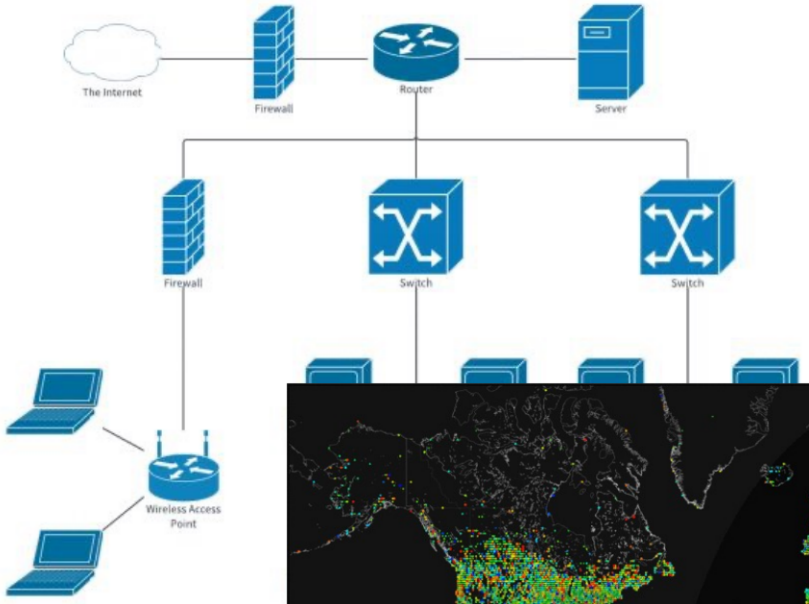
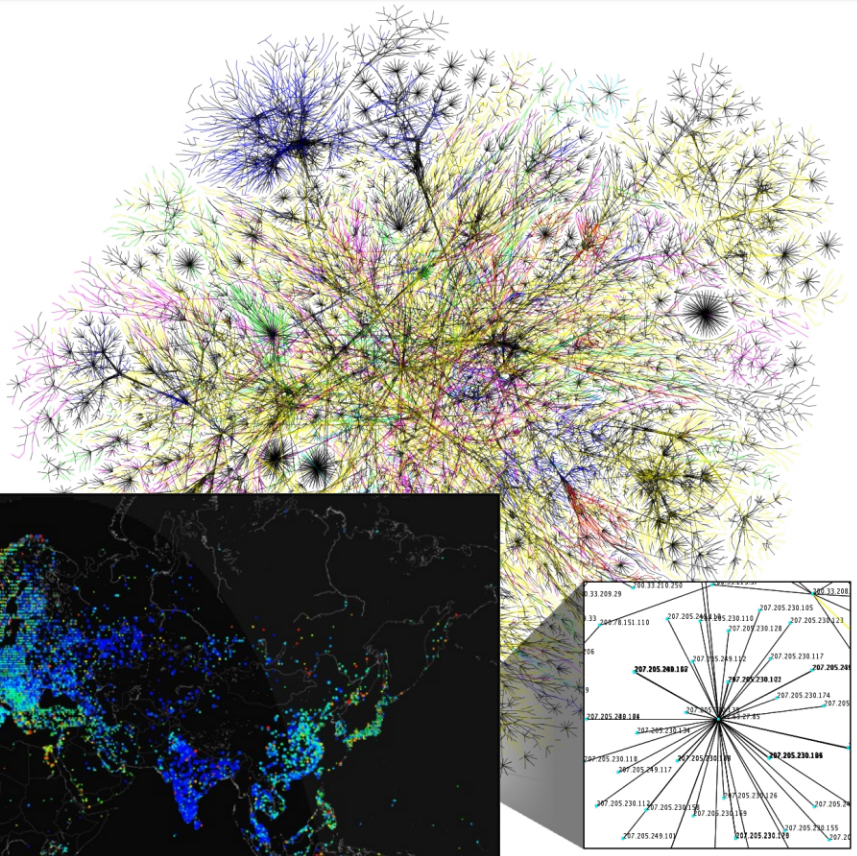
Computer and Network Security

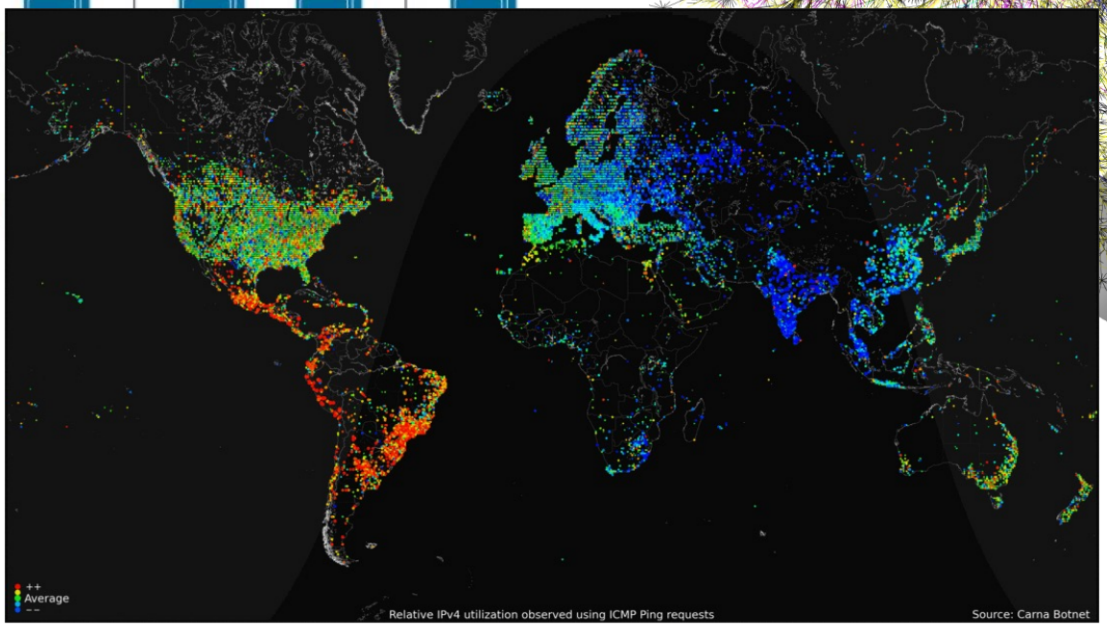
Lecture 16: Network Background/Attacks

COMP-5370/6370
Fall 2025

The Internet is Complicated



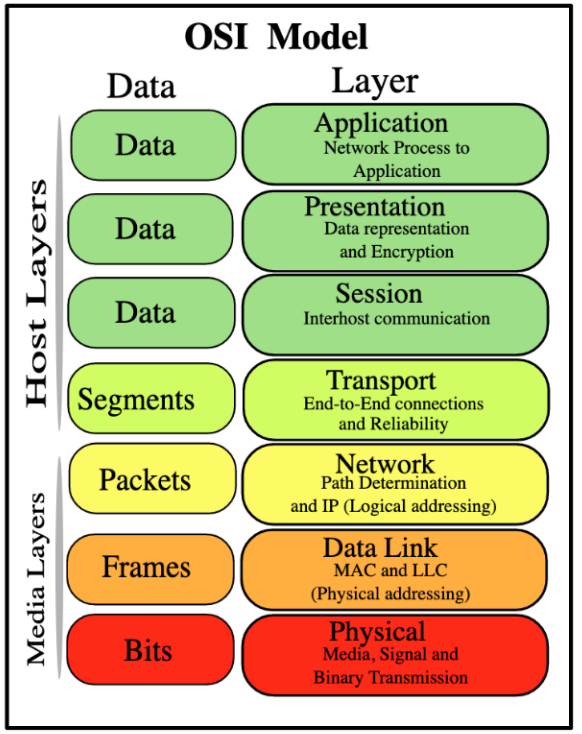




Theoretical OSI Model

The **OSI Model** was the original attempt at standardizing and partitioning requirements of communicating.

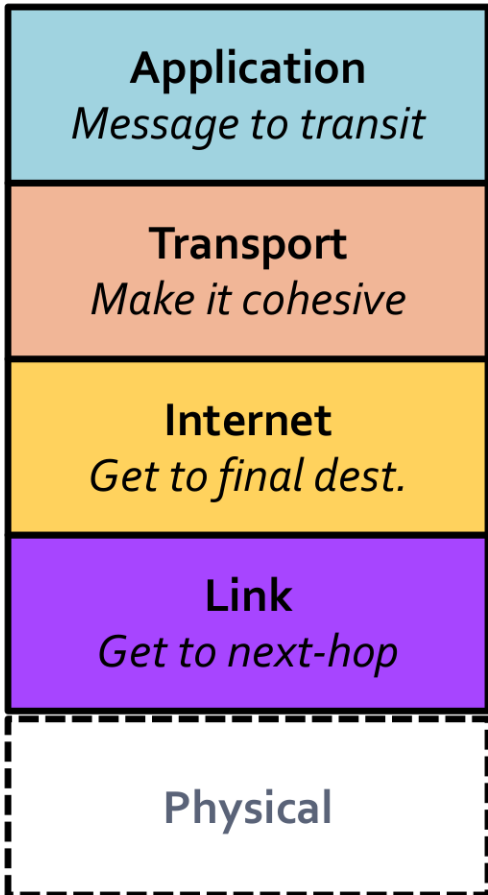
- Seven "*independent*" layers
- Replace layer X and others remain unchanged
- Formalized in the 1970s



TCP/IP Model

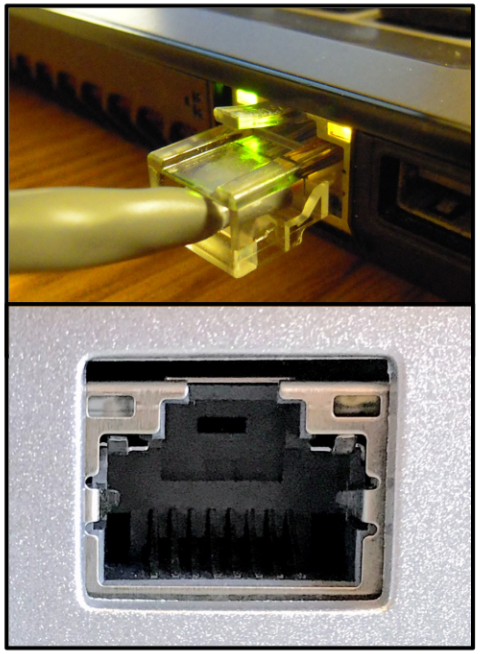
The **TCP/IP Model** is way of thinking about and conceptualizing the various protocols used in network communications.

- Reduced "OSI Model"
- Specifics differ greatly based on the source, time, and writer
- Is **NOT** a perfect representation of the real-world



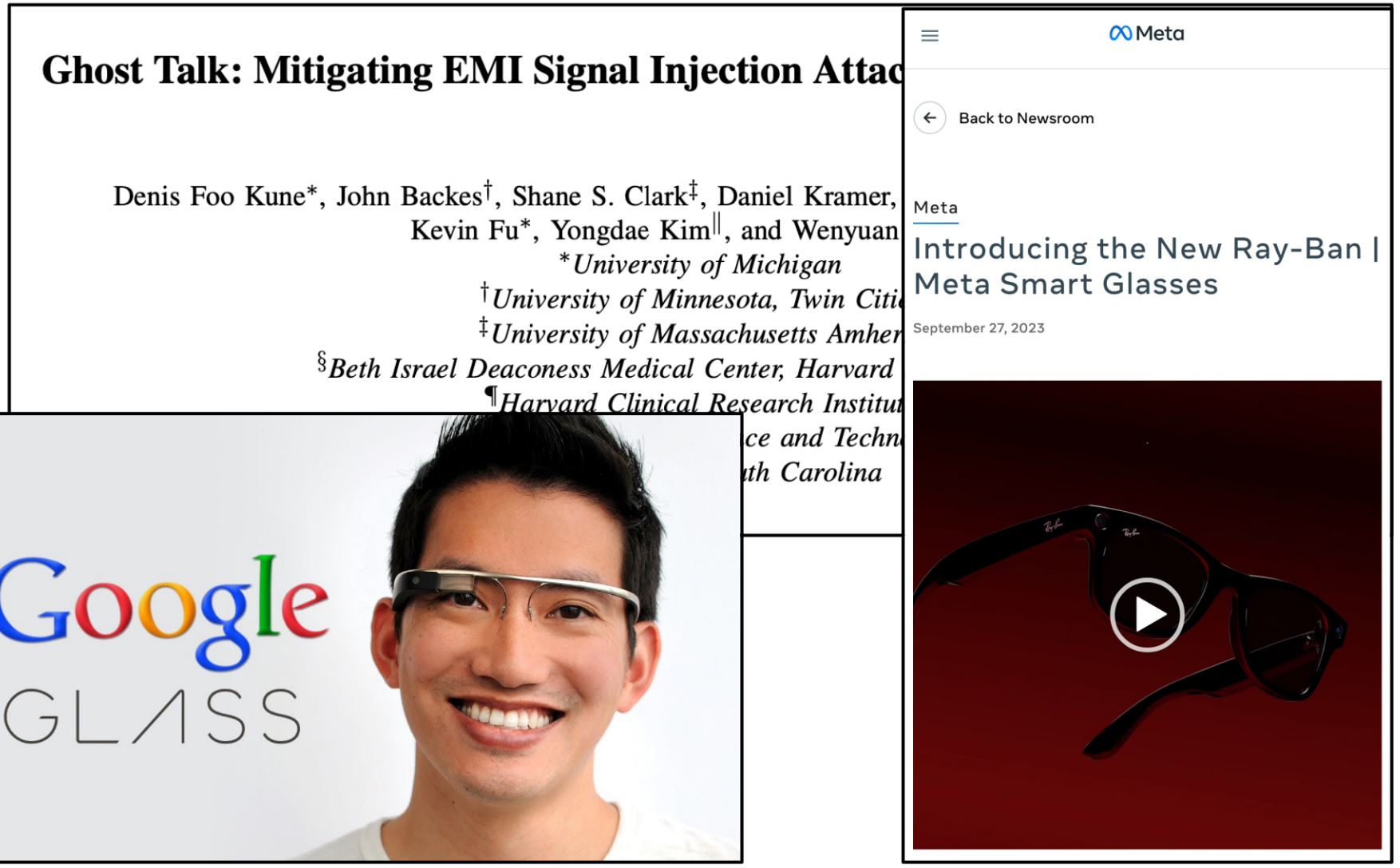
Physical Layer

The **physical layer** is the actual encoding mechanism used to represent bits in a physical form.



- How data is turned into bits
  - AM/FM, QPSK, PWM, …
- How bits are sent/received
  - RF, electrical impulse, light, …
- Largely EE-/ECE-world not CS

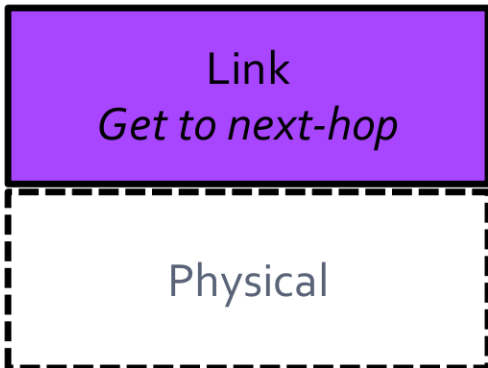
Physical


Ghost Talk: Mitigating EMI Signal Injection Attac
Denis Foo Kune*, John Backes†, Shane S. Clark‡, Daniel Kramer,
Kevin Fu*, Yongdae Kim‖, and Wenyuan
*University of Michigan
†University of Minnesota, Twin Citi
‡University of Massachusetts Amher
§Beth Israel Deaconess Medical Center, Harvard
¶Harvard Clinical Research Institut
ce and Techn
th Carolina
Google
GLASS
Meta
Back to Newsroom
Meta
Introducing the New Ray-Ban | Meta Smart Glasses
September 27, 2023

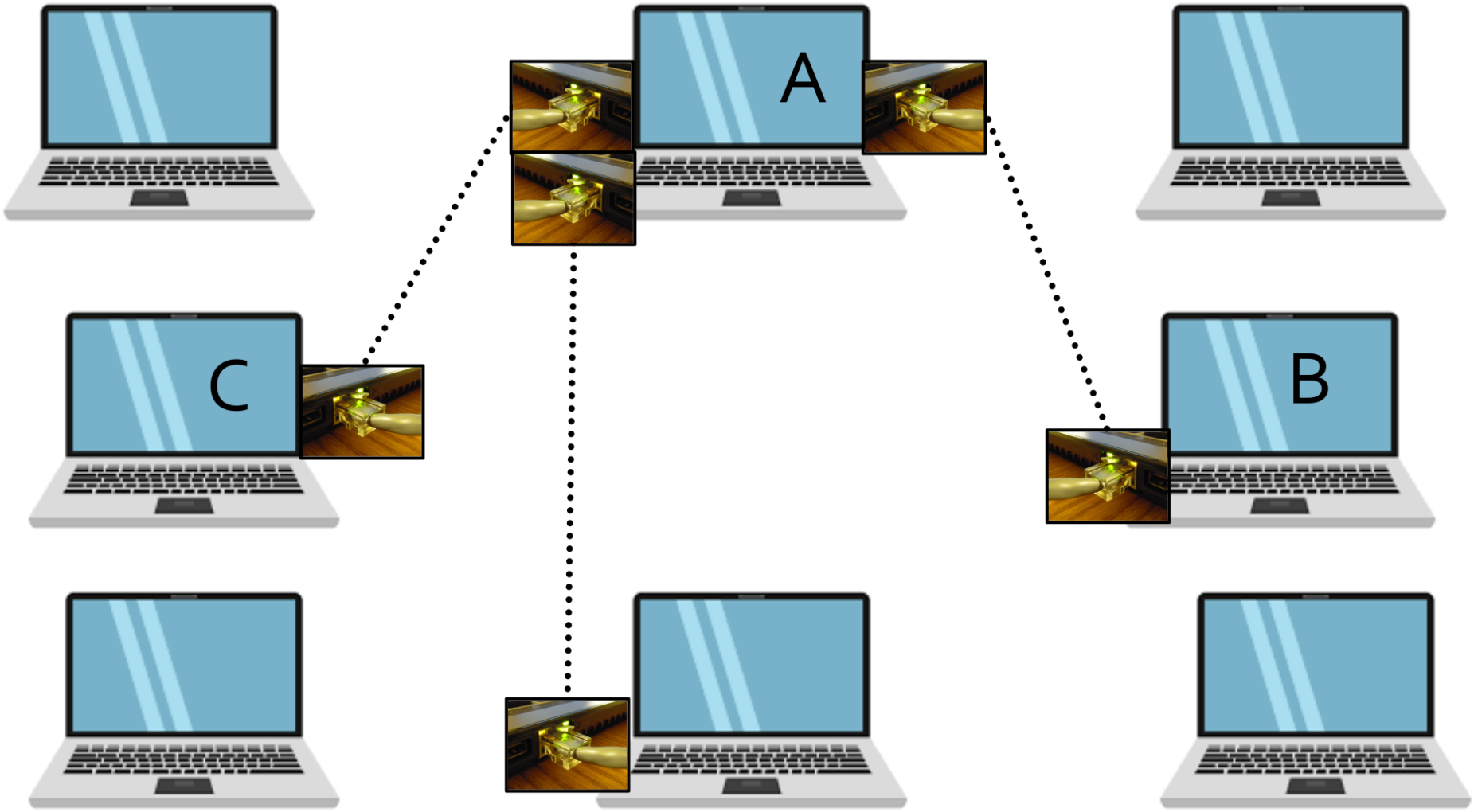
Link Layer

The **link layer** is responsible for addressing and transiting between endpoints on the same Local Area Network (LAN).

- Usually a relatively small physical distance
  - A room, a group of rooms, a floor, etc
- Very useful when bootstrapping to higher-level protocols

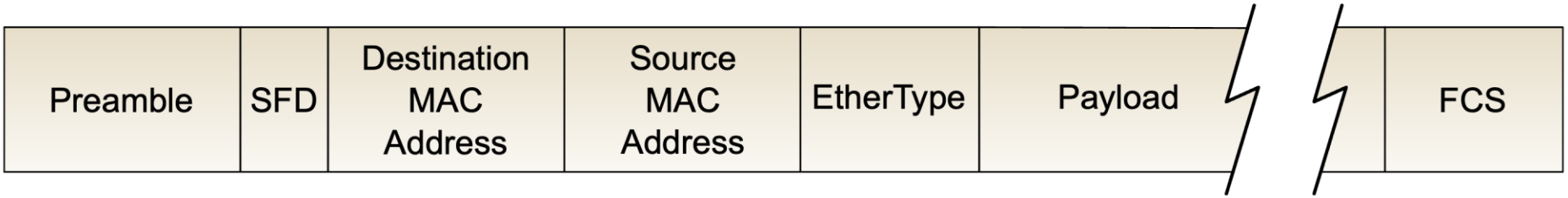


Naïve Link Layer Data Flow



Ethernet Protocol

- Media Access Control (MAC) addresses
  - DE:AD:BE:EF:4D:AD
  - **“MAC address” != “Cryptographic MAC”**
- Must be “locally unique” addresses
  - 3-byte manufacturer + 3-byte device ID
  - Are **NOT** globally unique

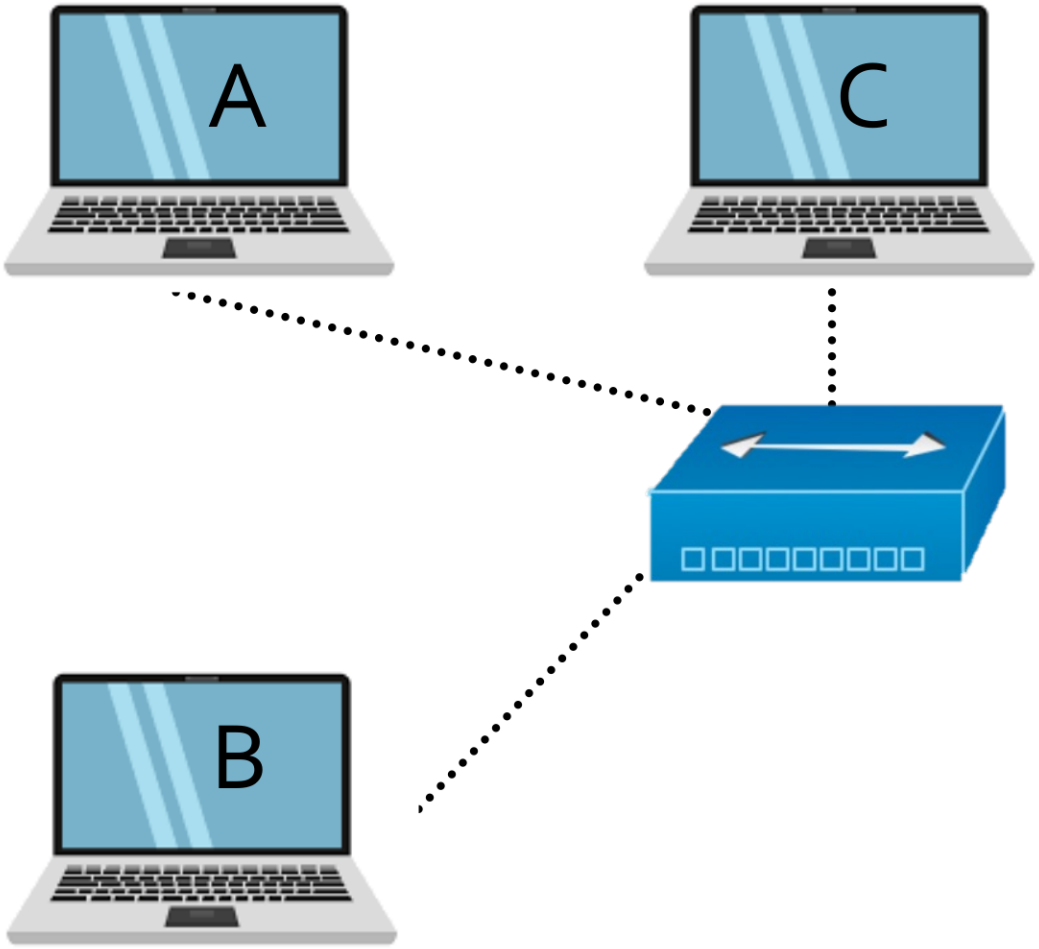


Link-Layer Data Flow

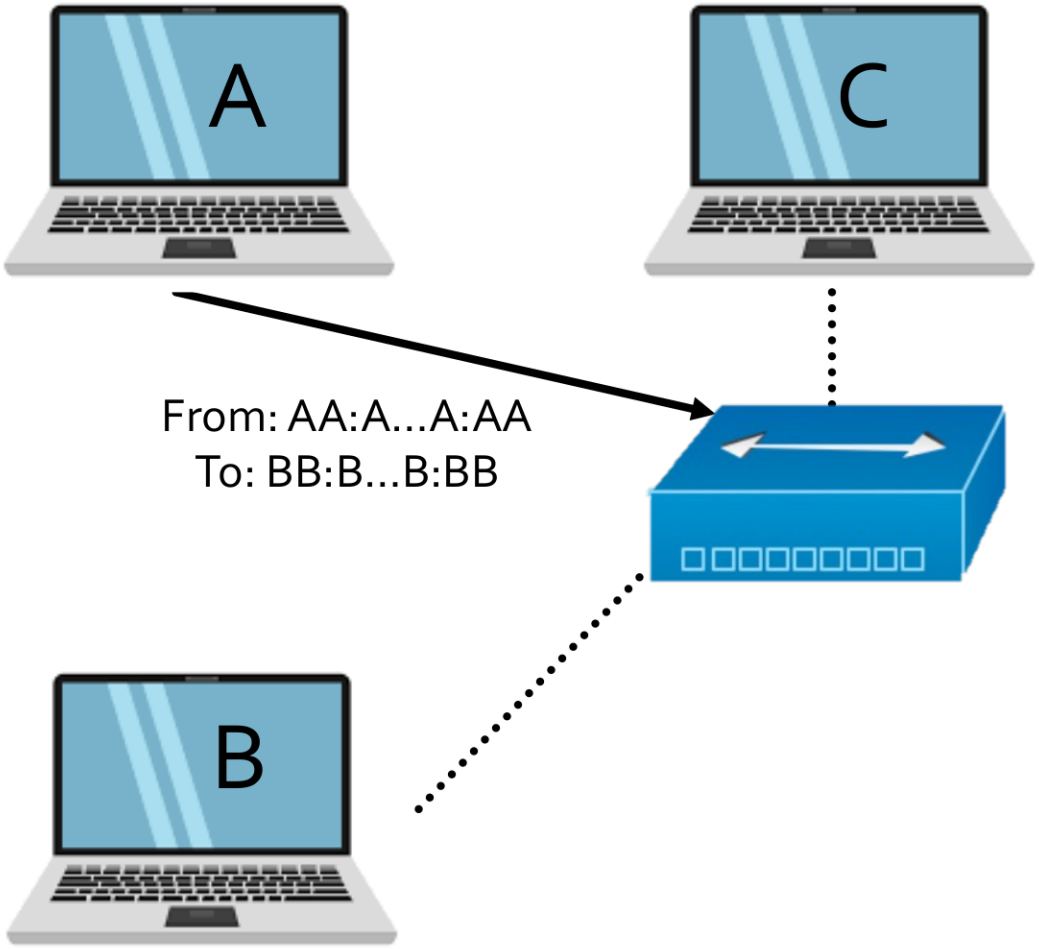
Ethernet Hub

- “packet repeater”

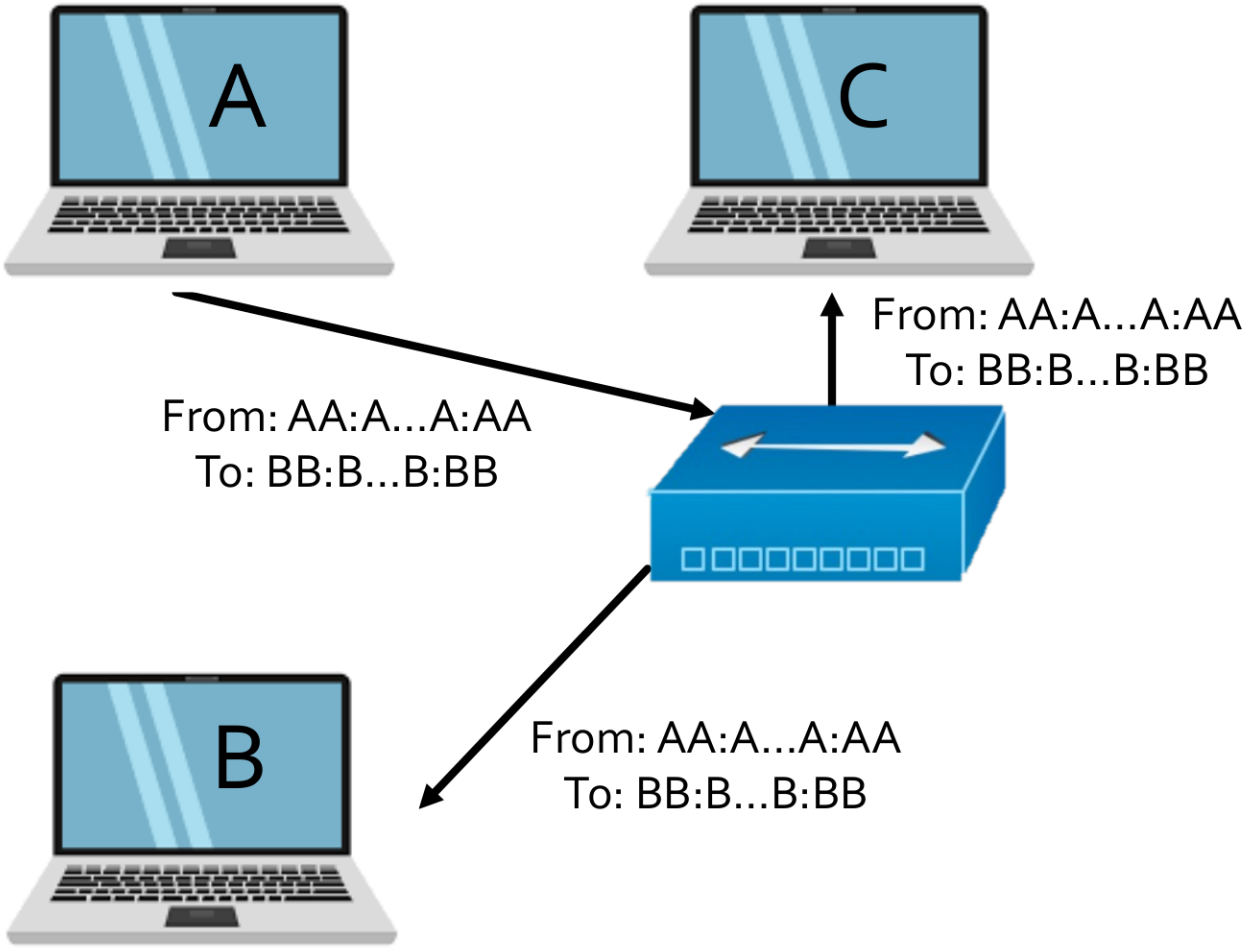
Link Layer Data Flow (hub)



Link Layer Data Flow (hub)



Link Layer Data Flow (hub)



Link Layer Data Flow

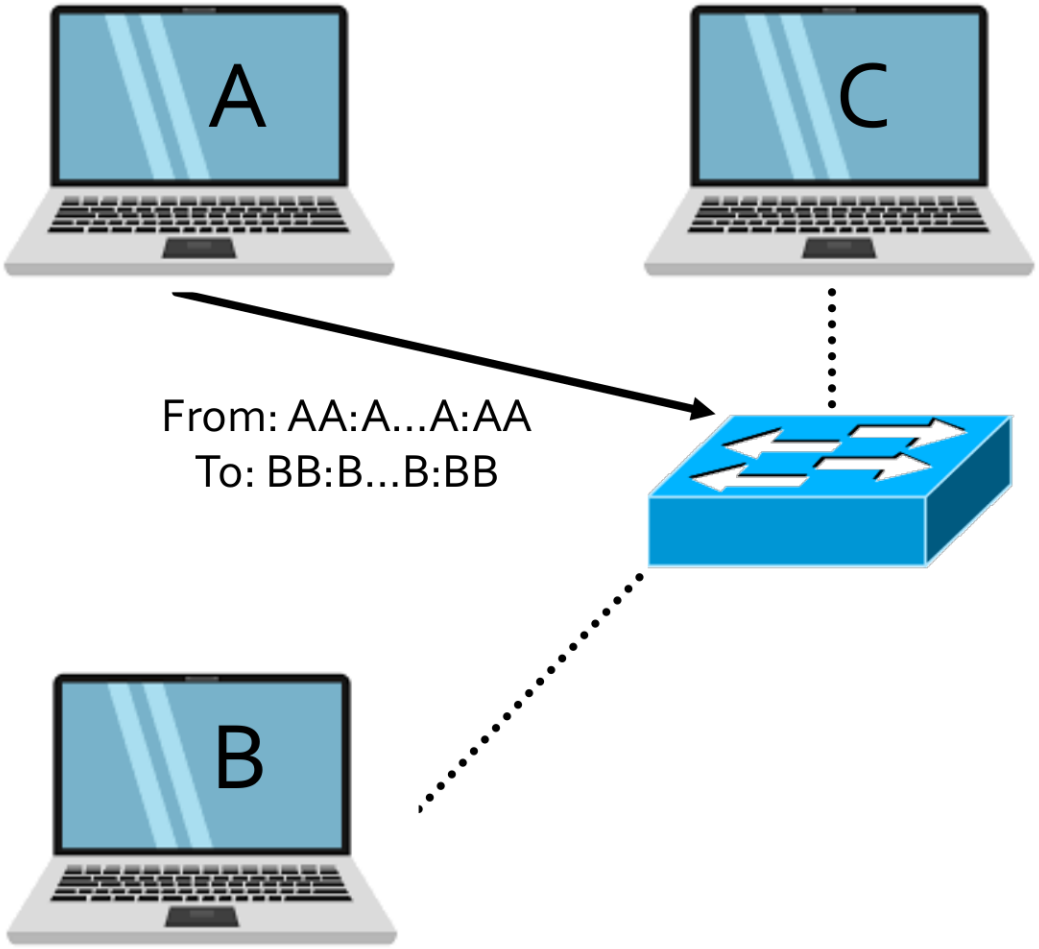
Ethernet Hub

- "packet repeater"
- Packet-In HW port 1
- Packet-Out HW ports 2, 3, 4

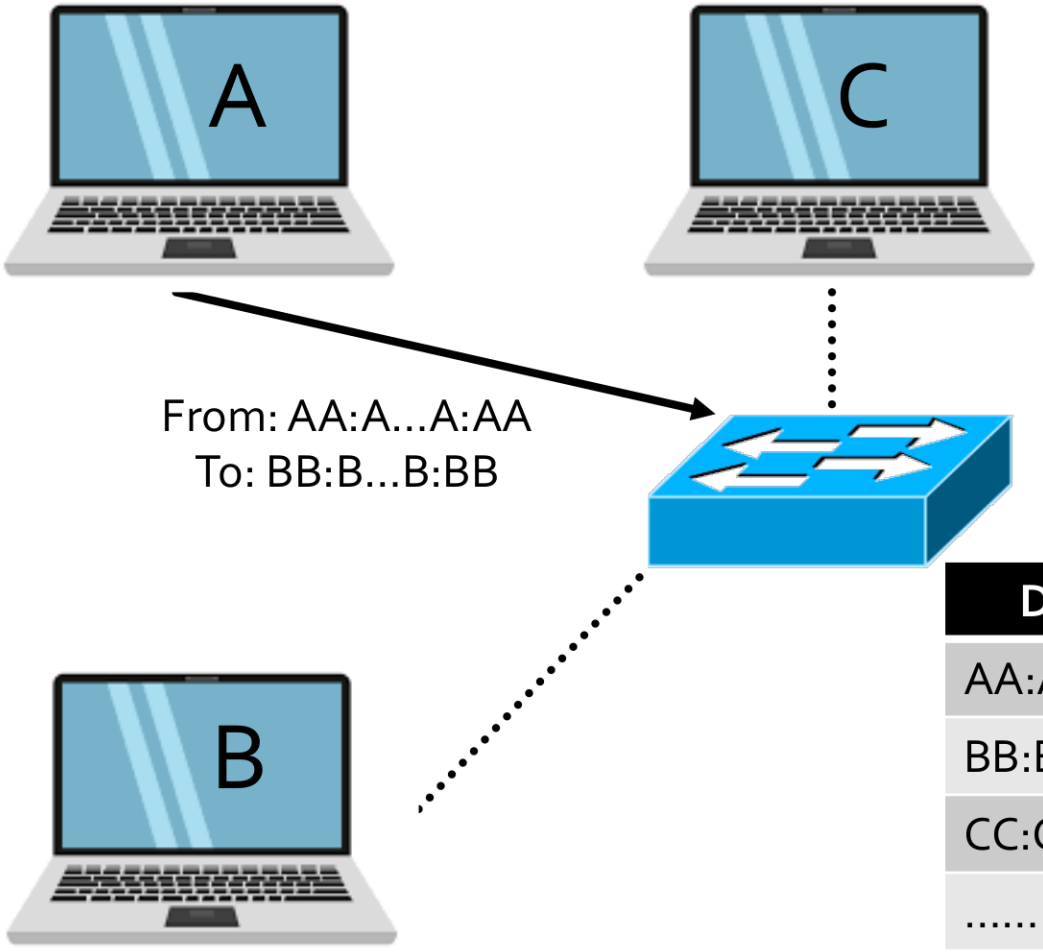
Ethernet Switch

- "packet dispatcher"

Link Layer Switching

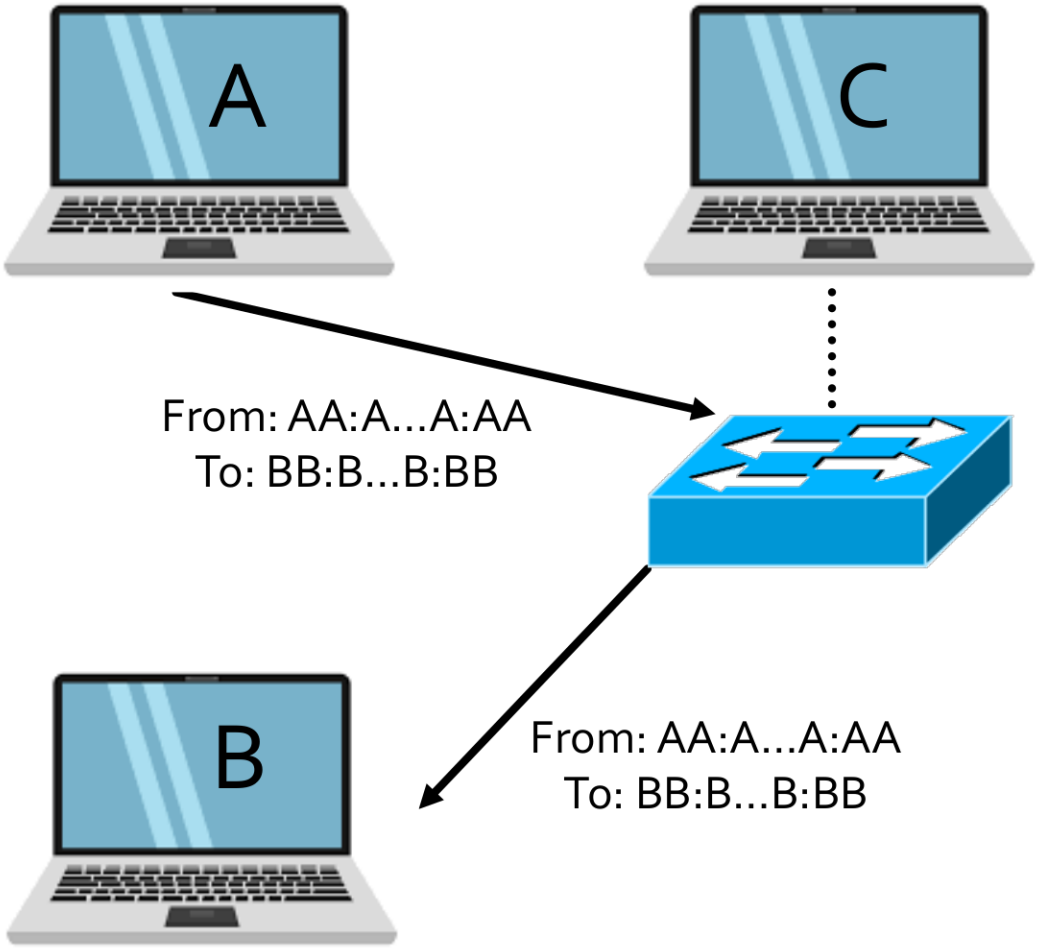


Link Layer Switching



Dest MAC	Interface Port
AA:A…A:AA	1
BB:B…B:BB	2
CC:C…C:CC	3
…….	…….

Link Layer Switching



Link-Layer Packet Passing

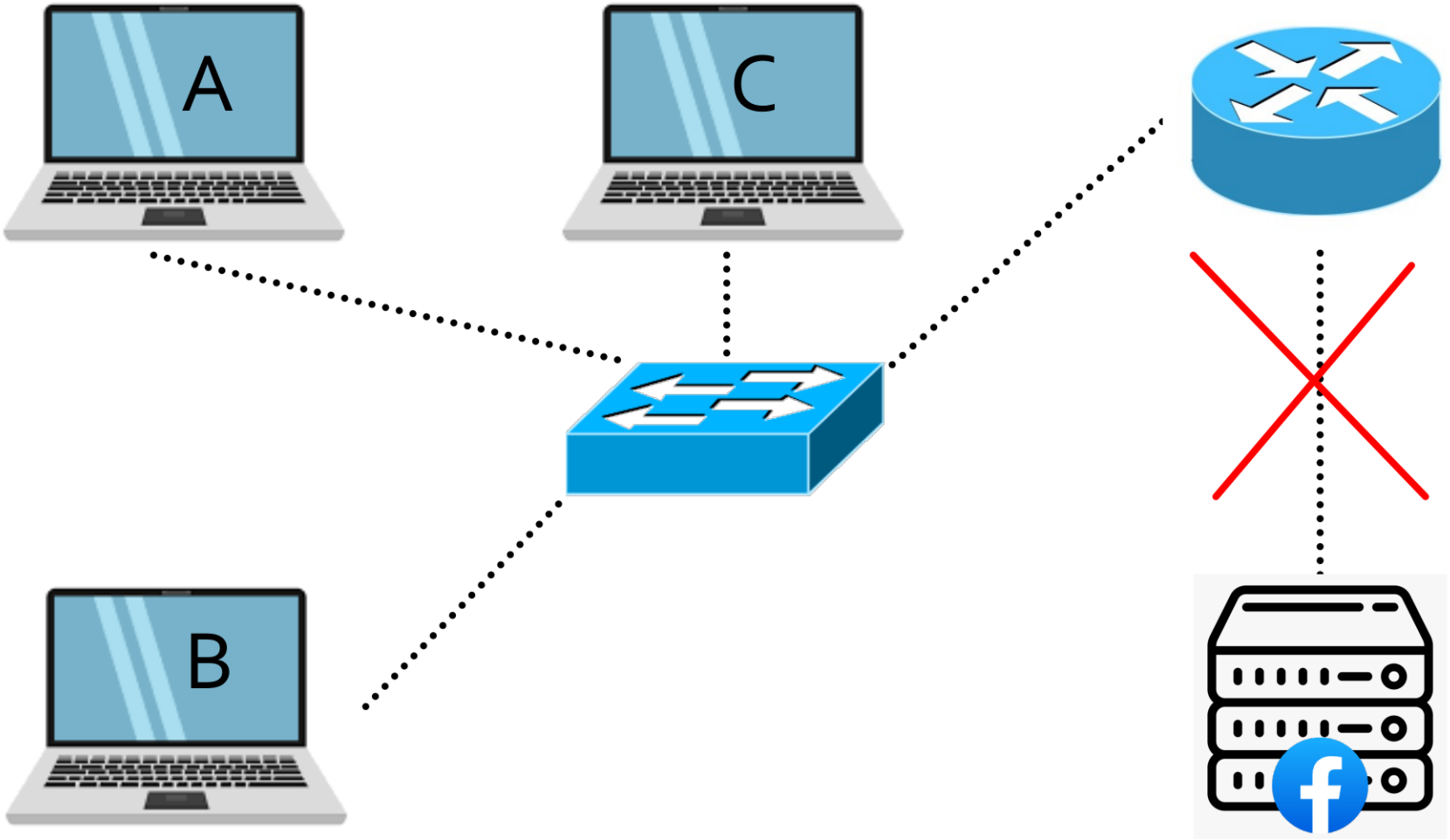
Ethernet Hub

- "packet repeater"
- Packet-In HW port 1
- Packet-Out HW ports 2, 3, 4

Ethernet Switch

- "packet dispatcher"
- Packet-In HW port 1
- Lookup dest MAC in MAC → port table
- Send to specific port

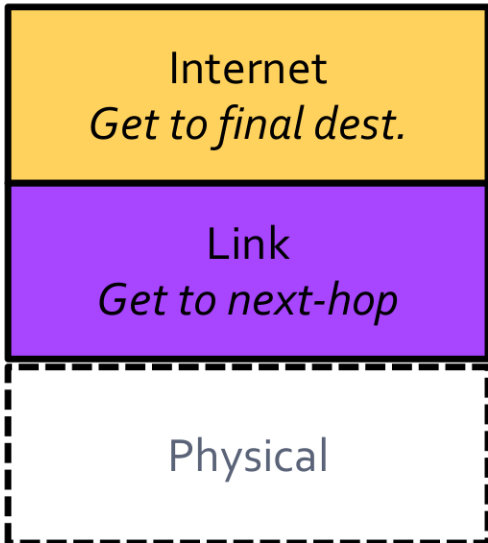
Link Layer Switching



Internet Layer

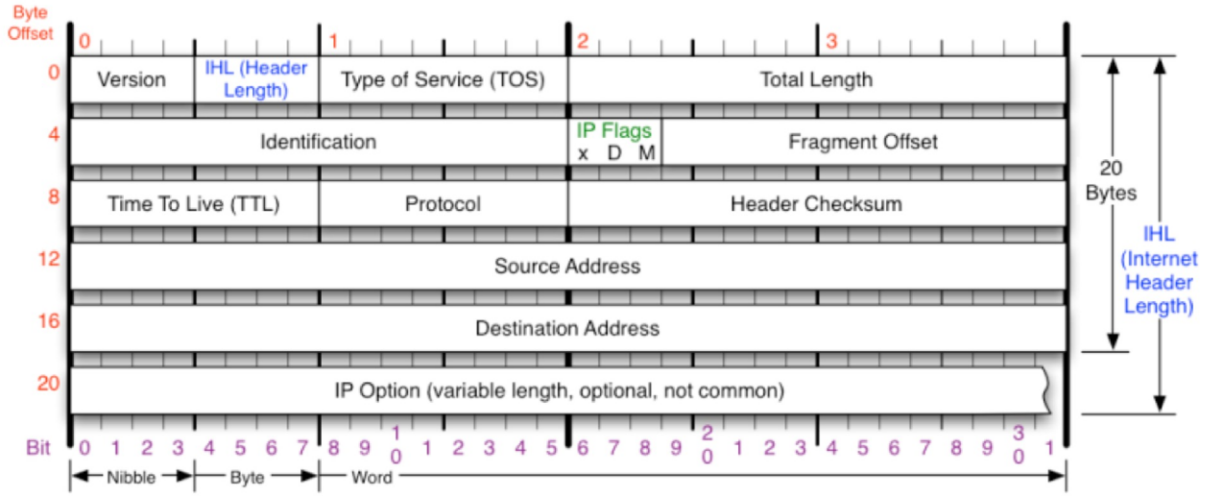
The **Internet layer** is responsible for addressing and transiting between endpoints on *different* LANs connected via a Wide Area Network (WAN).

- Acts as a LAN interconnect
- Requires a shared addressing and encoding scheme



Internet Protocol v4

- Commonly used for client-side addressing
- 4-byte address (~4 billion total)
  - 192.168.1.30, 1.1.1.1, 130.160.0.54, …
- Networks often use “CIDR notation”
  - `1.1.1.0/24 = mask 255.255.255.0 = 1.1.1.0–1.1.1.255`



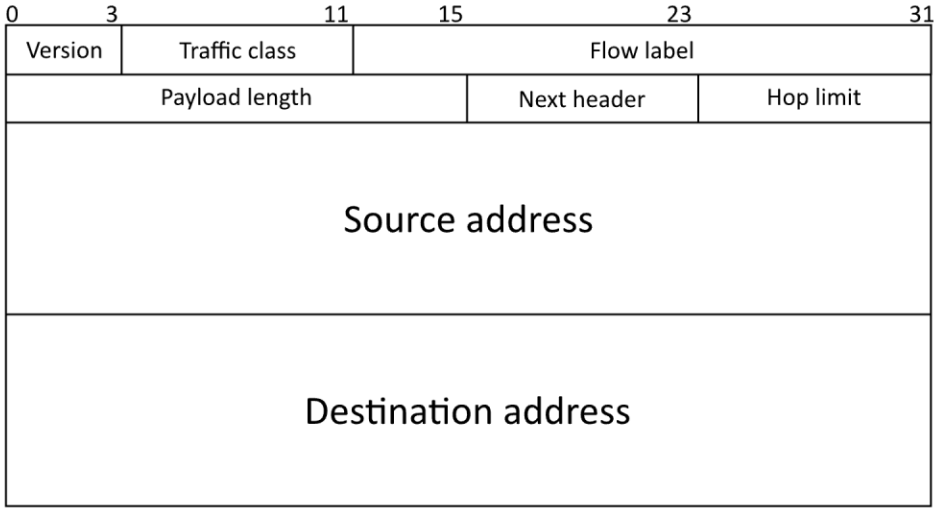
Reserved IPv4 Networks

There are network IP ranges that are "*special*" and should/can only be used for specific purposes.

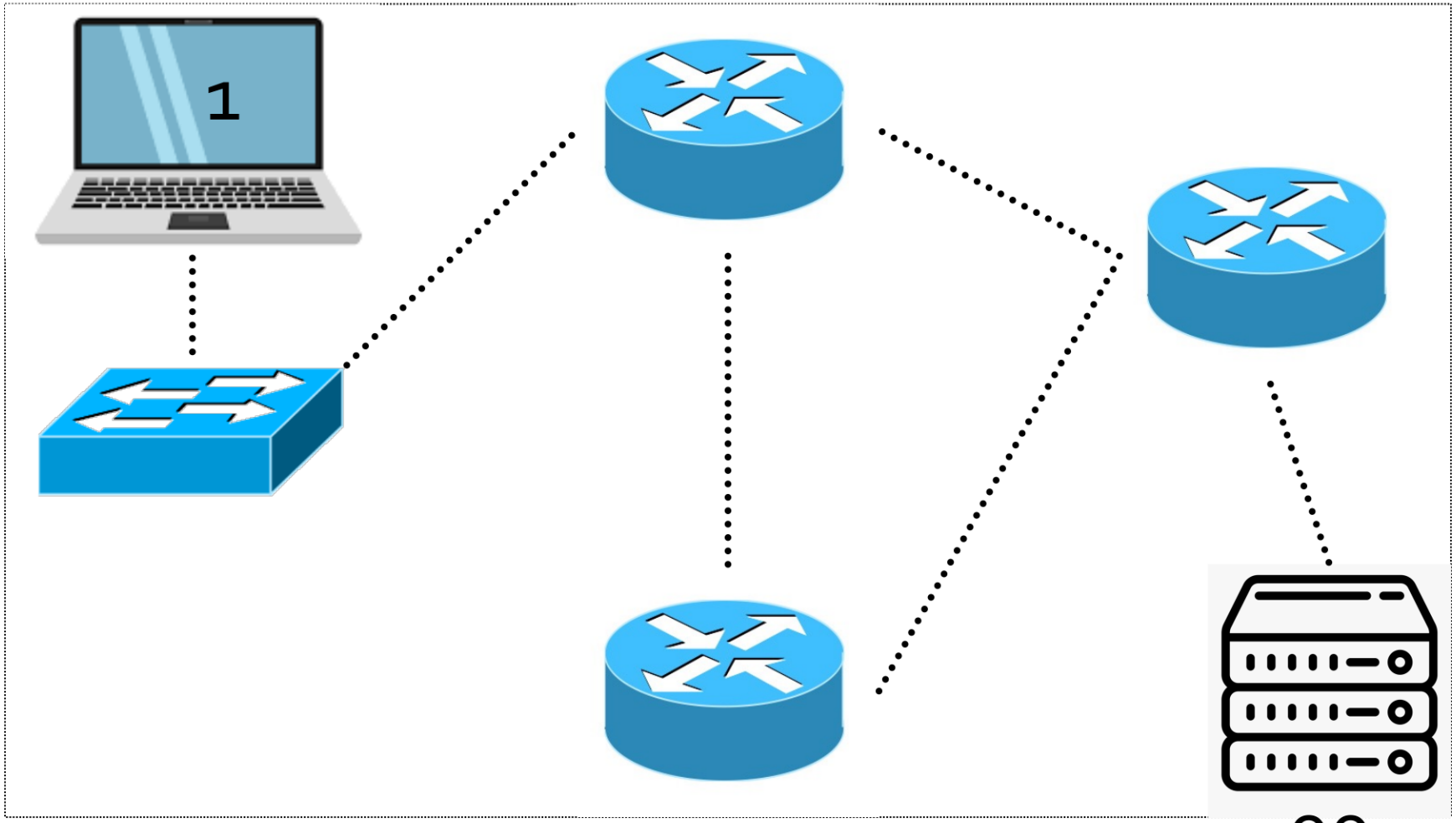
- Loopback: 127.0.0.0/8
- Broadcast: 255.255.255.255/32
- Private (internal routing only)
  - 10.0.0.0/8
  - 172.16.0.0/16
  - 192.168.0.0/16

Internet Protocol v6

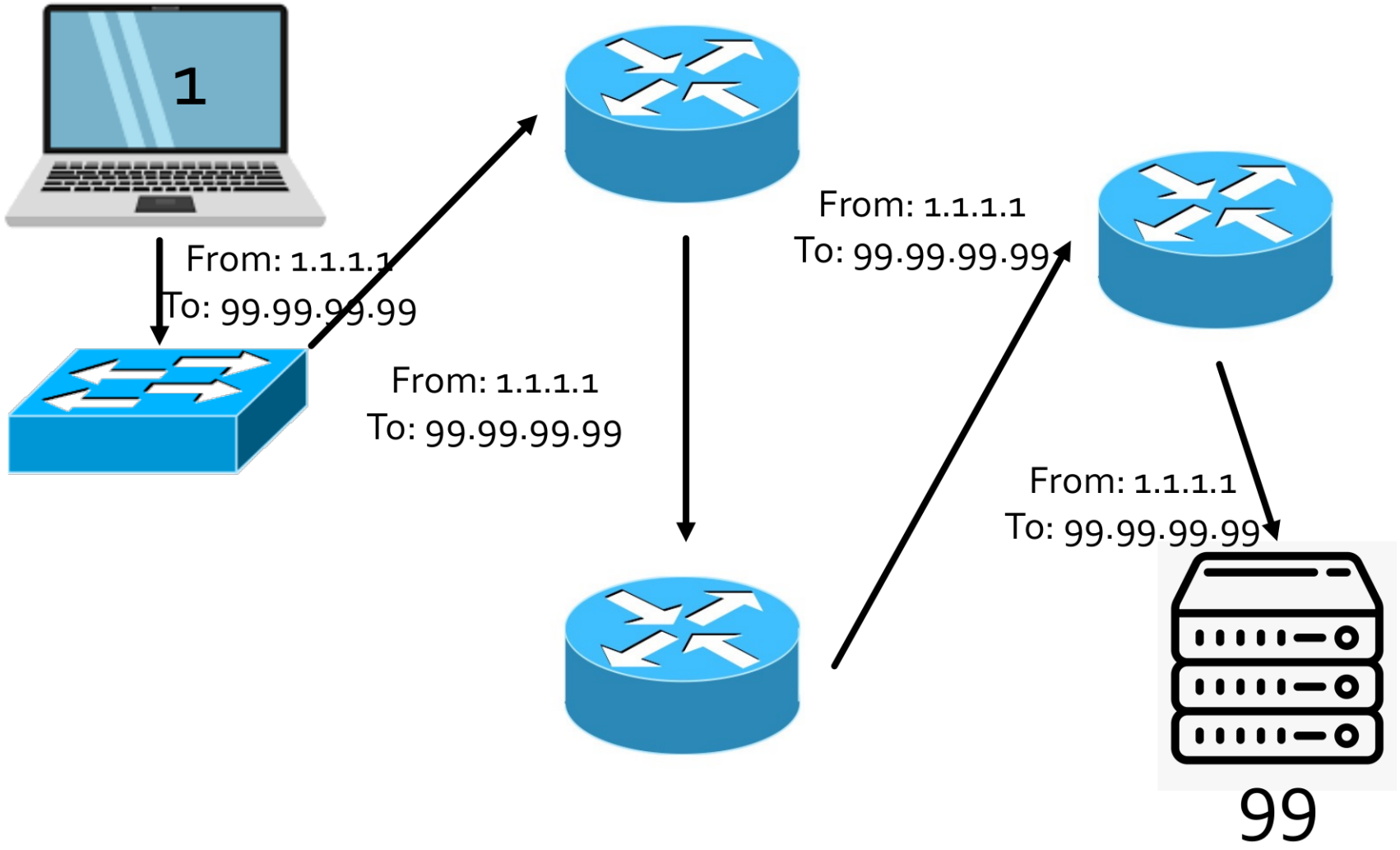
- Sometimes used for server-side addressing
- 16-byte address (2^{64} total == ~18 quintillion)
  - 2001:0db8:0000:0000:0000:ff00:0042:8329
- Will be the standard in ~~1999~~, ~~2004~~, ~~2008~~, ~~2013~~, ~~2016~~, ~~2017~~, ~~2020~~, 2030?



Internet Layer Data Flow



Internet Layer Data Flow



Internet Layer Data Flow

From: 1.1.1.1
To: 99.99.99.99

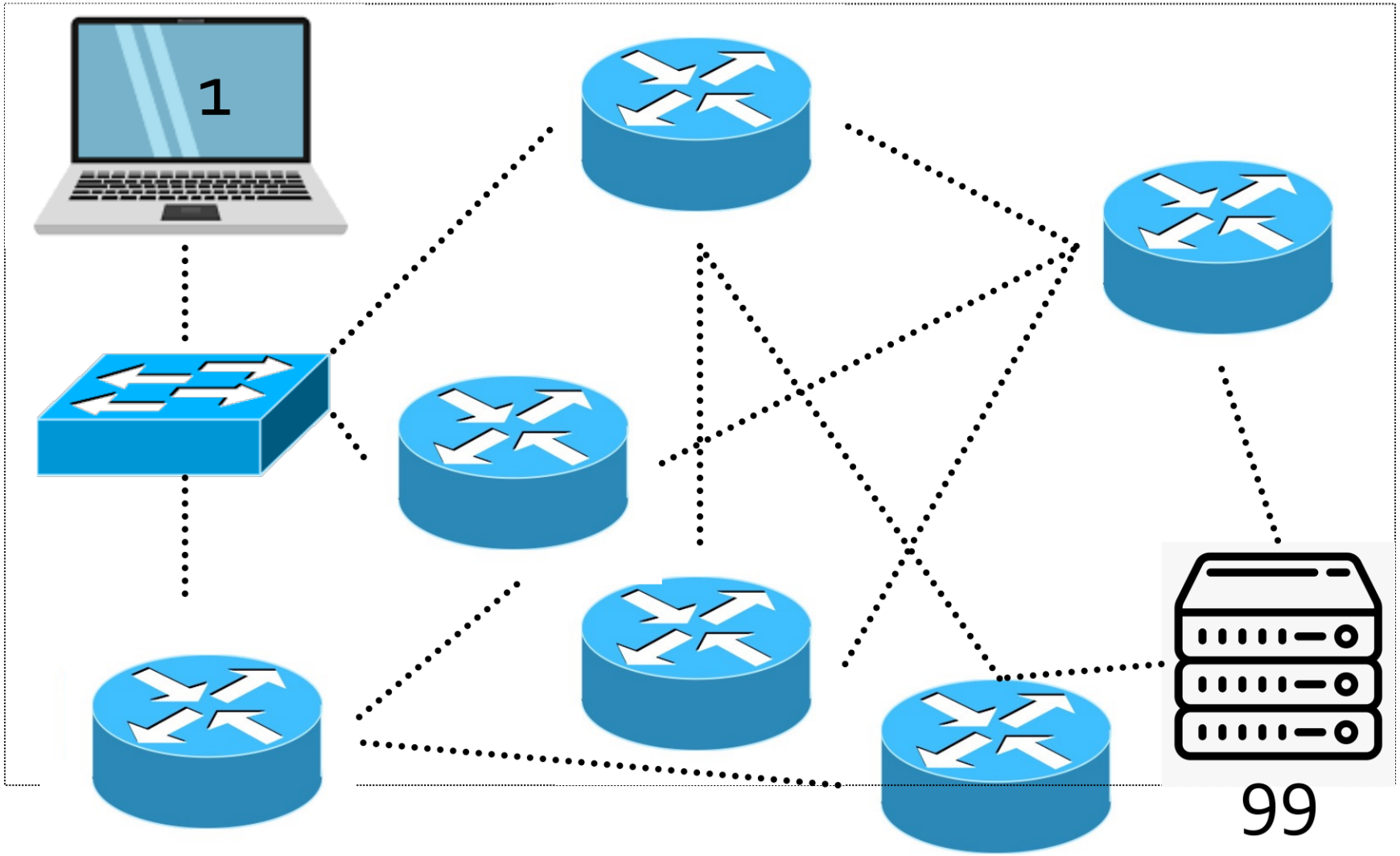
From: 1.1.1.1
To: 99.99.99.99

From: 1.1.1.1
To: 99.99.99.99

From: 1.1.1.1
To: 99.99.99.99

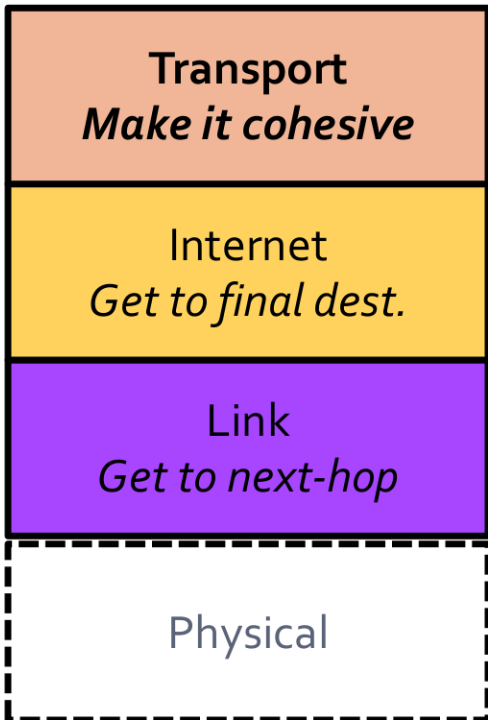
Routing policies are complicated

Internet Layer Data Flow



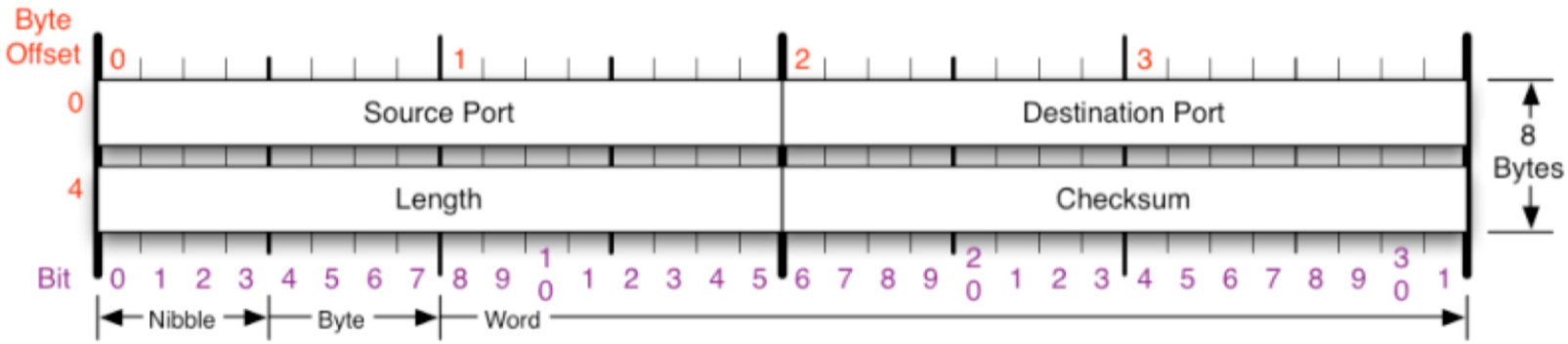
Transport Layer

The **Transport layer** is responsible for ensuring that the data is processed in an orderly and complete manner.



User Datagram Protocol (UDP)

- **Connectionless** protocol
- Used when:
  - Dropped packets are OK or recovery to be handled at the application layer



UDP Data Flow

From: Time Client
To: Time Server
Msg: What time is it?

From: Time Server
To: Time Client
Msg: It's 2pm.

UDP Data Flow (packet loss)

From: Time Client
To: Time Server
Msg: What time is it?

From: Time Server
To: Time Client
Msg: It's 2pm.

From: Time Client
To: Time Server
Msg: What time is it?

From: Time Server
To: Time Client
Msg: It's 2pm.

UDP Data Flow (packet loss)

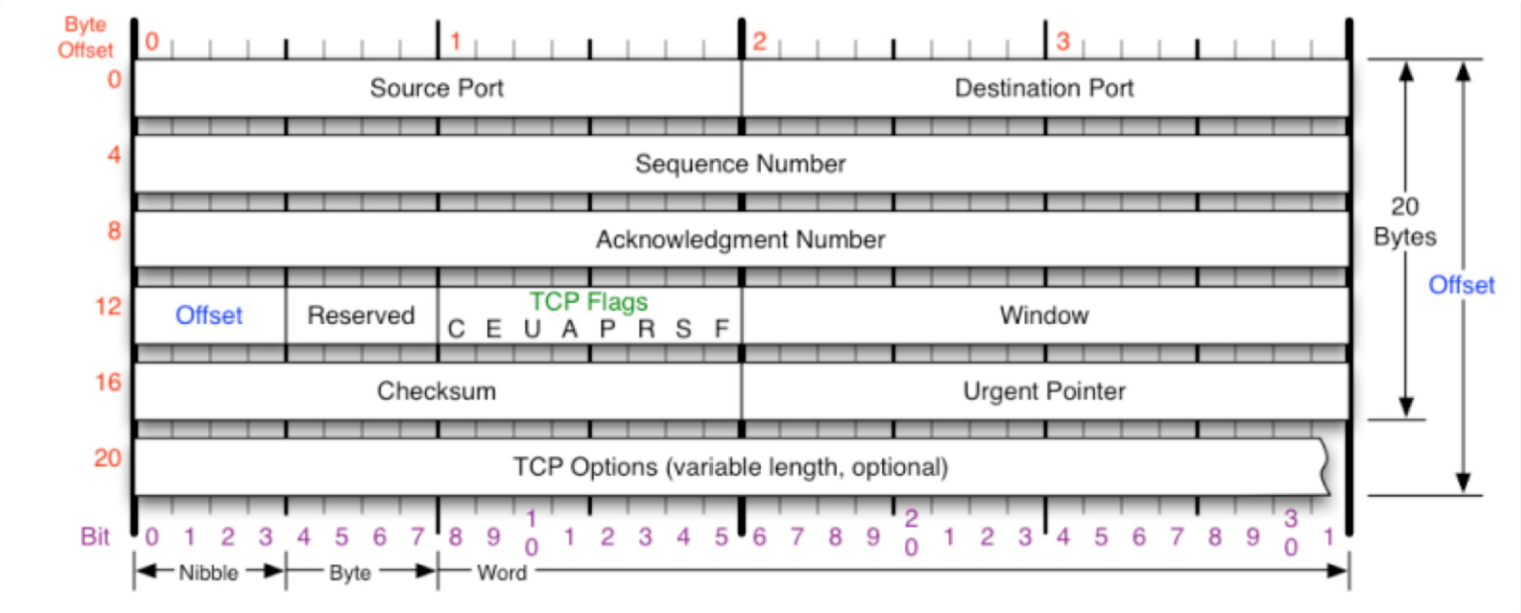
From: Time Client
To: Time Server
Msg: What time is it?

From: Time Client
To: Time Server
Msg: What time is it?

From: Time Server
To: Time Client
Msg: It's 2pm.

Transmission Control Protocol (TCP)

- **Connection-oriented** protocol
- Usually the default for communications
- Handles orderly bit stream details
  - Dropped packets, congestion control, etc



Three-Way TCP Handshake

From: Web Client
To: Web Server **SYN**
Msg: You there?

From: Web Server
To: Web Client **SYN-ACK**
Msg: Yeah

From: Web Client
To: Web Server **ACK-ACK**
Msg: OK, let's talk.

--- BEGIN CONTENT ---

TCP Acknowledgements

From: Web Server
To: Web Client
Msg: Here's ½ a picture

From: Web Client
To: Web Server
Msg: Got it

From: Web Server
To: Web Client
Msg: Here's the other half

From: Web Client
To: Web Server
Msg: Got it

TCP Batch Acknowledgements

From: Web Server
To: Web Client
Msg: Here's ½ a picture

From: Web Server
To: Web Client
Msg: Here's the other half

From: Web Client
To: Web Server
Msg: Got both of them

TCP Handling Out-of-Order/ Dropped Packets

From: Web Server
To: Web Client
Msg: Here's ½ a picture

From: Web Server
To: Web Client
Msg: Here's the 2nd half

From: Web Client
To: Web Server
Msg: Got 2nd half only

From: Web Server
To: Web Client
Msg: Here's the 1st half

Transport Layer Addressing

Application ports are used to address packets to *applications* running on device.

- Are a SW “port” not a HW “port”
- Often implicit but can be explicit
  - “Auburn website” == 131.204.138.170:80
  - Google’s “Honest” DNS == 8.8.8.8:53

Three-Way TCP Handshake

From: Web Client:75839
To: Web Server:80 **SYN**
Msg: You there?

From: Web Server:80
To: Web Client:75839 **SYN-ACK**
Msg: Yeah

From: Web Client:75839
To: Web Server:80 **ACK-ACK**
Msg: OK, let's talk.

Transport Layer Addressing

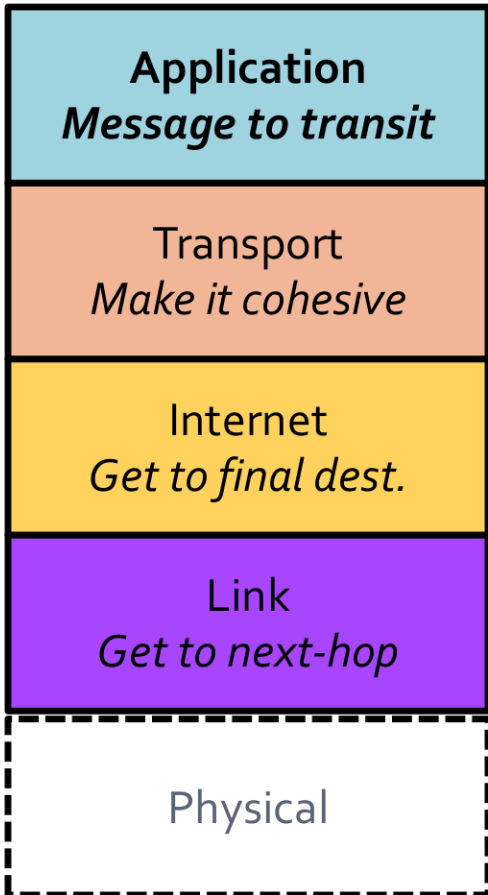
Application ports are used to address packets to *applications* running on device.

- Are a SW “port” not a HW “port”
- Often implicit but can be explicit
  - “Auburn website” == 131.204.138.170:80
  - Google’s “Honest” DNS == 8.8.8.8:53
- Servers often use canonical ports
  - TCP/80 == HTTP, TCP/443 == HTTPS
  - UDP/53 == DNS, UDP/123 == NTP

Application Layer

The **application layer** is the highest-layer protocol and handles the logical interactions between endpoints.

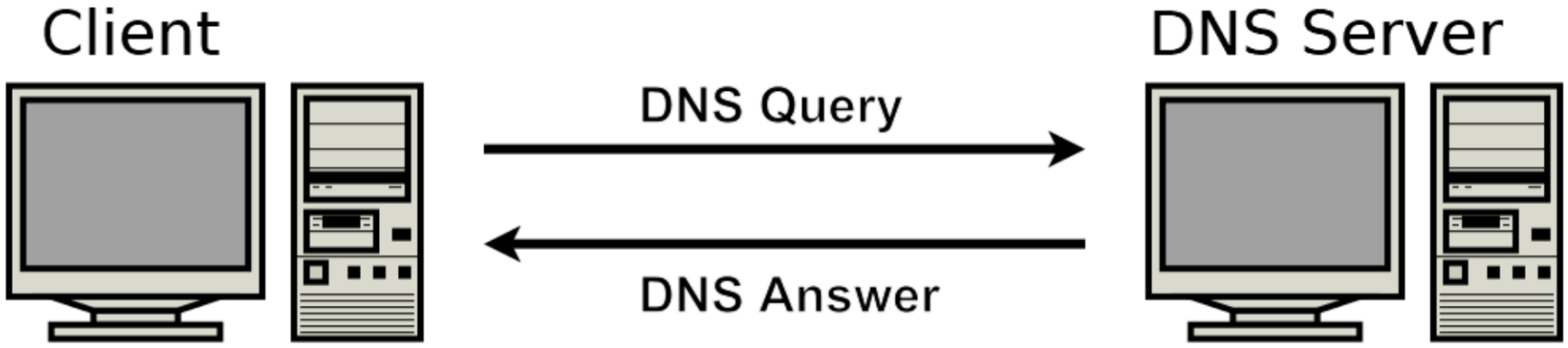
- Most well-known protocols
  - DNS, HTTP, SMTP, etc



DNS Protocol

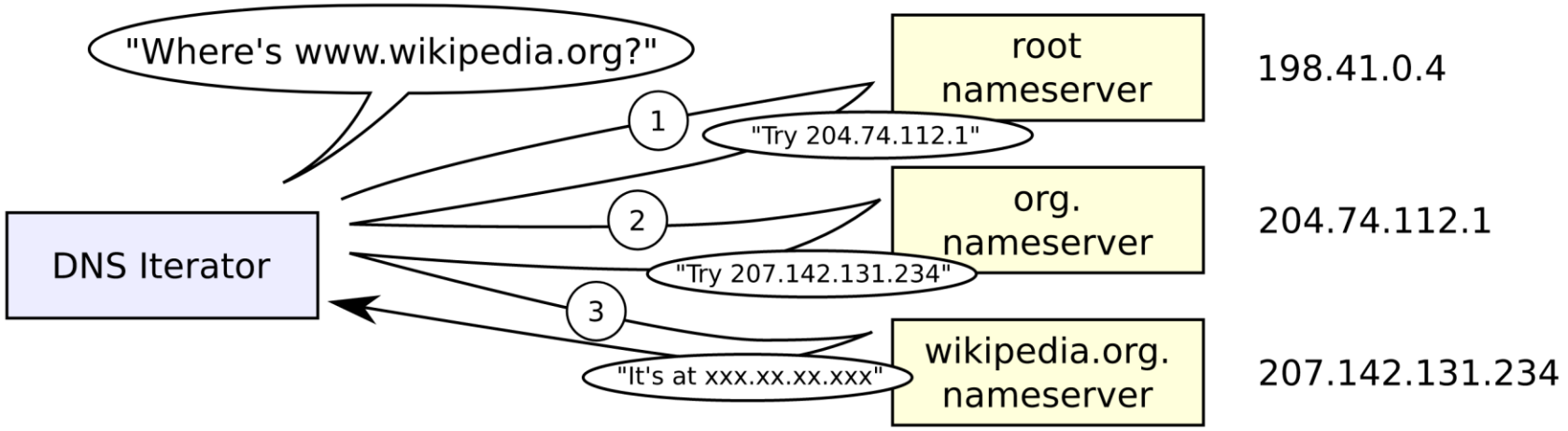
The **Domain Name System (DNS) protocol** converts memorable names to routable IP addresses used for passing traffic.

- facebook.com → 31.13.65.36
- auburn.edu → 131.204.138.170



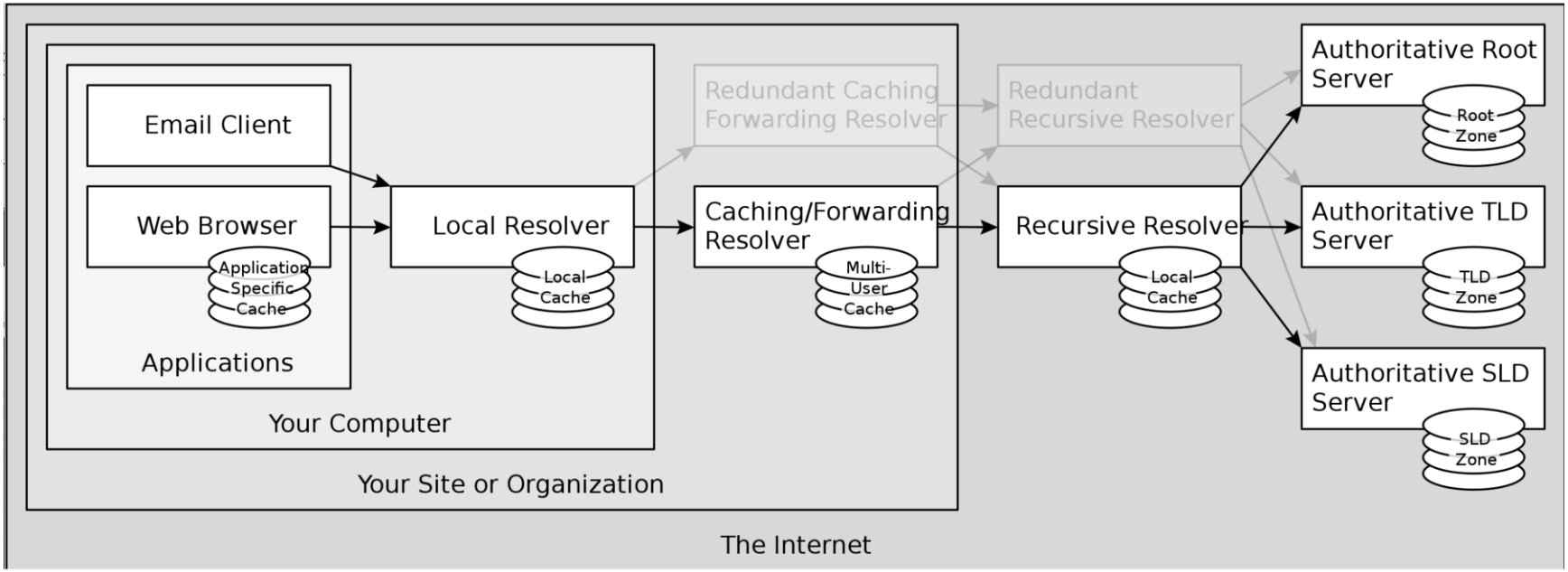
Recursive DNS Look-Up

A **recursive DNS lookup** is when a server iterates over the DNS labels in reverse-order to find the name manually.



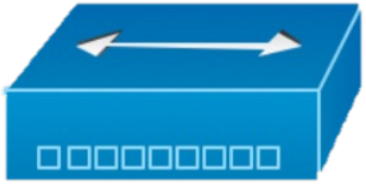
DNS in the Real-World

Real-world DNS is very complicated due to independent caching and resolvers.



Network Devices

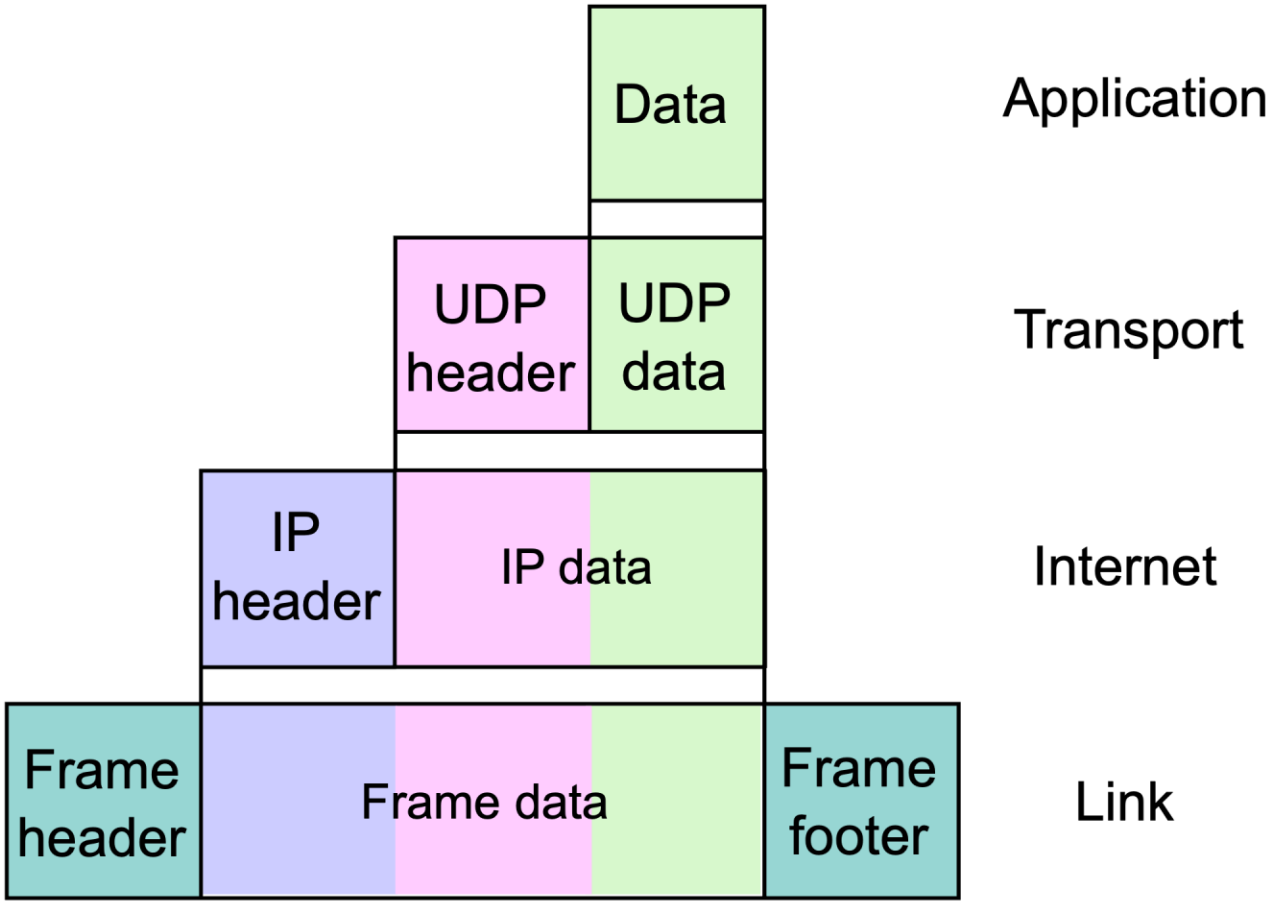
- **Hubs** are L1 devices
  - Packet comes in, packets go-out



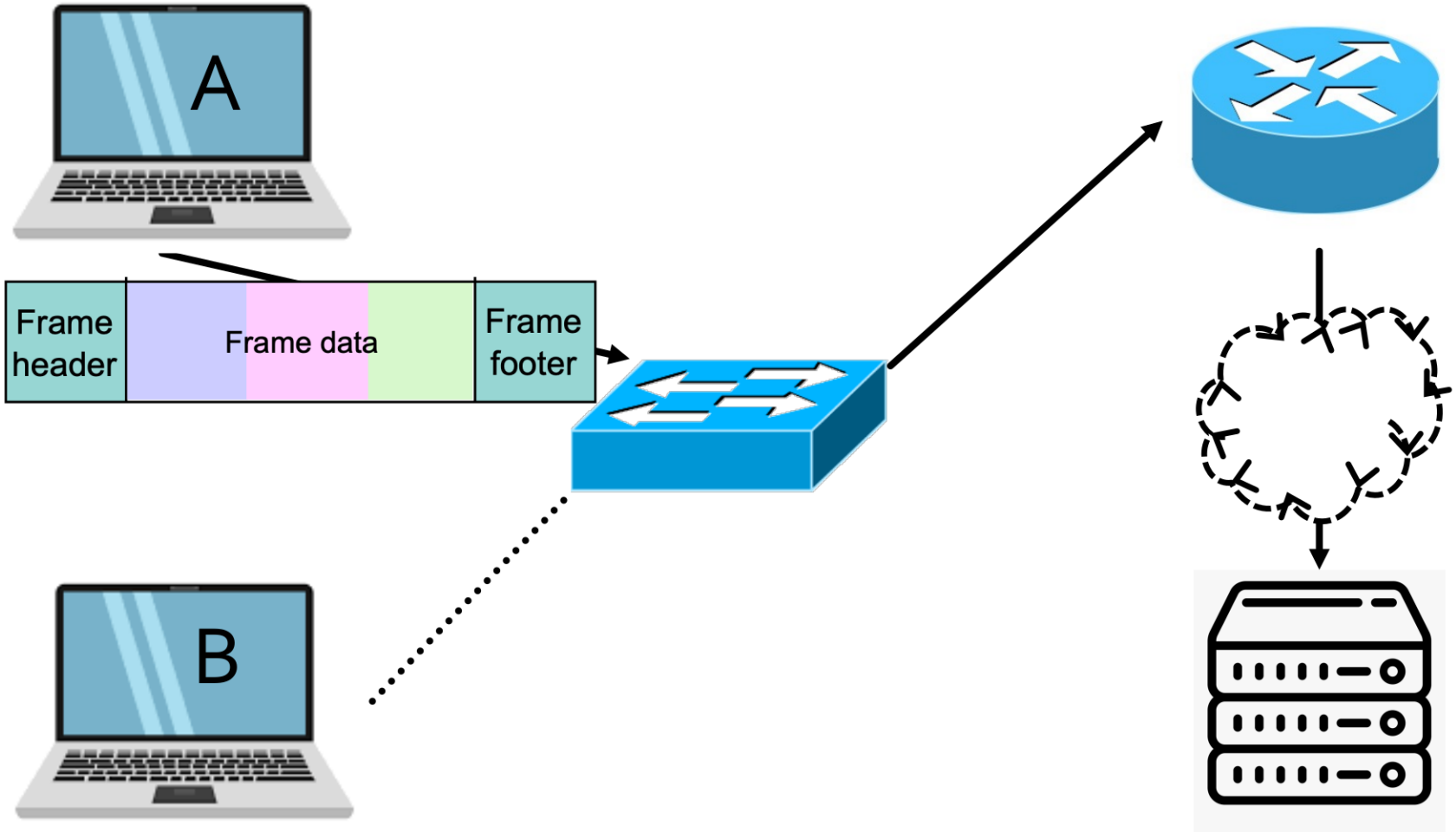
- **Switches** are L2 devices
  - Dispatch packets via MAC address
  - "L3 switches" are common but are not what we're talking about

- **Routers** are L3 devices
  - Dispatch packets via IP address
  - Lots of things called "routers" aren't actually **routers** (but some are)

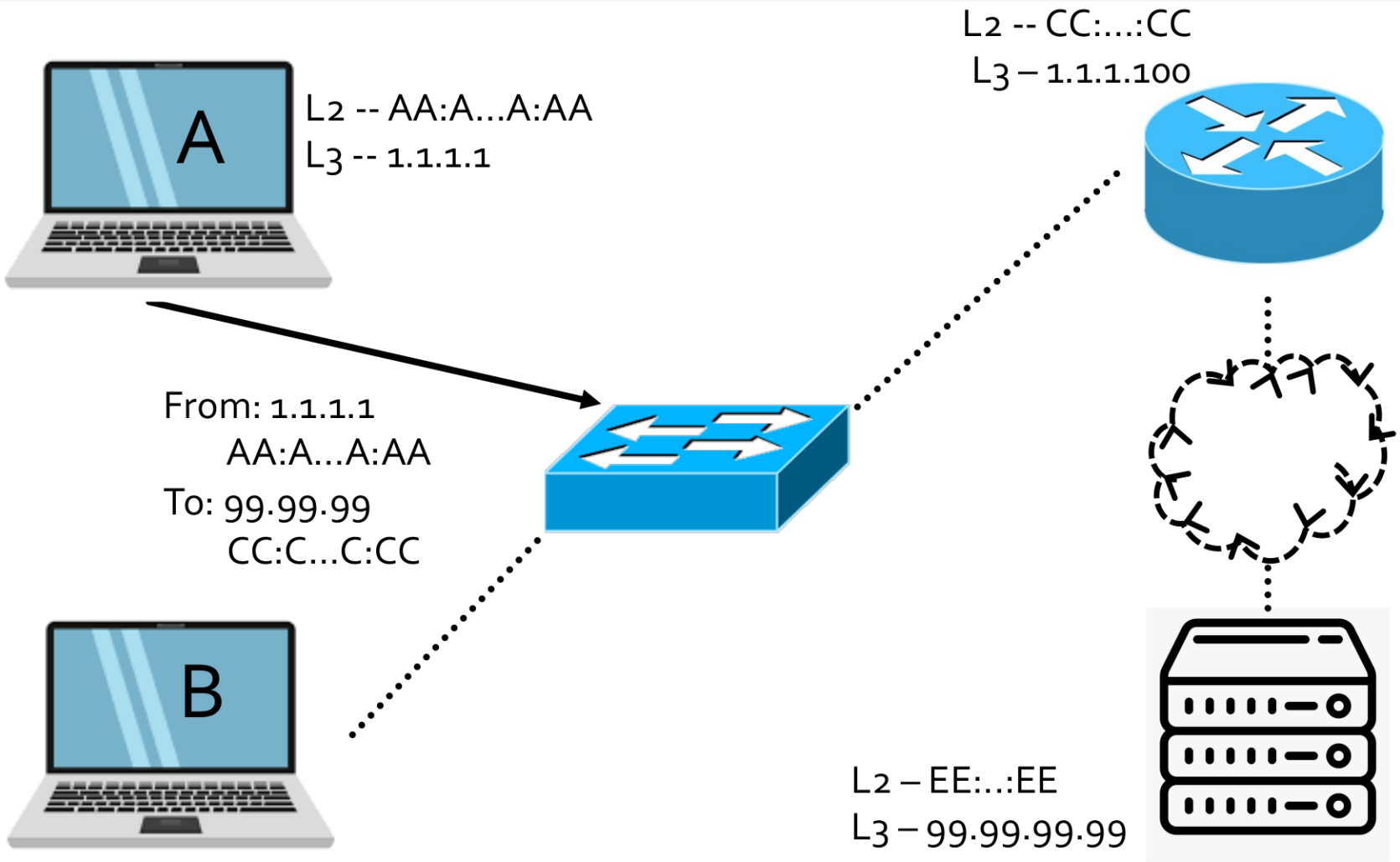
Data Encapsulation



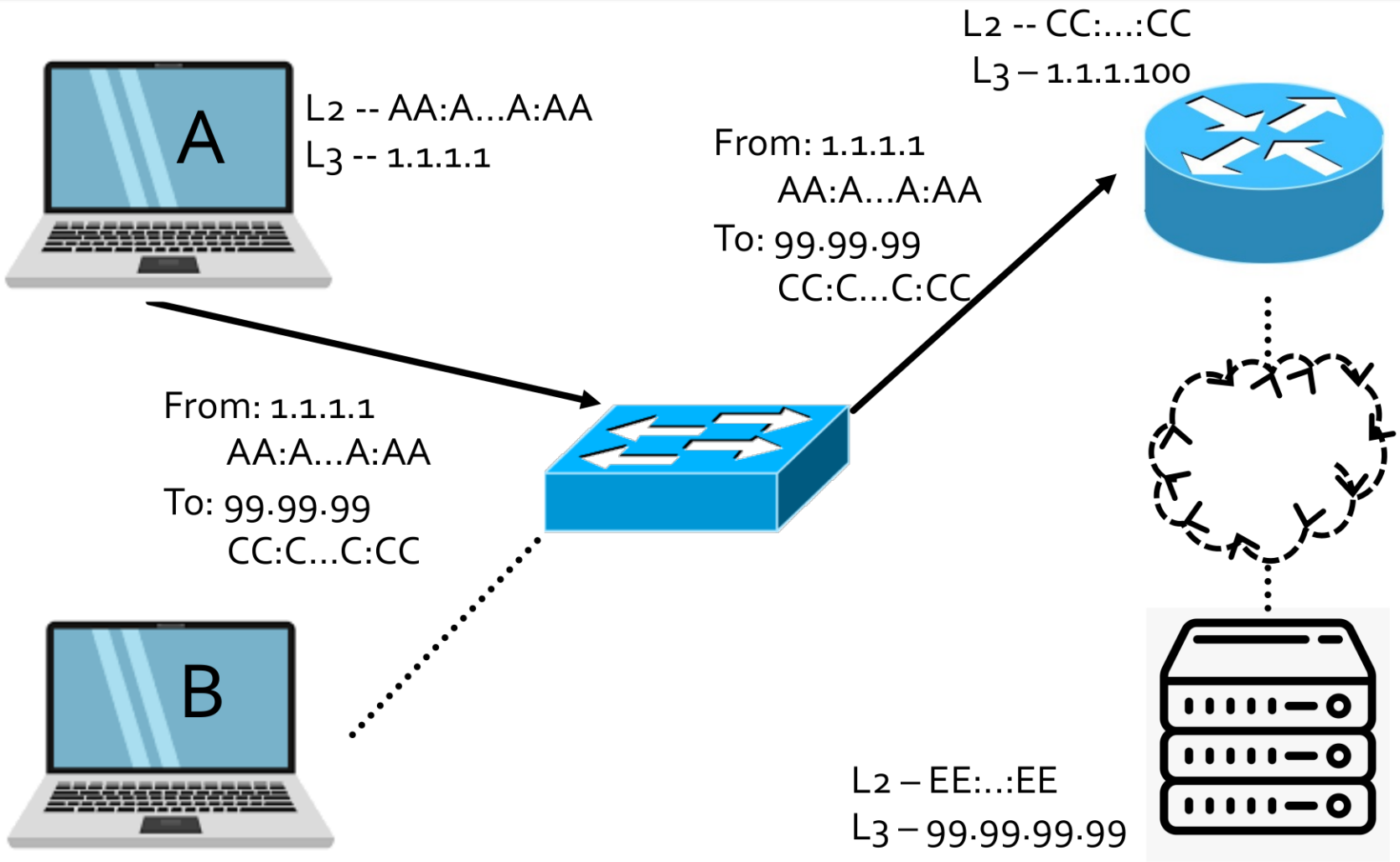
Link Layer Data Flow (external)



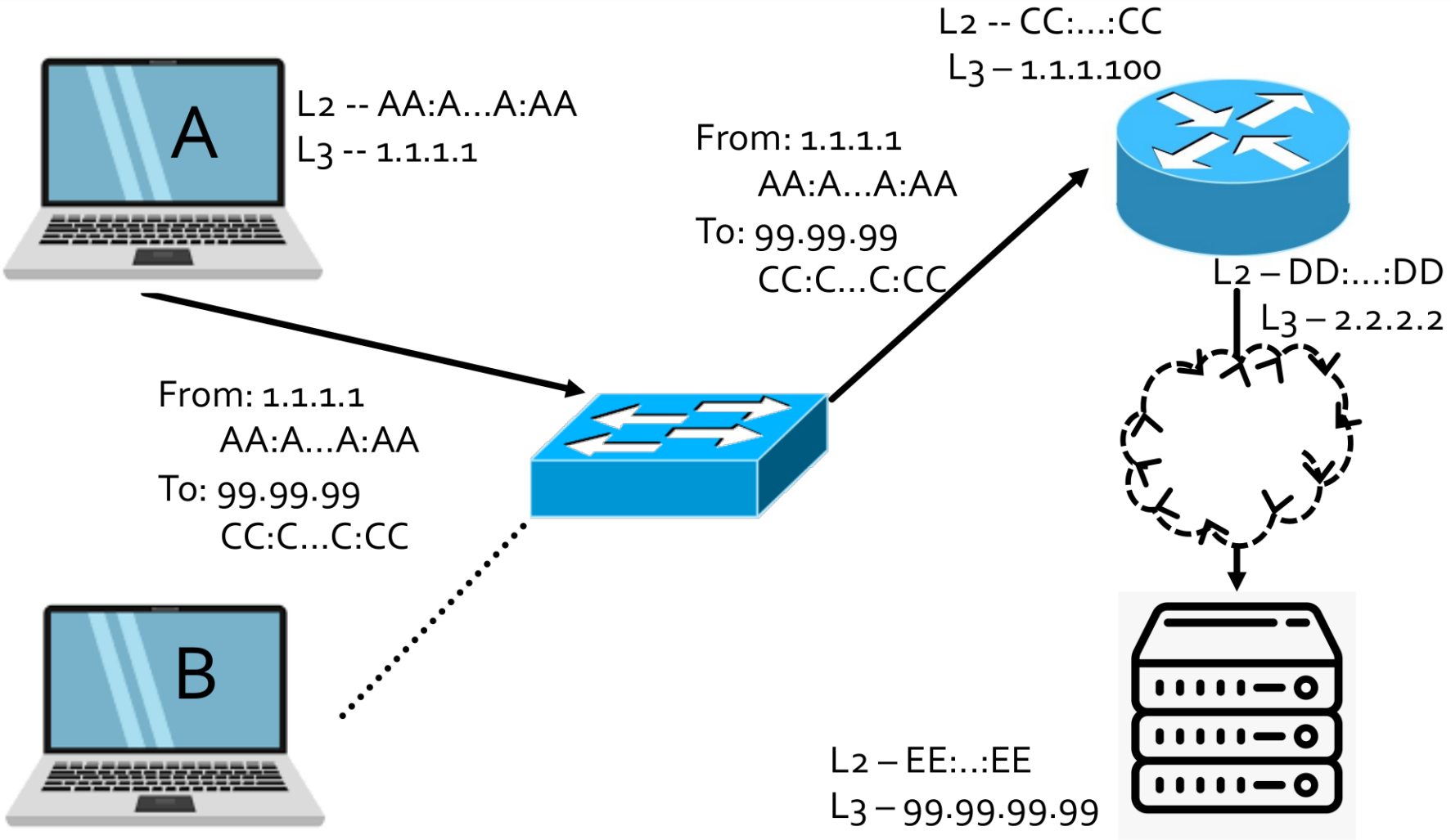
Link Layer Data Flow (external)



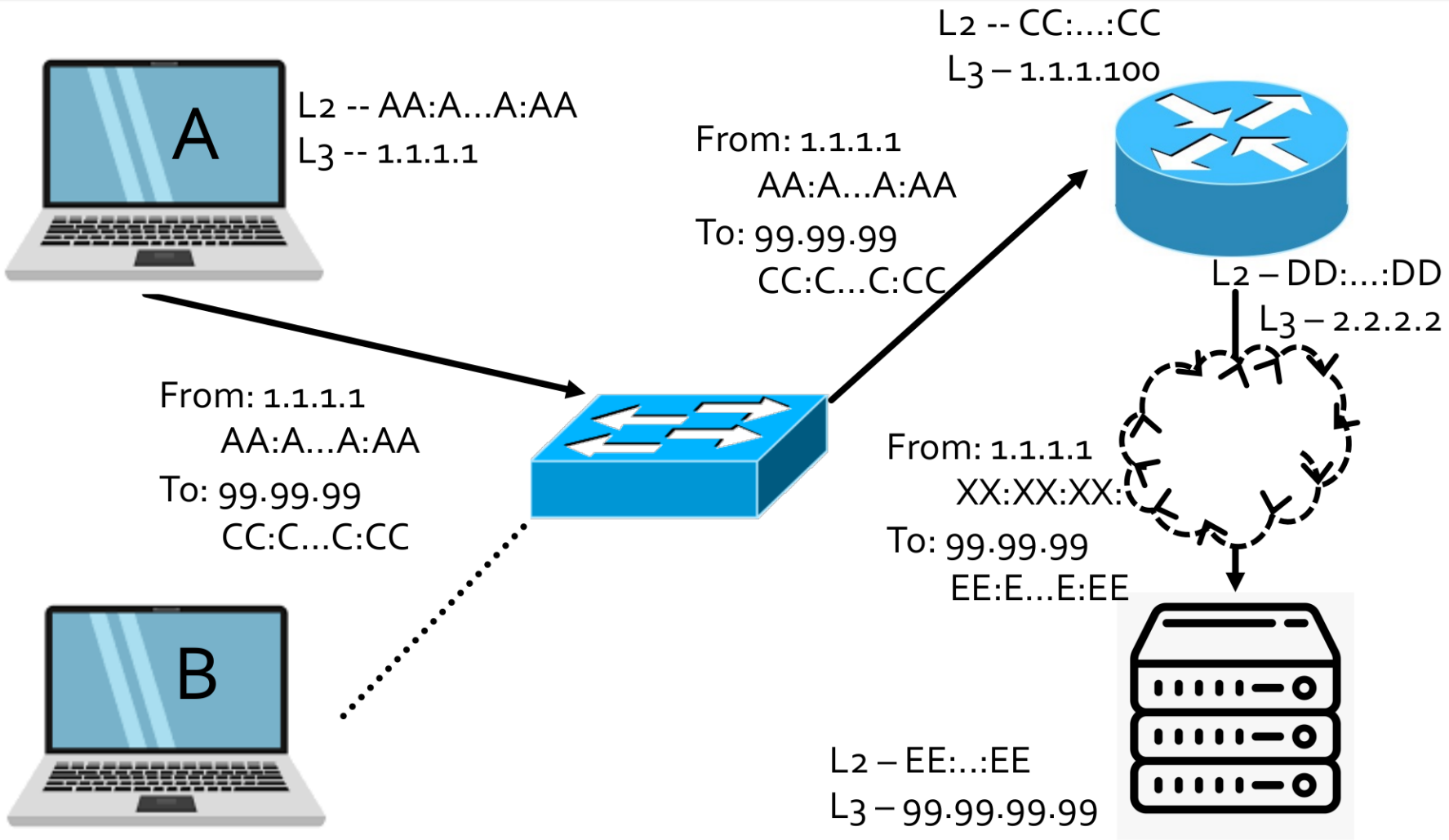
Link Layer Data Flow (external)



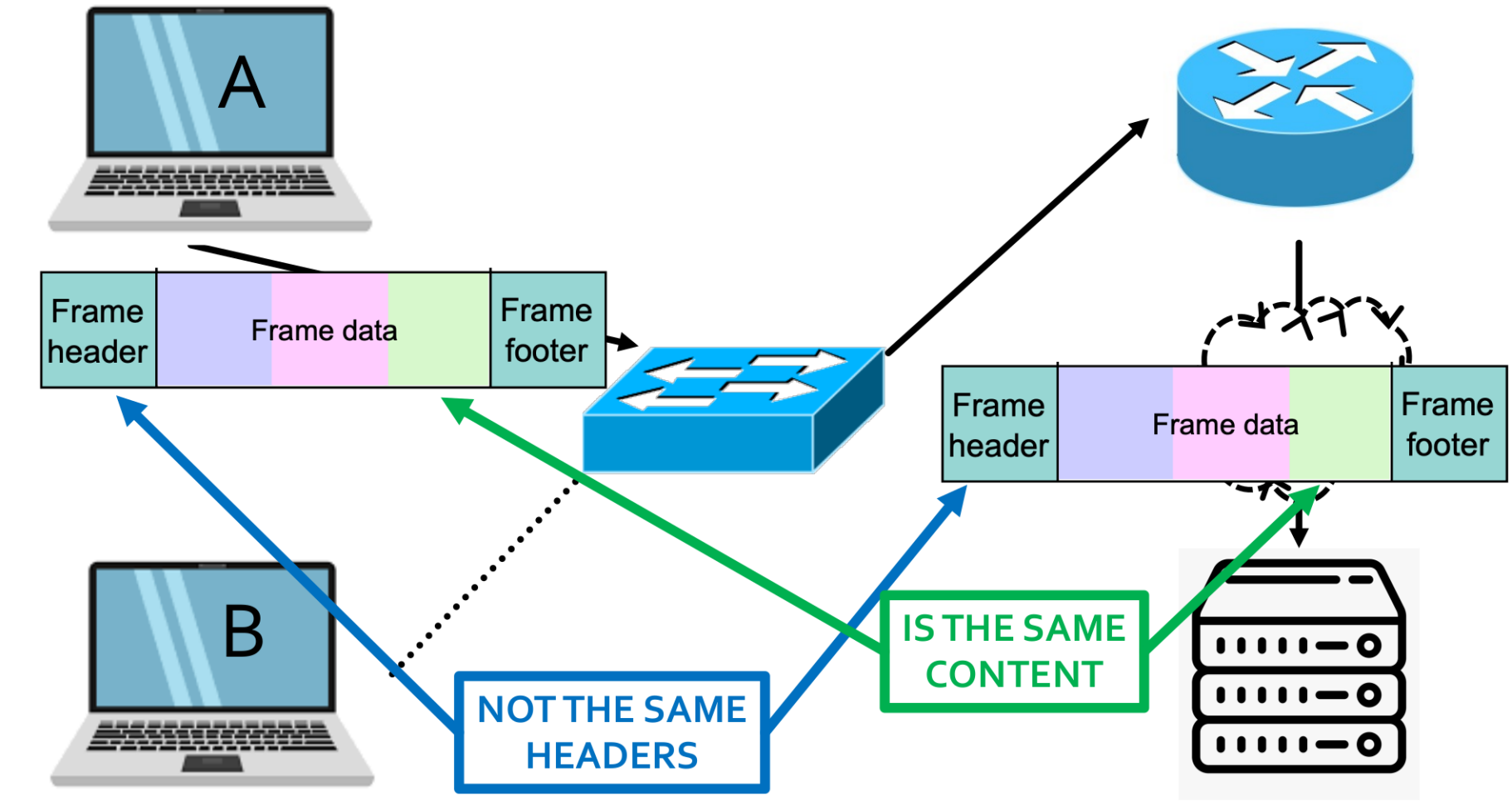
Link Layer Data Flow (external)



Link Layer Data Flow (external)



Link Layer Data Flow (external)

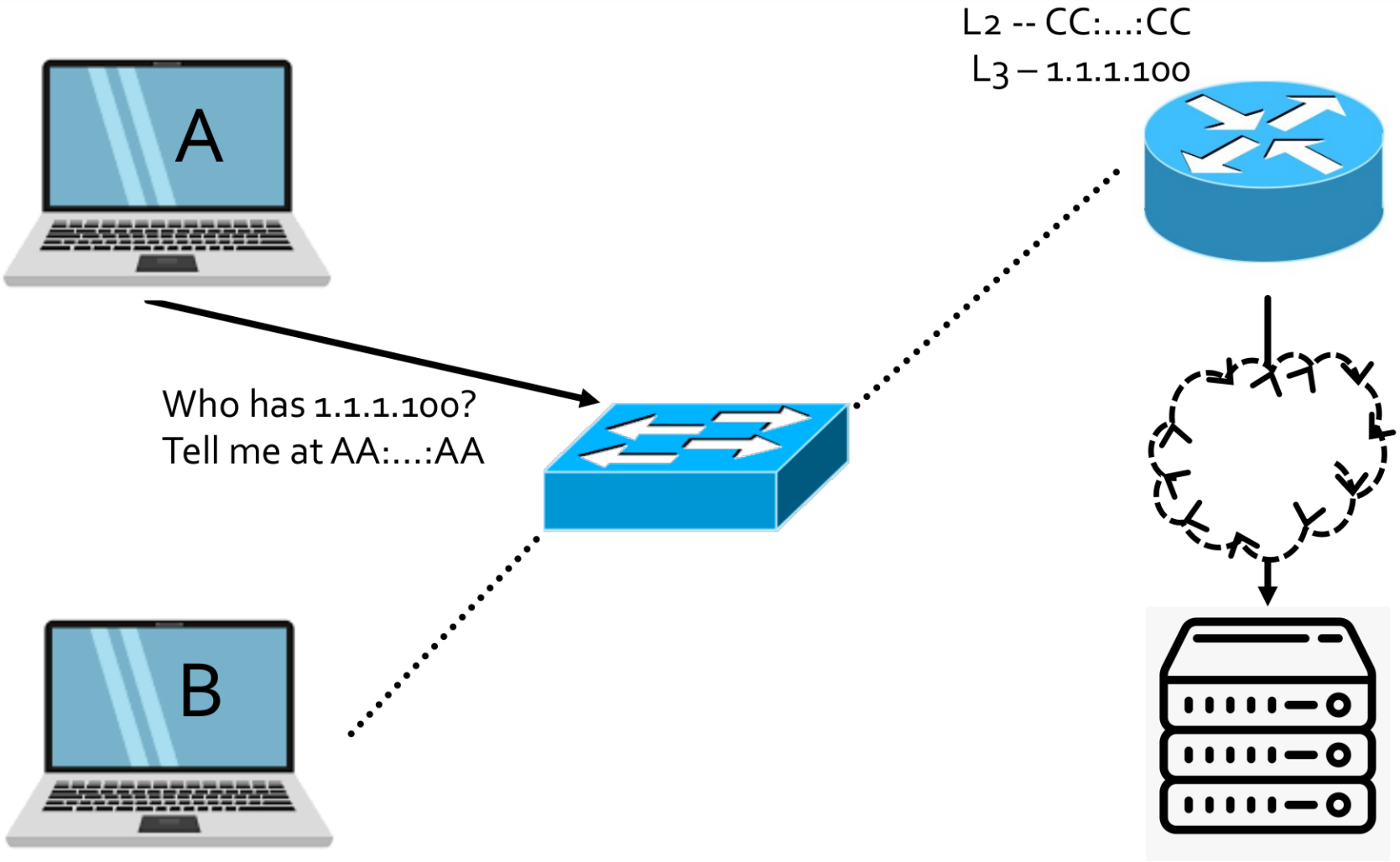


ARP Protocol

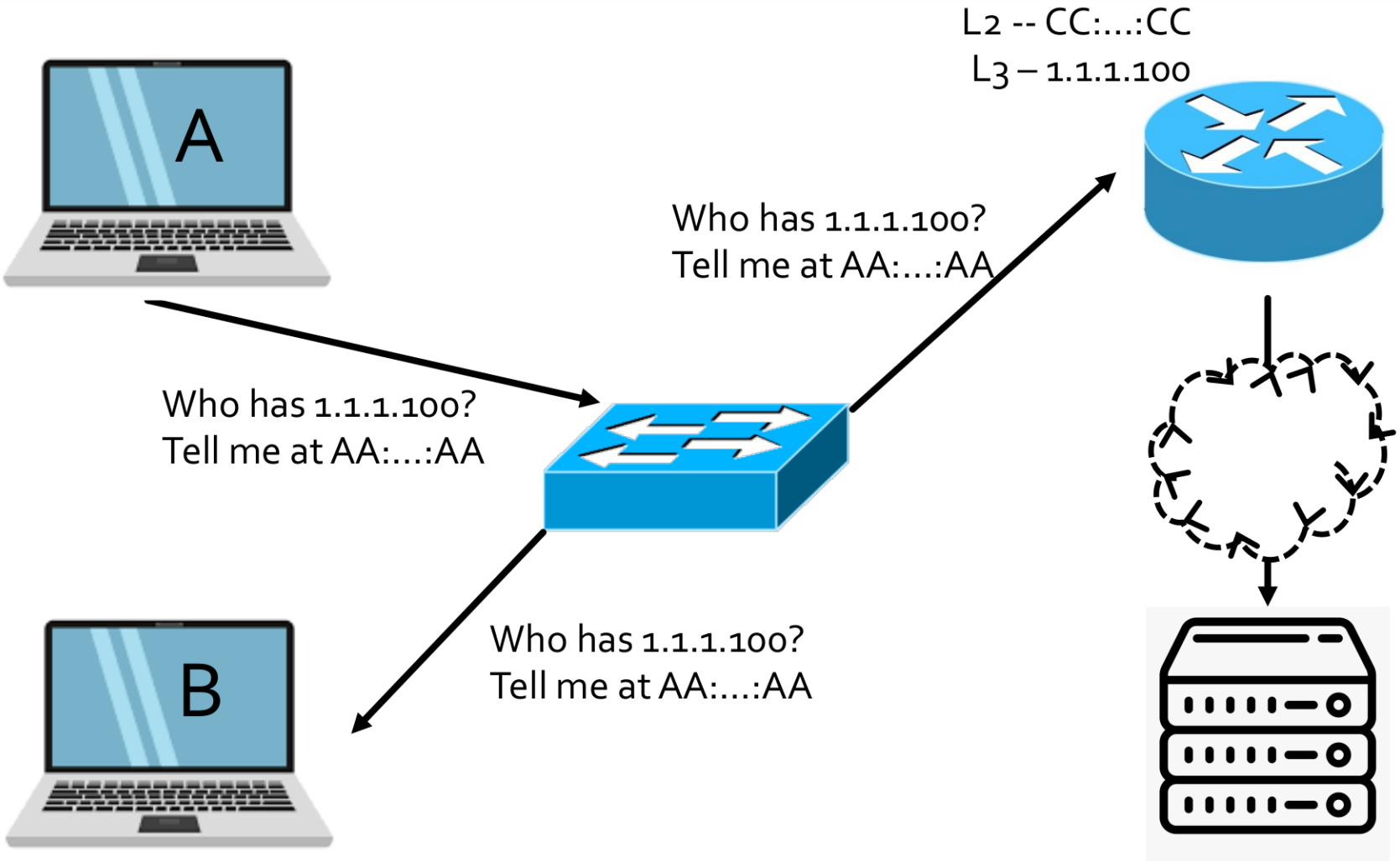
The **Address Resolution Protocol (ARP)** allows hosts to map IP ←→ MAC addresses.

- ARP Announcements
  - “I have IP 1.1.1.1 and my MAC is XX:X…X:XX”
- ARP Probe
  - “What MAC is associated with 1.1.1.1?”

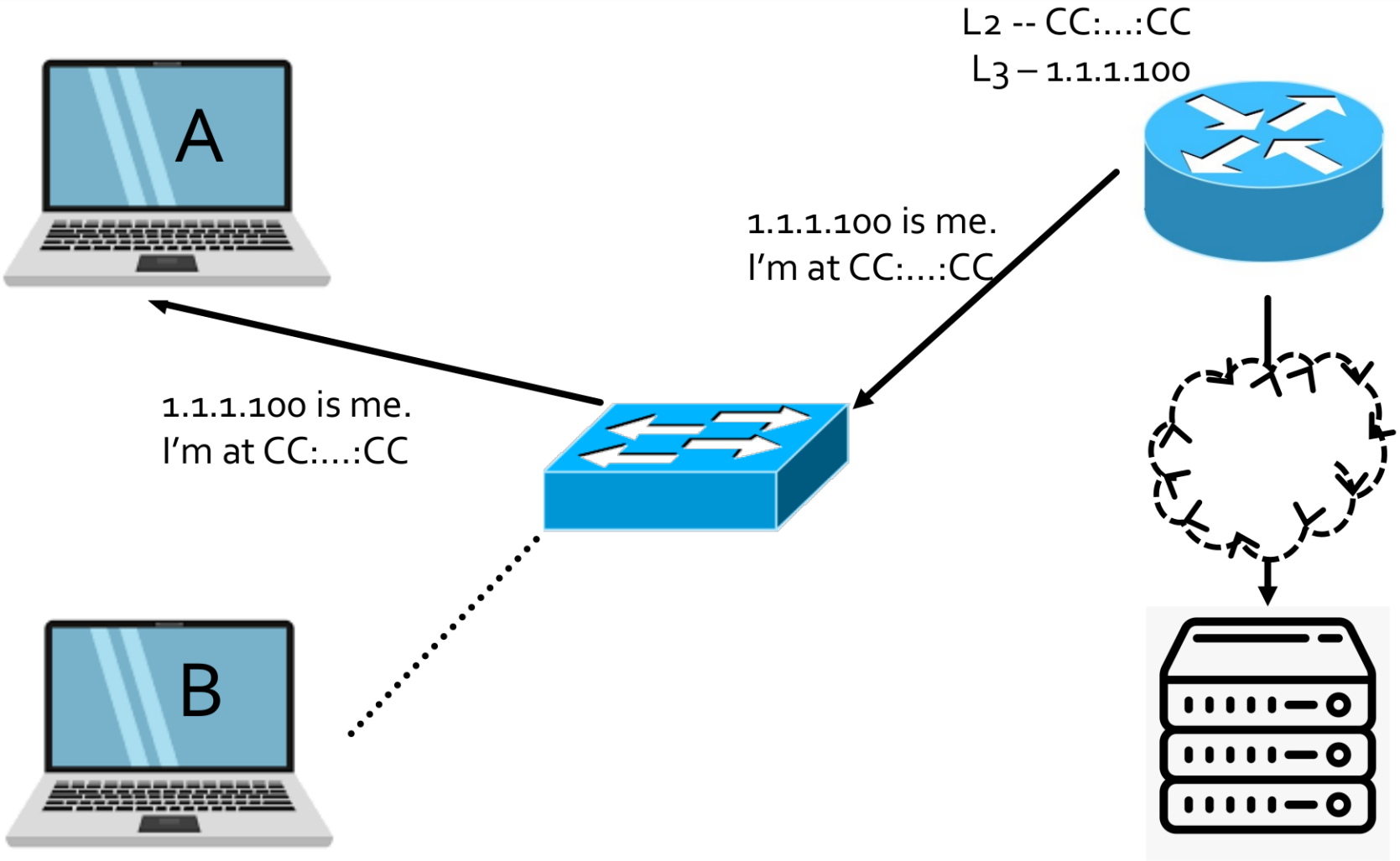
ARP Protocol



ARP Protocol



ARP Protocol



ARP Protocol

The **Address Resolution Protocol (ARP)** allows hosts to map IP ←→ MAC addresses.

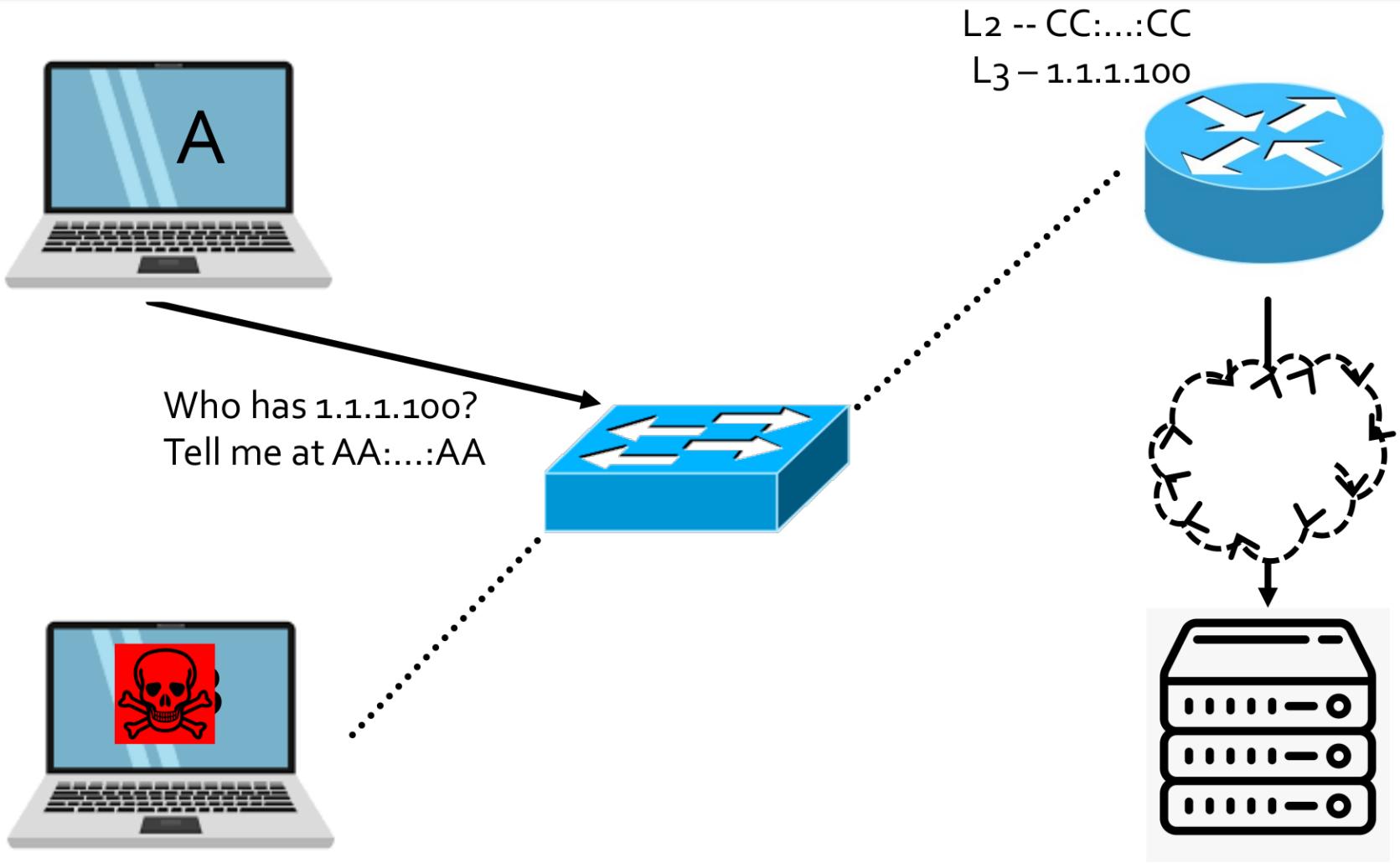
- ARP Announcements
  - "I have IP 1.1.1.1 and my MAC is XX:X…X:XX"
- ARP Probe
  - "What MAC is associated with 1.1.1.1?"
- L2 Default gateway
  - The host to send to if not on local network

ARP Poisoning/Spoofing

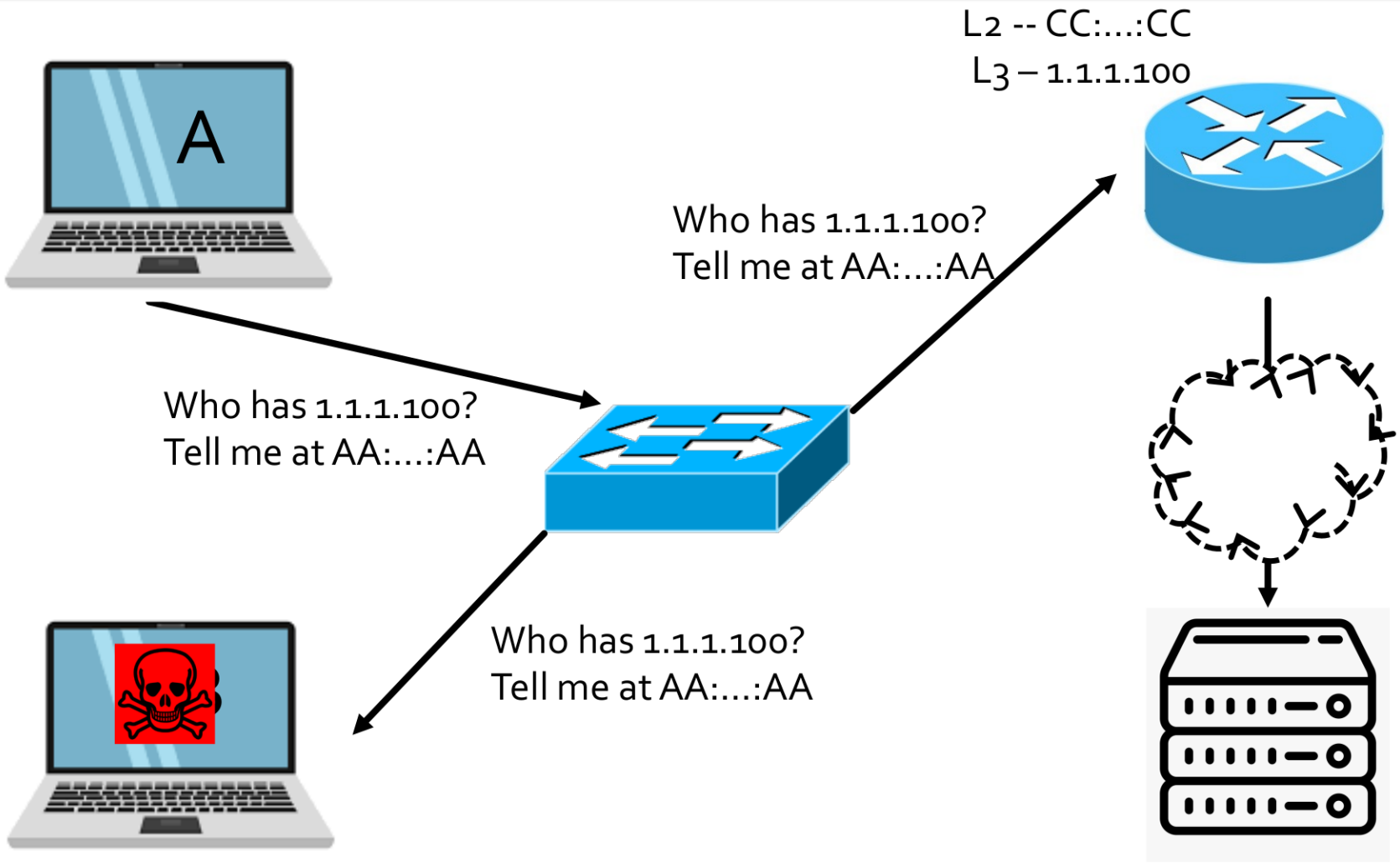
ARP Poisoning/Spoofing Attacks are a set of techniques used to misdirect traffic inside a LAN via L2.

- Malicious actor on local network

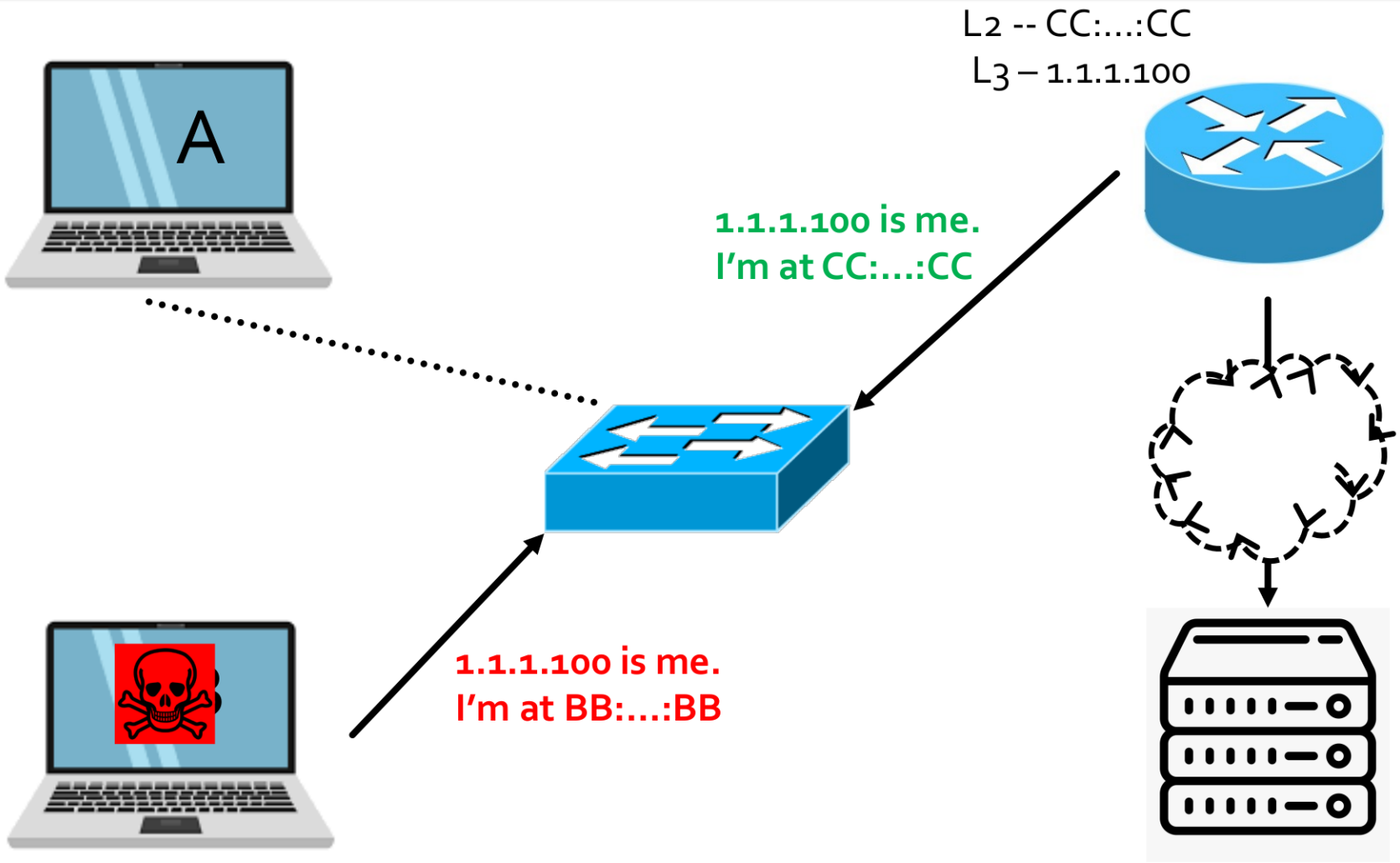
ARP Poisoning/Spoofing



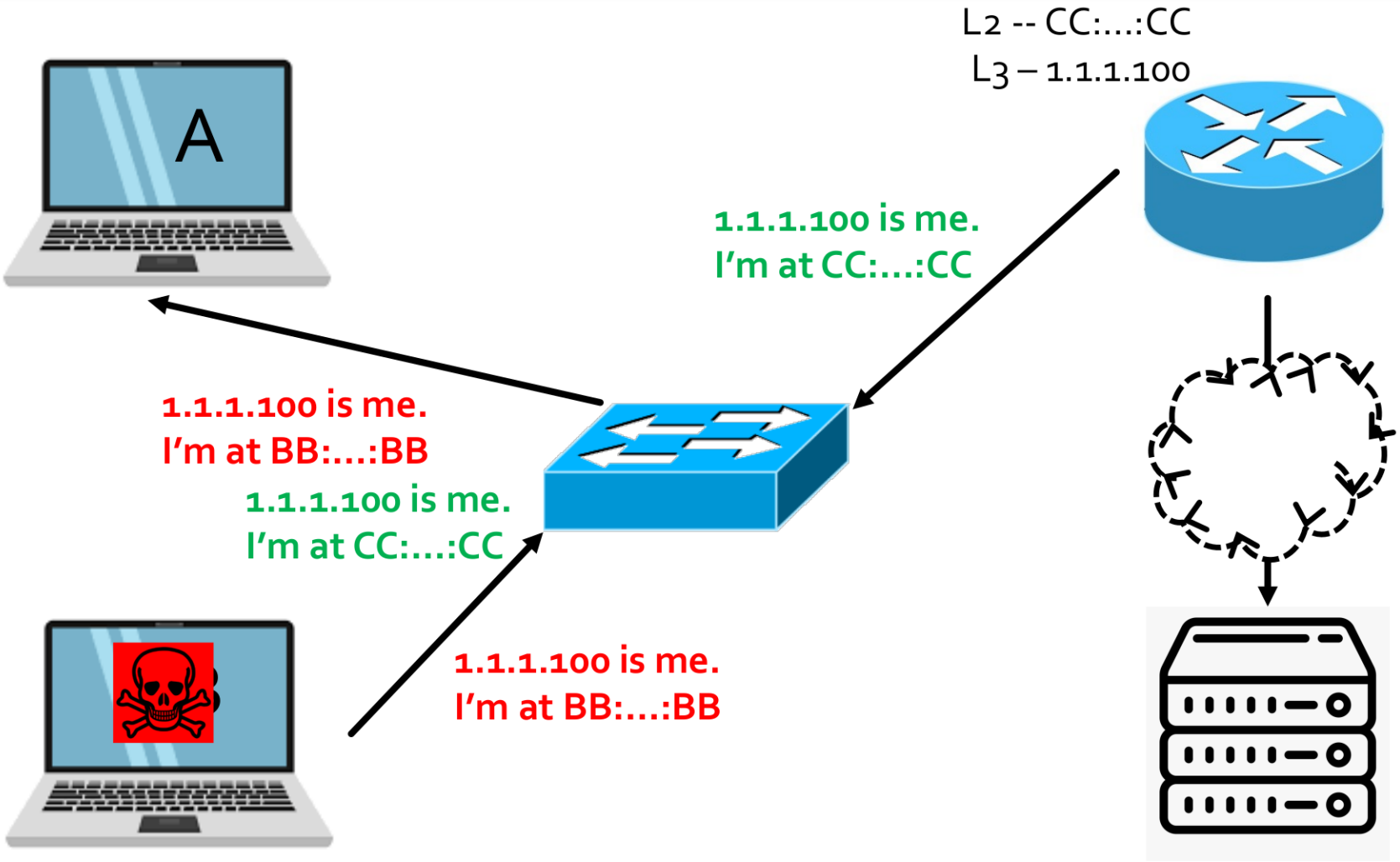
ARP Poisoning/Spoofing



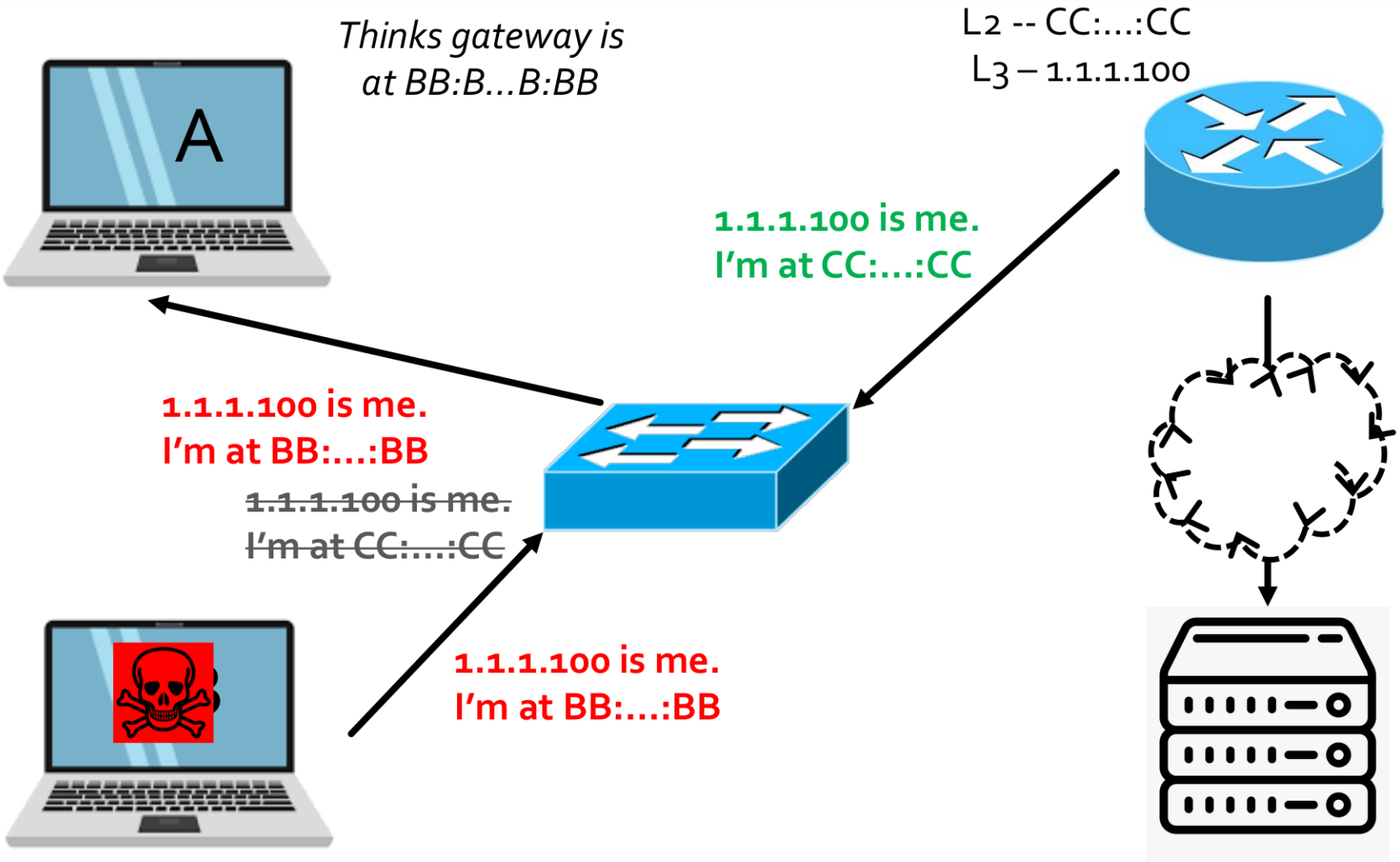
ARP Poisoning/Spoofing



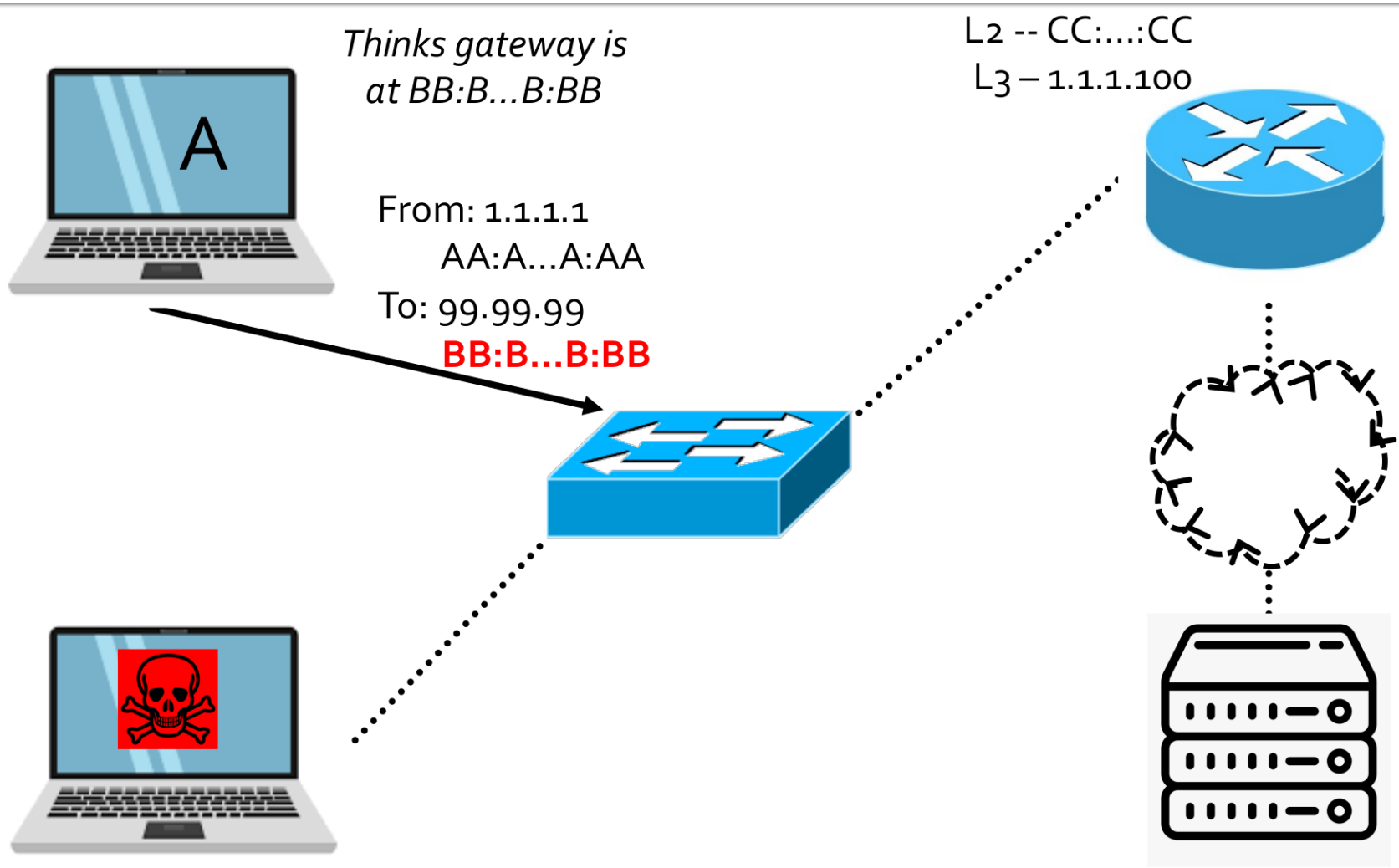
ARP Poisoning/Spoofing



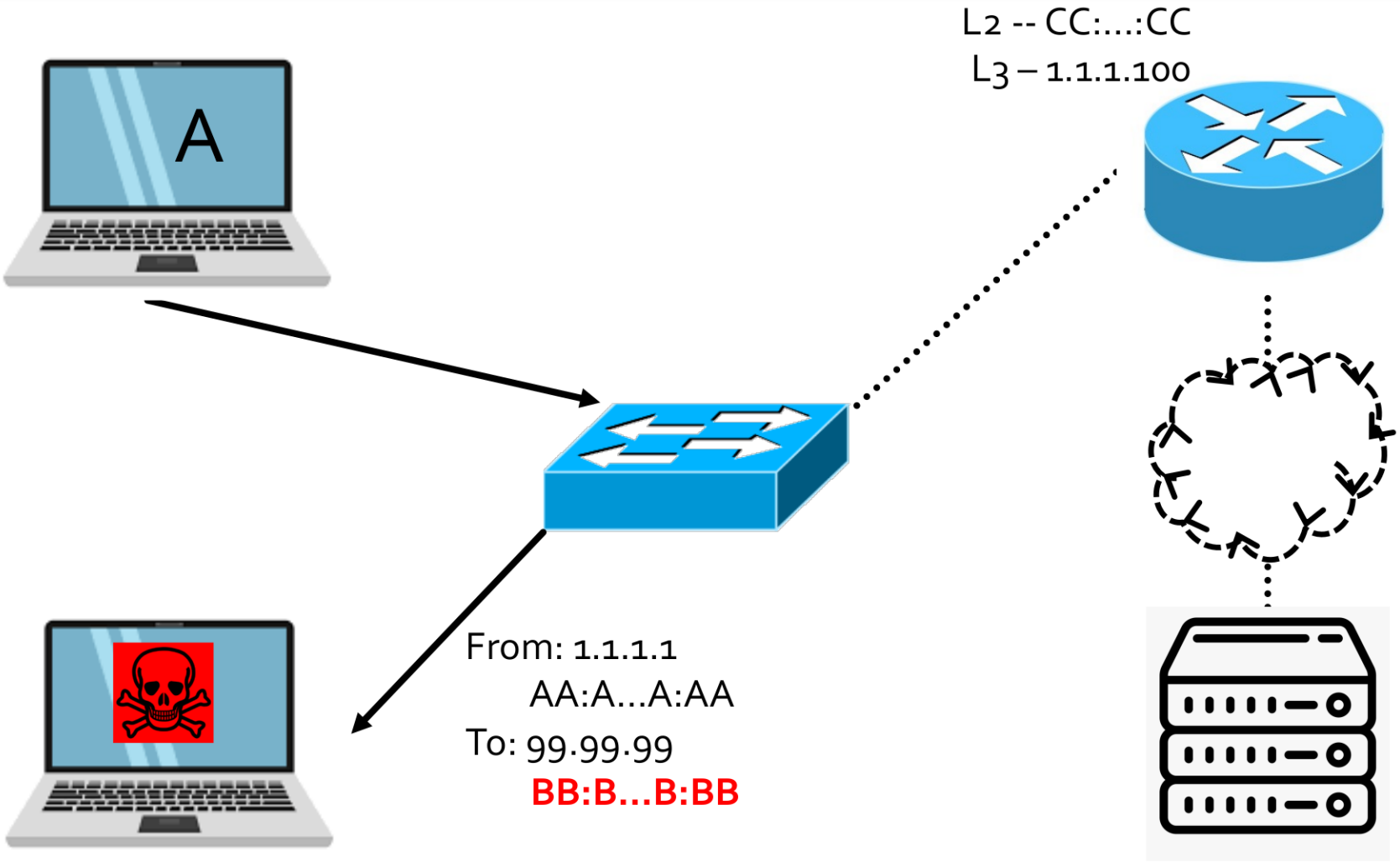
ARP Poisoning/Spoofing



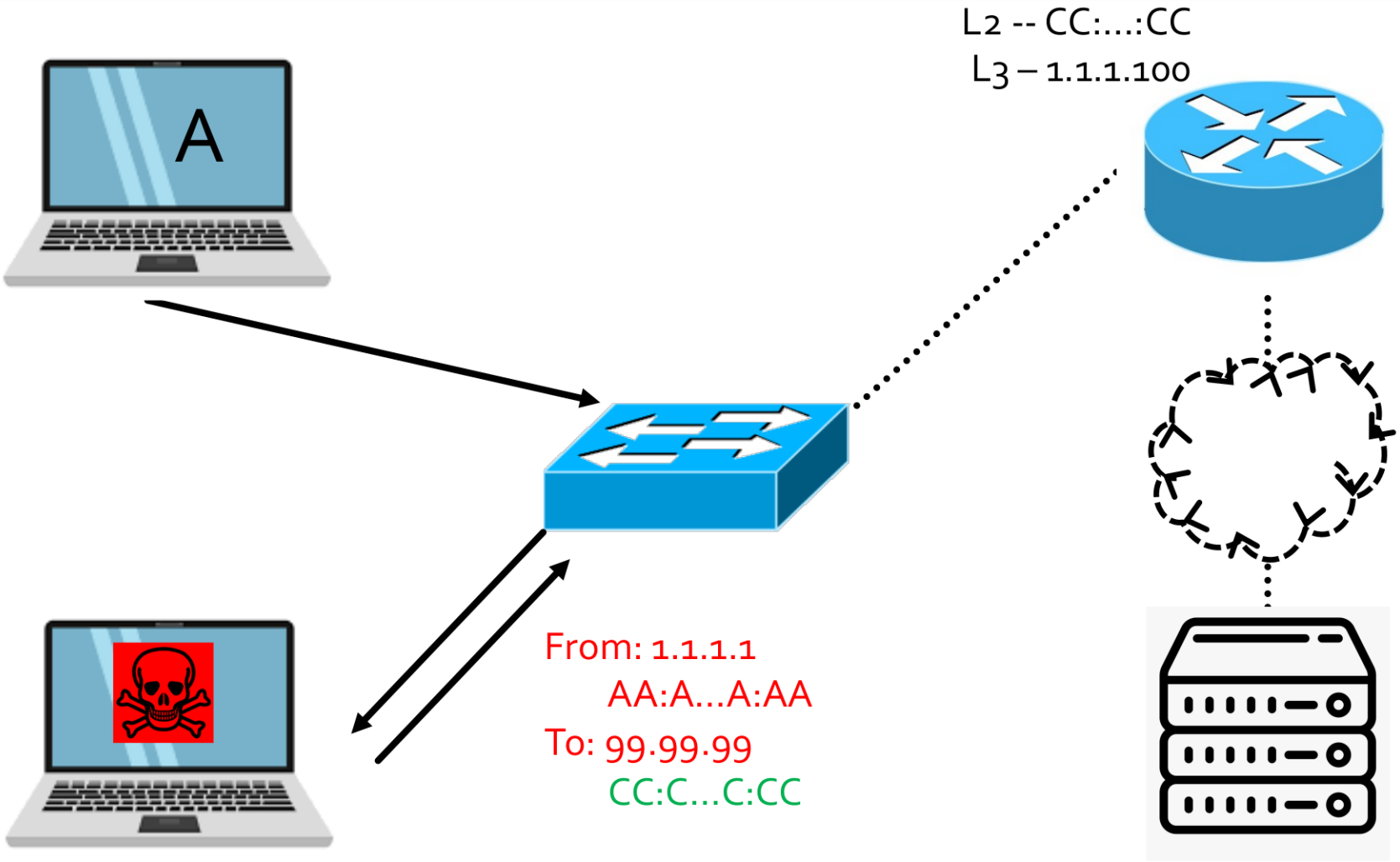
ARP Poisoning/Spoofing



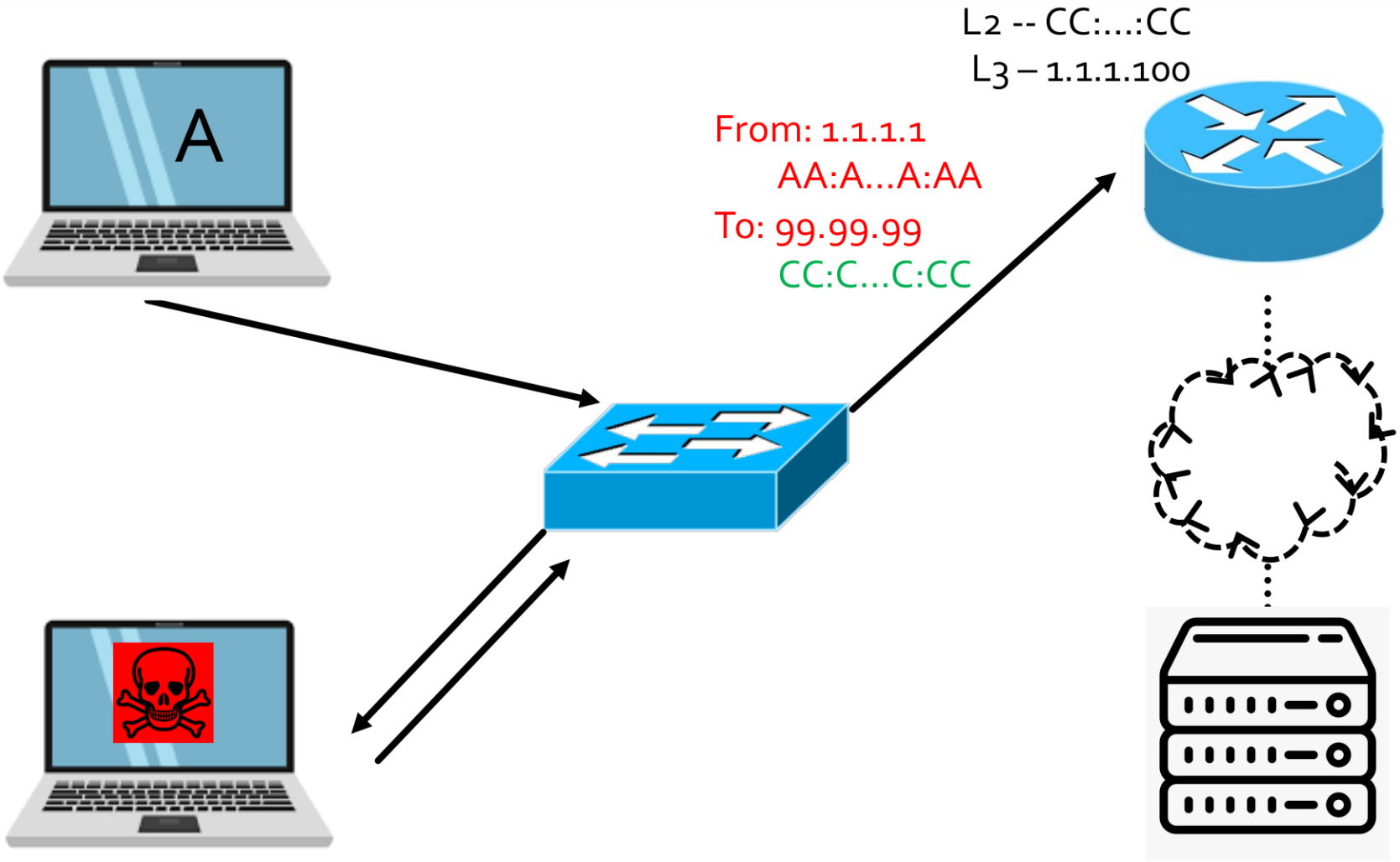
ARP Poisoning/Spoofing



ARP Poisoning/Spoofing



ARP Poisoning/Spoofing

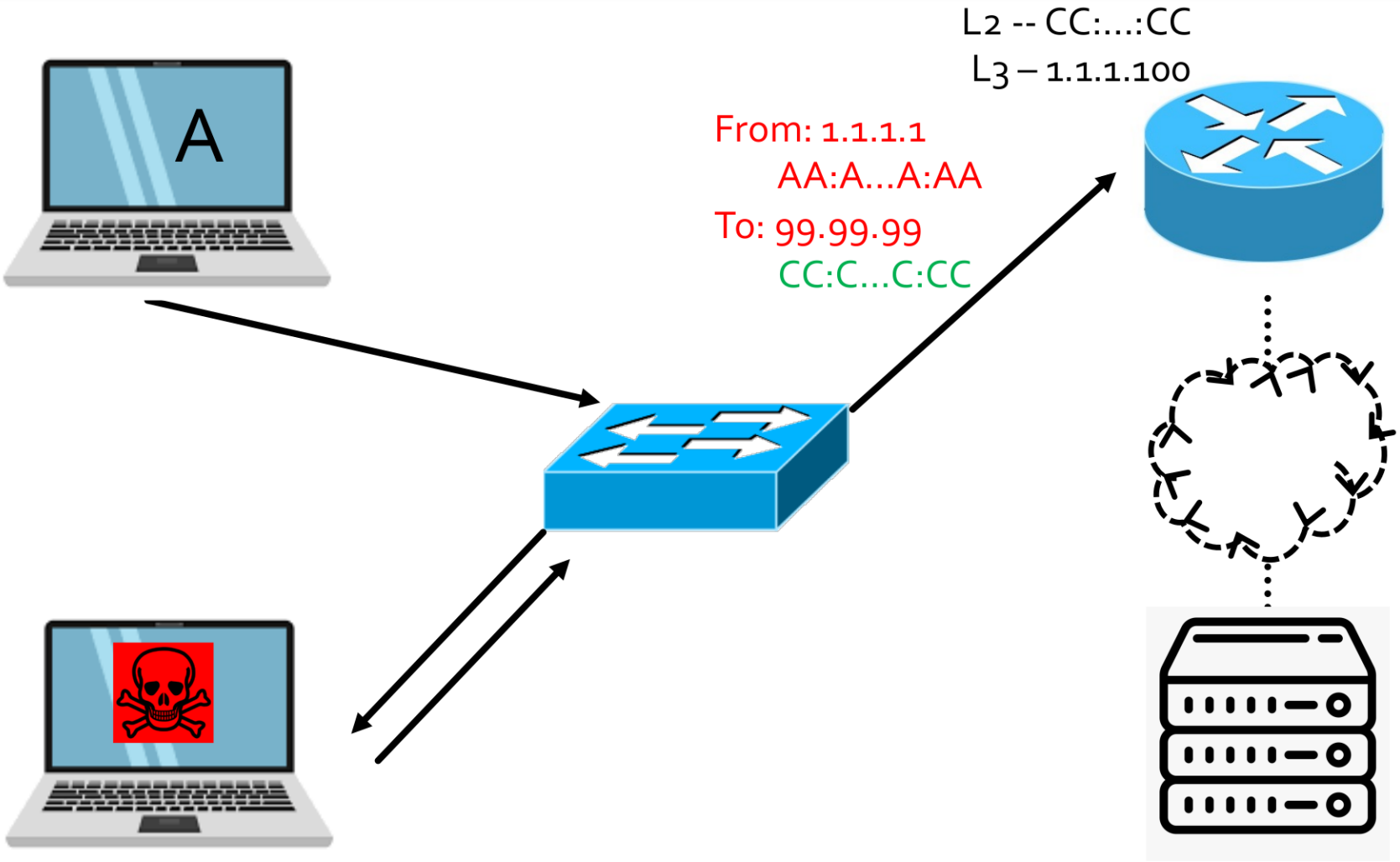


ARP Poisoning/Spoofing

ARP Poisoning/Spoofing Attacks are a set of techniques used to misdirect traffic inside a LAN via L2.

- Malicious actor on local network
- Can be used to:
  - DoS another client
  - Cause network thrashing
  - **Intercept traffic**

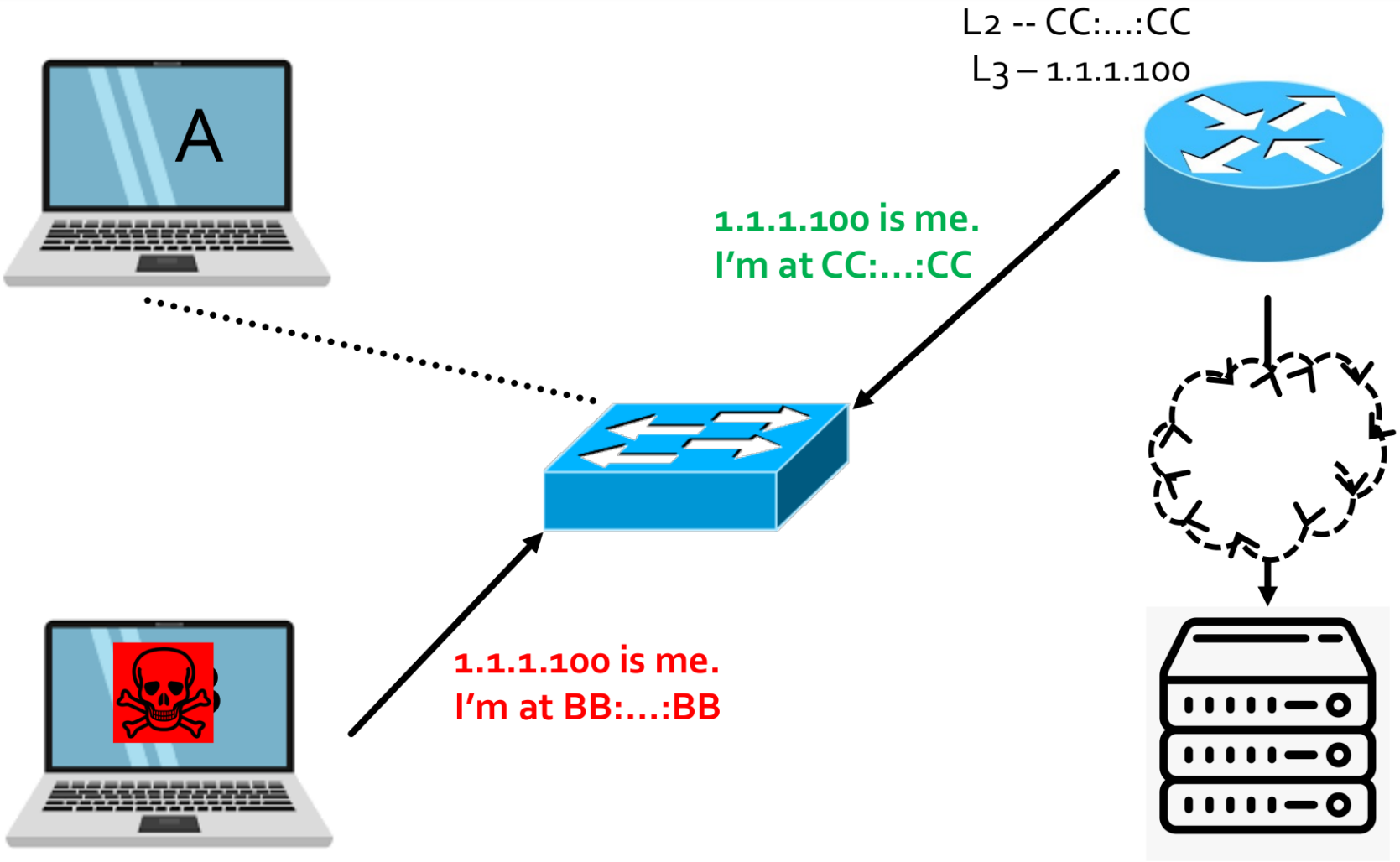
ARP Poisoning/Spoofing



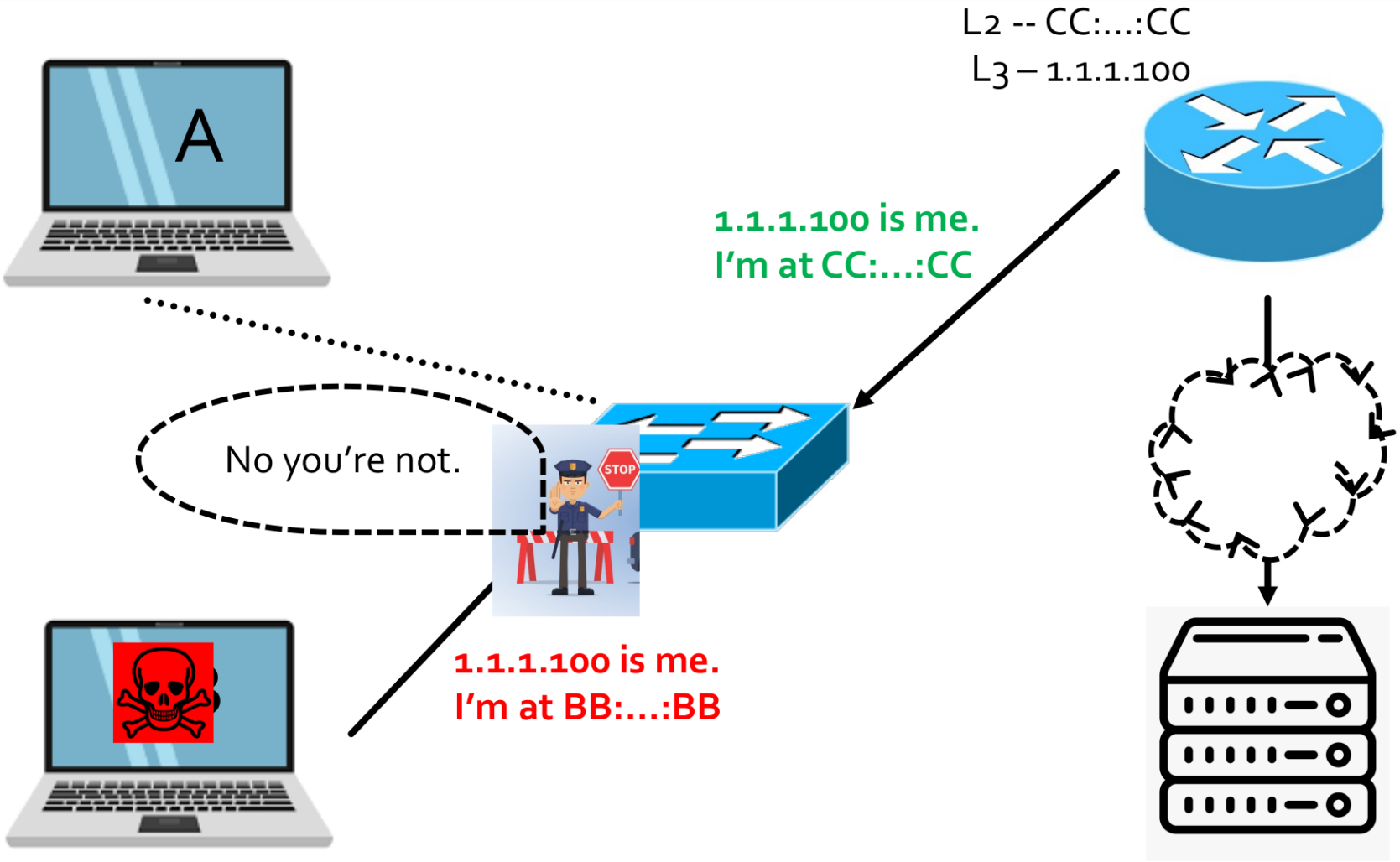
Defending against ARP Attacks

- More intelligent switching
  - “Sticky” MACs: 1 physical port = 1 MAC addr
  - 802.1X: Authenticate physical port access
  - *<many more>*

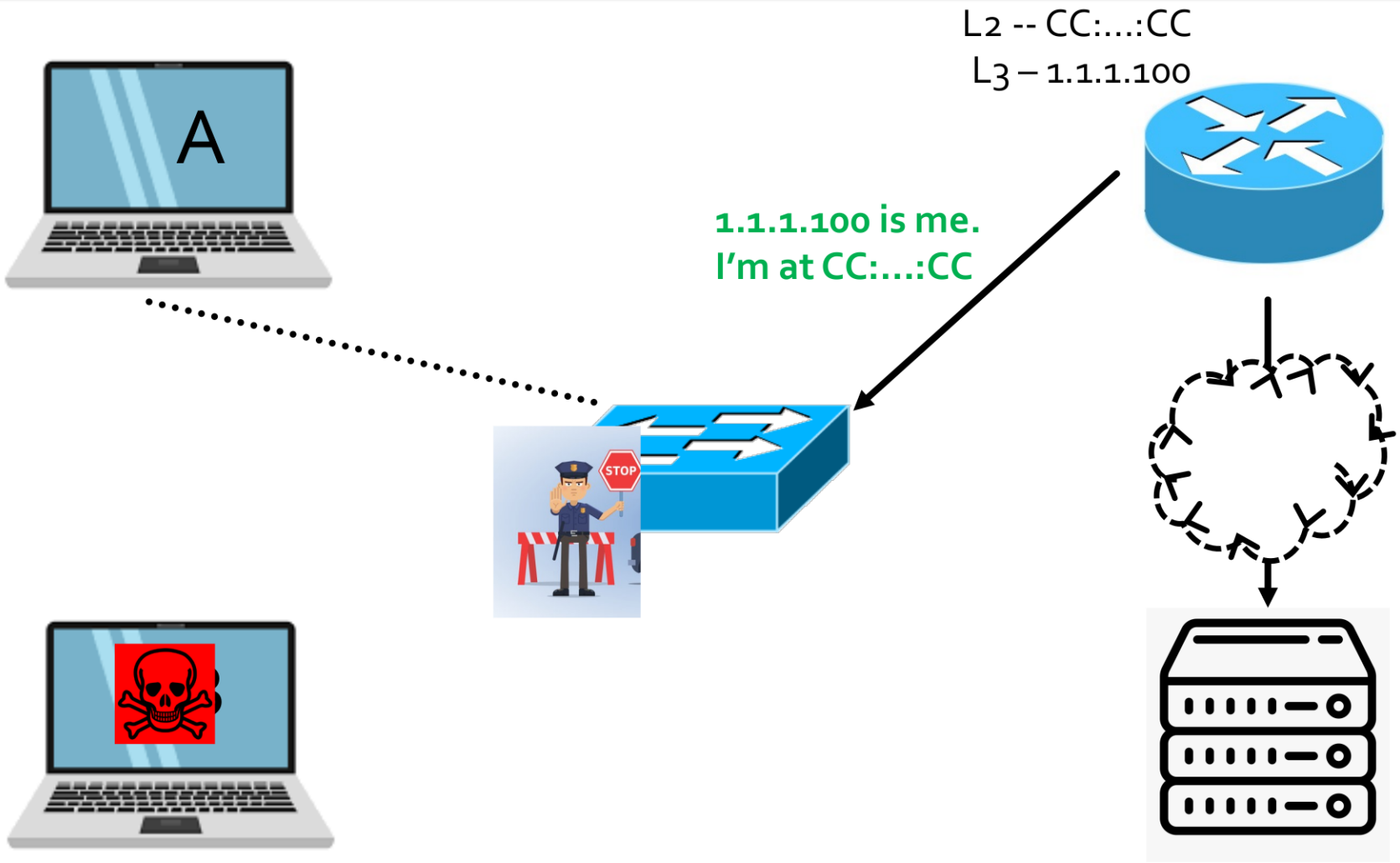
"Sticky" MACs


L2 -- CC:...:CC
L3 – 1.1.1.100
A
1.1.1.100 is me.
I'm at CC:...:CC
1.1.1.100 is me.
I'm at BB:...:BB

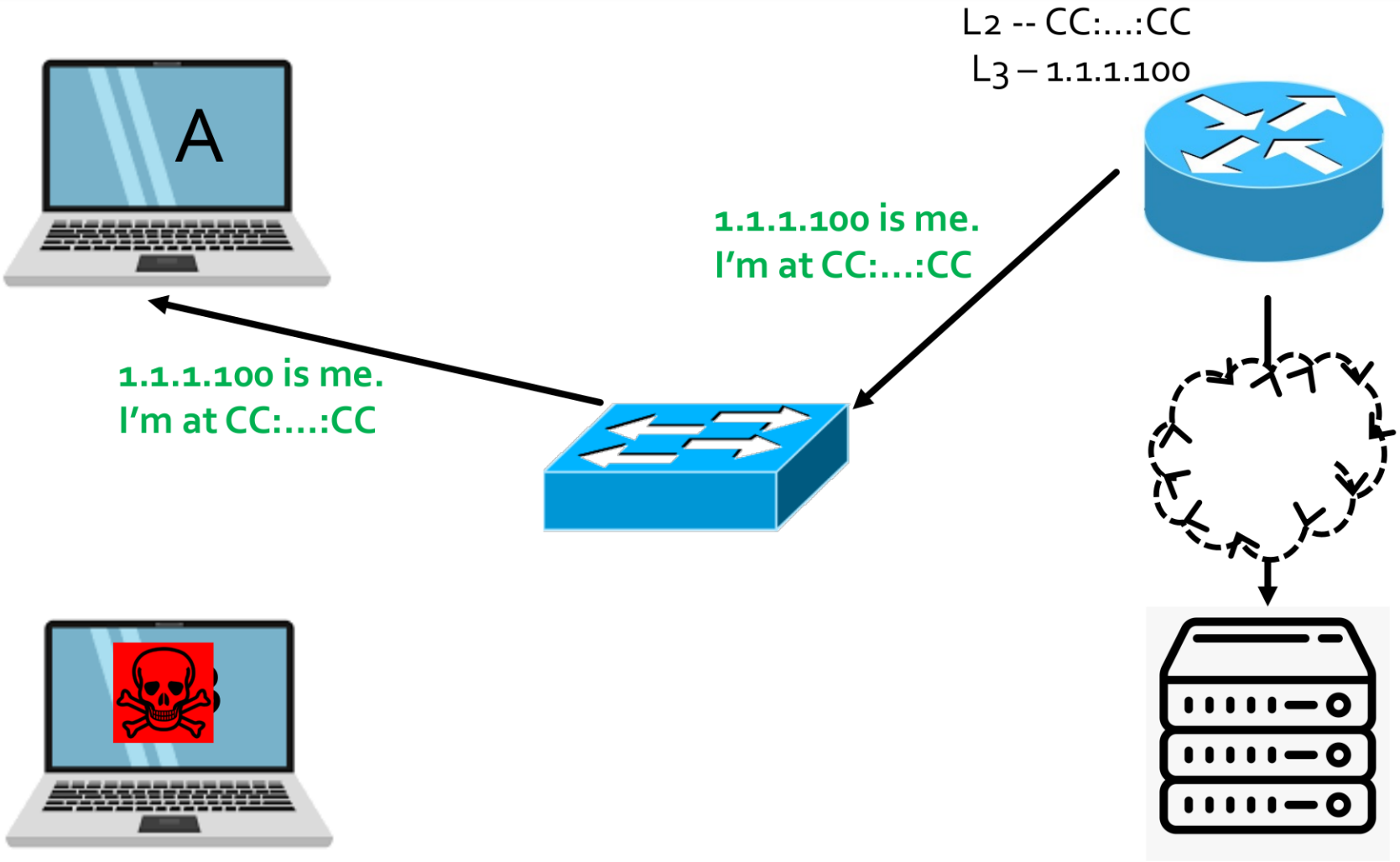
"Sticky" MACs


L2 -- CC:...:CC
L3 – 1.1.1.100
A
1.1.1.100 is me.
I'm at CC:...:CC
No you're not.
STOP
1.1.1.100 is me.
I'm at BB:...:BB

"Sticky" MACs



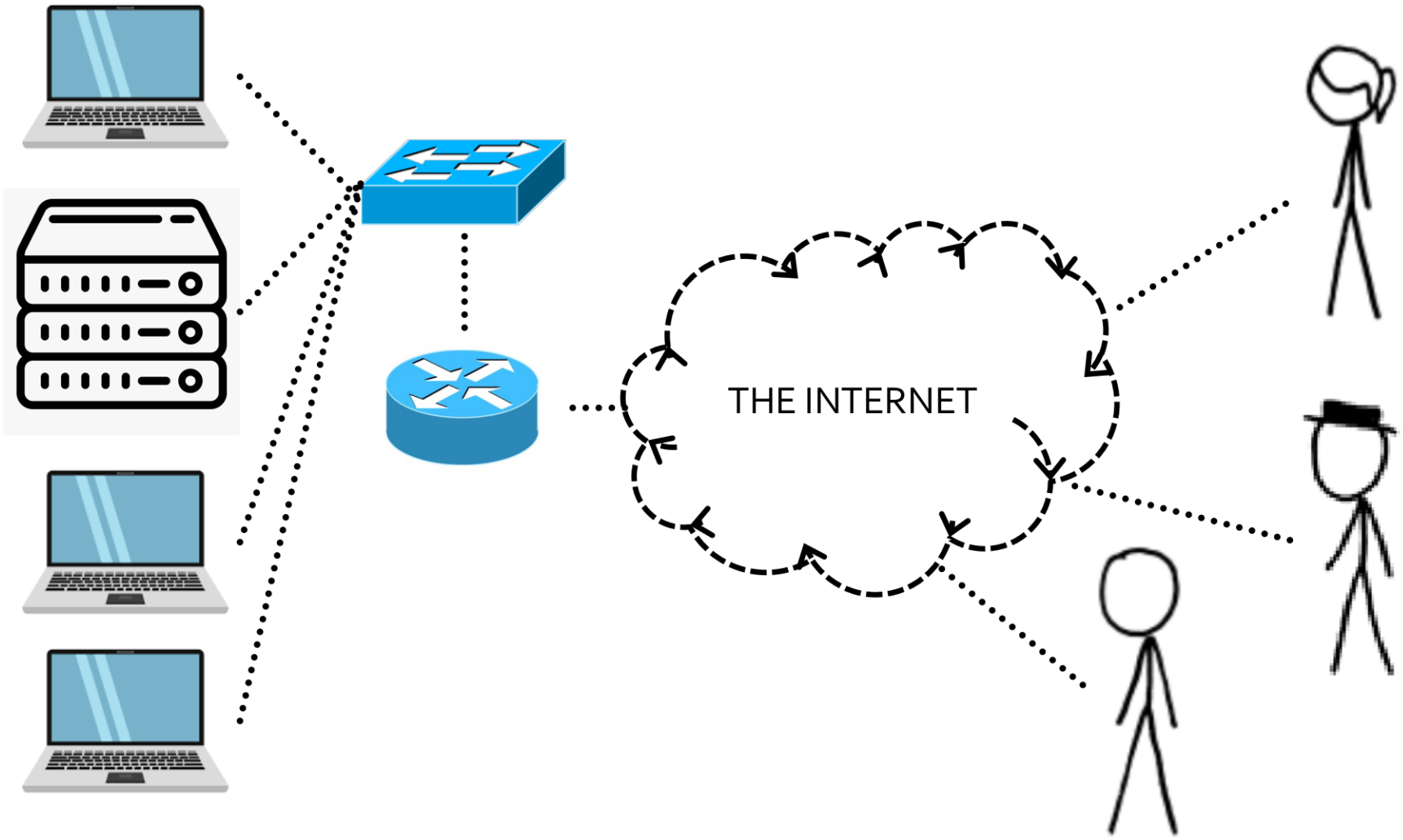
"Sticky" MACs


L2 -- CC:...:CC
L3 – 1.1.1.100
A
1.1.1.100 is me.
I'm at CC:...:CC
1.1.1.100 is me.
I'm at CC:...:CC

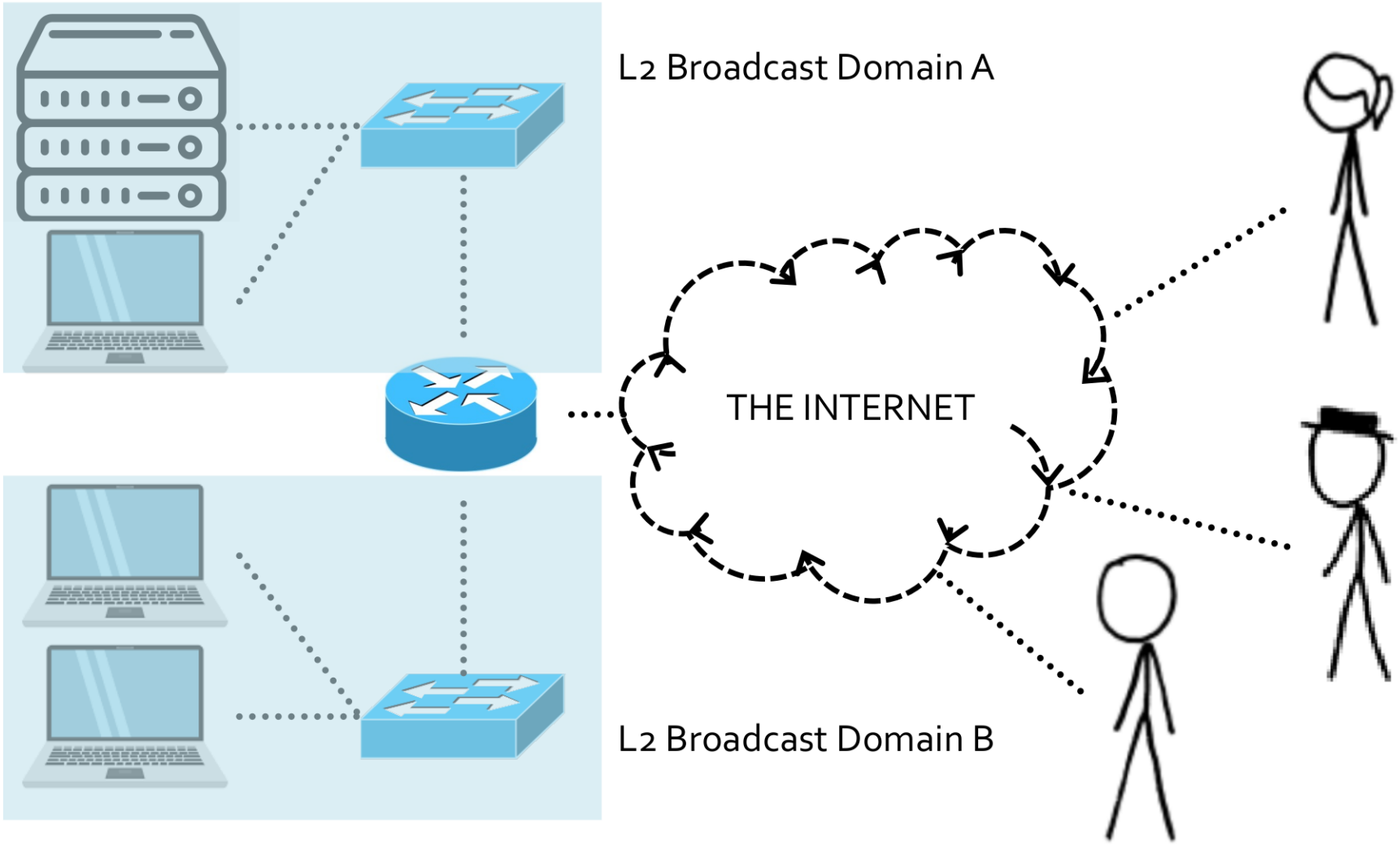
Defending against ARP Attacks

- More intelligent switching
  - "Sticky" MACs: 1 physical port = 1 MAC addr
  - 802.1X: Authenticate physical port access
  - *<many more>*
- More intelligent topology
  - Network partitioning
  - Reduce the "blast radius" of an attack via network segmentation

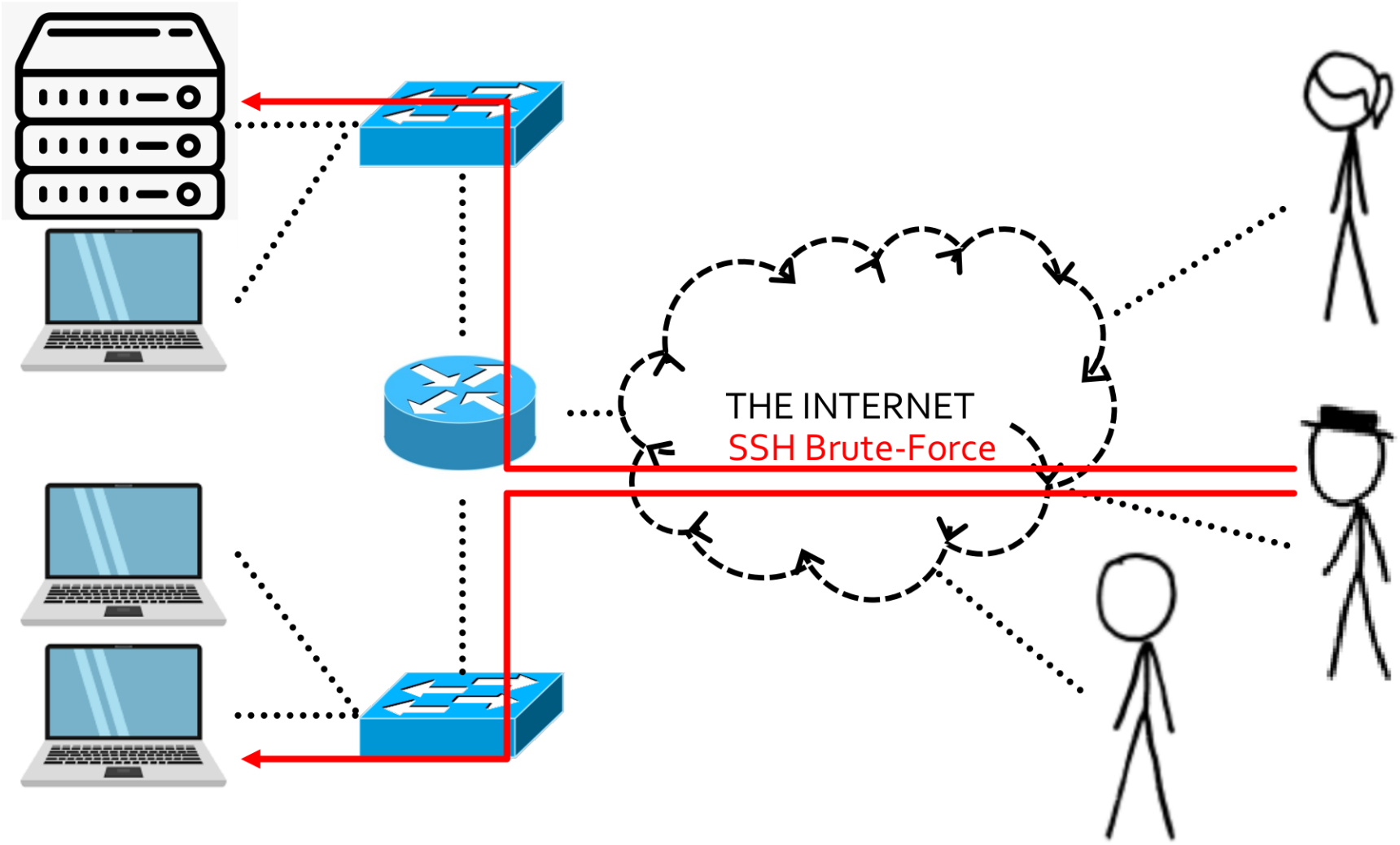
Bad Network Topology



Better Network Topology



External Attacks



Firewalls

A **firewall** is a generic name for a network-level defense tactic that blindly applies a rule-based policy to network traffic.

Firewalls

A **firewall** is a generic name for a network-level defense tactic that blindly applies a rule-based policy to network traffic.

- Operate on L3 and L4 (IPs and TCP/UDP)
- Logically, the rules are straight-forward
  - **DO** allow port 80 (HTTP)
  - **DON'T** allow port 22 (SSH)
  - **DON'T** allow port 21 (FTP) **UNLESS** comes from <*remote office IP*>

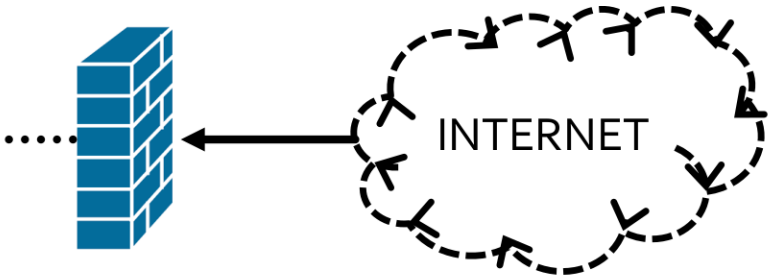
Firewall Implementations

- Blacklisting traffic
  - Match rule? **BLOCK**
- Whitelisting traffic
  - Match rule? **ALLOW**

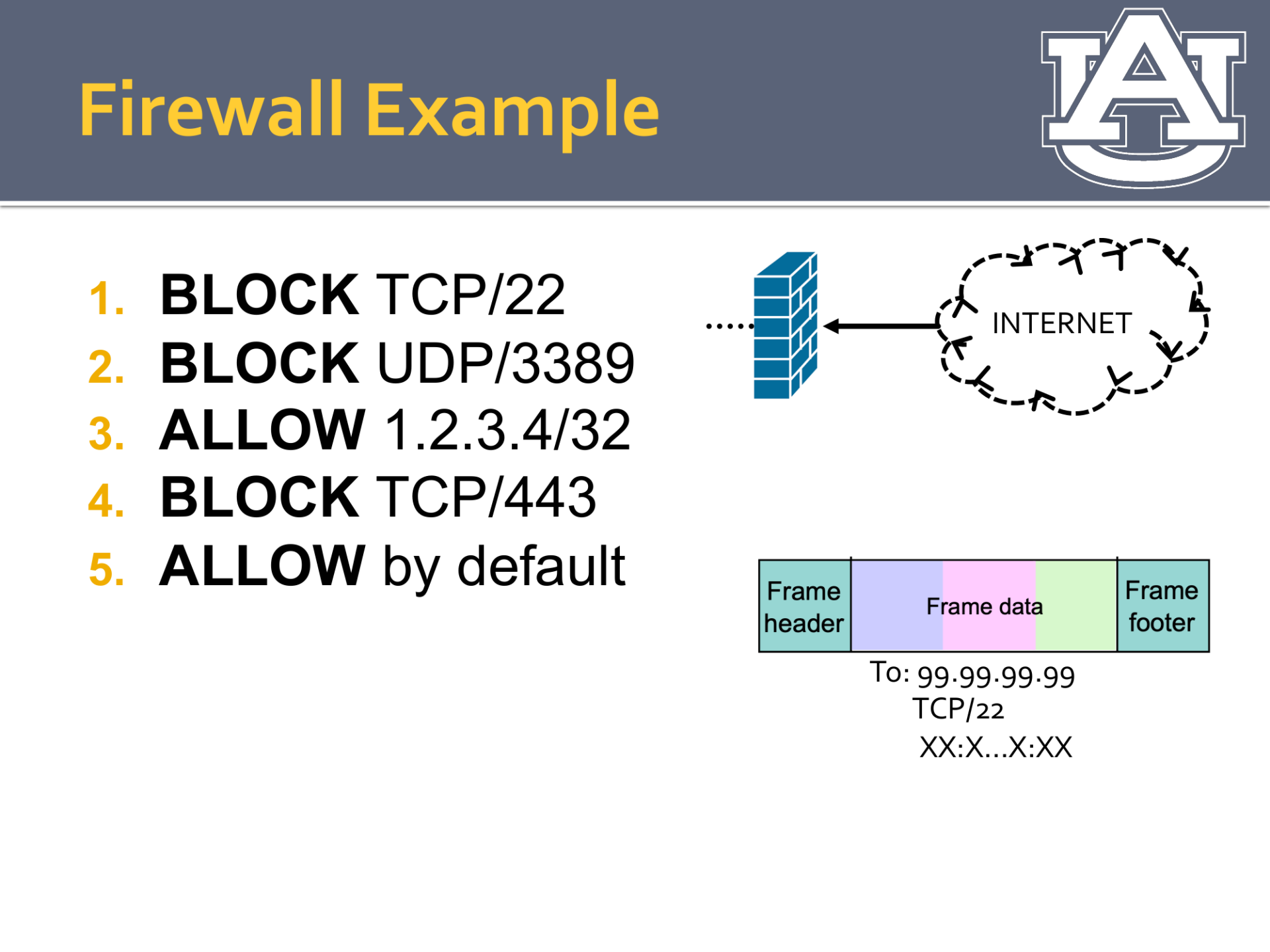
There's always a default action if doesn't match a specific rule!

Firewall Example

1. **BLOCK** TCP/22
2. **BLOCK** UDP/3389
3. **ALLOW** 1.2.3.4/32
4. **BLOCK** TCP/443
5. **ALLOW** by default



Firewall Example

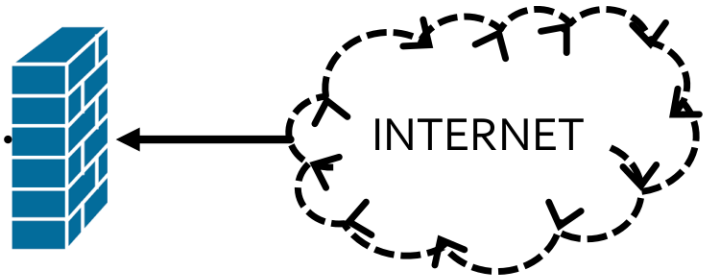


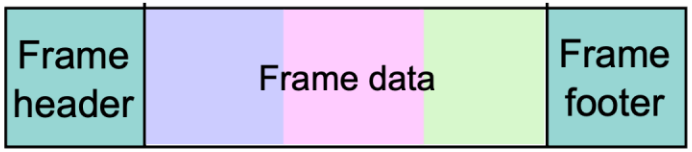
1. **BLOCK** TCP/22
2. **BLOCK** UDP/3389
3. **ALLOW** 1.2.3.4/32
4. **BLOCK** TCP/443
5. **ALLOW** by default

To: 99.99.99.99
TCP/22
XX:X…X:XX

Firewall Example

1. **BLOCK** TCP/22
2. **BLOCK** UDP/3389
3. **ALLOW** 1.2.3.4/32
4. **BLOCK** TCP/443
5. **ALLOW** by default

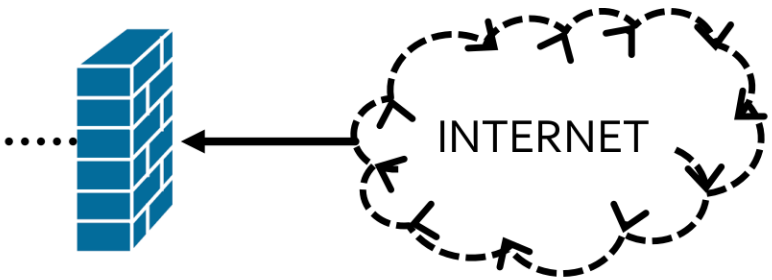


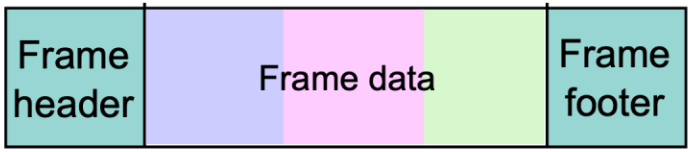


To: 99.99.99.99
TCP/22
XX:X…X:XX

Firewall Example

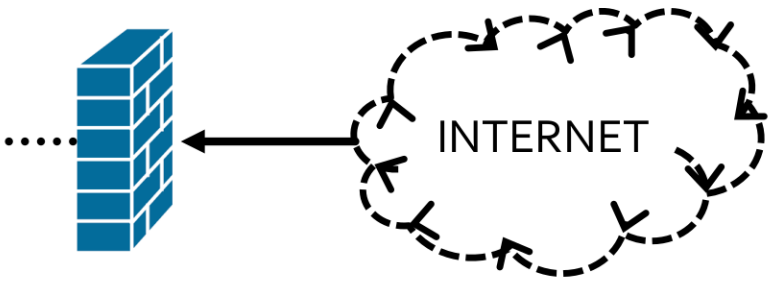
1. **BLOCK** TCP/22
2. **BLOCK** UDP/3389
3. **ALLOW** 1.2.3.4/32
4. **BLOCK** TCP/443
5. **ALLOW** by default



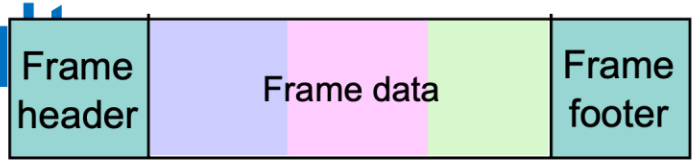


To: 1.42.13.37
UDP/53
XX:X…X:XX

Firewall Example



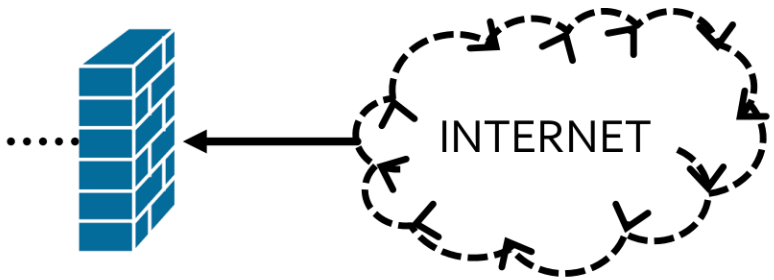
1. **BLOCK** TCP/22
2. **BLOCK** UDP/3389
3. **ALLOW** 1.2.3.4/32
4. **BLOCK** TCP/443
5. **ALLOW** by defau

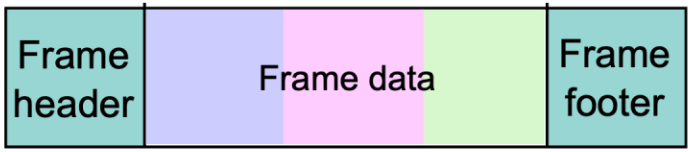


To: 1.42.13.37
UDP/53
XX:X…X:XX

Firewall Example

1. **BLOCK** TCP/22
2. **BLOCK** UDP/3389
3. **ALLOW** 1.2.3.4/32
4. **BLOCK** TCP/443
5. **ALLOW** by default

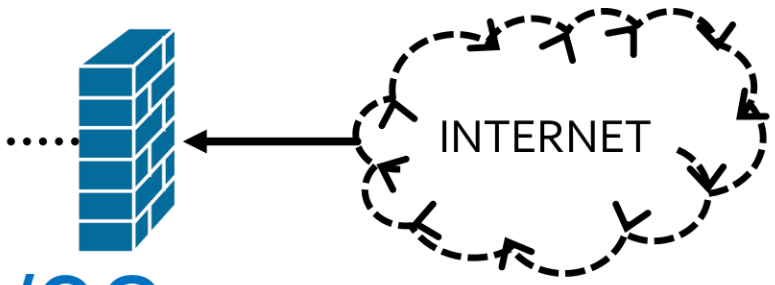


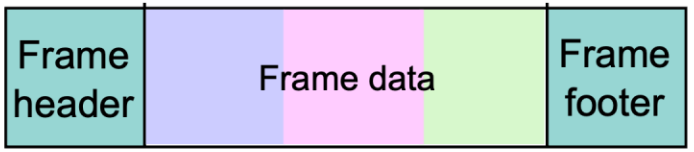


To: 1.2.3.4
TCP/443
XX:X…X:XX

Firewall Example

1. **BLOCK** TCP/22
2. **BLOCK** UDP/3389
3. **ALLOW** 1.2.3.4/32
4. **BLOCK** TCP/443
5. **ALLOW** by default





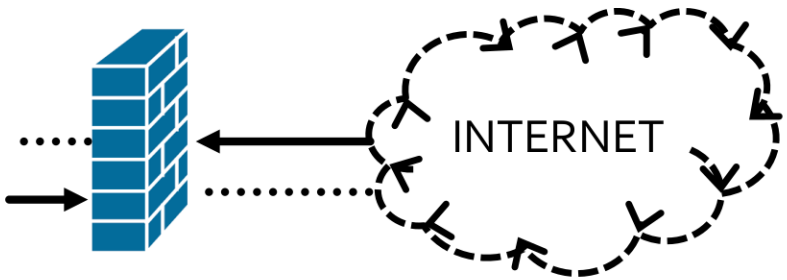
To: 1.2.3.4
TCP/443
XX:X…X:XX

Firewall Implementations

- Blacklisting traffic
  - Match rule? **BLOCK**
- Whitelisting traffic
  - Match rule? **ALLOW**
- Always have a “default rule”

FIREWALL RULES ARE DIRECTIONAL

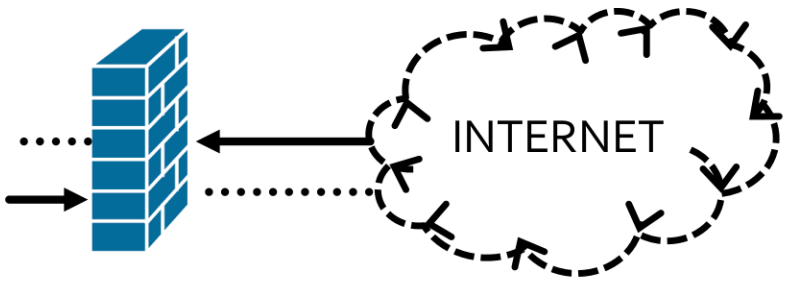
Firewall Example

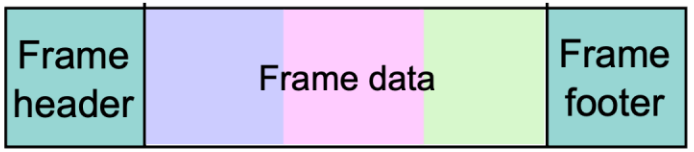


1. **BLOCK** inbound to:TCP/22
2. **BLOCK** inbound to:UDP/3389
3. **ALLOW** inbound to:1.2.3.4/32
4. **BLOCK** inbound to:TCP/443
5. **ALLOW** inbound by default
6. **ALLOW** outbound to:TCP/80
7. **ALLOW** outbound to:UDP/53
8. **BLOCK** outbound to:99.99.99.99/32
9. **ALLOW** outbound to:TCP/443
10. **ALLOW** outbound by default

Firewall Example

1. **BLOCK** inbound to:TCP/22
2. **BLOCK** inbound to:UDP/3389
3. **ALLOW** inbound to:1.2.3.4/32
4. **BLOCK** inbound to:TCP/443
5. **ALLOW** inbound by default
6. **ALLOW** outbound to:TCP/80
7. **ALLOW** outbound to:UDP/53
8. **BLOCK** outbound to:99.99.99.99/32
9. **ALLOW** outbound to:TCP/443
10. **ALLOW** outbound by default





INBOUND

To: 1.2.3.4

TCP/443

XX:X…X:XX

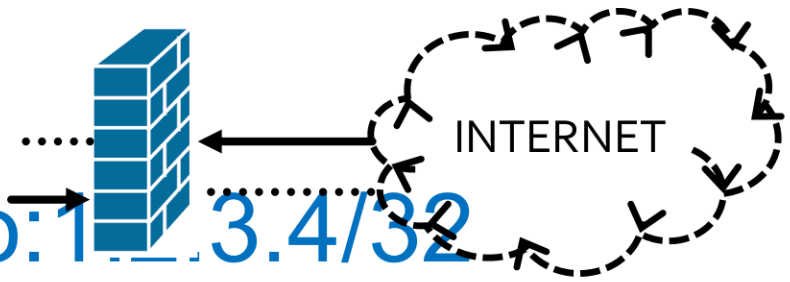
From: 6.7.8.9

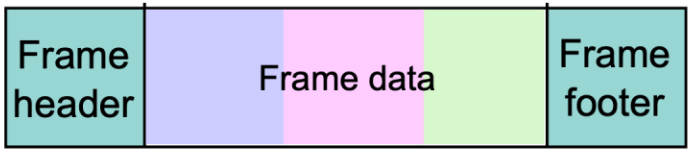
TCP/337

XX:X…X:XX

Firewall Example

1. **BLOCK** inbound to:TCP/22
2. **BLOCK** inbound to:UDP/3389
3. **ALLOW** inbound to:1.2.3.4/32
4. **BLOCK** inbound to:TCP/443
5. **ALLOW** inbound by default
6. **ALLOW** outbound to:TCP/80
7. **ALLOW** outbound to:UDP/53
8. **BLOCK** outbound to:99.99.99.99/32
9. **ALLOW** outbound to:TCP/443
10. **ALLOW** outbound by default

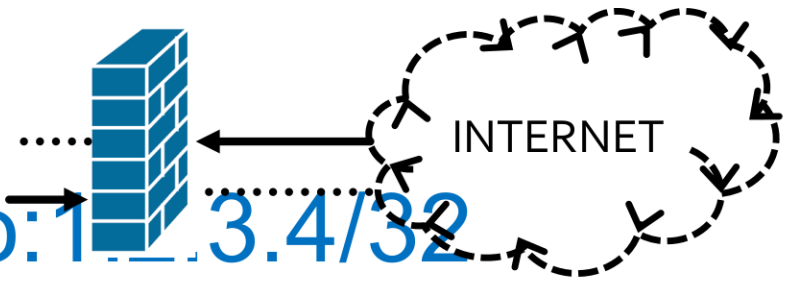


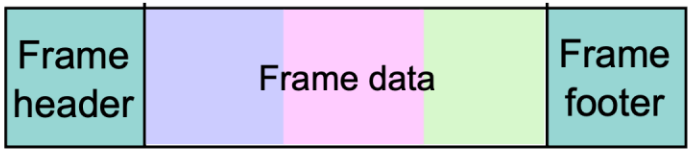


INBOUND
To: 1.2.3.4
TCP/443
XX:X…X:XX
From: 6.7.8.9
TCP/337
XX:X…X:XX

Firewall Example

1. **BLOCK** inbound to:TCP/22
2. **BLOCK** inbound to:UDP/3389
3. **ALLOW** inbound to:1.2.3.4/32
4. **BLOCK** inbound to:TCP/443
5. **ALLOW** inbound by default
6. **ALLOW** outbound to:TCP/80
7. **ALLOW** outbound to:UDP/53
8. **BLOCK** outbound to:99.99.99.99/32
9. **ALLOW** outbound to:TCP/443
10. **ALLOW** outbound by default

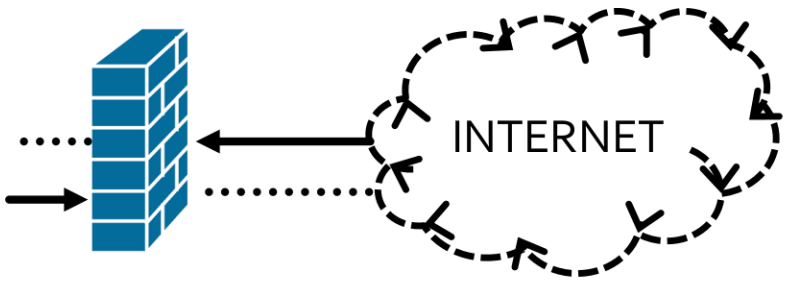


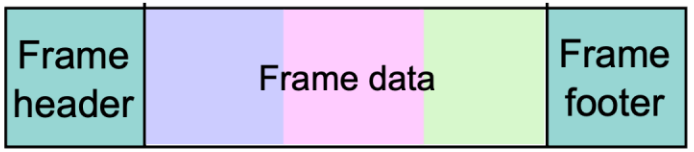


INBOUND
To: 1.2.3.4
TCP/443
XX:X…X:XX
From: 6.7.8.9
TCP/337
XX:X…X:XX

Firewall Example

1. **BLOCK** inbound to:TCP/22
2. **BLOCK** inbound to:UDP/3389
3. **ALLOW** inbound to:1.2.3.4/32
4. **BLOCK** inbound to:TCP/443
5. **ALLOW** inbound by default
6. **ALLOW** outbound to:TCP/80
7. **ALLOW** outbound to:UDP/53
8. **BLOCK** outbound to:99.99.99.99/32
9. **ALLOW** outbound to:TCP/443
10. **ALLOW** outbound by default





OUTBOUND

To: 1.2.3.4

TCP/443

XX:X…X:XX

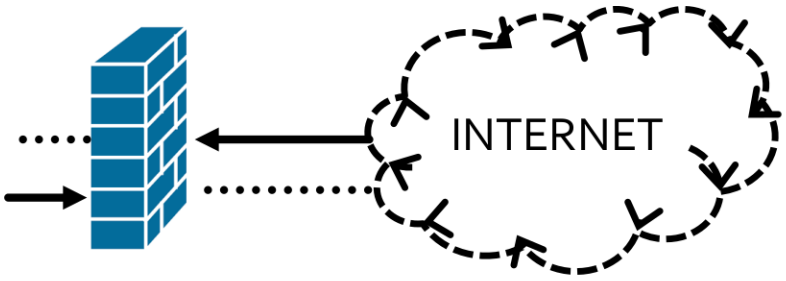
From: 6.7.8.9

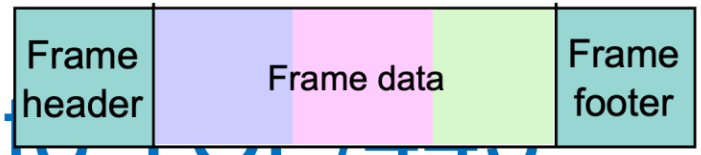
TCP/337

XX:X…X:XX

Firewall Example

1. **BLOCK** inbound to:TCP/22
2. **BLOCK** inbound to:UDP/3389
3. **ALLOW** inbound to:1.2.3.4/32
4. **BLOCK** inbound to:TCP/443
5. **ALLOW** inbound by default
6. **ALLOW** outbound to:TCP/80
7. **ALLOW** outbound to:UDP/53
8. **BLOCK** outbound to:99.99.99.99/32
9. **ALLOW** outbound to:1.2.3.4/443
10. **ALLOW** outbound by default

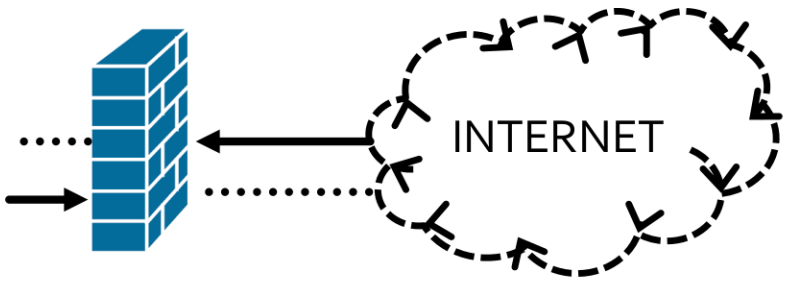


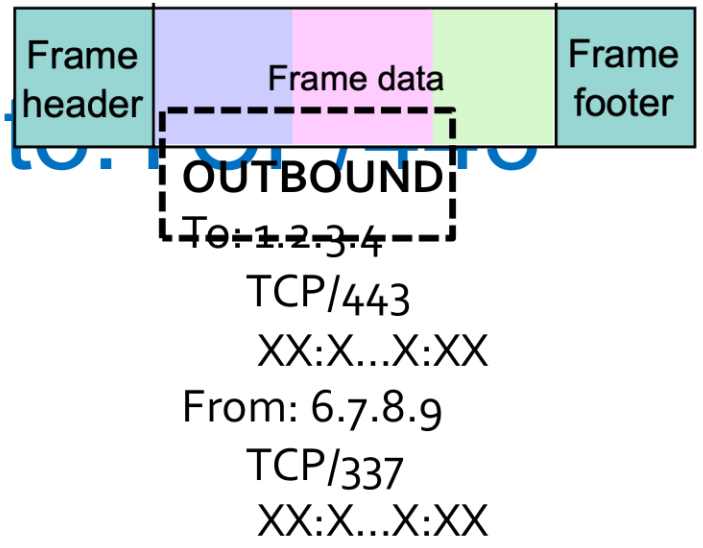


OUTBOUND
To: 1.2.3.4
TCP/443
XX:X…X:XX
From: 6.7.8.9
TCP/337
XX:X…X:XX

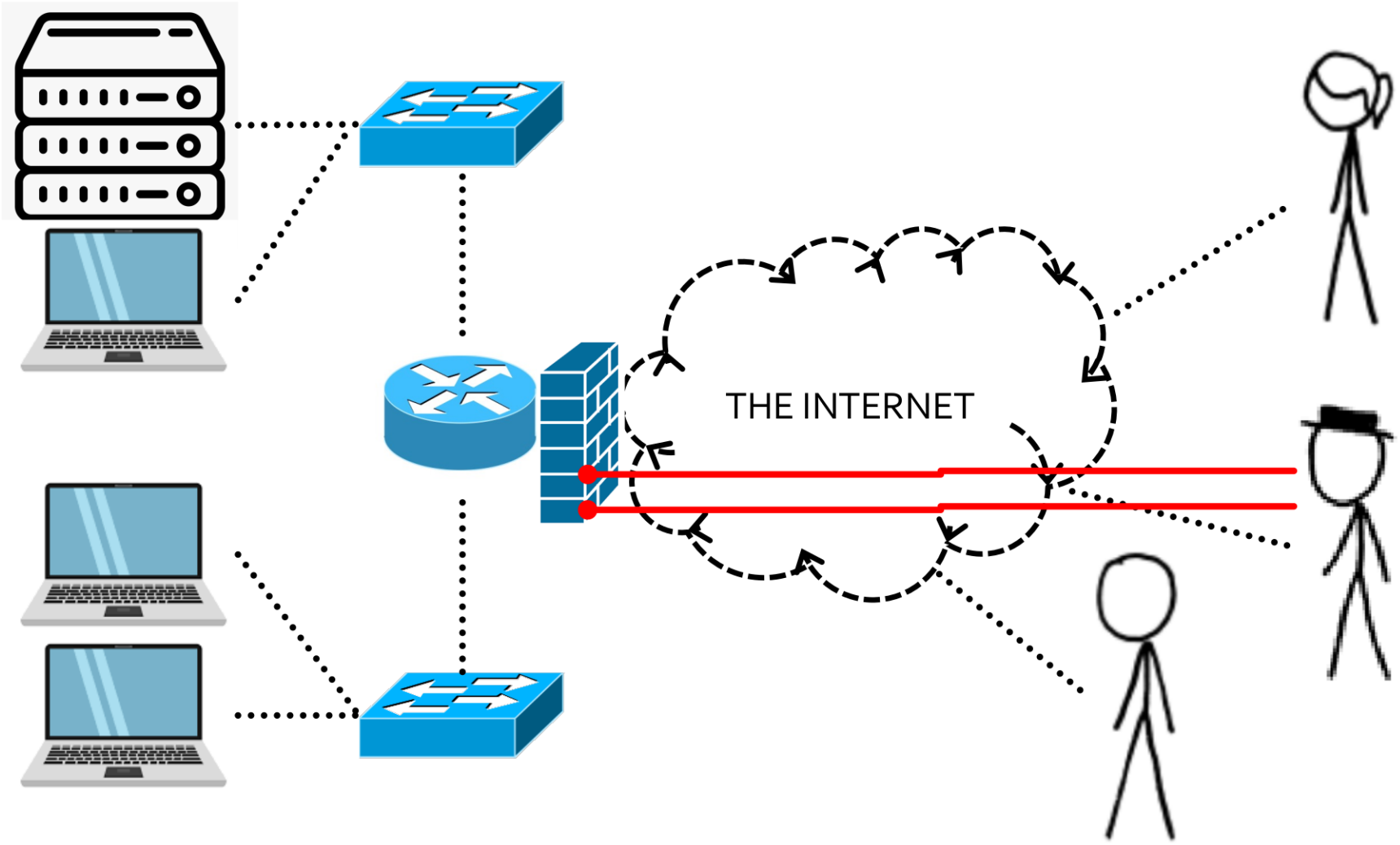
Firewall Example

1. **BLOCK** inbound to:TCP/22
2. **BLOCK** inbound to:UDP/3389
3. **ALLOW** inbound to:1.2.3.4/32
4. **BLOCK** inbound to:TCP/443
5. **ALLOW** inbound by default
6. **ALLOW** outbound to:TCP/80
7. **ALLOW** outbound to:UDP/53
8. **BLOCK** outbound to:99.99.99.99/32
9. **ALLOW** outbound to:1.2.3.4
10. **ALLOW** outbound by default



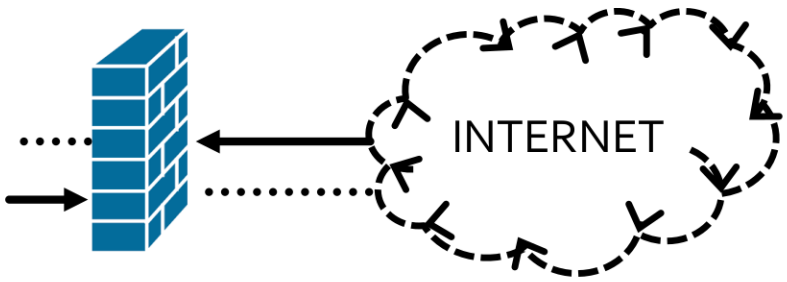


External Attacks



Semi-Realistic Firewall Rules

1. **ALLOW** outbound to:UDP/75
2. **BLOCK** inbound to:TCP/43
3. **ALLOW** inbound to:1.2.3.4/32
4. **BLOCK** inbound to:TCP/443
5. **ALLOW** outbound to:TCP/80
6. **BLOCK** outbound to:99.99.99.99/32
7. **ALLOW** outbound to:TCP/443
8. **BLOCK** inbound to:TCP/389
9. **BLOCK** inbound to:TCP/43
10. **ALLOW** outbound to:UDP/53
11. **BLOCK** outbound to:UDP/53
12. **ALLOW** outbound by default
13. **BLOCK** inbound to:TCP/243
14. **BLOCK** inbound to:TCP/443
15. **BLOCK** inbound to:TCP/694
16. **BLOCK** outbound to:UDP/1111
17. **BLOCK** outbound to:UDP/435
18. **BLOCK** outbound to:UDP/3943
19. **BLOCK** outbound to:UDP/954
20. **ALLOW** inbound by default

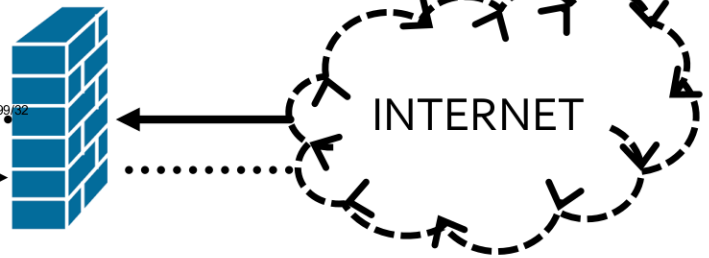


Realistic Firewall Rules

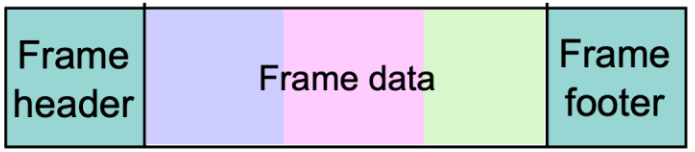
1. **ALLOW** outbound to:UDP/75
2. **BLOCK** inbound to:TCP/43
3. **ALLOW** inbound to:1.2.3.4/32
4. **BLOCK** inbound to:TCP/443
5. **ALLOW** outbound to:TCP/80
6. **BLOCK** outbound to:99.99.99.99/32
7. **ALLOW** outbound to:TCP/443
8. **BLOCK** inbound to:TCP/389
9. **BLOCK** inbound to:TCP/43
10. **ALLOW** outbound to:UDP/53
11. **BLOCK** outbound to:UDP/53
12. **ALLOW** outbound by default
13. **BLOCK** inbound to:TCP/243
14. **BLOCK** inbound to:TCP/443
15. **BLOCK** inbound to:TCP/694
16. **BLOCK** outbound to:UDP/1111
17. **BLOCK** outbound to:UDP/435
18. **BLOCK** outbound to:UDP/3943
19. **BLOCK** outbound to:UDP/954
20. **ALLOW** inbound by default
21. **ALLOW** outbound to:UDP/75
22. **BLOCK** inbound to:TCP/43
23. **ALLOW** inbound to:1.2.3.4/32
24. **BLOCK** inbound to:TCP/443
25. **ALLOW** outbound to:TCP/80
26. **BLOCK** outbound to:99.99.99.99/32
27. **ALLOW** outbound to:TCP/443
28. **BLOCK** inbound to:TCP/389
29. **BLOCK** inbound to:TCP/43
30. **ALLOW** outbound to:UDP/53
31. **BLOCK** outbound to:UDP/53
32. **ALLOW** outbound by default
33. **BLOCK** inbound to:TCP/243
34. **BLOCK** inbound to:TCP/443
35. **BLOCK** inbound to:TCP/694
36. **BLOCK** outbound to:UDP/1111
37. **BLOCK** outbound to:UDP/435
38. **BLOCK** outbound to:UDP/3943
39. **BLOCK** outbound to:UDP/954
40. **ALLOW** inbound by default
41. **ALLOW** outbound to:UDP/75
42. **BLOCK** inbound to:TCP/43
43. **ALLOW** inbound to:1.2.3.4/32
44. **BLOCK** inbound to:TCP/443
45. **ALLOW** outbound to:TCP/80
46. **BLOCK** outbound to:99.99.99.99/32
47. **ALLOW** outbound to:TCP/443
48. **BLOCK** inbound to:TCP/389
49. **BLOCK** inbound to:TCP/43
50. **ALLOW** outbound to:UDP/53
51. **BLOCK** outbound to:UDP/53
52. **ALLOW** outbound by default
53. **BLOCK** inbound to:TCP/243
54. **BLOCK** inbound to:TCP/443
55. **BLOCK** inbound to:TCP/694
56. **BLOCK** outbound to:UDP/1111
57. **BLOCK** outbound to:UDP/435
58. **BLOCK** outbound to:UDP/3943
59. **BLOCK** outbound to:UDP/954
60. **ALLOW** inbound by default

1. **ALLOW** outbound to:UDP/75
2. **BLOCK** inbound to:TCP/43
3. **ALLOW** inbound to:1.2.3.4/32
4. **BLOCK** inbound to:TCP/443
5. **ALLOW** outbound to:TCP/80
6. **BLOCK** outbound to:99.99.99.99/32
7. **ALLOW** outbound to:TCP/443
8. **BLOCK** inbound to:TCP/389
9. **BLOCK** inbound to:TCP/43
10. **ALLOW** outbound to:UDP/53
11. **BLOCK** outbound to:UDP/53
12. **ALLOW** outbound by default
13. **BLOCK** inbound to:TCP/243
14. **BLOCK** inbound to:TCP/443
15. **BLOCK** inbound to:TCP/694
16. **BLOCK** outbound to:UDP/1111
17. **BLOCK** outbound to:UDP/435
18. **BLOCK** outbound to:UDP/3943
19. **BLOCK** outbound to:UDP/954
20. **ALLOW** inbound by default
21. **ALLOW** outbound to:UDP/75
22. **BLOCK** inbound to:TCP/43
23. **ALLOW** inbound to:1.2.3.4/32
24. **BLOCK** inbound to:TCP/443
25. **ALLOW** outbound to:TCP/80
26. **BLOCK** outbound to:99.99.99.99/32
27. **ALLOW** outbound to:TCP/443
28. **BLOCK** inbound to:TCP/389
29. **BLOCK** inbound to:TCP/43
30. **ALLOW** outbound to:UDP/53
31. **BLOCK** outbound to:UDP/53
32. **ALLOW** outbound by default
33. **BLOCK** inbound to:TCP/243
34. **BLOCK** inbound to:TCP/443
35. **BLOCK** inbound to:TCP/694
36. **BLOCK** outbound to:UDP/1111
37. **BLOCK** outbound to:UDP/435
38. **BLOCK** outbound to:UDP/3943
39. **BLOCK** outbound to:UDP/954
40. **ALLOW** inbound by default
41. **ALLOW** outbound to:UDP/75
42. **BLOCK** inbound to:TCP/43
43. **ALLOW** inbound to:1.2.3.4/32
44. **BLOCK** inbound to:TCP/443
45. **ALLOW** outbound to:TCP/80
46. **BLOCK** outbound to:99.99.99.99/32
47. **ALLOW** outbound to:TCP/443
48. **BLOCK** inbound to:TCP/389
49. **BLOCK** inbound to:TCP/43
50. **ALLOW** outbound to:UDP/53
51. **BLOCK** outbound to:UDP/53
52. **ALLOW** outbound by default
53. **BLOCK** inbound to:TCP/243
54. **BLOCK** inbound to:TCP/443
55. **BLOCK** inbound to:TCP/694
56. **BLOCK** outbound to:UDP/1111
57. **BLOCK** outbound to:UDP/435
58. **BLOCK** outbound to:UDP/3943
59. **BLOCK** outbound to:UDP/954
60. **ALLOW** inbound by default

1. **ALLOW** outbound to:UDP/75
2. **BLOCK** inbound to:TCP/43
3. **ALLOW** inbound to:1.2.3.4/32
4. **BLOCK** inbound to:TCP/443
5. **ALLOW** outbound to:TCP/80
6. **BLOCK** outbound to:99.99.99.99/32
7. **ALLOW** outbound to:TCP/443
8. **BLOCK** inbound to:TCP/389
9. **BLOCK** inbound to:TCP/43
10. **ALLOW** outbound to:UDP/53
11. **BLOCK** outbound to:UDP/53
12. **ALLOW** outbound by default
13. **BLOCK** inbound to:TCP/243
14. **BLOCK** inbound to:TCP/443
15. **BLOCK** inbound to:TCP/694
16. **BLOCK** outbound to:UDP/1111
17. **BLOCK** outbound to:UDP/435
18. **BLOCK** outbound to:UDP/3943
19. **BLOCK** outbound to:UDP/954
20. **ALLOW** inbound by default
21. **ALLOW** outbound to:UDP/75
22. **BLOCK** inbound to:TCP/43
23. **ALLOW** inbound to:1.2.3.4/32
24. **BLOCK** inbound to:TCP/443
25. **ALLOW** outbound to:TCP/80
26. **BLOCK** outbound to:99.99.99.99/32
27. **ALLOW** outbound to:TCP/443
28. **BLOCK** inbound to:TCP/389
29. **BLOCK** inbound to:TCP/43
30. **ALLOW** outbound to:UDP/53
31. **BLOCK** outbound to:UDP/53
32. **ALLOW** outbound by default
33. **BLOCK** inbound to:TCP/243
34. **BLOCK** inbound to:TCP/443
35. **BLOCK** inbound to:TCP/694
36. **BLOCK** outbound to:UDP/1111
37. **BLOCK** outbound to:UDP/435
38. **BLOCK** outbound to:UDP/3943
39. **BLOCK** outbound to:UDP/954
40. **ALLOW** inbound by default
41. **ALLOW** outbound to:UDP/75
42. **BLOCK** inbound to:TCP/43
43. **ALLOW** inbound to:1.2.3.4/32
44. **BLOCK** inbound to:TCP/443
45. **ALLOW** outbound to:TCP/80
46. **BLOCK** outbound to:99.99.99.99/32
47. **ALLOW** outbound to:TCP/443
48. **BLOCK** inbound to:TCP/389
49. **BLOCK** inbound to:TCP/43
50. **ALLOW** outbound to:UDP/53
51. **BLOCK** outbound to:UDP/53
52. **ALLOW** outbound by default
53. **BLOCK** inbound to:TCP/243
54. **BLOCK** inbound to:TCP/443
55. **BLOCK** inbound to:TCP/694
56. **BLOCK** outbound to:UDP/1111
57. **BLOCK** outbound to:UDP/435
58. **BLOCK** outbound to:UDP/3943
59. **BLOCK** outbound to:UDP/954
60. **ALLOW** inbound by default



1. **ALLOW** outbound to:UDP/75
2. **BLOCK** inbound to:TCP/43
3. **ALLOW** inbound to:1.2.3.4/32
4. **BLOCK** inbound to:TCP/443
5. **ALLOW** outbound to:TCP/80
6. **BLOCK** outbound to:99.99.99.99/32
7. **ALLOW** outbound to:TCP/443
8. **BLOCK** inbound to:TCP/389
9. **BLOCK** inbound to:TCP/43
10. **ALLOW** outbound to:UDP/53
11. **BLOCK** outbound to:UDP/53
12. **ALLOW** outbound by default
13. **BLOCK** inbound to:TCP/243
14. **BLOCK** inbound to:TCP/443
15. **BLOCK** inbound to:TCP/694
16. **BLOCK** outbound to:UDP/1111
17. **BLOCK** outbound to:UDP/435
18. **BLOCK** outbound to:UDP/3943
19. **BLOCK** outbound to:UDP/954
20. **ALLOW** inbound by default
21. **ALLOW** outbound to:UDP/75
22. **BLOCK** inbound to:TCP/43
23. **ALLOW** inbound to:1.2.3.4/32
24. **BLOCK** inbound to:TCP/443
25. **ALLOW** outbound to:TCP/80
26. **BLOCK** outbound to:99.99.99.99/32
27. **ALLOW** outbound to:TCP/443
28. **BLOCK** inbound to:TCP/389
29. **BLOCK** inbound to:TCP/43
30. **ALLOW** outbound to:UDP/53
31. **BLOCK** outbound to:UDP/53
32. **ALLOW** outbound by default
33. **BLOCK** inbound to:TCP/243
34. **BLOCK** inbound to:TCP/443
35. **BLOCK** inbound to:TCP/694
36. **BLOCK** outbound to:UDP/1111
37. **BLOCK** outbound to:UDP/435
38. **BLOCK** outbound to:UDP/3943
39. **BLOCK** outbound to:UDP/954
40. **ALLOW** inbound by default
41. **ALLOW** outbound to:UDP/75
42. **BLOCK** inbound to:TCP/43
43. **ALLOW** inbound to:1.2.3.4/32
44. **BLOCK** inbound to:TCP/443
45. **ALLOW** outbound to:TCP/80
46. **BLOCK** outbound to:99.99.99.99/32
47. **ALLOW** outbound to:TCP/443
48. **BLOCK** inbound to:TCP/389
49. **BLOCK** inbound to:TCP/43
50. **ALLOW** outbound to:UDP/53
51. **BLOCK** outbound to:UDP/53
52. **ALLOW** outbound by default
53. **BLOCK** inbound to:TCP/243
54. **BLOCK** inbound to:TCP/443
55. **BLOCK** inbound to:TCP/694
56. **BLOCK** outbound to:UDP/1111
57. **BLOCK** outbound to:UDP/435
58. **BLOCK** outbound to:UDP/3943
59. **BLOCK** outbound to:UDP/954
60. **ALLOW** inbound by default



OUTBOUND

To: 3.43.43.53

TCP/133

XX:X…X:XX

From: 48.5.84.66

TCP/4935

XX:X…X:XX

Computer and Network Security

Lecture 16: Network Background/Attacks

COMP-5370/6370
Fall 2025