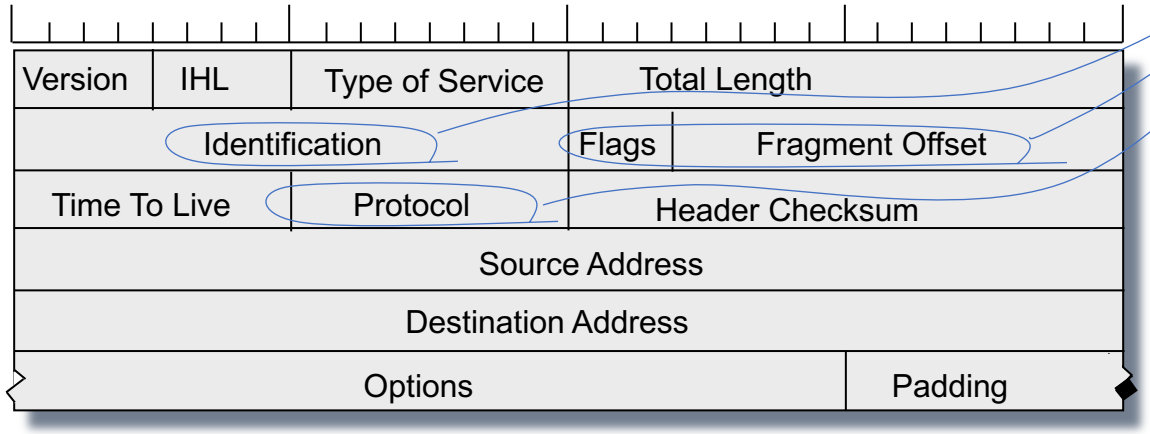
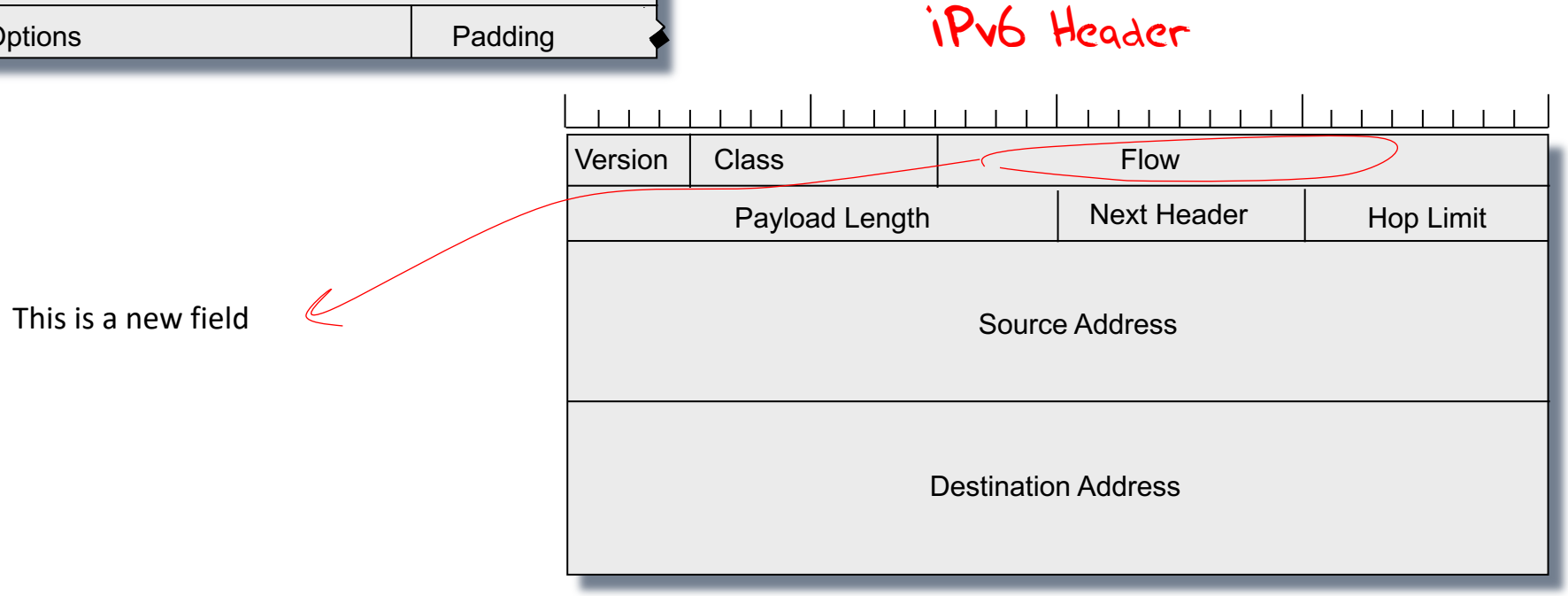
IPv6 Fragmentation and EH behaviours

Geoff Huston, Joao Damas
APNIC Labs

IPv6: Packet Header Changes



These three fields in the IPv4 header were pushed into the Extension Header chain, and do not appear in every IPv6 packet



This is a new field

IPv6: Packet Header Changes

- Type of Service changed to Traffic Class
  - (yet to find a useful agreed role, even ECN!)
- A Flow Label added
  - (but yet to find a useful role!)
- Header Checksum becomes a media layer function
- Options, Protocol fields replaced by chained Extension Headers
- **Packet ID and Fragmentation Controls become an Extension Header**

IPv6: Packet Header Changes

- Type of Service changed to Traffic Class
  - (yet to find a useful agreed role, even ECN!)
- A Flow Label added
  - (but yet ...)
- Fragmentation becomes a media layer function
  - Options, Protocol fields replaced by chained Extension Headers
- **Packet ID and Fragmentation Controls become an Extension Header**

The substantive change with IPv6 is the handling of Fragmentation and the use of Extension Headers

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

This is bad!
Really bad!

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

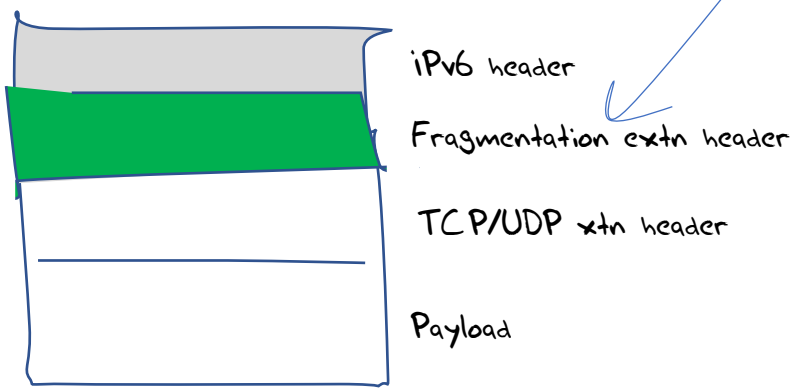
This is also bad!

Why is Frag Drop so high?

Some possible reasons for the high drop rate:

- Packet Fragments are often dropped by firewalls
- IPv6 Path MTU measures rely on ICMPv6 (as there is no ability for the router to fragment on the fly), and ICMP messages are commonly blocked
- Extension Header chains may either not be supported in router, or may only supported in the processor path (slow path)

IPv6 Fragmentation Extension Header Handling



The extension header sits between the IPv6 packet header and the upper level protocol header for the leading fragged packet, and sits between the header and the trailing payload frags for the trailing packets

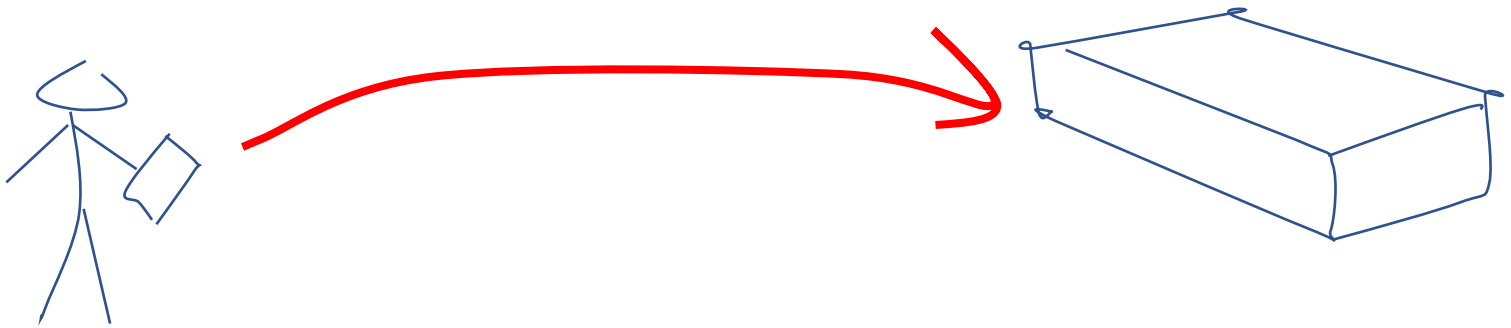
Practically, this means that transport-protocol aware packet processors/switches need to decode the extension header chain, if its present, which can consume additional cycles to process/switch a packet – and the additional time is not predictable. For trailing frags there is no transport header!

Or the transport-protocol aware unit can simply discard all IPv6 packets that contain extension headers!

Are the effects of middleware symmetric?

The RFC7872 experiments sent altered IPv6 packets **towards** well-known servers and observed whether the server received and reconstructed the altered packet by seeing whether the server responded (or not)

Sending large fragmented queries towards servers is not all that common – the reverse situation of receiving big responses is more common

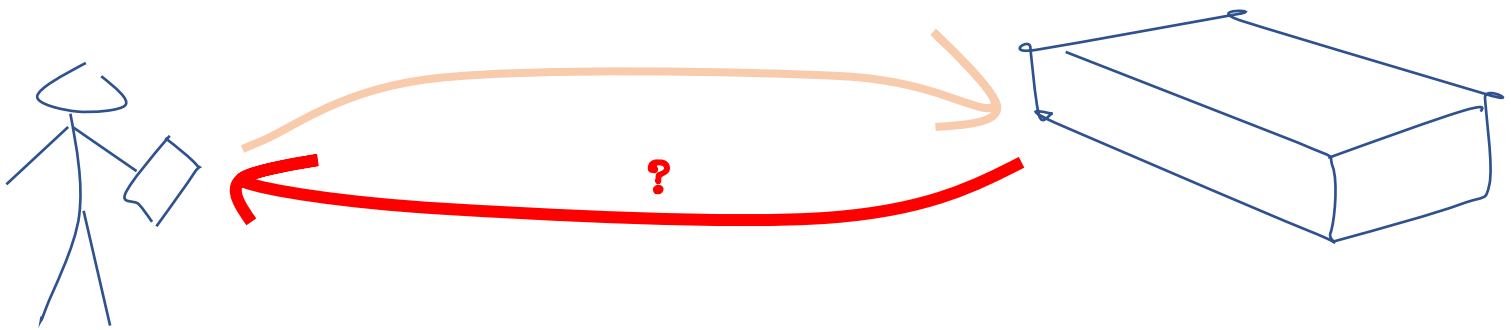


Are the effects of middleware symmetric?

The RFC7872 experiments sent altered IPv6 packets **towards** well-known servers and observed whether the server received and reconstructed the altered packet by seeing whether the server responded (or not)

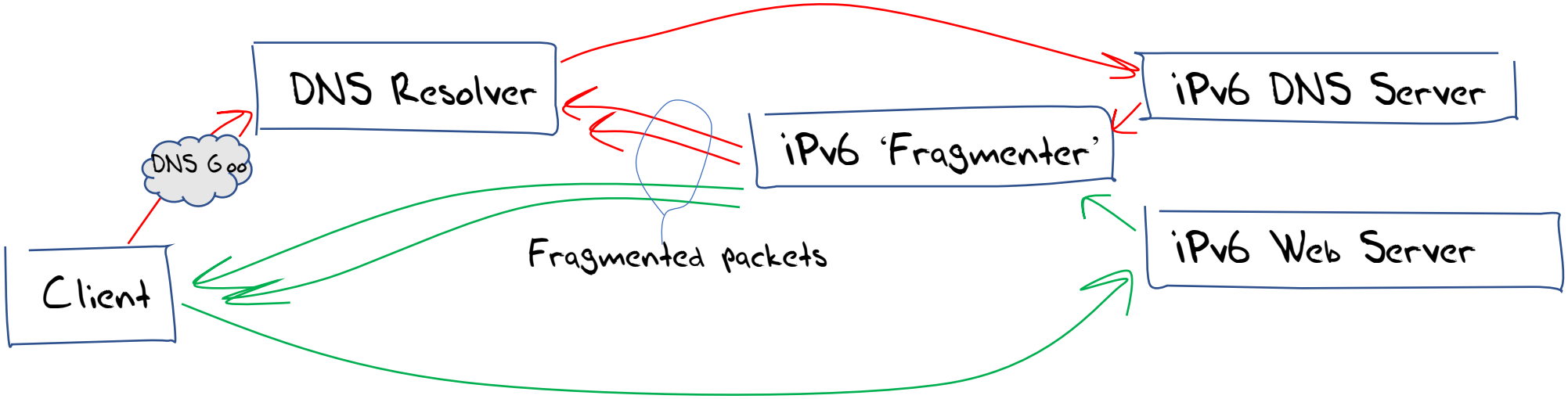
Sending large fragmented queries towards servers is not all that common – the reverse situation of receiving big responses is more common

What happens if we try to reproduce this experiment by looking at what happens when we send various forms of altered IPv6 packets **back** from servers – what's the drop rate of this reverse case for packets from servers to end-clients?



IPv6 Fragmentation Extension Header Handling

We used an ad-based measurement system, using a custom packet fragmentation wrangler as a front end to a DNS and Web server to test IPv6 fragmentation behaviour



APNIC Test - August 2017

- Use APNIC IPv6 measurement platform to test the drop rate of IPv6 fragmented 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%

This is an improvement over the RFC 7872 measurement, which reported a 28÷ drop rate client to server

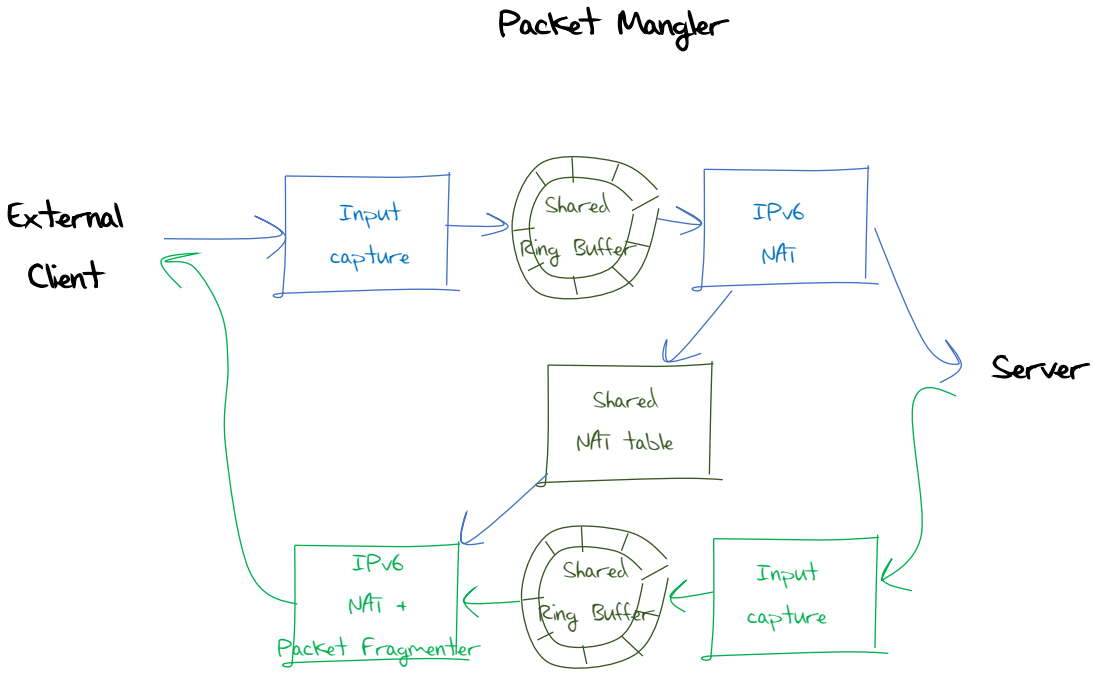
APNIC Test - 2021 onward

Re-work of the 2017 measurement experiment

- Same server-to-client TCP session fragmentation mechanism
- Uses a dedicated middlebox to fragment outgoing packets to improve packet handling capacity of the experiment
- 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

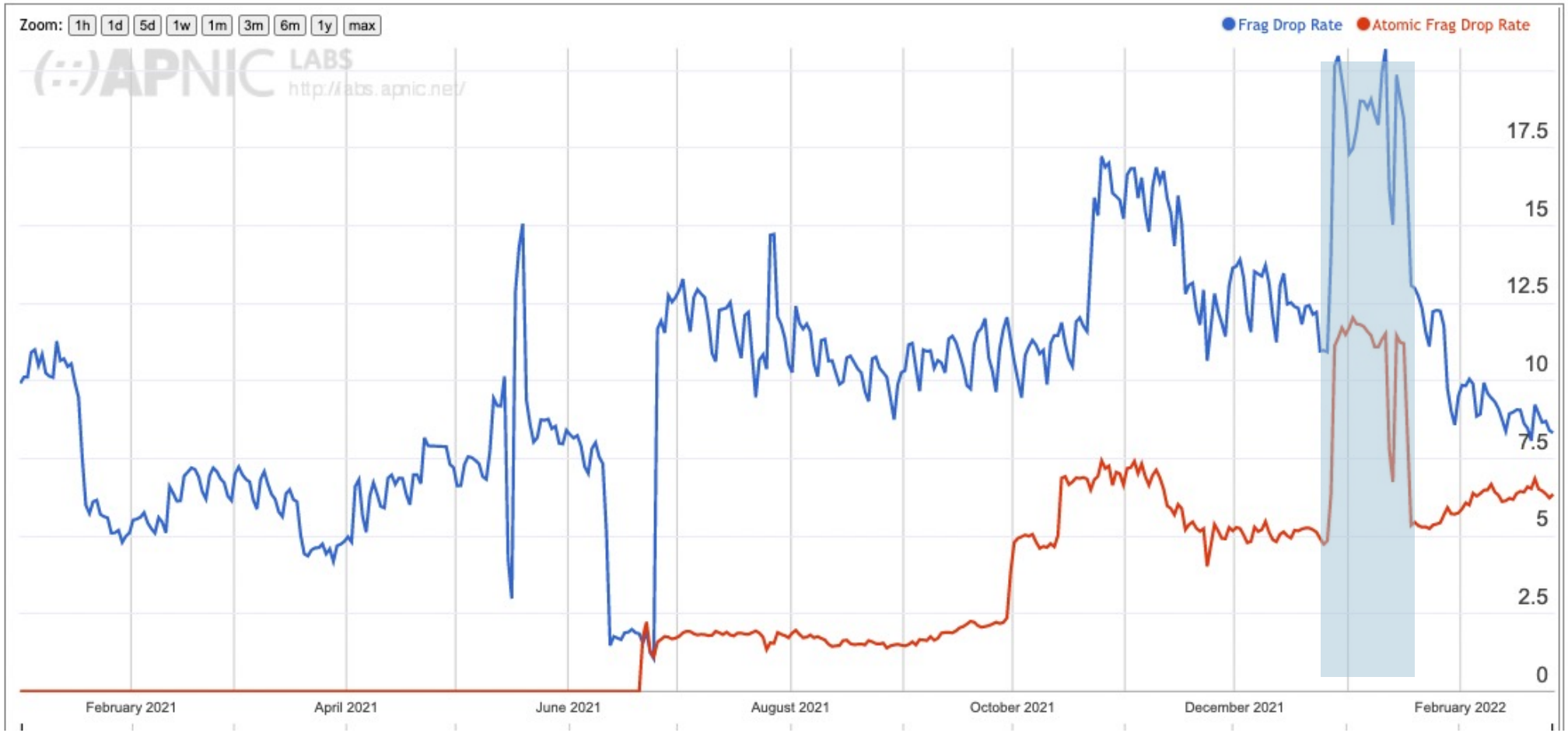
IPv6 Packet Mangler Design

- Easier said than done when we are limited to user-space code on standard cloud processing platforms
- We needed a promiscuous capture mechanism that works in user space. We used the *libpcap* libraries to perform packet capture, and used *IP packet filters* to stop the kernel's persistent desire to send RST packets
- The problem is that the pcap libraries have no buffers so were are dropping packets under load. We resorted to using multiple processes and shared memory ring buffers to improve throughput
- We also split the back end server and the packet mangling function to separate units, so we also implemented an IPv6 NAT function in the packet mangler again as a shared memory structure



2021 Fragmentation Drop Rate

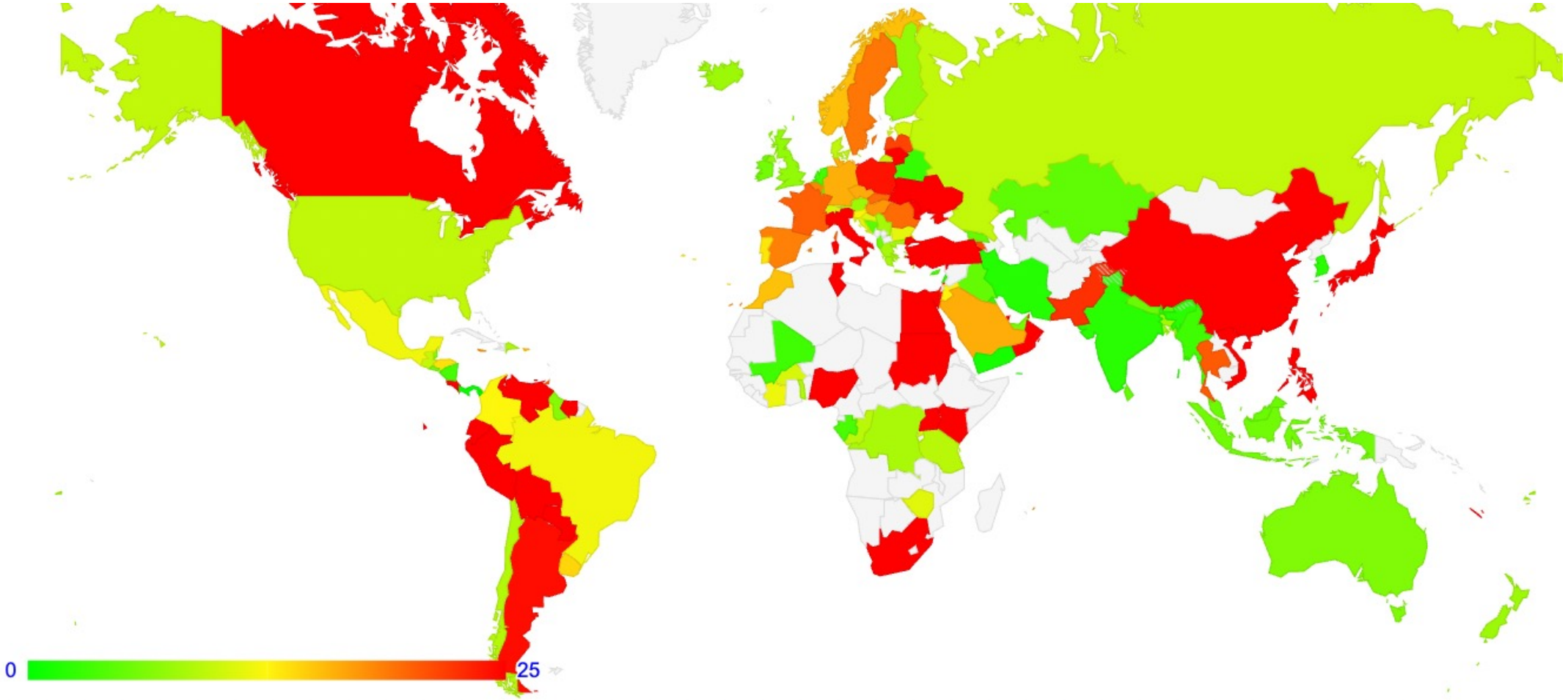
Use of V6FRAG Drop Rate for World (XA)



This is a significant improvement over 2017 data

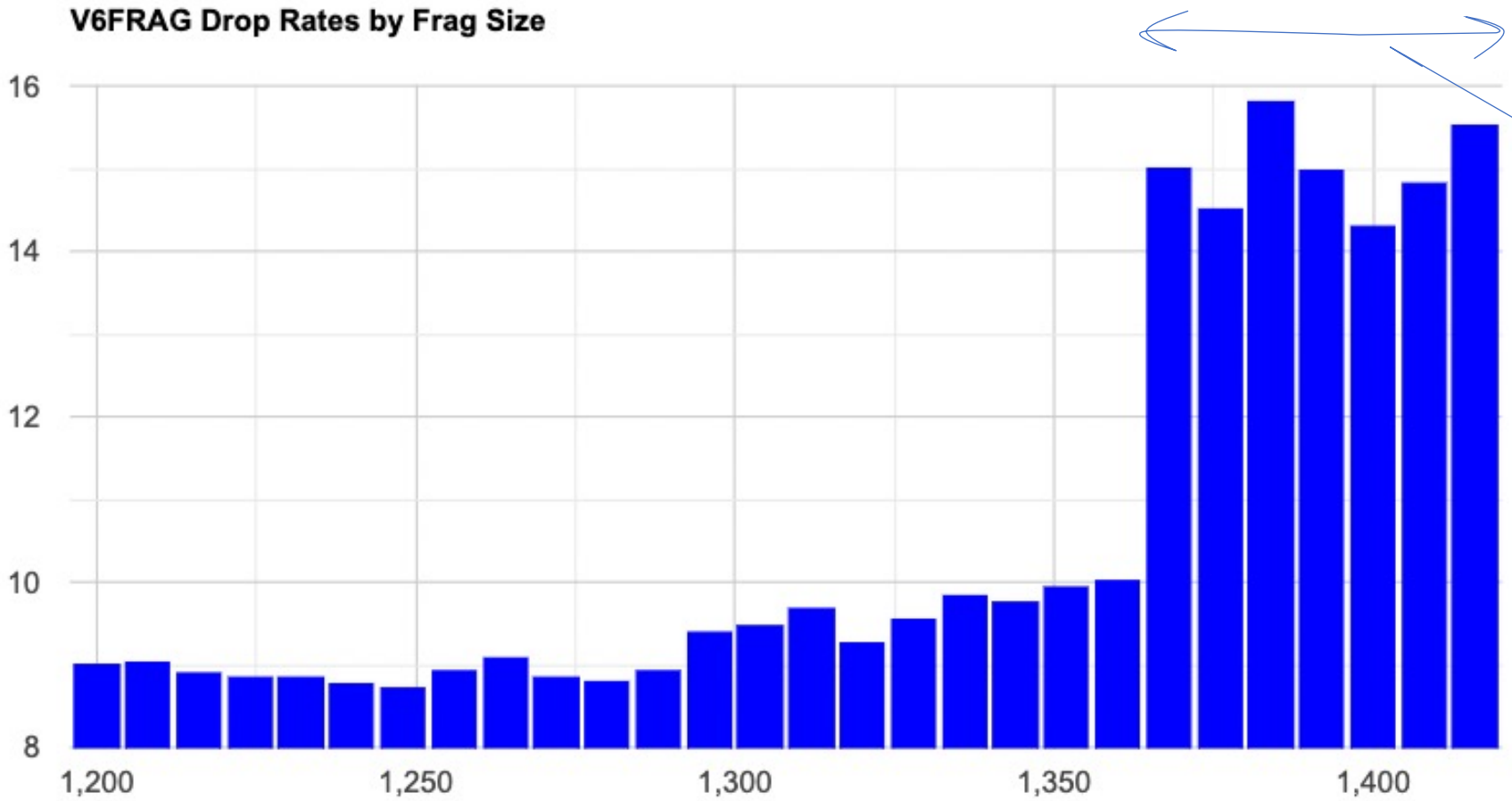
Since 2017 there are 10x the number of IPv6 users and the fragmentation drop rate has come down by 2/3 – we appear to be getting better at handling IPv6 fragments in the longer term!

2021 Fragmentation Drop Rate



More recent IPv6 deployments appear to be better at frag handling than more mature ones

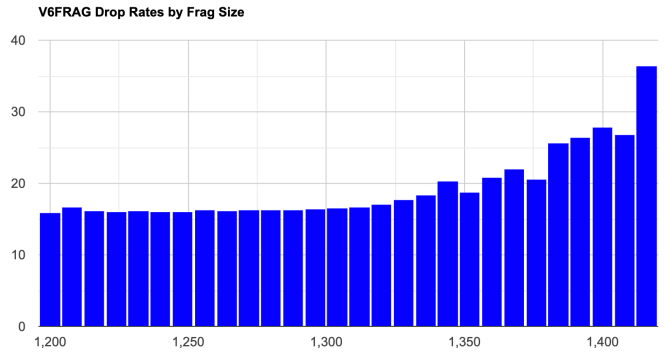
Drop Rate by Size



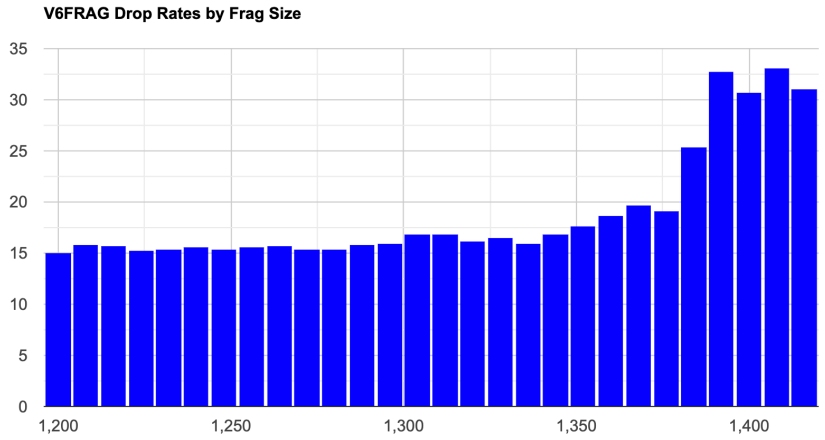
This is unexpected. Why does the drop rate increase so markedly when the fragmented packet size passes over the threshold of 1,360 octets?

Drop Size Profile by Region

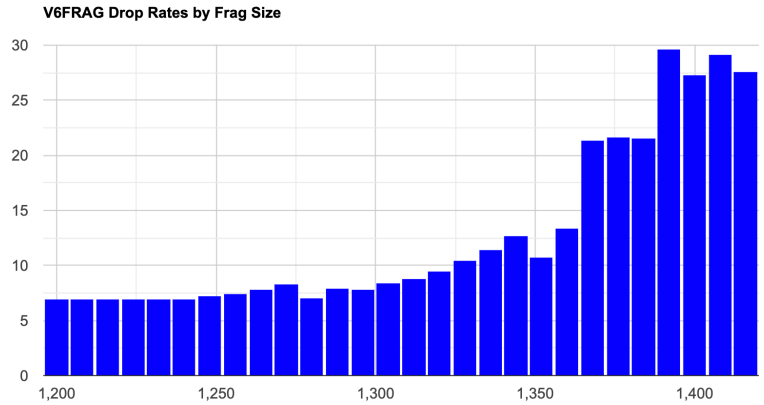
Americas



Europe



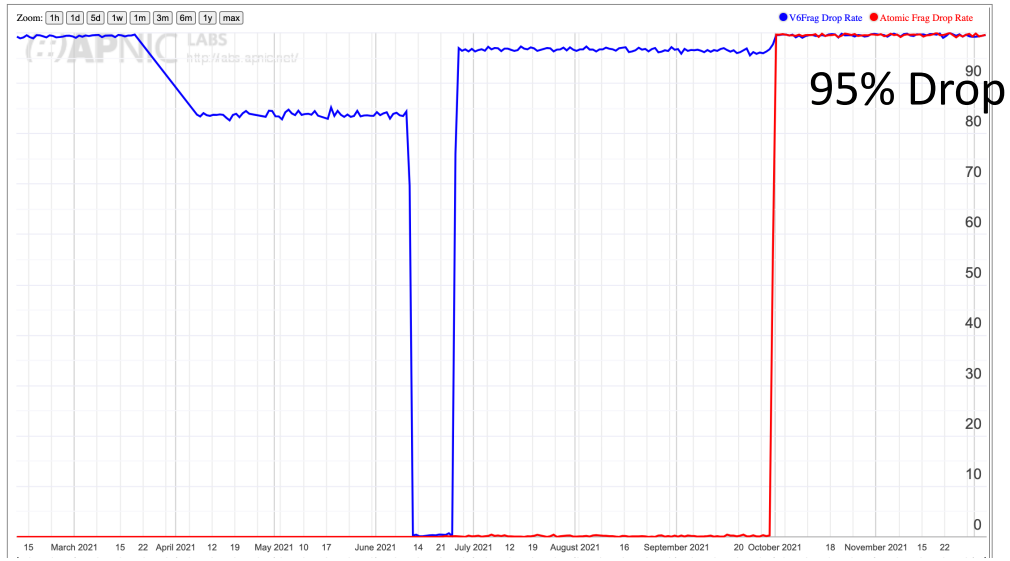
Asia



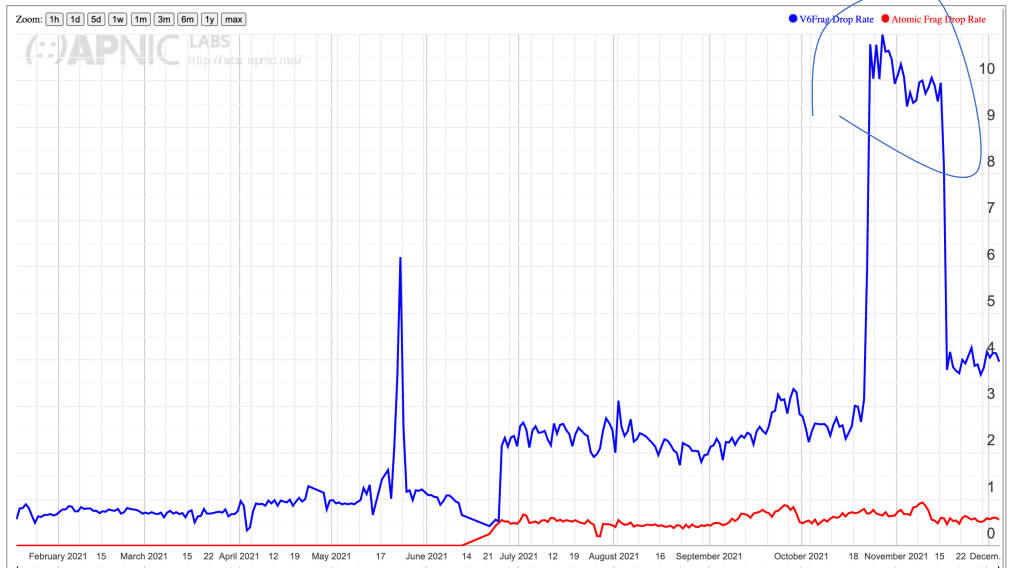
Why?

- 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)



V6Frag Drop Measurement for AS45609: BHARTI-MOBILITY-AS-AP Bharti Airtel Ltd. AS for GPRS Service, India (IN)



4% Drop!

Why?

Other potential factors that could contribute:

- Local security policies in user-facing edge devices
- IPv6 EH may trigger “slow path” processing in network equipment that could lead to higher drop rates
- IPv6 Path MTU woes!
- TCP MSS settings interfere with the measurement

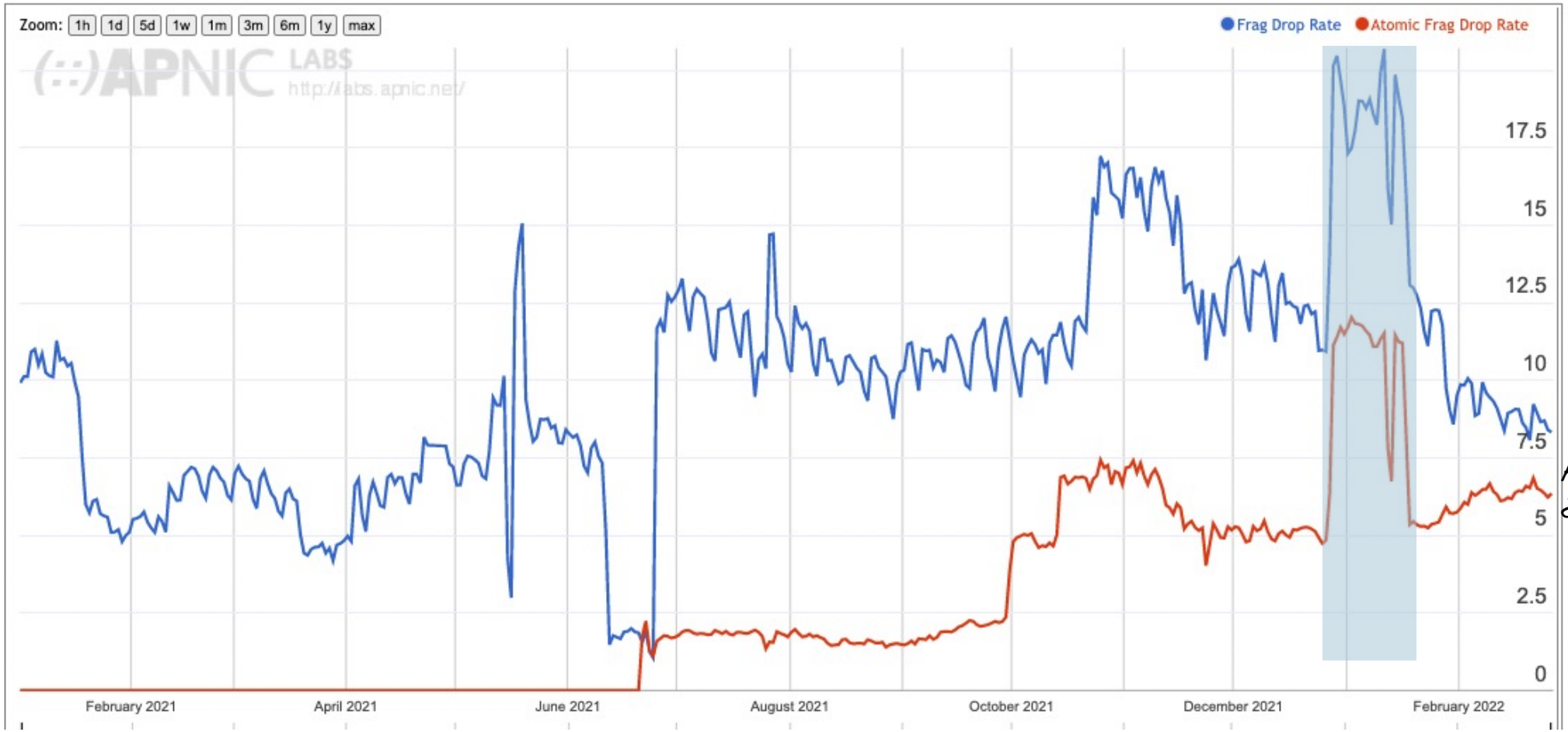
Frag Drop or EH Drop?

We added a further test to try and see the difference between Fragmentation and Extension Headers

- We used an ”atomic” fragment, which is a IPv6 packet with a Fragmentation Header where the fragment offset is 0 and the M (more) bit is also 0

2021 Fragmentation Drop Rate

Use of V6FRAG Drop Rate for World (XA)

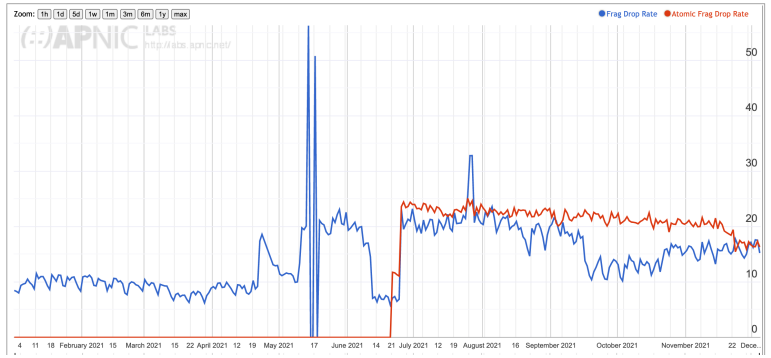


Atomic Fragment drop rate is 6÷

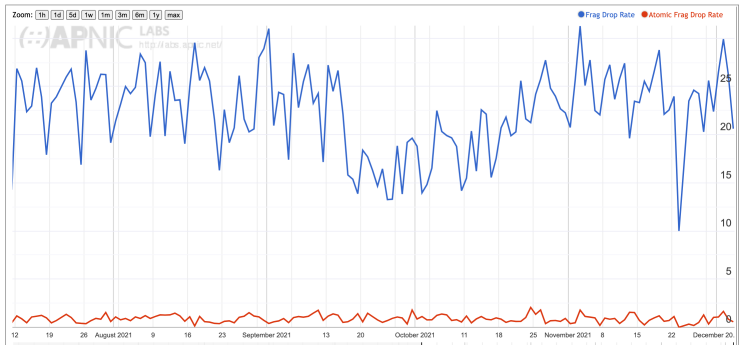
Frags vs Atomic Frags

- Most of the time the Atomic Frag drop rate is ~3x lower than the Fragmented packet drop rate
- Except when it's not!

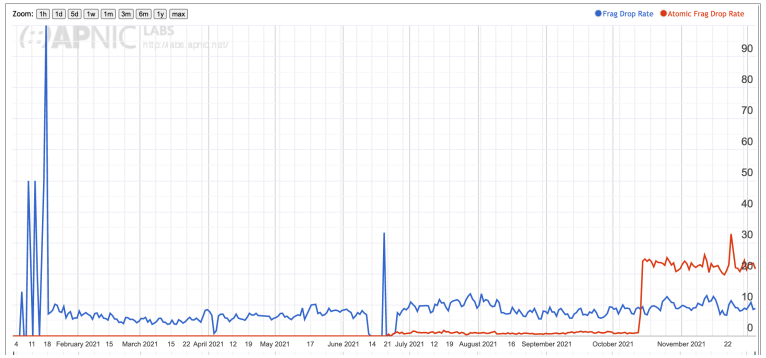
Use of V6FRAG Drop Rate for Germany (DE)



Use of V6FRAG Drop Rate for Italy (IT)



Use of V6FRAG Drop Rate for Australia (AU)



Frag Drop or EH Drop?

- The Atomic Frag data isn't really as informative as we would've liked
- We thought that all hosts would accept incoming packets with an Atomic Frag header
  - After all the Atomic Frag header is a no-op for the packet!
- So if there are Atomic Frag packet drops it should be a network effect
  - The Atomic Frag drop rate should always be less than the fragmented packet drop rate
- But this is not always the case in our data
- So we looked for other Extension Headers to test

EH DST Drop?

- What's another innocuous Extension Header we can use?
- There is the PADN Destination Option
  - In this way can can compare the network treatment of extension headers to the treatment of fragmentation headers
- Let's use it!
  - Because padding does not relying on any particular functionality in the host, so hosts should accept it
- In theory Destination Options should be handled by the network as a neutral option, as the option signalling is about the destination host, not the network elements that switch the packet in transit

Destination Option Drop Rate

January 2022: 94.5% drop rate

Wow! That's awesomely bad!

It seems that most hosts are dropping incoming packets with unexpected destination options, whether they contain directives of just padding or other directives, but we need to test this against various commonly used IPv6 protocol stacks to test this a little more

EH HBH Drop?

- What's another innocuous Extension Header we can use?
- There is the PADN Hop-by-Hop Option
  - In this way can can compare the network treatment of extension headers to the treatment of fragmentation headers
- Let's try this
  - Because padding does not relying on any particular functionality in the host, so hosts should accept it
- Again, this is a simple padding option so no special processing is being requested from the network's switches

Hop-by-Hop Option Drop Rate

February 2022: 99.5% drop rate

Wow! That's awesomely even badder!

It seems that most hosts and routers are dropping incoming packets with destination and hop-by-hop options, but we need to test this some more

Summary

- The network is slowly improving it's handling of fragmented IPv6 packets
  - In 5 years its gone from *unusably bad* to *tolerably poor*
  - Recent IPv6 deployments appear to show more robust general handling of IPv6 packets
- Destination Extension Headers and Hop-by-Hop Extension Headers are a completely lost cause - they are not usefully supported on public Internet infrastructure

Lessons Learned on Fragmentation

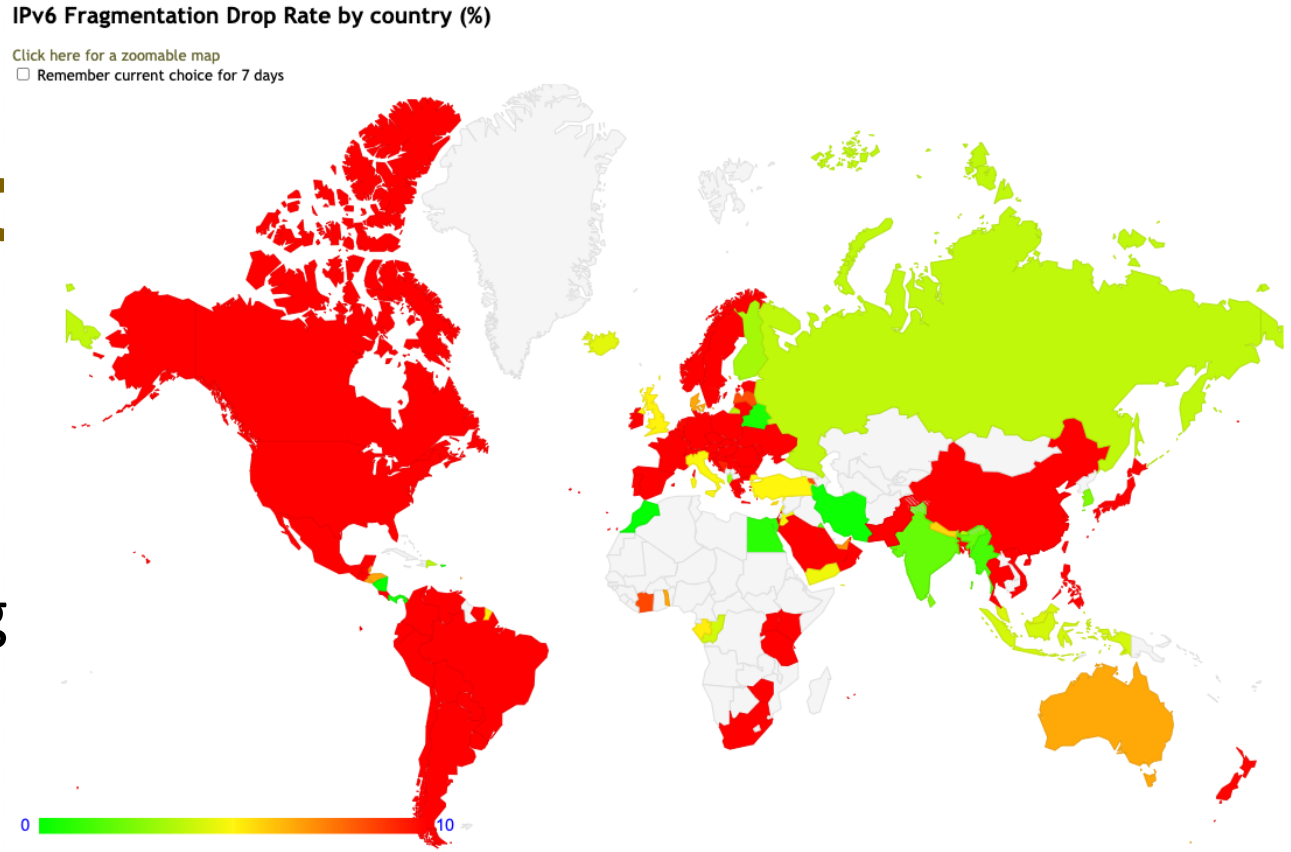
- Don’t Fragment outgoing packets
- Don’t rely on PMTUD
- More generally, don’t rely on Extension Headers nor on ICMP6 integrity
- Pick your TCP MSS setting carefully:
  - 1280 is a robust local MTU size – but possibly too conservative
  - 1350 is survivable as a MTU size – the Goldilocks choice
  - 1400 is risky – but survivable in most cases
  - 1500 is a poor choice – this leads to visible failure cases

Lessons Learned on HBH and Dst EHs

- Don’t

Daily Report

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



7 day average (08/05/2021 - 14/05/2021)

Window (Days) 7

Redraw

Code	Region	Frag Drop Rate	V6 Samples
XA	World	8.10%	18,509,740
XC	Americas	14.14%	4,474,221
XE	Europe	13.84%	1,569,881
XB	Africa	12.41%	143,540
XF	Oceania	7.13%	85,580
XG	Unclassified	5.26%	20,677
XD	Asia	5.11%	12,215,841

That's it!