Mobile Communications

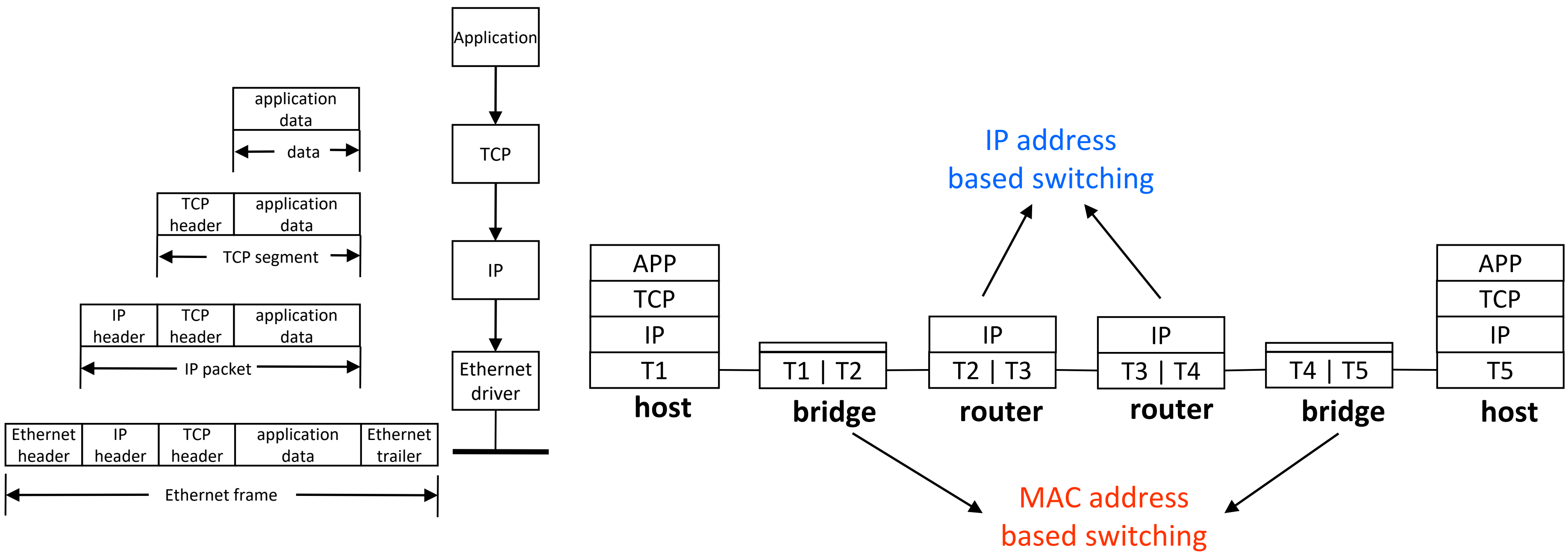
Mobile Networking

Manuel P. Ricardo

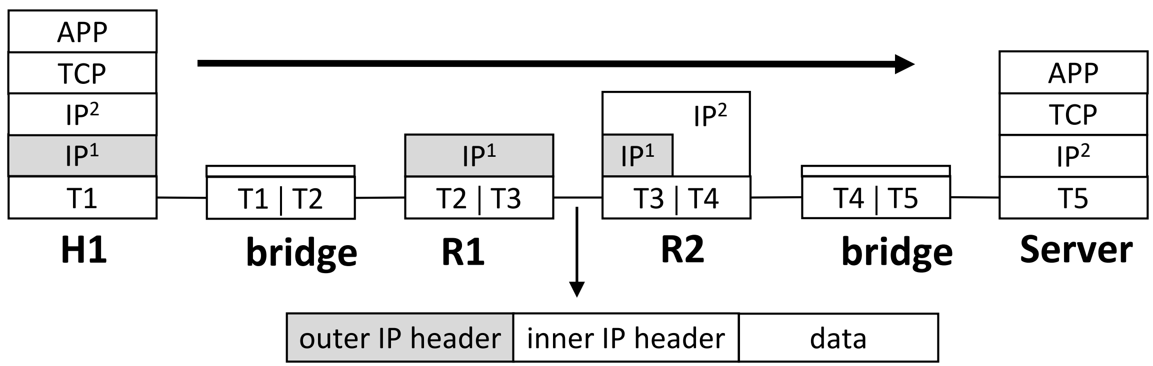
Faculdade de Engenharia da Universidade do Porto

- *What is a tunnel?*
- *How does Mobile IPv6 work?*
- *What are the characteristics mobile ad-hoc networks?*
- *How does AODV work?*
- *How does OLSR work?*
- *What routing metrics are used in wireless networks?*

Traditional TCP/IP Communications Stack

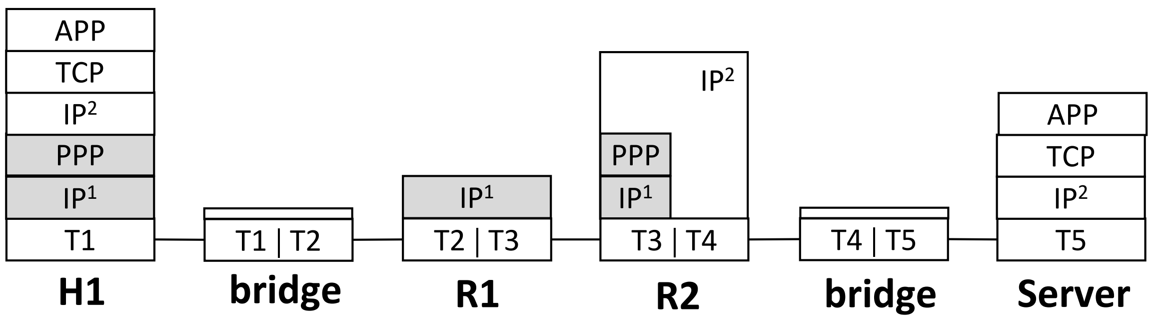


Tunnel IP-in-IP

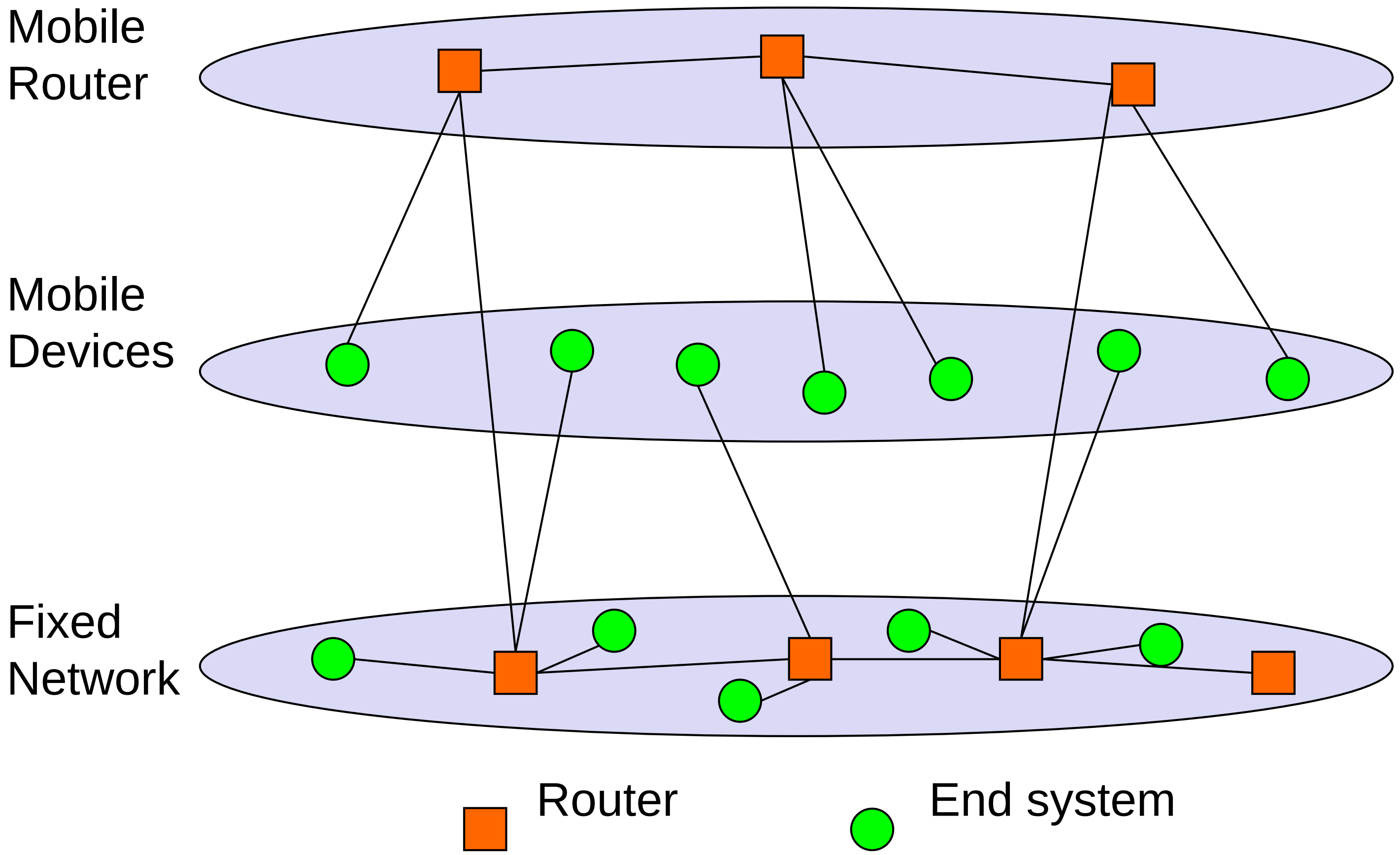


<table>
<tr><td>ver.</td><td>IHL</td><td>TOS</td><td colspan="2">length</td></tr>
<tr><td colspan="3">IP identification</td><td>flags</td><td>fragment offset</td></tr>
<tr><td colspan="2">TTL</td><td>IP-in-IP</td><td colspan="2">IP checksum</td></tr>
<tr><td colspan="5">SA= IP address of H1 (IP^1)</td></tr>
<tr><td colspan="5">DA= IP address of R2 (IP^1)</td></tr>
<tr><td>ver.</td><td>IHL</td><td>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">SA= IP address of H1 (IP^2)</td></tr>
<tr><td colspan="5">DA= IP address of Server (IP^2)</td></tr>
<tr><td colspan="5">TCP/UDP/ ... payload</td></tr>
</table>

Tunnel PPP over IP (E.g PPTP)



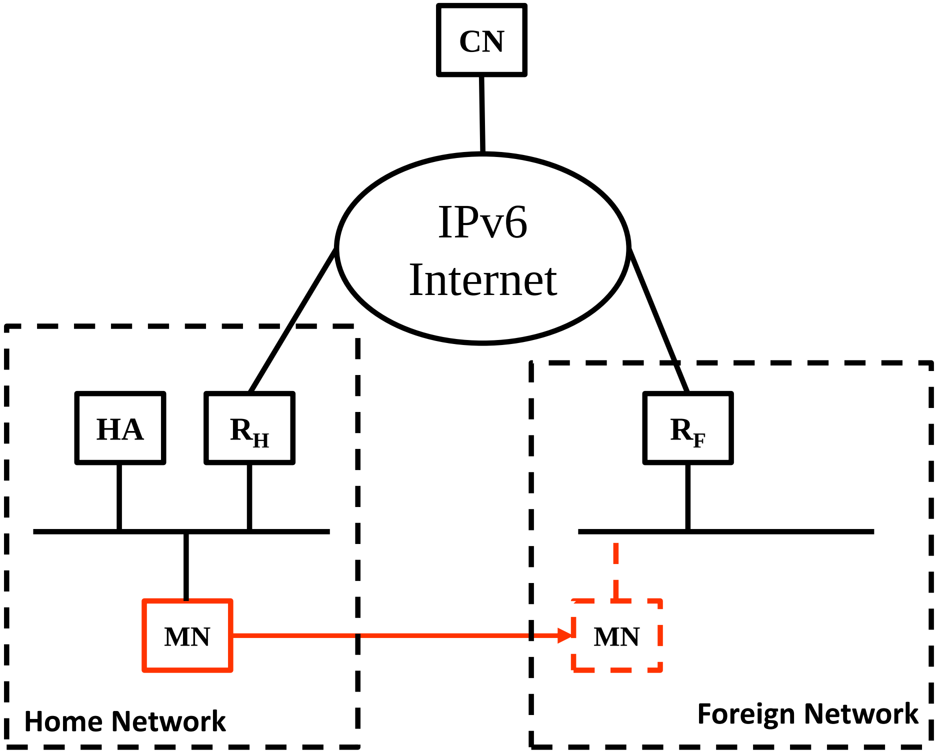
Mobile Networking



Mobile IPv6 (Mobile Devices)

Motivation

How to implement mobility at the IP layer?



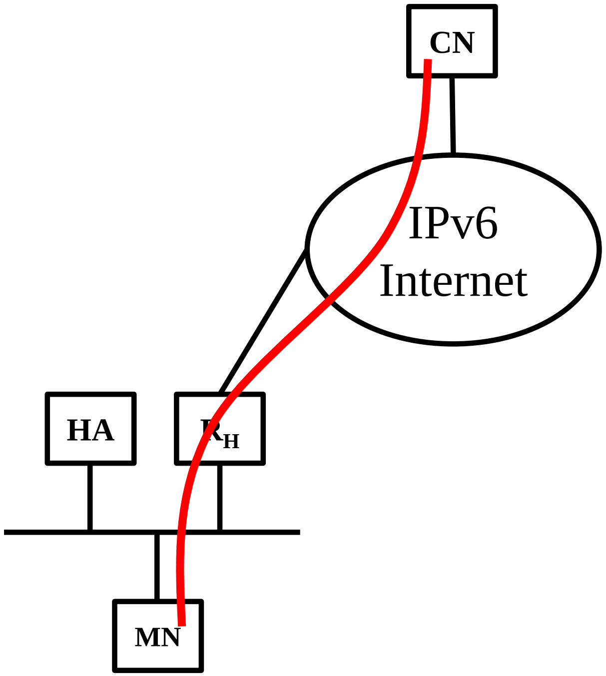
MN - Mobile Node
HA – Home Agent
CN - Correspondent Node
R - Router

Possible Solutions

- DHCP plus dynamic DNS
  - MN in the foreign network
    gets new IP address, uses same name
  - Current TCP connections will break
  - Works with existing Internet

- Mobile IPv6
  - Mobile Node maintains its original IP address
  - Mobile Node gets a second IP address (*care of address*)
  - Enables TCP session continuity
  - Requires mobility aware nodes

MN at Home Network



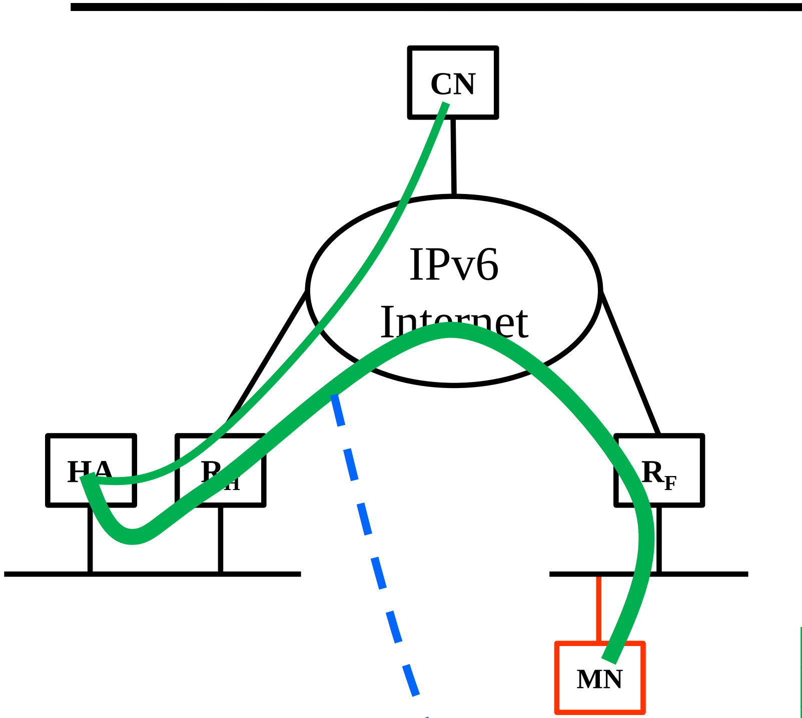
Home Network

ping

```
CN              HA            MN
|echo request|               |
+--------------------------->|
|echo reply   |              |
|<---------------------------+
|             |              |
```

Standard exchange of packets

MN visits a Foreign Network (./..)



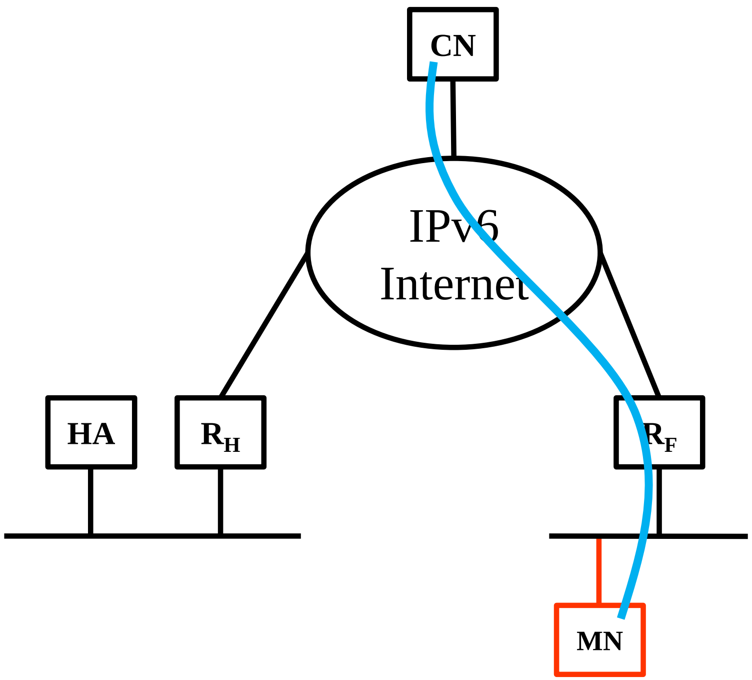

```
CN            HA            MN            RF            MN'
|             |             |             |             |
|             |             |MN moves     |             |
|             |             +-------------------------->|
|             |             |             |radv         |
|             |             |             +------------>|
|             |binding update(CoA)        |             | CoA
|             |<----------------------------------------+
|             |binding ack  |             |             |
|             +---------------------------------------->|
|echo request |             |             |             |
+ -----------+=========================================>|
| echo reply  |             |             |             |
|<-----------+==========================================+
|             |             |             |             |
```

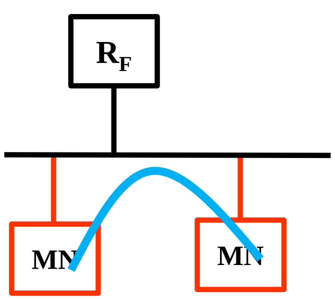
ver.	IHL	TOS	length	
IP identification			flags	fragment offset
TTL		*IP-in-IP*	IP checksum	
IP address of HA				
Care-of address COA				
ver.	IHL	TOS	length	
IP identification			flags	fragment offset
TTL		lay. 4 prot.	IP checksum	
IP address of CN				
IP address of MN				
TCP/UDP/ ... payload				

MN visits a Foreign Network (../..)

- MN acquires a second IP address (Care Of Address)
  - by DHCP or by listening ICMP Router Advertisement message sent by R_F
- MN informs HA about its new address
  - MN sends Binding-Update; HA sends Binding-Acknowledge
  - These are IPv6 messages using a new options mobility header
- HA
  - starts behaving as MN
  - receives traffic to MN@home
  - tunnels this traffic to the CoA of MN
- MN sends traffic to HA, using the tunnel

MN optimizes the Route to CN (./..)






```
CN            HA            MN            RF          MN'
|             |             |             |            |
|echo request|              |             |            |
+ -----------==========================================>|
|echo reply   |             |             |            |
|<------------|========================================+
|             |             |             |            |
|home test init             |             |            |
|<------------|========================================+
|care of test init          |             |            |
|<-----------------------------------------------------+
|care of test|              |             |            |
+----------------------------------------------------->|
|home test    |             |             |            |
+-------------|=======================================>|
|binding update             |             |            |
|<-----------------------------------------------------+
|binding ack                |             |            |
+----------------------------------------------------->|
|echo request|              |             |            |
+----------------------------------------------------->|
|  echo reply |             |             |            |
|<-----------------------------------------------------+
|             |             |             |            |
```

MN Optimizes the Route to CN (../..)

- MN detects packet received in tunnel
- Optionally, it decides to optimize the route to the CN

- MN informs CN about its new address
  - MN sends Binding-Update; CN sends Binding-Acknowledge
  - These are IPv6 messages using a new options mobility header

- Traffic starts to be exchanged directly between MN←→CN
  - MN→CN: use of options destination header
  - CN→MN: use of option routing header

Route Optimization

- IPv6 packets in the CN → MN direction
  - CN
    - Before sending a packet to MN, reads its Bindings cache
    - Is there is no entry → packet sent as usual
    - If there is an entry
      - Sends packet to *CareOfAddress* (IPv6 destination address = *CareOfAddress*)
      - Includes in the packet a *RoutingHeade*r having 2 hops (list of addresses to be visited by packet)

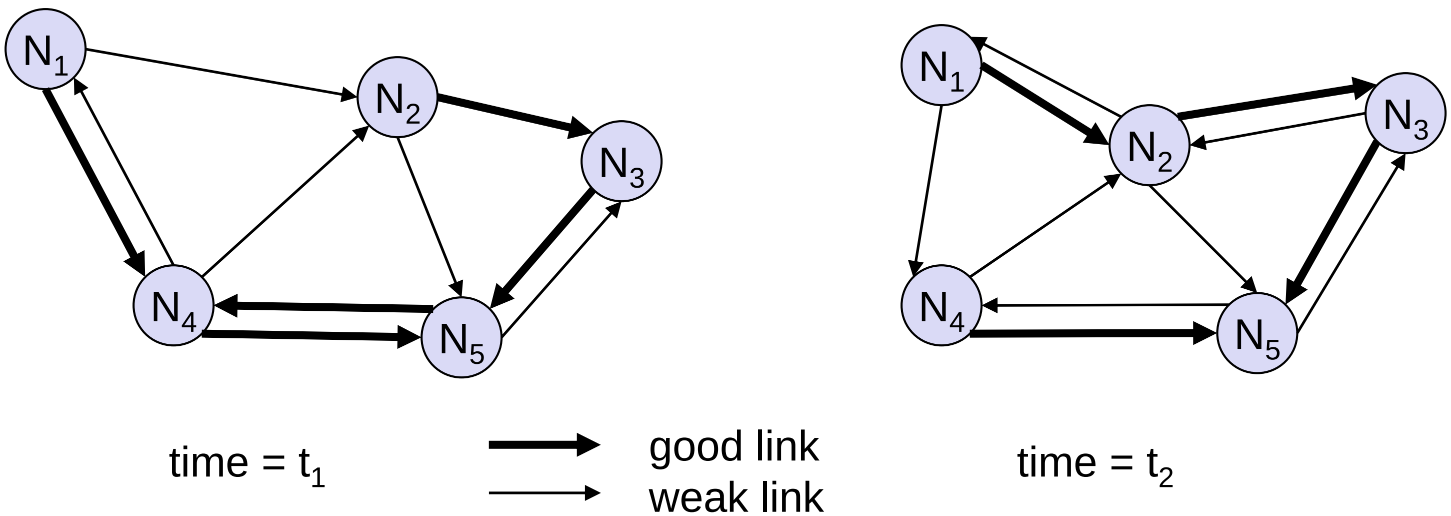
        1º hop → CareOfAddress; 2º hop → MN HomeAddress
  - MN
    - Receives packet in CareOfAddress
    - Forwards packet to itself (MN home address)
- IPv6 packets in the MN → CN direction
    - Source address = *CareOfAddress*
    - Inclusion of *DestinationHeader* with information about *HomeAddress*
    - CN replaces *HomeAddress* in the packet source address

      so that the socket structure may contain the correct information → *HomeAddress*

```
CN           HA           MN           RF         MN'
|echo request|            |            |          |
+------------------------>|            |          |
|echo reply  |            |            |          |
|<------------------------+            |          |
|            |            |MN moves    |          |
|            |            +---------------------->|
|            |            |            |radv      |
|            |            |            +--------->|
|            | binding update          |          |
|            |<-------------------------------------+
|            |binding ack |            |          |
|            +----------------------------------->|
|echo request|            |            |          |
+ -----------=====================================>|
|echo reply  |            |            |          |
|<-----------|======================================+
|            |            |            |          |
|home test init           |            |          |
|<-----------|======================================+
|care of test init        |            |          |
|<--------------------------------------------------+
|care of test|            |            |          |
+------------------------------------------------->|
|home test   |            |            |          |
+------------|====================================>|
|binding update           |            |          |
|<--------------------------------------------------+
|binding ack              |            |          |
+------------------------------------------------->|
|echo request|            |            |          |
+------------------------------------------------->|
| echo reply |            |            |          |
|<--------------------------------------------------+
|            |            |            |          |
```

Ad-hoc Networks (Mobile Routers)

Characteristics of Ad-hoc Networks



Ad-hoc network

» Dynamic topology ← moving nodes

» Link characteristics vary along the time

» Asymmetric links

Routing in Ad-hoc Networks

- Conventional routing protocols

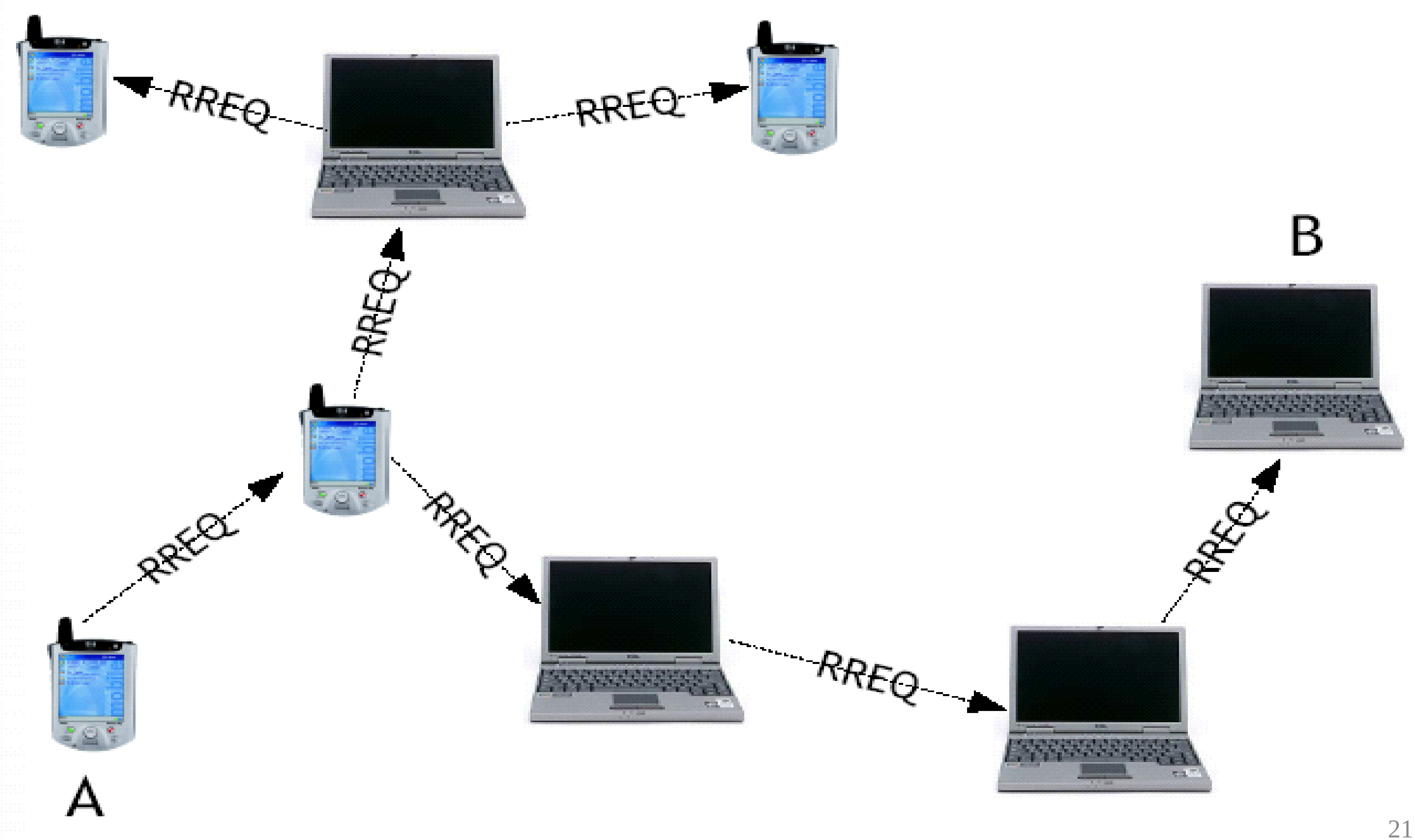
  built for wired networks → topology varies slowly

- In Ad-hoc networks
  - » Dynamic topology → information required to be refreshed more frequently
  - » Wireless node may have scarce resources (bandwidth, energy) …

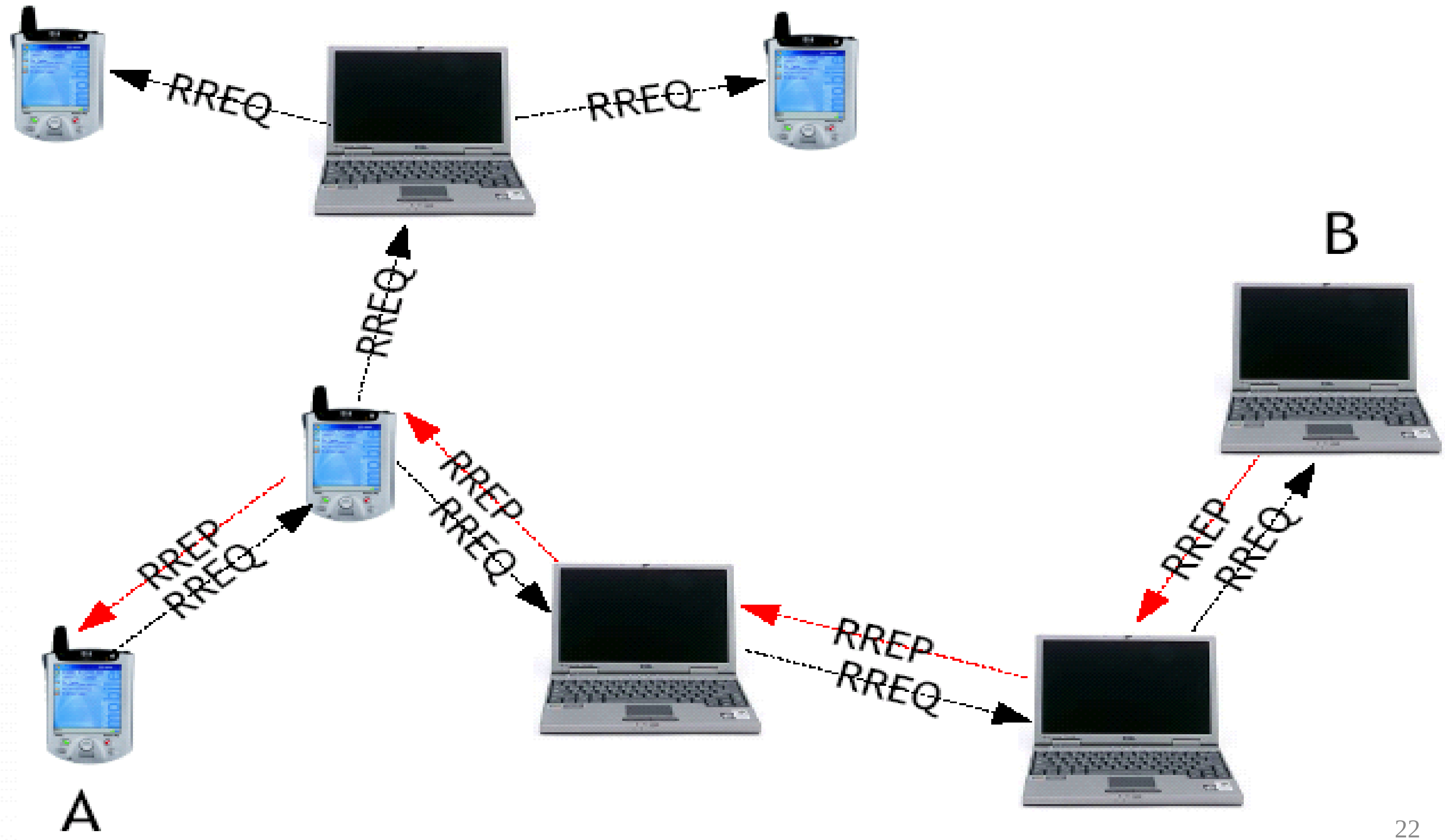
- New routing strategies required for ad-hoc networks
  - » Reactive - example: AODV
  - » Pro-active - example: OLSR

AODV – A needs to send data packet to B

AODV – A sends RouteRequest



AODV – B replies with RouteReply



AODV - Characteristics

- *AODV: Ad hoc On-Demand Distance Vector* routing protocol

- Decision to find a route: when a packet has to be sent
- Broadcast of *Route-request*
- Intermediate nodes get routes to node A
- *Route-reply* sent *unicast* through same path of received *Route-request*
- Intermediate nodes get also route to node B
- Routes have *Time-to-live*, in every node
- Needs symmetric graph

Pro-active Routing Protocols

- Routes built using continuous control traffic
- Routes are maintained

- Advantages, disadvantages
  - Constant control traffic ➔ radio resources consumed with signaling
  - Routes always available

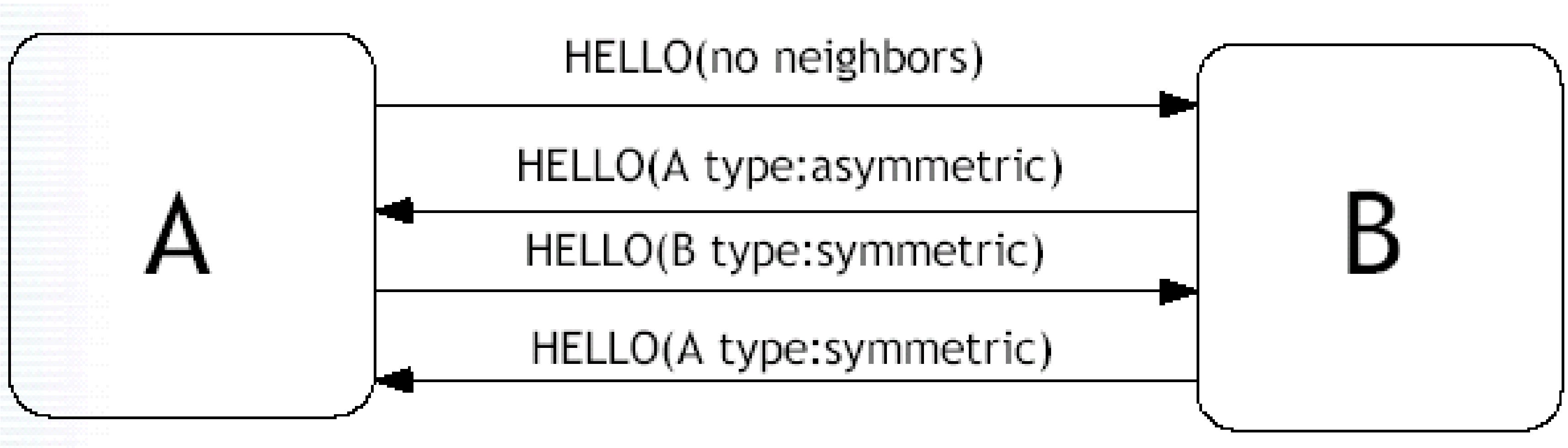
- Example – OLSR (RFC 3626)
  - OLSR - Optimized Link-State Routing protocol

OLSR – Main functions

- Detection of links to neighbour nodes
- Optimized forwarding / flooding (MultiPoint Relaying)

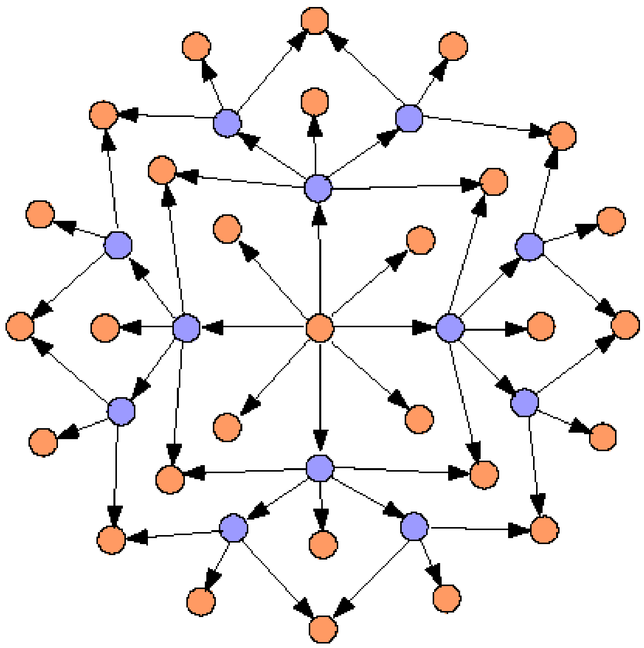
OLSR – Detecting Links to Neighbor Nodes

- Using *HELLO* messages
- All nodes transmit periodically *HELLO* messages
- *HELLO* messages group neighbour by their state



OLSR – MultiPoint Relaying (MPR)

- MultiPoint Relaying (MPR)
  - » Special nodes in the network
  - » Used to
    - – Limit number of nodes retransmitting packets
    - – Reduce number duplicated retransmissions



- Each node selects its MPRs, which must
  - » Be at 1 hop distance
  - » Have symmetric links to the node
- Set of MPRs selected by a node
  - » Must be minimum
  - » Must enable communication with every 2-hop-away nodes
- Node is MPR if it is selected by other node

OLSR – Link State

- OSPF (*Open Shortest Path First*), in wired networks
  - » Every node floods the network with information about its links state (neighbors)
  - » By receiving this information from the other nodes each node calculates its shortest path tree to the other nodes in the network

- OLSR does the same, using 2 optimizations
  - » Only the MPR nodes send link state messages (Topology Control Messages)

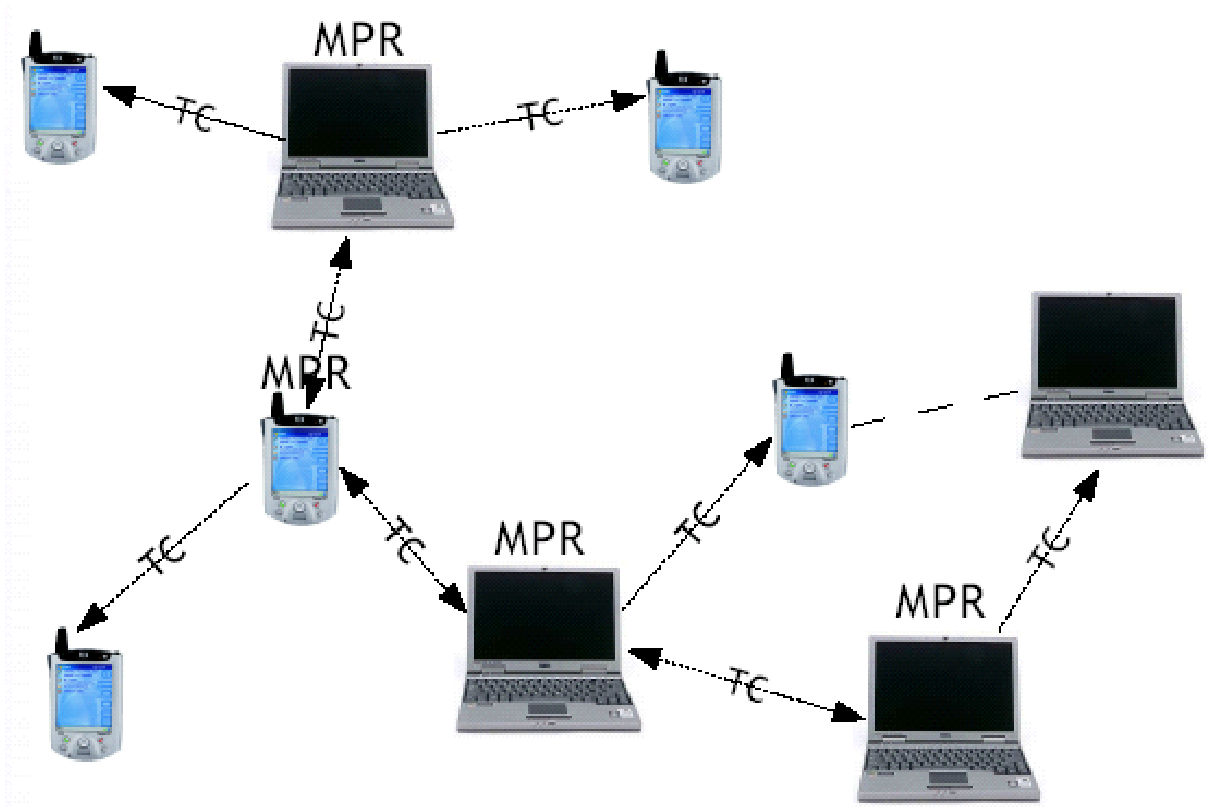
    ➔ Smaller number of nodes sending signaling messages
  - » Only nodes associated to MPR are declared in link state message

    ➔ Reduced message length

OLSR – Link state, example

- Messages which declare the links state

  Topology Control Messages (TC)

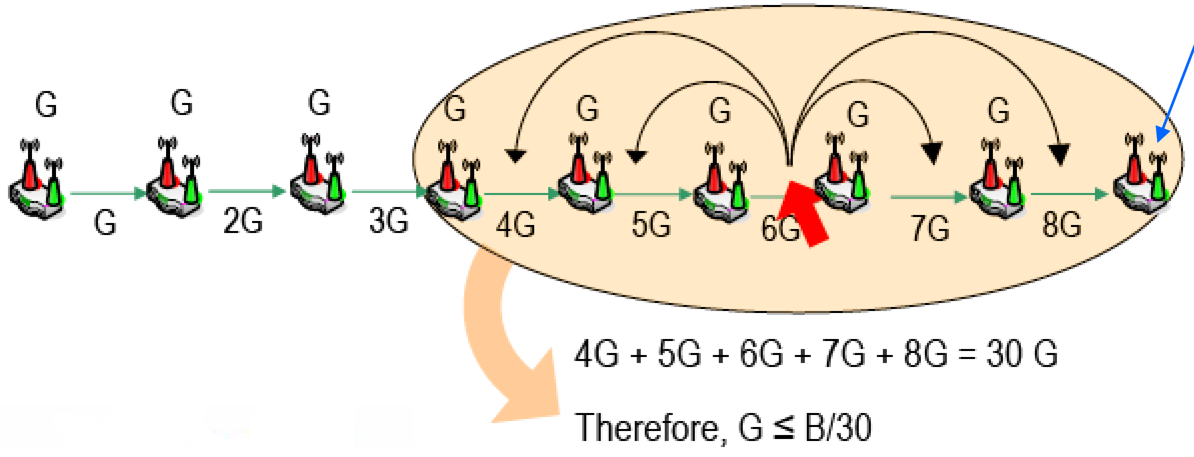


Routing metrics

(assuming layer 2 adopts CSMA/CA and
all the nodes operate in the same radio channel)

Multiple traffic flows in a shared medium

- Assume all nodes send same traffic flow of G bit/s towards GW
- **B** (bit/s) is the capacity of wireless medium
- Forwarding implies that a frame is transmitted multiple times
- In a collision domain only one frame at time is transmitted



Routing Metric for Wireless Channels – Expected Transmission Count (ETX)

- *ETX* – mean number of times a data frame will be transmitted

- Successful frame transmission probabilities, in a link
  - » S_f: probability that DATA frame is successfully received
  - » S_r: probability that ACK packet is successfully received

- $$ETX = \frac{1}{S_f \times S_r}, \quad ETX \in [1, +\infty[$$

- Example: $S_f=0.6$, $S_r=0.5$ → $ETX=3,3$

- Shortest path between two nodes
  - » path having the minimum sum of ETXs

Routing metrics for Wireless Channels - Expected Transmission Time (ETT)

- *ETT* – mean time required to transmit a frame with success
- Improves ETX by considering also link transmission bitrate R
- Packet size: *S*, Link transmission bitrate: *R*
- $ETT=ETX*S/R$ *(sec)*
- Shortest path between two nodes
  - Path having the minimum sum of *ETT*s

IEEE 802.11s – Airtime Link Metric

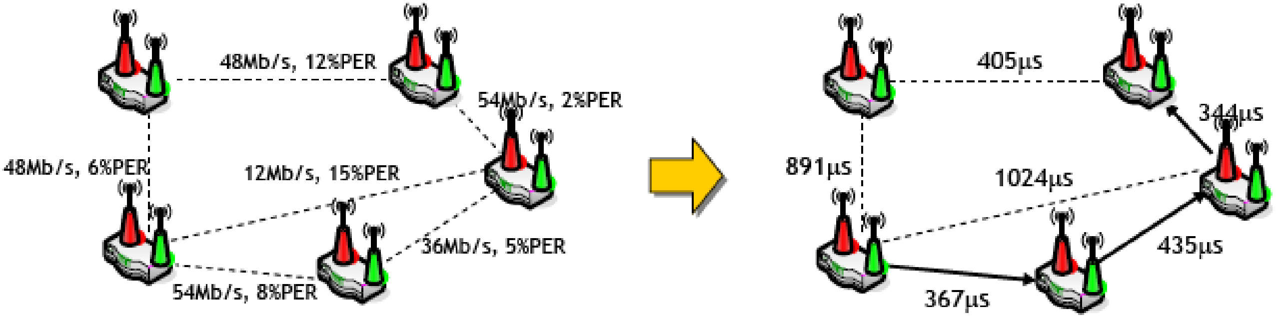
802.11s default routing metric: Airtime link Cost

- Amount of time c_a required to transmit a frame

$$c_a = \left[O_{ca} + O_p + \frac{B_t}{r} \right] \times \frac{1}{1 - e_{pt}}$$

- r = transmission bitrate
- e_{pt} = frame error ratio

Parameter	Value (802.11a)	Value (802.11b)	Description
O_{ca}	75μs	335μs	Channel access overhead
O_p	110μs	364μs	Protocol overhead
B_t	8224	8224	Number of bits in test frame



IPv6

The Need of a New IP

- IPv4
  - Small addressing space (2^{32} bits)
  - Solutions used to overcome this problem:

    private networks (NAT), classless networks (CDIR)

- IETF developed new IP version: **IPv6**
  - Same principles of IPv4
  - Improvements
  - Header re-defined

IPv6 – Improvements

- 128 bit addresses (16 octets, 8 *shorts*). No classes
- Better QoS support (flow label)
- Native security functions (peer authentication, data encryption)
- Autoconfiguration (*Plug-n-play)*

Address Representation

- 8 x 16 bit, hexadecimal. Separated by **:**

  47CD : 1234 : 3200 : 0000 : 0000 : 4325 : B792 : 0428

- Compressed format: **FF01:0:0:0:0:0:0:43 → FF01::43**

- Loopback address: **::1**

- Network prefix described by **/** , same as IPv4
  - » **FEDC:BA98:7600::/40 → network prefix = 40 bits**

Addresses – Link-Local, Global Unicast, Anycast

- Link-Local
  - » Used for communication between hosts in the same LAN /link
  - » Address built from network interface MAC address
  - » Routers do not foward packets having a Link-Local destination address
- Global Unicast
  - » Global addresses
  - » Address: network prefix + computer identifier
  - » Structured prefixes
- Multicast
  - » Group address; packet received by all the members of the group
- Anycast
  - » Group address; packet is received by any (only one, the closest) member of the group

Address Formats

n bits	m bits	128-n-m bits
001 global rout prefix	subnet ID	interface ID

Global Unicast address (2000::/3)

10 bits	54 bits	64 bits
1111111010	0	interface ID

Link-Local Unicast address (fe80::/10)

8 bits	4	4	112 bits
11111111	flags	scope	group ID

Multicast Address
Scope – link, global
(ff::/8)

n bits	128-n bits
subnet prefix	00000000000000

Anycast address

Link Local Address - Identifier IEEE EUI-64

Method to create a IEEE EUI-64 identifier from an IEEE 48bit MAC identifier.
This is to insert two octets, with hexadecimal values of 0xFF and 0xFE,
in the middle of the 48 bit MAC (between the company_id and vendor supplied id).
For example, the 48 bit IEEE MAC with global scope:

```
+----------------+----------------+----------------+
|cccccc0gcccccccc|ccccccccmmmmmmmm|mmmmmmmmmmmmmmmm|
+----------------+----------------+----------------+  00:C0:DF:08:D5:99
```

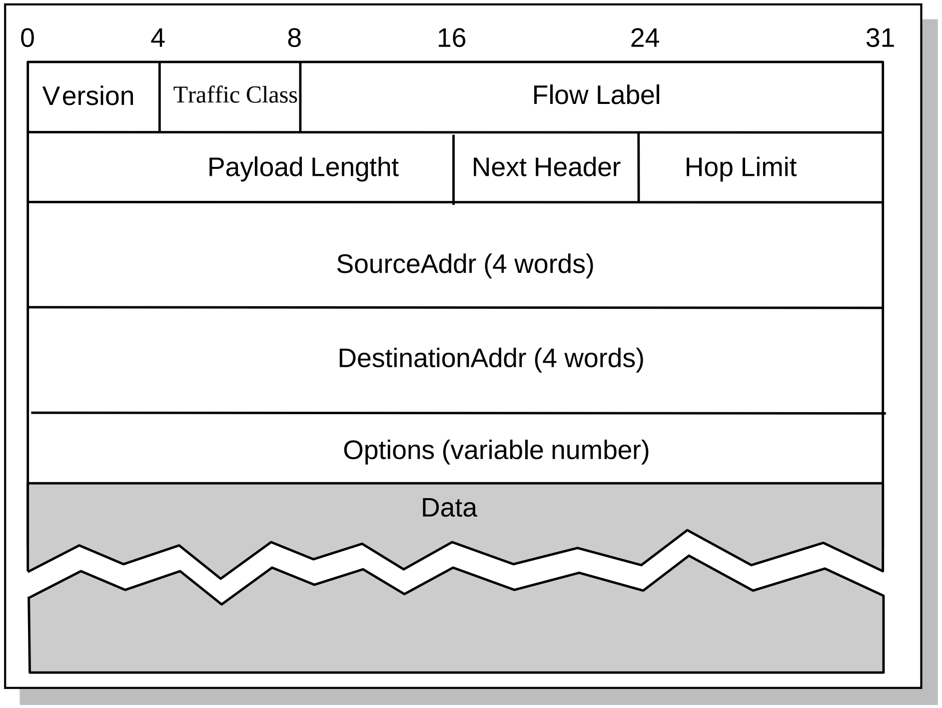
where "c" are the bits of the assigned company_id, "0" is the value of the universal/local bit to indicate global scope, "g" is individual/group bit, and "m" are the bits of the manufacturer-selected extension identifier.
The interface identifier would be of the form:

```
+----------------+----------------+----------------+----------------+
|cccccc1gcccccccc|cccccccc11111111|11111110mmmmmmmm|mmmmmmmmmmmmmmmm|
+----------------+----------------+----------------+----------------+

                                                fe80::2c0:dfff:fe08:d599
```

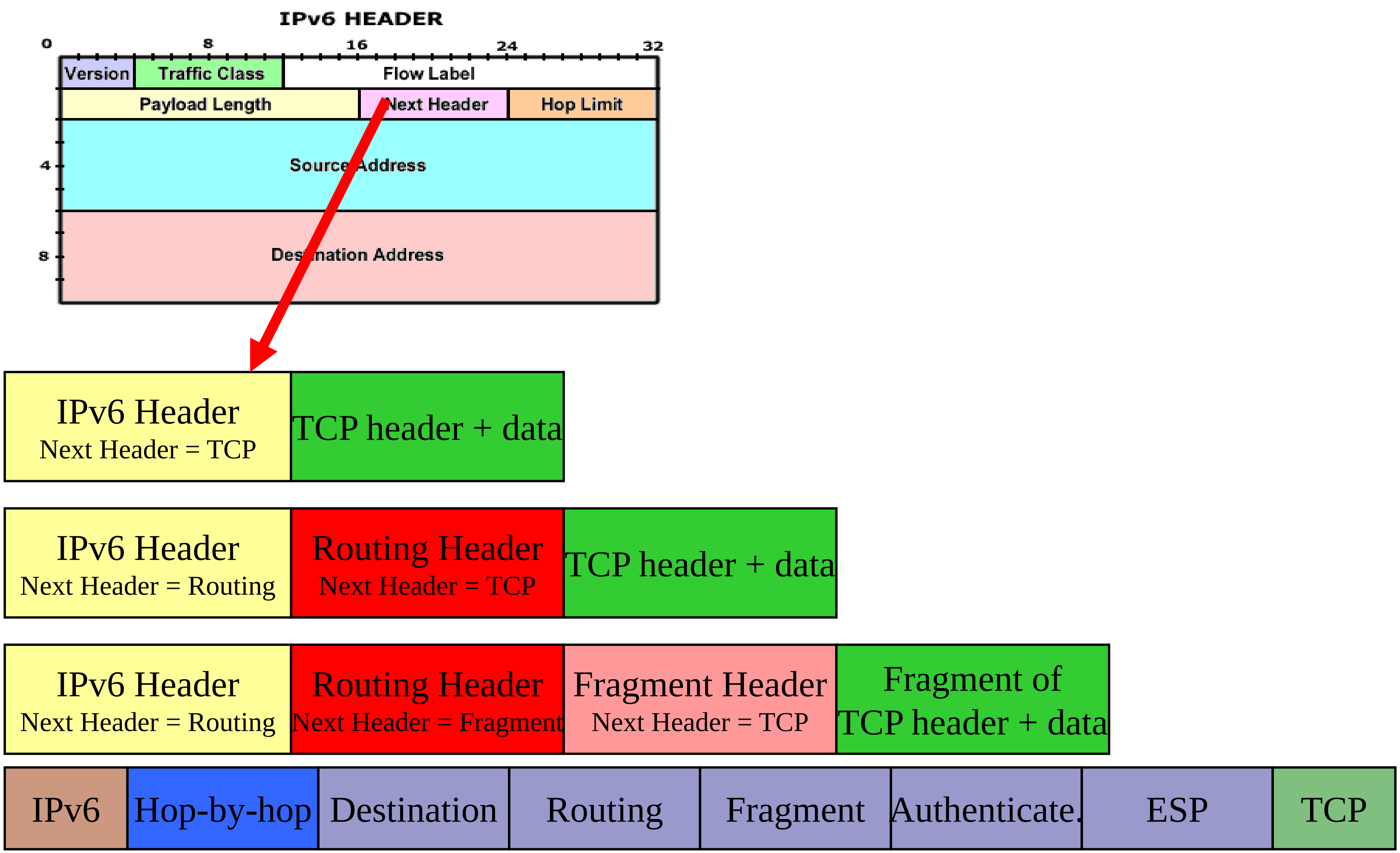
IPv6 Header

- Flow label → identifies packet flow
  - » QoS, resource reservation
  - » Packets receive same service



- Payload length
  - » Header not included
- Hop limit = TTL in IPv4
- Next header
  - » Identifies next header/extension header
- Options → included as extension headers

Extension Headers



Extension Headers

- Hop-by-hop
  - additional information, inspected by every node traversed by the packet
  - other extension headers inspected only at the destination/pre-defined nodes
- Destination
  - information for the destination node
- Routing
  - list of nodes to be visited by the packet
- Fragmentation
  - made by the source, that must also find PMTU (Path Maximum Transmission Unit)
- Authentication
  - signature of packet header
- ESP
  - data encryption

Neighbor Discovery (ND) Protocol

- IPv6 node uses ND protocol to
  - » Find other nodes in the same link /LAN
  - » Find a node's MAC address ➔ ND substitutes ARP
  - » Find router(s) in its network
  - » Mantaining information about neighbour nodes

- ND protocol is similar to the IPv4 functions
  - » ARP IPv4
  - » ICMP Router Discovery
  - » ICMP Redirect

ND Messages

- ICMP messages (over IP); use *Link Local* addresses
- ***Neighbor Solicitation***
  Sent by a host to obtain MAC address of a neighbour / verify its presence
- ***Neighbor Advertisement:*** Answer to the request (solicitation)
- ***Router Advertisement***
  Information about the network prefix; periodic or under request
  Sent by router to IP address *Link Local multicast*
- ***Router Solicitation:*** host solicits from router a *Router Advertisment* message
- ***Redirect***: Used by a router to inform an host about the best route to a destination

Homework

- Review slides
- Read books
  - » From Schiller: Chap. 8 - Mobile Network layer
  - » From Garg: Chap. 14 - Mobile Network and Transport Layer
- Pay attention. The previous books
  - » Discuss mainly Mobile IPv4 while we have discussed Mobile IPv6

    (Mobile IPv6 has does not use a Foreign Agent (FA) and it takes advantage of the Neighbor Discovery protocol)
- Read relevant RFCs
  - » RFC 2460: Internet Protocol, Version 6 (IPv6) Specification
  - » RFC 4861: Neighbor Discovery for IP Version 6 (IPv6)
  - » RFC 6275: Mobility Support in IPv6
  - » RFC 3561: AODV
  - » RFC 3626: OLSR
- Answer questions at moodle