1.0.0.0/8

Merit

Eric Wustrow
Manish Karir

APNIC

George Michaelson
Geoff Huston

University of Michigan

Micheal Bailey
Farnam Jahanian

Background

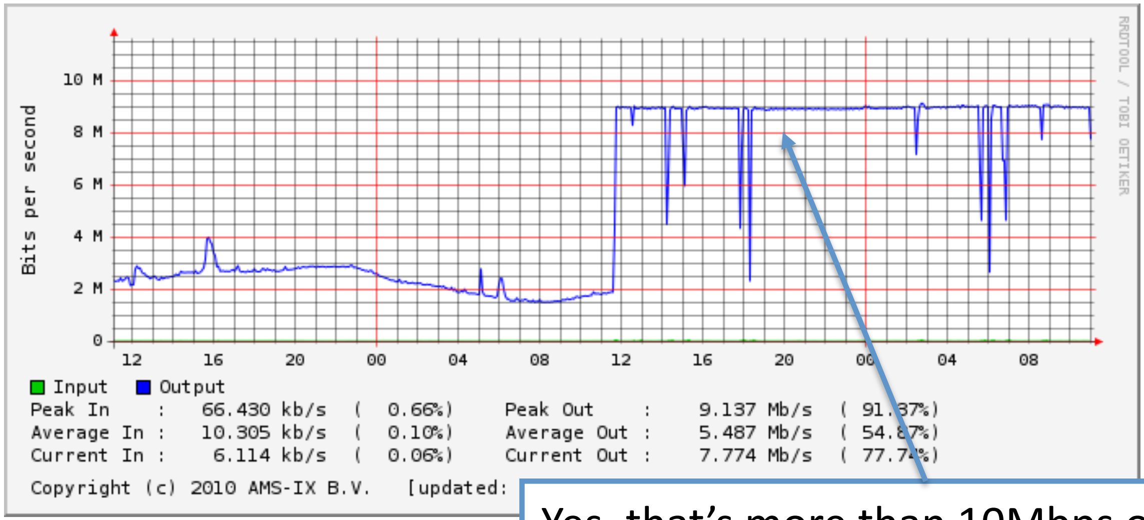
- We are now down to the last 16 /8s in IPv4 for allocation
- There is a growing concern that these blocks are increasingly less desirable
  - 'Who said the water at the bottom of the barrel of IPv4 addresses will be very pure?' – NANOG POST
  - "+1" – NANOG POST ;)
- IANA allocated 1.0.0.0/8 to APNIC in January 2010

Today’s Talk

- What is normal for an unallocated block? Is 1.0.0.0/8 any different?
  - Amount of traffic
  - Protocols used
  - Ports used
  - Source and destination distributions
- If it is different, why is it different?
- What can we do about it?

First Evidence that Something is Fishy

- 27 January 2010 RIPE NCC announces 1.1.1.0/24, 1.2.3.0/24, 1.50.0.0/22 and 1.255.0.0/16
- http://labs.ripe.net/content/pollution-18

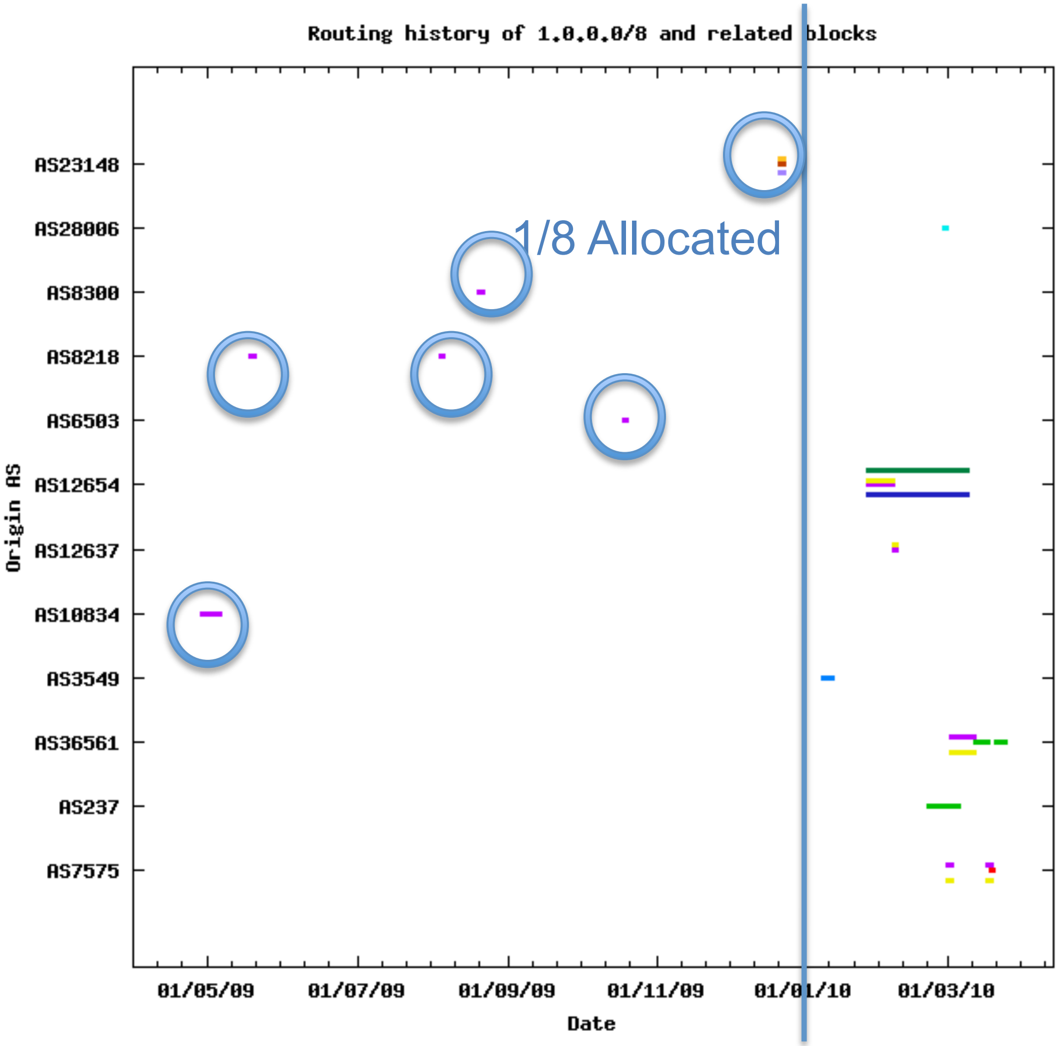


```
Peak In    :   66.430 kb/s  (  0.66%)    Peak Out    :    9.137 Mb/s  ( 91.37%)
Average In :   10.305 kb/s  (  0.10%)    Average Out :    5.487 Mb/s  ( 54.87%)
Current In :    6.114 kb/s  (  0.06%)    Current Out :    7.774 Mb/s  ( 77.74%)

Copyright (c) 2010 AMS-IX B.V.   [updated:
```

Yes, that’s more than 10Mbps of traffic!

Routing of 1.0.0.0/8



1.0.0.0/24	AS7575
1.0.0.0/8	AS237
	AS36561
1.1.0.0/24	AS3549
1.1.1.0/24	AS10834
	AS12637
	AS12654
	AS36561
	AS6503
	AS7575
	AS8218
	AS8300
1.10.25.0/24	AS28006
1.120.0.0/13	AS23148
1.2.3.0/24	AS12637
	AS12654
	AS36561
	AS7575
1.255.0.0/16	AS12654
1.40.0.0/13	AS23148
1.50.0.0/22	AS12654
1.80.0.0/13	AS23148

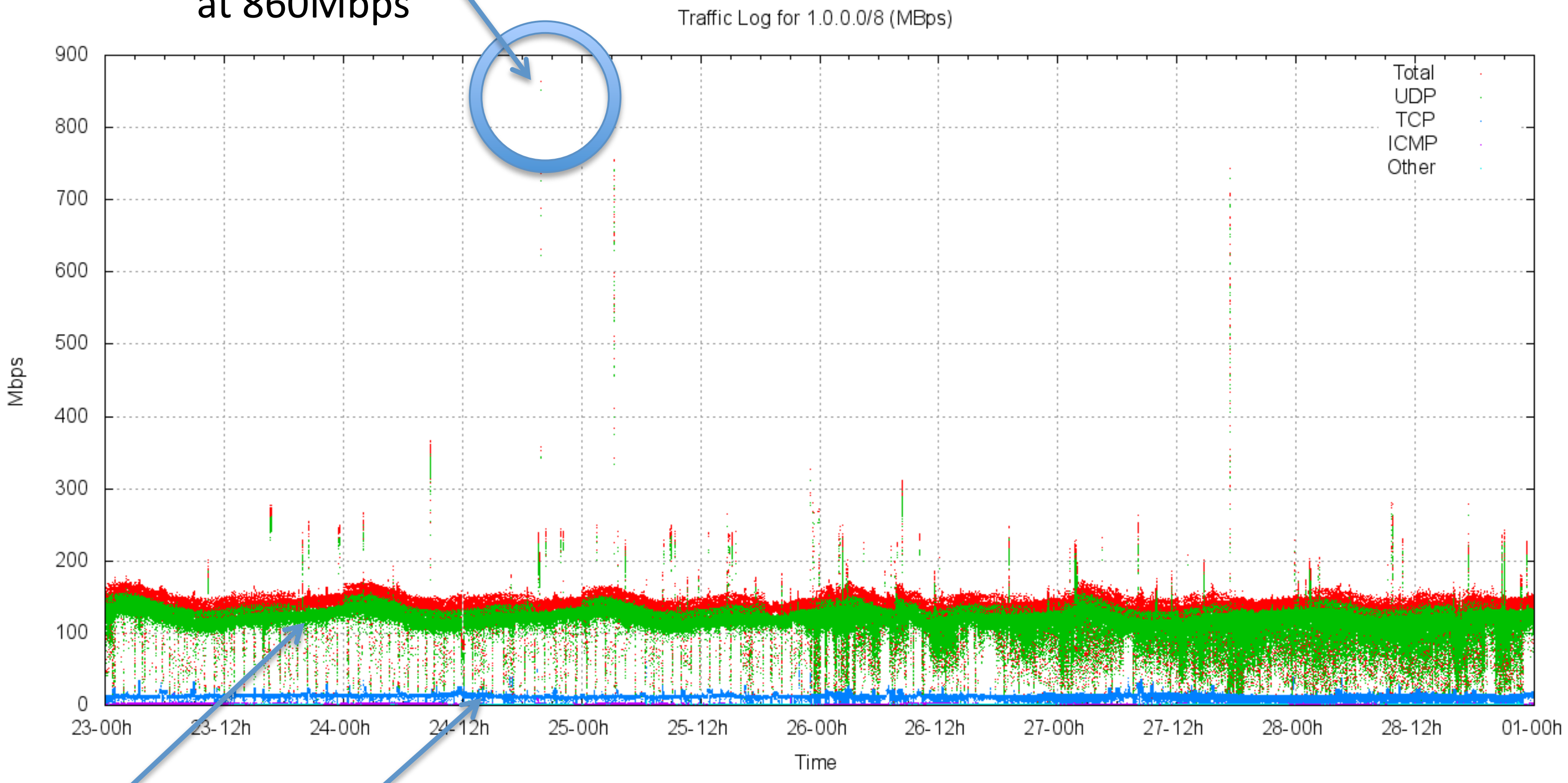
http://albatross.ripe.net/cgi-bin/rex.pl

Ok but how much of a problem is this?

- Merit (AS237) announced 1.0.0.0/8 from 23 Feb until 1 March 2010
  - Collected 7.9Tb of packet capture data

Traffic to 1.0.0.0/8

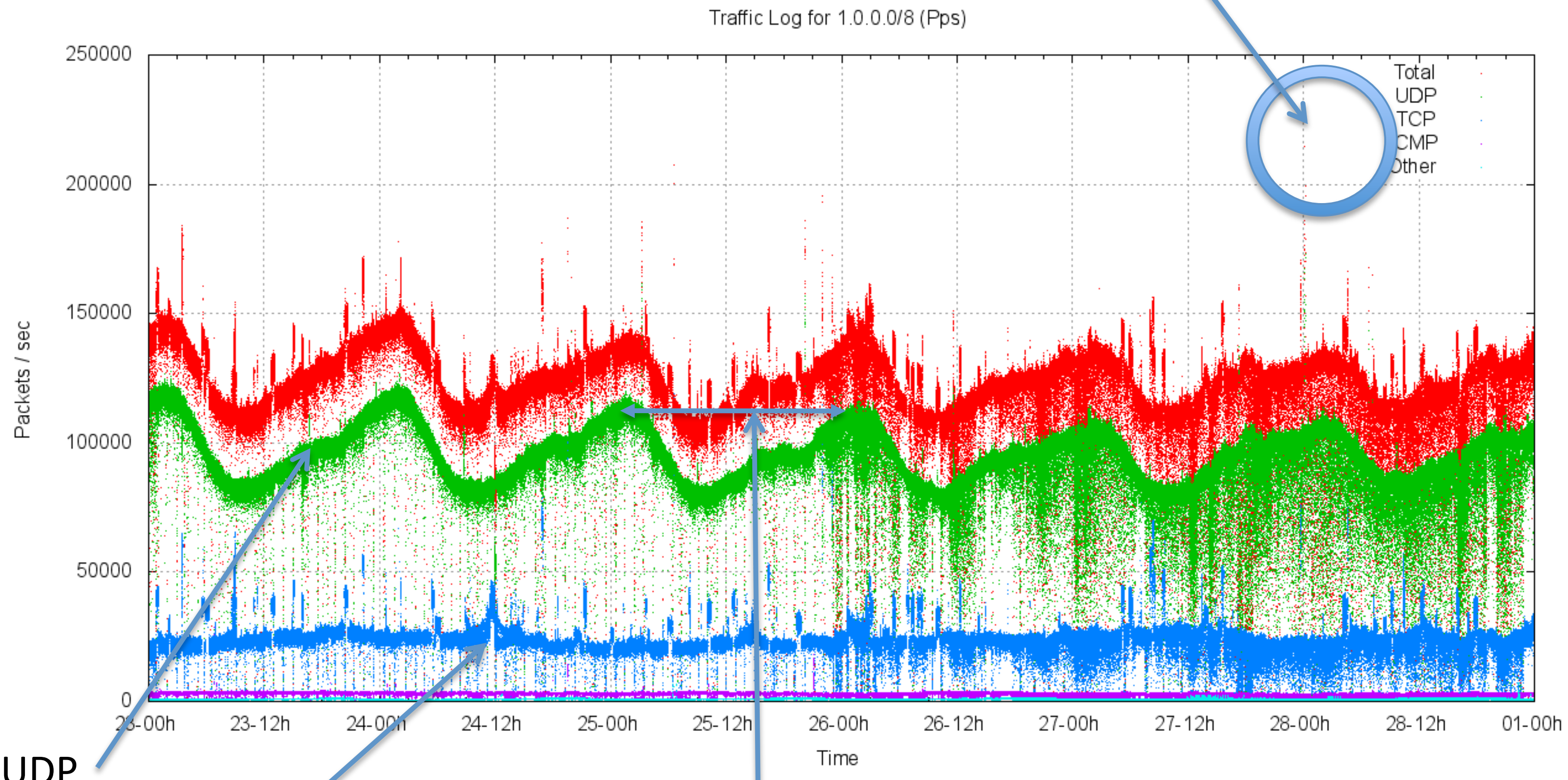
Peak Burst
at 860Mbps



UDP

TCP

Packet Rate to 1.0.0.0/8



Peak Burst
at 220Kpps

UDP

TCP

Marked UDP diurnal pattern

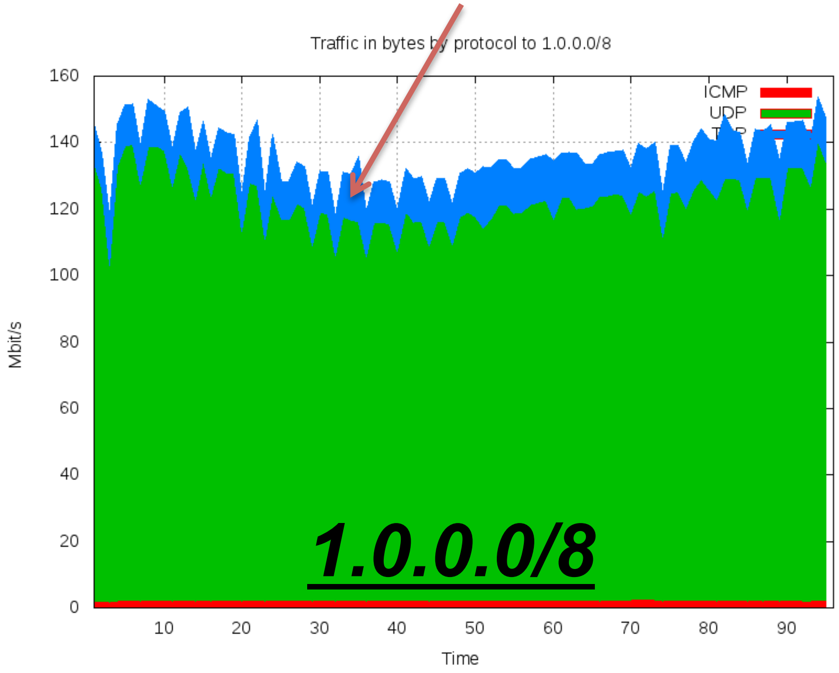
But how abnormal is this?

- Merit (AS237) announced 1.0.0.0/8 from 23 Feb until 1 March 2010
- Merit announced 35.0.0.0/8 during the same period. Unused minus a single /17 block.

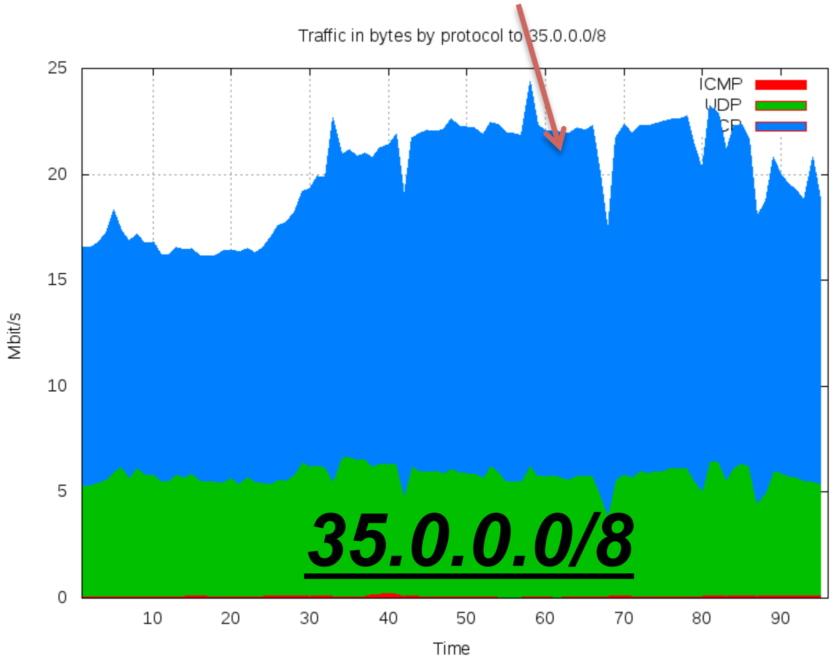
Is 1/8 Normal? No Way!

Total Volume

130-150 Mbps ≠ 15-25 Mbps



1.0.0.0/8



35.0.0.0/8

1. UDP
2. TCP
3. ICMP

Protocol Distribution

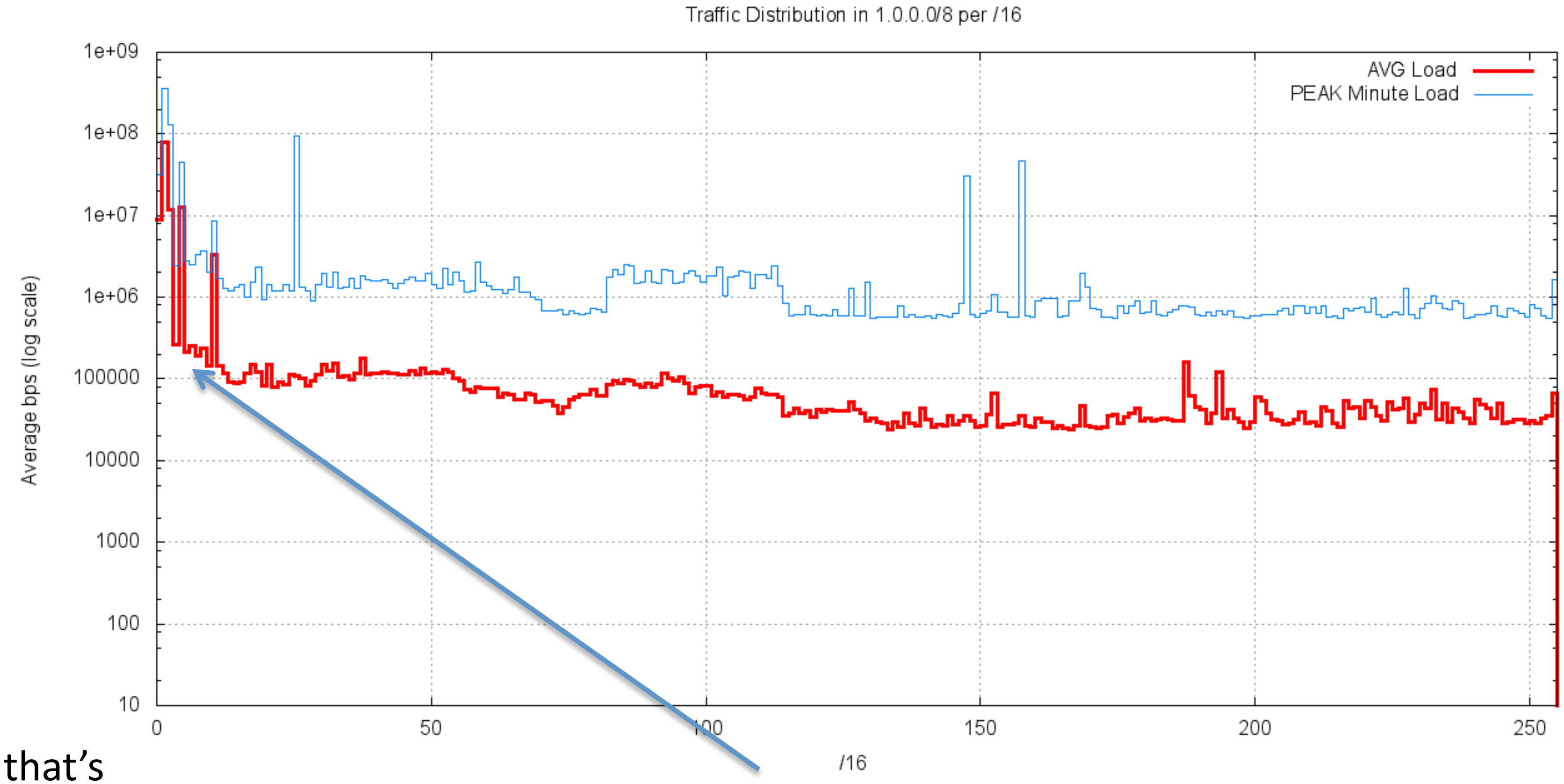
≠

1. TCP
2. UDP
3. ICMP

Comparing Pollution Types

- 1/8 (% of packets):
  - Scanning: 17.9% (12.5B)
  - Backscatter: 1.9% (1.34B)
  - Misconfiguration (Other): 80.2%
- 35/8 (% of packets):
  - Scanning: 69.7% (15.5B)
  - Backscatter: 6.2% (1.39B)
  - Misconfiguration (Other): 24.1%

What's going on?



Yes, that's
a Log Scale!

The "hot spots" appear to lie in the low /16s

Top 10 Contributors are 75% of Packets

Subnet /24	Packets	%
1.1.1.0	4797420185	44.5
1.4.0.0	1884458639	17.5
1.0.0.0	1069156477	9.9
1.2.3.0	199452209	1.8
1.1.168.0	62347104	0.5
1.10.10.0	26362000	0.2
1.0.168.0	18988771	0.1
1.1.0.0	18822018	0.1
1.0.1.0	14818941	0.1
1.2.168.0	12484394	0.1

1.1.1.1:15206

- For 1/8, 34.5% of all packets (and 50.1% of all bytes) received are UDP packets to 1.1.1.1, destination port 15206.
  - Compare to 35/8, which on the same UDP port (across the entire /8) received a total of 4703 packets (0.00066%) in one day.

What are they?

- Most of the payloads looks like version 2 RTP packets
  - 75% of all bytes to this port have 0x8000 first 16 bits (first two bits is the version number and the next 14 all 0)
  - the majority of packets are 214 bytes in size (89.4%)
  - the vast majority (97.3%) of them are even ports (hinting at RTP data)
- Hand full of bad applications devices
  - All this coming from only 1036 /24s in 1 day of data
  - And from only 1601 source ports seemingly unrelated to the ephemeral port ranges

It turns out, the 1.0.0.0/8 traffic is mostly audio data!

- Took one stream, from XXX.148.35.10, source port 13464 and noticed the PT field was 00
  - PCMU, a raw-ish (compressed dynamic range) audio wave format.
- Converted this into a .au file using wireshark, and it is indeed an audio file. Take a listen for yourself:

1.4.0.0

- For 1/8, 17.5% of all packets (and 10% of all bytes) received are UDP packets to 1.4.0.0, destination port 33368, 514, 33527, 3072, 33493
  - Surprisingly most of these could be interpreted as DNS traffic of different types, A, AAAA, MX, etc.
  - Possibly sourced from ASUS ADSL modem
  - Most appear to be misdirected queries:
    - hotelnikkohimeji.co.jp.
    - x.myspacecdn.com
    - typepad.com
    - th411.photobucket.com

1.2.3.4:5001

- Traffic to 1.2.3.0 is 1.8% of all packets
- Iperf traffic to 1.2.3.4 is roughly 10Mbps of traffic from less than a 100 unique sources
- The top contributor (a single IP from 41.194.0.0/16) sent roughly 70M pkts/day

rfc1918 analysis (or is it rfc32263?)

- Some other popular destinations are 1.1.168.0, 1.0.168.0, 1.2.168.0?
- Most of the packets are going to:1.1.168.192, 1.0.168.192, 1.2.168.192.
- These IPs are really just 192.168.x.1, in host-byte order (little-endian), someone is not doing a proper htonl(ip_addr); somewhere, and we are catching the data.
- Destination port 80, over UDP (yeah...UDP, not TCP), length = 1, and data of 0x31

What can we do about it?

- APNIC suggested that the following /24s be withheld from general allocation:
  - 1.0.0.0/24
  - 1.1.1.0/24
  - 1.2.3.0/24
  - 1.4.0.0/24
  - 1.10.10.0/24
- If further investigation reveals that the traffic to any of these /24s abates to a normal background level in the future, then these addresses would be returned to the APNIC unallocated address pool at that time.

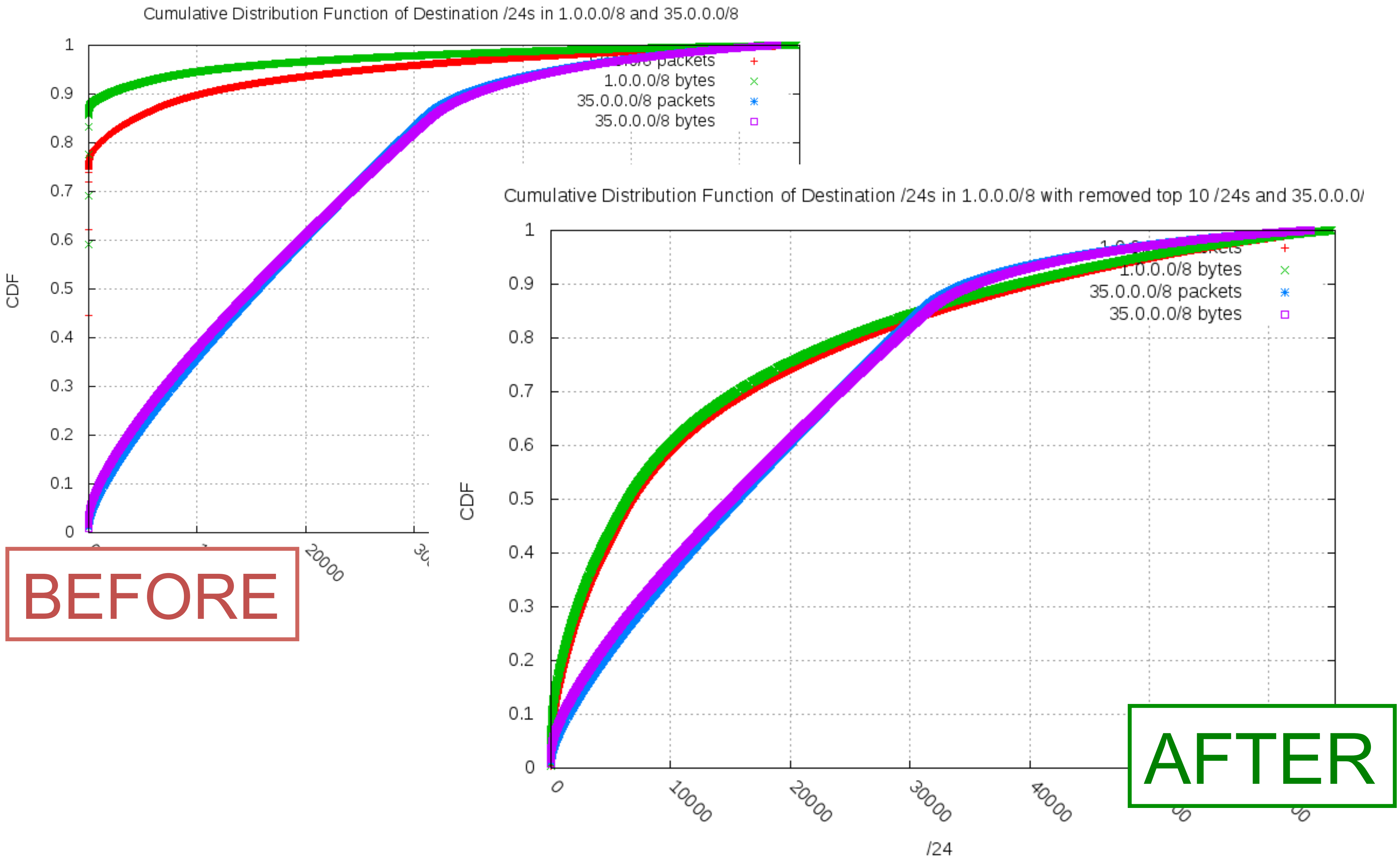
What can we do about it (cont)?

- It is recommended that the following /16s be temporarily marked as reserved and withheld from general allocation by APNIC:

1.0.0.0/16	1.5.0.0/16	1.20.0.0/16
1.1.0.0/16	1.6.0.0/16	1.32.0.0/16
1.2.0.0/16	1.7.0.0/16	1.37.0.0/16
1.3.0.0/16	1.8.0.0/16	1.187.0.0/16
1.4.0.0/16	1.10.0.0/16	

- These /16s should be marked as allocated to APNIC R&D to allow further short term experimentation in the distribution of unsolicited background traffic to these addresses to be conducted by APNIC

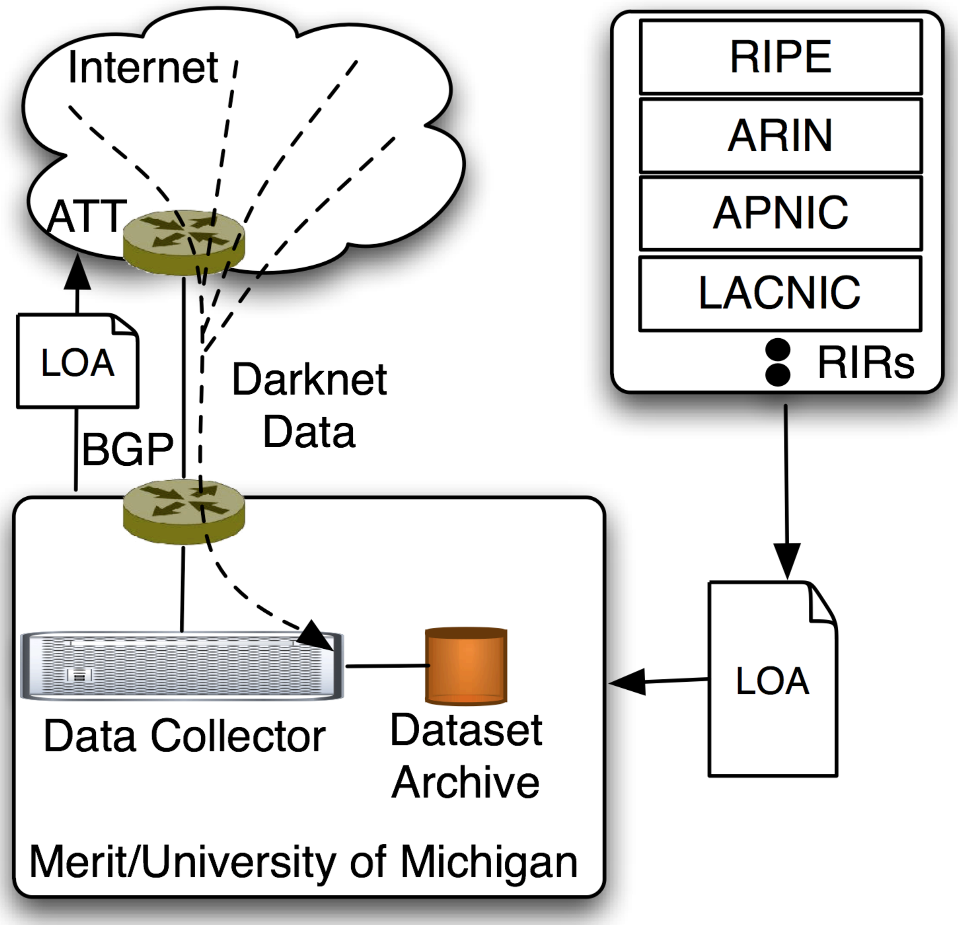
Would eliminating hotspots help?



The Broader View

- Pollution is not limited to 1/8. Evidence of similar types of pollution in 50/8, 107/8, 14/8, 223/8
- Hotspots can exist in strange and unusual places
- Pollution can come from strange and unusual sources (in addition to scanning and backscatter)
  - System Misconfiguration – syslog, DNS
  - Programming errors – htonl(), bit-torrent
  - Hardcoded defaults – SIP, dsl modems
  - Experiments gone wild! – iperf testing
- Need to develop a consistent methodology for identifying these hotspots and a policy on cleanup or quarantine

A Framework for Internet Pollution Analysis



- Work with RIRs to identify upcoming allocation
- Obtain LOA
- Advertise, Collect, Analyze, Archive, Provide to research community
- Cleanup/Quarantine recommendations

Conclusions (1)

- Unchecked Internet pollution has the potential to render portions of valuable address space unusable
- In some cases cleanup is actually possible if you can identify the source (IP, application, system, protocol, document)
- Internet pollution is only one aspect of usability of an address block
  - Reclaimed address space might be on blacklists such as SPAM and botnet lists
- Current approach is to return a polluted block and request an alternate allocation, but that might not be feasible for much longer

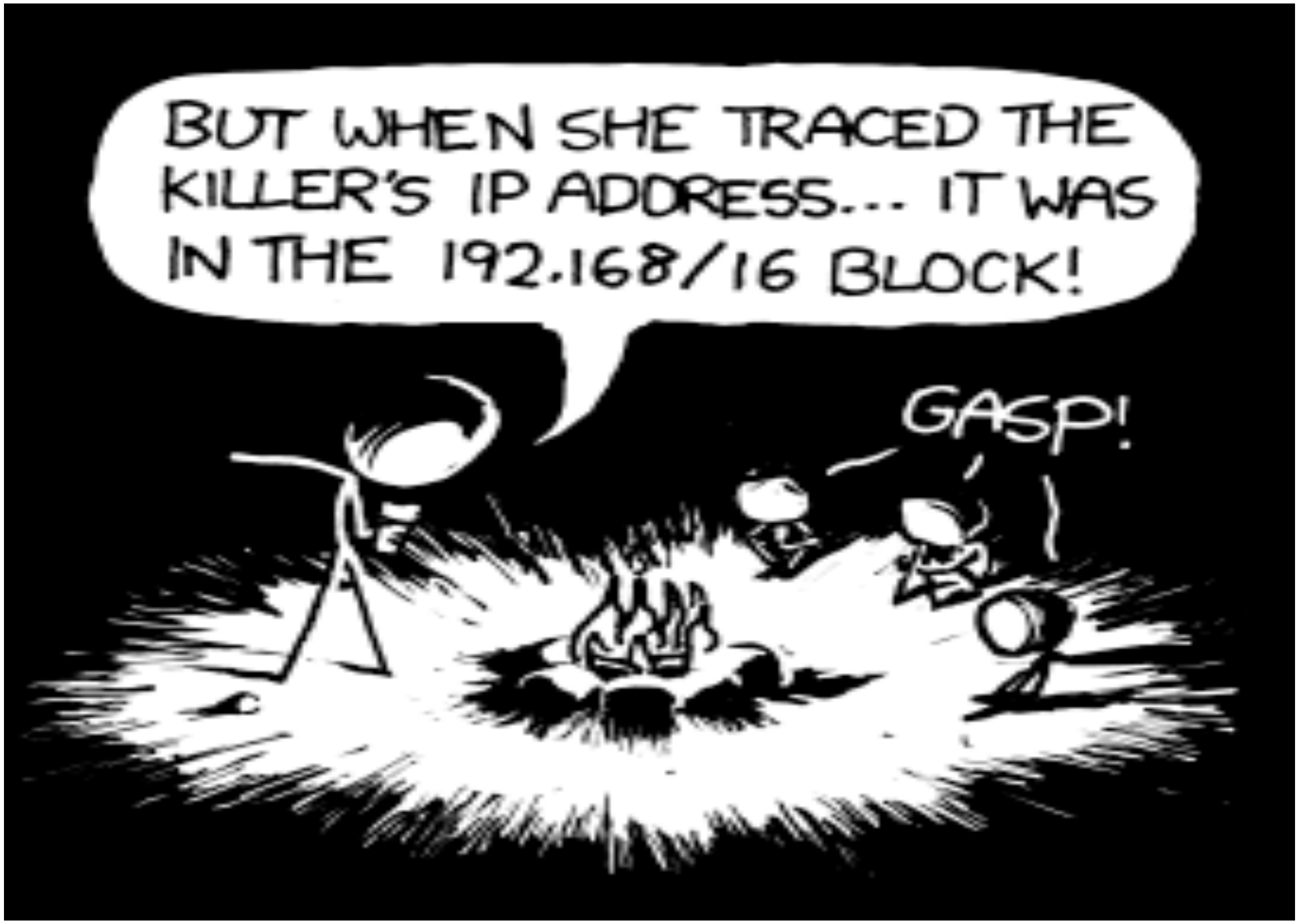
Conclusions (2)

- Who is responsible for the quality of the address block being allocated, does this have the potential to affect pricing should an address space market emerge
- We currently have collected data for 8 x.0.0.0/8 net blocks - 2 more in the next few weeks.
- Roughly 10TB of data collected - will be made available to researchers/community via the DHS funded PREDICT data repository

Additional Reading

- Some additional details:
  - Tech Report: https://www.eecs.umich.edu/techreports/cse/2010/CSE-TR-564-10.pdf
  - http://www.potaroo.net/studies/14-223-slash8/14-223-slash8.html
  - http://software.merit.edu/darknet

Obligatory


BUT WHEN SHE TRACED THE KILLER'S IP ADDRESS... IT WAS IN THE 192.168/16 BLOCK!
GASP!

[Source: http://xkcd.com/742/]