CS 168
Software-Defined Networking (SDN)

Fall 2022

Guest Lecture: Scott Shenker

CS168.io
Be Forewarned....

- First classroom lecture in four years...
  - ...so this could be very rough

- Please ask questions when I’m floundering
  - To save us all the embarrassment

- When I ask a question, not looking for an answer
  - I’m asking you to think!

- Lecture will start slowly, with lots of generalities
  - This is to establish the necessary context
  - But things get more specific towards the end
Goal For Today

- Provide the “why” of software-defined networking
  - Why was it needed, why did it come about,...
  - Some history, some gossip, and the post-hoc rationale
  - *An exercise in retrospective architectural thinking*

- Almost none of the real “how”
  - Go read papers (RCP, 4D, NOX, ONIX, NetVirt, Fabric, …)
  - Sylvia is actively working on new ways of doing SDN
  - But the main point of SDN isn’t in the details of how...

- I am presenting the “canonical” version of SDN
  - Which absolutely no one uses in its pure form
  - But is the best way to understand what SDN is conceptually
We Begin With Two Questions
Q#1: What Is SDN?

- SDN is a way of managing networks
  - Will clarify what that means later in lecture

- However, SDN is not a revolutionary technology…
  - …just a way of organizing network functionality

- But that’s all the Internet architecture is….
  - The Internet architecture isn’t clever, but it is deeply wise

- SDN isn’t clever, and we can only hope it is wise
  - We’ll find out in thirty or forty years…
Q#2: Where did SDN come from?

- ~2004: Research on new management paradigms
  - RCP, 4D [Princeton, CMU,…]
  - SANE, Ethane [Stanford/Berkeley]
  - Industrial efforts with similar flavor (most not published)

- 2008: Software-Defined Networking (SDN)
  - NOX Network Operating System [Nicira]
  - OpenFlow switch interface [Stanford/Nicira]

- 2011: Open Networking Foundation (ONF)
  - **Board**: Google, Yahoo, Verizon, DT, Msft, Fbook, NTT, GS
  - **Members**: Cisco, Juniper, HP, Dell, Broadcom, IBM, …
Where did SDN really come from?

Martín Casado
Current Status Of SDN

- SDN accepted as **right way to do networking**
  - Commercialized, in production use, growing revenue
    - E.g., in use at Google/MSoft/Amazon, carriers partially adopted
  - Not fully adopted by router vendors, so this talk will often refer to pre-SDN practices in the present tense

- Was an insane level of SDN hype, and backlash…
  - SDN doesn’t work miracles, merely makes things easier

- But the real question is: **why the rapid adoption?**
  - 2004: idea, 2008: design, 2011: industry **frenzy**
  - This is incredibly fast for networking!
Why The Rapid Adoption?

- When a technology is adopted so quickly, it must be addressing a significant pain point.
  - Especially true in networking, which changes very slowly.

- SDN was addressing two huge pain points.

  1. Cisco’s extreme market power
     - Explains why vendors jumped on SDN.
  2. The poor state of network management
     - Explains why customers cared about SDN.
Network Management
What is network management?

- Recall the two “planes” of networking
  - **Data plane**: forwarding packets
    - Based on local forwarding state
  - **Control plane**: computing that forwarding state
    - Involves coordination with rest of system
- Broad definition of “network management”:
  - *Everything having to do with the control plane*
Original Goals For The Control Plane

- **Basic connectivity**: route packets to destination
  - Forwarding state computed by routing protocols
  - Globally (intradomain) distributed algorithms

- **Interdomain policy**: find policy-compliant paths
  - Done by globally (interdomain) distributed BGP

For long time, these were the only relevant goals!

- **What other goals are relevant now?**

Here are a few examples…
Isolation Of Logical LANs

- L2 bcast protocols often used for discovery
  - Useful, unscalable, invasive

- Want multiple logical LANs on a physical network
  - Retain usefulness, cope with scaling, provide isolation

- Use VLANs (virtual LANs) tags in L2 headers
  - Controls where broadcast packets go
  - Can create multiple logical L2 networks
  - Routers connect these logical L2 networks

- No universal method for setting VLAN state
Access Control

- Operators want to limit access to various hosts
  - “Don’t let laptops access backend database machines”
  - Crucial for security
- This can be imposed by routers using ACLs
  - ACL: Access Control List
- Example entry in ACL: <header template; drop>
  - If not port 80, drop
  - If source address = X, drop
- These are typically configured manually
  - And often implemented in firewalls
Traffic Engineering

- Choose routes to spread traffic load across links

- Two main methods:
  - Setting up MPLS tunnels (*MPLS is layer 2.5*)
  - Adjusting weights in OSPF

- Often done with centralized computation
  - Take snapshot of topology and load
  - Compute appropriate MPLS/OSPF state
  - Send state to network
Net management now has many goals

- Achieving these goals is job of the control plane...
- ...which currently involves many mechanisms

- **Globally distributed**: routing algorithms
- **Manual/scripted configuration**: ACLs, VLANs
- **Centralized computation**: Traffic engineering
Bottom Line

- Many different control plane mechanisms
- Each designed from scratch for their intended goal
- Encompassing a wide variety of implementations
  - Distributed, manual, centralized,…
- And none of them particularly well designed
- **Network control plane was a complicated mess!**
  - With mediocre functionality…
- Big contrast with simple and functional dataplane
Questions?
How Have We Managed To Survive?

- Network admins must master this complexity
  - Understand all aspects of networks
  - Must keep myriad details in mind

- Networks require large expert admin staffs
  - Much larger than compute admin staffs
  - This is how we survive, by mastering complexity

- This ability to master complexity is both a blessing
  - …and a curse!
A Simple Story About Complexity...

- ~1985: Don Norman visits Xerox PARC
  - Talks about user interfaces and stick shifts
  - Do you even know what a stick shift is?
What Was His Point?

- The ability to **master complexity** is valuable
  - But not the same as the ability to **extract simplicity**

- Each has its role:
  - When first getting systems to work, *master complexity*
    - *Stick shifts!*
  - When making system easy to use, *extract simplicity*
    - *Automatic transmissions!*

- You will never succeed in extracting simplicity
  - *If you don’t recognize it is a different skill set than mastering complexity!*
What Is *My* Point?

- Networking had never made the distinction...
  - And therefore never made the transition from mastering complexity to extracting simplicity for control plane

- Until SDN, focused on mastering complexity
  - Networking “experts” are those that know all the details

- Network management had suffered as a result

- *Simplify network mngmt requires extracting simplicity*
  - And we had never bothered to do that for control plane
Forcing People To Make Transition?

- We are really good at mastering complexity
  - And it had worked for us for decades, why change?

- How do you make people change?
  - Make them cry!

- A personal story about algebra and complexity
  - School problems:
    \[
    3x + 2y = 8 \quad x + y = 3
    \]
  - My father’s problems:
    \[
    327x + 26y = 8757 \quad 45x + 57y = 7776
    \]
  - My response: (1) I cried, (2) I learned algebra
How Do You Make Network Operators Cry?

What convinced network operators that they needed SDN?
Step 1: Large datacenters

- 100,000s machines; 10,000s switches
- Pushing the limits of what we could handle….
Step 2: Multiple tenancy

- Large datacenters can host many customers
  - Gave rise to the modern public cloud

- Each customer gets their own logical network
  - Customer should be able to set policies on this network
  - ACLs, VLANs, etc.

- If there are 1000 customers, that adds 3 oom
  - Where oom = orders of magnitude

- This went way beyond what we could handle
  - Because our control plane is so primitive!
Net Operators Were Now Weeping…

- They had been beaten by complexity
- The era of ad hoc control mechanisms was over
- We needed a simpler, more systematic design
  - We needed algebra, not arithmetic…

- But note the contrast between banks and multitenant datacenters:
  - One group willing to continue mastering complexity
  - The other was defeated, and needed something new
    - And they were desperate, which is why the rapid adoption
What Do We Do Now?

- We had been defeated by complexity in DCs
- So we had to “extract simplicity”!
- So how do you “extract simplicity”?
An Example Transition: Programming

- Machine languages: no abstractions
  - Had to deal with low-level details
  - Mastering complexity was crucial

- Higher-level languages: OS and other abstractions
  - File system, virtual memory, abstract data types, ...

- Modern languages: even more abstractions
  - Object orientation, garbage collection, ...

Abstractions key to extracting simplicity
“The Power of Abstraction”

“Modularity based on abstraction is the way things get done”

−Barbara Liskov

Abstractions ➔ Interfaces ➔ Modularity
What About Network Abstractions?

- Consider the data and control planes separately
- Different tasks, so naturally different abstractions
Abstractions for Data Plane: Layers

Applications
...built on...
Reliable (or unreliable) transport
...built on...
Best-effort global packet delivery
...built on...
Best-effort local packet delivery
...built on...
Physical transfer of bits
Control Plane Abstractions
Many Control Plane Mechanisms

- Variety of goals, no modularity:
  - **Routing**: distributed routing algorithms
  - **Isolation**: ACLs, VLANs, Firewalls,…
  - **Traffic engineering**: adjusting weights, MPLS,…

- Control Plane: mechanism without abstraction
  - *Too many mechanisms, not enough functionality*
SDN: An Exercise in Finding Control Plane Abstractions
Lecture So Far

- We have motivated the need for SDN
  - Networks admins were using arithmetic...
  - ...but suddenly needed algebra

- We now talk about how SDN met that need
  - What is the “algebra” of network management
How do you find abstractions?

- You start with a task you need to perform.
- You then decompose the task.
- ...and define abstractions for each subtask.
- Let’s do that for the control plane.
- Basic task is to compute forwarding state.
- But this task has several subtasks or constraints.
Task: Compute Forwarding State

- Consistent with low-level hardware/software
  - Which might depend on particular vendor

- Based on entire network topology
  - Because many control decisions depend on topology

- For all routers/switches in network
  - Every router/switch needs forwarding state
The Pre-SDN Approach

- Design one-off mechanisms that deal with all three
  - E.g., routing protocols deal with all three subproblems

- A sign of how much we love complexity

- No other field would do it this way!

- They would define abstractions to handle each subtask independently

- …and so should we!

- And that is what leads to SDN
Separate Concerns With Abstractions

1. Be compatible with low-level hardware/software
   Need an abstraction for general forwarding model

2. Make decisions based on entire network
   Need an abstraction for network state

3. Compute configuration of each physical device
   Need an abstraction that simplifies configuration
Abs#1: Forwarding Abstraction

- Express intent independent of implementation
  - Don’t want to deal with proprietary HW and SW

- OpenFlow is one proposal for forwarding
  - Standardized interface to switch
  - Configuration in terms of flow entries: <header, action>

- Design details concern exact nature of:
  - Header matching
  - Allowed actions
Separate Concerns With Abstractions

1. Be compatible with low-level hardware/software
   Need an abstraction for general forwarding model

2. Make decisions based on entire network
   Need an abstraction for network state

3. Compute configuration of each physical device
   Need an abstraction that simplifies configuration
Abs#2: Network State Abstraction

- Abstract away various distributed mechanisms

- Abstraction: **global network view**
  - Annotated network graph provided through an API

- Implementation: “Network Operating System”
  - Runs on servers in network (“controllers”)
  - Replicated for reliability

- Information flows both to and from NOS
  - Information *from* routers/switches to form “view”
  - Configurations *to* routers/switches to control forwarding
Network Operating System

- Think of it as a centralized link-state algorithm
- Switches send connectivity info to controller
- Controller computes forwarding state
  - Some control program that uses the topology as input
- Controller sends forwarding state to switches
  - Using forwarding abstraction (OpenFlow)
- Controller is replicated for resilience
  - System is only “logically centralized”
Control Program

Distributed algorithm running between neighbors

Complicated task-specific distributed algorithm

Network OS

routing, access control, etc.

Global Network View

Network of Switches and/or Routers
Major Change In Paradigm

- Control program: **Configuration = Function(view)**
  - Configuration means set of forwarding entries

- Control mechanism now program using NOS API

- Not a distributed protocol, just a graph algorithm
  - All distributed algorithms in NOS

- Configurations are passed to switches by NOS
Software Defined Network (So Far)

routing, access control, etc.

Control Program

Global Network View

Network OS
Questions?
Separate Concerns With Abstractions

1. Be compatible with low-level hardware/software
   Need an abstraction for general forwarding model

2. Make decisions based on entire network
   Need an abstraction for network state

3. Compute configuration of each physical device
   Need an abstraction that simplifies configuration

   Otherwise, control program must compute forwarding entries for every switch in network…. 
Abs#3: Specification Abstraction

- Control mechanism specifies desired behavior
  - Whether it be isolation, access control, or QoS

- It should not be responsible for implementing that behavior on physical network infrastructure
  - Requires configuring the forwarding tables in each switch

- Proposed abstraction: **abstract view** of network
  - Abstract view models only enough detail to **specify goals**
  - Will depend on task semantics
  - Now called “intention-based” networking
  - Think of this as providing a compiler...
Simple Example: Access Control

Abstract Network View

Global Network View
Software Defined Network

Abstract Network View

Virtualization Layer

Global Network View

Network OS
Software Defined Network

Control Program

Configuration of Abstract Network View

Virtualization Layer

Configuration of Global Network View

Network OS

Diagram showing the layers of a Software Defined Network (SDN) with Control Program at the top, followed by the Virtualization Layer, then the Configuration of Global Network View, and finally the Network OS at the bottom with interconnected nodes.
What This Really Really Means
Routing Application

- Look at graph of network
- Compute routes
- Give to SDN platform, which passes on to switches
- *Graph algorithm, not distributed protocol*
Access Control Application

- Control program decides who can talk to who
  - E.g., based on security category

- Pass this information to SDN platform

- Appropriate ACL flow entries are added to network
  - In the right places (based on the topology)

- The control program that decides who can talk to whom doesn’t care what the network looks like!
Clean Separation Of Concerns

- **Control program**: specify goals on abstract view
  - Driven by **Operator Requirements**

- **Virt. Layer**: abstract view $\leftrightarrow$ global view
  - Implements goals on network (as in global view)
  - Driven by **Specification Abstraction** for particular task

- **NOS**: global view $\leftrightarrow$ physical switches
  - API: driven by **Network State Abstraction**
  - Switch interface: driven by **Forwarding Abstraction**
SDN: *Layers* For The Control Plane

Control Program

Abstract Network View

Virtualization Layer

Global Network View

Network OS
Questions?
Abstrns Don’t Remove Complexity

- NOS, Virtualization are complicated pieces of code

- SDN merely localizes the complexity:
  - Simplifies interface for control program (use-specific)
  - Pushes complexity into *reusable* code (SDN platform)
  - This is the trajectory of computer science

- This is the big payoff of SDN: modularity!
  - The core distribution mechanisms can be reused
  - Control programs only deal with their specific function
Why Is SDN Important?

● As a design:
  ● It is more modular, enabling faster innovation
  ● Control programs become very simple!

● As an academic endeavor:
  ● Provides abstractions that enable systematic reasoning
  ● Can reason about control program, without looking at each switch…

● As a change in the ecosystem:
  ● Open switch interfaces reduce vendor lock-in
  ● Not clear that this will happen (why?)
Common Questions About SDN?

- Is SDN less scalable, secure, resilient,…?
- Can SDN be extended to the WAN?
- Is OpenFlow the right forwarding abstraction?
- Is SDN incrementally deployable?
Common Questions About SDN?

- Is SDN less scalable, secure, resilient,…? No
- Can SDN be extended to the WAN? Yes
- Is OpenFlow the right forwarding abstraction? No
- Is SDN incrementally deployable? Yes

How can this be?
What About Deployment?

- Most of SDN’s design is in software on servers
  - NOS and virtualization layer run on servers
  - Deploying these components is easy!

- But all routers must support OpenFlow
  - To provide information to the SDN controllers
  - To receive flow entries from the SDN controllers

- Requires replacing all routers in network
  - Routers are closed/proprietary, vendors won’t upgrade

- So the question is…
How Did We Get This Deployed?

- Get everyone to buy new OpenFlow switches?
- That is a completely ludicrous approach
  - *Though one we believed in at Nicira for a while*
- So, how did we deploy SDN?
  - Without them buying new switching hardware
  - And in some cases not even talking to the networking team at the company....
- Think about it....
Fact #1

- Most additional control plane functionality can be implemented at the edge
  - Access control, LAN Isolation, traffic engineering,…
  - Think about this for a second..

- Network core merely needs to deliver packets
  - Pre-SDN networking technologies pretty good at this
  - i.e., control plane for core only has its original task

- So only need to add SDN at network edge…

- This edge/core split arises in other contexts
  - E.g., MPLS, which has been widely adopted
Fact #2

- The operators who were crying were from large multitenant datacenters.

- They run hypervisors on their hosts, to support VMs initiated by tenants.

- These two facts gave us an opening.
Deployment In Virtualized Datacenters

- Virtualization (VMs) is supported by hypervisors
- Hypervisors use virtual switches to connect VMs
- Make this virtual (software) switch SDN-compatible
  - And you’ll be able to deploy SDN without any new HW
- Open vSwitch was an OpenFlow-capable vSwitch
  - Developed by Nicira, inserted into in Linux, Xen, etc.
- SDN now deployable without any HW deployment!
- This applies only to multitenant datacenters
  - But they were our only customers!
Network in Regular Setting
Network in Virtualized Setting
Virtual Switches (vSwitch)

- vSwitch is first-hop switch for all VMs
  - vSwitch sends packet to other VMs, or to physical network

- vSwitch is a software switch
  - If it supports OpenFlow, can be controlled by NOS
Physical View of Virtualized Network
Logical View of Virtualized Network

All edge switches are vSwitches
vSwitches are Sufficient in VDCs

- vSwitches enough to implement most CP functions
  - Access control, QoS, mobility, migration, monitoring,…

- Physical network becomes static crossbar
  - Crossbar: just delivers packets from edge-to-edge
  - Simple to implement and manage
  - Mostly static (only responds to changes inside core)

- Edge handles all dynamic/configured functions
  - Tracking VM movement
  - Access control policies
  - …
Managing Physical Network

Physical Switches

NOS
Managing Virtualized Network

NOS only needs to control vSwitches at edge

Physical network is logical crossbar
vSwitches as Insertion Point

- Can insert new functionality into datacenters with
  - Hypervisors with OpenFlow-enabled vSwitch
  - Network Operating System (on servers)

- No change to physical infrastructure
  - Legacy hosts
  - Legacy network components

- This last issue isn’t just a technical point
  - The network remaining completely unchanged is huge!
Deploying SDN in VDCs

- Because the network is completely unchanged, the deployment can be managed by the compute team, not the networking team.
  - Which have very different perspectives

- Networking team:
  - Very conservative, need to “not fail”, HW-oriented

- Compute team:
  - More nimble, need to deliver functionality, SW-oriented

- Initial SDN deployments were not network-driven
  - Which is what made them possible!
Questions?

About this lecture, or anything else...
Demo #1: DHCP and ARP
Welcome to Wireshark

Capture

...using this filter: [Enter a capture filter ...]  [All interfaces shown]

Wi-Fi: en0
Loopback: lo0
awdl0
llw0
utun0
utun1
utun2
Thunderbolt 0: en1
Thunderbolt 0: en2
Thunderbolt 1: en3
Thunderbolt 2: en4
gf0
stf0
ap1
Thunderbolt Bridge: bridge0
Cisco remote capture: ciscodump
Random packet generator: randpkt
SSH remote capture: sshdump
UDP Listener remote capture: udpdump
Wi-Fi remote capture: wifidump

Learn

User’s Guide  Wiki  Questions and Answers  Mailing Lists  SharkFest  Wireshark Discord

You are running Wireshark 4.0.1 (v4.0.1-0-ge9f3970b1527). You receive automatic updates.
ifconfig en0:
  flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
  options=4433<RXCSUM,TSO4,TSO6,CHANNEL_ID,PARTIAL_CSUM,ZEROINVERT_CSUM>
  ether 14:7d:da:3c:7e:19
  nd6 options=201<PERFORMNUD,DAD>
  media: autoselect
  status: inactive
ifconfig en0

en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1000
    options=6432<RXCSUM,TXCSUM,TSQ4,TSQ6,CHANNEL_10,PARTIAL_CSUM,ZEROCSUM> ether 14:77:da:3c:17:e19
    nd6 options=201<PERFORMNUD,DAD>
    media: autoselect
    status: inactive
<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
</table>

Capturing from Wi-Fi: en0

Wi-Fi: en0: <live capture in progress>
<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>::</td>
<td>ff02::fffd:ee29</td>
<td>ICMPv6</td>
<td>86</td>
<td>Neighbor Solicitation for fe80::1cc2:5d0b:1ed3:ee29</td>
</tr>
<tr>
<td>2</td>
<td>0.001543</td>
<td>Apple_3c:7e:19</td>
<td>euro:00:1c:3e:19</td>
<td>ARP</td>
<td>42</td>
<td>Who has 192.168.4.17 Tell 172.168.4.20</td>
</tr>
<tr>
<td>3</td>
<td>0.002456</td>
<td>0.0.0.0</td>
<td>255.255.255.255</td>
<td>DHCP</td>
<td>342</td>
<td>DHCP Request - Transaction ID 0x1d1be7a</td>
</tr>
<tr>
<td>4</td>
<td>0.004740</td>
<td>fe80::1cc2:5d0b:1e21</td>
<td>ff02::2</td>
<td>ICMPv6</td>
<td>62</td>
<td>Router Solicitation</td>
</tr>
<tr>
<td>5</td>
<td>0.007789</td>
<td>142.250.191.37</td>
<td>192.168.4.20</td>
<td>TLSv1</td>
<td>195</td>
<td>Application Data, Application Data</td>
</tr>
<tr>
<td>6</td>
<td>0.008143</td>
<td>54.188.33.22</td>
<td>192.168.4.20</td>
<td>TLSv1</td>
<td>168</td>
<td>Application Data, Application Data</td>
</tr>
<tr>
<td>7</td>
<td>0.008151</td>
<td>eero_29:e6:ed</td>
<td>Apple_3c:7e:19</td>
<td>ARP</td>
<td>42</td>
<td>192.168.4.1 is at 00:ab:48:29:6e:ed</td>
</tr>
<tr>
<td>8</td>
<td>0.008384</td>
<td>fe80::2ab:48ff:fe21</td>
<td>fe80::1cc2:5d0b:1e21</td>
<td>ICMPv6</td>
<td>110</td>
<td>Router Advertisement from 00:ab:48:29:6e:ed</td>
</tr>
<tr>
<td>9</td>
<td>0.008547</td>
<td>Apple_3c:7e:19</td>
<td>Broadcast</td>
<td>ARP</td>
<td>42</td>
<td>ARP Announcement for 192.168.4.20</td>
</tr>
<tr>
<td>10</td>
<td>0.008558</td>
<td>::</td>
<td>ff02::ff1b:3503</td>
<td>ICMPv6</td>
<td>86</td>
<td>Neighbor Solicitation for fdf2:2a4:5f02:1:1019:c20:81b:3503</td>
</tr>
<tr>
<td>11</td>
<td>0.019601</td>
<td>192.168.4.1</td>
<td>192.168.4.20</td>
<td>DHCP</td>
<td>342</td>
<td>DHCP ACK - Transaction ID 0x1d1be7a</td>
</tr>
<tr>
<td>12</td>
<td>0.033803</td>
<td>192.168.4.20</td>
<td>224.0.8.251</td>
<td>MDNS</td>
<td>352</td>
<td>Standard query response 0x000000 PTR, cache flush sarahmacbook.local PTR, cache flush sarahmacbook.local P</td>
</tr>
<tr>
<td>13</td>
<td>0.033872</td>
<td>fe80::1cc2:5d0b:1e21</td>
<td>ff02::2fb</td>
<td>MDNS</td>
<td>372</td>
<td>Standard query response 0x000000 PTR, cache flush sarahmacbook.local PTR, cache flush sarahmacbook.local P</td>
</tr>
<tr>
<td>14</td>
<td>0.087214</td>
<td>eero_29:e6:ed</td>
<td>Apple_3c:7e:19</td>
<td>ARP</td>
<td>42</td>
<td>192.168.4.1 is at 00:ab:48:29:6e:ed</td>
</tr>
<tr>
<td>15</td>
<td>0.087262</td>
<td>192.168.4.20</td>
<td>50.0.1.1</td>
<td>DNS</td>
<td>73</td>
<td>Standard query 0x941a AAAA ipv4only.arpa</td>
</tr>
</tbody>
</table>

> Frame 2: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface en0, id 0
> Address Resolution Protocol (request)
1. 0.000000
   ::
   ff02::1:ffd3:ea29
   ICMPv6 86 Neighbor Solicitation for fe80::1:cc5:d0b:1ed3:ea29

2. 0.001141
   Apple_3c:7e:19
   eero_29:e6:ed
   ARP 42 Who has 192.168.4.1? Tell 192.168.4.1

3. 0.008256
   0.0.0.0
   255.255.255.255
   DHCP 342 DHCP Request - Transaction ID 0xda18e7a

4. 0.008470
   fe80::1:cc2:5d0b:1ed3:ea29
   ff02::2
   ICMPv6 102 Router Solicitation

5. 0.007789
   142.250.191.37
   192.168.4.20
   TLsv1_0 195 Application Data, Application Data

6. 0.008143
   54.188.33.22
   192.168.4.20
   TLsv1_0 186 Application Data, Application Data, Application Data

7. 0.008151
   eero_29:e6:ed
   Apple_3c:7e:19
   ARP 42 192.168.4.1 is at 00:ab:48:29:e6:ed

8. 0.008384
   fe80::2ab:48ff:fe29:fe80::1:cc2:5d0b:1ed3:ea29
   ICMPv6 110 Router Advertisement from 00:ab:48:29:e6:ed

9. 0.008547
   Apple_3c:7e:19
   Broadcast
   ARP 42 ARP Announcement for 192.168.4.20

10. 0.008550
    ::
    ff02::1:ff1:3503
    ICMPv6 86 Neighbor Solicitation for fdf2:2a4:ff02:1:810c:2a8:81:3503

11. 0.019061
    192.168.4.1
    192.168.4.20
    DHCP 342 DHCP ACK - Transaction ID 0xda18e7a

12. 0.033883
    224.0.0.251
    192.168.4.20
    MDNS 352 Standard query response 00000000, cache flush sarahmacbook.local PTR, cache flush sarahmacbook.local

13. 0.033883
    fe80::1:cc2:5d0b:1ed3:ea29
    ff02::2
    MDNS 372 Standard query response 00000000, cache flush sarahmacbook.local PTR, cache flush sarahmacbook.local

14. 0.074501
    Apple_3c:7e:19
    Broadcast
    ARP 42 Who has 192.168.4.1? Tell 192.168.4.1

15. 0.087214
    eero_29:e6:ed
    Apple_3c:7e:19
    ARP 42 192.168.4.1 is at 00:ab:48:29:e6:ed

16. 0.087262
    192.168.4.20
    50.0.1.1
    DNS 73 Standard query 0x941a AAAA ipv4only.arpa
<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>::</td>
<td>ff02::1:fd3:ea29</td>
<td>ICMPv6</td>
<td>86</td>
<td>Neighbor Solicitation for fe80::1cc2:5d0b:1ed3:ea29</td>
</tr>
<tr>
<td>2</td>
<td>0.000141</td>
<td>Apple_3c:7e:19</td>
<td>euro_3c:7e:19</td>
<td>ARPS</td>
<td>0</td>
<td>Who has 192.168.4.28 Tell 192.168.4.29</td>
</tr>
<tr>
<td>3</td>
<td>0.002456</td>
<td>0.0.0.0</td>
<td>255.255.255.255</td>
<td>DHCP</td>
<td>342</td>
<td>DHCP Request - Transaction ID 0x1814e7a</td>
</tr>
<tr>
<td>4</td>
<td>0.004740</td>
<td>fe80::1cc2:5d0b:1e...</td>
<td>ff02::2</td>
<td>ICMPv6</td>
<td>62</td>
<td>Router Solicitation</td>
</tr>
<tr>
<td>5</td>
<td>0.007789</td>
<td>142.250.191.37</td>
<td>192.168.4.20</td>
<td>TLSv1</td>
<td>195</td>
<td>Application Data, Application Data</td>
</tr>
<tr>
<td>6</td>
<td>0.008143</td>
<td>54.188.33.22</td>
<td>192.168.4.20</td>
<td>TLSv1</td>
<td>168</td>
<td>Application Data, Application Data</td>
</tr>
<tr>
<td>7</td>
<td>0.008151</td>
<td>eero_29:e6:ed</td>
<td>Apple_3c:7e:19</td>
<td>ARP</td>
<td>42</td>
<td>192.168.4.1 is at 00:ab:48:29:6e:ed</td>
</tr>
<tr>
<td>8</td>
<td>0.008384</td>
<td>fe80::2ab:48ff:fe2...</td>
<td>fe80::1cc2:5d0b:1e...</td>
<td>ICMPv6</td>
<td>110</td>
<td>Router Advertisement from 00:ab:48:29:6e:ed</td>
</tr>
<tr>
<td>9</td>
<td>0.008547</td>
<td>Apple_3c:7e:19</td>
<td>Broadcast</td>
<td>ARP</td>
<td>42</td>
<td>ARP Announcement for 192.168.4.20</td>
</tr>
<tr>
<td>10</td>
<td>0.008550</td>
<td>::</td>
<td>ff02::1:3503</td>
<td>ICMPv6</td>
<td>86</td>
<td>Neighbor Solicitation for fdf2:2a4:5f02:1:1019:c20:81b:3503</td>
</tr>
<tr>
<td>11</td>
<td>0.019001</td>
<td>192.168.4.1</td>
<td>192.168.4.20</td>
<td>DHCP</td>
<td>342</td>
<td>DHCP ACK - Transaction ID 0x1814e7a</td>
</tr>
<tr>
<td>12</td>
<td>0.033803</td>
<td>192.168.4.28</td>
<td>224.0.0.251</td>
<td>MDNS</td>
<td>352</td>
<td>Standard query response 0x000000 PTR, cache flush saramacbook.local PTR, cache flush saramacbook.local</td>
</tr>
<tr>
<td>13</td>
<td>0.033872</td>
<td>fe80::1cc2:5d0b:1e...</td>
<td>ff02::fb</td>
<td>DHCP</td>
<td>372</td>
<td>Standard query response 0x000000 PTR, cache flush saramacbook.local PTR, cache flush saramacbook.local</td>
</tr>
<tr>
<td>14</td>
<td>0.087214</td>
<td>eero_29:e6:ed</td>
<td>Apple_3c:7e:19</td>
<td>ARP</td>
<td>42</td>
<td>192.168.4.1 is at 00:ab:48:29:6e:ed</td>
</tr>
<tr>
<td>15</td>
<td>0.087262</td>
<td>192.168.4.20</td>
<td>50.0.1.1</td>
<td>DNS</td>
<td>73</td>
<td>Standard query 0x941a AAAA ipv4onlyarpa</td>
</tr>
</tbody>
</table>

> Frame: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface en0, id 0
> Address Resolution Protocol (ARP)
<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00s000</td>
<td>::</td>
<td>ff02::1:fd3:ea29</td>
<td>ICMPv6</td>
<td>86</td>
<td>Neighbor Solicitation for fe80::1:cc2::5d0b:1ed3:ea29</td>
</tr>
<tr>
<td>2</td>
<td>0.00s417</td>
<td>euro_3c7e:19</td>
<td>255.255.255.255</td>
<td>ARP</td>
<td>42</td>
<td>Who has 192.168.64.17 Tell 192.168.4.20</td>
</tr>
<tr>
<td>3</td>
<td>0.00s456</td>
<td>0.0.0.0</td>
<td>255.255.255.255</td>
<td>DHCP</td>
<td>342</td>
<td>DHCP Request - Transaction ID 0xd1b1e7a</td>
</tr>
<tr>
<td>4</td>
<td>0.00s474</td>
<td>fe80::1:cc2::5d0b:1e...</td>
<td>ff02::2</td>
<td>ICMPv6</td>
<td>62</td>
<td>Router Solicitation</td>
</tr>
<tr>
<td>5</td>
<td>0.00s789</td>
<td>142.250.191.37</td>
<td>192.168.4.20</td>
<td>TLSv1</td>
<td>195</td>
<td>Application Data, Application Data</td>
</tr>
<tr>
<td>6</td>
<td>0.00s814</td>
<td>54.188.33.22</td>
<td>192.168.4.20</td>
<td>TLSv1</td>
<td>186</td>
<td>Application Data, Application Data</td>
</tr>
<tr>
<td>7</td>
<td>0.00s815</td>
<td>eero_29:e6:ed</td>
<td>Apple_3c7e:19</td>
<td>ARP</td>
<td>42</td>
<td>192.168.4.1 is at 00:ab:48:29:e6:ed</td>
</tr>
<tr>
<td>8</td>
<td>0.00s838</td>
<td>fe80::2ab:48ff:fe2...</td>
<td>fe80::1:cc2::5d0b:1e...</td>
<td>ICMPv6</td>
<td>110</td>
<td>Router Advertisement from 00:ab:48:29:e6:ed</td>
</tr>
<tr>
<td>9</td>
<td>0.00s854</td>
<td>Apple_3c7e:19</td>
<td>Broadcast</td>
<td>ARP</td>
<td>42</td>
<td>ARP Announcement for 192.168.4.20</td>
</tr>
<tr>
<td>10</td>
<td>0.00s855</td>
<td>::</td>
<td>ff02::ff1:3503</td>
<td>ICMPv6</td>
<td>86</td>
<td>Neighbor Solicitation for fdf2:2a4:5f02:1:1019:c28:81b:3503</td>
</tr>
<tr>
<td>11</td>
<td>0.00s960</td>
<td>192.168.4.1</td>
<td>192.168.4.20</td>
<td>DHCP</td>
<td>342</td>
<td>DHCP ACK - Transaction ID 0xd1b1e7a</td>
</tr>
<tr>
<td>12</td>
<td>0.03s803</td>
<td>192.168.4.20</td>
<td>224.0.0.251</td>
<td>DNS</td>
<td>352</td>
<td>Standard query response 0x00000 PTR, cache flush sarahmacbook.local PTR, cache flush sarahmacbook.local PTR</td>
</tr>
<tr>
<td>13</td>
<td>0.03s872</td>
<td>fe80::1:cc2::5d0b:1e...</td>
<td>ff02::ff</td>
<td>DNS</td>
<td>372</td>
<td>Standard query response 0x00000 PTR, cache flush sarahmacbook.local PTR, cache flush sarahmacbook.local PTR</td>
</tr>
<tr>
<td>14</td>
<td>0.00s450</td>
<td>Apple_3c7e:19</td>
<td>Broadcast</td>
<td>ARP</td>
<td>42</td>
<td>Who has 192.168.4.17 Tell 192.168.4.20</td>
</tr>
<tr>
<td>15</td>
<td>0.00s724</td>
<td>eero_29:e6:ed</td>
<td>Apple_3c7e:19</td>
<td>ARP</td>
<td>42</td>
<td>192.168.4.1 is at 00:ab:48:29:e6:ed</td>
</tr>
<tr>
<td>16</td>
<td>0.00s726</td>
<td>192.168.4.20</td>
<td>50.0.1.1</td>
<td>DNS</td>
<td>373</td>
<td>Standard query 0x941a AAAA ipv4onlyarpa</td>
</tr>
</tbody>
</table>

> Frame 2: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface en0, id 0
> Address Resolution Protocol (request)

Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
Sender MAC address: Apple_3c7e:19 (14:7:da:3c:7e:19)
Sender IP address: 192.168.4.20
Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.4.20
Parameter Request List Item: (46) NetBIOS over TCP/IP Node Type

- Option: (57) Maximum DHCP Message Size
  Length: 2
  Maximum DHCP Message Size: 1500

- Option: (61) Client identifier
  Length: 7
  Hardware type: Ethernet (0x01)
  Client MAC address: Apple_3c7e:19 (14:7d:da:3c7e:19)

- Option: (50) Requested IP Address (192.168.4.20)
  Length: 4
  Requested IP Address: 192.168.4.20

- Option: (51) IP Address Lease Time
  Length: 4
  IP Address Lease Time: (776000s) 90 days

- Option: (12) Host Name
  Length: 12
  Host Name: sarahtombook

- Option: (255) End
  Option End: 255

Padding: 000000
<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>::</td>
<td>::</td>
<td>ICMPv6</td>
<td>86</td>
<td>Neighbor Solicitation for fe80::1cc2:5d0b:1ed3:ea29</td>
</tr>
<tr>
<td>2</td>
<td>0.001941</td>
<td>Apple_3c:7e:19</td>
<td>eero_29:e6:ed</td>
<td>ARP</td>
<td>42</td>
<td>Who has 192.168.4.17 Tell 192.168.4.20</td>
</tr>
<tr>
<td>3</td>
<td>0.002456</td>
<td>::</td>
<td>255.255.255</td>
<td>DHCP</td>
<td>342</td>
<td>DHCP Request - Transaction ID 0xd1B1e7a</td>
</tr>
<tr>
<td>4</td>
<td>0.004740</td>
<td>fe80::1cc2:5d0b:bc::</td>
<td>::</td>
<td>ICMPv6</td>
<td>62</td>
<td>Router Solicitation</td>
</tr>
<tr>
<td>5</td>
<td>0.007789</td>
<td>142.250.191.37</td>
<td>192.168.4.20</td>
<td>TSv1</td>
<td>195</td>
<td>Application Data, Application Data</td>
</tr>
<tr>
<td>6</td>
<td>0.008143</td>
<td>54.188.33.22</td>
<td>192.168.4.20</td>
<td>TSv1</td>
<td>168</td>
<td>Application Data, Application Data, Application Data</td>
</tr>
<tr>
<td>7</td>
<td>0.008151</td>
<td>eero_29:e6:ed</td>
<td>Apple_3c:7e:19</td>
<td>ARP</td>
<td>42</td>
<td>192.168.4.1 is at 00:ab:48:29:e6:ed</td>
</tr>
<tr>
<td>8</td>
<td>0.008384</td>
<td>fe80::2ab:48ff:fe::</td>
<td>fe80::1cc2:5d0b:bc::</td>
<td>ICMPv6</td>
<td>110</td>
<td>Router Advertisement from 00:ab:48:29:e6:ed</td>
</tr>
<tr>
<td>9</td>
<td>0.008547</td>
<td>Apple_3c:7e:19</td>
<td>Broadcast</td>
<td>ARP</td>
<td>42</td>
<td>ARP Announcement for 192.168.4.20</td>
</tr>
<tr>
<td>10</td>
<td>0.008558</td>
<td>::</td>
<td>ff02::ff1b:3503</td>
<td>ICMPv6</td>
<td>86</td>
<td>Neighbor Solicitation for ff02::ff1b:3503</td>
</tr>
<tr>
<td>11</td>
<td>0.019001</td>
<td>192.168.4.1</td>
<td>192.168.4.20</td>
<td>DHCP</td>
<td>342</td>
<td>DHCP ACK - Transaction ID 0xd1B1e7a</td>
</tr>
<tr>
<td>12</td>
<td>0.033803</td>
<td>192.168.4.20</td>
<td>224.0.0.251</td>
<td>MDNS</td>
<td>352</td>
<td>Standard query response 0x0000 PTR, cache flush sarahmacbook.local PTR, cache flush sarahmacbook.local</td>
</tr>
<tr>
<td>13</td>
<td>0.033872</td>
<td>ff02::ff1b:3503</td>
<td>00:00:00</td>
<td>MDNS</td>
<td>372</td>
<td>Standard query response 0x0000 PTR, cache flush sarahmacbook.local PTR, cache flush sarahmacbook.local</td>
</tr>
<tr>
<td>14</td>
<td>0.074501</td>
<td>Apple_3c:7e:19</td>
<td>Broadcast</td>
<td>ARP</td>
<td>42</td>
<td>192.168.4.1 Tell 192.168.4.20</td>
</tr>
<tr>
<td>15</td>
<td>0.087214</td>
<td>eero_29:e6:ed</td>
<td>Apple_3c:7e:19</td>
<td>ARP</td>
<td>42</td>
<td>192.168.4.1 is at 00:ab:48:29:e6:ed</td>
</tr>
<tr>
<td>16</td>
<td>0.087262</td>
<td>192.168.4.20</td>
<td>50.0.1.1</td>
<td>DNS</td>
<td>73</td>
<td>Standard query 0x941a AAAA ipv4only.arpa</td>
</tr>
</tbody>
</table>

Frame 7: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface en0, id 0
Address Resolution Protocol (reply)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender IP address: 192.168.4.1
Target MAC address: Apple_3c:7e:19 (14:7d:da:3c:7e:19)
Target IP address: 192.168.4.20
<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>::</td>
<td>f02::1:fc3:ea29</td>
<td>ICMPv6</td>
<td>86</td>
<td>Neighbor Solicitation for fe80::1:cc2:5d0:b:1ed:ea29</td>
</tr>
<tr>
<td>2</td>
<td>0.001941</td>
<td>eero_29:6e:ed</td>
<td>eero_29:6e:ed</td>
<td>ARP</td>
<td>42</td>
<td>Who has 192.168.4.17 Tell 192.168.4.20</td>
</tr>
<tr>
<td>3</td>
<td>0.002456</td>
<td>0.0.0.0</td>
<td>255.255.255.255</td>
<td>DHCP</td>
<td>342</td>
<td>DHCP Request - Transaction ID 0xd181e7a</td>
</tr>
<tr>
<td>4</td>
<td>0.004740</td>
<td>fe80::1:cc2:5d0:b:1fe2::</td>
<td>ff02::2</td>
<td>ICMPv6</td>
<td>62</td>
<td>Router Solicitation</td>
</tr>
<tr>
<td>5</td>
<td>0.007789</td>
<td>142.250.191.37</td>
<td>192.168.4.20</td>
<td>TSLv1</td>
<td>195</td>
<td>Application Data, Application Data</td>
</tr>
<tr>
<td>6</td>
<td>0.008143</td>
<td>54.188.33.22</td>
<td>192.168.4.20</td>
<td>Application Data, Application Data, Application Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.008151</td>
<td>eero_29:6e:ed</td>
<td>Apple_3c:7e:19</td>
<td>ARP</td>
<td>42</td>
<td>192.168.4.1 is at 00:ab:48:29:6ed</td>
</tr>
<tr>
<td>8</td>
<td>0.008384</td>
<td>fe80::1:cc2:5d0:b:1fe2::</td>
<td>fe80::1:cc2:5d0:b:1fe2::</td>
<td>ICMPv6</td>
<td>110</td>
<td>Router Advertisement from 00:ab:48:29:6e:ed</td>
</tr>
<tr>
<td>9</td>
<td>0.008547</td>
<td>Apple_3c:7e:19</td>
<td>Broadcast</td>
<td>ARP</td>
<td>42</td>
<td>ARP Announcement for 192.168.4.20</td>
</tr>
<tr>
<td>10</td>
<td>0.008550</td>
<td>::</td>
<td>ff02::1:ff1:3b503</td>
<td>ICMPv6</td>
<td>86</td>
<td>Neighbor Solicitation for fdf2::ca4::f02::1::1019::c20::81::3b503</td>
</tr>
</tbody>
</table>

DHCP: ACK (5)
- Option: (54) DHCP Server Identifier (192.168.4.1)
  - Length: 4
  - DHCP Server Identifier: 192.168.4.1
- Option: (51) IP Address Lease Time
  - Length: 4
  - IP Address Lease Time: (2880s) 8 hours
- Option: (1) Subnet Mask (255.255.252.0)
  - Length: 4
  - Subnet Mask: 255.255.252.0
- Option: (3) Router
  - Length: 4
  - Router: 192.168.4.1
- Option: (6) Domain Name Server
  - Length: 8
  - Domain Name Server: 50.0.0.1.1, 50.0.0.2.2
- Option: (255) End
  - Option End: 255
  - Padding: 0000000000000000000000000000000000000000000000000000000000000000
Frame 17: 130 bytes on wire (1040 bits), 130 bytes captured (1040 bits) on interface en0, id 0
> User Datagram Protocol, Src Port: 55, Dst Port: 59741
> Domain Name System (response)
> Transaction ID: 0x941a
> Flags: 0x0180 Standard query response, No error
> Questions: 1
> Answer RRs: 0
> Authority RRs: 1
> Additional RRs: 0
> Queries
> Authoritative nameservers
> [Request: 16]
[Time: 0.021599080 seconds]
Frame 16: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface en0
> Domain Name System (query)
> Flags: 0x0100 Standard query
> Questions: 1
> Answer RRs: 0
> Authority RRs: 0
> Additional RRs: 0

Queries
> ipv4onlyarpa: type AAAA, class IN [Response In: 17]

> Frame 15: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface en0
> Domain Name System (query)
> Flags: 0x0100 Standard query
> Questions: 1
> Answer RRs: 0
> Authority RRs: 0
> Additional RRs: 0

Queries
> ipv4onlyarpa: type AAAA, class IN [Response In: 17]
Questions?
Demo #2: Web
Part 1: HTTP - Level
CS168 Web Example

Look at this network diagram!
Look at this network diagram!
Look at this network diagram!
CS168 Web Example

Look at this network diagram!
Look at this network diagram!
CS186 Web Example

Look at this network diagram!
CS168 Web Example

Look at this network diagram!
Look at this network diagram!
Look at this network diagram!
CS168 Web Example

Look at this network diagram!
Look at this network diagram!
CS168 Web Example

Look at this network diagram!
Look at this network diagram!
Look at this network diagram!
Look at this network diagram!
CS168 Web Example

Look at this network diagram!
Look at this network diagram!
Part 2: Packet-Level
CS168 Web Example

Look at this network diagram!
CS168 Web Example

Look at this network diagram!
CS168 Web Example

Look at this network diagram!
CS168 Web Example

Look at this network diagram!
CS168 Web Example

Look at this network diagram!
Look at this network diagram!
Look at this network diagram!
<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>2.593772</td>
<td>2607:1f4b:4000:a000</td>
<td>2606:5c0:d803:1513</td>
<td>TCP</td>
<td>98</td>
<td>65183 - 443 [SYN] Seq=65535 Win=6513 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>32</td>
<td>2.626819</td>
<td>2607:1f4b:4000:a000</td>
<td>2606:5c0:d803:1513</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=65535 Win=13164 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>33</td>
<td>2.626867</td>
<td>2607:1f4b:4000:a000</td>
<td>2606:5c0:d803:1513</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>34</td>
<td>2.627591</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>35</td>
<td>2.627592</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>36</td>
<td>2.628032</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>37</td>
<td>2.628033</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>38</td>
<td>2.628038</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>39</td>
<td>2.628039</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>40</td>
<td>2.628040</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>41</td>
<td>2.628041</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>42</td>
<td>2.628042</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>43</td>
<td>2.628043</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>44</td>
<td>2.628044</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>45</td>
<td>2.628045</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>46</td>
<td>2.628046</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>47</td>
<td>2.628047</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>48</td>
<td>2.628048</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>49</td>
<td>2.628049</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>50</td>
<td>2.628050</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>51</td>
<td>2.628051</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
<tr>
<td>52</td>
<td>2.628052</td>
<td>2606:5c0:d803:1513</td>
<td>2607:1f4b:4000:a000</td>
<td>TCP</td>
<td>86</td>
<td>65183 - 443 [ACK] Ack=526 Win=6625 Len=1548 MSS=44 Tsa=1676261006 Tsrc=0 SACK_PERM</td>
</tr>
</tbody>
</table>

Transmission Control Protocols

Source Port: 65183
Destination Port: 443
Seq: 0
Len: 0

TCP segment of a reassembled PDU
CS168 Web Example

Look at this network diagram!
Sylvia Ratnasamy
Soda Hall 413
UC Berkeley
Berkeley, CA 94720-1776 USA
Email: first-name@eecs.berkeley.edu

Biography

I am an Associate Professor of Computer Science at the University of California, Berkeley. My research focuses on the design and implementation of networked systems. I co-lead the SPAN Center for networking research. A brief bio is here.

Teaching

CS268: Spring 2019, Spring 2016, Spring 2015
CS168: Fall 2014
EE122: Fall 2013

Service

Member of the Technical Steering Committee: ACM SIGCOMM (2015 - to date), ACM Sigmetrics 2011, ACM PRESTO Workshop 2010

Program Co-Chair: ACM SIGCOMM 2014, Usenix NSDI 2011, ACM PRESTO Workshop 2010

Member of the Program Committee: SIGCOMM 2016, ANCS 2016

HonNets 2012
NSDI 2010, CommNets 2010
SOSP 2009, HotPower 2009, Presto 2009

NSDI 2007, SIGCOMM 2007
Sylvia Ratnasamy  
Soda Hall 4143  
UC Berkeley  
Berkeley, CA 94720-1776 USA  
Email: first-name at eecs dot berkeley dot edu

Biography

I am an Associate Professor of Computer Science at the University of California, Berkeley. My research focuses on the design and implementation of networked systems. I co-lead the SPAN Center for networking research. A brief bio is here.

Teaching

CS268: Spring 2019 , Spring 2016 , Spring 2015 , Spring 2013  
CS168: Fall 2014  
EE122: Fall 2013

Service

Member of the Technical Steering Committee: ACM SIGCOMM (2015 - to date), ACM SenSys (2008 - 2013)  
Program Co-Chair: ACM SIGCOMM 2014, Usenix NSDI 2011, ACM PRESTO Workshop 2010  
Member of the Program Committee: SIGCOMM 2016, ANCS 2016  
HotNets 2012  
NSDI 2010, CommNets 2010  
NSDI 2006, SIGCOMM 2006
<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td>2.572142</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>1514</td>
<td>Seq=1 Ack=1 Win=2048 Len=1448 Tsvval=1913937172 TSecr=1854437767</td>
</tr>
<tr>
<td>45</td>
<td>2.572150</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>1514</td>
<td>Seq=1 Ack=1 Win=2048 Len=1448 Tsvval=1913937172 TSecr=1854437767</td>
</tr>
<tr>
<td>47</td>
<td>2.572575</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>66</td>
<td>Seq=1 Ack=3121 Win=421 Len=0 Tsvval=1854455783 TSecr=1913937172</td>
</tr>
<tr>
<td>48</td>
<td>2.584568</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>66</td>
<td>Seq=1 Ack=3121 Win=421 Len=0 Tsvval=1854455783 TSecr=1913937172</td>
</tr>
<tr>
<td>50</td>
<td>2.615459</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>66</td>
<td>Seq=1 Ack=2897 Win=2002 Len=8 Tsvval=1913937215 TSecr=1854455745</td>
</tr>
<tr>
<td>52</td>
<td>2.615883</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>66</td>
<td>Seq=1 Ack=2897 Win=2002 Len=8 Tsvval=1913937215 TSecr=1854455745</td>
</tr>
<tr>
<td>56</td>
<td>2.615591</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>66</td>
<td>Seq=3435 Ack=321 Win=421 Len=0 Tsvval=1854455745 TSecr=1913937172</td>
</tr>
<tr>
<td>57</td>
<td>2.615594</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>66</td>
<td>Seq=7241 Ack=321 Win=421 Len=0 Tsvval=1854455745 TSecr=1913937172</td>
</tr>
<tr>
<td>58</td>
<td>2.615604</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>66</td>
<td>Seq=6809 Ack=321 Win=421 Len=0 Tsvval=1854455745 TSecr=1913937172</td>
</tr>
<tr>
<td>59</td>
<td>2.615607</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>66</td>
<td>Seq=18137 Ack=321 Win=421 Len=0 Tsvval=1854455745 TSecr=1913937172</td>
</tr>
<tr>
<td>57</td>
<td>2.615609</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>66</td>
<td>Seq=11585 Ack=321 Win=421 Len=0 Tsvval=1854455745 TSecr=1913937172</td>
</tr>
<tr>
<td>58</td>
<td>2.615612</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>66</td>
<td>Seq=13833 Ack=321 Win=421 Len=0 Tsvval=1854455745 TSecr=1913937172</td>
</tr>
<tr>
<td>59</td>
<td>2.615622</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>66</td>
<td>Seq=2669 Ack=321 Win=421 Len=0 Tsvval=1854455745 TSecr=1913937172</td>
</tr>
<tr>
<td>60</td>
<td>2.615690</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>66</td>
<td>Seq=3121 Ack=14481 Win=1867 Len=8 Tsvval=1913937215 TSecr=1854455745</td>
</tr>
<tr>
<td>61</td>
<td>2.615686</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>66</td>
<td>Seq=14481 Ack=321 Win=421 Len=0 Tsvval=1854455745 TSecr=1913937172</td>
</tr>
<tr>
<td>62</td>
<td>2.615689</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>66</td>
<td>Seq=35929 Ack=321 Win=421 Len=0 Tsvval=1913937215 TSecr=1854455745</td>
</tr>
<tr>
<td>63</td>
<td>2.615670</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>66</td>
<td>Seq=3121 Ack=17377 Win=1821 Len=8 Tsvval=1913937215 TSecr=1854455745</td>
</tr>
<tr>
<td>64</td>
<td>2.615680</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>66</td>
<td>Seq=3121 Ack=17443 Win=1828 Len=8 Tsvval=1913937215 TSecr=1854455745</td>
</tr>
</tbody>
</table>

**Urgent Pointer:** 0

**Options:** (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps

**[Time/Stamp]**

**[SEQ/ACK analysis]**

**TCP payload (224 bytes)**

**TCP segment data (1224 bytes)**

**[Reassembled TCP segments (312 bytes): #44(#448), #45(#448), #46(#224)]**

**HyperText Transfer Protocol**

**GET /syivia/ HTTP/1.1

Host: people.eecs.berkeley.edu
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15.7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/185
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9

[Full request URL] http://people.eecs.berkeley.edu/syivia/

[HTTP request 1/3]

[Response in fast] 43

[Next Request In Frame: 69]
### Network Traffic Capture

<table>
<thead>
<tr>
<th>No</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td>2.571214</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>1514</td>
<td>5646 = 80 [ACK] Seq=1 Ack=1 Win=2048 Len=1448 T RVA=1913937172 T Seq=1854437767 [TCP segment of a reassembled PDU]</td>
</tr>
<tr>
<td>45</td>
<td>2.572510</td>
<td>128.32.32.190</td>
<td>128.32.244.190</td>
<td>TCP</td>
<td>1514</td>
<td>5646 = 80 [ACK] Seq=1 Ack=1 Win=2048 Len=1448 T RVA=1913937172 T Seq=1854437767 [TCP segment of a reassembled PDU]</td>
</tr>
<tr>
<td>46</td>
<td>2.573717</td>
<td>128.32.37.50</td>
<td>128.32.244.190</td>
<td>HTTP</td>
<td>200</td>
<td>GET /my/ua/ HTTP/1.1</td>
</tr>
<tr>
<td>47</td>
<td>2.575775</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>66</td>
<td>80 = 5646 [ACK] Seq=1 Ack=3121 Win=421 Len=0 T RVA=1854455783 T Seq=1913937172</td>
</tr>
<tr>
<td>48</td>
<td>2.615046</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>66</td>
<td>80 = 5646 [ACK] Seq=1 Ack=3121 Win=421 Len=0 T RVA=1854455783 T Seq=1913937172</td>
</tr>
<tr>
<td>49</td>
<td>2.615407</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>66</td>
<td>80 = 5646 [ACK] Seq=1 Ack=3121 Win=421 Len=0 T RVA=1854455783 T Seq=1913937172</td>
</tr>
<tr>
<td>50</td>
<td>2.615459</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>66</td>
<td>80 = 5646 [ACK] Seq=1 Ack=3121 Win=421 Len=0 T RVA=1854455783 T Seq=1913937172</td>
</tr>
<tr>
<td>51</td>
<td>2.615583</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>66</td>
<td>80 = 5646 [ACK] Seq=1 Ack=3121 Win=421 Len=0 T RVA=1854455783 T Seq=1913937172</td>
</tr>
<tr>
<td>52</td>
<td>2.615588</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>66</td>
<td>80 = 5646 [ACK] Seq=1 Ack=3121 Win=421 Len=0 T RVA=1854455783 T Seq=1913937172</td>
</tr>
<tr>
<td>53</td>
<td>2.615591</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>66</td>
<td>80 = 5646 [ACK] Seq=1 Ack=3121 Win=421 Len=0 T RVA=1854455783 T Seq=1913937172</td>
</tr>
<tr>
<td>54</td>
<td>2.615594</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>66</td>
<td>80 = 5646 [ACK] Seq=1 Ack=3121 Win=421 Len=0 T RVA=1854455783 T Seq=1913937172</td>
</tr>
<tr>
<td>55</td>
<td>2.615604</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>66</td>
<td>80 = 5646 [ACK] Seq=1 Ack=3121 Win=421 Len=0 T RVA=1854455783 T Seq=1913937172</td>
</tr>
<tr>
<td>56</td>
<td>2.615606</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>66</td>
<td>80 = 5646 [ACK] Seq=1 Ack=3121 Win=421 Len=0 T RVA=1854455783 T Seq=1913937172</td>
</tr>
<tr>
<td>57</td>
<td>2.615608</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>66</td>
<td>80 = 5646 [ACK] Seq=1 Ack=3121 Win=421 Len=0 T RVA=1854455783 T Seq=1913937172</td>
</tr>
<tr>
<td>58</td>
<td>2.615611</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>1514</td>
<td>80 = 5646 [ACK] Seq=1 Ack=3121 Win=421 Len=0 T RVA=1854455783 T Seq=1913937172</td>
</tr>
<tr>
<td>59</td>
<td>2.615652</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>1514</td>
<td>5646 = 80 [ACK] Seq=3121 Ack=3121 Win=421 Len=1448 T RVA=1913937172 T Seq=1854455745 [TCP segment of a reassembled PDU]</td>
</tr>
<tr>
<td>60</td>
<td>2.615690</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>66</td>
<td>80 = 5646 [ACK] Seq=1 Ack=3121 Win=421 Len=0 T RVA=1913937172 T Seq=1854455745 [TCP segment of a reassembled PDU]</td>
</tr>
<tr>
<td>61</td>
<td>2.615868</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>66</td>
<td>80 = 5646 [ACK] Seq=1 Ack=3121 Win=421 Len=0 T RVA=1913937172 T Seq=1854455745 [TCP segment of a reassembled PDU]</td>
</tr>
<tr>
<td>62</td>
<td>2.615869</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>TCP</td>
<td>66</td>
<td>80 = 5646 [ACK] Seq=1 Ack=3121 Win=421 Len=0 T RVA=1913937172 T Seq=1854455745 [TCP segment of a reassembled PDU]</td>
</tr>
<tr>
<td>63</td>
<td>2.615870</td>
<td>128.32.244.190</td>
<td>128.32.37.50</td>
<td>HTTP</td>
<td>132</td>
<td>HTTP/1.1 200 OK (text/html)</td>
</tr>
</tbody>
</table>

### Flags
- #: Opt (PSH, ACK)
- Window: 421
- [Calculated window size: 421]
- [Window size scaling factors: 1 (unknown)]
- Checksum: 66f0b8 (Unverified)
- [Checksum Status: Unverified]
- Urgent Pointer: 8

### Options
- [TCP options: 12 bytes, No-Operation (NOP), No-Operation (NOP), Timestamps]

### Hypertext Transfer Protocol

```
HTTP/1.1 200 OK
Content-Length: 176
Content-Type: text/html

<html>
<head>
<title>Welcome to Apache!</title>
</head>
<body>
<h1>Welcome to Apache!</h1>
<p>This is the Apache HTTP server.</p>
</body>
</html>
```

---

**Thunderbolt Ethernet Slot: 0 - end of capture in progress.**
<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>63</td>
<td>21:56:59</td>
<td>128.32.32.440</td>
<td>128.32.32.440</td>
<td>TCP</td>
<td>1514</td>
<td>Seq:1321 Ack:1481 Win:1867 Len:0 Tvs:193937215 Tcpr:1854555745</td>
</tr>
<tr>
<td>64</td>
<td>21:56:59</td>
<td>128.32.32.440</td>
<td>128.32.32.440</td>
<td>HTTP</td>
<td>132</td>
<td>HTTP/1.1 200 OK</td>
</tr>
<tr>
<td>65</td>
<td>21:56:59</td>
<td>128.32.32.440</td>
<td>128.32.32.440</td>
<td>TCP</td>
<td>66</td>
<td>Seq:1321 Ack:1737 Win:1821 Len:0 Tvs:193937215 Tcpr:1854555745</td>
</tr>
<tr>
<td>68</td>
<td>21:56:59</td>
<td>128.32.32.440</td>
<td>128.32.32.440</td>
<td>TCP</td>
<td>66</td>
<td>Seq:1321 Ack:17443 Win:2408 Len:0 Tvs:193937215 Tcpr:1854555745</td>
</tr>
<tr>
<td>69</td>
<td>21:56:59</td>
<td>128.32.32.440</td>
<td>128.32.32.440</td>
<td>TCP</td>
<td>66</td>
<td>Seq:1321 Ack:17443 Win:2408 Len:0 Tvs:193937215 Tcpr:1854555745</td>
</tr>
<tr>
<td>70</td>
<td>21:56:59</td>
<td>128.32.32.440</td>
<td>128.32.32.440</td>
<td>TCP</td>
<td>66</td>
<td>Seq:1321 Ack:17443 Win:2408 Len:0 Tvs:193937215 Tcpr:1854555745</td>
</tr>
<tr>
<td>71</td>
<td>21:56:59</td>
<td>128.32.32.440</td>
<td>128.32.32.440</td>
<td>TCP</td>
<td>66</td>
<td>Seq:1321 Ack:17443 Win:2408 Len:0 Tvs:193937215 Tcpr:1854555745</td>
</tr>
<tr>
<td>72</td>
<td>21:56:59</td>
<td>128.32.32.440</td>
<td>128.32.32.440</td>
<td>TCP</td>
<td>66</td>
<td>Seq:1321 Ack:17443 Win:2408 Len:0 Tvs:193937215 Tcpr:1854555745</td>
</tr>
<tr>
<td>73</td>
<td>21:56:59</td>
<td>128.32.32.440</td>
<td>128.32.32.440</td>
<td>TCP</td>
<td>66</td>
<td>Seq:1321 Ack:17443 Win:2408 Len:0 Tvs:193937215 Tcpr:1854555745</td>
</tr>
<tr>
<td>74</td>
<td>21:56:59</td>
<td>128.32.32.440</td>
<td>128.32.32.440</td>
<td>TCP</td>
<td>66</td>
<td>Seq:1321 Ack:17443 Win:2408 Len:0 Tvs:193937215 Tcpr:1854555745</td>
</tr>
<tr>
<td>75</td>
<td>21:56:59</td>
<td>128.32.32.440</td>
<td>128.32.32.440</td>
<td>TCP</td>
<td>66</td>
<td>Seq:1321 Ack:17443 Win:2408 Len:0 Tvs:193937215 Tcpr:1854555745</td>
</tr>
<tr>
<td>76</td>
<td>21:56:59</td>
<td>128.32.32.440</td>
<td>128.32.32.440</td>
<td>TCP</td>
<td>66</td>
<td>Seq:1321 Ack:17443 Win:2408 Len:0 Tvs:193937215 Tcpr:1854555745</td>
</tr>
<tr>
<td>77</td>
<td>21:56:59</td>
<td>128.32.32.440</td>
<td>128.32.32.440</td>
<td>TCP</td>
<td>66</td>
<td>Seq:1321 Ack:17443 Win:2408 Len:0 Tvs:193937215 Tcpr:1854555745</td>
</tr>
<tr>
<td>78</td>
<td>21:56:59</td>
<td>128.32.32.440</td>
<td>128.32.32.440</td>
<td>TCP</td>
<td>66</td>
<td>Seq:1321 Ack:17443 Win:2408 Len:0 Tvs:193937215 Tcpr:1854555745</td>
</tr>
<tr>
<td>80</td>
<td>21:56:59</td>
<td>128.32.32.440</td>
<td>128.32.32.440</td>
<td>TCP</td>
<td>66</td>
<td>Seq:1321 Ack:17443 Win:2408 Len:0 Tvs:193937215 Tcpr:1854555745</td>
</tr>
</tbody>
</table>

**Internet Protocol Version 4, Src: 128.32.32.37.38, Dst: 128.32.32.440**

- **Source Port:** 65466
- **Destination Port:** 80
- **Sequence Number:** 0037
- **Header Length:** 32 bytes
- **Flags:** 0x80 (PSH, ACK)
- **Window:** 2048
- **Checksum:** 0x9c3c (unverified)
- **Urgent Pointer:** 0
- **Options:** 12 bytes, No-Operation (NOP), No-Operation (NOP), Timestamps
- **Timestamps:**

**Transmission Control Protocol, Src Port:** 65466, **Dest Port:** 80, **Seq:** 0037, **Ack:** 17443, **Len:** 1904

- **Type:** IPv4 (0x0008)
- **Internet Protocol Version:** 4
- **Source Port:** 65466
- **Destination Port:** 80
- **Sequence Number:** 0037
- **Header Length:** 32 bytes
- **Flags:** 0x80 (PSH, ACK)
- **Window:** 2048
- **Checksum:** 0x9c3c (unverified)
- **Urgent Pointer:** 0
- **Options:** 12 bytes, No-Operation (NOP), No-Operation (NOP), Timestamps
- **Timestamps:**
Questions?