Fragmentation

Geoff Huston
APNIC Labs

Before Packets…

Digitised telephone networks switched **time**

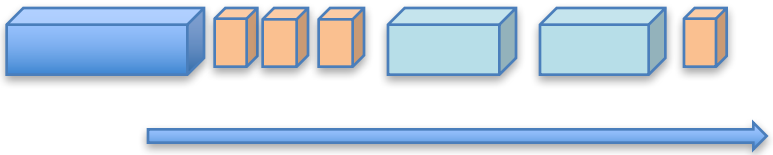
- Each active network transaction was a 56K constant bit rate data stream
- Each stream was divided into 8,000 7 bit samples per second
- Each 7 bit sample was aggregated with other samples and packed into frames
- Each frame was switched at 8K frames per second

Packets are Different

- Computers do not require constant bit rate
- They can optimise their data rates to make efficient use of the network
- They can vary the packet size to match the requirements of the application and the network
- They do not rely on a network state – each packet contains information in the header to allow it to be passed to the destination

Packet Networks are Different

- The range of packet sizes supported in a network represents a set of engineering trade-offs
  - Bit error rate of the underlying media
  - Desired carriage efficiency
  - Transmission speed vs packet switching speed



Media Packet Sizes

- Ethernet 64 – 1,500 octets
  - These numbers were derived from the original CSMA-CD design
- FDDI 4,532 octets
- Frame Relay 46 – 4,470 octets
- ATM 53 octets

BER, Framing, FEC (or not), Jitter, HOL blocking all play a role in the design tradeoffs for media packet sizes

The IEEE Jumbogram Fiasco

- 1500 octets was fine for 10Mbps
  - 800 packets per second
- But at 100Gbps?
  - 8,000,000 pps

- So why not allow for larger packets?
- Yes, but what size?
  - IEEE found themselves incapable of standardizing which size to pick

Protocol Design

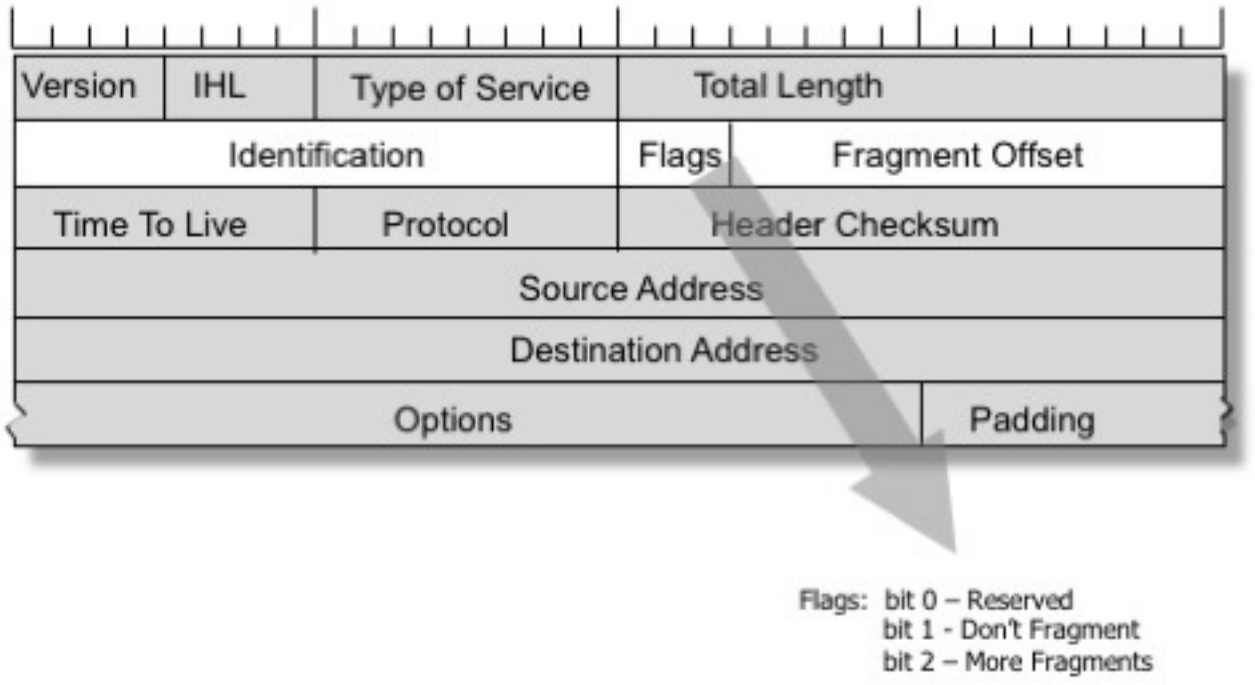
- Adopt a fixed packet size approach
  - Tends to be a lower number (ATM)
  - Decreases carriage efficiency and increases packet switching loads
- Adopt a variable size approach
  - Maximises applicability
  - Maximises carriage efficiency
  - But the protocol needs to cope with packet size mismatch as a packet traverses multiple networks

IPv4 Packet Design

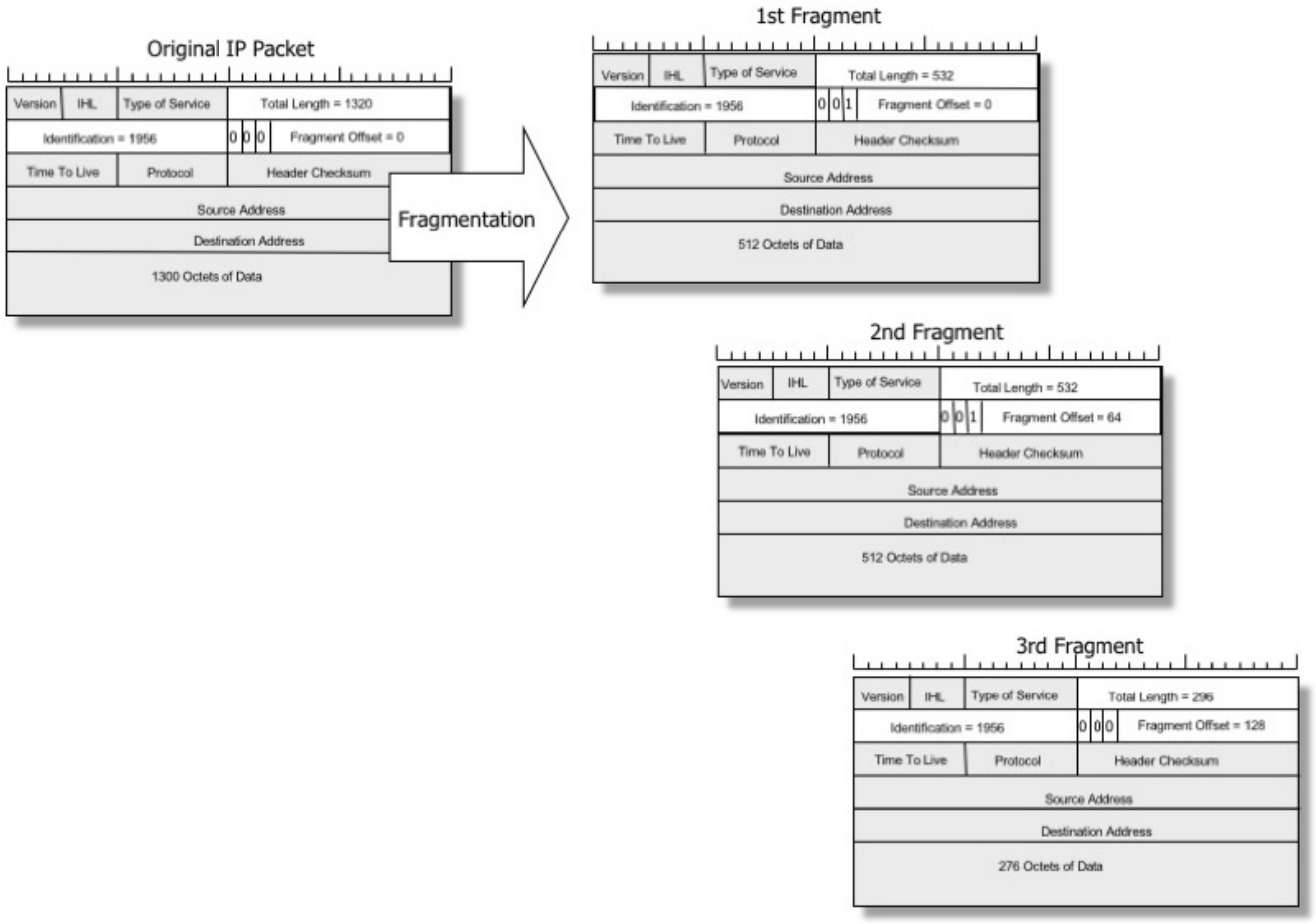
FORWARD fragmentation

- If a router cannot forward a packet on its next hop due to a packet size mismatch then it is permitted to fragment the packet, preserving the original IP header in each of the fragments

IPv4 Fragmentation Control



IPv4 Fragmentation



IPv4 and the Don't Fragment bit

If Fragmentation is not permitted by the source, then the router discards the packet. The router may send an ICMP to the packet source with an Unreacahble code (Type 3, Code 4)

Later implementations added a MTU size to this ICMP message

ICMP messages are extensively filtered in the Internet so applications should not count on receiving these messages

Trouble at the Packet Mill

- Lost frags require a resend of the entire packet – this is far less efficient than repairing a lost packet
- Fragments represent a security vulnerability as they are easily spoofed
- Fragments represent a problem to firewalls – without the transport headers it is unclear whether frags should be admitted or denied
- Packet reassembly consumes resources at the destination

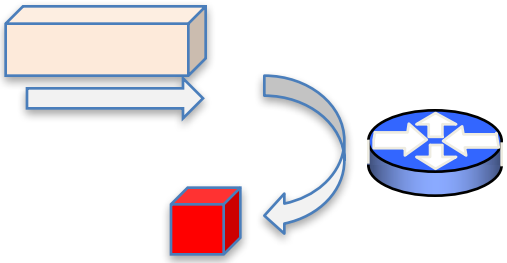
The thinking at the time…

Fragmentation was a Bad Idea!

Kent, C. and J. Mogul, "Fragmentation Considered Harmful", Proc. SIGCOMM '87 Workshop on Frontiers in Computer Communications Technology, August 1987

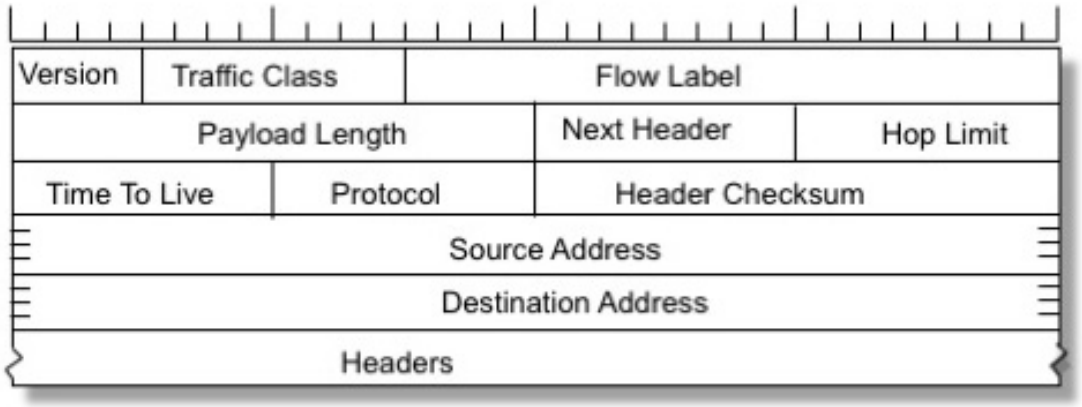
IPv6 Packet Design

- Attempt to repair the problem by effectively jamming the DON'T FRAGMENT bit to ON
- IPv6 uses BACKWARD signalling
  - When a packet is too big for the next hop a router should send an ICMP6 TYPE 2 (Packet Too Big) message to the source address and include the MTU of the next hop.

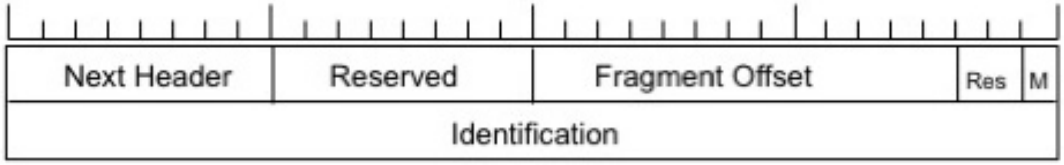


IPV6 Source Fragmentation

IPv6 Packet Header



IPv6 Fragmentation Header



What changed? Whats the same?

- Both protocols may fragment a packet at the source
- Both protocols support a Packet Too Big signal from the interior of the network to the source
- Only IPv4 routers may generate fragments on-the-fly

What does "Packet Too Big" mean anyway?

errrrr

What does "Packet Too Big" mean anyway?

- For TCP it means that the active session referred to in the ICMP payload* should drop its session MSS to match the MTU **

* IPv4: assuming that the payload contains the original IP header

** assuming that the ICMP is genuine

What does "Packet Too Big" mean anyway?

- For UDP its not clear:
  - The offending packet has gone away!
  - Some IP implementations appear to ignore it
  - Others add a host entry to the local IP Forwarding table that records the MTU

Problems

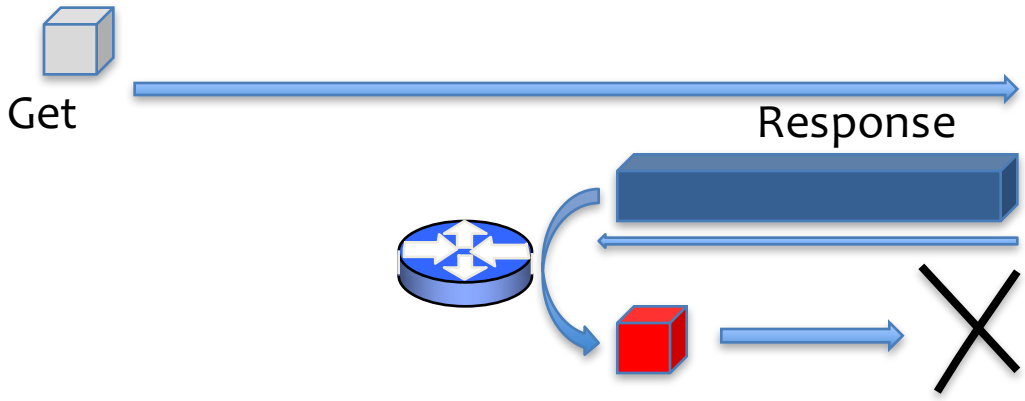
ICMP is readily spoofed

- An attacker may send a fragment stream with a maximum fragment offset value causing a potential memory starvation issue on the destination
- A attacker may send partially overlapping fragments
- An attacker may spoof ICMP PTB messages with very low MTU values
- An attacker may spoof a stream of ICMP PTB messages with random IPv6 source addresses

Problems

ICMP is widely filtered

- leading to black holes in TCP sessions
  - GET is a small HTTP packet
  - The response can be arbitrarily large, and if there is a path MTU mismatch the response can wedge



Problems

ICMP is widely filtered

- Ambiguity in UDP
  - Is packet loss due to congestion or MTU mismatch?
  - Should I give up, resend or revert to TCP?

Problems

Backward signalling is unreliable

- In no other part of the IP protocol is it assumed that the source address of an IP packet is reliably reachable by anything other than the addressed destination
- Source addresses are not necessarily “real”
  - MPLS
  - IP tunnels
  - SDN

Who should control packet fragmentation?

Is fragmentation and reassembly an IP-Layer function or a Application Session function?

- i.e. is the DNS use of EDNS0 buffer size options with explicit max MTU signalling under application control a better approach than allowing the IP layer to manage this?

e.g. DNS and EDNS0

- The DNS is the most critical user of UDP
- Developments in security protocol evolution has lead to larger DNS responses
- DNS took on explicit fragmentation control with EDNS0
  - Loss of fragment coherence leads to re-query with lower buffer size, which leads to fallback to TCP
  - This is controlled by the DNS application, not the network layer

Where now?

- Deprecate Fragmentation
  - Bonica had a draft on this for IPv6

Where now?

- Walk away from variable size and adopt a single max MTU for IP?
  - But what is the “single” MTU size?

    1500?

    1280?

    1272?

    ???

Where now?

- Keep fragmentation, and try and make ICMP PTB work consistently and reliably across all of the Internet for IPv6 (and IPv4)?

or

- Just do nothing and just hope it all goes away!

Thanks!