Mapping the Great Void Smarter scanning for IPv6 Richard Barnes, Rick Altmann, Daniel Kerr BBN Technologies

Agenda

- Challenges for mapping the IPv6 Internet
- Some approaches to smarter scanning
 - CIDR++
 - Registry information
 - Addressing heuristics
- Empirical results

Background: IPv6 is big

IPv6 address space is big

• How do you select the networks you trace to?

- Ark IPv4: Each /24 covered by a BGP prefix
- Ark IPv6: One per prefix advertised in BGP
- Supposing we view a /48 as functionally similar to a / 24...
 - IPv4: 12,577,420 / 24s advertised (~ 2^{23.6})
 - IPv6: 3,523,931,041 /48s advertised (~2^{31.7})
 ... and that's with the current level of IPv6 deployment
- And really, /48s get subdivided too

http://www.caida.org/workshops/isma/1102/slides/aims1102_yhyun_ark.pdf RouteViews RIB from WIDE collector, 2011/12/22

General Approach: Adaptive Probing

- Learn from previous rounds of probes to predict where you should probe next
- In the IPv4 context, focus has been on reducing impact of comprehensive measurement traffic
 - DoubleTree / Interface Set Cover algorithms find minimal set of paths to cover all interfaces
- In IPv6, focus is more on discovering the most subnets / interfaces in a feasible number of measurements
 - Some algorithms don't scale to IPv6 (e.g., subnet-centric)

Smarter Scanning

Going beyond BGP

- To tell two networks apart in measurements, we need to trace to a target in each of them
- Finding networks via pure random scanning within BGP-announced prefixes doesn't scale
- Start with BGP, add more information
 - Small amounts of randomness
 - Registration information (WHOIS)
 - Information gathered in earlier scans

Testing Methodology

- 5 nodes from commercial VPS services
- ICMP Paris traceroutes to selected targets
- Metric: Discovered addresses (no alias resolution)



Baseline: BGP

Technique	Traceroute Targets / Monitor	Monitors	Total Measureme nts	Discovered Interface Addresses	Gain Rate (New Hops Per Trace)

BGP+4

- Some networks do a little bit of subdivision of an advertised prefix, but maybe not much
- Take each prefix from BGP
- Compute 16 subnets you can get by adding 4 random bits
 - Random scanning, but bounded increase in work (16x)

BGP+4

Technique	Traceroute Targets / Monitor	Monitors	Total Measureme nts	Discovered Interface Addresses	Gain Rate (New Hops Per Trace)
BGP	8380	5	41900	16986	0.405
BGP+4	73407	5	367035	20434	0.056

BGP∩WHOIS + Rand48

- People sometimes register WHOIS information at a higher level of granularity than they advertise in BGP
- Download bulk WHOIS information and build a list of prefixes from inet6num objects
- Find routable WHOIS prefixes, covered by prefixes advertised in BGP
- If a given BGP prefix has no more specifics in WHOIS, sample five random /48s

BGP∩WHOIS + Rand48

Prefix		Network	BGP	Gain
2a02:f8:7:1a::/64	IT	AISA-NET-1	/32	32
2a01:4f8:141:22::/64	DE	FORMER-03-GMBH	/32	32
2406:4800::/64	SG	DOCOMOinterTouch-HQ-V6	/40	24
2405:2000:ff10::/56	IN	CHN-CXR-TATAC	/32	24
2607:f6f0:100::/56	US	EQUINIX-EDMA-V6-CORP-01	/40	16
2001:42c8:ffd0:100::/56	ZA	CAPETOWN-KLT-TATA	/32	24

BGP∩WHOIS + Rand48

Technique	Traceroute Targets / Monitor	Monitors	Total Measureme nts	Discovered Interface Addresses	Gain Rate (New Hops Per Trace)
BGP	8380	5	41900	16986	0.405
BGP+4	73407	5	367035	20434	0.056
BGP ∩ WHOIS + Rand48	90817	4	363268	40074	0.110

Sequence Completion

- As we do traceroutes, we get addresses back in the source addresses of responses
- Sometimes these addresses hint at the use of addressing schemes
- Look for runs within each hex digit, then complete sequences

2001:db8:1:47c8::797f 2001:db8:1:47c9::47db 2001:db8:1:47cb::8a03 2001:db8:1:47cd::4d33 2001:db8:1:47cf::b221 2001:db8:1:47c7::/48 2001:db8:1:47c8::/48 2001:db8:1:47c9::/48 2001:db8:1:47ca::/48 2001:db8:1:47cb::/48 2001:db8:1:47cc::/48 2001:db8:1:47cd::/48 2001:db8:1:47cf::/48 2001:db8:1:47cf::/48

Sequence Completion



2a01:198:200:000::/52 2a01:198:200:100::/52 2a01:198:200:200::/52 2a01:198:200:300::/52 2a01:198:200:400::/52 2a01:198:200:500::/52 2a01:198:200:600::/52 2a01:198:200:700::/52 2a01:198:200:800::/52 2a01:198:200:900::/52

Scanning within the /40... Completing the sequence...

Sequence Completion

Technique	Traceroute Targets / Monitor	Monitors	Total Measureme nts	Discovered Interface Addresses	Gain Rate (New Hops Per Trace)
BGP	8380	5	41900	16986	0.405
BGP+4	73407	5	367035	20434	0.056
BGP ∩ WHOIS + Rand48	90817	4	363268	40074	0.110
Sequence Completion	21279.75	4	85119	22919	0.269

How much did we learn?



Overlap in Discovered Interfaces



Overlap in Discovered Interfaces



Broader or Deeper?

Three techniques show similar hop count distributions
BGP+WHOIS lower mean, but greater max by 5 hops

CDF of Paris Traceroute Hop Count



Conclusions

- CIDR prefixes derived from BGP hide a lot of topology information
- New techniques add both detail and depth relative to scanning based on BGP prefixes alone
 - "Augmented BGP": BGP+4, BGP+WHOIS
 - Inference from discovered addresses
- Each technique seems to cover different parts of the network, so combination is necessary
- Future work: Incorporate better algorithms (e.g., ISC)

Digression: Security Appliances

- There are apparently security appliances out there that respond to ICMP requests for every address in a subnet
 - Show up in measurements as highly active networks / highly connected nodes
 - May be useful for mapping out subnet boundaries
- "20% test" detects with high confidence
 - If 2 of 10 randomly chosen addresses within a network respond to pings ...
 - ... then there's probably one of these devices there.

Digression: Security Appliances



2001:0760:ffff:0124:0000:0000:0000:0043

2a02:27e8:ffff:0000:0000:0000:0000:0001

2a02:27e8:8306:8554 ff0d:4de3:d5b3:7407 2a02:27e8:1b65:b6c849bf:e806:c663:58a7 2a02:27e8:27f7:b118:b664:0a6f:e8af:2423 2a02:27e8:22c6:8fe2tb2e5;7c15:0019;99e8 2a02:27e8:10f9:8b197e61:2cd5:3fd1:a965 2a02:27e8:125d:e753381fd:1a67;45e3;063b 2a02:27e8:f202:5d7862ab:f59b/c9af/97bb 2a02:27e8:52be:55be462d:352e:bf2f;ec14 2a02:27e8:b0c6:466 23c:7/17c:9159:0ae6 2a02:27e8:1001:0f5002af:f0da;3caf:a46a 2a02:27e8:23a3:0fbf 25e:d396:678b:d741 2a02:27e8:2834:8c9 31/ca:daff:83a2:2eba 2a02:27e8:2001:efb44d26:10e5:e902:6993 2a02:27e8:1acd:174 2314:e4ce:66d0:0dca 2a02:27e8:0001:762b6c39:d3d0:46ec:abed

2001:0760:ffff:ffff:0000:0000:0000:00df 2a02:27e8:1528:eab10:3cb:d7de:00b2:7170 2a02:27e8:7529:6337cde9:c5ec:a41e:43f5 2a02:27e8:0001:0d9ba261:8690:71e7:5390 2a02:27e8:206a;b1006c0f:779e:b664:be1f 2a02:27e8;11e6;b7a0383:bf1a:0100:70ef 2a02:27e8:2001;b2c0f520b:1811:27d1:1305 2a02:27e8:afae:ebb16bd4:f963:f15d:1c3c 2a02;27e8;2e69;7f4;654d1:684d:54f0;7420 2a02;27e8:1001:1143;11b:3076:b126:d346 2a02;27e8;2cc4;801;363;e0cd:7d58:5567 2a02:27e8:2f7a:d634cd484:6bd5:e0b7:ad76 2a02:27e8:42d9:2c0a8c83:da8b:80d7:33fd 2a02:27e8:2001:fc8fa7528:c462:737a:ee7b 2a02:27e8:cd2b:31ea88c7:b4c4:4b3f:e789 2a02:27e8:1188:4eb14251:4dc7:3ca6:1095 2/a02:27e8:112c:c14d:87c0:69a8:6f96:bc63 2a02;27e8;1097:c9a67e7a:4671:2948:102a 2a02:27e8:2711:b6707df6:13e8:d140:c719 2a02:27e8:2837:1a1ce83fb:02ec:f43a:8b22 2a02:27e8:0001:e8eadc7b:122e:9c58:dec6 2a02:27e8/f2df;0721546f:5530:732d:3a8e 2a02:27e8/a6ea:b0d97767:3f82:14f2:b61f 2a02:27e8:29d9:77b2af55:54da:1871:d57f 2a02:27e8:2a9b:ca750553:41cc:84ab:5e73 2a02:27e8:15a0/7ad94790:f168:a589:57b7 2a02:27e8:1039:c56a415c:eeac:8cdd:c4f8 2a02:27e8:bc50:333thfcb5:5c07:69ea:670c 2a02:27e8:1c5c:833ema38:825c:9400:9ab3 2a02:27e8:2f99:043f 67db:a548:0320:8028 2a02:27e8:1001:be797bdf:eb4d:9de4:921d

Thanks!

Richard Barnes <rbarnes@bbn.com> Rick Altmann Daniel Kerr