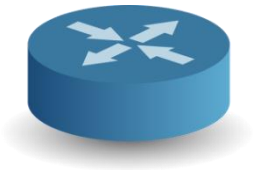
CMPN415
Computer Networks

Part FOUR

Internetworking – Quality of Service

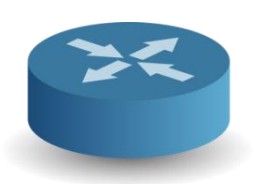


Most Important Slides

Internetworking

Internetworking joins multiple, different networks into a single larger network

- How networks differ
- How networks can be connected / Tunneling
- Fragmentation



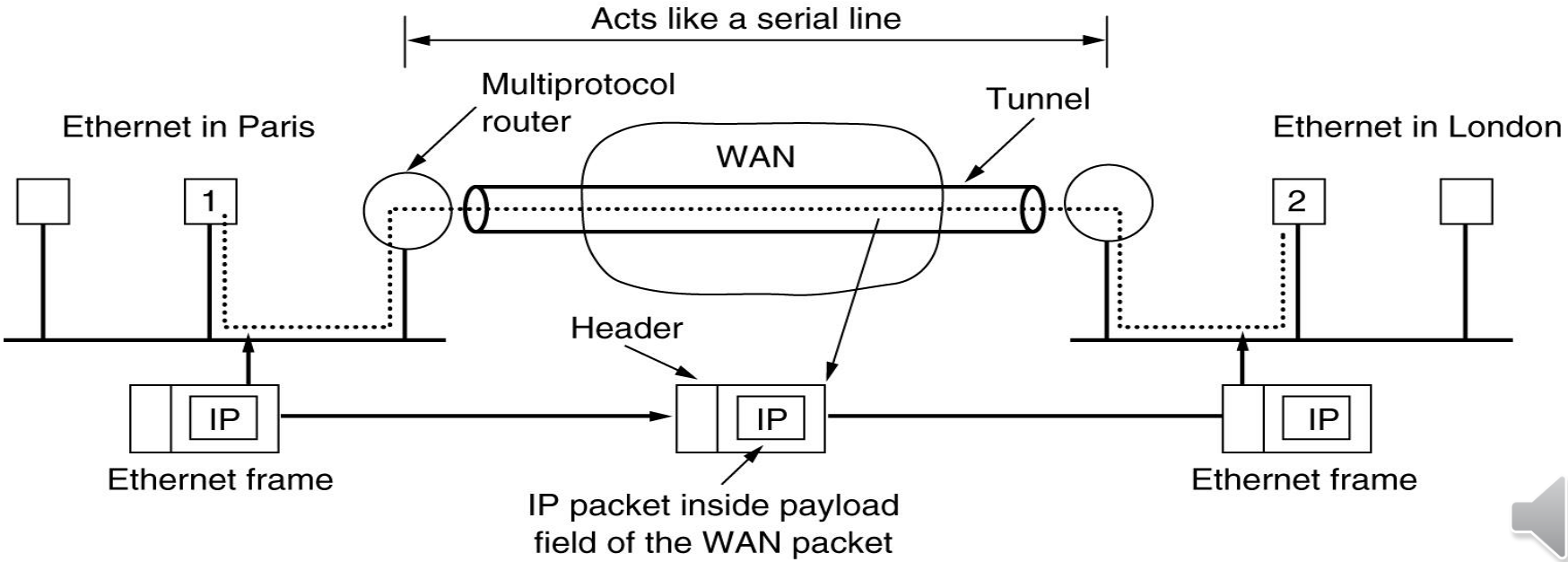
How Networks Differ

Differences can be large; complicates internetworking

Item	Some Possibilities
Service offered	Connectionless versus connection oriented
Addressing	Different sizes, flat or hierarchical
Broadcasting	Present or absent (also multicast)
Packet size	Every network has its own maximum
Ordering	Ordered and unordered delivery
Quality of service	Present or absent; many different kinds
Reliability	Different levels of loss
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, packet, byte, or not at all

Tunneling

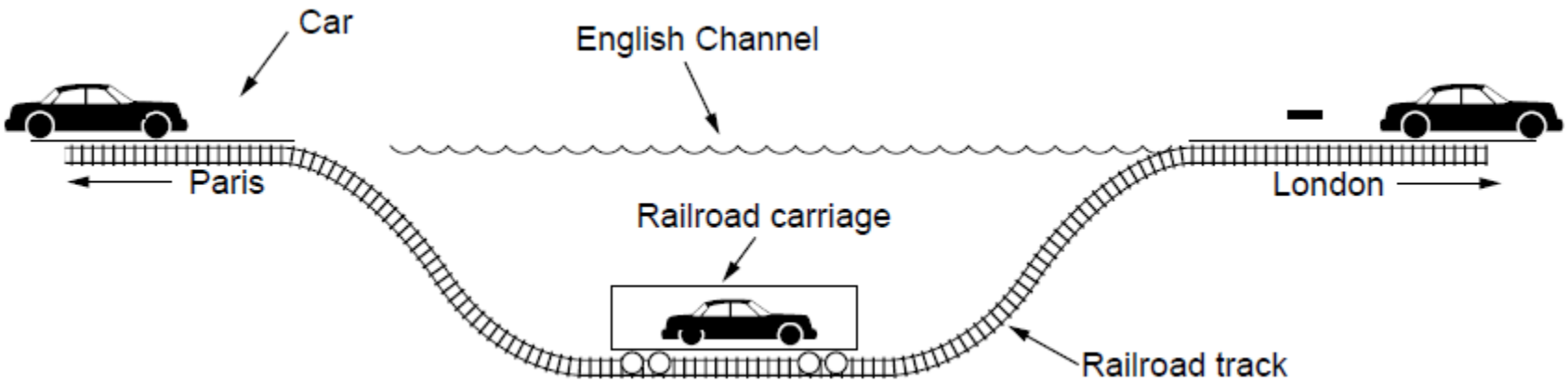
- Tunneling is encapsulating one packet into another
- Very *versatile. Many applications*
  - *E.g. Mobility, security, inter-networking,…, etc*
- Only MP router needs to understand both protocols
- Works if endpoint networks are similar



Tunneling

Tunneling analogy:

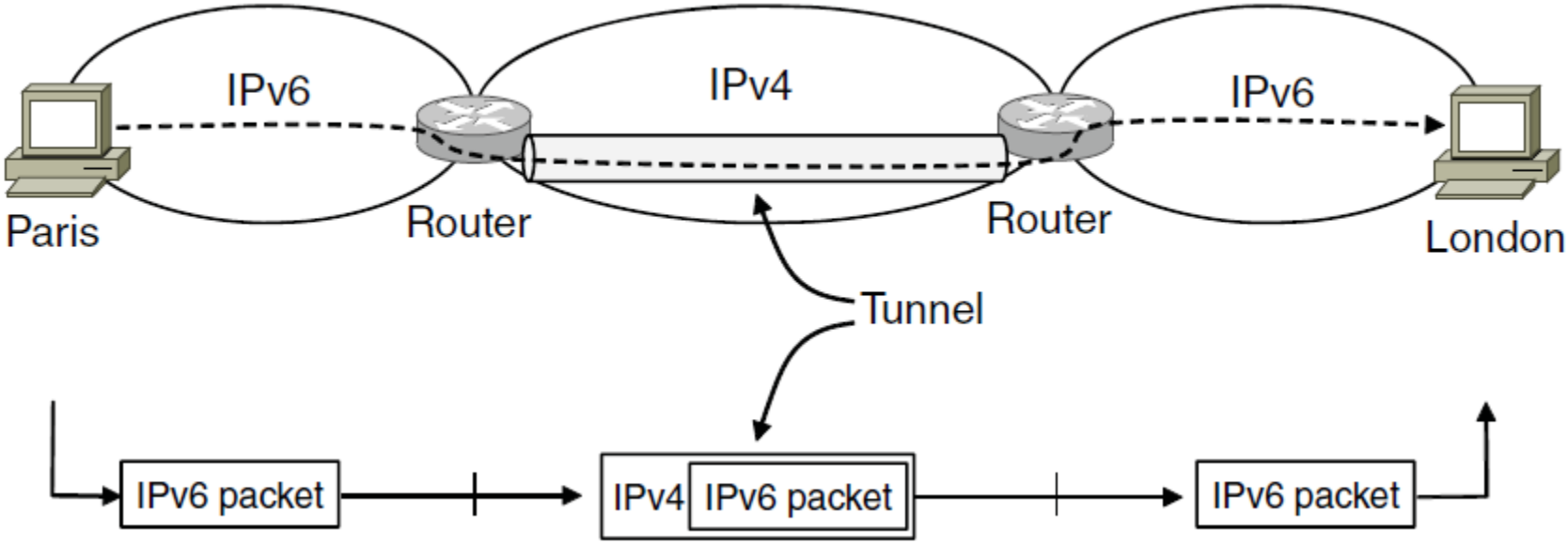
- tunnel is a link; packet can only enter/exit at ends



Tunneling

Connects two networks through a middle one

- Packets are encapsulates over the middle



Fragmentation

- Different Networks have different maximum Transmission Units (MTUs). Example

MTU of layer **n**

- It is the largest **payload** size carried by the packet/frame at **layer n**
- Example
  - Ethernet is L2 protocol. The largest payload is 1500. Hence the L2 MTU is 1500
  - IPv4 packet transmitted inside Ethernet requires a minimum of 20 bytes header. Hence the L3 MTU in this case in 1480

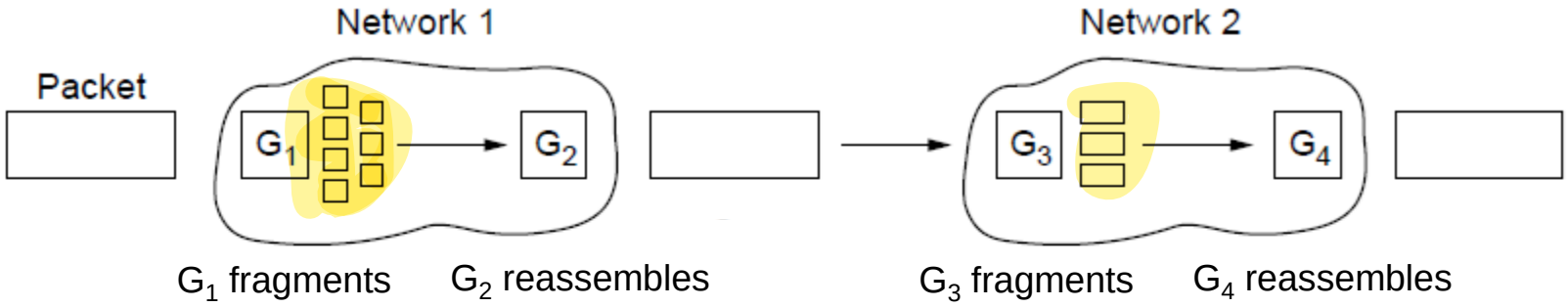
Possible solutions

- Route only through networks that can accommodate source MTU
  - Problem: Destination network may have a smaller MTU
  - Problem: The only available path may have smaller MTU
- Find the smallest MTU along the path and force the source to use the smallest MTU. Problems:
  - Path may change
  - All routers need to agree on an MTU discovery protocol
  - Inefficiency
- Allow intermediate routers to **fragment** packets
  - Re-assembly is not simple

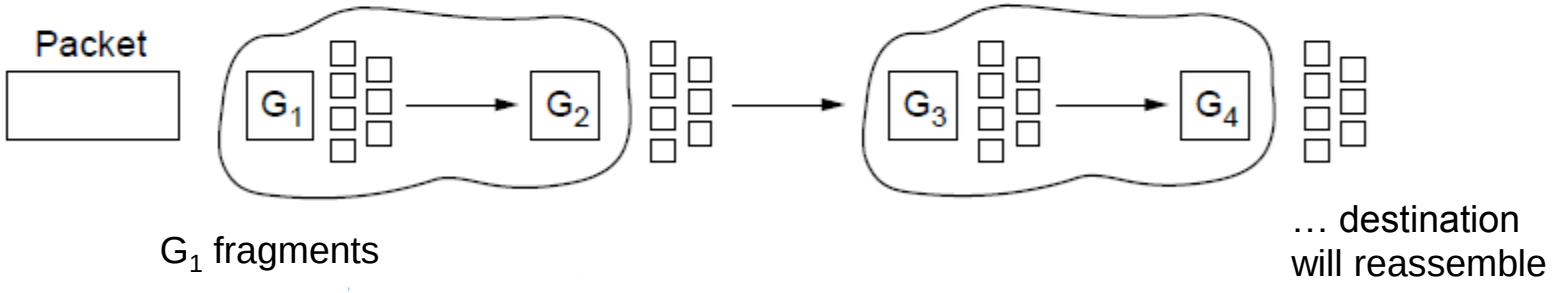
Packet Fragmentation

Networks have different packet size limits for many reasons

- Large packets sent with fragmentation & reassembly



Transparent – packets fragmented / reassembled in each network



Non-transparent – fragments are reassembled at destination

Packet Fragmentation

Transparent

- Advantage
  - Simple from point of view of end points
  - Hosts need not have re-assembly functionality
  - No need for complexity of fragmenting already fragmented packets
- Disadvantage
  - Must use the same exit point for all fragments. May be impossible to connectionless networks
  - Need to add reassembly functionality to *routers* ⇒ complexity
  - Overhead of successive re-assembly and then re-fragmenting

Non-transparent

- Advantage
  - Routers need not re-assemble ⇒ Simpler routers
  - Different fragments can use different paths
- Disadvantage
  - Little bit extra complexity to re-fragment fragments
  - All endpoints need to have re-assembly functionality
  - Overhead of small fragments pass through large MTU networks

Reassembly

Reassembly is harder than fragmentation

- Fragments may arrive out of order
- Fragments may be lost

We need to tag packets to be able to reassemble them correctly

Hierarchical (tree-like) numbering

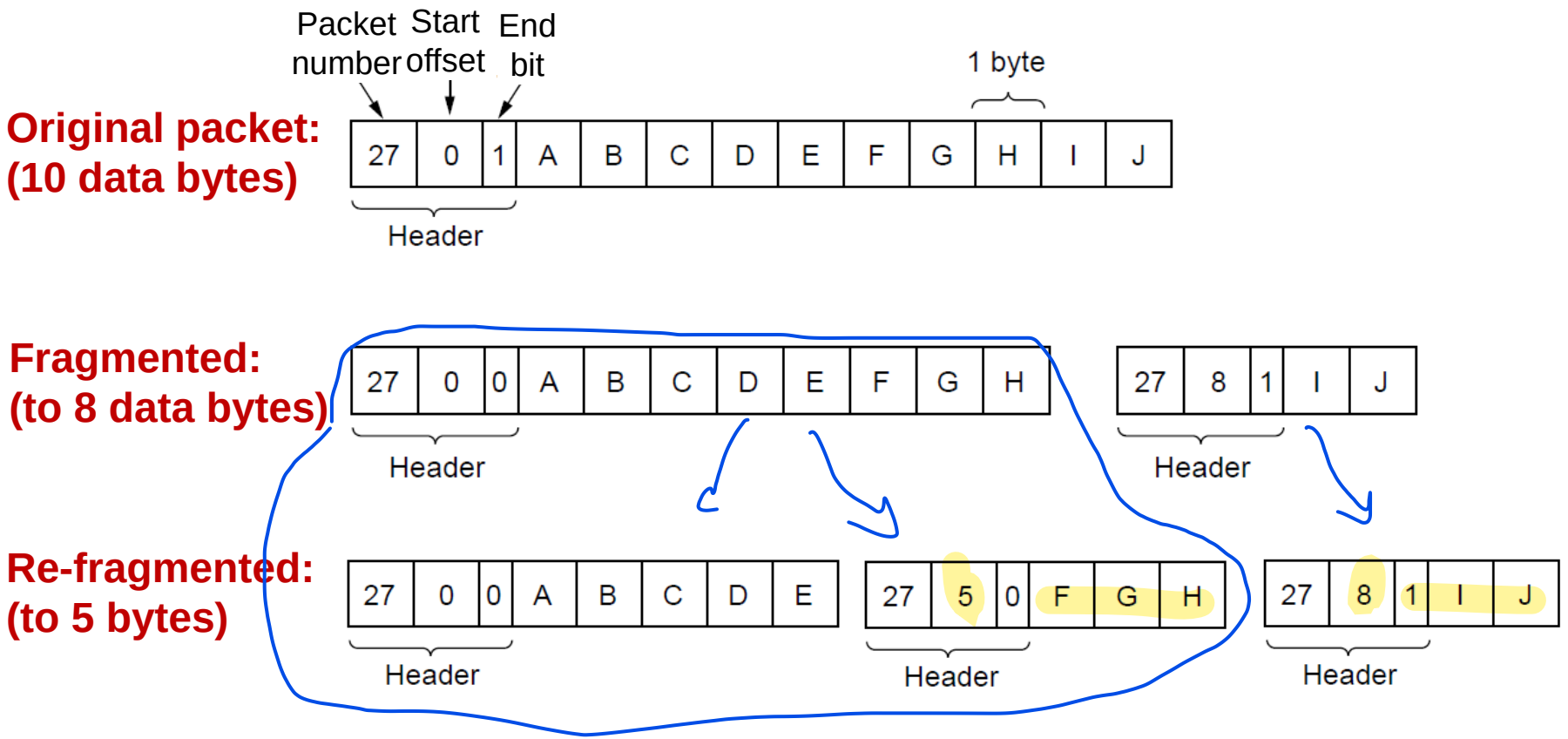
- Assign each packet a number *P*.
- Number fragment P.0, P.1, P.2,…
- If a fragment needs to be re-fragmented, add another level.
  - E.g. fragment 1 needs to be re-fragmented P.1.0, P.1.1,…,

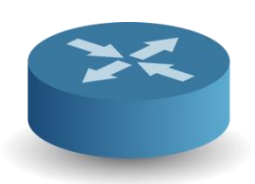
There is a serious problem with this scheme, what is it?

Packet Fragmentation

Example of IP-style fragmentation:

Flat numbering

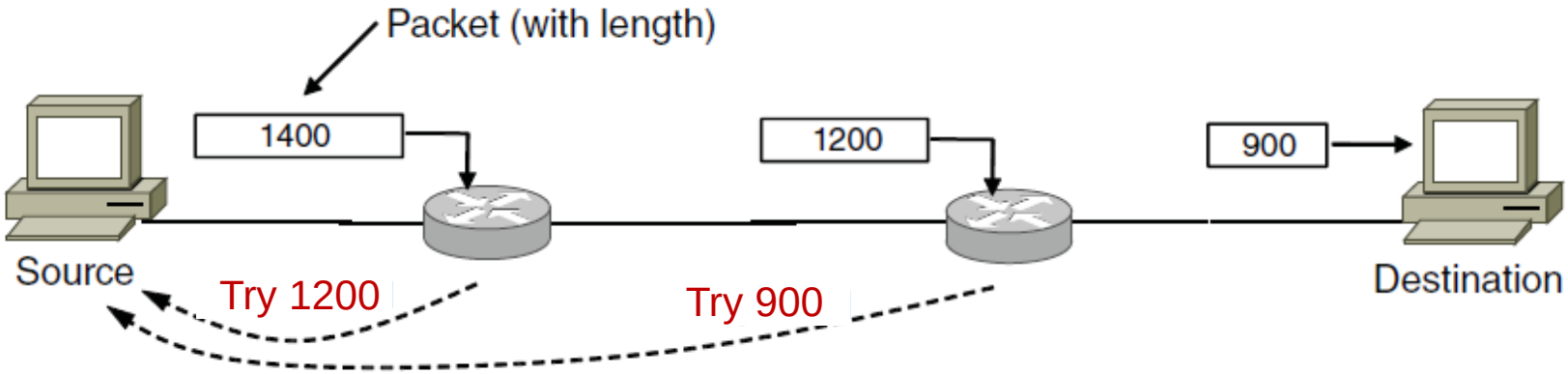




Packet Fragmentation (3)

Path MTU Discovery avoids network fragmentation

- Routers return MTU (Max. Transmission Unit) to source and discard large packets



General Notes of Fragmentation

- If a fragment is lost, the entire packet needs to be retransmitted
- Reassembly node need to have some sort of timeout to avoid waiting for a lost fragment forever
- Reassembly needs to be done in a single central place
  - Problem with distributed forwarding architecture
- Fragmentation can produce very small fragments (***can you provide an example?***)
- Fragmentation can be used for DOS attacks (***how?***)
- *So why is fragmentation good?*
  - It allows a inter-connection of heterogeneous networks