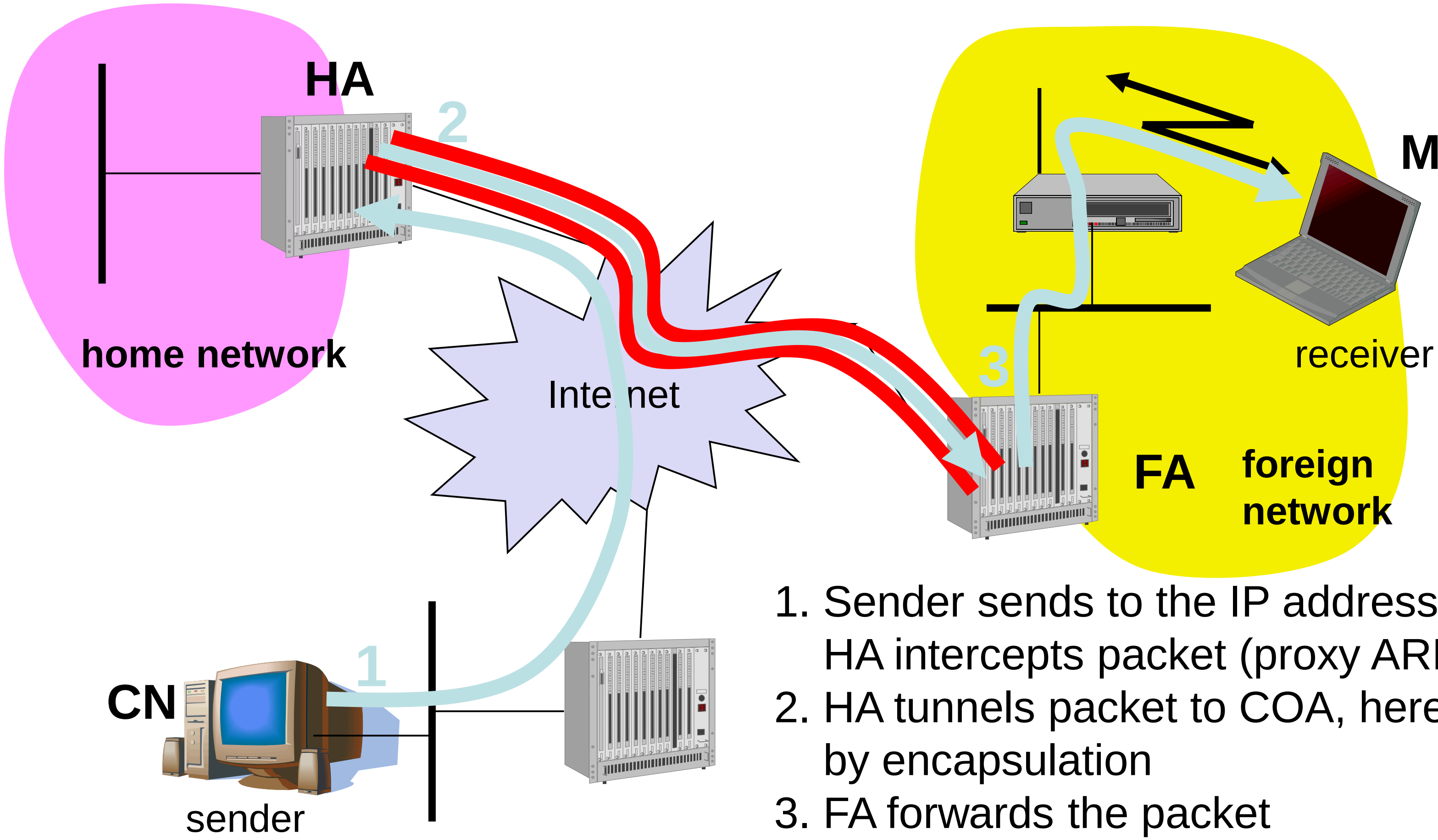


Data Transfer from the Mobile System



1. Sender sends to the IP address of the receiver as usual, FA works as default router

Data Transfer to the Mobile System

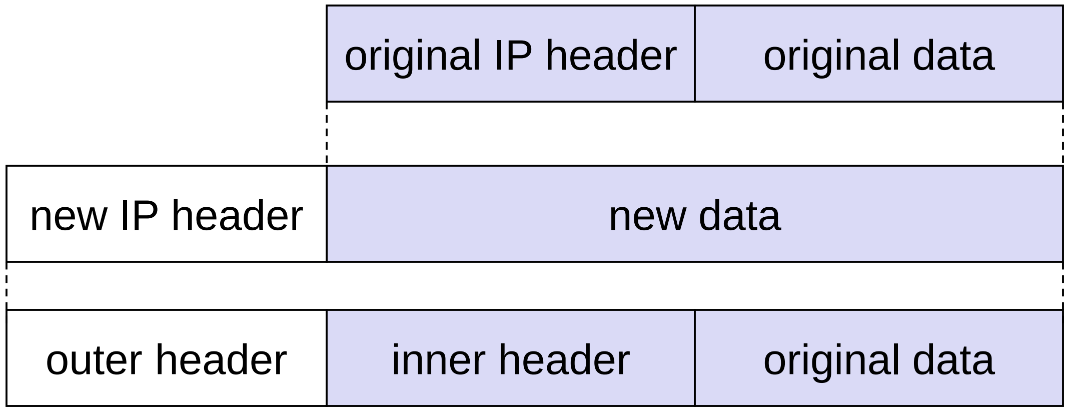


1. Sender sends to the IP address of MN, HA intercepts packet (proxy ARP)
2. HA tunnels packet to COA, here FA, by encapsulation
3. FA forwards the packet to the MN

Mobile IP Protocol Actions

- Agent Advertisement
  - HA and FA periodically send advertisement messages into their physical subnets
  - MN listens to these messages and detects, if it is in the home or a foreign network (standard case for home network)
  - MN reads a COA from the FA advertisement messages
- Registration (always limited lifetime!)
  - MN signals COA to the HA via the FA, HA acknowledges via FA to MN
  - these actions have to be secured by authentication
- Route Advertisement
  - HA advertises the IP address of the MN (as for fixed systems), i.e. standard routing information
  - routers adjust their entries, these are stable for a longer time (HA responsible for a MN over a longer period of time)
  - packets to the MN are sent to the HA,
  - independent of changes in COA/FA

Encapsulation



Encapsulation I

- Encapsulation of one packet into another as payload
  - e.g. IPv6 in IPv4 (6Bone), Multicast in Unicast (Mbone)
  - here: e.g. IP-in-IP-encapsulation, minimal encapsulation or GRE (Generic Record Encapsulation)
- IP-in-IP-encapsulation (mandatory, RFC 2003)
  - tunnel between HA and COA

<table>
<tr><td>ver.</td><td>IHL</td><td>DS (TOS)</td><td colspan="2">length</td></tr>
<tr><td colspan="3">IP identification</td><td>flags</td><td>fragment offset</td></tr>
<tr><td colspan="2">TTL</td><td>IP-in-IP</td><td colspan="2">IP checksum</td></tr>
<tr><td colspan="5">IP address of HA</td></tr>
<tr><td colspan="5">Care-of address COA</td></tr>
<tr><td>ver.</td><td>IHL</td><td>DS (TOS)</td><td colspan="2">length</td></tr>
<tr><td colspan="3">IP identification</td><td>flags</td><td>fragment offset</td></tr>
<tr><td colspan="2">TTL</td><td>lay. 4 prot.</td><td colspan="2">IP checksum</td></tr>
<tr><td colspan="5">IP address of CN</td></tr>
<tr><td colspan="5">IP address of MN</td></tr>
<tr><td colspan="5">TCP/UDP/ ... payload</td></tr>
</table>

Encapsulation II

- Minimal encapsulation (optional)
  - avoids repetition of identical fields
  - e.g. TTL, IHL, version, DS (RFC 2474, old: TOS)
  - only applicable for unfragmented packets, no space left for fragment identification

<table>
<tr><td>ver.</td><td>IHL</td><td colspan="2">DS (TOS)</td><td colspan="2">length</td></tr>
<tr><td colspan="4">IP identification</td><td>flags</td><td>fragment offset</td></tr>
<tr><td colspan="2">TTL</td><td colspan="2">min. encap.</td><td colspan="2">IP checksum</td></tr>
<tr><td colspan="6">IP address of HA</td></tr>
<tr><td colspan="6">care-of address COA</td></tr>
<tr><td colspan="2">lay. 4 protoc.</td><td>S</td><td>reserved</td><td colspan="2">IP checksum</td></tr>
<tr><td colspan="6">IP address of MN</td></tr>
<tr><td colspan="6">original sender IP address (if S=1)</td></tr>
<tr><td colspan="6">TCP/UDP/ ... payload</td></tr>
</table>

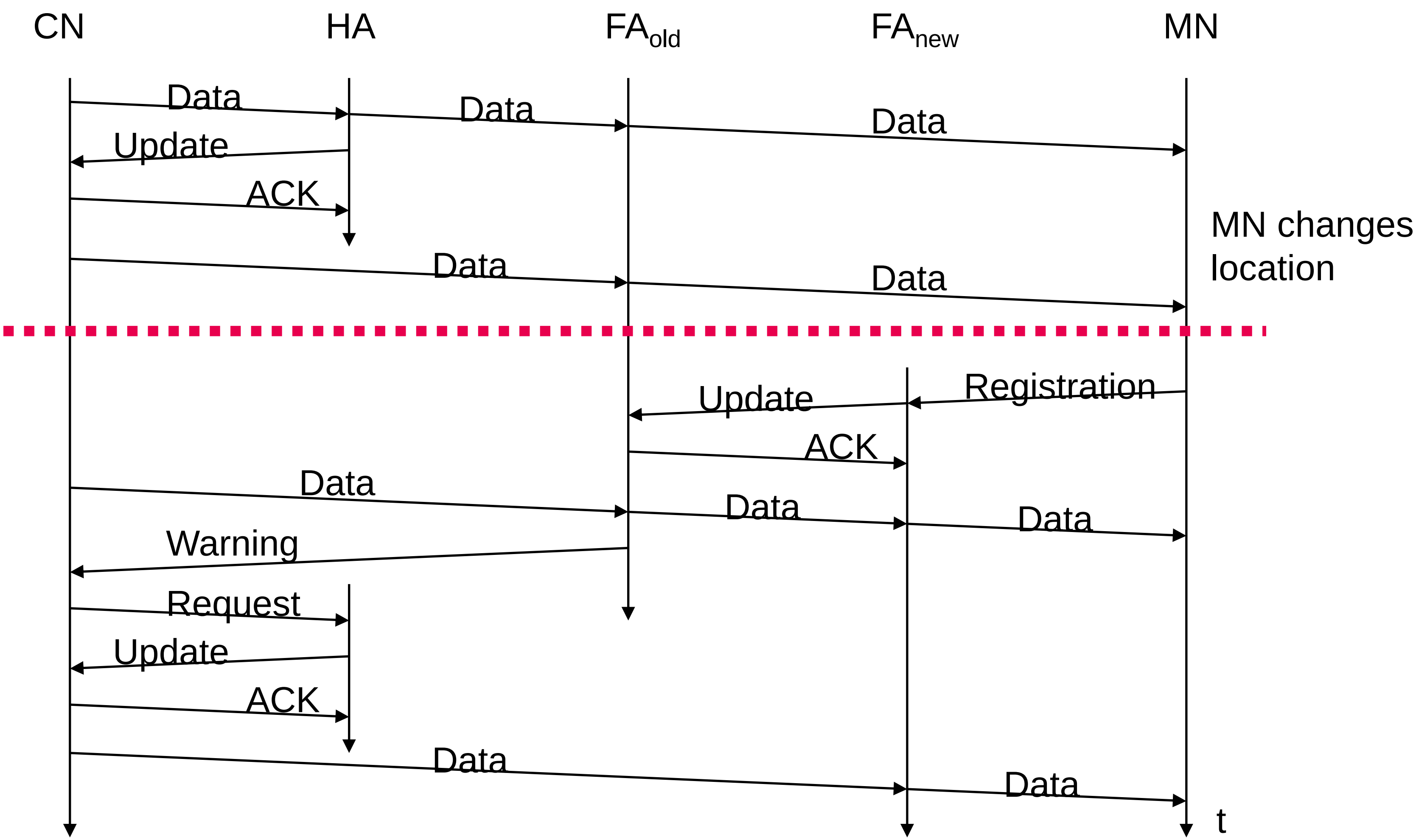
Mobile IP: Motion Detection

- detect when MH moved to new IP subnet, triggers new registration
- two primary mechanisms, others MAY be used:
  - algorithm 1 based on lifetime in agent advertisement:
    - MH records lifetime, updates it with every advertisement
    - upon expiration, assume that contact with agent is lost
    - register with an agent for which advertisement was received and whose lifetime is not yet expired
  - algorithm 2 uses network prefixes
    - compare newly received agent advertisements with network prefix of currently used care-of address
    - if prefixes differ, assume that MH moved
    - upon expiration of current registration, MH MAY choose to register with new FA

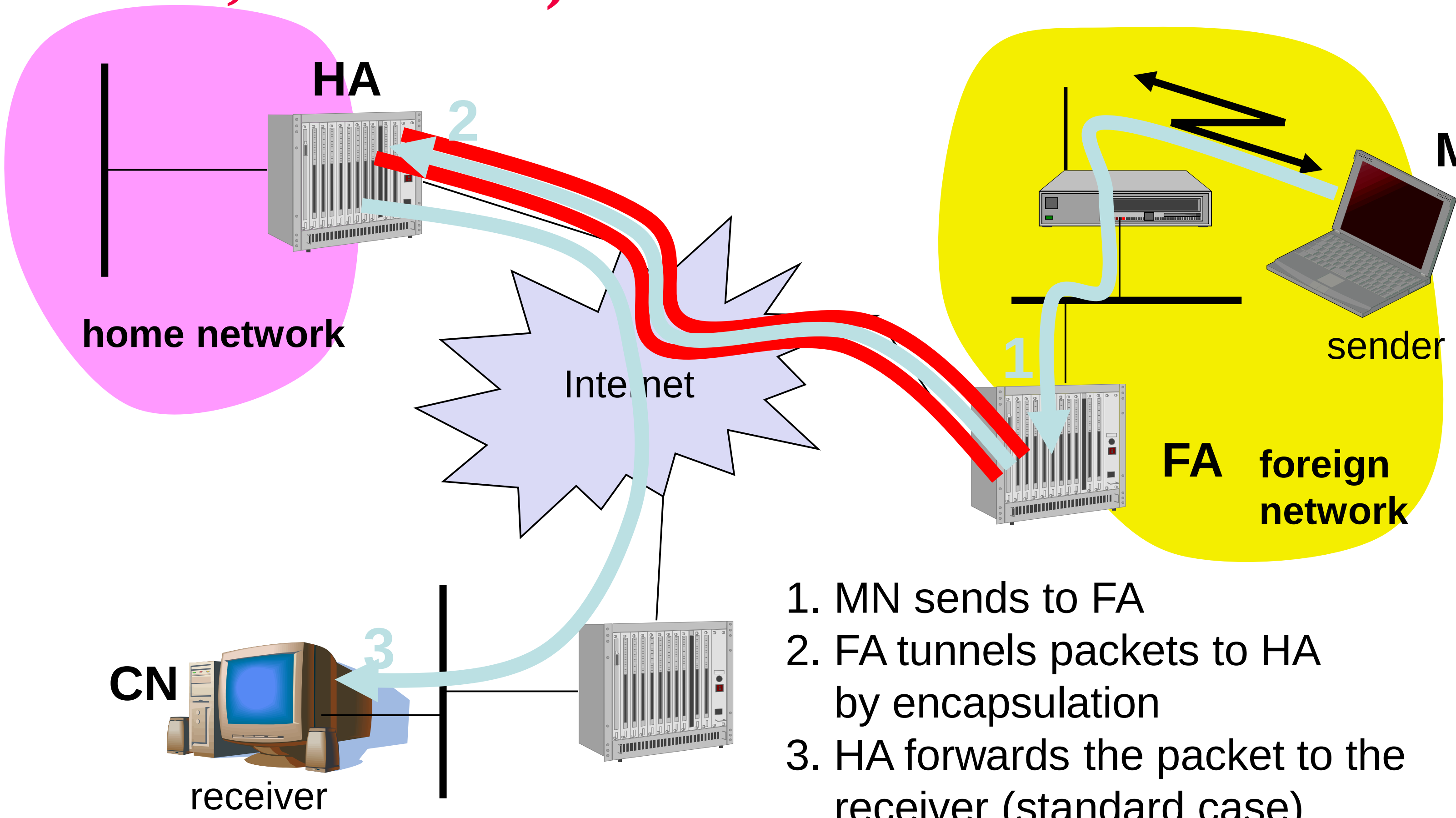
Optimization of Packet Forwarding

- Triangular Routing
  - sender sends all packets via HA to MN
  - higher latency and network load
- "Solutions"
  - sender learns the current location of MN
  - direct tunneling to this location
  - HA informs a sender about the location of MN
  - big security problems!
- Change of FA
  - packets on-the-fly during the change can be lost
  - new FA informs old FA to avoid packet loss, old FA now forwards remaining packets to new FA
  - this information also enables the old FA to release resources for the MN

Change of Foreign Agent



Reverse Tunneling (RFC 3024, was: 2344)



1. MN sends to FA
2. FA tunnels packets to HA by encapsulation
3. HA forwards the packet to the receiver (standard case)

Mobile IP with Reverse Tunneling

- Router accept often only "topological correct" addresses (firewall!)
  - a packet from the MN encapsulated by the FA is now topological correct
  - furthermore multicast and TTL problems solved (TTL in the home network correct, but MN is too far away from the receiver)
- 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)
- The standard is backwards compatible
  - the extensions can be implemented easily and cooperate with current implementations without these extensions
  - Agent Advertisements can carry requests for reverse tunneling

IP Micro-Mobility Support

- Micro-mobility support:
  - Efficient local handover inside a foreign domain without involving a home agent
  - Reduces control traffic on backbone
  - Especially needed in case of route optimization

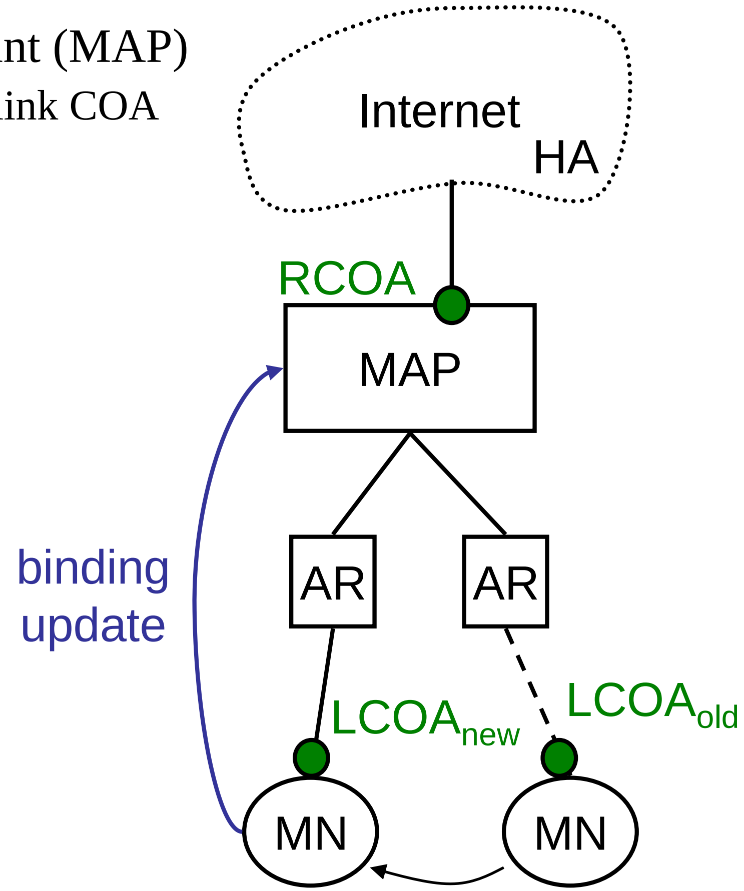
- Example approaches:
  - Cellular IP
  - HAWAII
  - Hierarchical Mobile IP (HMIP)
  - Others in the paper

- Important criteria:
  Security, Efficiency, Scalability, Transparency, Manageability

Hierarchical Mobile IPv6 (HMIPv6)

- Operation:
  - Network contains mobility anchor point (MAP)
    - mapping of regional COA (RCOA) to link COA (LCOA)
  - Upon handover, MN informs MAP only
    - gets new LCOA, keeps RCOA
  - HA is only contacted if MAP changes

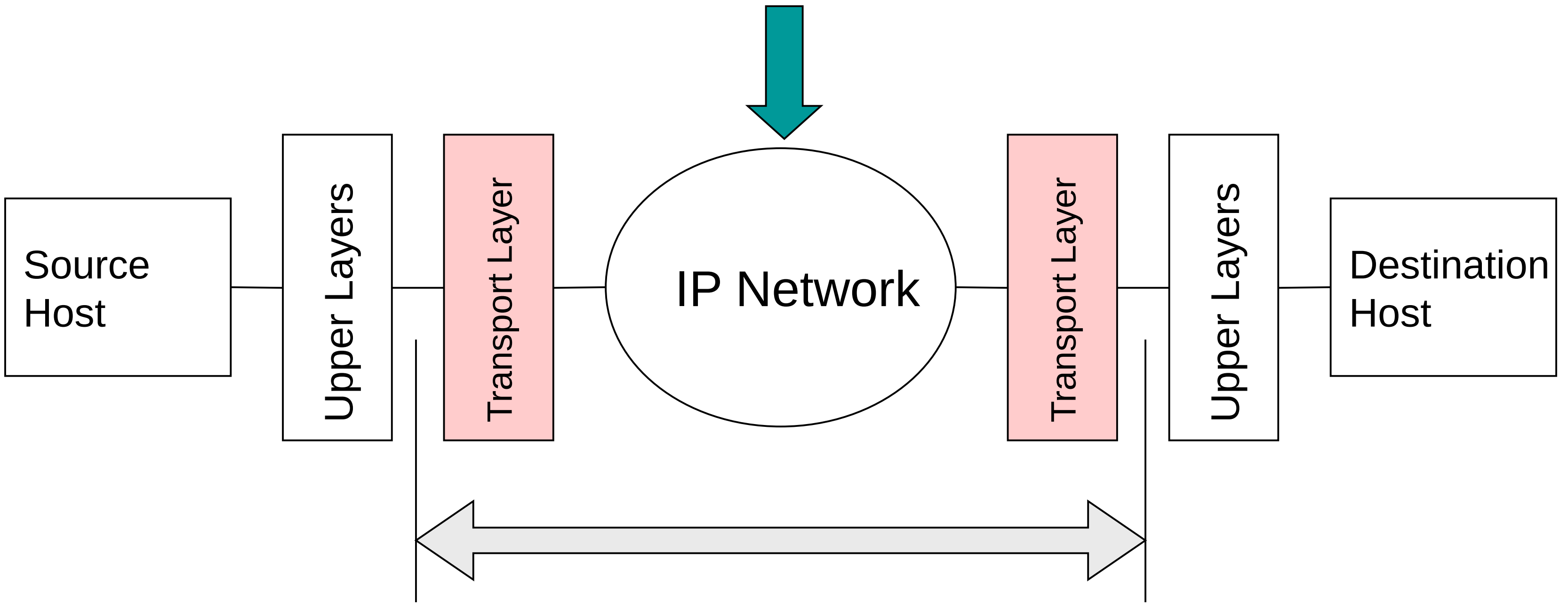
- Security provisions:
  - no HMIP-specific security provisions
  - binding updates should be authenticated



Transport Protocol

- What is the role of the "Transport Layer" ?

The IP Network DOES NOT guarantee delivery !!



The transport layer provides more reliable delivery

TCP and Mobile Computing

- TCP is popular transport layer protocol
- Designed for wired networks
  - low error rate
  - requirement to share bottlenecks
- Key assumptions in TCP are:
  - packet loss is indication of congestion, not transmission error
  - rather aggressive response to congestion is needed to ensure fairness and efficiency
- Wireless links and mobile computing violate these assumptions:
  - packets lost due to unreliable physical media
  - packets can get lost due to mobility (handover, route failure)

TCP and Mobile Computing

- Packet losses over wireless link often in bursts
  - will trigger slow start rather than fast retransmit
- Packet loss no indication of congestion
  - reduction of congestion window will reduce throughput
  - getting back to previous window size may take long
- Problem caused by mismatch of wireless link properties with assumptions underlying TCP design
- Multiple suggestions to improve TCP performance:
  - link-level retransmissions: improve reliability of wireless link
  - network layer solutions: SNOOP
  - transport layer solutions: I-TCP (indirect TCP), Mowgli
  - session layer solutions: establish end-to-end session layer connection, manages two separate TCP connections

Link Layer Mechanisms
Forward Error Correction

- Forward Error Correction (FEC) can be use to correct small number of errors
- Correctable errors hidden from the TCP sender
- FEC incurs overhead even when errors do not occur
  - Adaptive FEC schemes can reduce the overhead by choosing appropriate FEC dynamically
- FEC does not guard/protect from packet loss due to handover

Link Layer Mechanisms

Link Level Retransmissions

- Link level retransmission schemes retransmit a packet at the link layer, if errors are detected
- Retransmission overhead incurred only if errors occur
  - unlike FEC overhead

In general

- Use FEC to correct a small number of errors
- Use link level retransmission when FEC capability is exceeded

Link Level Retransmissions

Issues

- How many times to retransmit at the link level before giving up?
  - Finite bound -- semi-reliable link layer
  - No bound -- reliable link layer
- What triggers link level retransmissions?
  - Link layer timeout mechanism
  - Link level acks (negative acks, dupacks, …)
  - Other mechanisms (e.g., Snoop, as discussed later)
- How much time is required for a link layer retransmission?
  - Small fraction of end-to-end TCP RTT
  - Large fraction/multiple of end-to-end TCP RTT

Link Level Retransmissions

Issues

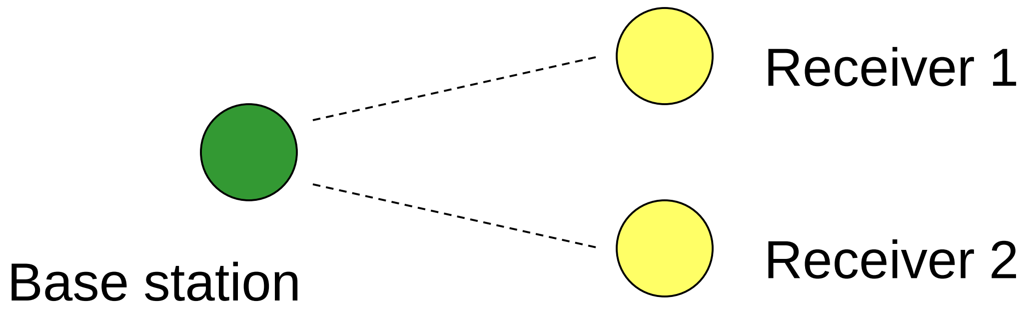
- Should the link layer deliver packets as they arrive, or deliver them in-order?
  - Link layer may need to buffer packets and reorder if necessary so as to deliver packets in-order

Carleton UNIVERSITY
Canada's Capital University

Link Level Retransmissions

Issues

- Retransmissions can cause head-of-the-line blocking

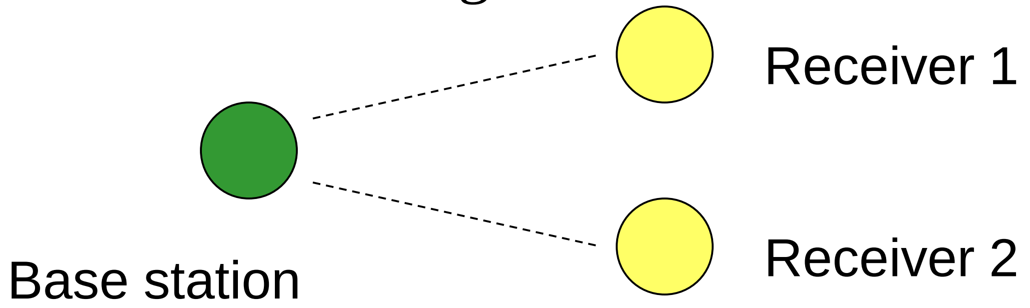


- Although link to receiver 1 may be in a bad state, the link to receiver 2 may be in a good state
- Retransmissions to receiver 1 are lost, and also block a packet from being sent to receiver 2

Link Level Retransmissions

Issues

- Retransmissions can cause congestion losses



- Attempting to retransmit a packet at the front of the queue, effectively reduces the available bandwidth, potentially making the queue at base station longer
- If the queue gets full, packets may be lost, indicating congestion to the sender
- Is this desirable or not ?

Link Level Retransmissions An Early Study

- The sender's Retransmission Timeout (RTO) is a function of measured RTT (round-trip times)
  - **Link level retransmits increase RTT, therefore, RTO**
- **If errors not frequent**, RTO will **not** account for RTT variations due to link level retransmissions
  - When errors occur, the sender may timeout & retransmit before link level retransmission is successful
  - Sender and link layer both retransmit
  - Duplicate retransmissions (interference) waste wireless bandwidth
  - Timeouts also result in reduced congestion window

A More Accurate Picture

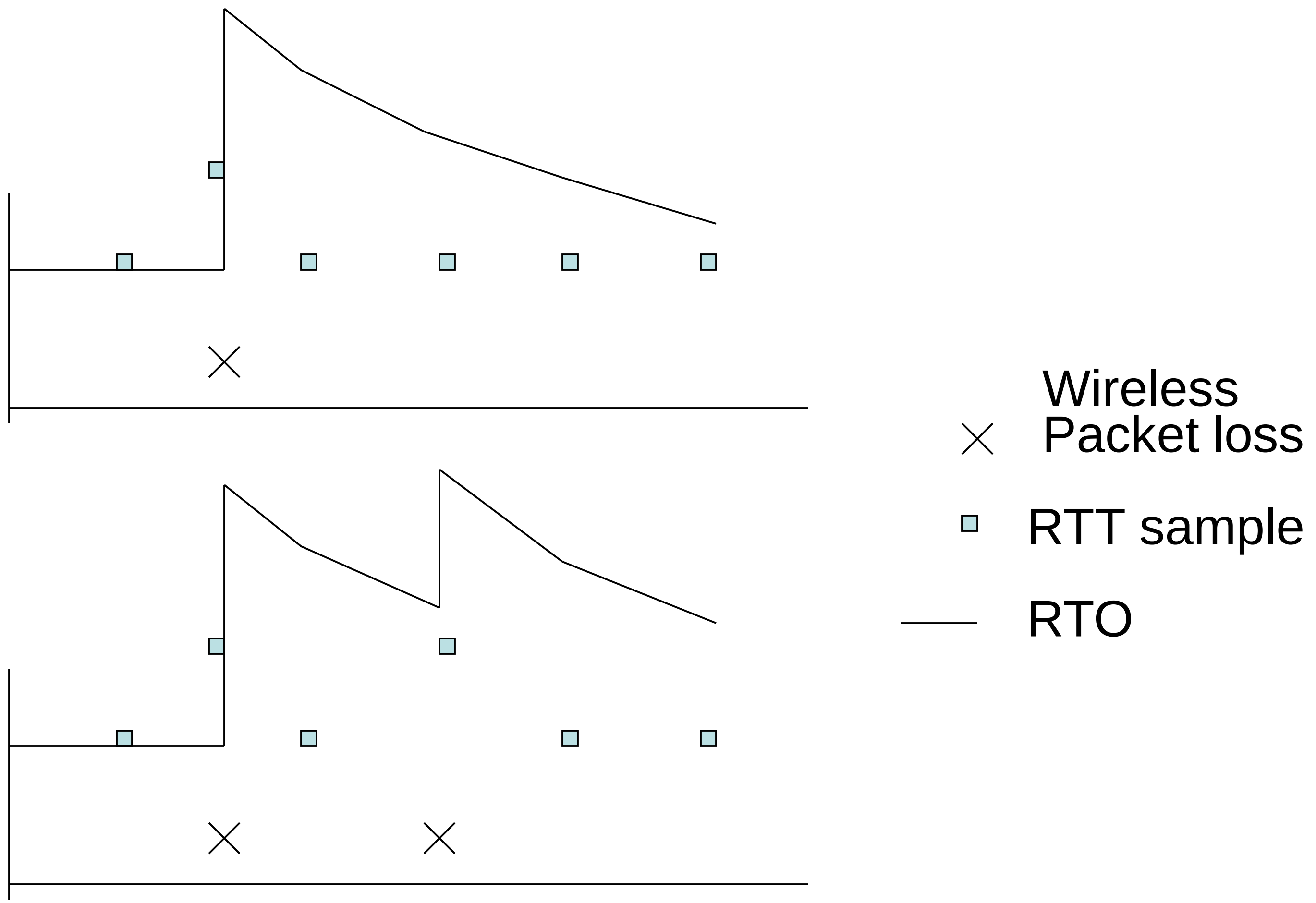
- Early analysis does not accurately model real TCP stacks
- With large **RTO granularity**, interference is unlikely, if time required for link-level retransmission is small compared to TCP RTO
  - Standard TCP RTO granularity is often large (500 ms)
  - Minimum RTO (2*granularity) is large enough to allow a small number of link level retransmissions, if link level RTT is relatively small
  - Interference due to timeout not a significant issue when wireless RTT small, and RTO granularity large

Link Level Retransmissions

A More Accurate Picture

- **Frequent errors** increase RTO significantly on slow wireless links
  - RTT on slow links large, retransmissions result in large variance, pushing RTO up
  - Likelihood of interference between link layer and TCP retransmissions smaller
  - But congestion response will be delayed due to larger RTO
  - When wireless losses do cause timeout, much time wasted

RTO Variations



Large TCP Retransmission Timeout Intervals

- Good for reducing interference with link level retransmits
- Bad for recovery from congestion losses
- Need a timeout mechanism that responds appropriately for both types of losses

Link Layer Schemes: Summary

When is a reliable link layer beneficial to TCP performance?

- if it provides *almost in-order* delivery
- TCP retransmission timeout large enough to tolerate additional delays due to link level retransmits
- Basic ideas:
  - Hide wireless losses from TCP sender
  - Link layer modifications needed at both ends of wireless link
    - TCP need not be modified

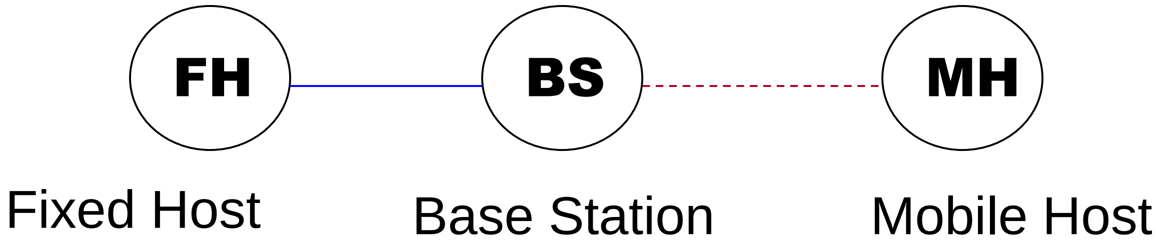
Split Connection Approach

- End-to-end TCP connection is broken into one connection on the wired part of route and one over wireless part of the route
- A single TCP connection split into two TCP connections
  - if wireless link is not last on route, then more than two TCP connections may be needed

Split Connection Approach

- Connection between wireless host MH and fixed host FH goes through base station BS

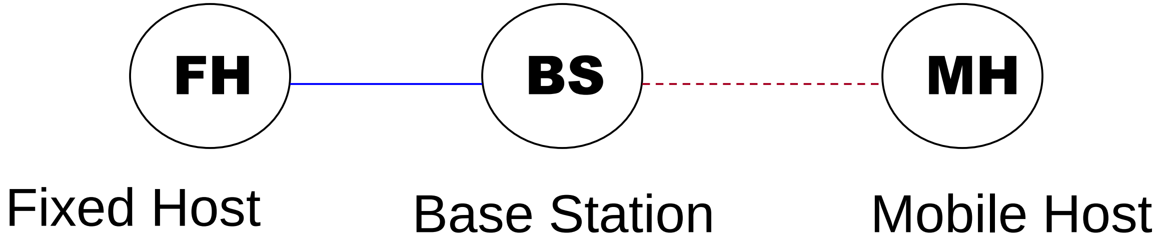
- FH-MH = FH-BS + BS-MH



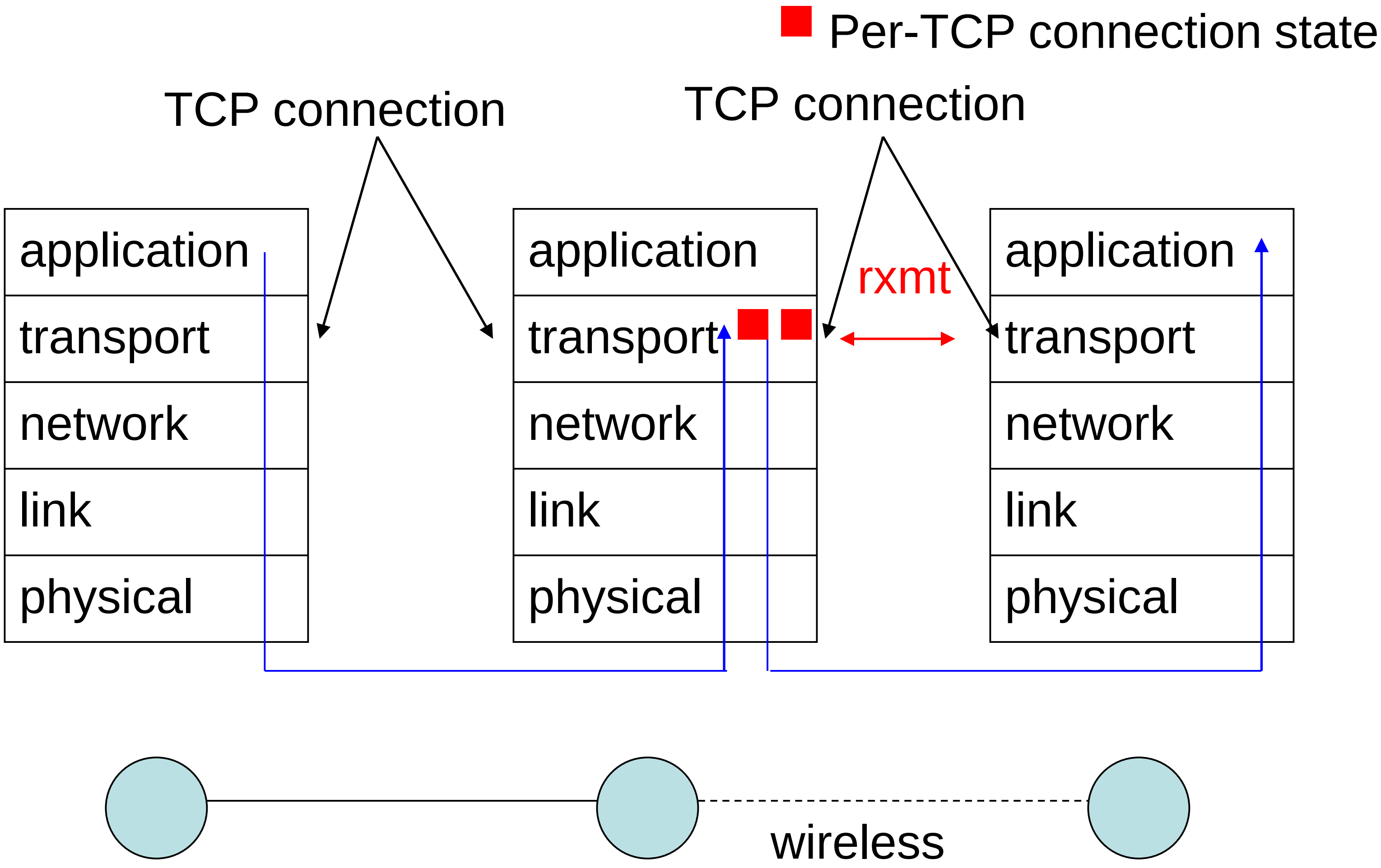
Split Connection Approach

- Split connection results in independent flow control for the two parts

- Flow/error control protocols, packet size, time-outs, may be different for each part



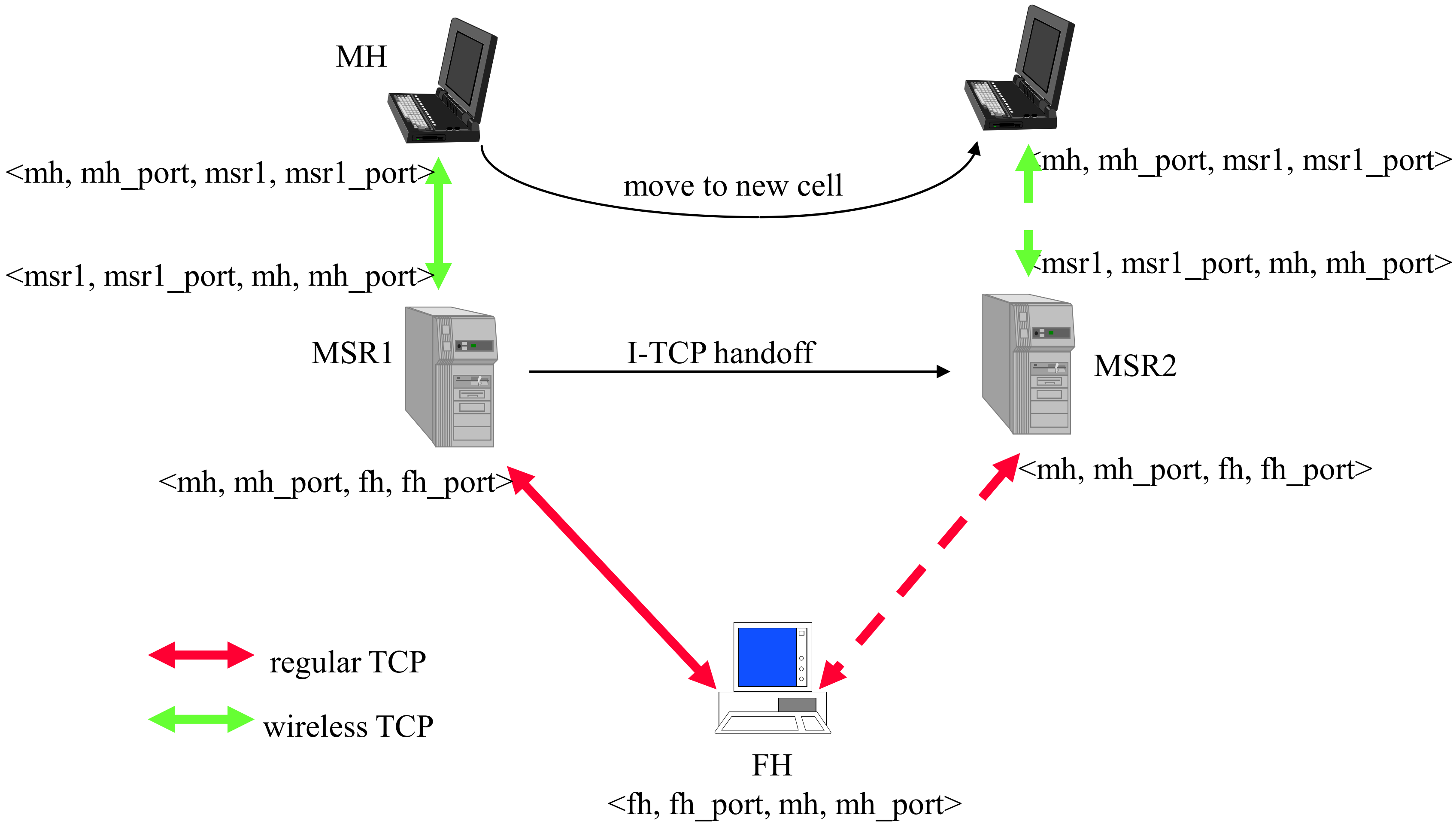
Split Connection Approach



I-TCP

- basic idea: split communication between mobile host (MH) and fixed host (FH) into two separate interactions
- each connection can be tuned to accommodate the special characteristics of the underlying physical media
  - use standard TCP between MSR and FH, both on wired backbone
  - special wireless TCP between MH and MSR, where packet loss does not trigger congestion avoidance

I-TCP: Connection Setup



I-TCP

- throughput improved, particularly for wide-area connections, compared to regular TCP

Connection Type	No moves	Overlapped cells	Disjoint cells, 0 sec between	Disjoint cells, 1 sec between
Regular TCP	65.49 kB/s	62.59 kB/s	38.66 kB/s	23.73 kB/s
I-TCP	70.06 kB/s	65.37 kB/s	44.83 kB/s	36.31 kB/s

I-TCP performance over local area

Connection Type	No moves	Overlapped cells	Disjoint cells, 0 sec between	Disjoint cells, 1 sec between
Regular TCP	13.35 kB/s	13.26 kB/s	8.89 kB/s	5.19 kB/s
I-TCP	26.78 kB/s	27.97 kB/s	19.12 kB/s	16.01 kB/s

I-TCP performance over wide area

Split Connection Approach: Classification

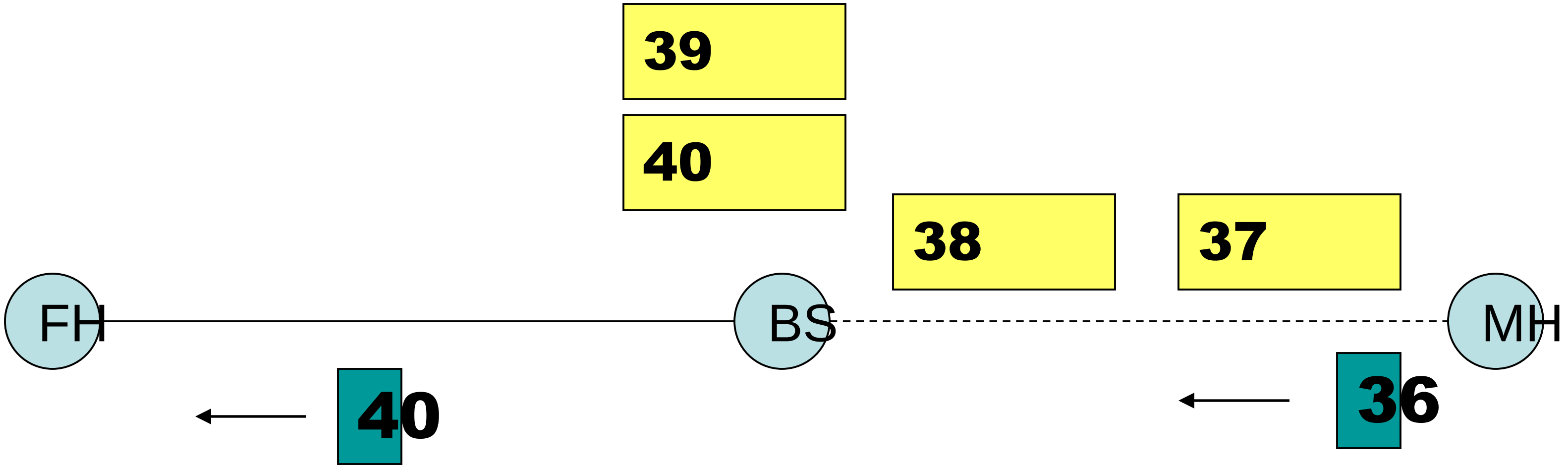
- Hides transmission errors from sender
- Primary responsibility at base station
- If specialized transport protocol used on wireless, then wireless host also needs modification

Split Connection Approach: Advantages

- BS-MH connection can be *optimized* independent of FH-BS connection
  - Different flow / error control on the two connections
- Local recovery of errors
  - Faster recovery due to relatively shorter RTT on wireless link
- Good performance achievable using **appropriate** BS-MH protocol
  - Standard TCP on BS-MH performs poorly when multiple packet losses occur per window (timeouts can occur on the BS-MH connection, stalling during the timeout interval)
  - **Selective acks improve performance for such cases**

Split Connection Approach: Disadvantages

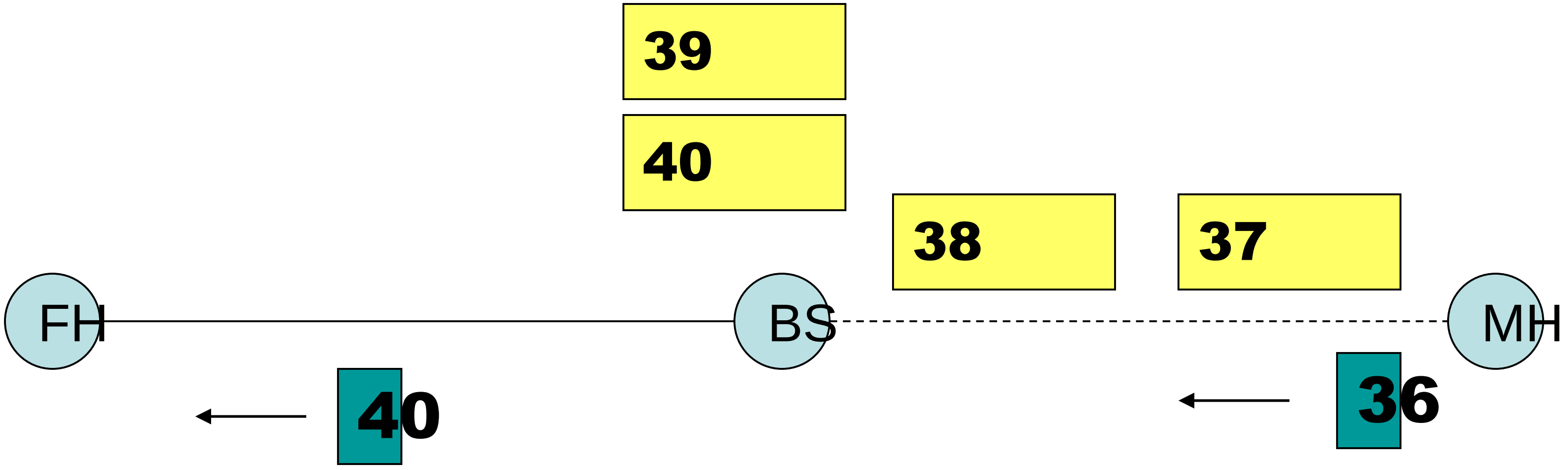
- End-to-end semantics violated
  - ack may be delivered to sender, before data delivered to the receiver
  - May not be a problem for applications that do not rely on TCP for the end-to-end semantics



Split Connection Approach: Disadvantages

- BS (MSR in I-TCP) retains hard state

  BS failure can result in loss of data (unreliability)
  - If BS fails, packet 40 will be lost
  - Because it is ack'd to sender, the sender does not buffer 40

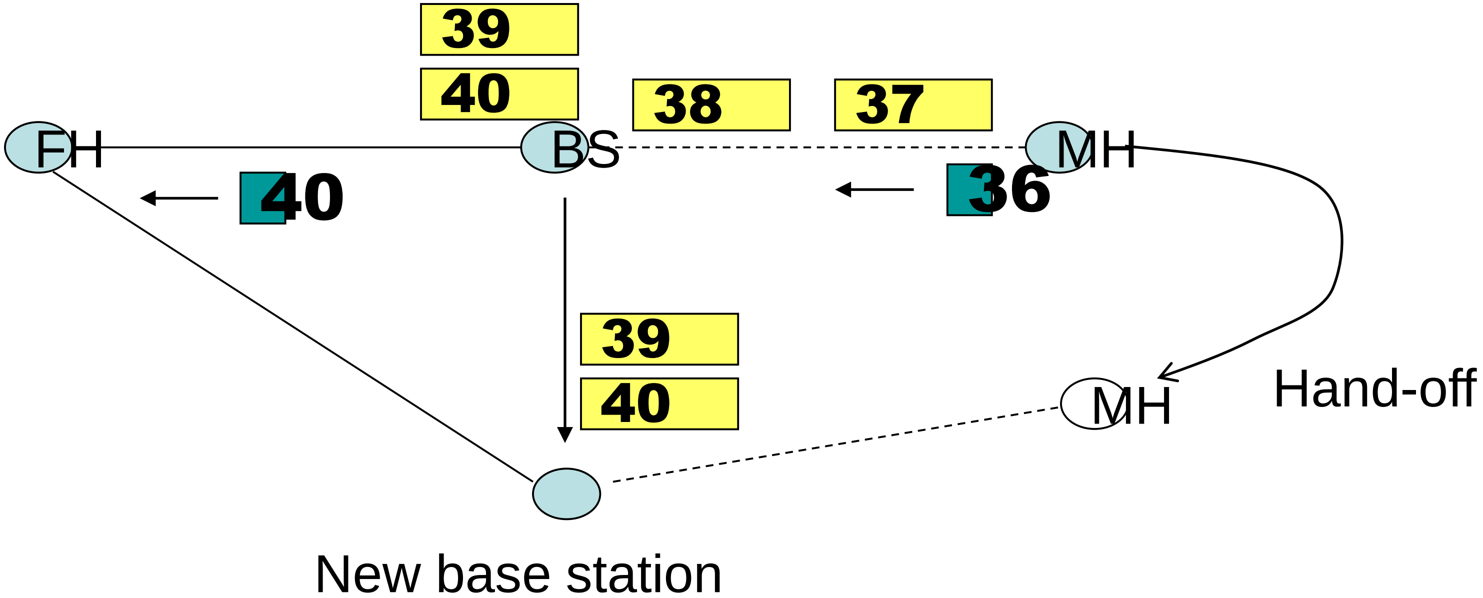


Split Connection Approach: Disadvantages

- BS retains hard state

  Hand-off latency increases due to state transfer

  – Data that has been ack'd to sender, must be moved to new base station

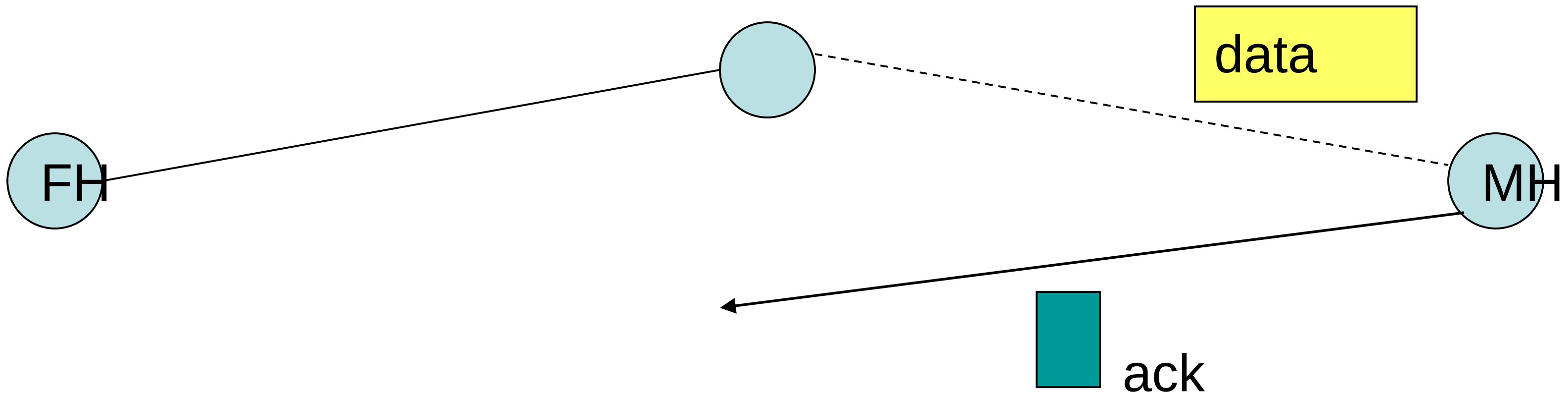


Split Connection Approach: Disadvantages

- Buffer space needed at BS for each TCP connection
  - BS buffers tend to get full, when wireless link slower (one window worth of data on wired connection could be stored at the base station, for each split connection)
- Window on BS-MH connection reduced in response to errors
  - may not be an issue for wireless links with small delay-bw product

Split Connection Approach: Disadvantages

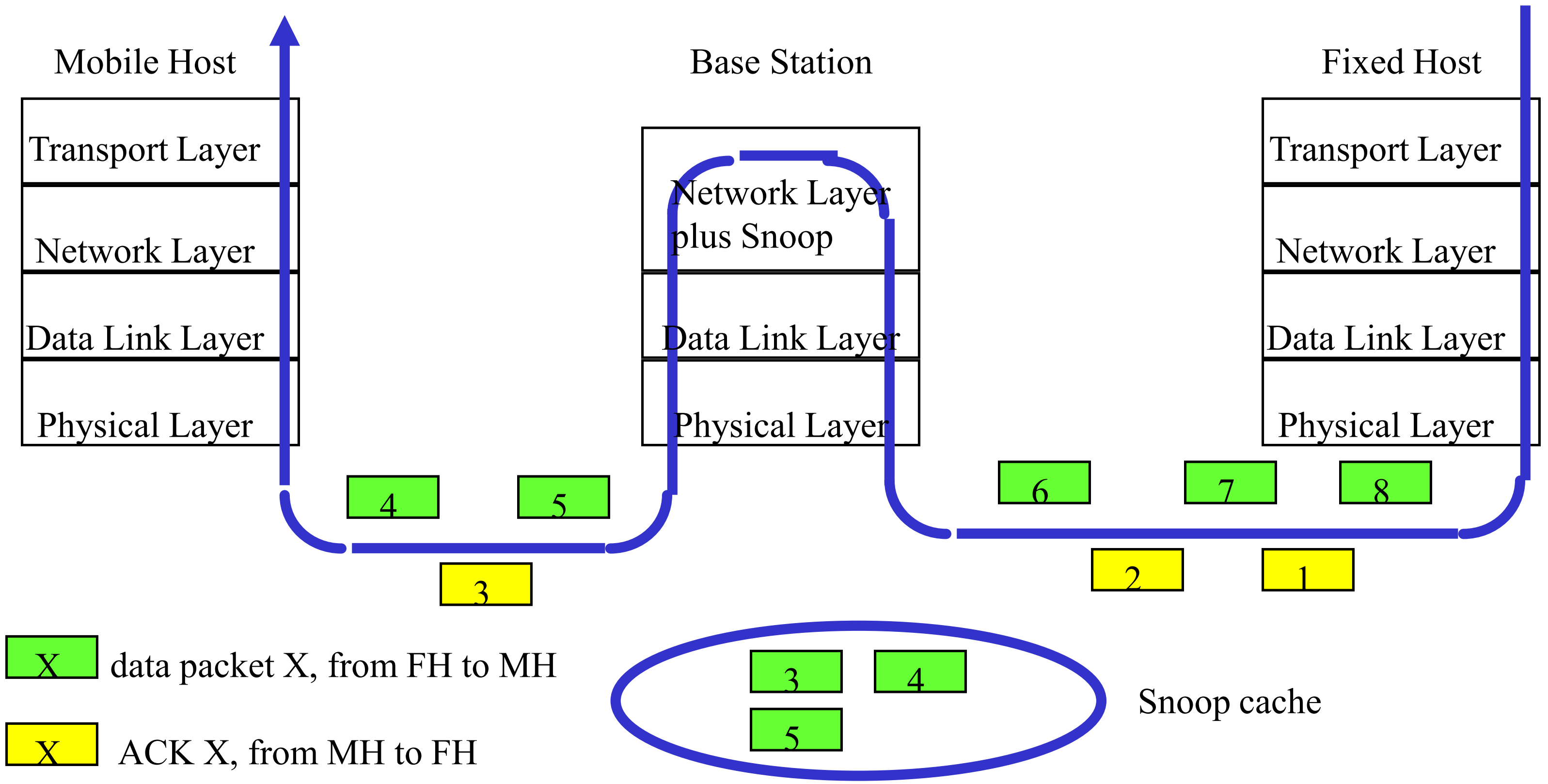
- Extra copying of data at BS
  – copying from FH-BS socket buffer to BS-MH socket buffer
  – increases end-to-end latency
- May not be useful if data and acks traverse different paths (both do not go through the base station)
  – Example: data on a satellite wireless hop, acks on a dial-up channel



Snoop: Network Layer Solution

- idea: modify network layer software at base station
- changes are transparent to MH and FH
  - no changes in TCP semantics (unlike I-TCP)
  - less software overhead (packets pass TCP layer only twice)
  - no application relinking on mobile host
- modifications are mostly in caching packets and performing local retransmissions across the wireless link by monitoring (*snooping*) on TCP acks
- results are impressive:
  - Speed ups of up to 20 times over regular TCP
  - More robustness when dealing with multiple packet losses

Snoop: Architecture



Snoop: Description of Protocol

- processing packets from FH
  - new packet in the normal TCP sequence:
    - cache and forward to MH
  - packet out-of sequence and cached earlier:
    - sequence number > last ack from MH: packet probably lost, forward it again
    - otherwise, packet already received at MH, so drop
      - but: original ACK could have been lost, so fake ACK again
  - packet out-of sequence and not cached yet:
    - packet either lost earlier due to congestion or delivered out-of-order: cache packet and mark as retransmitted, forward to MH

Snoop: Description of Protocol

- processing ACKs from MH:
  - new ACK: common case, initiates cleaning up of snoop cache, update estimate of round-trip time for wireless link, forward ACK to FH
  - spurious ACK: less than last ACK seen, happens rarely. Just drop ACK and continue
  - duplicate ACK: indicates packet loss, one of several actions:
    - packet either not in cache or marked as retransmitted: pass duplicate ACK on to FH
    - first duplicated ACK for cached packet: retransmit cached packet immediately and at high priority, estimate number of expected duplicate ACKs, based on # of packets sent after missing one
    - expected successive duplicate ACKs: ignore, we already initiated retransmission. Since retransmission happens at higher priority, we might not see total number of expected duplicate ACKs

Snoop: Description of Protocol

- design does not cache packets from MH to FH
  - bulk of packet losses will be between MH and base
  - but snooping on packets generates requests for retransmissions at base much faster than from remote FH
  - enhance TCP implementation at MH with "selective ACK" option:
    - base keeps track of packets lost in a transmission window
    - sends bit vector back to MH to trigger retransmission of lost packets
- mobility handling:
  - when handoff is requested by MH or anticipated by base station, nearby base stations begin receiving packets destined for MH, priming their cache
  - caches synchronized during actual handoff (since nearby bases cannot snoop on ACKs)

Snoop: Performance

- no difference in very low error rate environment (bit error rate < $5x10^{-7}$)
- for higher bit error rates, Snoop outperforms regular TCP by a factor of 1 to 20, depending on the bit error rate (the higher, the better Snoop's relative performance)
- even when every other packet was dropped over the wireless link, Snoop still allowed for progress in transmission, while regular TCP came to a grinding halt
- Snoop provides high and consistent throughput, regular TCP triggers congestion control often, which leads to periods of no transmission and very uneven rate of progress

Snoop: Evaluation

- most effort spent on direction FH->MH
  - authors argue that not much can be done for MH->FH
    - losses occur over first link, the unreliable wireless link
- Internet drops 2%-5% of IP packets, tendency rising
  - assume that IP packet is lost in wired part of network:
    - receiver (FH) will issue duplicate ACKs
    - this should trigger fast retransmit rather than slow start (?)
    - nothing is done to ensure that ACKs are not dropped over last link
    - retransmission of data packet over wireless link is subject to unreliable link and low bandwidth again
  - Snoop could potentially benefit from caching packets in both directions
    - how would this differ from link-layer retransmission policy?

TCP over Wireless: Summary

- Many proposals focus on downlink only
- Many proposals, most try to avoid changing TCP interface or semantics
- Topics ignored:
  - asymmetric bandwidth on uplink and downlink (for example in some cable or satellite networks)
  - wireless link extends over multiple hops, such as in an ad-hoc network: connections fail due to spurious disconnections or route failures in ad-hoc networks
  - fairness
  - Performance-Enhancing Proxies (PEP): RFC 3135
    - Break end-to-end semantics and security
    - Can exist at transport layer or application layer

Messaging over Lossy Networks

- TCP: problem well understood and studied for a long time
- Newer version: M2M (machine-to-machine) communication
  - Add "wrinkle": keep protocol simple to run protocol stack on small embedded devices (controller in a dish washer, microwave, etc.)
  - Sample protocols:
    - MQTT: publish-subscribe based "light weight" messaging protocol for connections with remote locations where a small code footprint is required and/or network bandwidth is limited
    - Constrained Application Protocol (CoAP): a transfer protocol for use with constrained nodes and constrained (e.g., low-power, lossy) networks