# Mobile IP and  Mobile Transport Protocols

# Preliminaries

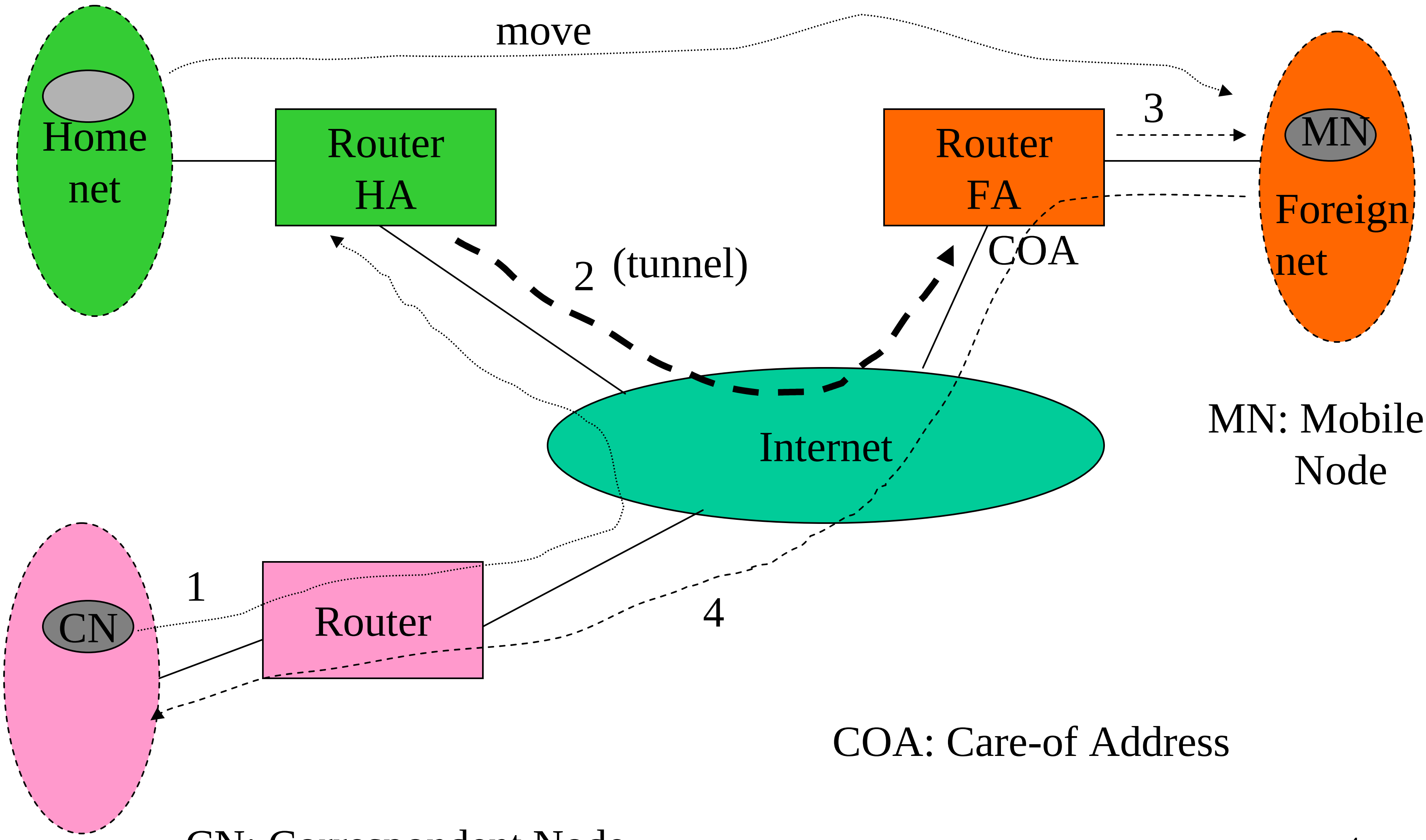
- **IP routing**
  - **Works on a hop-by-hop basis using a routing table**
  - **32 bits: 129.97.92.42**
  - **Address = subnet + host (Mobility ➔ No packet for you)**
  - **Two parts**
    - » **Routing protocol: constructs routing tables**
    - » **Packet forwarding: Uses a routing table**

| Target/Prefix-Length | Next Hop | Interface |
|---|---|---|
| 7.7.7.99/32 (host specific) | R 1 | a |
| 7.7.7.0/24 (network prefix) | R 2 | b |
| 0.0.0.0/0 (default) | R 3 | c |

# The need for Mobile IP

- Hosts and routers base their forwarding decision on **network prefix** portion of an IP address.
- When a host moves from its **home link** to a **foreign link**, the host becomes unreachable.
  - Home link: The link on which a node should be located. (This link has been assigned the same network-prefix as the node’s IP address.)
  - Foreign link: Any link other than a node’s home link.

# Entities and Packet delivery




MN: Mobile Node

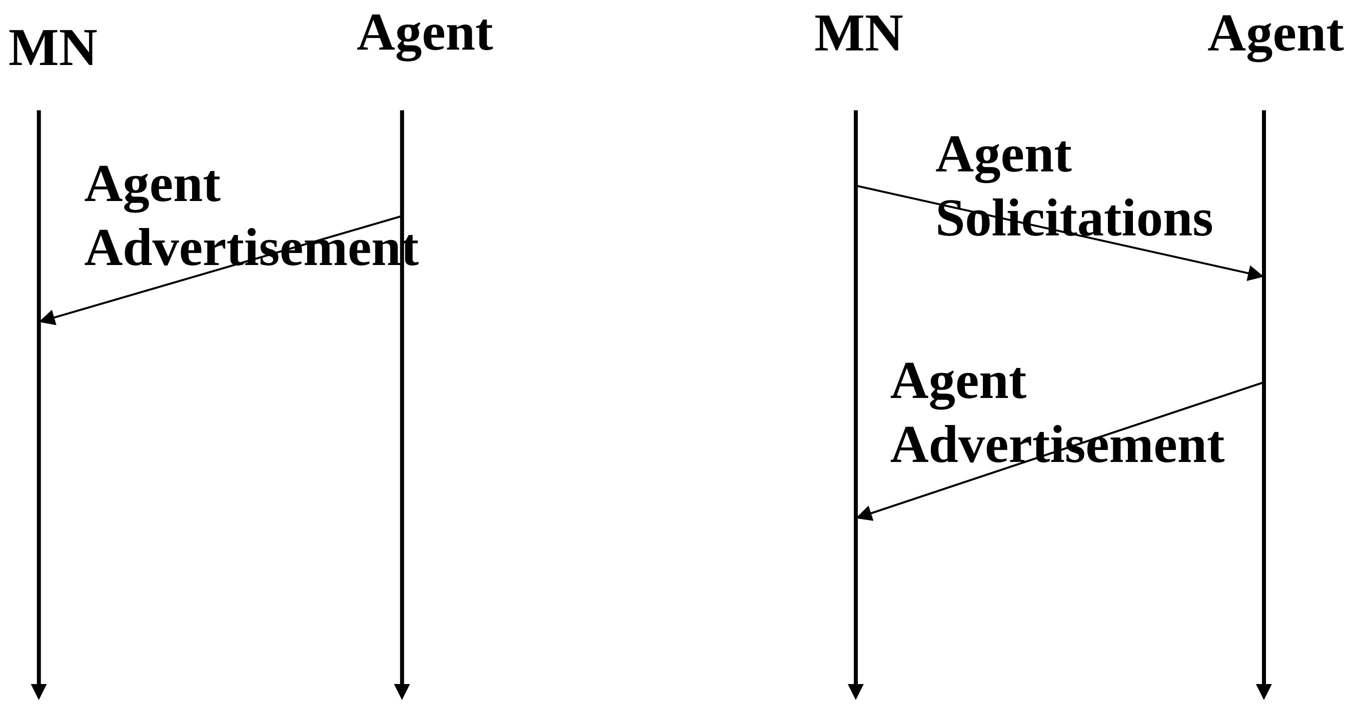
COA: Care-of Address

CN: Correspondent Node

# What is Agent Discovery?

- A mobile node
  - Determines whether it is currently connected to its home link or a foreign link.
  - Detects whether it has moved from one link to another.
  - Obtains a COA when connected to a foreign link.

# Agent Discovery?



# Agent Discovery?

- Important Fields in *Agent Advertisement* message
  - **IP Source Address (HA or FA)**
    - **Know if you are home or away.**
  - **COA fields: one or more IP addresses**
    - **Select one**
  - **Lifetime**
    - **How soon the MN will hear from the agent again.**

# Move Detection

- Move detection using *Lifetime*: If you don't hear from the FA after *Lifetime*
    1. **Register with the next FA from which you receive an AA message.**
    2. **Broadcast an Agent Solicitation message.**
- Using Network-Prefixes
    - **For each advertised route, there is a prefix length in IP header.**
        - » **A different network prefix means the node has moved.**

# Mobile IP Registration

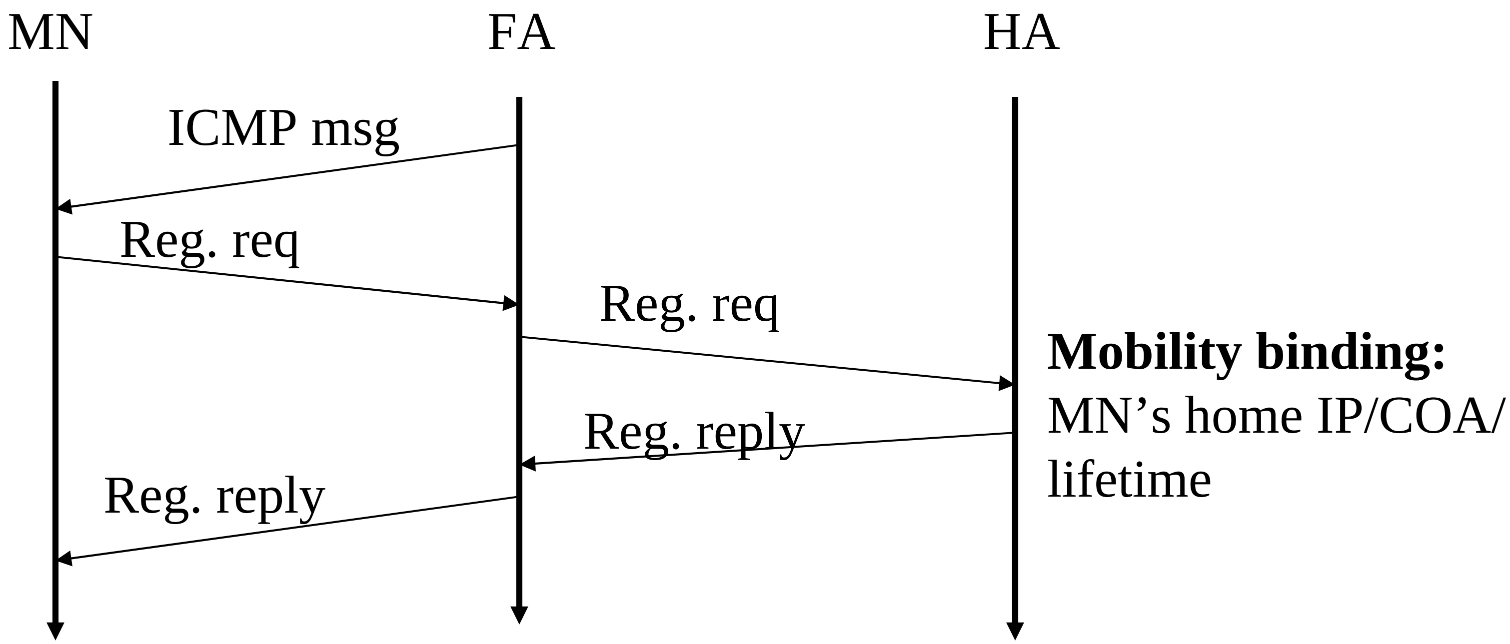
– This is a process by which an MN
  - Requests routing service from an FA.
  - Informs its HA of its current COA.
  - Renews a registration which is due to expire.
  - Deregisters when it returns to its home link.
    - **Act like a fixed host (no use of Mobile IP features.)**

# Important fields of a *Registration Request*

- Source/Destination addresses
- Mobile node's home address
- Home agent address
- COA
- Time to Live

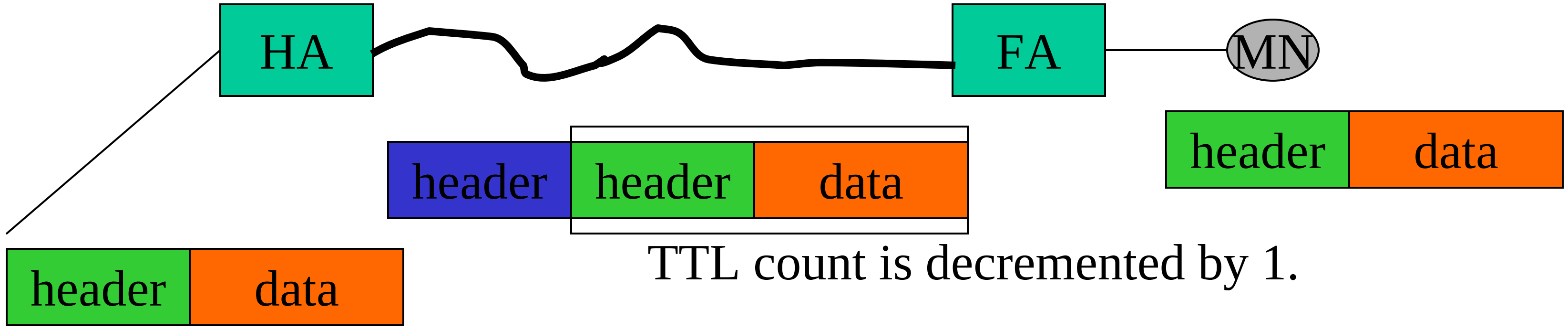
# Registration

- After receiving a COA, MN registers with HA



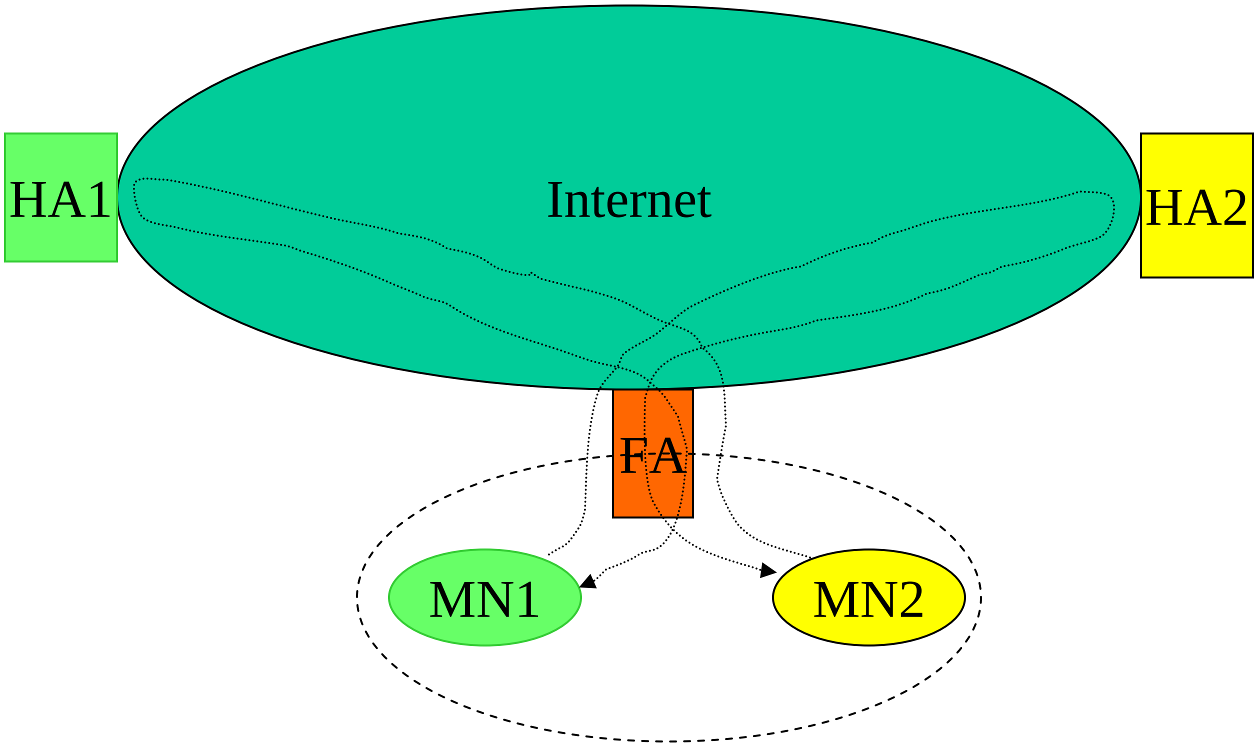
# Tunneling + Encapsulation

- **Tunnel**: data pipe between HA and COA
- Tunneling is achieved by **encapsulation**.

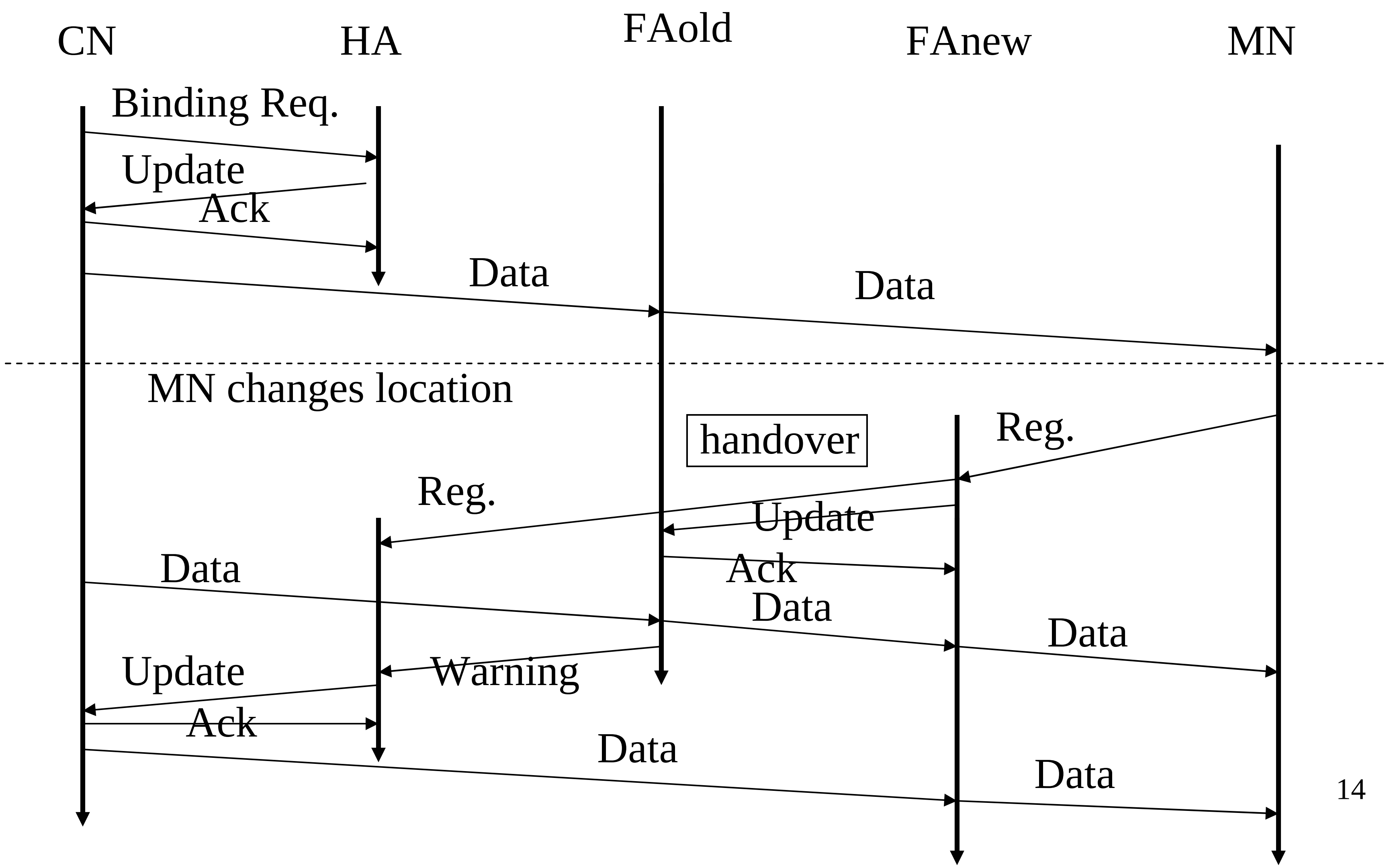



IP-in-IP

# Need for optimization



# Optimized mobile IP



# IPv6

- Basic features
- Mobility for IPv6
- Mobile IPv6

# How does IPv6 differ from IPv4

- 128 bits in IPv6 (implication to be discussed)
- Rigorous definition of optional header fields (in the form of extension headers)
  - IPv6 Base header ➔ Extension header ➔ ….

# IPv6 Base header

- Version (4 bits) = 6
- Priority (4 bits): among packets from same source
- Flow (24 bits): Might find application
- Payload length (16 bits)
- Next header (8 bits): Type of header following this.
- Hop limit(8 bits): Decremented by 1 after each hop
- Source address (128 bits)
- Destination address (128 bits)

# IPv6 extension header

- Hop-by-hop Options header: Contains options to be examined by every router along the path.
- Destination Options header: Examined by the destination.
- Upper-layer header: The TCP or app header, or an IPv6 or IPv4 header in case of a tunnel.
- IP Authentication header
- Routing header

# Mobile IPv6: its components

- An MN determines its location.
- On a foreign link, an MN acquires a **COA—collocated.**
- The MN notifies its HA of its COA.
- The MN may notify selected correspondents of its COA.
- Packets are forwarded to an MN using tunneling.
- Correspondents knowing the COA directly send packets to the MN.

# IPv6 Agent Discovery Protocol

- Knows whether it is on its home link or foreign link.
- Knows whether it has moved from one link to another.
- Obtains a COA when connected to a foreign link.

MN, routers, and HA cooperate to accomplish the above.

# IPv6: Location and movement detection

- Routers and HA broadcast *Router Advertisement* (RA) messages from time to time.
- If *prefix*(RA) matches with MN's home address, the MN is on its home link. *Notify* HA of your return.
- If there is no match, compare *prefix*(RA) with those of previous advertisements to see if it has moved.
- If MN has moved, <u>acquire a new COA</u> (to be explained).

# IPv6: Obtaining a COA address

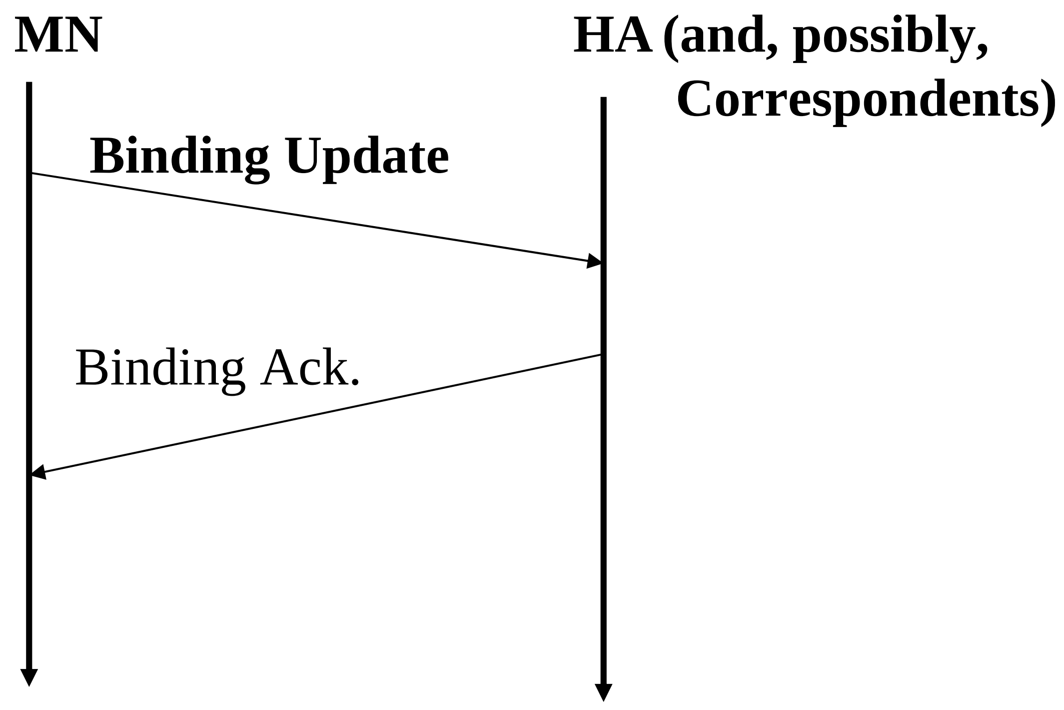
- In IPv6, COAs are collocated. (There is no FA as the address space is huge.)
- Two ways of obtaining a COA:
  - **Stateful Address Autoconfiguration**
    - **Ask a server (DHCPv6) for an address and use it.**
  - **Stateless Address Autoconfiguration**
    - **COA = network prefix @ interface token**

      **(Interface token is link dependent ID, such as MAC address)**

# IPv6: An MN informing others of its COA

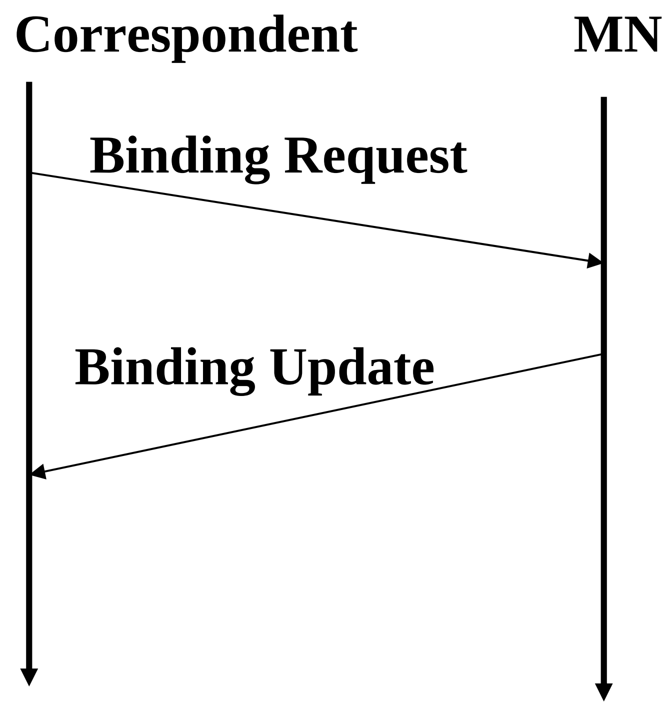
- ***Notification* to HA**
  - **The HA uses the COA as the exit-point of a tunnel to get packets to a mobile node.**
- ***Notification* to correspondents**
  - **Correspondents use COA to route packets directly to the mobile node. ➔ IPv6 supports route optimization.**

# IPv6: Notification scenario




IPv6: Update and Ack are IPv6 extended headers.
IPv4: Registration and Ack are UDP/IP messages.

# IPv6: Notification scenario



# IPv6: Correspondent sends data to MN

- Correspondent knows MN’s COA:
  - **Destination = COA and**
    **Routing Header = HA ➔ MN consumes the packet**
- Correspondent <u>does not know</u> MN’s COA
  - **Packet sent as usual to the HA of the MN**
  - **HA tunnels the packet to the COA.**
    - » **Tunnel ➔ Correspondent does not know the COA**
    - » **Send a *Binding Update* to Correspondent.**

# Mobile transport layer

## Outline

- Traditional TCP
- Implication on mobility
- Different solutions

# TCP

- Network layer **addresses** a host.
- TCP ports allow addressing of applications.
- TCP: end-to-end
  - Lossless
  - In-order
- Traditional TCP (factors affecting performance)
  - Congestion control
  - Slow start

# Congestion control

- Packet loss
  - Congestion, no buffer, router drops packets
- Receiver notices a **gap**
  - Does not ask for selective retransmission
  - All in-sequence packets up to the missing one are ACKed.

# Congestion control

- Sender
  - Notices the missing ACK for the lost packet
  - Assumes packet loss due to congestion
  - Slows down the transmission rate
  - All senders facing the same congestion slow down ← reason for survival of the Internet

# Slow start

- TCP's response to congestion detection
- Congestion window (CW): exp growth
    - Start: CW = 1, send a packet
    - Receive ACK ➔ CW = 2, send 2 packets
    - Receive 2 ACK ➔ CW = 4, …..
    - Exp growth stops at **congestion threshold (CT)**
    - Next is linear growth (1 by 1)

# Slow start

- Linear growth continues until
  - Timeout occurs: missing ACK
  - Continuous ACK for same packet ➔
    Perceived congestion ➔ **CT = CW/2, CW = 1**

# Fast recovery/fast retransmit

- Two reasons leading to a reduced CT
    - Timeout
    - Receiving continuous ACKs for same packet
- Timeout
    - Real congestion
- Continuous ACKs for same packet
    - No real congestion (retransmit)

# Implications on mobility

- Wireless system
  - Higher error rate
  - Packet loss is much more common
  - Layer-2 retransmission may not be good solution
    - could trigger TCP retransmissions
  - Mobility itself can cause packet loss
    - During handover
    - Rerouting problem, and not wireless access
  - Congestion: **not the main reason** for packet loss

  ➔ Degraded TCP performance

# Implications on mobility

- No drastic change in TCP is possible
  - Installed based of TCP is too large
  - Slow start keep the Internet going

  ➔Changes must be compatible

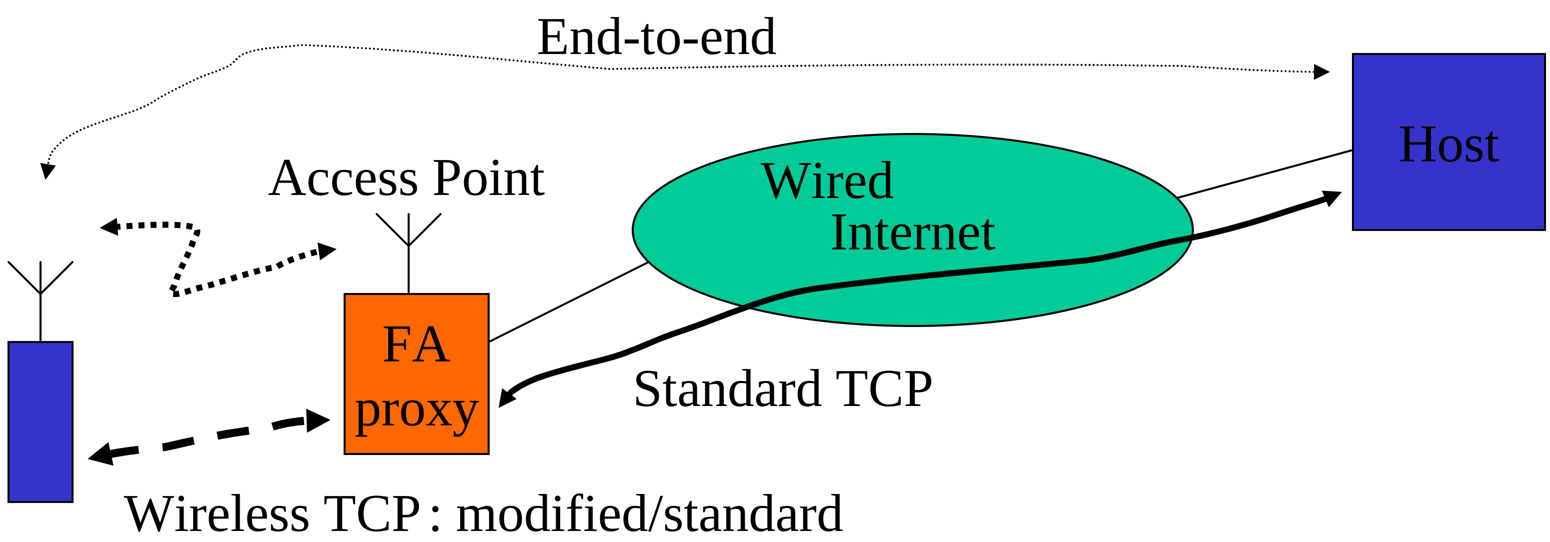
  Must not jeopardize cautious behavior

# TCPs for mobile systems

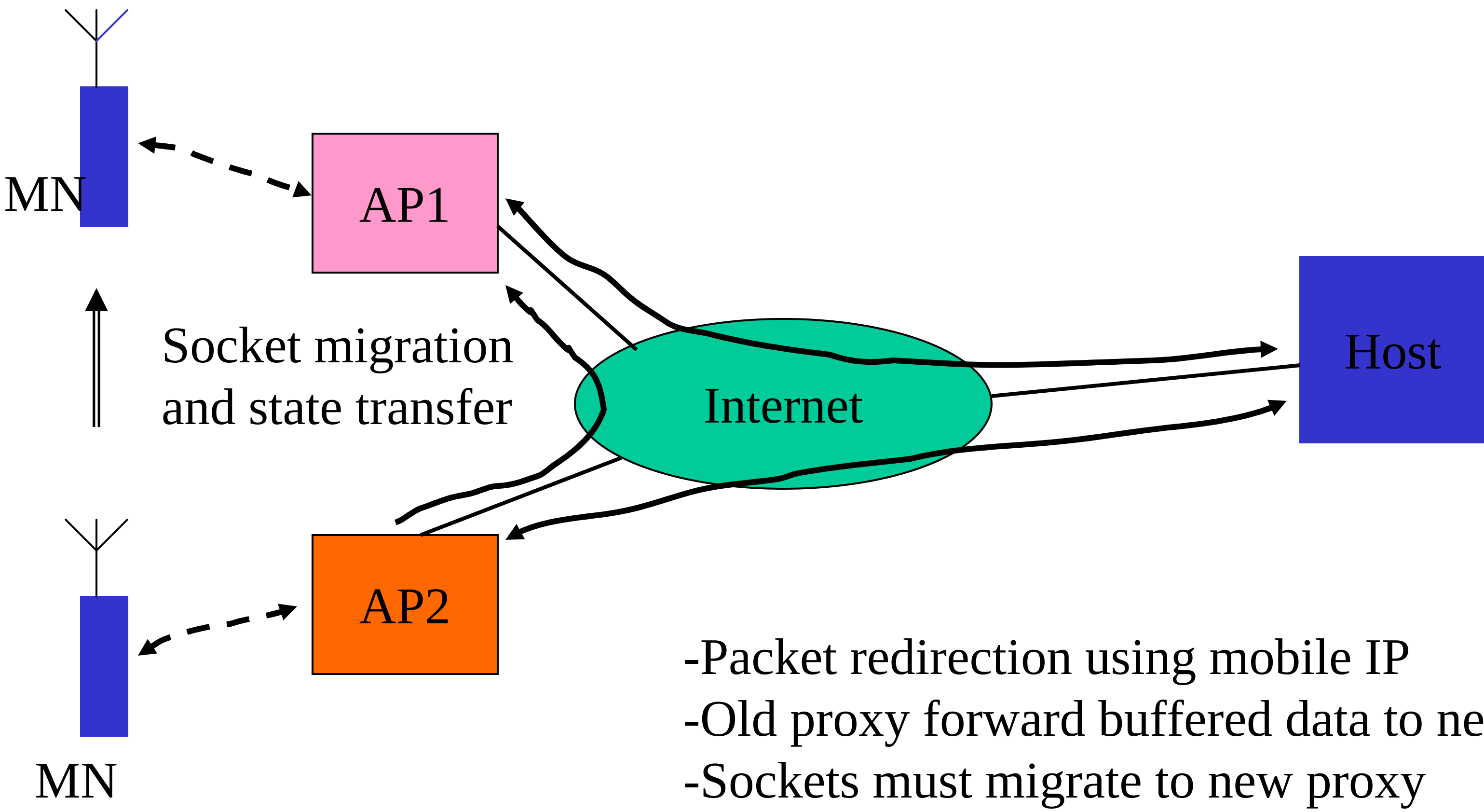
- Indirect TCP
- Snooping TCP
- Mobile TCP
- Mechanisms for better performance
  - Fast retransmit/fast recovery
  - Selective retransmission

# Indirect TCP

- Motivation
  - Poor TCP performance over wireless links
  - No change to TCP within fixed network



# I-TCP handover

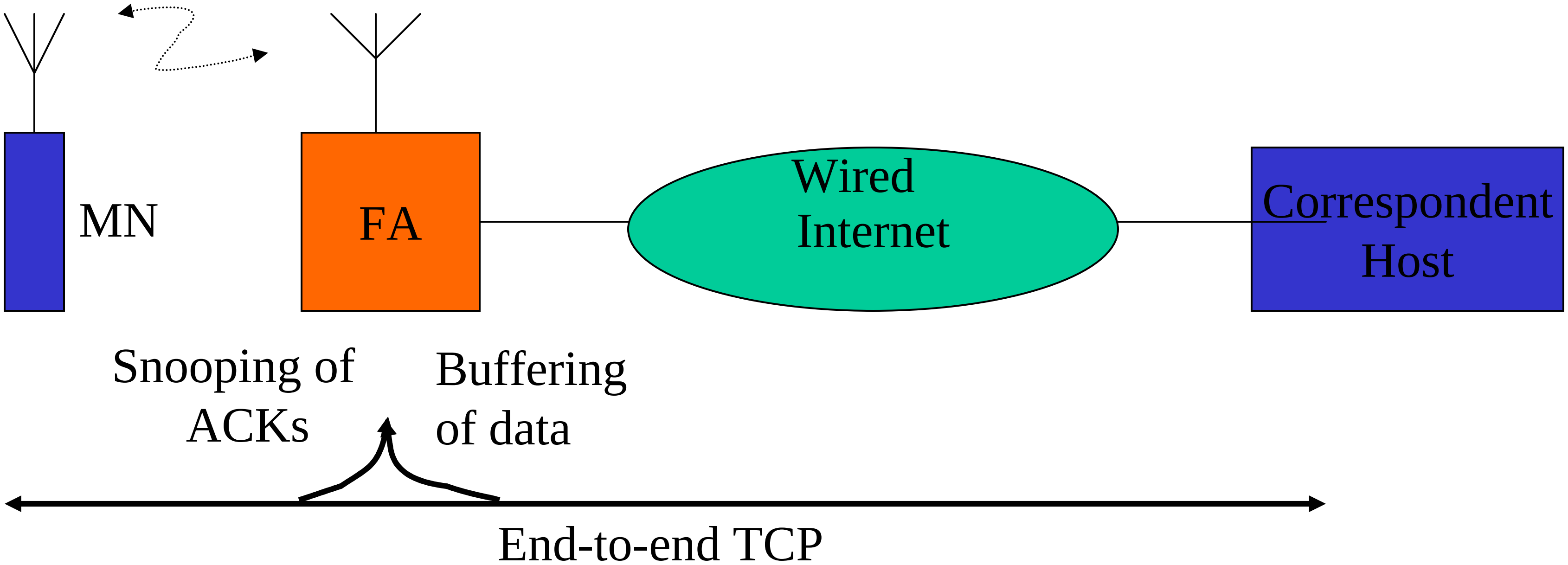



-Packet redirection using mobile IP
-Old proxy forward buffered data to new proxy
-Sockets must migrate to new proxy
    -Seq #, addresses, ports
-No new connection is established

# I-TCP

- Advantages
  - No change to standard TCP
  - No propagation of loss on wireless link to fixed net
  - Wireless TCP can be locally improved
- Disadvantages
  - Loss of end-to-end semantics.

# Snooping TCP




FA: **Buffers** all packets with destination MN (unidirectional)
**Snoops** packet flow in both directions
Does **not** generate ACKs

# Snooping TCP

- Packets with **destination mobile** host
  - Buffered by FA
  - Until an ACK is received from MN
  - FA performs local retrans in case of loss on wireless link
  - Timeout or multiple ACKs received
    - MN retransmits from local buffer
    - Smaller timeout, better performance
  - Discards duplicates from correspondent host

# Snooping TCP

- Packets with dest **correspondent** host
  - FA snoops into the packet stream to **detect gaps**
  - Missing packet detected
    - – Sends a negative ACK (NACK) to MN
  - (Reordering is done by correspondent host)

# Snooping TCP

- Advantages
  - Preservation of end-to-end semantic
  - No handover of state when MN moves to a new FA
  - No forwarding of buffered data
  - Does not matter if the new FA uses the enhancement
- Disadvantages
  - Does not **isolate** the behavior of wireless link as good as I-TCP.
  - NACK: additional mechanism on mobile node
  - Useless if TCP protocol header is encrypted.

# Mobile TCP

- Wireless link
  - (Packet dropping: higher bit error, handover)
  - Lengthy and/or frequent **disconnections**
- Retransmission policy
  - Sender retransmits data controlled by a timer
  - Timeout interval doubles with each unsuccessful retransmission (max: **1 minute**)
  - 12 retransmissions

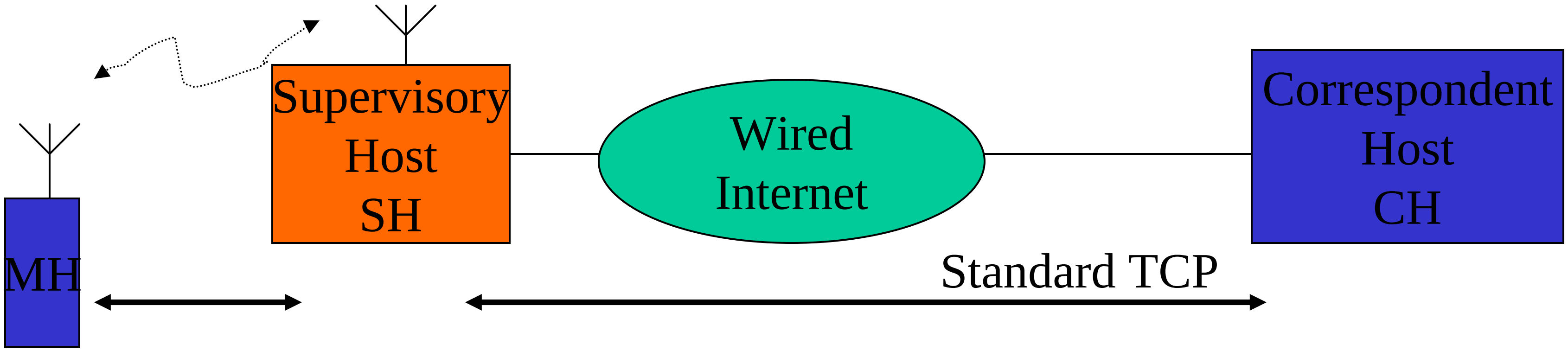
# Mobile TCP

- If connectivity is back before 12th retrans
  - No data is sent for a minute
  - Slow start mode
- I-TCP and disconnection
  - FA buffers a lot of data
  - FA forwards buffered data to new FA …
- Snooping TCP too is ineffective

# Mobile TCP

- Goals
  - I-TCP + S-TCP + adapt to lengthy/freq discon.
  - (Prevent sender window from shrinking)

# Mobile TCP




Assumes low bit error on wireless channel.
No caching/retransmission of data by SH
Lost on wireless link: retransmission by original sender ← end-to-end

SH **monitors** all packets sent to MH and ACKs from MH.
If SH does not see an ACK, it **chokes** the sender (window size = 0)

Window size = 0 → sender goes into **persistent mode (**no state change**)**



**SH detects connectivity:** Reopen sender's window

# Fast retransmit/fast recovery

- Moving to a new FA causes **timeout/loss**
- ➔ TCP goes into **slow start**
- Force TCP into fast retransmit/fast recovery
    - Registration at a new FA ➔ send duplicated ACKs
- Disadvantage
    - Coordination between TCP and IP at MH
    - May result in more duplicates

# Selective retransmission

- TCP ACKs are **cumulative**
- Loss of **one** packet
  - retransmit **all** starting from the lost packet
- RFC 2018 (1996)
  - Request for **selective** retransmission