IPv6 and Fragmentation

Geoff Huston AM
Chief Scientist, APNIC

I have just a few minutes

So I will skip forward to slide 31

What follows here in the pack is a quick explanation of IP fragmentation and why it's useful and why it doesn't work!

The full pack is available here

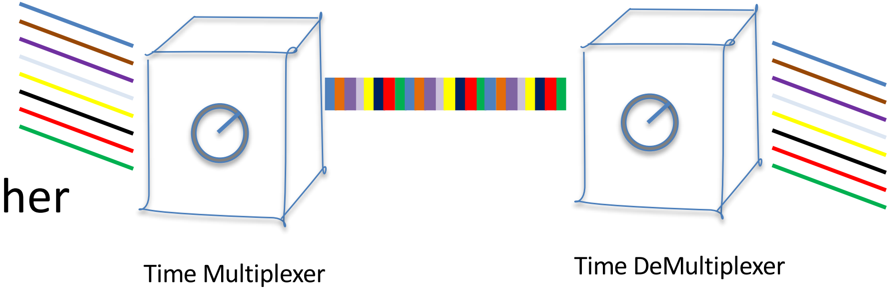
Fragmentation

- IP is one of the few protocols that allowed packets to be fragmented by the network
- This has been both a fundamental strength and a major weakness for IP
- Lets look at fragmentation in a bit more detail

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



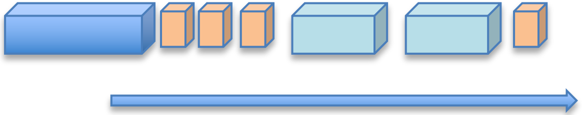
Packets are Different

- Computers do not require constant bit rate virtual circuits
- 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, etc all play a role in the design tradeoffs for media packet sizes

Aside: The IEEE Jumbogram Fiasco

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

- So why not allow for larger packets?
- Yes, but what size?
  - IEEE found themselves incapable of standardizing which size to pick
  - So they ended up picking none!

Packet Protocol Design

EITHER use a fixed packet size approach

- Tends to be a lower number (see ATM)
- Decreases carriage efficiency and increases packet switching loads

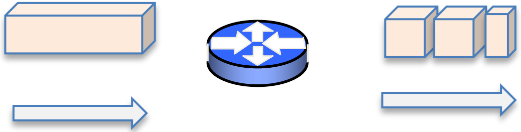
OR use a variable size approach

- Maximises applicability
- Maximises carriage efficiency
- But the protocol needs to cope with packet size mismatch as a packet traverses multiple networks

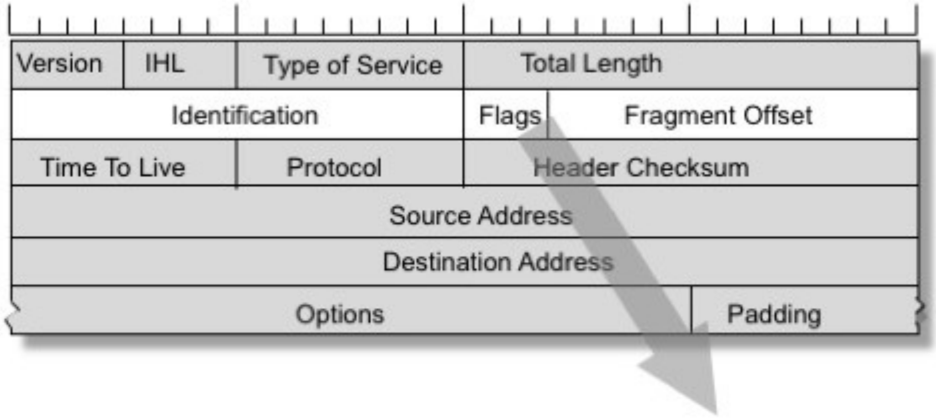
IPv4 Packet Design

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



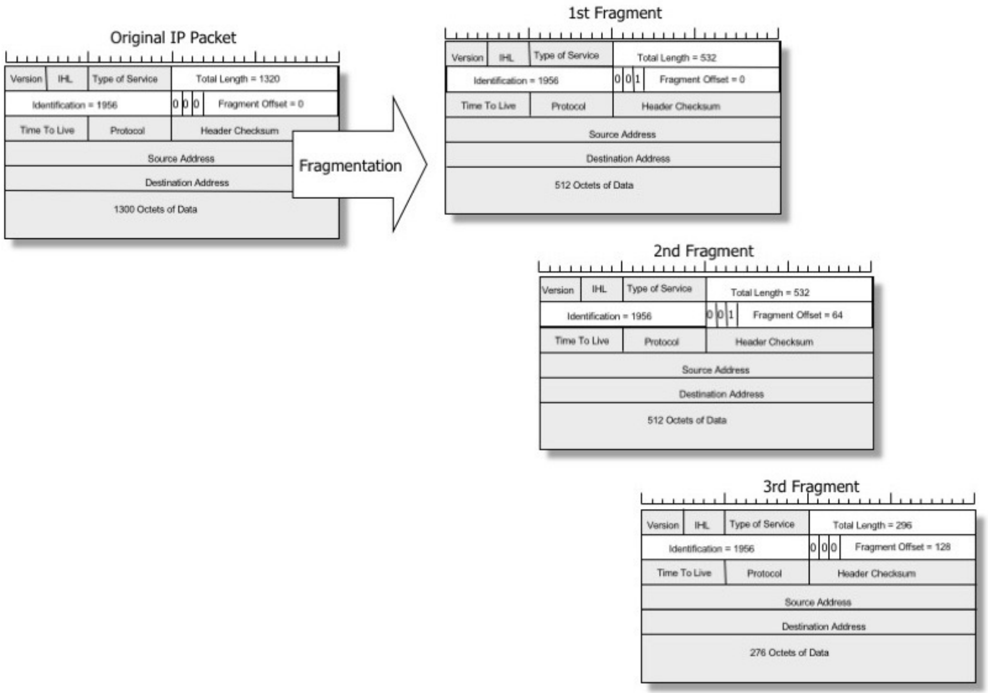
IPv4 Fragmentation Control



Version
IHL
Type of Service
Total Length
Identification
Flags
Fragment Offset
Time To Live
Protocol
Header Checksum
Source Address
Destination Address
Options
Padding

Flags: bit 0 – Reserved
bit 1 - Don't Fragment
bit 2 – More Fragments

IPv4 Fragmentation



Original IP Packet

Version	IHL	Type of Service	Total Length = 1320
Identification = 1956		0 0 0	Fragment Offset = 0
Time To Live	Protocol		Header Checksum

Source Address

Destination Address

1300 Octets of Data

1st Fragment

Version	IHL	Type of Service	Total Length = 532
Identification = 1956		0 0 1	Fragment Offset = 0
Time To Live	Protocol		Header Checksum

Source Address

Destination Address

512 Octets of Data

2nd Fragment

Version	IHL	Type of Service	Total Length = 532
Identification = 1956		0 0 1	Fragment Offset = 64
Time To Live	Protocol		Header Checksum

Source Address

Destination Address

512 Octets of Data

3rd Fragment

Version	IHL	Type of Service	Total Length = 296
Identification = 1956		0 0 0	Fragment Offset = 128
Time To Live	Protocol		Header Checksum

Source Address

Destination Address

276 Octets of Data

IPv4 and "Don't Fragment"

If Fragmentation is not permitted by the source, (by setting the Don’t Fragment bit) 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 IPv4 implementations added a MTU size to this diagnostic ICMP message to indicate how to repair the problem

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

Trouble at the Packet Mill

- Lost frags require a resend of the entire packet
- The 16-bit identification field represents a ceiling to the number of packets in flight for high-speed high-latency systems
- Fragments represent a problem to firewalls
  - without the transport headers (which are only in the leading fragment) it is unclear whether subsequent frags should be admitted or denied
- Fragments represent a massive problem to ECMP per-flow load balancers
- Packet reassembly consumes resources at the destination

The thinking at the time…

Fragmentation was, all things considered, a net Bad Idea!

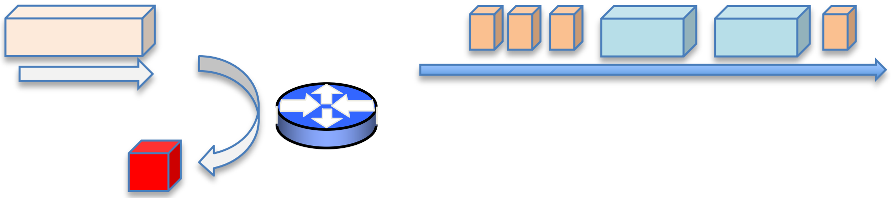
Kent, C. and J. Mogul, "Fragmentation Considered Harmful", Proc. SIGCOMM '87 Workshop on Frontiers in Computer Communications Technology, August 1987

IPv6 Packet Design

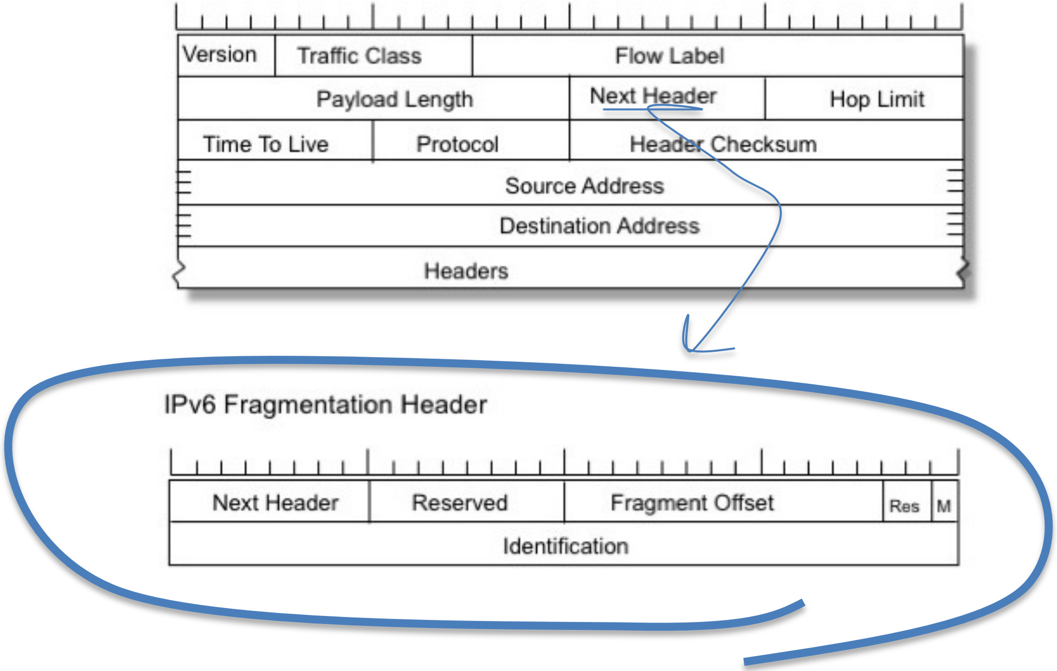
- Attempt to repair the problem by effectively jamming the DON'T FRAGMENT bit to ON
  - Which effectively prohibits on-the-fly fragmentation by intermediate switches

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

Version	Traffic Class	Flow Label	
Payload Length		Next Header	Hop Limit
Time To Live	Protocol	Header Checksum	
Source Address			
Destination Address			
Headers			

IPv6 Fragmentation Header

Next Header	Reserved	Fragment Offset	Res	M
Identification				

What changed? What's the same?

- All IPv4 packets have Fragmentation Control fields.
- Only Fragmented IPv6 packets have IPv6 Extension headers added to the packet
- IPv4 sources and routers may generate fragments
- Only IPv6 sources may fragment a packet
- Both protocols support a “Packet Too Big” ICMP diagnostic signal from the interior of the network to the source

What does "Packet Too Big" mean anyway?

errrrr

What does "Packet Too Big" mean anyway?

- Clearly the packet was too big to be delivered, and this is a notice to the sources to that effect
- All well and good, but what is the source meant to do then?

It's a Layering Problem

- Fragmentation was seen as an IP level problem
  - It was meant to be agnostic with respect to the upper level (transport) protocol
- But we don’t treat it like that
  - And we expect different transport protocols to react to fragmentation notification in different ways

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, and re-send unacknowledged data **, ***

i.e. you should **never** see IPv6 fragments in TCP!

* IPv4: assuming that the payload contains the original IP + TCP headers

** assuming that the ICMP is genuine

*** and if that's too hard, set a per destination MTU value from the ICMP and hope that the TCP session is able to get itself out of its wedged state and resend the data within the new MTU

What does "Packet Too Big" mean anyway?

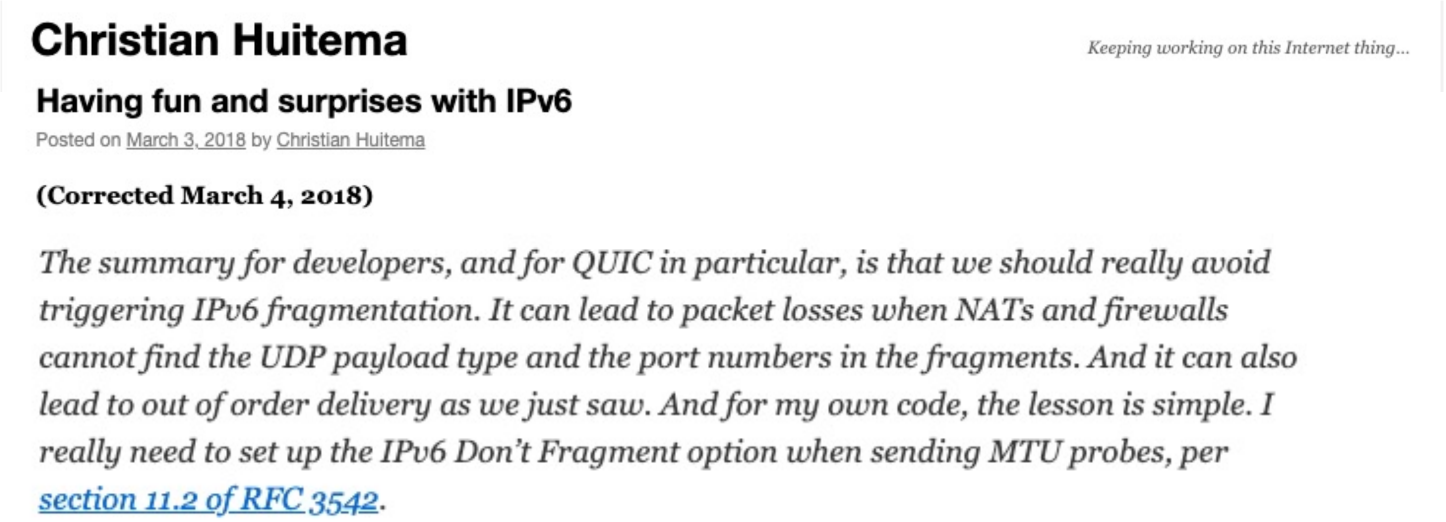
For **<u>UDP</u>** its not clear:

- – The offending packet has gone away!
- – Some IP implementations appear to ignore it *
- – The host should add an entry to the local IP forwarding table that records the MTU that should be used to send future packets to this destination

* This is bad!!!

What does "Packet Too Big" mean anyway?

For **QUIC**:



Christian Huitema

Keeping working on this Internet thing…

Having fun and surprises with IPv6

Posted on March 3, 2018 by Christian Huitema

(Corrected March 4, 2018)

The summary for developers, and for QUIC in particular, is that we should really avoid triggering IPv6 fragmentation. It can lead to packet losses when NATs and firewalls cannot find the UDP payload type and the port numbers in the fragments. And it can also lead to out of order delivery as we just saw. And for my own code, the lesson is simple. I really need to set up the IPv6 Don't Fragment option when sending MTU probes, per section 11.2 of RFC 3542.

https://huitema.wordpress.com/2018/03/03/having-fun-and-surprises-with-ipv6/

Problems

ICMP is readily spoofed:

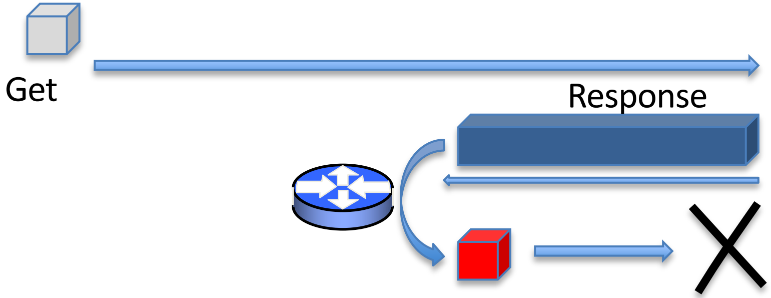
- An attacker may send a fragment stream with a maximum fragment offset value causing a potential memory starvation issue on the destination
- An attacker may send partially overlapping fragments
- An attacker may spoof ICMP PTB messages with very low MTU values
- An attacker may spoof a stream of ICMP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

- Leading to ambiguity in UDP
  - Is UDP 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

IPv6 Fragmentation: Adding an Extension Header

Extension Headers are a problem

- A number of implementations of network level packet processing equipment appears to be intolerant of IPv6 packets with Extension headers – so they drop them!
- IPv6 Fragmentation Control is an Extension Header
- Today’s network has a significant level of drop of IPv6 packets with fragmentation extension headers

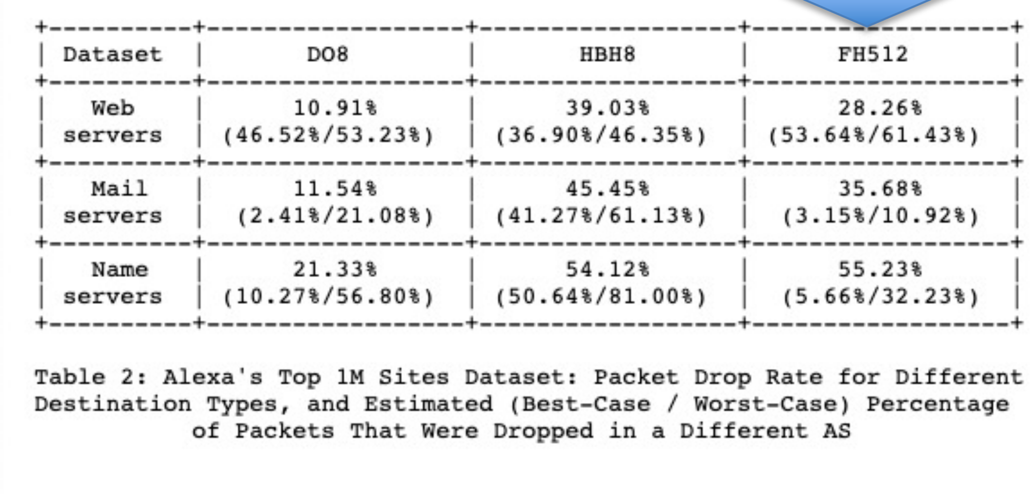
Now to Measurements…

How serious is this problem?

- How bad is fragmentation loss in IPv6?
- How bad is Extension Header loss in IPv6?

Initial Tests: 2014 (RFC 7872)

- August 2014 and June 2015
- Sent fragmented IPv6 packets towards “well known” IPv6 servers (Alexa 1M and World IPv6 Launch
- Drop Rate:



Dataset	DO8	HBH8	FH512
Web servers	10.91% (46.52%/53.23%)	39.03% (36.90%/46.35%)	28.26% (53.64%/61.43%)
Mail servers	11.54% (2.41%/21.08%)	45.45% (41.27%/61.13%)	35.68% (3.15%/10.92%)
Name servers	21.33% (10.27%/56.80%)	54.12% (50.64%/81.00%)	55.23% (5.66%/32.23%)

Table 2: Alexa's Top 1M Sites Dataset: Packet Drop Rate for Different Destination Types, and Estimated (Best-Case / Worst-Case) Percentage of Packets That Were Dropped in a Different AS

APNIC Test - August 2017

- Use APNIC IPv6 measurement platform to test the drop rate of IPv6 packets flowing in the opposite direction (server to client)

	Count	%
Tests	1,675,898	
ACK Fragmented Packets	1,324,834	79%
Fragmentation Loss	351,064	21%

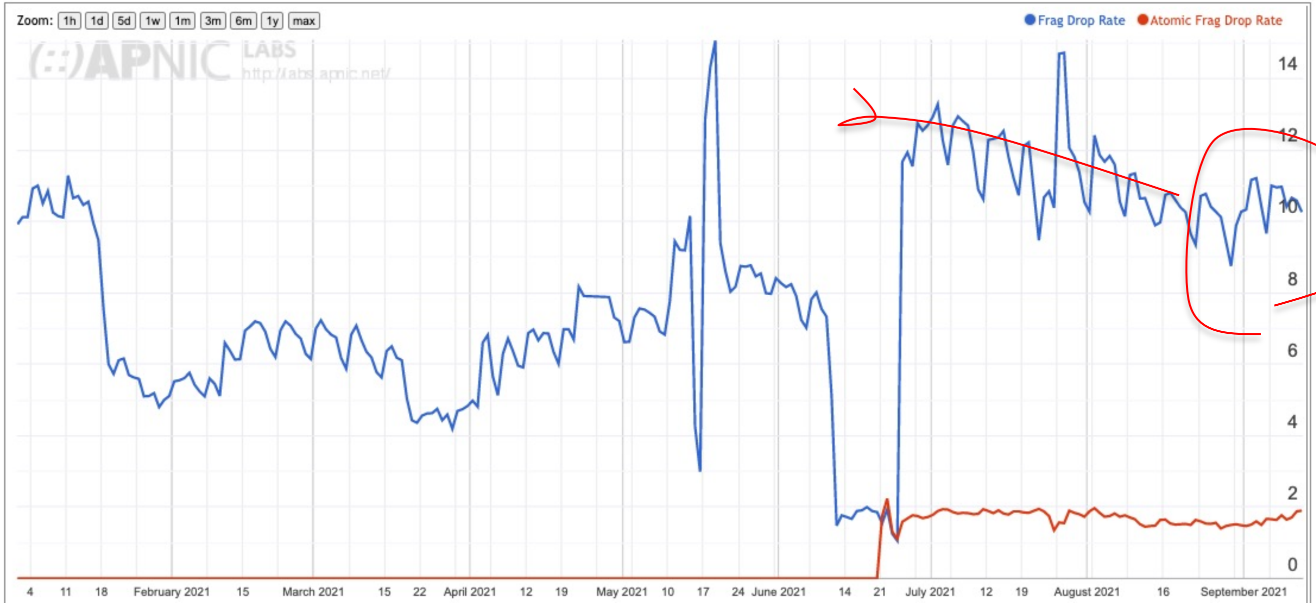
That's 21%
This is an improvement over the RFC 7872 measurement, but its still a really bad number!

APNIC Test - 2021

Re-work of the 2017 measurement experiment

- Same server-to-client TCP session fragmentation mechanism
- Uses a middlebox to fragment outgoing packets - drop is detected by a hung TCP session that fails to ACK the sequence number in the fragmented packet
- This time we randomly vary the initial fragmented packet size between 1,200 and 1,416 bytes
- Performed as an ongoing measurement

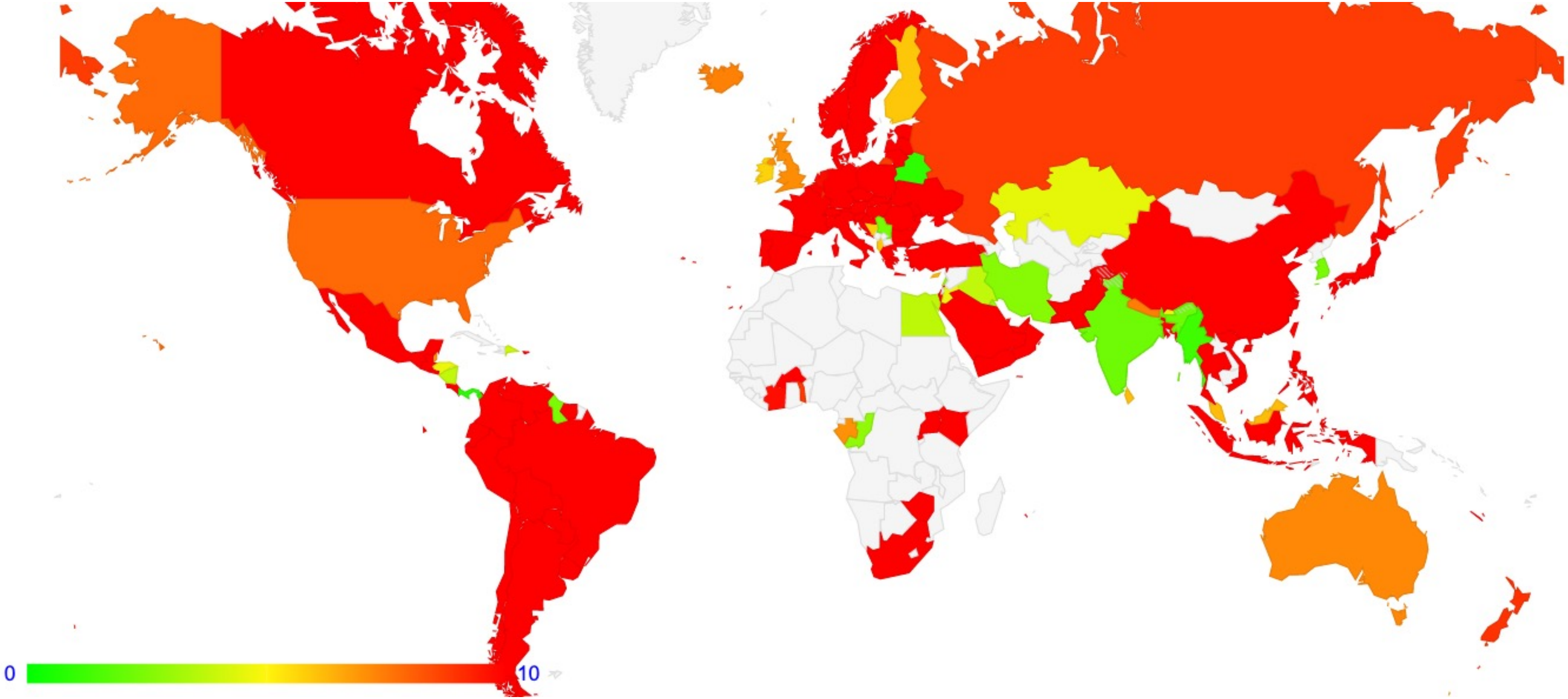
2021 Fragmentation Drop Rate



This is a significant improvement over 2017 data

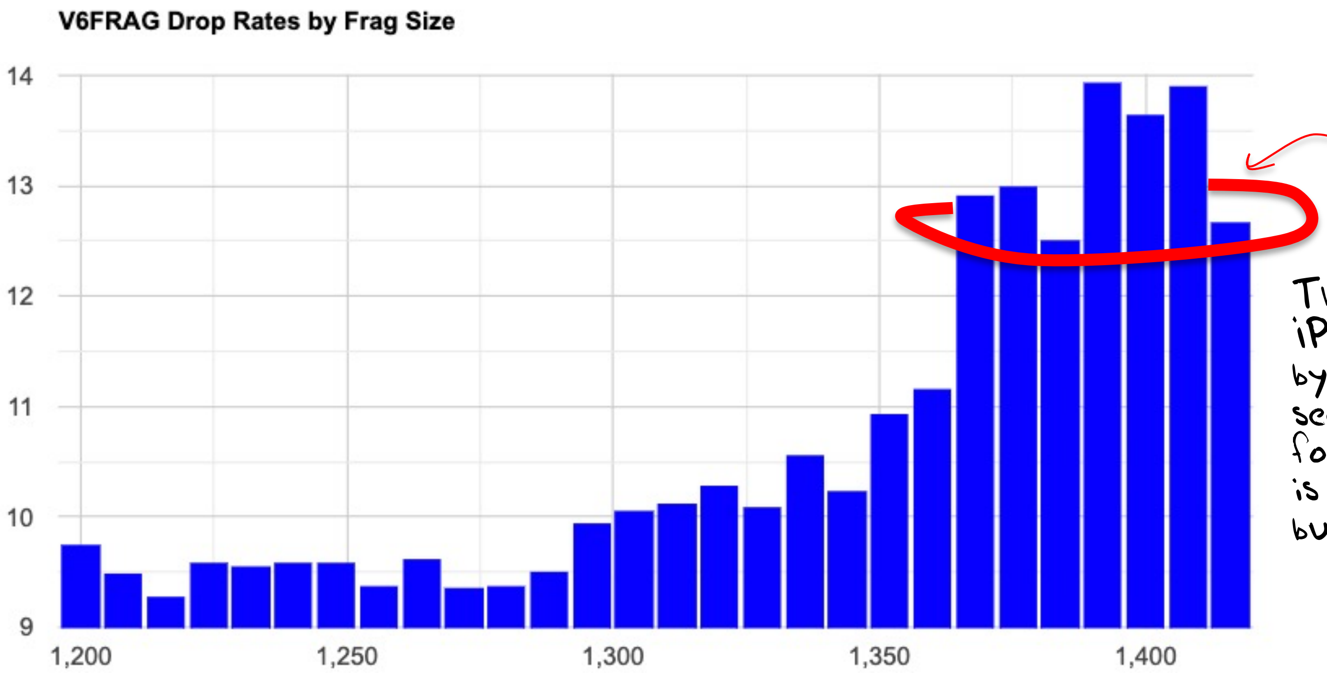
Since 2017 there are 10x the number of iPv6 users and the fragmentation drop rate has halved – we appear to be getting better at handling iPv6 fragments!

2021 Fragmentation Drop Rate



More recent iPv6 deployments appear to be a lot better than more mature ones

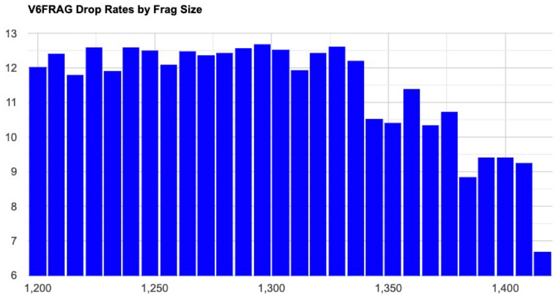
Drop Rate by Size



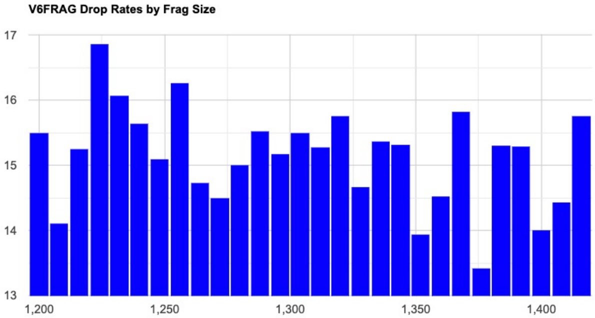
This is unexpected. At a total iPv6 packet size of 1408 bytes we did not expect to see higher packet drop rates for this packet size, as there is still an iP encapsulation budget of 92 bytes

Drop Size Profile by Region

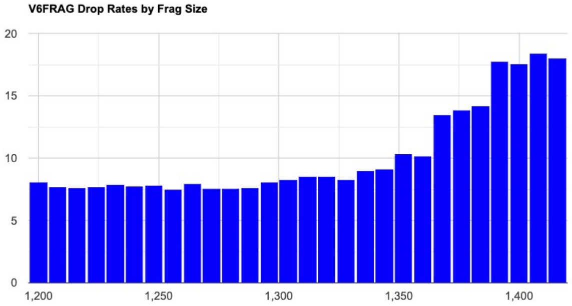
Americas



Europe



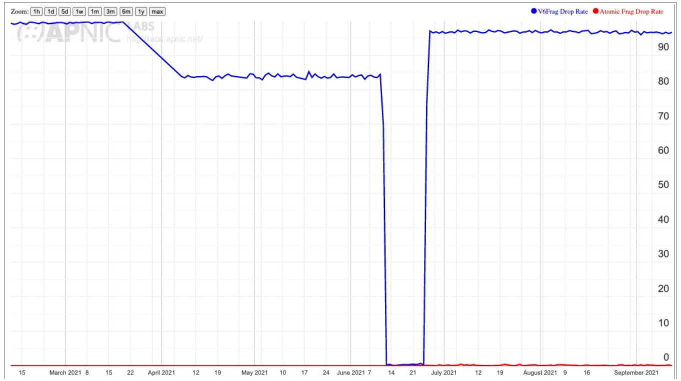
Asia



Why?

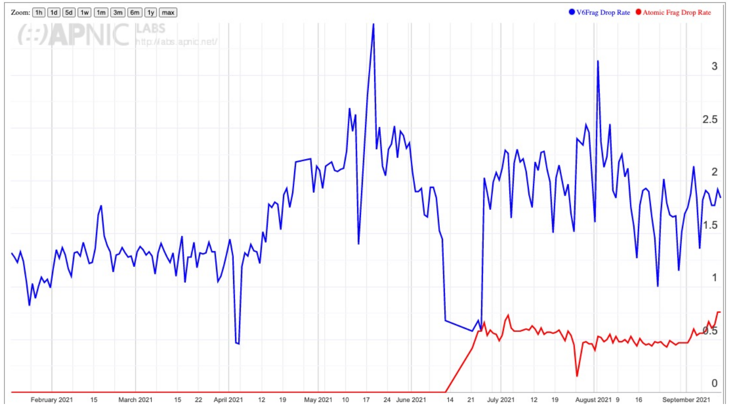
- Drop patterns vary across service providers, so there are probably contributary factors from network equipment and configurations

V6Frag Drop Measurement for AS852: TELUS Communications, Canada (CA)



80% Drop

V6Frag Drop Measurement for AS55836: RELIANCEJIO-IN Reliance Jio Infocomm Limited, India (IN)



2% Drop

Why?

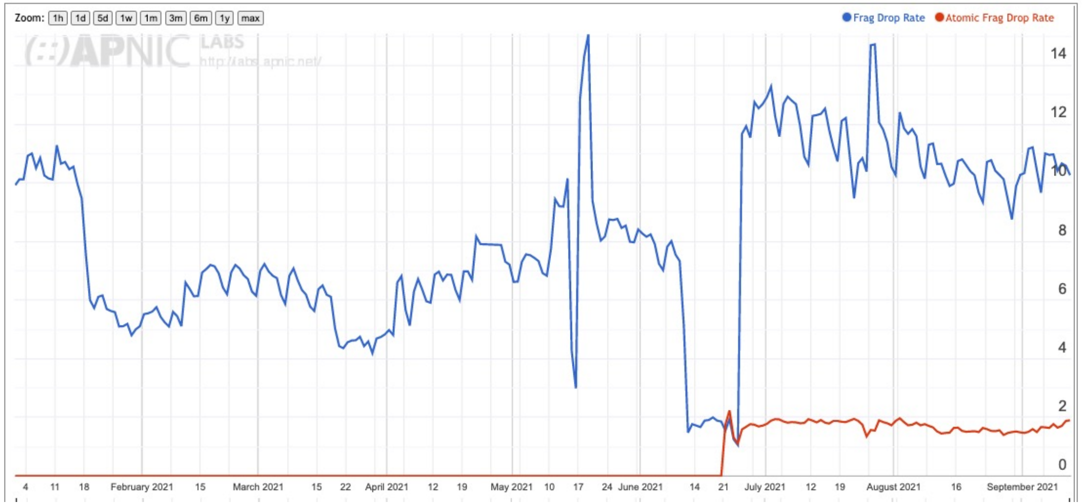
Other potential factors that could contribute:

- Local security policies
- IPv6 EH may trigger “slow path” processing in network equipment that could lead to higher drop rates
- IPv6 Path MTU woes!

"Atomic" Fragments

- It’s possible to add a “null” Fragmentation Extension header to a IPv6 packets

Use of V6FRAG Drop Rate for World (XA)



Atomic Frag Drop rate is 2%

The Atomic Frag Drop rate varies by region and by provider

- Europe – 8%
- Americas - 0.5%
- Asia – 1%

- AS3320 (DTAG, Germany) – 22%
- AS7922 (Comcast, US) – 0.3%
- AS55836 (Reliance Jio, India) - 0.3%
- AS54113 (Fastly) - 95%

Why?

Some possible explanations…

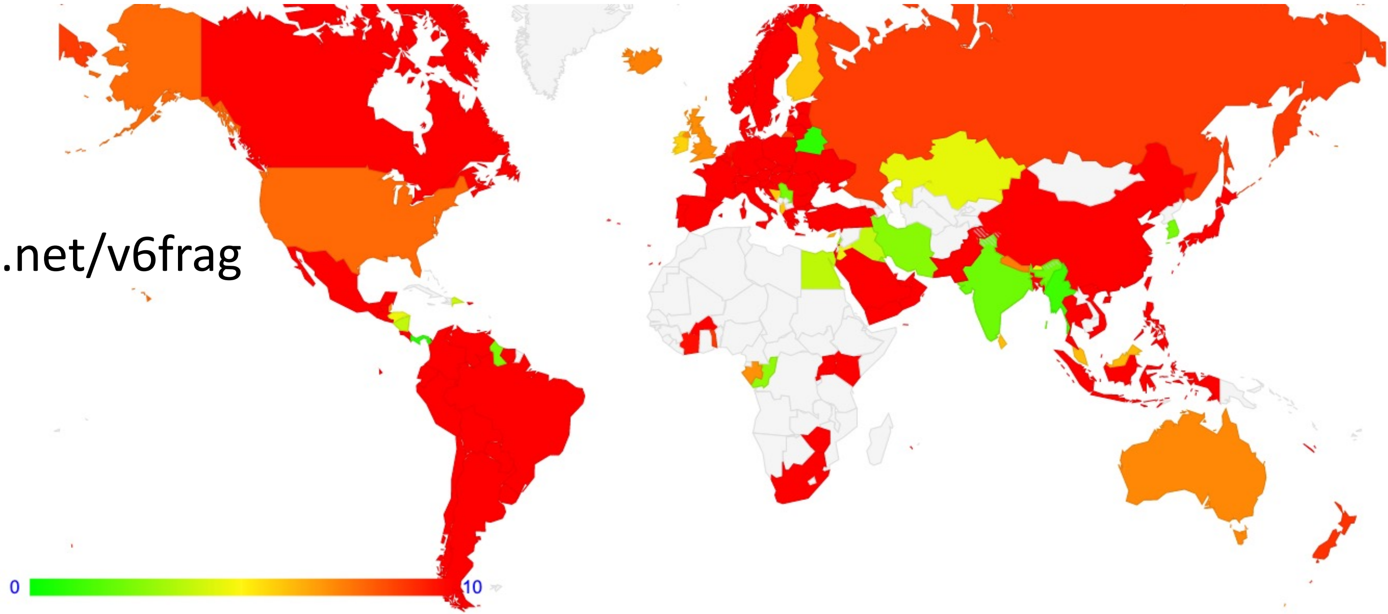
- Different dual stack transition architectures appear to have different behaviours with fragmentation and extension header handling
- Different use of LAG / ECMP approaches are variably tolerant of trailing frags with no transport header
  - The IPv6 Flow Label was meant to address this, but…
- Differing security stances with respect to fragment forwarding
- Different vendor equipment handles IPv6 packets differently
  - And ISPs don’t appear to care about a uniform handling setup across ISPs!
  - Because Dual Stack and fallback to IPv4 fixes everything – right?

Summary

- The IPv6 network is improving it’s handling of fragmented packets
- In 5 years its gone from ***unusably bad*** to ***tolerably poor in average, but terrible in some places***
- Recent IPv6 deployments appear to show more robust handling of IPv6 packets
  - Older IPv6 infrastructure appears to be less tolerant of both fragments and extension headers
- Smaller frags appear to be more robust than larger ones (if you are going to fragment a packet, prefer smaller fragment sizes, not larger ones)

Daily Report

https://stats.labs.apnic.net/v6frag



That's it!