Advanced Computer Networking (ACN)

IN2097 – WiSe 2023–2024

Prof. Dr.-Ing. Georg Carle

Sebastian Gallenmüller, Max Helm, Benedikt Jaeger,
Marcel Kempf, Patrick Sattler, Johannes Zirngibl

Chair of Network Architectures and Services
School of Computation, Information, and Technology
Technical University of Munich

Internet Protocol v4

Network layer

Internet addressing

ICMP

ARP

Internet-wide Measurements

Bibliography

Network layer

Internet addressing

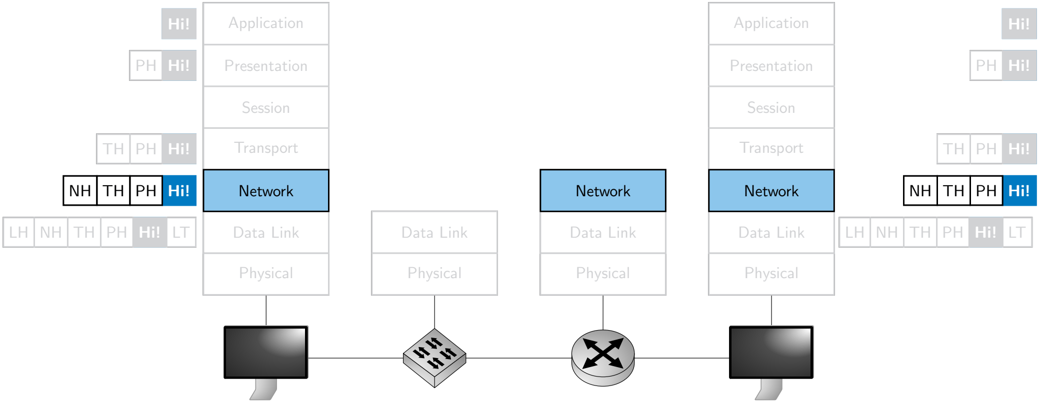
ICMP

ARP

Internet-wide Measurements

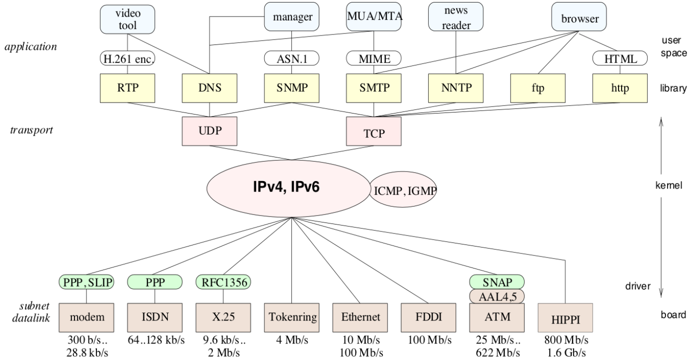
Bibliography

Network layer
Protocol entities in hosts and routers



Network layer
IP protocol model

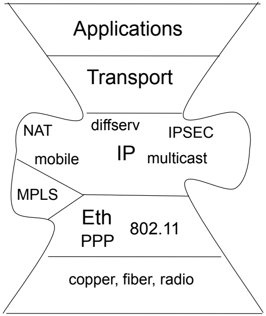
- Many application specific protocols over IP
- IP (with best effort service model) over many media specific LAN protocols
  - “Hourglass” model of IP
  - QoS support (RSVP, DiffServ) added to IP as an “afterthought”



Network layer
IP protocol stack evolution

- Specific requirements and use cases did lead to the development of additional protocols
  - protocol implementation added to specific nodes
  - additional protocols not supported everywhere

- TLS, DTLS
- TCP, UDP, SCTP, DCCP
- BGP, OSPF, IS-IS, RIP, RIPng, VRRP, PIM, IGMP, MLD
- IPsec, IKE, EAP
- IPv4, IPv6, ICMP
- VLAN, GTP, IP in IP, GRE, L2TP, MPLS



Network layer
Network prefix and host number

- L3/IP service goal: forward IP datagrams to destination IP subnet/host/interface
- IP address has role of locator & identifier:
  - network part (network identifier & locator)
  - host part (host identifier)

- Each IP network (often called subnetwork or subnet) has an IP address:
  - IP address of a network = Host number is set to all zeros, e.g., 128.143.0.0

- IP routers are devices that forward IP datagrams between IP networks

- Delivery of an IP datagram proceeds in 2 steps:
  1. Use network prefix to deliver IP datagram to the right network
  2. Once the network is reached, use the host L3 address to deliver to the right interface

Network layer
Host or router network layer functions

- IP protocol
  - addressing conventions
  - datagram format
  - packet handling conventions
- ICMP protocol
  - error reporting
  - router “signaling”
- Routing protocols
  - path selection
  - RIP, OSPF, BGP

Network layer
IPv4 datagram [1]

<table>
<tr><th>Offset</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td>0 B</td><td colspan="4">Version</td><td colspan="4">IHL</td><td colspan="8">TOS</td><td colspan="16">Total Length</td></tr>
<tr><td>4 B</td><td colspan="16">Identification</td><td colspan="3">Flags</td><td colspan="13">Fragment Offset</td></tr>
<tr><td>8 B</td><td colspan="8">TTL</td><td colspan="8">Protocol</td><td colspan="16">Header Checksum</td></tr>
<tr><td>12 B</td><td colspan="32">Source Address</td></tr>
<tr><td>16 B</td><td colspan="32">Destination Address</td></tr>
<tr><td>20 B</td><td colspan="32">Options / Padding (optional)</td></tr>
</table>

Abbreviations:

- IHL: Internet Header Length
- TOS: Type Of Service
- TTL: Time To Live

Internet Protocol v4

Network layer

Internet addressing

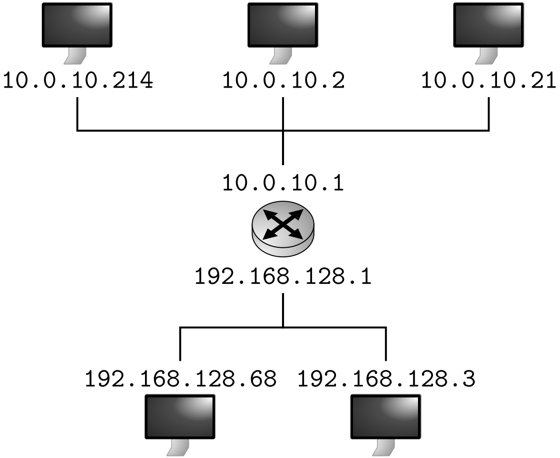
ICMP

ARP

Internet-wide Measurements

Bibliography

Internet addressing
IPv4 addressing

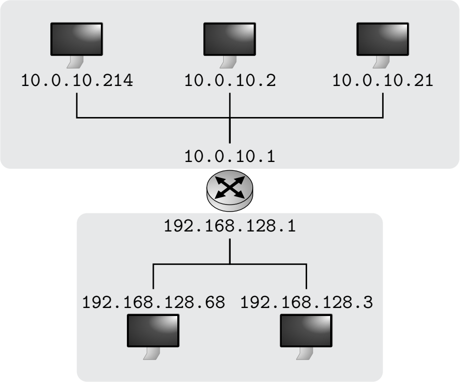


- IP address: 32-bit identifier for host, router interface
- Address space: 4.3 billion IPv4 addresses in theory
- Interface: connection between host/router and physical link
    - IP addresses associated with each interface

IP address notation: 4 numbers from 0 to 255 separated by dots

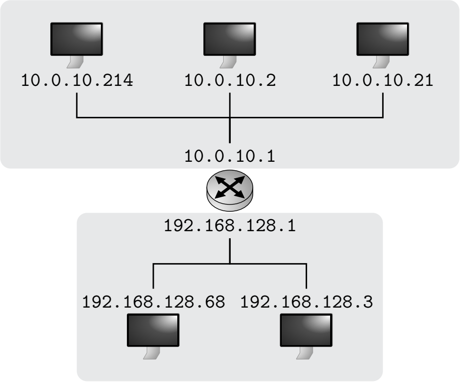
Internet addressing
Subnets

- What is a subnet?
  - device interfaces with same subnet part of IP address
  - can physically reach each other without intervening router



network with 2 subnets

Internet addressing
Subnets

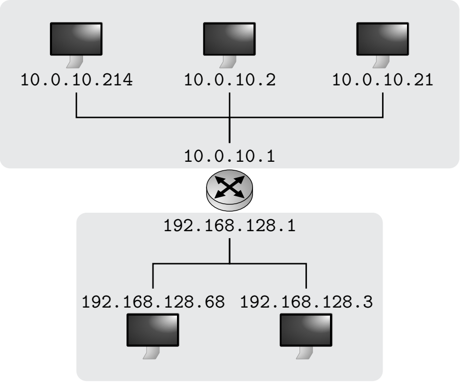


network with 2 subnets

- What is a subnet?
  - device interfaces with same subnet part of IP address
  - can physically reach each other without intervening router
- How to determine?
  - Detach interfaces from host or router

Internet addressing
Subnets

- What is a subnet?
  - device interfaces with same subnet part of IP address
  - can physically reach each other without intervening router
- How to determine?
  - Detach interfaces from host or router
- Splitting IP addresses
  - **Network part**: Addressing the network
  - **Host part**: Addressing the interface of a host

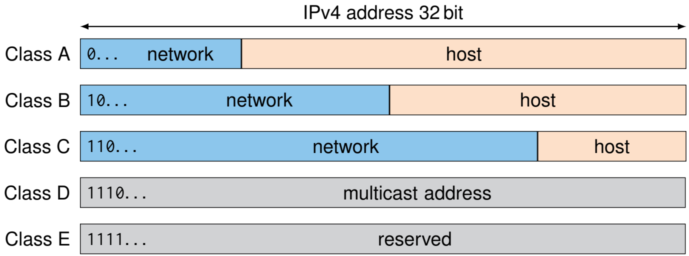


network with 2 subnets

$$
\begin{array}{ccc}
 & 192_{10}.\ 168_{10}.\ & 128_{10}.\ 1_{10} \\
= & 11000000_2.10101000_2.1_2 & 0000000_2.00000001_2 \\
 & \text{network part} & \text{host part}
\end{array}
$$

Internet addressing
Classful IP addresses (historic)

- Used from 1981 to 1993



class	start address		end address	relative portion
A	0.0.0.0	to	127.255.255.255	50.00 %
B	128.0.0.0	to	191.255.255.255	25.00 %
C	192.0.0.0	to	223.255.255.255	12.50 %
D	224.0.0.0	to	239.255.255.255	6.25 %
E	240.0.0.0	to	255.255.255.255	6.25 %

Internet addressing
Classless IP addresses

- CIDR: Classless InterDomain Routing (introduced in 1993, RFC 1519)
- Idea: intoduction of arbitrary subnet length
- Address format: `a.b.c.d/x`, where `x` corresponds to the number of bits in subnet portion of address

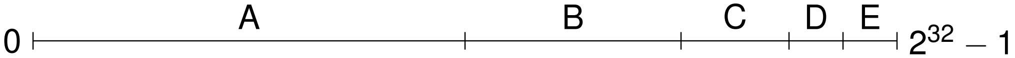
Internet addressing
Example: CIDR

- How to calculate the network address for interface address 192.168.128.1 with a prefix length of 17 bits?
  - CIDR notation: 192.168.128.1/17
  - Dotted decimal notation: 192.168.128.1/255.255.128.0

host address$_{10}$		192_{10}. 168_{10}. 128_{10}. 1_{10}
host address$_2$		$11000000_2.10101000_2.10000000_2.00000001_2$
network mask	&	$11111111_2.11111111_2.10000000_2.00000000_2$
network address$_2$		$11000000_2.10101000_2.10000000_2.00000000_2$
network address$_{10}$		192_{10}. 168_{10}. 128_{10}. 0_{10}

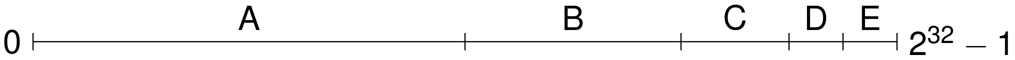
Internet addressing
Classful vs. CIDR

- Classful:

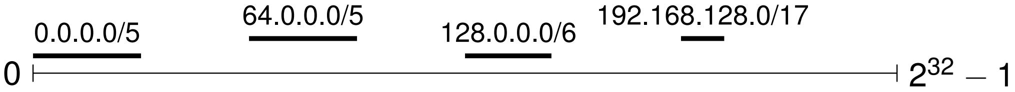


Internet addressing
Classful vs. CIDR

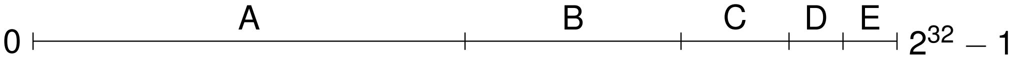
- Classful:



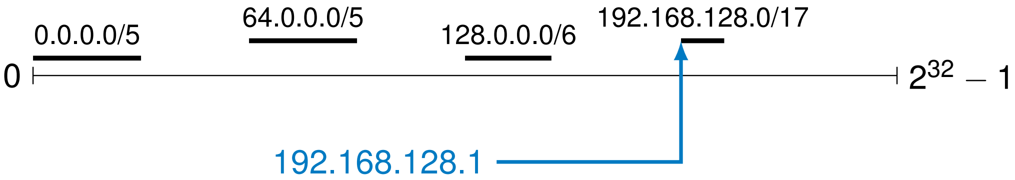
- CIDR:



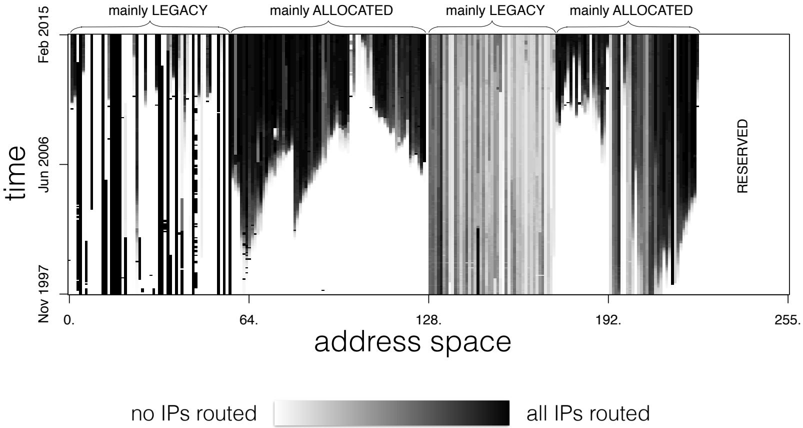
- Classful:



- CIDR:



Internet addressing
Current usage of IPv4 addresses[1]



[1] P. Richter, M. Allman, R. Bush, et al., “A primer on ipv4 scarcity,” ACM SIGCOMM Computer Communication Review, vol. 45, no. 2, pp. 21–31, 2015

Internet addressing
IPv4 address exhaustion

IPv4 addresses are a rare resource nowadays

- Inefficiency of Classful Internet Routing:
  - Class C (256 addresses) too small for small enterprises
  - Class B (65536 addresses) too small for large enterprises or universities
  - Class A (16 million addresses) too large
- Rise of Internet-connected devices: personal computers, mobile phones, Internet-of-Things, . . .
- Always-on devices: Sharing of IPv4 addresses become less viable

Various solutions proposed, the most notable one being private addresses:

- 10.0.0.0/8 - 24-bit block
- 172.16.0.0/12 - 20-bit block
- 192.168.0.0/16 - 16-bit block

Internet Protocol v4

Network layer

Internet addressing

ICMP

ARP

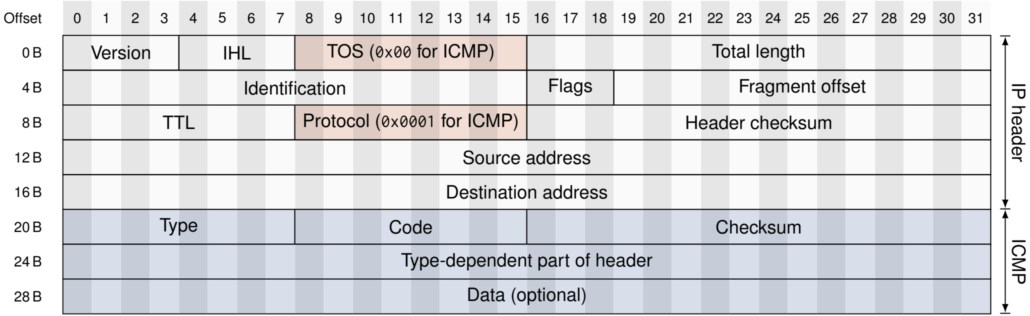
Internet-wide Measurements

Bibliography

ICMP
Internet Control Message Protocol (ICMP)

- RFC 792 [3]
- Network control plane protocol “above” IP:
  - ICMP messages carried in IP datagrams
  - Can be considered part of the IP layer
- Communicates error messages and other conditions that require attention
- Error messages are acted on by either. . .
  - IP layer, or
  - TCP, or UDP
- Some ICMP messages cause error notifications to be returned to user processes

ICMP
ICMP message format



- 15 different types
- Some types use a code to further specify the condition

ICMP message types

Two classes of ICMP messages:

- Query messages
  - Only kind of ICMP messages that generate another ICMP message
- Error messages
  - Contain IP header and at least first 8 bytes of datagram that caused the ICMP error to be generated.
  - Allows receiving ICMP module to associate the message with a particular protocol and process (port number)

ICMP
ICMP message types

- Cf. RFC 792 [3]

type	description
0	echo reply (ping)
3	destination unreachable (codes subsequent slide)
4	source quench (deprecated, RFC 6633)
5	redirect
8	echo request
9	router advertisement (MC, see RFC 1256)
10	router solicitation (MC, see RFC 1256)
11	time exceeded
12	parameter problem (bad IP header)
13	timestamp request
14	timestamp reply
15	info request
16	info reply
17	address mask request (see RFC 950)
18	address mask reply

ICMP
ICMP message types continued

- Cf. RFC 792 [3]

type	code	description
3	0	dest network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown

- Historically: ICMP content always contained IP header and first 8 bytes of IP payload that caused ICMP error message to be generated (RFC 792)
- Today: ICMP should contain as much data of the dropped message as possible up to a limit of 576 byte for the ICMP message (RFC 1812)

Internet Protocol v4

Network layer

Internet addressing

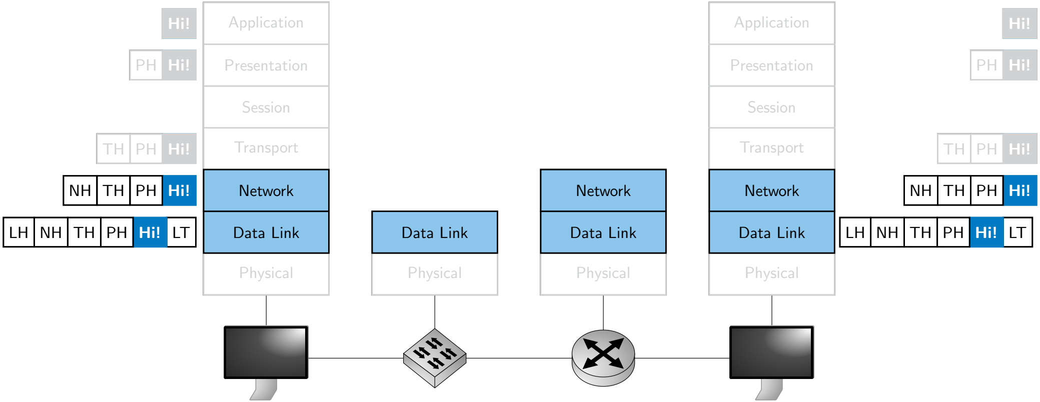
ICMP

ARP

Internet-wide Measurements

Bibliography

ARP
Connecting Link and Network Layer



ARP
MAC Addresses and IP Addresses

MAC (or LAN or physical or Ethernet) address

- L2 service: transmit frame from one interface to another physically-connected interface (same network) with specified destination address
- address length: 48 bit (for most LANs)
  - burned into network adapter ROM, or software settable
  - assumption: two hosts on the same LAN will not use the same Ethernet address

IP address: network-layer address

- L3 service: get datagram to destination IP subnet / host I/F
- L3 address: has role of locator & identifier (vs. HIP – Host Identity Protocol; LISP – Locator/ID Separator Protocol)
- address length: 32 bit (IPv4) or 128 bit (IPv6)
- address separated into:
  - network part (i.e. network identifier & locator)
  - host part (i.e. host identifier)

ARP
Address resolution

Mapping between addresses of different layers

- Examples:
  - IPv4 ⟶ MAC
  - MAC ⟶ IPv4

Mapping from L3 host address to MAC address

- Needed to identify correct L2 adapter of L3 address

→ Address Resolution Protocol (ARP)

Mapping from MAC address to L3 address

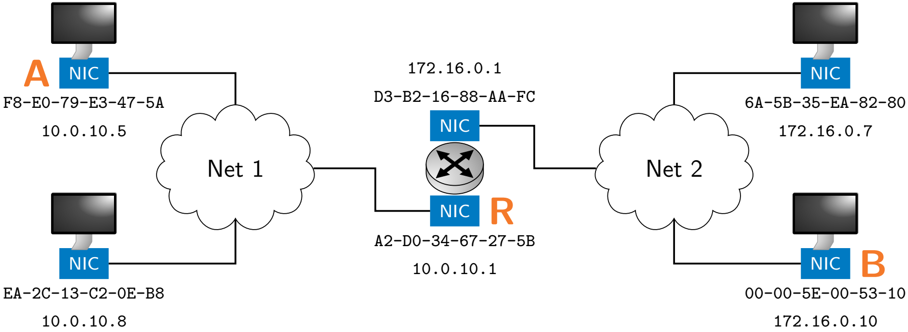
→ Reverse Address Resolution Protocol (RARP) (rarely used)

ARP
Addresses and names

	Example	Organization
MAC address	`6C:40:08:BD:A5:B4`	flat, permanent
IP address	`172.16.0.1`	topological (mostly)
Host name	`www.ietf.org`	hierarchical

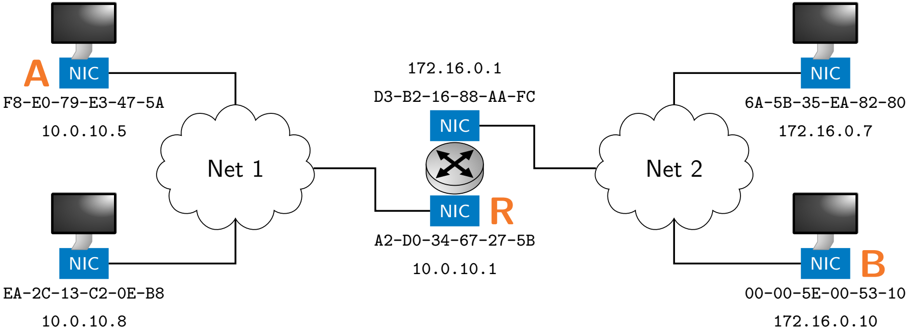
ARP
Addressing: routing to another LAN

- Example send datagram from A to B via R (assuming A knows B's IP address)
- The router manages two ARP tables one for Net 1 and one for Net 2



ARP

- A creates IP datagram with source IP addr. A, destination IP addr. B
- A uses ARP to get R's MAC address of R's interface 10.0.10.1
- A creates link-layer frame with R's MAC address as destination, frame contains A-to-B IP datagram
- A's NIC sends frame
- R's NIC receives frame
- R extracts IP datagram from Ethernet frame, sees it is destined to B
- R uses ARP to get B's MAC address
- R creates frame containing A-to-B IP datagram, sends it to B



ARP protocol: same LAN (network)

- A wants to send datagram to R's interface 10.0.10.1, while R's MAC address is not in A's ARP table.
- A broadcasts ARP query packet, containing R's IP address
  - destination MAC address = FF-FF-FF-FF-FF-FF
  - all hosts on LAN receive ARP query
- When R receives ARP packet, it replies to A with its (R's) MAC address
  - frame sent to A's MAC address
- A caches IP-to-MAC address pair in its ARP table until information times out
  - soft state: information that times out (goes away) unless refreshed
- ARP is “plug-and-play”:
  - nodes create their ARP tables without intervention from network administrator

ARP
ARP packet format

<table>
<tr><th>Offset</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td>0 B</td><td colspan="16">Hardware Type</td><td colspan="16">Protocol Type</td></tr>
<tr><td>4 B</td><td colspan="8">Hardware Addr. Length</td><td colspan="8">Protocol Addr. Length</td><td colspan="16">Operation</td></tr>
<tr><td>8 B</td><td colspan="32">Sender Hardware Address (first 32 bit)</td></tr>
<tr><td>12 B</td><td colspan="16">Sender Hardware Address (last 16 bit)</td><td colspan="16">Sender Protocol Address (first 16 bit)</td></tr>
<tr><td>16 B</td><td colspan="16">Sender Protocol Address (last 16 bit)</td><td colspan="16">Target Hardware Address (first 16 bit)</td></tr>
<tr><td>20 B</td><td colspan="32">Target Hardware Address (last 32 bit)</td></tr>
<tr><td>24 B</td><td colspan="32">Target Protocol Address</td></tr>
</table>

ARP details

ARP supports different protocols at L2 and L3

- any network protocol over any LAN/MAC protocol
- type and address length fields specified in ARP PDUs

Reverse ARP (RARP) cf. RFC 903 (rarely used)

L2 MAC fields (hardware)

- hardware type: 6 = IEEE802 (with LLC/SNAP)
- address length: 6 for a 6 byte long MAC address
- sender hardware address (SHA)
- target hardware address (THA)

L3 network fields (protocol)

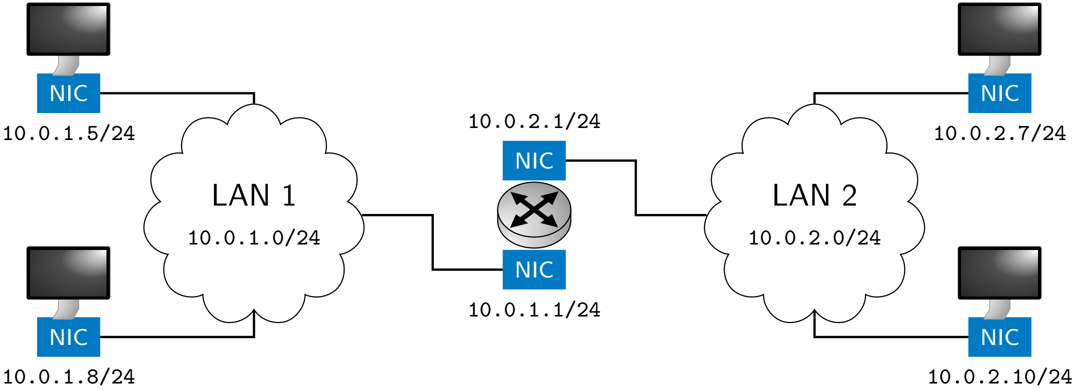
- protocol type: IP = 0800
- address length: 4 for 4 byte long IPv4 address
- sender protocol address (SPA)
- target protocol address (TPA)

Operation Code

- 01: request
- 02: reply
- 03: reverse request
- 04: reverse reply (for RARP)
- cf. http://www.iana.org/assignments/arp-parameters

ARP
Proxy ARP

- Proxy ARP: Host or router responds to ARP Request that arrives from one of its connected networks for a host that is on another of its connected networks.
- RFC 925: Multi-LAN Address Resolution



ARP
Proxy ARP - possible uses

Transparent subnet gatewaying

- Two LANs sharing same IP subnet, connected via router
- cf. RFC 1027 – Using ARP to Implement Transparent Subnet Gateways

Host joining LAN via dialup link

- Dialup router employs Proxy ARP

Host joining LAN via VPN

- VPN server employs Proxy ARP

Host separated via firewall

- Firewall employs Proxy ARP

ARP
ARP optimizations

When should a host send ARP requests?

- Before sending each IP packet?
  - No, each host/router maintains ARP table (IP address ⟶ MAC address mapping)
  - ARP request is only sent in case there is no entry for this IP address in the ARP table.

How to deal with hosts that change their addresses?

- Expiration timer is associated to each entry in the ARP table
  - ARP table entry is removed upon timer expiration
  - Some implementations send ARP request to revalidate before removing table entry
  - Some implementations remember when ARP table entries were used to avoid removing important entries

ARP
Things to know about ARP

What happens if an ARP request is made for a non-existing host?

- Several ARP requests are made with increasing time intervals between requests
- Eventually, ARP gives up

Gratuitous ARP Requests

- A host sends an ARP request for its own IP address
- Useful for detecting if an IP address has already been assigned.

ARP
Vulnerabilities of ARP

1. Since ARP does not authenticate requests or replies, ARP requests and replies can be forged
2. ARP is stateless: ARP replies can be sent without a corresponding ARP request
3. According to the ARP protocol specification, a node receiving an ARP packet (request or reply) must update its local ARP cache with the information in the sender fields. Updates also happen if the receiving node already has an entry for the IP address of the sender in its ARP cache. (This applies for ARP Request packets and for ARP Reply packets)

Typical exploitation of these vulnerabilities:

- A forged ARP request or reply can be used to update the ARP cache of a remote system with a forged entry (ARP Poisoning)
- This can be used to redirect IP traffic from/to other hosts
- ARP poisoning & ARP spoofing also can be performed by hosts within a WPA2-protected WLAN

Internet Protocol v4

Network layer

Internet addressing

ICMP

ARP

Internet-wide Measurements

Bibliography

Why do we measure the network?

Do we really have to?

- The network is well engineered
- Well documented protocols, mechanisms, . . .
- Everything built by humans
  - → No unknowns (compare this to physics)
- In theory, we can know everything that is going on

→ No need for measurements?!

Why do we measure the network?

Do we really have to?

- The network is well engineered
- Well documented protocols, mechanisms, . . .
- Everything built by humans
  - → No unknowns (compare this to physics)
- In theory, we can know everything that is going on

→ No need for measurements?!

But:

- Distributed multi-domain network
  - → Information only partially available
- Moving target
  - Requirements change
  - Growth, usage, structure changes
- Highly interactive system
- Heterogeneity in all directions
- The total is more than the sum of its pieces
- Built, driven, and used by humans
  - → Errors, misconfigurations, flaws, failures, misuse, . . .

Why do we measure the network?

Do we really have to?

- The network is well engineered
- Well documented protocols, mechanisms, ...
- Everything built by humans
  - → No unknowns (compare this to physics)
- In theory, we can know everything that is going on

→ No need for measurements?!

But:

- Distributed multi-domain network
  - → Information only partially available
- Moving target
  - Requirements change
  - Growth, usage, structure changes
- Highly interactive system
- Heterogeneity in all directions
- The total is more than the sum of its pieces
- Built, driven, and used by humans
  - → Errors, misconfigurations, flaws, failures, misuse, ...

Active network measurements are an important research area to understand the Internet and interactions between all its components.

Why do we measure the network?

Network provider view

- Manage traffic
  - Model reality
  - Predict future
  - Plan network
  - Avoid bottlenecks in advance
- Reduce cost
- Accounting

Why do we measure the network?

Network provider view

- Manage traffic
  - Model reality
  - Predict future
  - Plan network
  - Avoid bottlenecks in advance
- Reduce cost
- Accounting

Service provider view

- Get information about clients
- Adjust service to demands
- Reduce load on servers
- Accounting

Why do we measure the network?

Network provider view

- Manage traffic
  - Model reality
  - Predict future
  - Plan network
  - Avoid bottlenecks in advance
- Reduce cost
- Accounting

Service provider view

- Get information about clients
- Adjust service to demands
- Reduce load on servers
- Accounting

Client view

- Get the best possible service
- *Do I get what I paid for?*

Why do we measure the network?

Network provider view

- Manage traffic
  - Model reality
  - Predict future
  - Plan network
  - Avoid bottlenecks in advance
- Reduce cost
- Accounting

Service provider view

- Get information about clients
- Adjust service to demands
- Reduce load on servers
- Accounting

Client view

- Get the best possible service
- *Do I get what I paid for?*

Security view

- Detect malicious traffic
- Detect malicious hosts
- Detect malicious networks

Why do we measure the network?

Network provider view

- Manage traffic
  - Model reality
  - Predict future
  - Plan network
  - Avoid bottlenecks in advance
- Reduce cost
- Accounting

Service provider view

- Get information about clients
- Adjust service to demands
- Reduce load on servers
- Accounting

Client view

- Get the best possible service
- *Do I get what I paid for?*

Security view

- Detect malicious traffic
- Detect malicious hosts
- Detect malicious networks

Researcher view

- Understand the Internet better
- *Could our new routing algorithm handle all this real-world traffic?*
- ...

Active Internet-wide measurements effect the network, users and providers!

Problems:

- Creates additional traffic
- Creates load on routers and hosts
- Might uncover personal information
- Might be intrusive

Considerations:

- Scan with a moderate rate
- Distribute the load as good as possible
- Do not publish data without anonymization or limited access
- Inform about the scanning behavior and react to complaints

Info page at our chair[2]

Chair of Network Architectures and Services
TUM Department of Informatics
Technical University of Munich

Homepage
Members
Research
Publications
Projects
Interest Groups
Past Projects
Cooperation Partners
Talks
Teaching
Theses
Job Offers
Contact
Information for Students

Projects › GINO ›

Internet-wide scans

Why am I receiving traffic from the Technical University of Munich?

We conduct various regular and ad-hoc Internet-wide scans for protocols such as HTTPS, DNS, and BACnet. These are purely scientific and we never attempt to intrude into any system. We follow best practices laid out by the scientific community such as by Dittrich et al. [1], and Partridge and Allman [2].

We never attempt to abuse security vulnerabilities on your system, guess for passwords or upload files to your systems.

We may collect information about your service as far as they are publicly visible to everybody else on the Internet.

We hope that the data, we are collecting, helps us to understand the Internet better. We are an academic institution and will try to publish all our findings to a wider audience. However, we will never publish parts of our dataset which clearly identifies you or your company. More information about scans and publications can be found on the webpage of the Global INternet Observatory.

I do not want to be part of the research. How can I opt out?

You can just block connection attempts from our scanning systems or send us an E-Mail and we will add you and your network to our blacklist.

Host	IPv6 address	IPv4 address
planetlabX.net.in.tum.de	2001:4ca0:108:42::X	138.246.253.X
dallas	2600:3c00::f03c:91ff:fe3b:d2d	45.33.5.55
singapore	2400:8901::f03c:91ff:fe3b:d08	139.162.29.117

Contact

If you have further questions about our research, please contact us at scans@net.in.tum.de

References

1. D. Dittrich, E. Kenneally et al., “The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research,” US Department of Homeland Security, 2012. ↩
2. C. Partridge and M. Allman, “Ethical Considerations in Network Measurement Papers”, Communications of the ACM, 2016. ↩

[2] https://net.in.tum.de/projects/gino/scans.html

Ping

- Checks if host is reachable, alive
- Uses ICMP echo request/reply
- Copy packet data request reply

```
PING net.in.tum.de (131.159.15.24): 56 data bytes
64 bytes from 131.159.15.24: icmp_seq=0 ttl=63 time=4.033 ms
64 bytes from 131.159.15.24: icmp_seq=1 ttl=63 time=13.310 ms
64 bytes from 131.159.15.24: icmp_seq=2 ttl=63 time=58.955 ms
64 bytes from 131.159.15.24: icmp_seq=3 ttl=63 time=7.143 ms
^C
--- net.in.tum.de ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 4.033/20.860/58.955/22.246 ms
```

Listing 1: Sample output of ping

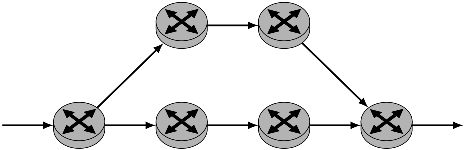
Traceroute

- Allows to follow path taken by packet
- Send UDP/TCP/. . . packets with increasing TTL to (unlikely) port
- ICMP replies: ‘time exceeded’; last ICMP message: ‘port unreachable’

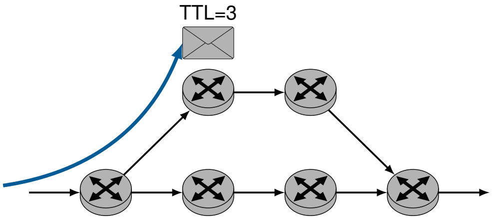
```
$ traceroute gaia.cs.umass.edu
1  scylla (131.159.20.11)  4.263 ms  2.531 ms  2.162 ms
2  nz-bb-net.informatik.tu-muenchen.de (131.159.252.149)  6.124 ms  15.174 ms  3.546 ms
3  nz-csr1-kw5-bb1.informatik.tu-muenchen.de (131.159.252.2)  2.925 ms  4.234 ms  3.033 ms
4  vl-3010.csr1-2wr.lrz.de (129.187.0.149)  5.082 ms  3.387 ms  4.694 ms
5  cr-gar1-be2-147.x-win.dfn.de (188.1.37.89)  3.254 ms  3.274 ms  2.967 ms
6  cr-fra2-hundredgige0-0-0-3.x-win.dfn.de (188.1.144.253)  13.139 ms  12.260 ms  15.702 ms
7  dfn.mx1.fra.de.geant.net (62.40.124.217)  11.365 ms  11.716 ms  16.314 ms
8  ae1.mx1.gen.ch.geant.net (62.40.98.108)  19.889 ms  26.193 ms  19.661 ms
9  ae4.mx1.par.fr.geant.net (62.40.98.152)  28.465 ms  27.664 ms  29.365 ms
10 et-3-1-0.102.rtsw.newy32aoa.net.internet2.edu (198.71.45.236)  104.199 ms  104.173 ms  109.925 ms
11 nox300gw1-i2-re.nox.org (192.5.89.221)  111.437 ms  110.232 ms  109.370 ms
12 umass-re-nox300gw1.nox.org (192.5.89.102)  113.755 ms  115.848 ms  110.634 ms
13 core1-rt-xe-0-0-0.gw.umass.edu (192.80.83.101)  118.469 ms  119.070 ms  114.279 ms
14 lgrc-rt-106-8-po-10.gw.umass.edu (128.119.0.233)  111.948 ms  111.992 ms  111.616 ms
15 128.119.3.32 (128.119.3.32)  112.194 ms  124.315 ms  111.624 ms
16 nscs1bbs1.cs.umass.edu (128.119.240.253)  114.384 ms  166.509 ms  113.220 ms
17 gaia.cs.umass.edu (128.119.245.12)  130.574 ms !Z  114.883 ms !Z  116.865 ms !Z
```

Listing 2: Sample output of traceroute

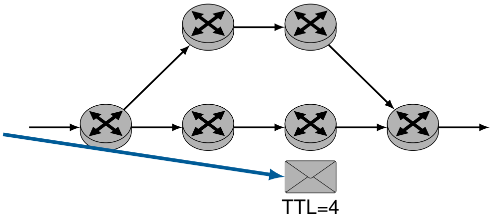
Traceroute: possible anomalies due to load balancing



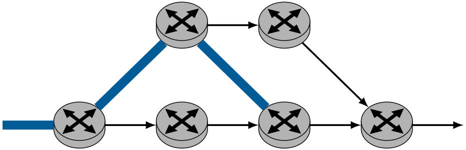
Traceroute: possible anomalies due to load balancing



Traceroute: possible anomalies due to load balancing



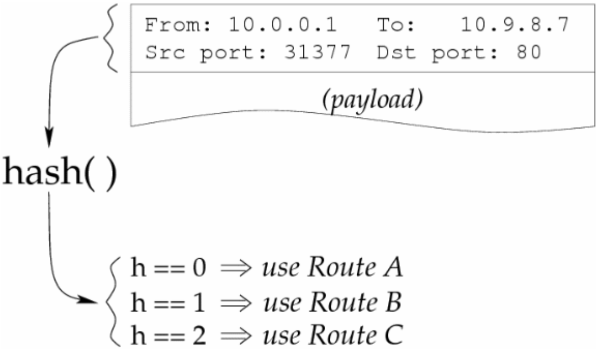
Traceroute: possible anomalies due to load balancing



Load balancing

Per Connection Load balancing:

- Hash *consistently* and use packet headers as *random* values
  - Packets from same TCP connection yield same hash value
  - No reordering within one TCP connection



Paris Traceoute

Idea: Vary header fields that are within the first 28 octets

- TCP: sequence number
- UDP: checksum field
  - Requires manipulation of payload to ensure correctness of checksum
- ICMP: combination of ICMP identifier and sequence number

Experiment results

- Certain routers use first four octets after IP header combined with IP fields for load balancing

Still fails on per packet load balancing

- MDA [4] tries to cover this problem

Further Traceroutes

There are further interesting traceroute tools, e.g.:

- yarrp [5]
  - Stateless
  - Highly parallel
- Scamper [6]
  - All-in-one tool
  - IPv4 & IPv6
  - Built-in alias resolution
- MDA [4]
  - Tries to identify all possible paths
  - Crafts specific packets to find new paths
  - Large overhead
- MDA-Lite [7]
  - Optimized MDA implementation
  - Trade off between performance and completeness

Tools
Nmap

Open-source network mapping tool

- https://nmap.org/
- First version in 1997

Modes of operation:

- Host discovery
- Service detection
- OS detection
- Execution of custom scripts

Nmap - Scanning Techniques

- TCP RAW socket scans with certain flags
  - SYN: Find open ports
  - NULL/FIN/Xmas:
    - According to RFC 793 all packets without SYN, ACK, RST result in RST if port is closed, and no response if port is open
    - NULL: No bit set
    - FIN: Only FIN set
    - Xmas: FIN+PUSH+URG
  - ACK: Determine filtered/unfiltered ports in a firewall
  - Window: Same as ACK, lists responses with Window > 0 in RST as open (implementation on certain firewalls)
  - Maimon: Send FIN+ACK, according to RFC 793 all hosts should respond with RST, no matter if port is open or closed
- TCP connect scans
- ICMP ping scan
- UDP payload scan

Nmap - Performance

Internet-wide scans using Nmap:

- Stateful scanning approach
  - Nmap keeps state for every packet in transit
  - Catch timeouts and send retry packets
- Performance
  - Full scan from one system takes 10 days (4k IP addr/sec) [8]
  - 25 Amazon EC2 instances $\rightarrow$ 25 hours (1.6k IP addr/sec) [9]
  - Typically 1 packet sent and 1 packet received per IP addr

ZMap

Adaptation of Nmap for Internet-wide scans

- https://zmap.io/
- Developed at the University of Michigan [10]
- First port-scanner to saturate 1 Gbit/s link: 1.4 Mpps
- Scan entire Internet in 45 minutes
- Later tweaked to saturate 10 Gbit/s link [11]: 14 Mpps

ZMap

Internet-wide scans

- Use TCP SYN or UDP payload scan to find open ports
- Input randomization
  - Pseudo-random number generator
  - Based on multiplicative group of integers modulo p ($2^{32} + 15$)
  - Map 32-bit integer to IPv4 address
- Possible to use multiple worker nodes (shards) on different machines
  - IP will only be scanned once in complete scan

ZMap - Approach

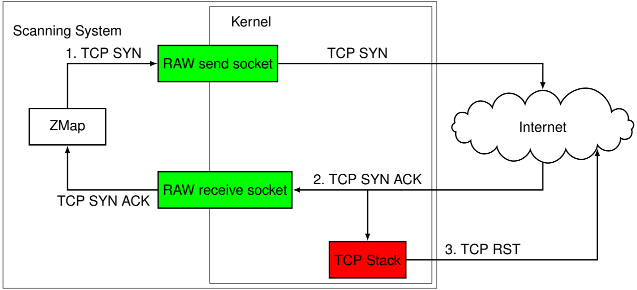
Stateless scanning

- No state for sent packets kept
- Timeout detection not possible
- How to identify responses belonging to scan?
  - Use IP ID = 54321
  - Generate validation based on packet input (e.g. destination IP) using AES
  - Store validation in packet which will be sent (e.g. in sequence number)
  - Validate validation (e.g. sequence number – 1) in received packet

ZMap - Approach

Separate send and receive threads using RAW sockets

- Use RAW socket to directly send and receive packets without kernel TCP stack
- No locking needed
- ZMap send and receive behavior:



ZMap - Approach

Separate probe and output modules

- Probe modules
  - Implement scanning technique
  - E.g. TCP SYN, TCP SYN-ACK, UDP payload
- Output modules
  - Implement processing and output of received responses
  - E.g. IP address only, CSV, database

ZMap - Additional Tools

ZMap is the basis of a large set of additional tools[3]:

- ZGrab
  - Stateful application-layer scanner
  - e.g. for HTTPS, SSH, BACNET
- ZDNS
  - utility for fast DNS lookups
- ZCrypto
  - TLS and X.509 library
  - Certificate parsing and TLS handshake transcription

3 https://zmap.io/

IPv4 ZMap Scans

State of the art:

- Full "0/0" scans

IPv4 ZMap Scans

State of the art:

- Full "0/0" scans
- Out of 4 B addresses only ~ 3.2 B are publicly reachable
  - Excludes private, reserved or announced addresses

IPv4 ZMap Scans

State of the art:

- Full "0/0" scans
- Out of 4 B addresses only ~ 3.2 B are publicly reachable
  - Excludes private, reserved or announced addresses
- Feasible with Nmap/ZMap
  - ZMap scan rate: 20k IP addr/s $\rightarrow$ 37h

IPv4 ZMap Scans

State of the art:

- Full "0/0" scans
- Out of 4 B addresses only ~ 3.2 B are publicly reachable
  - Excludes private, reserved or announced addresses
- Feasible with Nmap/ZMap
  - ZMap scan rate: 20k IP addr/s → 37h
- ZMap only provides information whether the address is responsive
  - e.g., an ICMP Ping is possible or a TCP Handshake

→ No information whether an actual service is available
  - Protocol-specific scanners for stateful protocols are required
- Continuous scans to observe changes in the network and deployment

IPv4 ZMap Scans

TCP Port Scan results:

- Conducted from a single vantage point
- First week of August 2022

Service	Port	Responsive
HTTP	80	63 185 323
HTTPS	443	55 797 463
CPE WAN Management	7547	43 118 258
SSH	22	25 612 566
SMTP	25	15 298 930
FTP	21	12 695 736
Alternative HTTP	8080	11 828 087
DNS	53	10 215 627
RDP	3389	8 135 255
Ephemeral Port	60000	7 332 835

IPv4 ZMap Scans

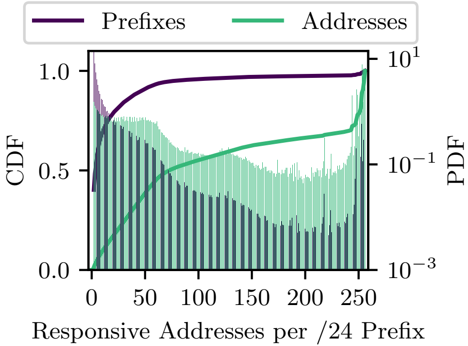
Distribution accross the Internet

- Based on /24 prefixes
- The smallest prefix routed on the Internet (within BGP)

Distribution accross the Internet

- Based on /24 prefixes
- The smallest prefix routed on the Internet (within BGP)

Port 443:

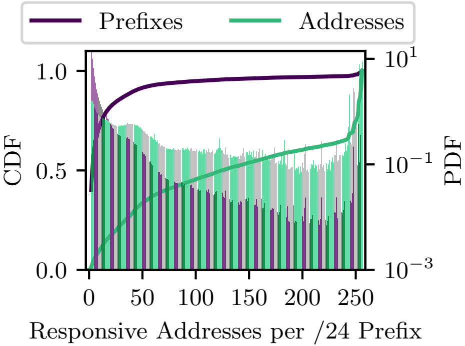


IPv4 ZMap Scans

Distribution accross the Internet

- Based on /24 prefixes
- The smallest prefix routed on the Internet (within BGP)

Port 80:

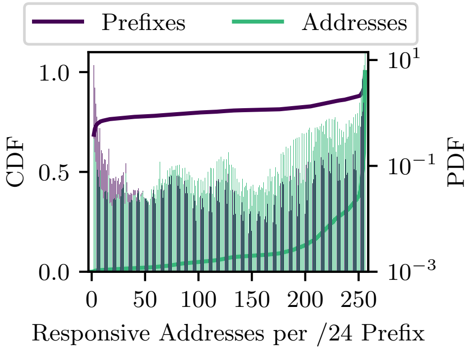


IPv4 ZMap Scans

Distribution accross the Internet

- Based on /24 prefixes
- The smallest prefix routed on the Internet (within BGP)

Port 60000:



Why are more than 90% of addresses responsive for some /24 prefixes?

- In some cases all addresses are used by individual servers.
- But other reasons can potentially be:

IPv4 ZMap Scans

Why are more than 90% of addresses responsive for some /24 prefixes?

- In some cases all addresses are used by individual servers.
- But other reasons can potentially be:
- Tarpits
  - Each address is responsive to slow down scanners

Why are more than 90% of addresses responsive for some /24 prefixes?

- In some cases all addresses are used by individual servers.
- But other reasons can potentially be:
- Tarpits
    - Each address is responsive to slow down scanners
- Proxies/Middleboxes
    - Devices terminate TCP handshakes for all addresses
    - Decide whether to drop or where to route traffic depending on higher layer services

Why are more than 90% of addresses responsive for some /24 prefixes?

- In some cases all addresses are used by individual servers.
- But other reasons can potentially be:
- Tarpits
  - Each address is responsive to slow down scanners
- Proxies/Middleboxes
  - Devices terminate TCP handshakes for all addresses
  - Decide whether to drop or where to route traffic depending on higher layer services
- CDNs, e.g., Cloudflare's addressing agility approach [12]
  - This technique decouples IP addresses from domain names and services.
  - The authoritative name server can select the addresses in the query response from a full prefix.
  - Used for on-demand, flexible load balancing.

Internet Protocol v4

Network layer

Internet addressing

ICMP

ARP

Internet-wide Measurements

Bibliography

[1] DARPA, Internet Protocol, https://tools.ietf.org/html/rfc791, 1981.

[2] P. Richter, M. Allman, R. Bush, and V. Paxson, “A primer on ipv4 scarcity,” ACM SIGCOMM Computer Communication Review, vol. 45, no. 2, pp. 21–31, 2015.

[3] J. Postel, Internet Control Message Protocol, https://tools.ietf.org/html/rfc792, 1981.

[4] D. Veitch, B. Augustin, R. Teixeira, and T. Friedman, “Failure control in multipath route tracing,” in IEEE INFOCOM 2009, 2009, pp. 1395–1403. DOI: 10.1109/INFCOM.2009.5062055.

[5] R. Beverly, “Yarrp'ing the internet: Randomized high-speed active topology discovery,” in Proceedings of the 2016 Internet Measurement Conference, ser. IMC '16, New York, NY, USA: Association for Computing Machinery, 2016, 413–420, ISBN: 9781450345262. [Online]. Available: https://doi.org/10.1145/2987443.2987479.

[6] M. Luckie, “Scamper: A scalable and extensible packet prober for active measurement of the internet,” in Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, ser. IMC '10, New York, NY, USA: Association for Computing Machinery, 2010, 239–245, ISBN: 9781450304832. DOI: 10.1145/1879141.1879171. [Online]. Available: https://doi.org/10.1145/1879141.1879171.

[7] K. Vermeulen, S. D. Strowes, O. Fourmaux, and T. Friedman, “Multilevel mda-lite paris traceroute,” in Proceedings of the Internet Measurement Conference 2018, ser. IMC '18, New York, NY, USA: Association for Computing Machinery, 2018, 29–42, ISBN: 9781450356190. DOI: 10.1145/3278532.3278536. [Online]. Available: https://doi.org/10.1145/3278532.3278536.

[8] O. Gasser, R. Holz, and G. Carle, “A deeper understanding of ssh: Results from internet-wide scans,” in 2014 IEEE Network Operations and Management Symposium (NOMS), IEEE, 2014, pp. 1–9.

[9] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman, “Mining your ps and qs: Detection of widespread weak keys in network devices,” in Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12), 2012, pp. 205–220.

[10] Z. Durumeric, E. Wustrow, and J. A. Halderman, “Zmap: Fast internet-wide scanning and its security applications,” in Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 13), 2013, pp. 605–620.

[11] D. Adrian, Z. Durumeric, G. Singh, and J. A. Halderman, “Zippier zmap: Internet-wide scanning at 10 gbps,” in Proceedings of the 8th USENIX Conference on Offensive Technologies, ser. WOOT’14, San Diego, CA: USENIX Association, 2014, pp. 8–8. [Online]. Available: `http://dl.acm.org/citation.cfm?id=2671293.2671301`.

[12] M. Fayed, L. Bauer, V. Giotsas, S. Kerola, M. Majkowski, P. Odintsov, J. Sitnicki, T. Chung, D. Levin, A. Mislove, C. A. Wood, and N. Sullivan, “The Ties That Un-Bind: Decoupling IP from Web Services and Sockets for Robust Addressing Agility at CDN-Scale,” in Proceedings of the 2021 ACM SIGCOMM 2021 Conference, ser. SIGCOMM ’21, New York, NY, USA: Association for Computing Machinery, 2021.

Internet Protocol v4
Acknowledgements

- Jim Kurose, University of Massachusetts, Amherst
- Keith Ross, Polytechnic Institute of NYC
- Olivier Bonaventure, University of Liege
- Srinivasan Keshav, University of Cambridge