Internet Engineering

241-461

Robert Elz

kre@munnari.OZ.AU
kre@coe.psu.ac.th

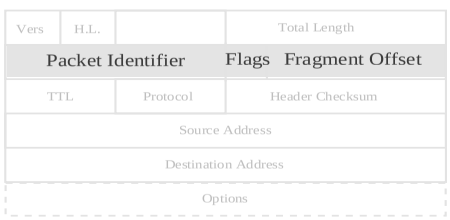
http://fivedots.coe.psu.ac.th/~kre

Dealing with Link Layer

- Link Layer
  - Has its own addressing
  - Has packet size limits
    - maximum packet size (usually)
    - minimum packet size (sometimes)
- IP must cope with this
  - Minimum packet size
    - Easy
      - if IP packet is too small
        append zeroes, make it big enough
      - receiver knows original length
        from IP header packet length field
      - repeated for each link crossed
        separately and independently

Maximum Packet Size

- If IP packet is too big
  - Broken into fragments
    - Each fragment is a new IP packet
    - Sent to destination
      - Reassembled only at destination



- Packet Identifier
  - Identifies original IP packet
    - together with addresses & protocol
    - allows all fragments to be collected
      - so they can be reassembled

Contents

Why Fragmentation

- Packets can be big
  - 65535 bytes
    - including header - IPv4
- Links usually have MTU
  - Maximum Transmission Unit
  - Limit on maximum packet size
    - Ethernet 1500 (+ ethernet headers)
- Packet might be bigger than MTU
  - What to do?

Why Fragmentation (2)

- Large truck - Low bridge
  - Use a different route
  - Can packets do that?
- Packet routing
  - path to destination address only
  - Nothing about packet sizes
  - Perhaps no route will handle big packets
- So NO cannot alter route
  - for different packet sizes
  - (Really could, but too much work already)
- What other choices?

Why Fragmentation (3)

- Large Truck - Shaky bridge
  - Truck too heavy
  - No other route to destination
- Take load from big truck
  - use several smaller trucks
  - each weigh much less
    - able to cross the bridge
- Same idea for packets
  - Take load from packet
  - Divide load
  - Place in several smaller packets (fragments)
    - Now packets get through link
    - Even with small MTU

Fragmentation!

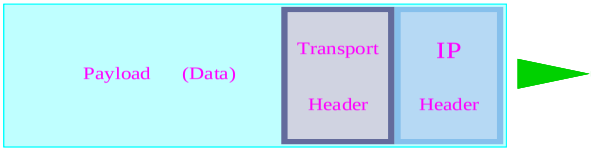
Contents

Why Reassembly

- Reassembly
  - Putting the fragments together again
  - Get original packet back
- Why?
- Don’t do this with truck & load
  - Once on small truck
  - Take load to destination
  - Deliver to recipient
- What is different about packets?

Why Reassembly (2)

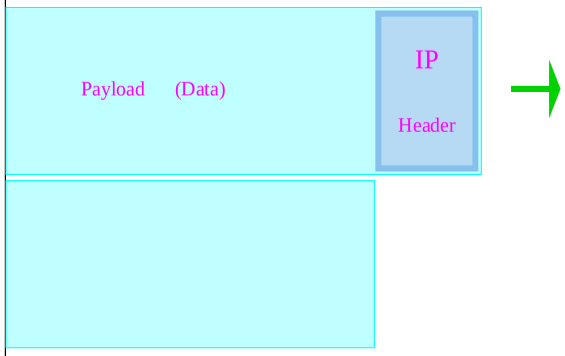
- Consider an IP packet



- Packet arrives at small MTU link
  - Must be fragmented

Why Reassembly (3)

- To fragment we take off the load



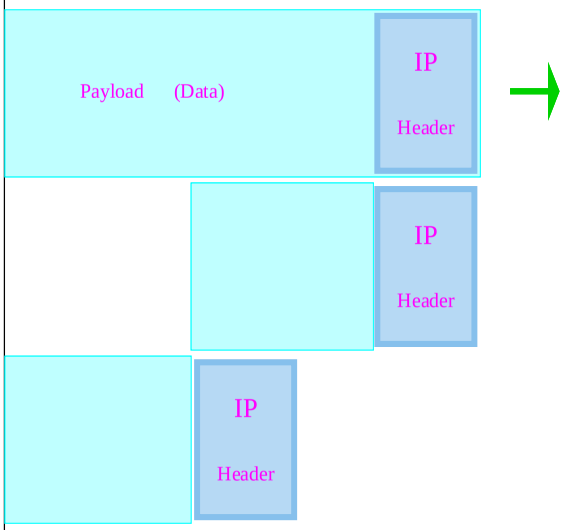
Why Reassembly (3)

- To fragment we take off the load

Divide load into smaller par

Why Reassembly (3)

- To fragment we take off the load



Why Reassembly (3)

- To fragment we take off the load

put each

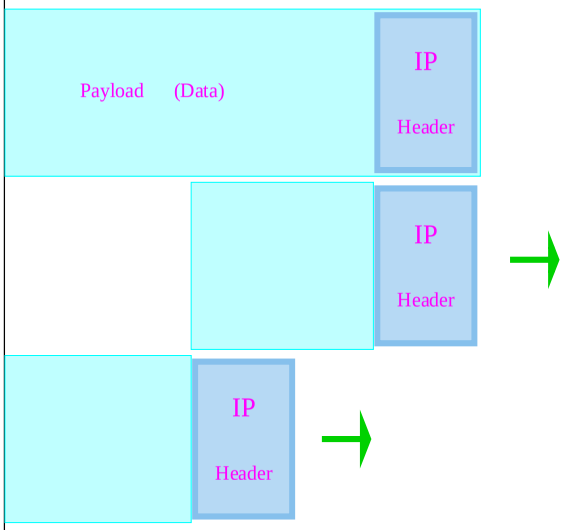
in an IP

(add IP

each

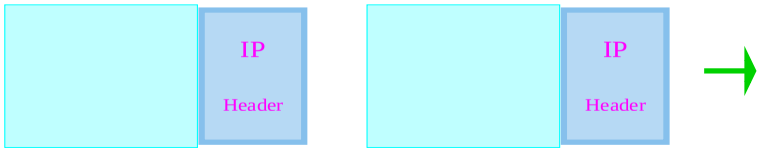
Why Reassembly (3)

- To fragment we take off the load



Why Reassembly (4)

- Fragments arrive at destination



  - Ready to be delivered
- Transport header controls delivery

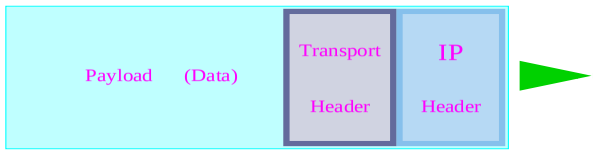
  - First fragment delivered OK

(not really)

  - Second fragment??
    - No transport header - how to deliver??

Why Reassembly (5)

- Must get all payload back
  - into one packet
  - including transport header
    - and all transport data



  - Now packet can be delivered

Reassembly

Contents

When to Fragment

- When do we fragment ?
- The easy question!
  - Only when required
    - Fragmenting adds overhead
    - Extra headers
  - When packet is too big
    - For the link it is to use
      - Forwarding table tells us this
      - From routing path calculations
    - Compare MTU of link with packet size
- Always fragment?
  - No - sender can prohibit
    - (more later)

Contents

When to Reassemble

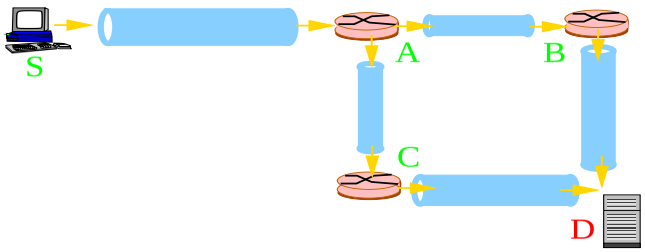
- When do we reassemble packet?
- The interesting question!
  - As soon as possible ?
  - After "bad" link is passed ?
- No.
  - Why??
- Might be more small MTU links
  - Would need to fragment again ??
- Yes.

When to Reassemble

- When do we reassemble packet?
- The interesting question!
  - As soon as possible ?
  - After "bad" link is passed ?
- No.
  - Why??
- Might be more small MTU links
  - Would need to fragment again ??
- Yes. But...
  - Not a very good reason
  - Not the important reason

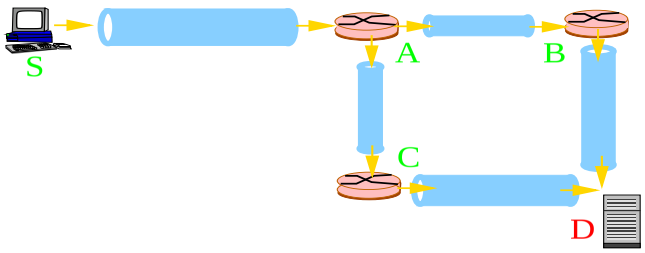
When to Reassemble (2)

- Imagine this network:



  - Thickness of PIPES
    - indicates MTU of link
- Paths from S to D
  - S - A - B - D
  - S - A - C - D
    - Equal cost
      - Can use either
      - Can use both

When to Reassemble (2)



- Large packet sent by S
  - Size less than MTU(S-A)
    - (MTU of link from S to A)
  - Size greater than MTU(A-B) and MTU(A-C)
  - Packet must be fragmented at A
- Assume one fragment sent to B
  - Other fragment sent to C
  - What happens if B wants to reassemble? C?

When to Reassemble (3)

- Only safe place to reassemble
  - At destination host
  - Packet was addressed to it
    - All fragments addressed to it
    - All arrive at Destination eventually
      - Or are lost forever
- Destination host
  - can reassemble
    - all fragments arrive
  - must reassemble
    - Packet cannot be delivered
    - until entire packet is back together
- No router on path can reassemble
  - no guarantee all fragments arrive there
  - must wait for last fragment to reassemble
    - if sent elsewhere (not to this router)
    - wait for very long time