CS 43: Computer Networks
Putting it all together…

Kevin Webb

Swarthmore College

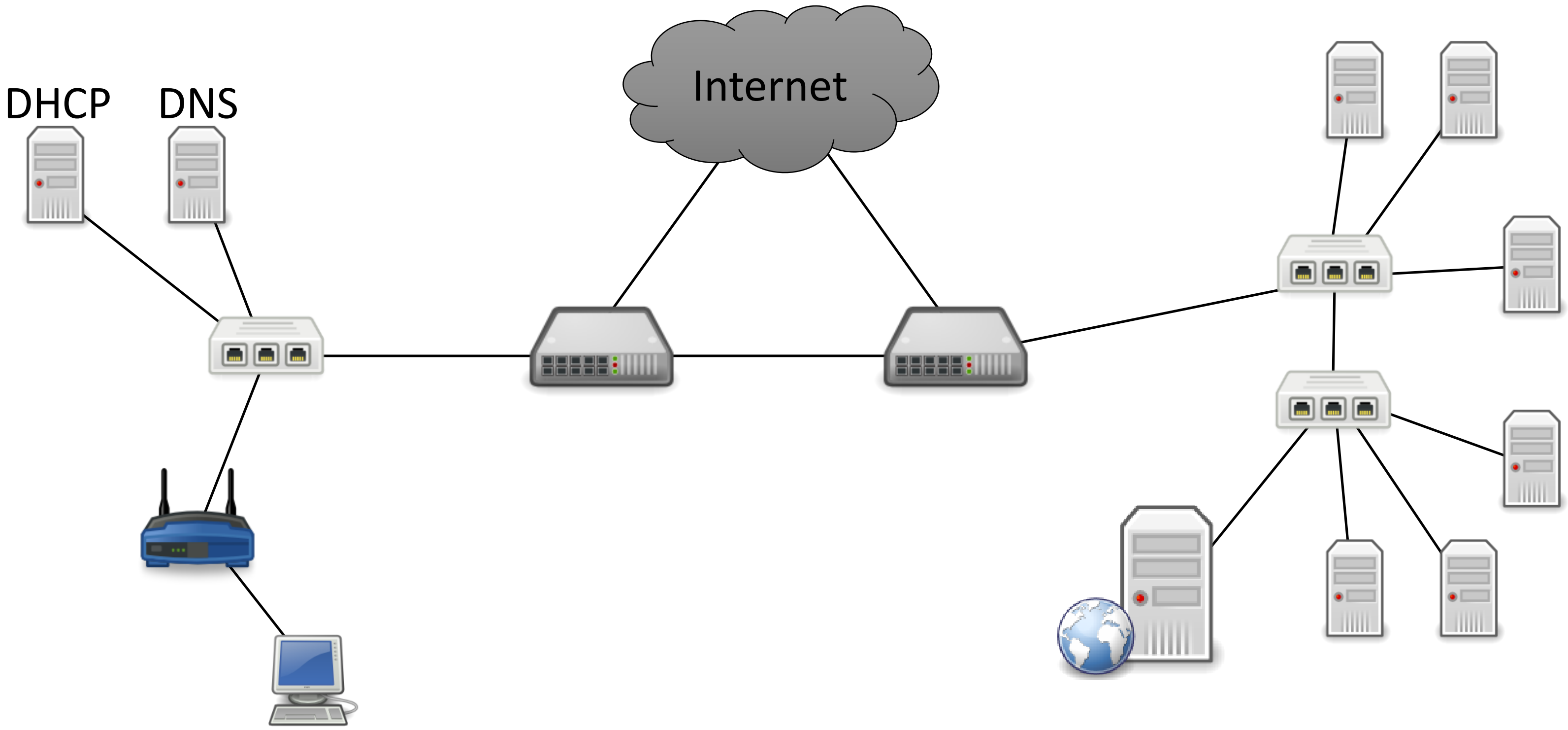
April 28, 2022

Putting it all together…

- What happens when a user shows up to a new network and wants to access a web site?

(Please stop and ask questions if anything is unclear!)

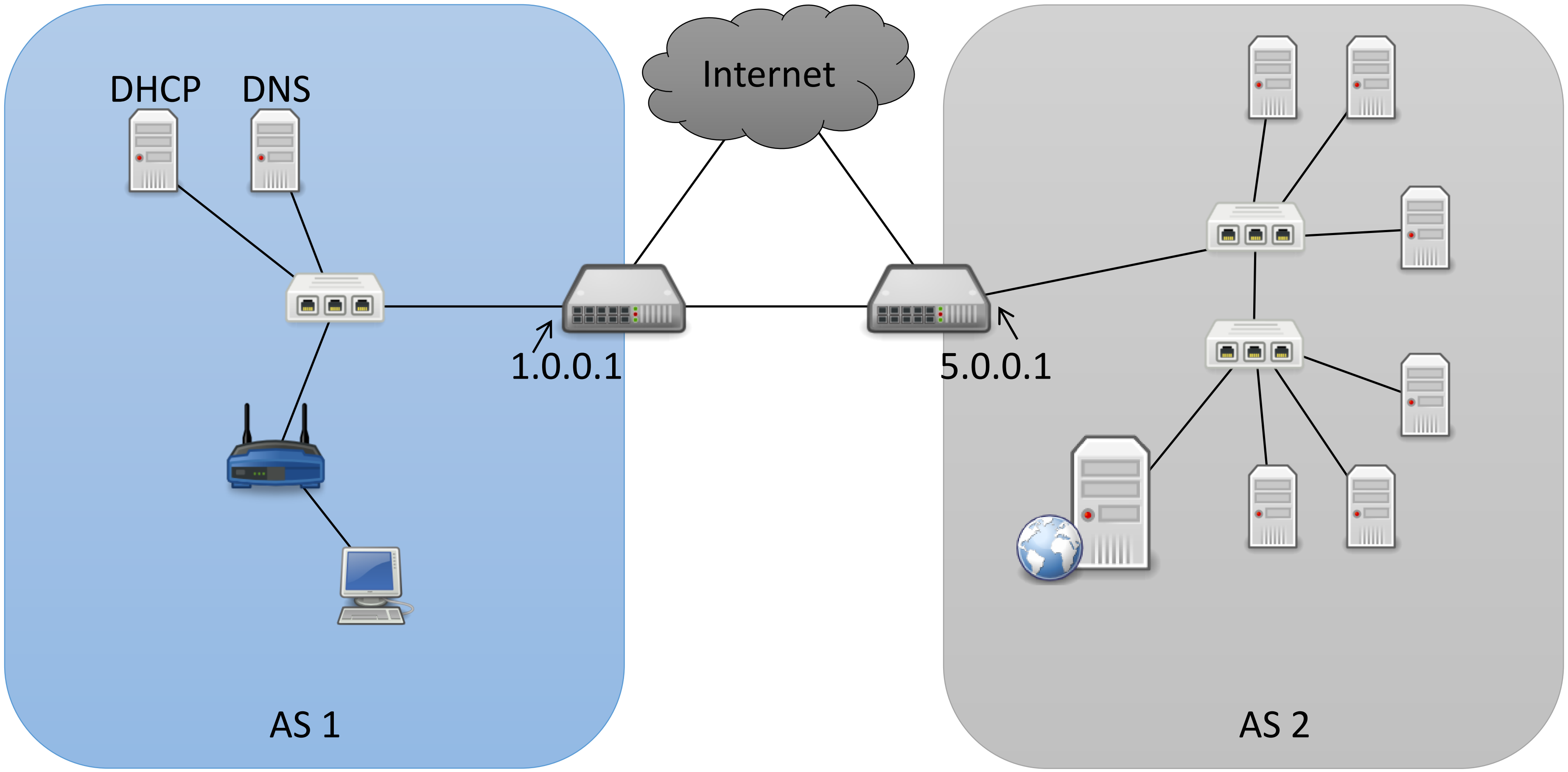
Scenario



Scenario

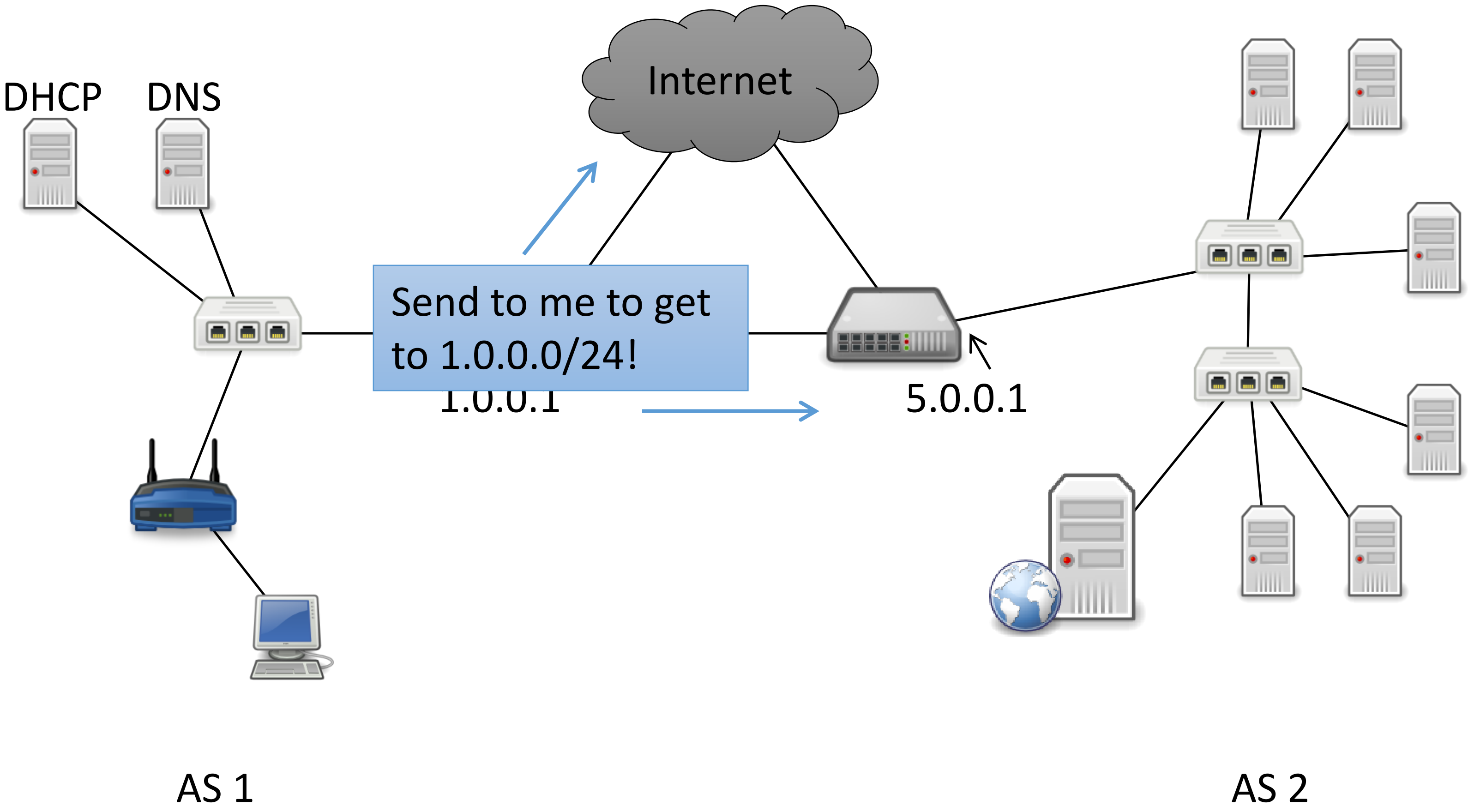
Network: 1.0.0.0/24
24 bits: network
8 bits: host

Network: 5.0.0.0/16
16 bits: network
16 bits: host



Step 0: Routing Protocol

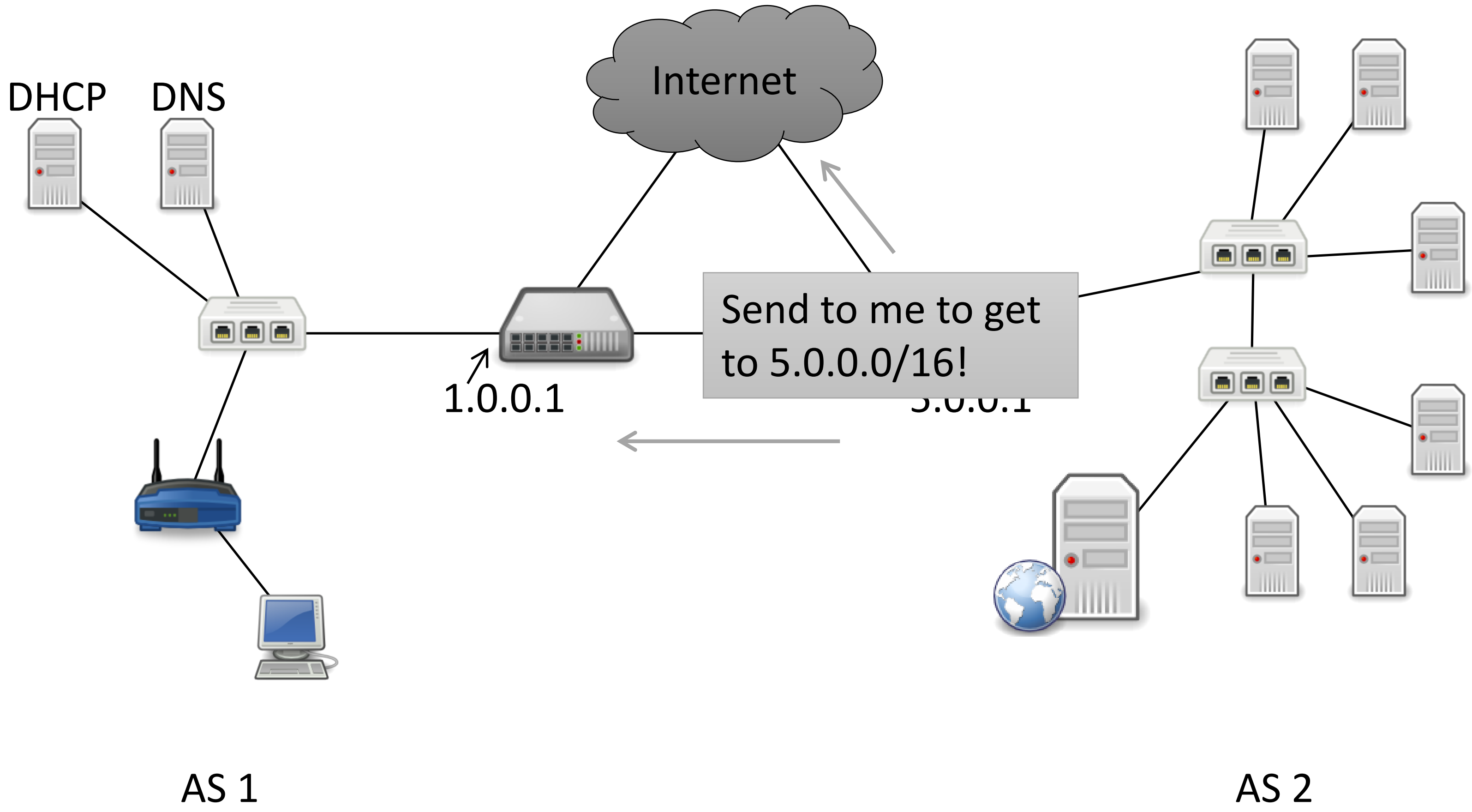
Before anyone starts sending data, we'll assume the routers have run a routing protocol (BGP) to learn about each other.



Send to me to get to 1.0.0.0/24!

Step 0: Routing Protocol

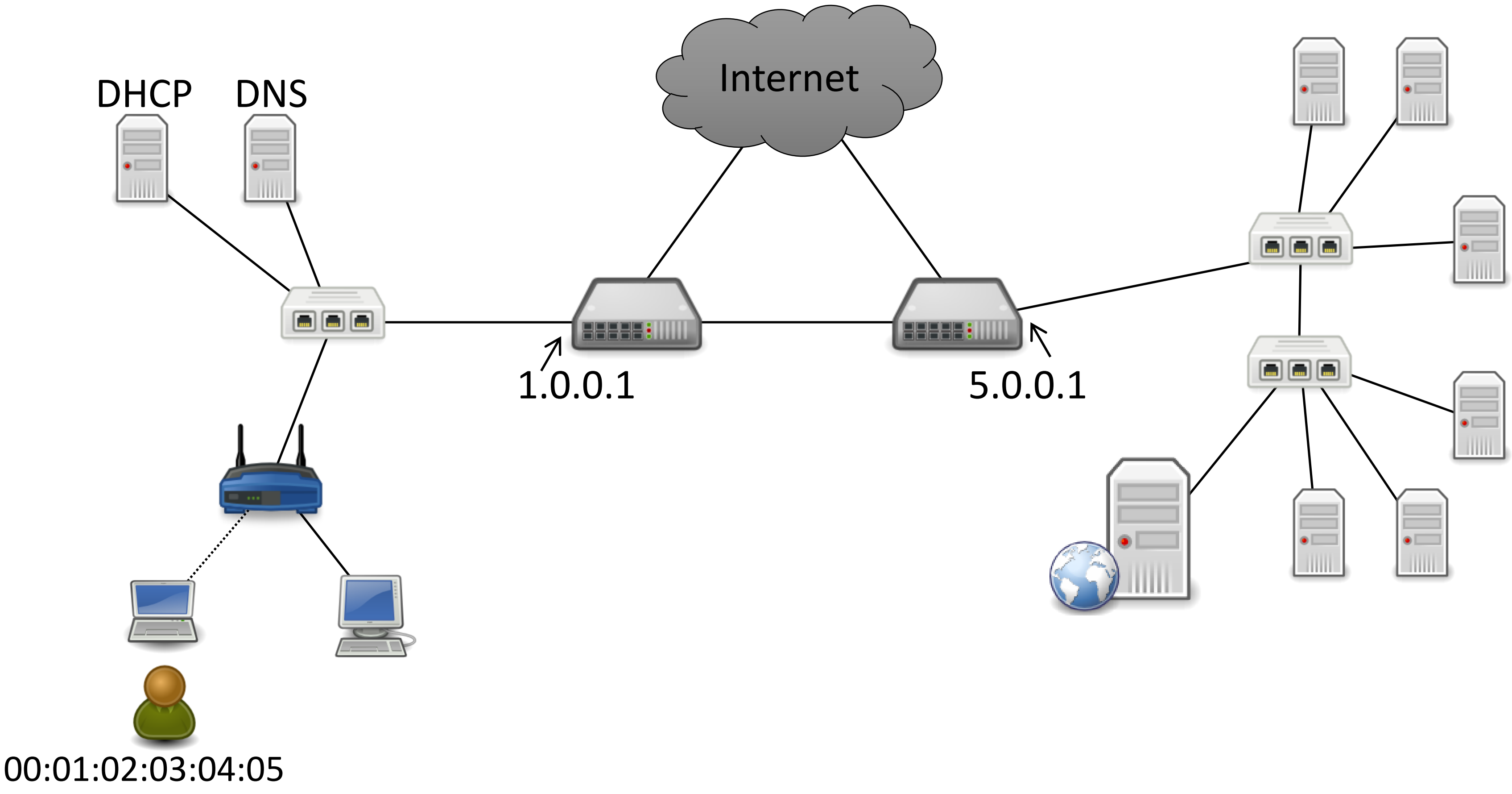
Before anyone starts sending data, we'll assume the routers have run a routing protocol (BGP) to learn about each other.



Send to me to get to 5.0.0.0/16!

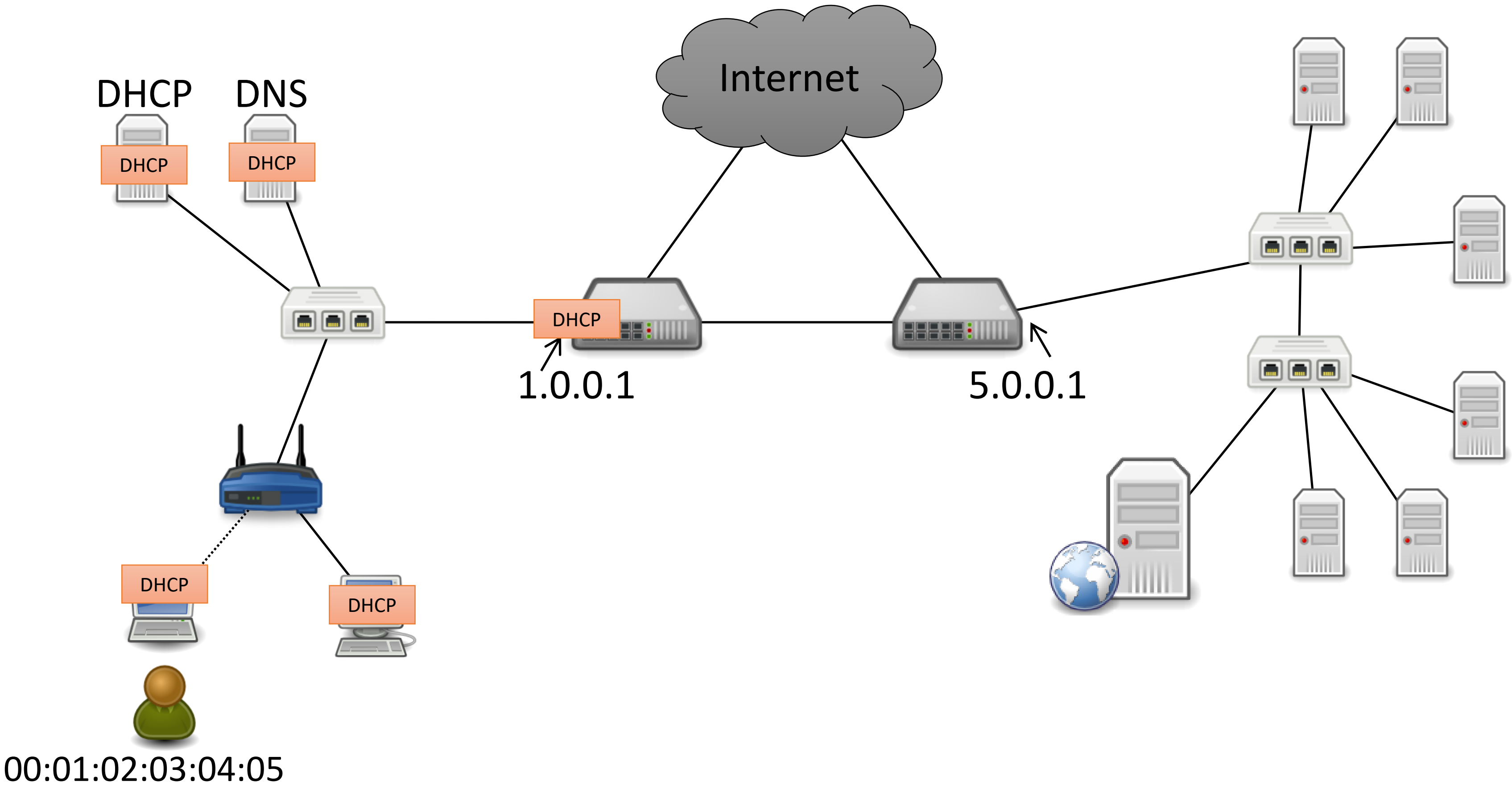
Step 1: User Joins Network

User arrives and needs an IP address.
They bring MAC address with them (built in to hardware).



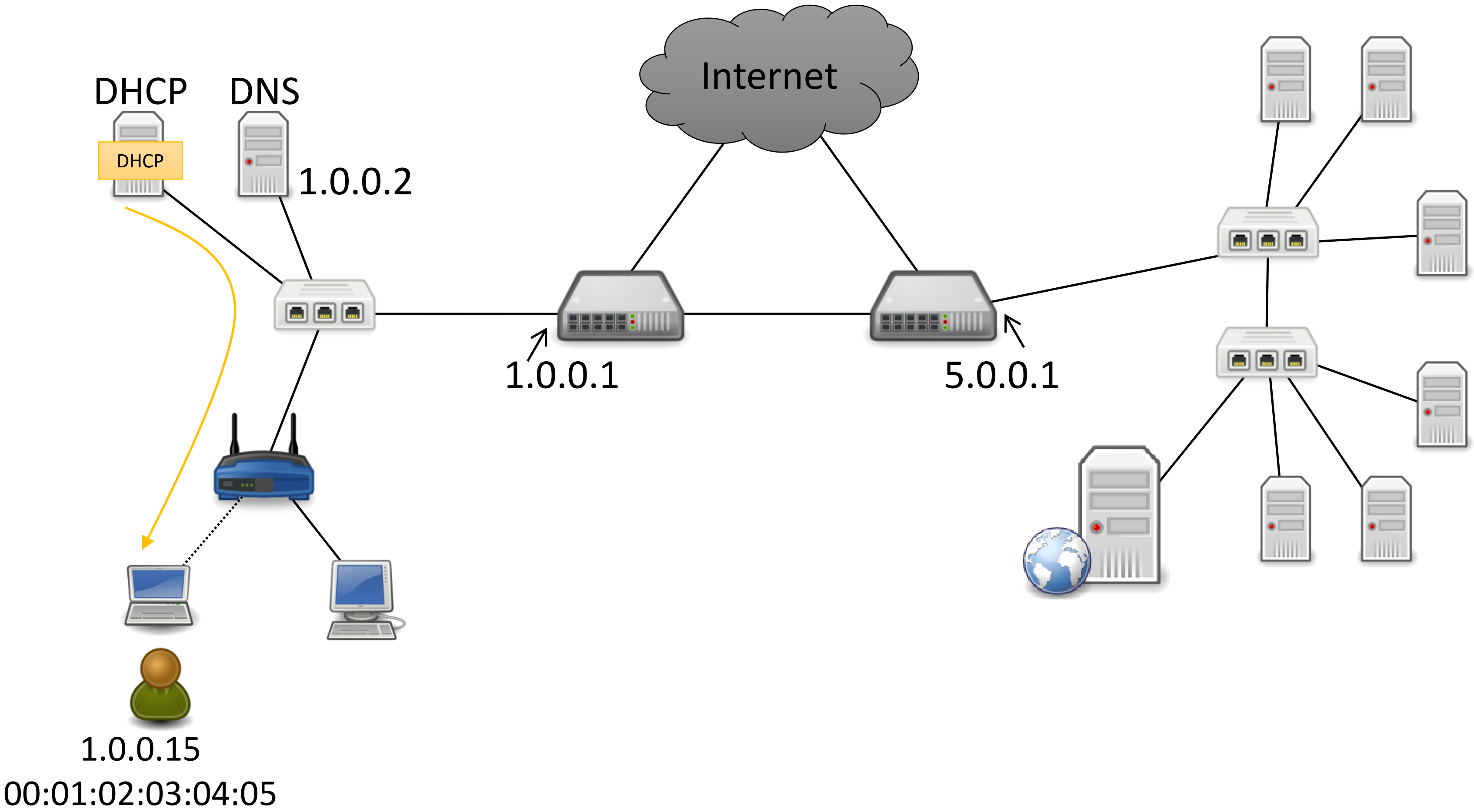
Step 1: User Joins Network

User broadcasts DHCP DISCOVER message to acquire IP address. (Alternative, they manually enter IP config details.)



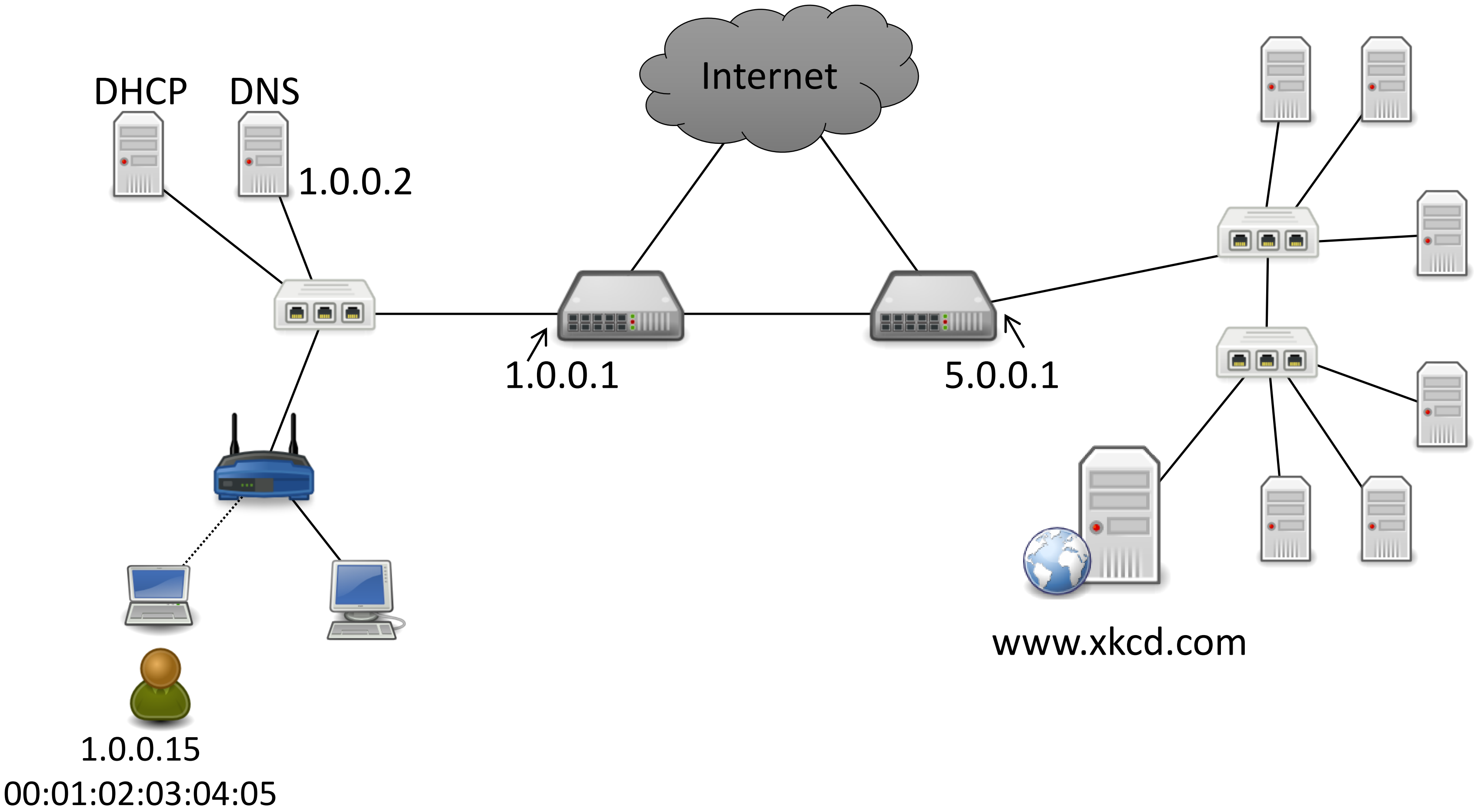
Step 1: User Joins Network

DHCP server responds with: IP address (1.0.0.15), subnet mask (255.255.255.0), gateway (1.0.0.1), and DNS server (1.0.0.2).



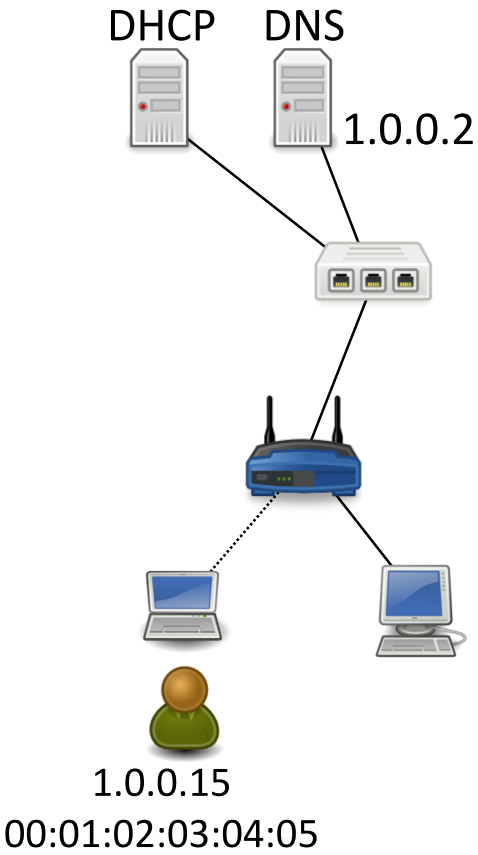
Step 2: User Resolves Name

Suppose user tries to access website: www.xkcd.com
Must resolve name using DNS. Query local resolver.



Step 2: User Resolves Name

User’s PC must answer: is the DNS resolver (1.0.0.2) I was given by DHCP server on my subnet? (Local vs. Internet)



my address

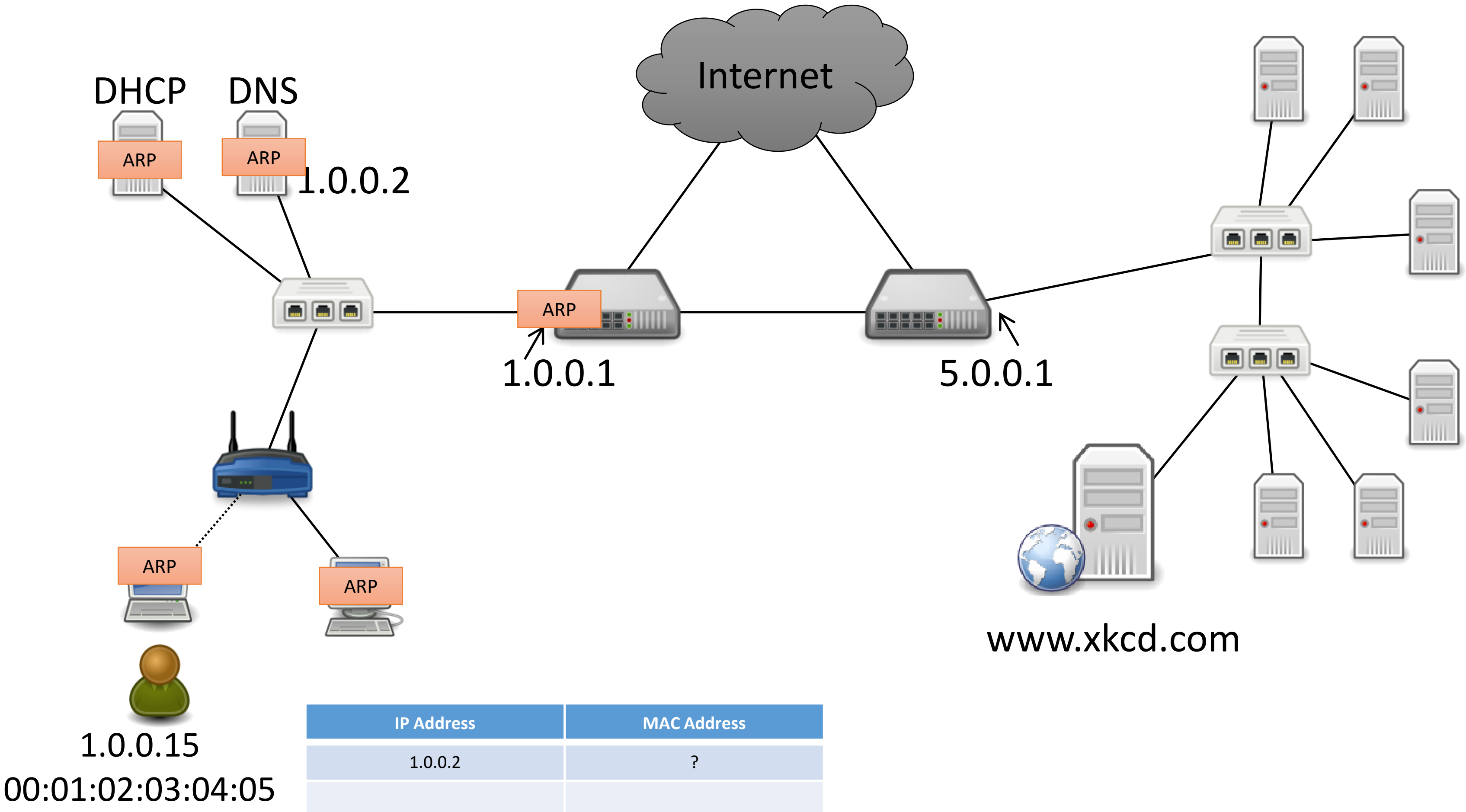
subnet mask

1.0.0.15:	00000001	00000000	00000000	00001111
255.255.255.0:	11111111	11111111	11111111	00000000
ANDed together: my network prefix	00000001	00000000	00000000	
	‖	‖	‖	
target address 1.0.0.2	00000001	00000000	00000000	00000010

Match! It’s local. Send directly, no need to go through Internet gateway (router).

Step 2: User Resolves Name

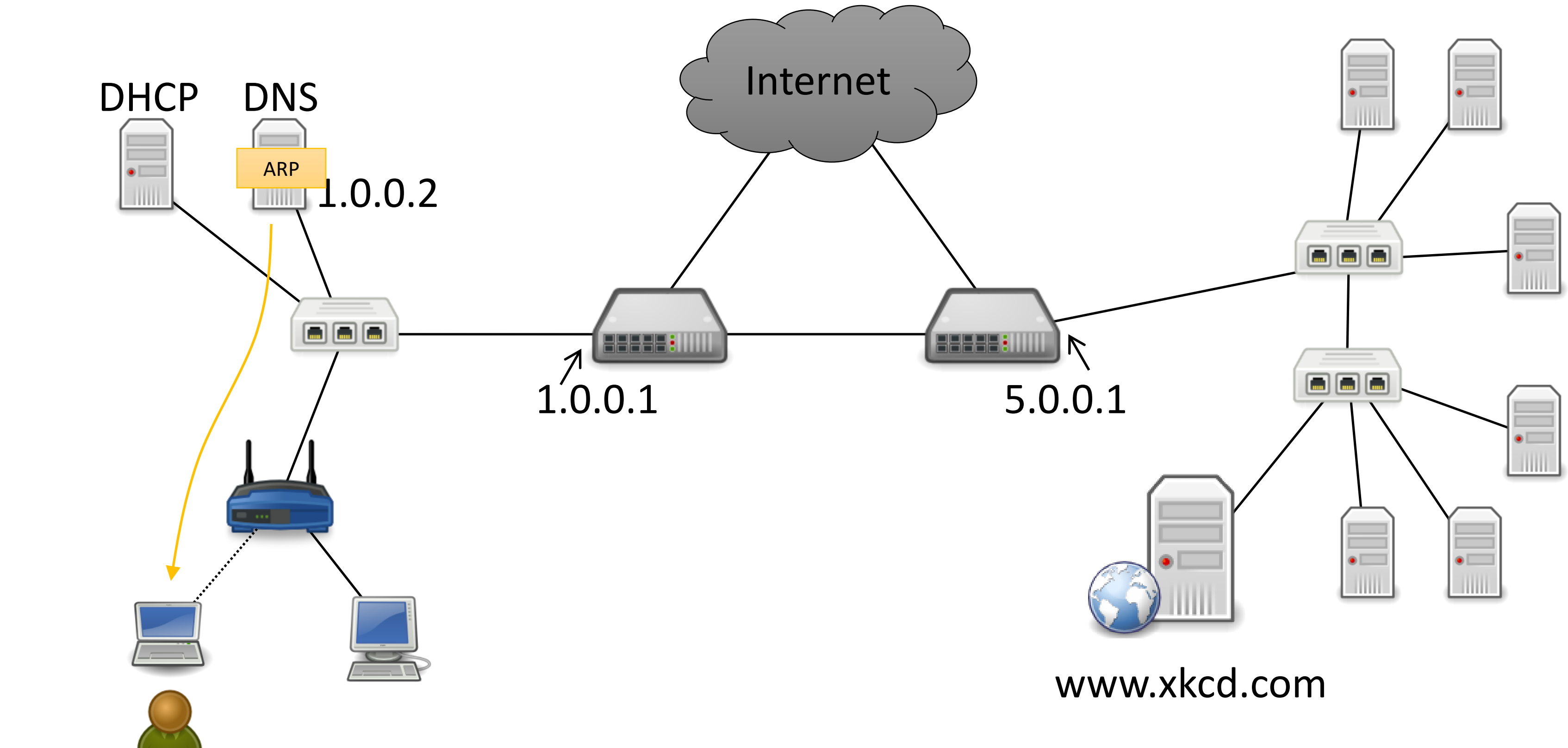
User’s PC does NOT know DNS server’s MAC address!
Broadcast ARP request looking for 1.0.0.2!



IP Address	MAC Address
1.0.0.2	?

Step 2: User Resolves Name

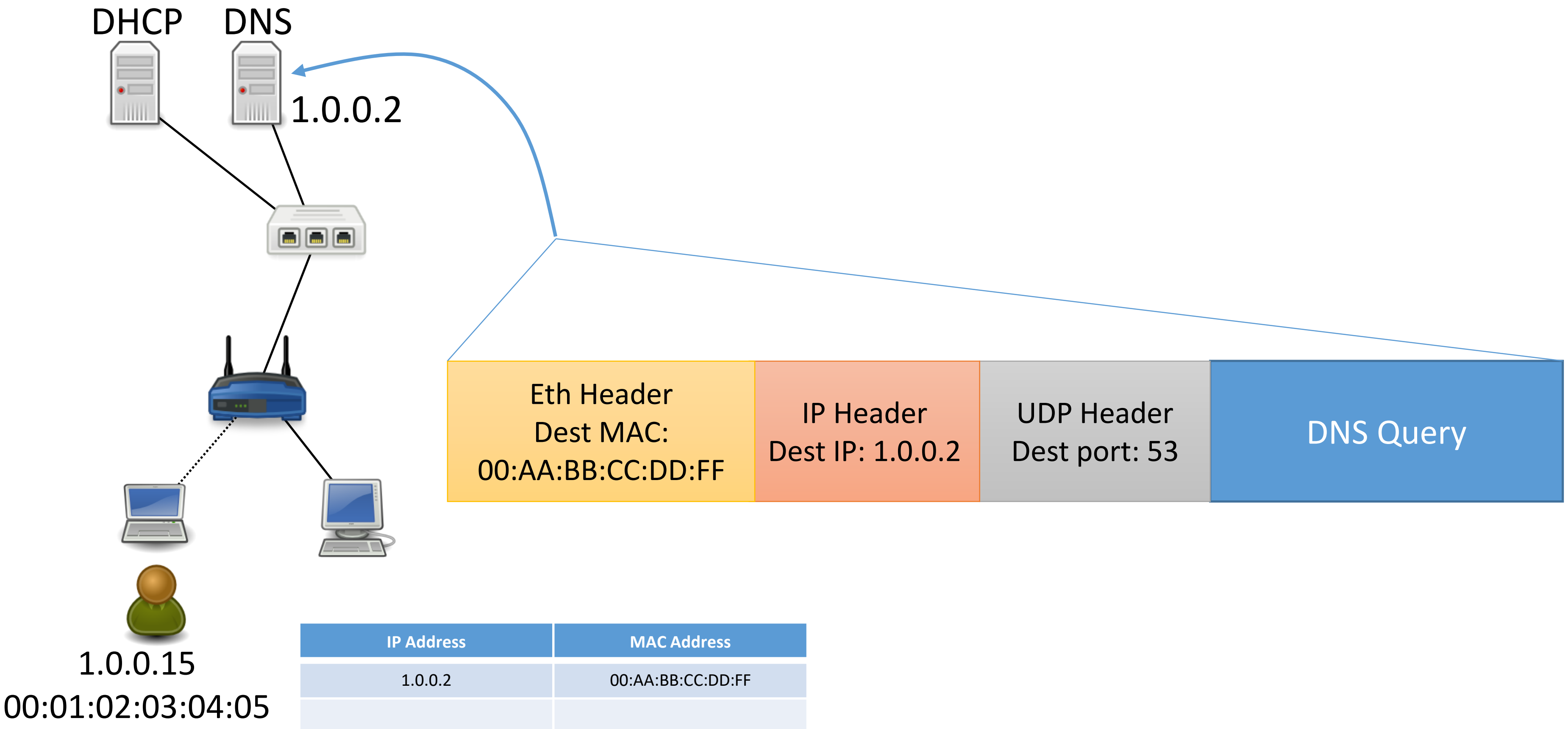
DNS server responds with MAC address.



IP Address	MAC Address
1.0.0.2	00:AA:BB:CC:DD:FF

Step 2: User Resolves Name

User queries local DNS resolver for www.xkcd.com.
Resolver runs necessary queries (root, TLD, etc.)



DHCP

DNS

1.0.0.2

Eth Header
Dest MAC:
00:AA:BB:CC:DD:FF

IP Header
Dest IP: 1.0.0.2

UDP Header
Dest port: 53

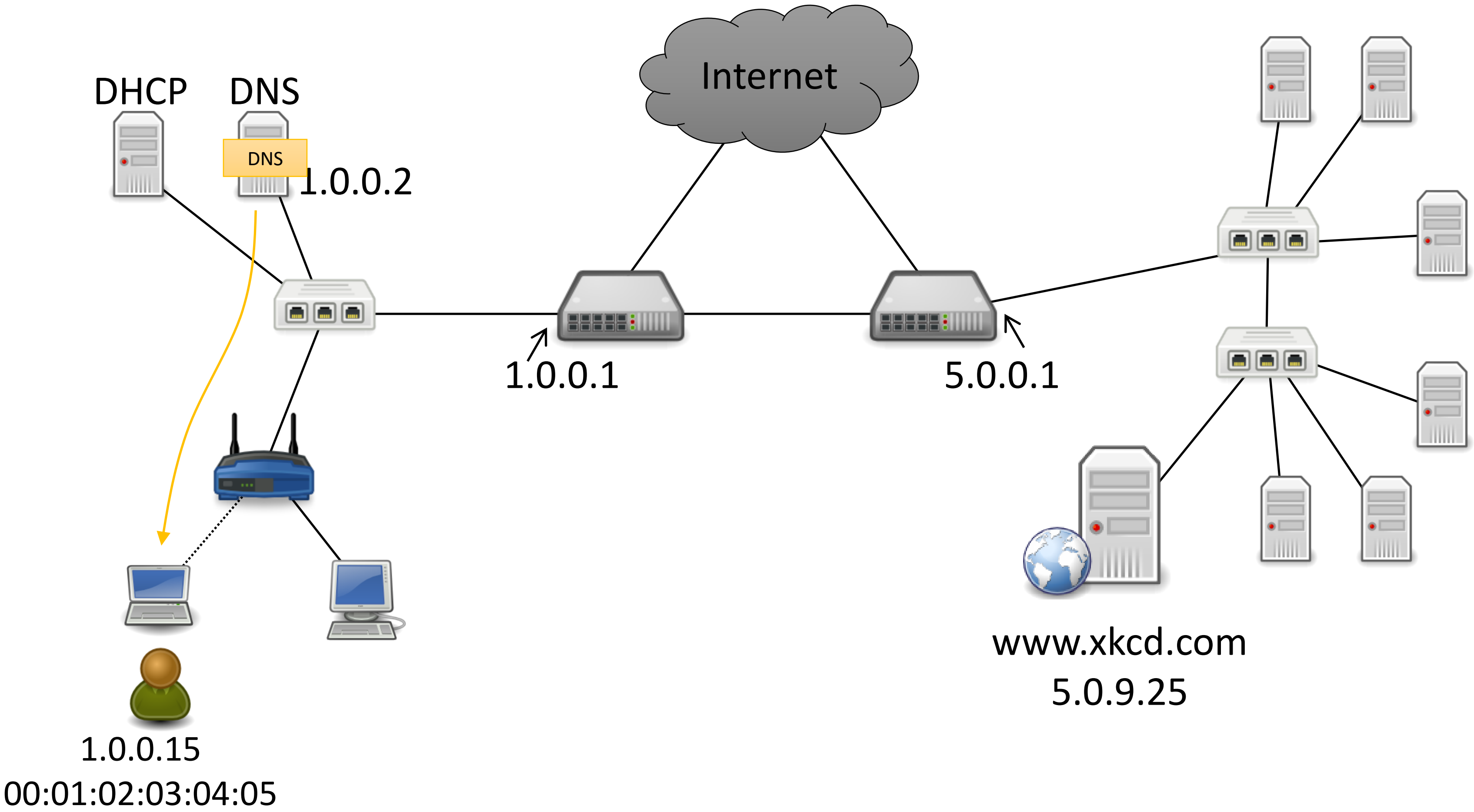
DNS Query

1.0.0.15
00:01:02:03:04:05

IP Address	MAC Address
1.0.0.2	00:AA:BB:CC:DD:FF

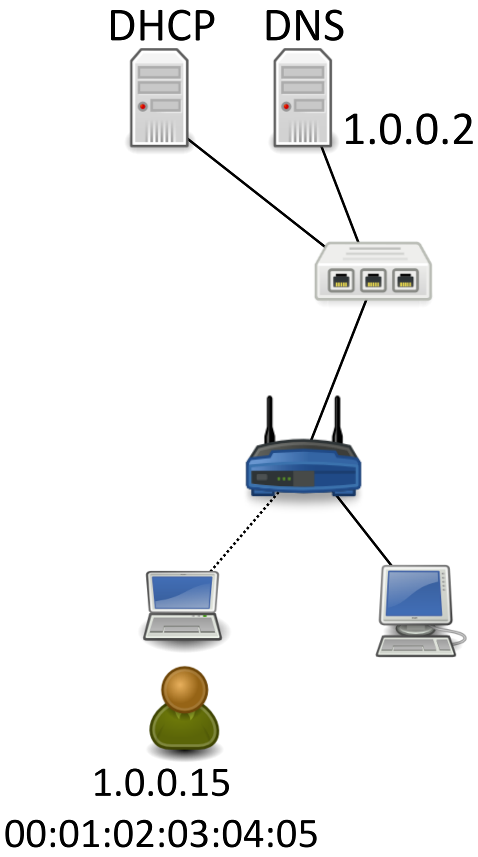
Step 2: User Resolves Name

DNS reply says that www.xkcd.com is 5.0.9.25.



Step 3: Establish a TCP Connection

User’s PC must answer: is the destination (5.0.9.25)
on my subnet? (Local vs. Internet)



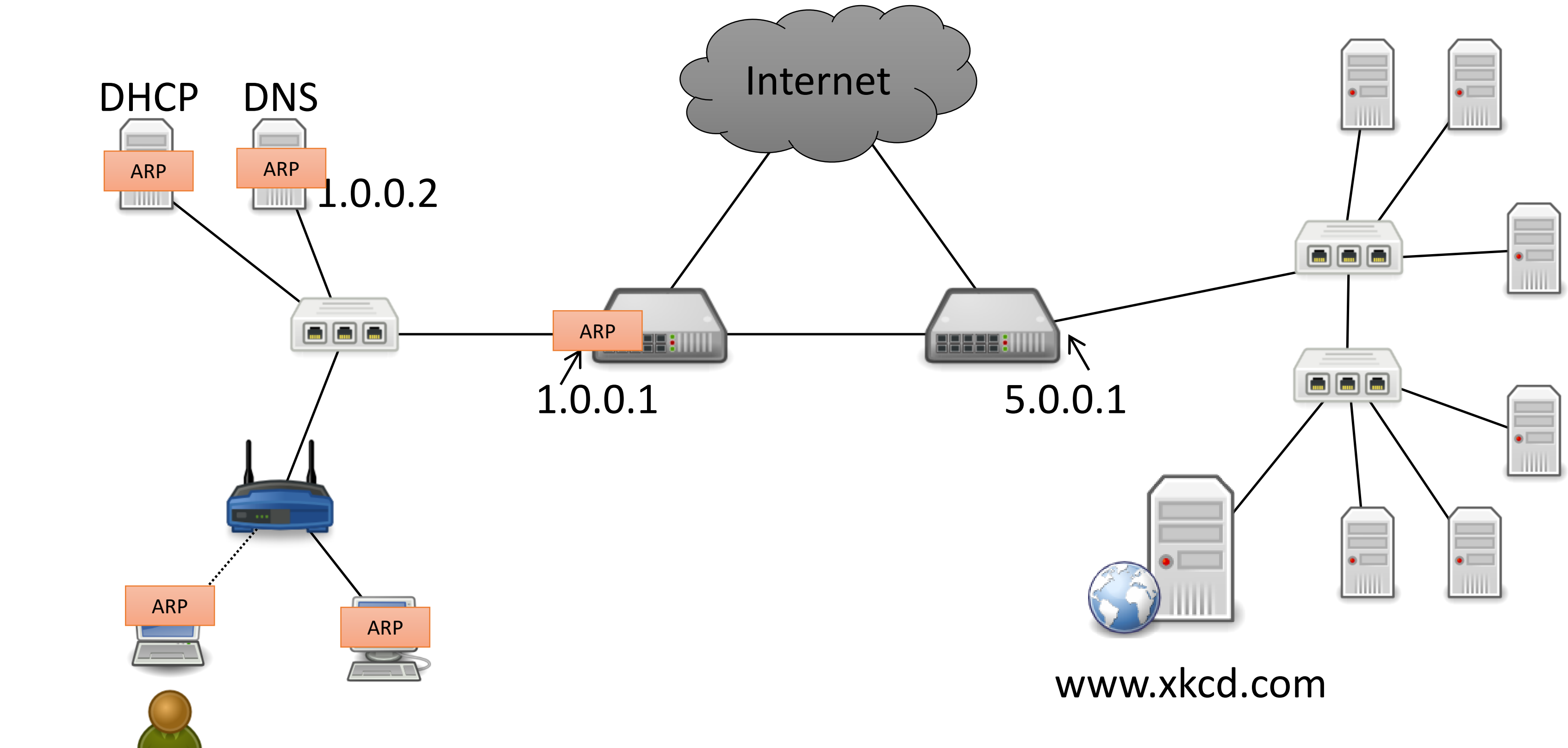
my address   subnet mask

1.0.0.15:	00000001	00000000	00000000	00001111
255.255.255.0:	11111111	11111111	11111111	00000000
ANDed together: my network prefix	00000001	00000000	00000000	
target address 5.0.9.25	00000101	00000000	00001001	00011101

No Match! Send it to the default gateway (router that connects to the Internet) that DHCP gave us (1.0.0.1).

Step 3: Establish a TCP Connection

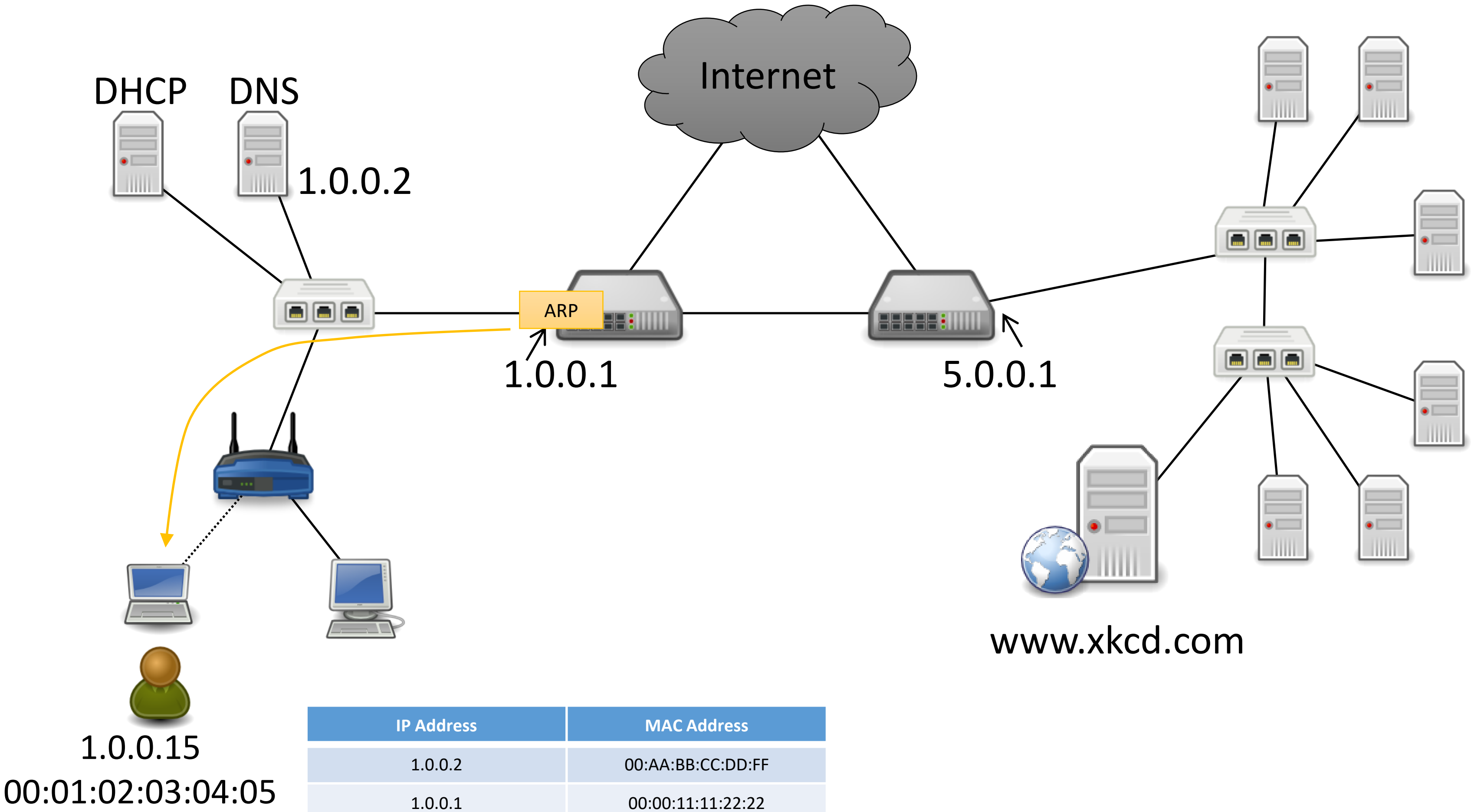
User’s PC does NOT know router’s MAC address!
Broadcast ARP request looking for 1.0.0.1!



IP Address	MAC Address
1.0.0.2	00:AA:BB:CC:DD:FF
1.0.0.1	?

Step 3: Establish a TCP Connection

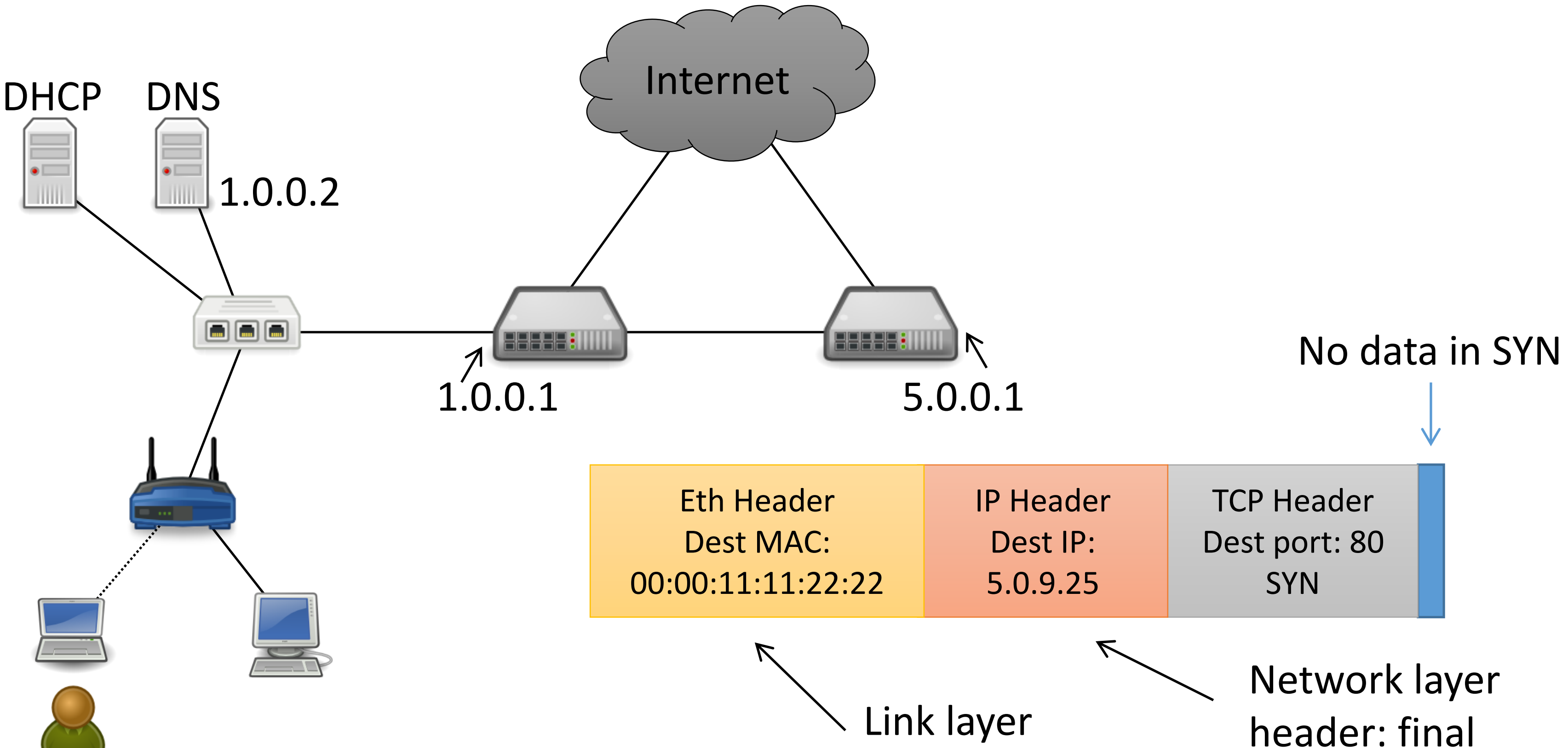
Router responds with MAC address.



IP Address	MAC Address
1.0.0.2	00:AA:BB:CC:DD:FF
1.0.0.1	00:00:11:11:22:22

Step 3: Establish a TCP Connection

Send TCP SYN to the destination, start 3-way handshake.



No data in SYN

Eth Header Dest MAC: 00:00:11:11:22:22	IP Header Dest IP: 5.0.9.25	TCP Header Dest port: 80 SYN

Link layer header: next hop destination!

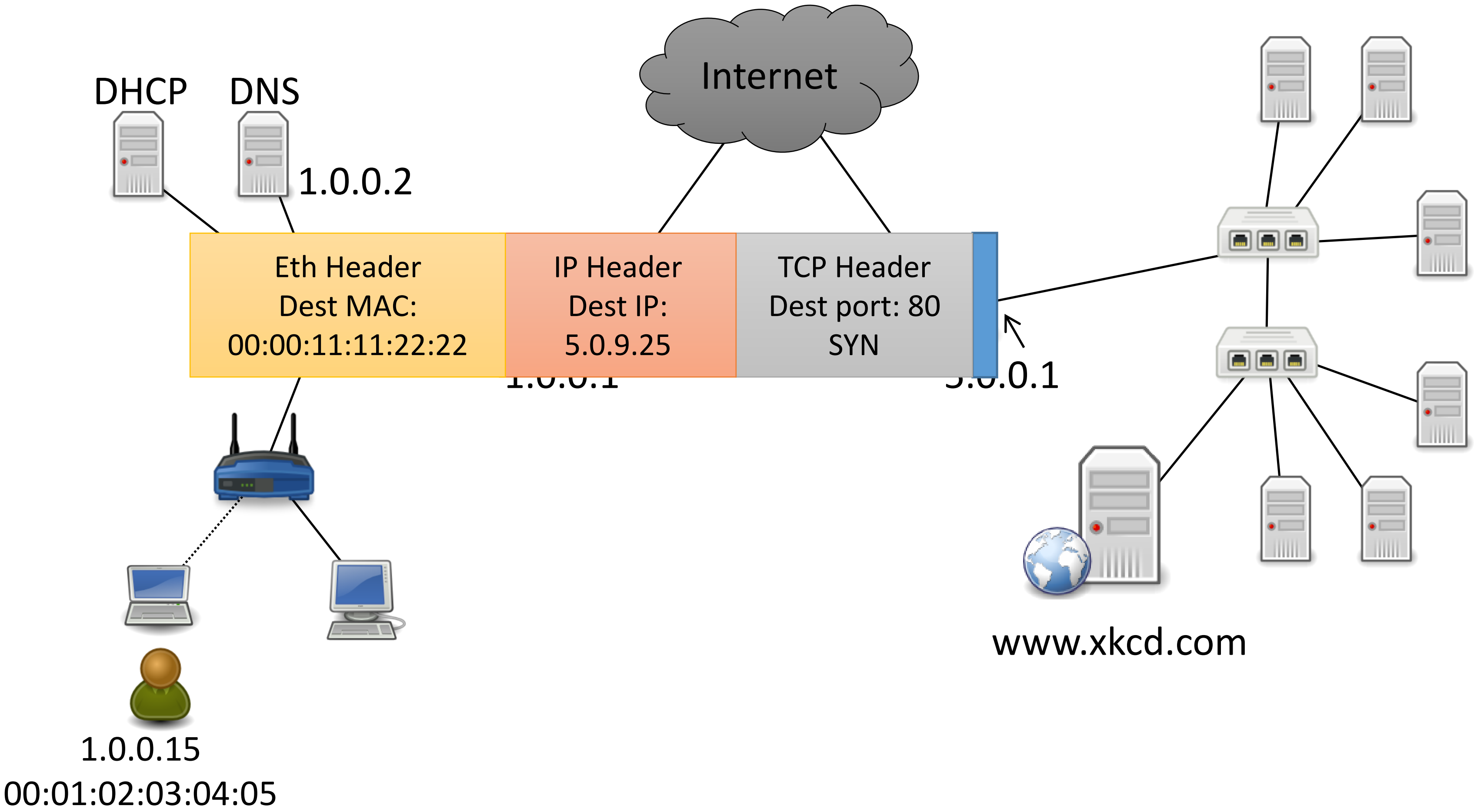
Network layer header: final destination!

1.0.0.15
00:01:02:03:04:05

IP Address	MAC Address
1.0.0.2	00:AA:BB:CC:DD:FF
1.0.0.1	00:00:11:11:22:22

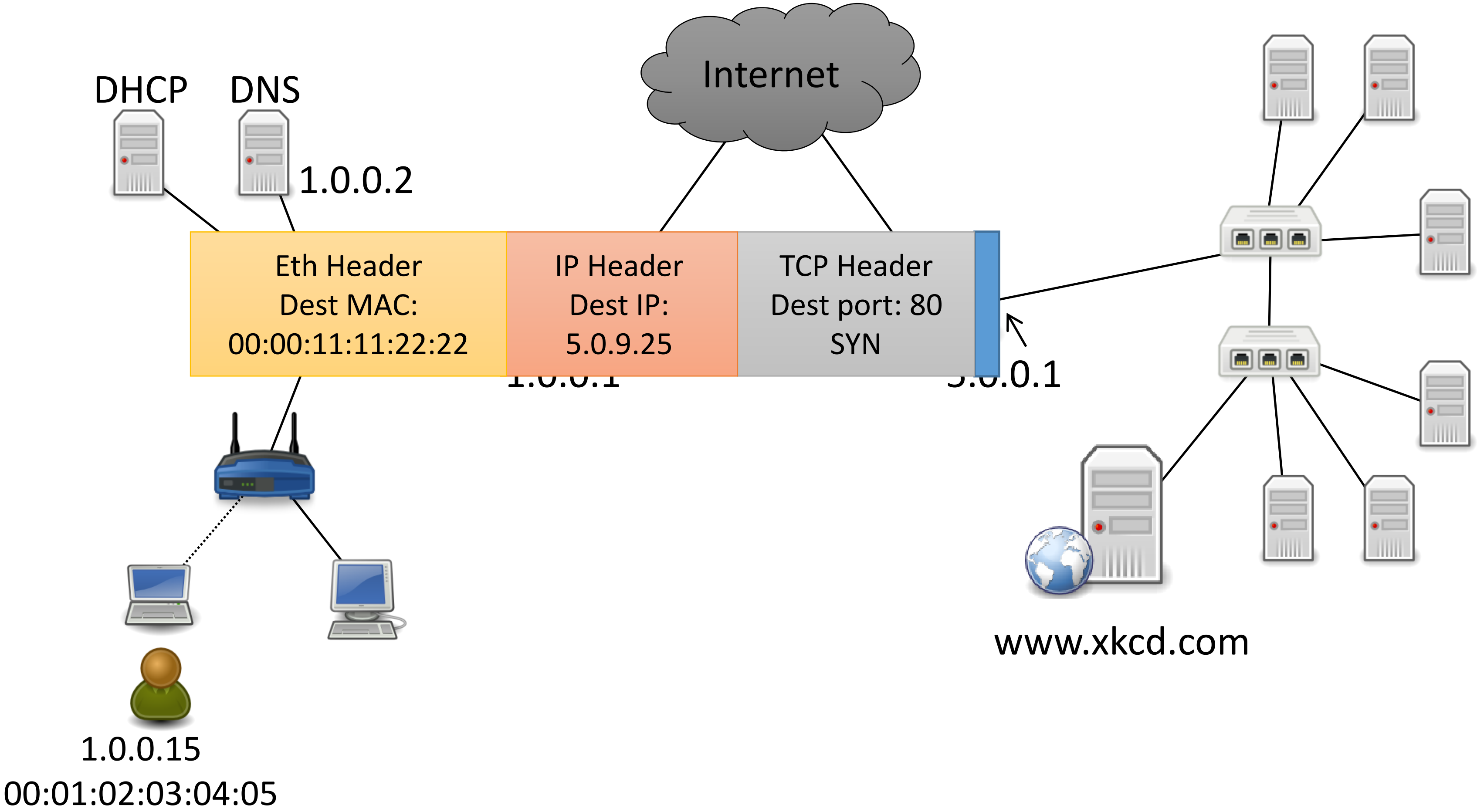
Step 3: Establish a TCP Connection

Send SYN to router.
NOTE: while the switch moves the frame to router, it is not ever addressed directly.



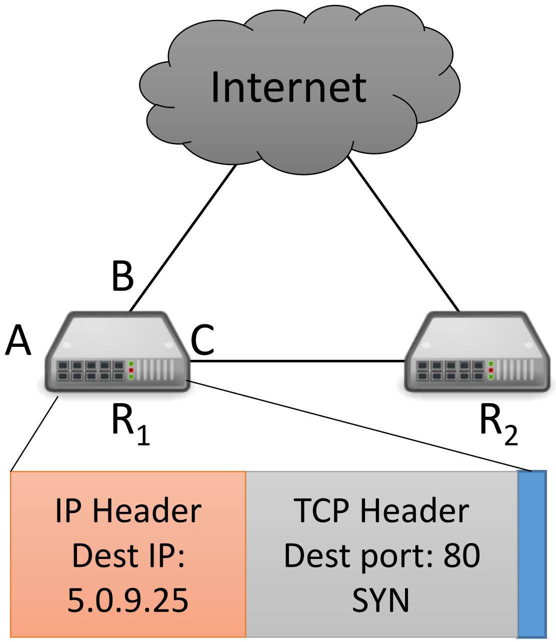
Step 3: Establish a TCP Connection

Router removes Ethernet header.



Step 3: Establish a TCP Connection

Router R_1 compares destination IP with its forwarding table, looks for longest prefix match.

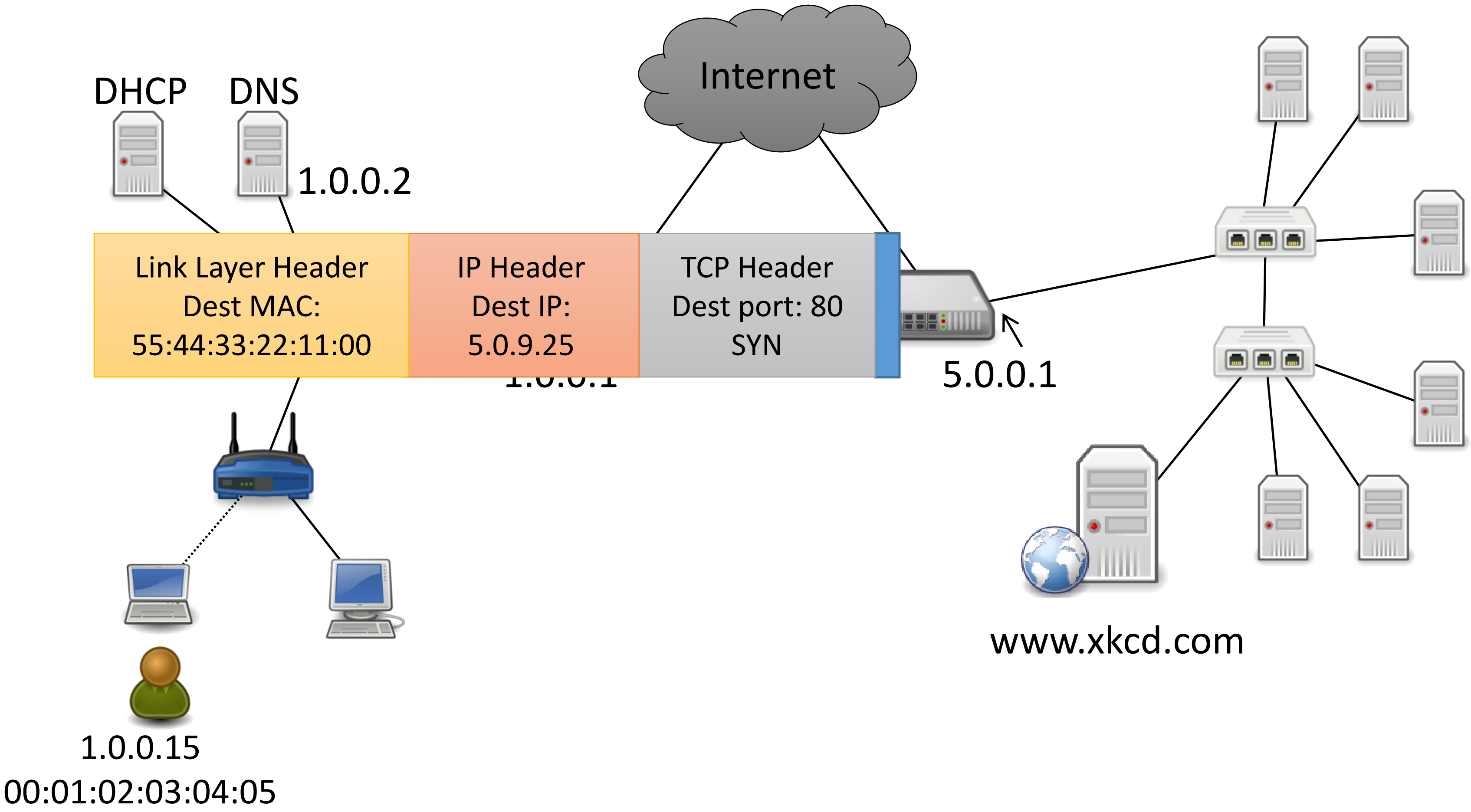


Prefix	Output Port	Next Router's Link Layer Addr
1.0.0.0/24	A	(N/A - no router there)
…	…	…
5.0.0.0/8	B	Some Internet router's address
5.0.0.0/16	C	R_2's Address: 55:44:33:22:11:00
…		

Best match: 5.0.0.0/16 -> Output port C
Destination MAC: 55:44:33:22:11:00

Step 3: Establish a TCP Connection

Router R_1 constructs frame and forwards it to R_2.



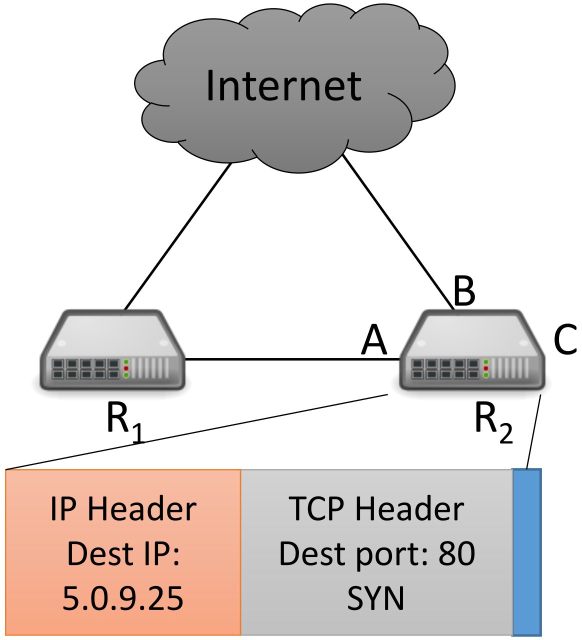
Link Layer Header
Dest MAC:
55:44:33:22:11:00

IP Header
Dest IP:
5.0.9.25

TCP Header
Dest port: 80
SYN

Step 3: Establish a TCP Connection

Router R_2 compares destination IP with its forwarding table, looks for longest prefix match.

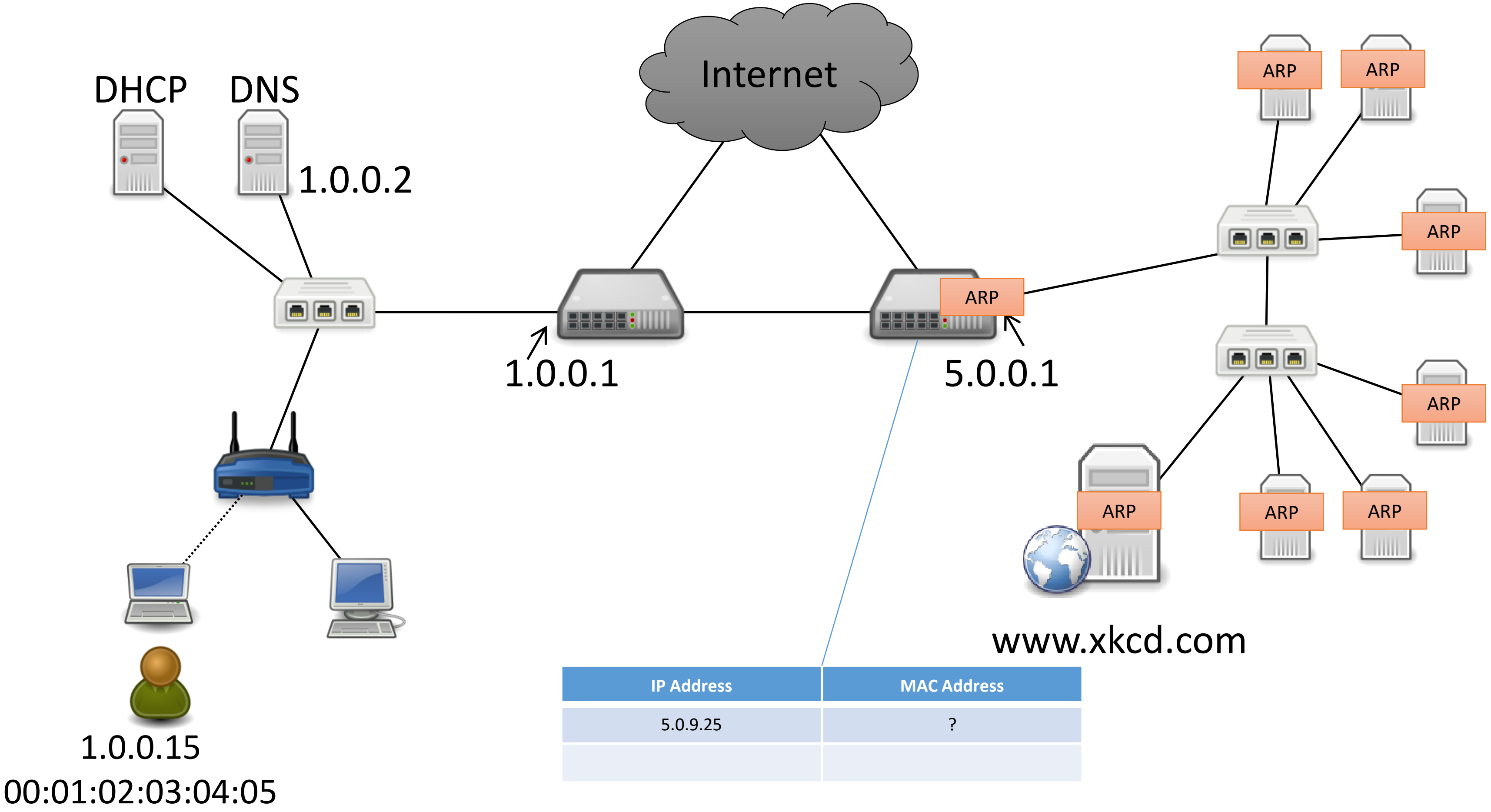


Prefix	Output Port	Next Router's Link Layer Addr
1.0.0.0/24	A	R_1's Address
…	…	…
5.0.0.0/8	B	Some Internet router's address
5.0.0.0/16	C	(N/A - no router there)
…		

Best match: 5.0.0.0/16 -> Output port C
Destination MAC: ?

Step 3: Establish a TCP Connection

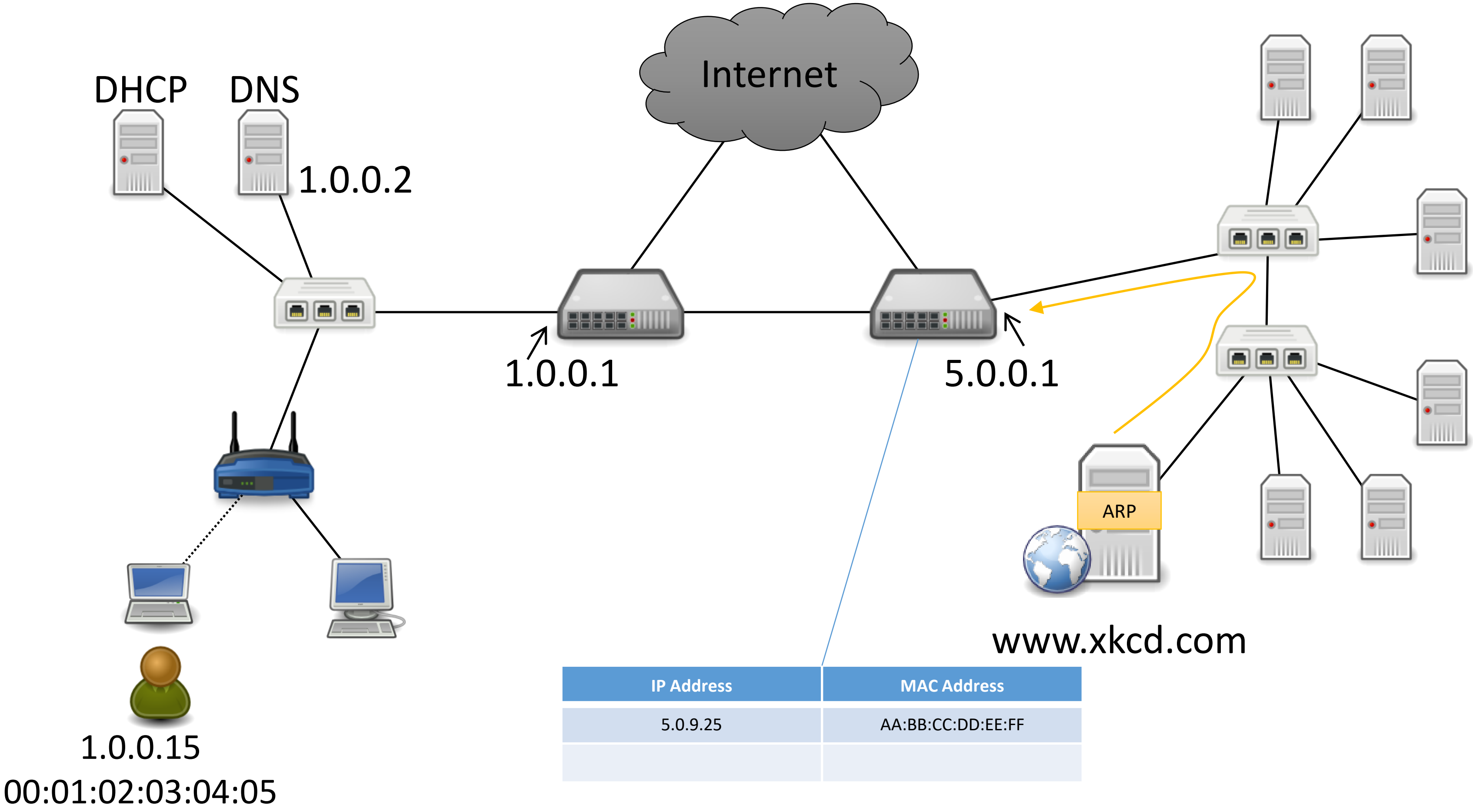
R_2 does NOT know destination's MAC address!
Broadcast ARP request looking for 5.0.9.25!
Data packet is queued while waiting for ARP to resolve.



IP Address	MAC Address
5.0.9.25	?

Step 3: Establish a TCP Connection

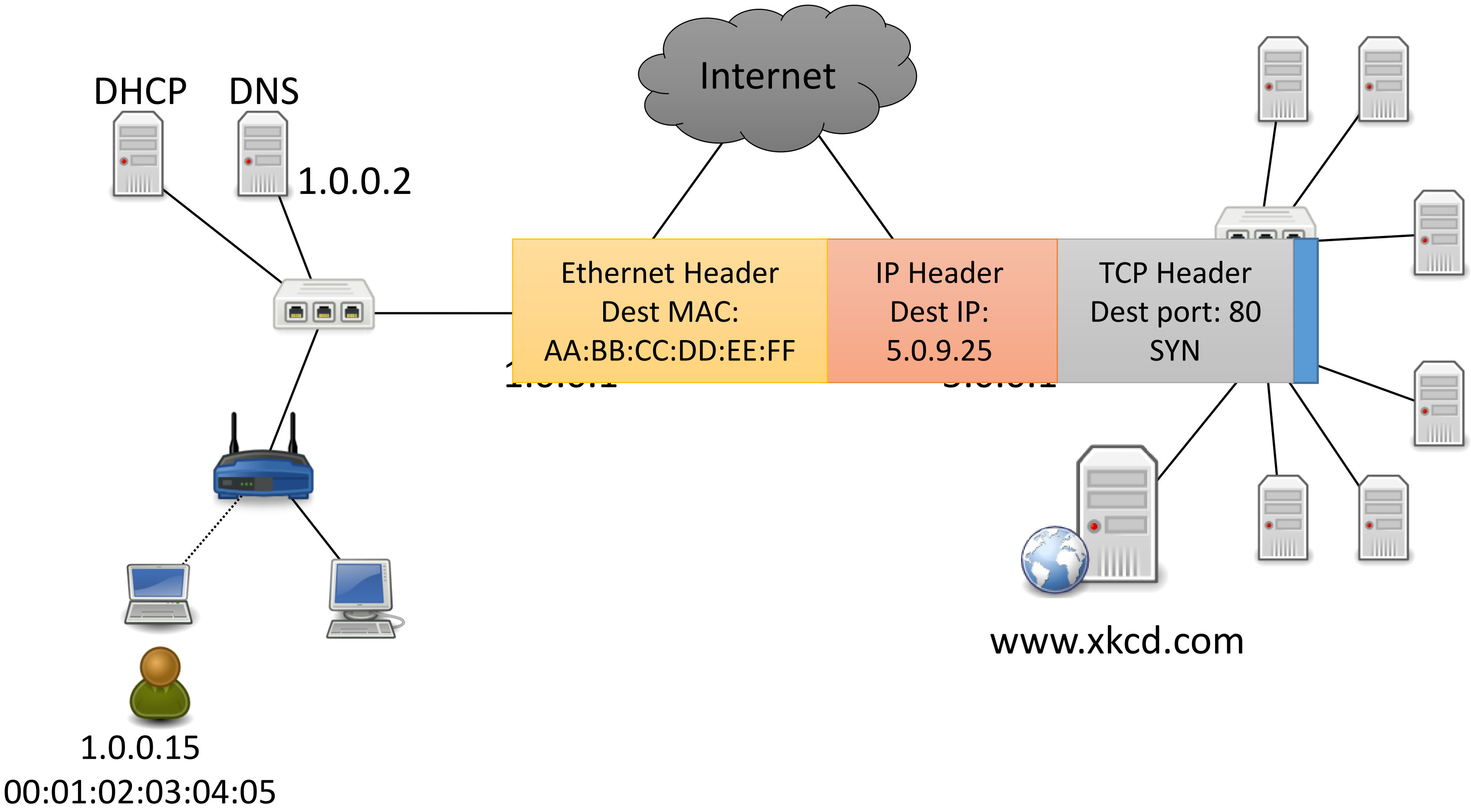
Host replies with MAC address.



IP Address	MAC Address
5.0.9.25	AA:BB:CC:DD:EE:FF

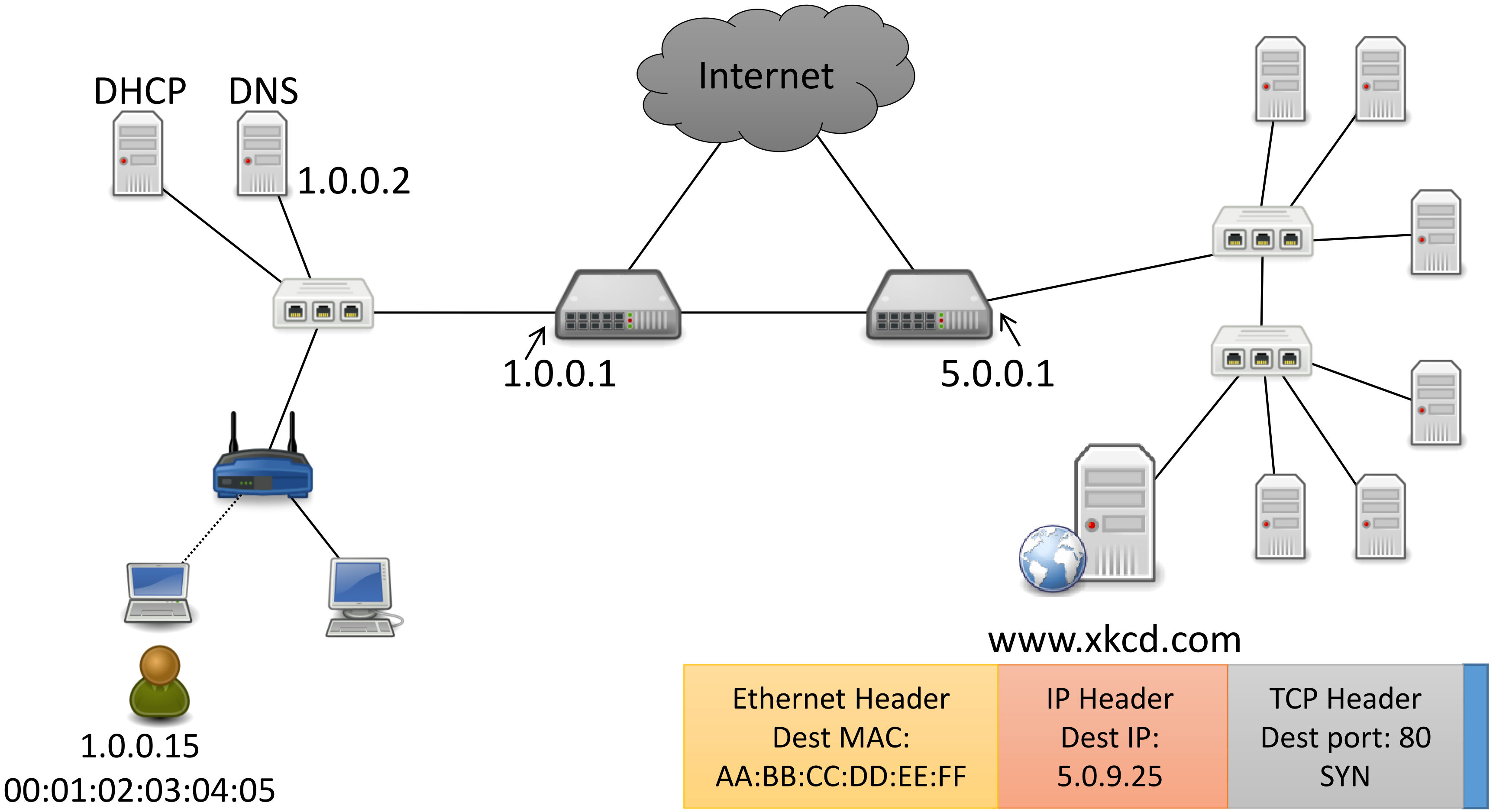
Step 3: Establish a TCP Connection

R_2 constructs frame, forwards it to destination.



Mission Accomplished!

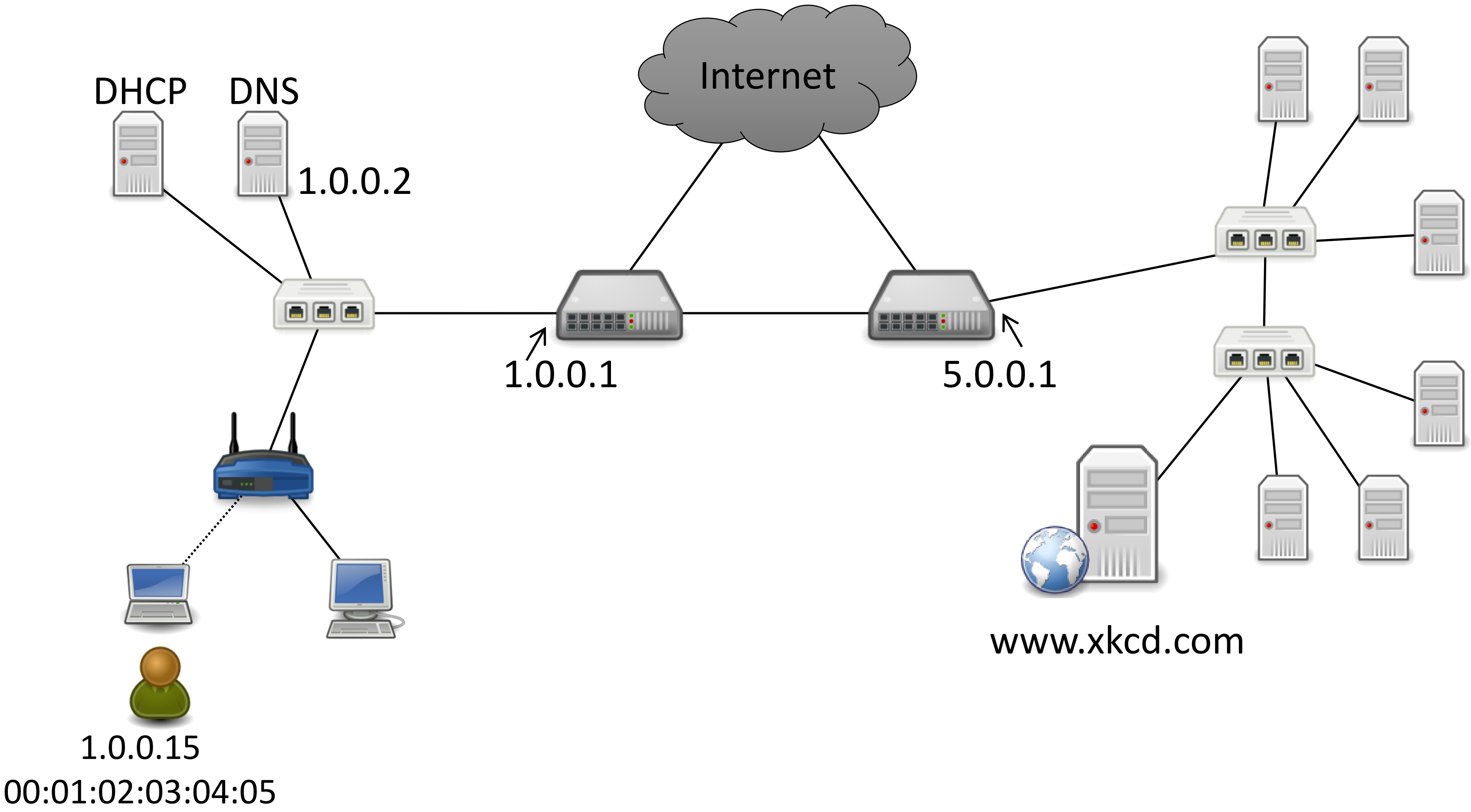
Destination peels off headers, generates reply (SYN+ACK).



Ethernet Header	IP Header	TCP Header	
Dest MAC:	Dest IP:	Dest port: 80	
AA:BB:CC:DD:EE:FF	5.0.9.25	SYN	

Mission Accomplished!

Process repeats in the opposite direction, without the ARPs this time. (MAC addresses were recently used, thus cached.)



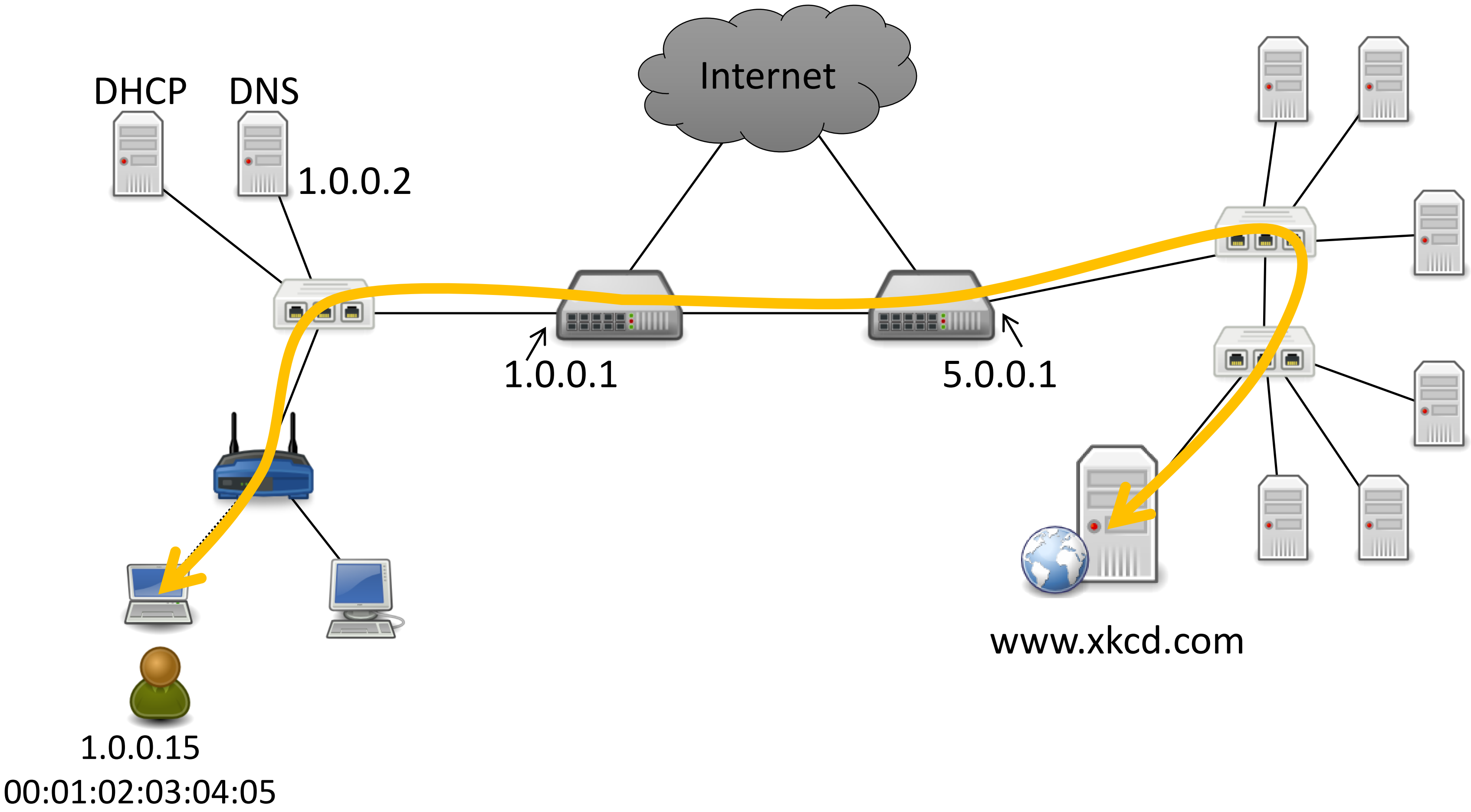
Steady State

- With DNS cached and ARP entries cached, host encapsulates data in TCP, IP, Eth headers and sends to router. Router forwards.

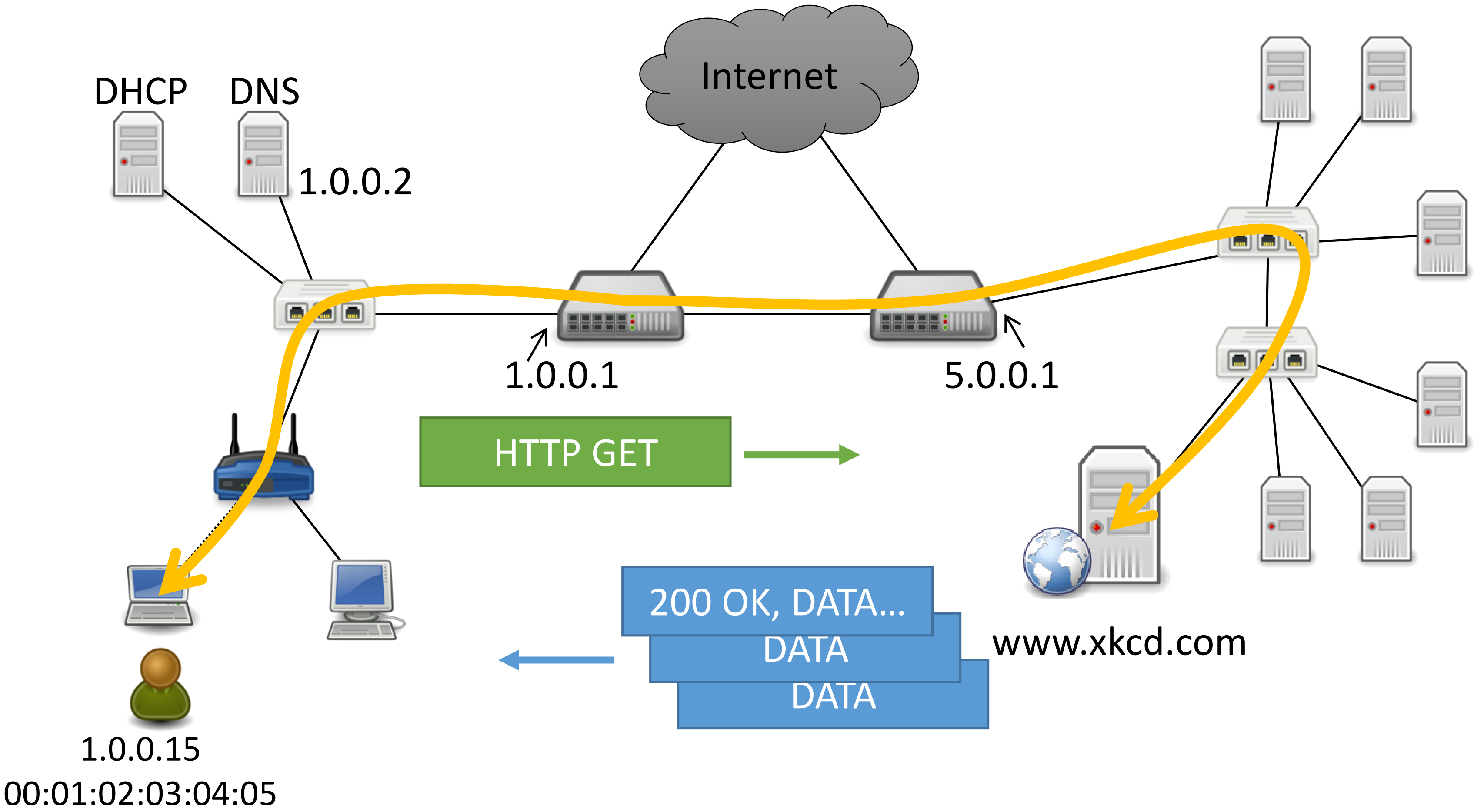
- Even *with* all the DNS/ARP, all that stuff happens in < 1 second
  (besides step 0: routing protocol)

Mission Accomplished!

Process repeats in the opposite direction, without the ARPs this time. (MAC addresses were recently used, thus cached.)



With TCP connection established, application can send HTTP request / reply.



Questions?