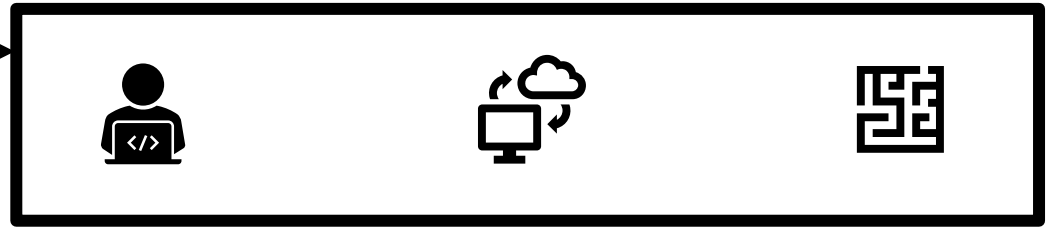
Design and Implementation of Content Routing Protocol

Mareesh Kumar Issar

March 9, 2020

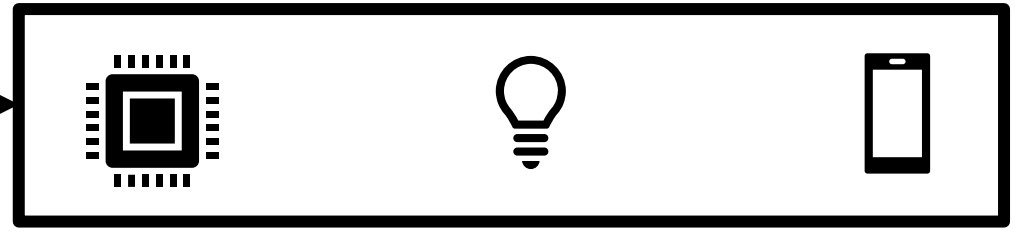
BACKGROUND

Graduate



Content Routing Protocol, Web application for stock price prediction, Minesweeper, Maze runner, Bayesian Hunting, and image colorizer

Undergraduate

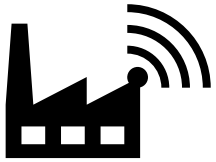


Electric Solar Vehicle Championship, Kalam Nanosatellite, Visible light communication and Robotics club.

MOTIVATION

- IoT
  - Smart connected IoT systems like smart homes, smart cities, and smart factories.
  - Challenges in accessing these devices.
  - Content more important than device.

- Issues with existing network protocols:
  - Latency
  - Network congestion



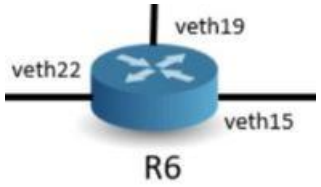
CHALLENGES

- How to mitigate the drawbacks of traditional protocols?
- Can we create a content centric network protocol?
- What if we can fetch content from nearby devices?
- Can we reduce the latency?

- Key idea: Design a network protocol by assigning unique id to each content.

KEY COMPONENTS

- Router



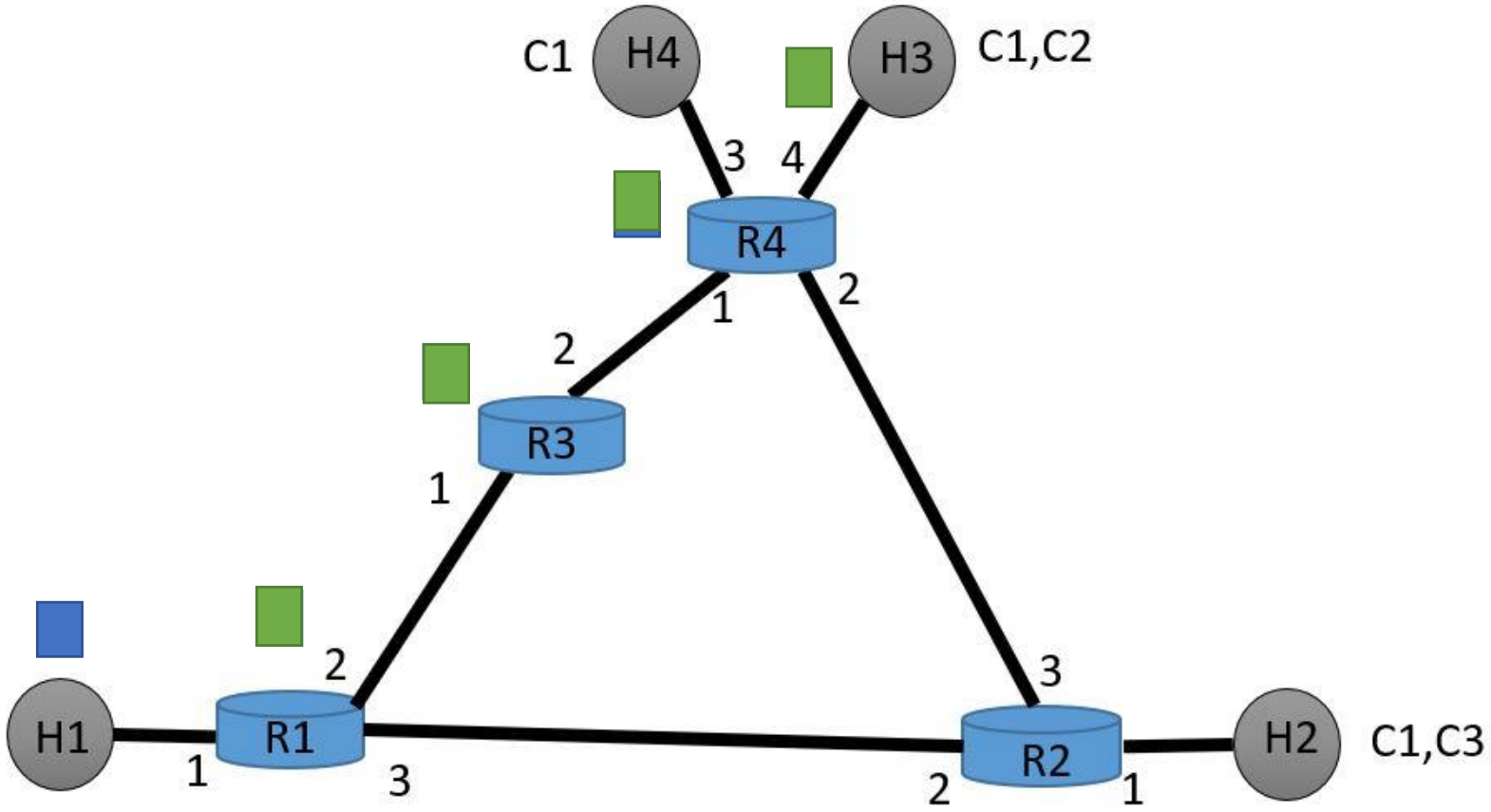
- Host device

- Packets

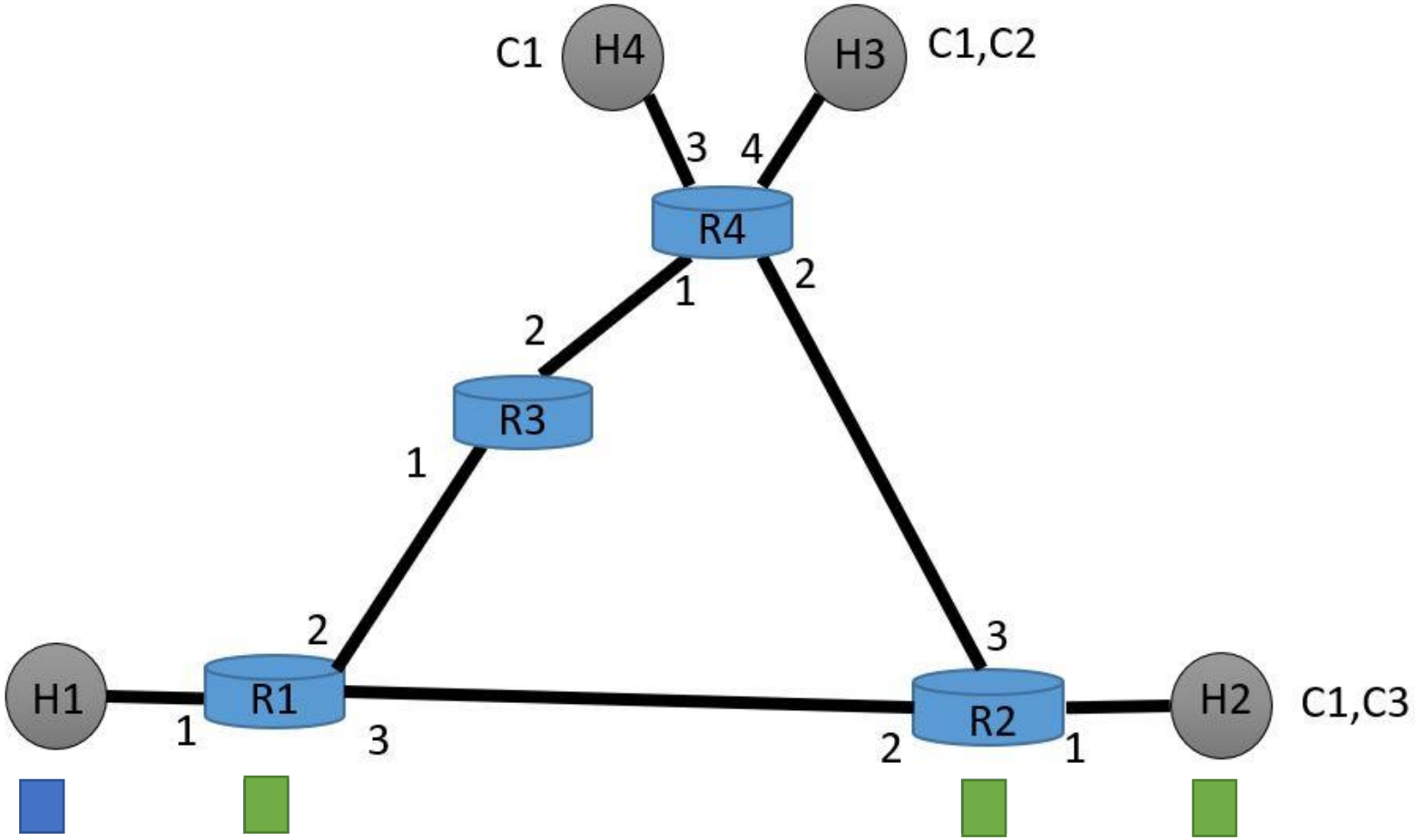
- Client

- Server

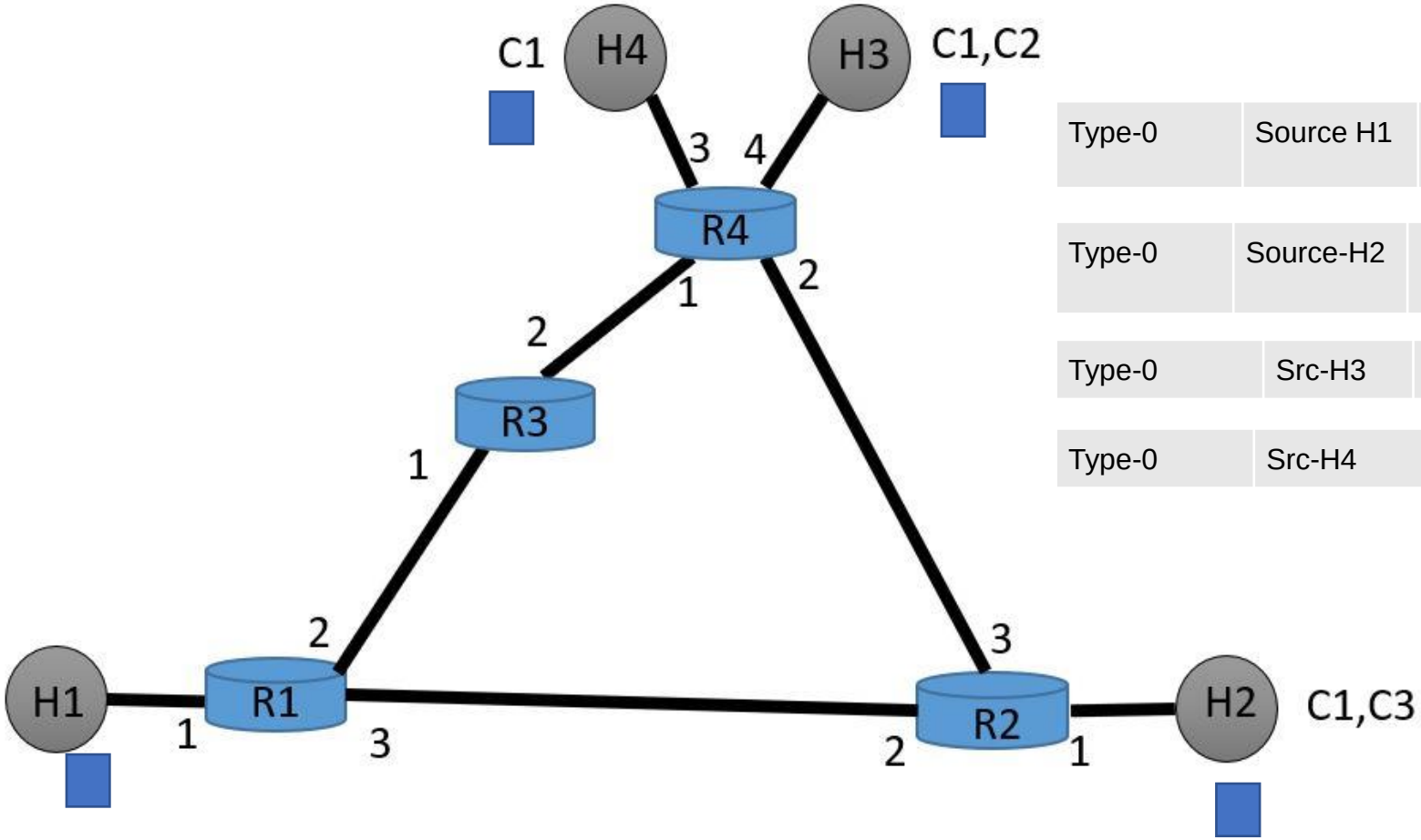
TRADITIONAL MECHANISM (UNICAST)



PROPOSED MECHANISM (ANYCAST)

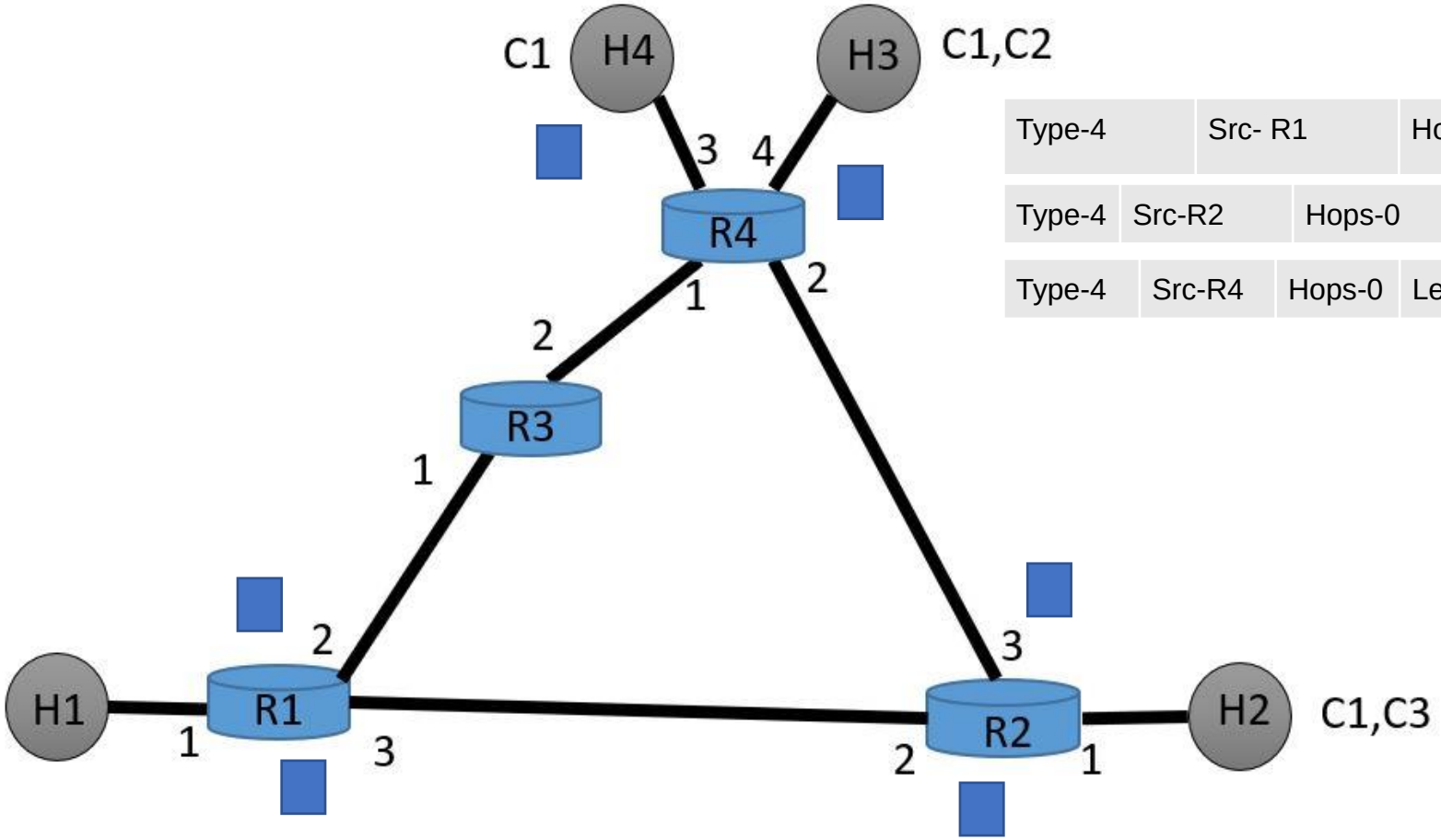


BOOTSTRAP AND DISCOVERY



Type-0	Source H1	Length 0	NULL	
Type-0	Source-H2	Length 2	C1	C3
Type-0	Src-H3	Length 2	C1	C2
Type-0	Src-H4	Length 1	C1	

ROUTING UPDATE



Type-4	Src- R1	Hops-0	Length-1	H1		
Type-4	Src-R2	Hops-0	Length-3	H2	C1	C3
Type-4	Src-R4	Hops-0	Length-4	H3	H4	C1	C2

ROUTING UPDATE

- **STEP 1**: In the bootstrap phase.
- **STEP 2:** After one time tick.
- **STEP 3:** After two time ticks.

- **At Router 1**
- **Host Routing Table:**

Host ID	Hop Count	Port No.
H1	0	1
H2	1	3
H3	2	3
H4	2	3

Content Routing Table:

Content ID	Hop Count	Port No.
C1	1	3
C3	1	3
C2	2	3

ROUTING UPDATE

- **STEP 1**: In the bootstrap phase.
- **STEP 2:** After one time tick.
- **STEP 3:** After two time ticks.

- **At Router 2**
- **Host Routing Table:**

Host ID	Hop Count	Port No.
H2	0	1
H1	1	2
H3	1	3
H4	1	3

Content Routing Table:

Content ID	Hop Count	Port No.
C1	0	1
C3	0	1
C2	1	3

ROUTING UPDATE

- **STEP 1**: In the bootstrap phase.
- **STEP 2:** After one time tick.
- **STEP 3:** After two time ticks.

- **At Router 3**
- **Host Routing Table:**

Host ID	Hop Count	Port No.
H1	1	1
H3	1	2
H4	1	2
H2	2	1

Content Routing Table:

Content ID	Hop Count	Port No.
C1	1	2
C2	1	2
C3	2	1

ROUTING UPDATE

- **STEP 1**: In the bootstrap phase.
- **STEP 2:** After one time tick.
- **STEP 3:** After two time ticks.

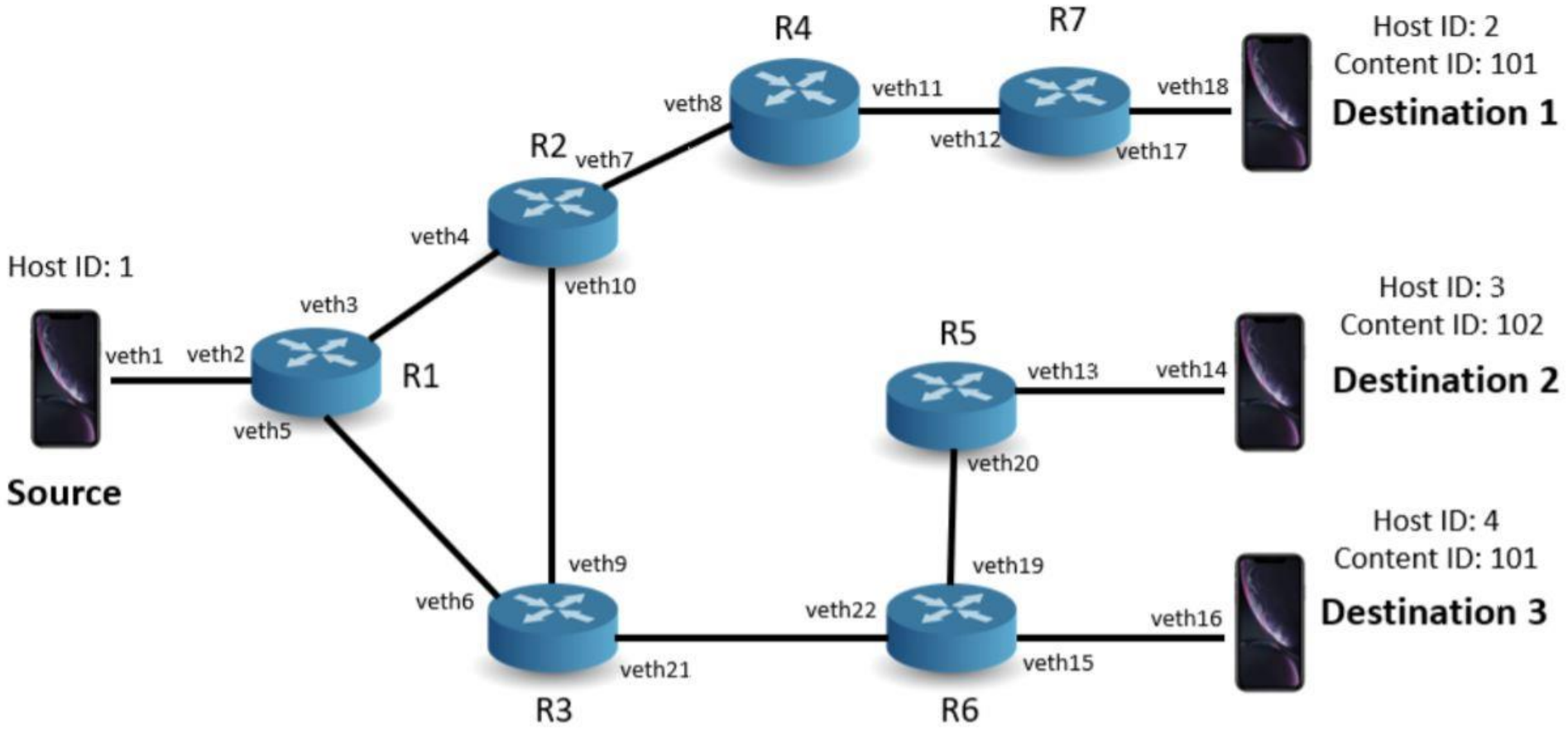
- **At Router 4**
- **Host Routing Table:**

Host ID	Hop Count	Port No.
H3	0	4
H4	0	3
H2	1	2
H1	2	1

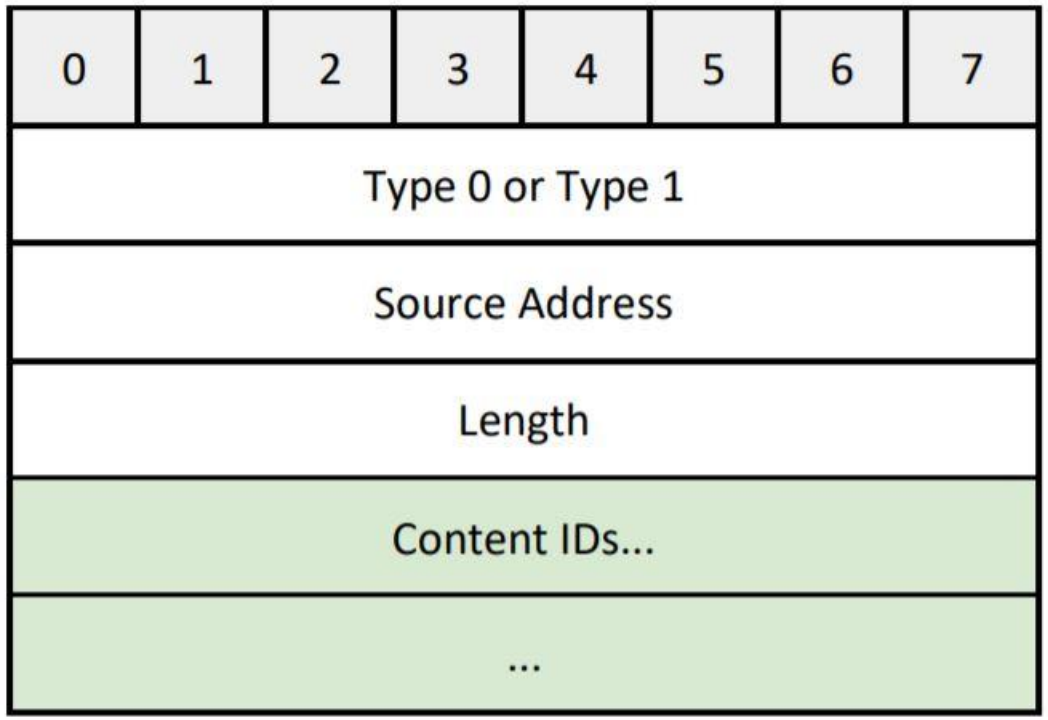
Content Routing Table:

Content ID	Hop Count	Port No.
C1	0	4
C2	0	4
C3	1	2

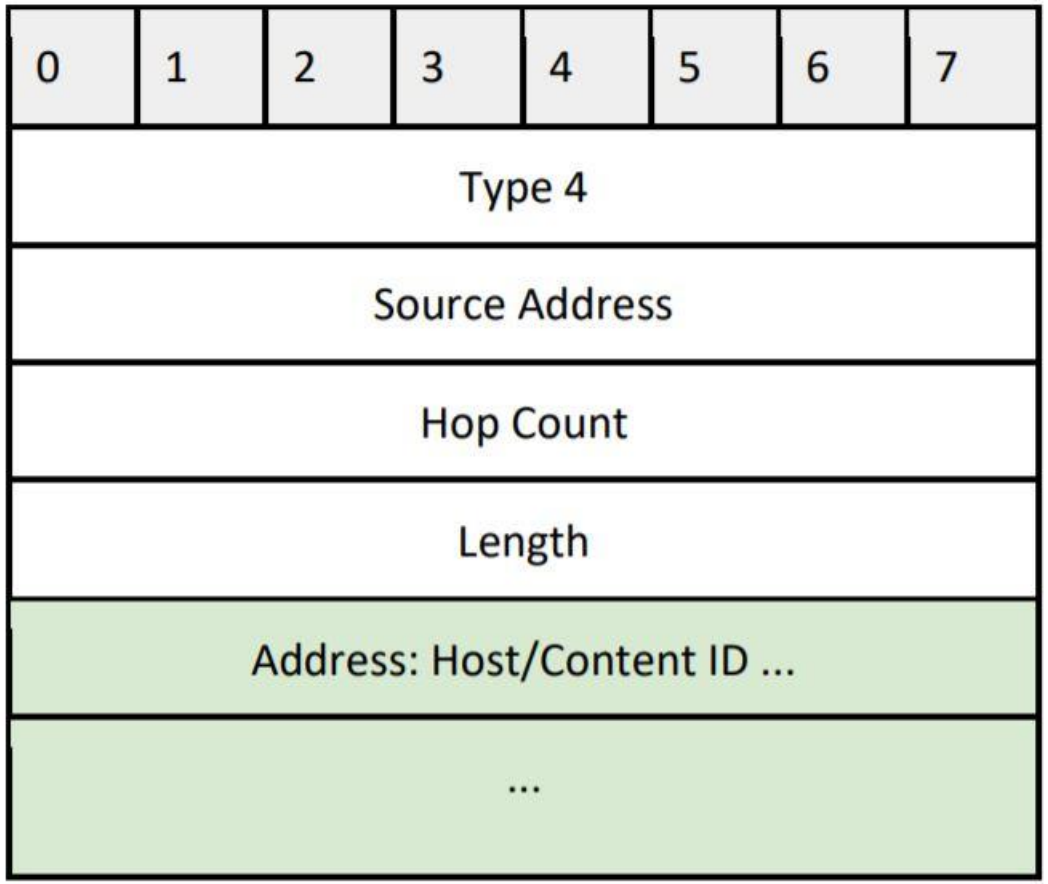
NETWORK TOPOLOGY



PACKET STRUCTURE

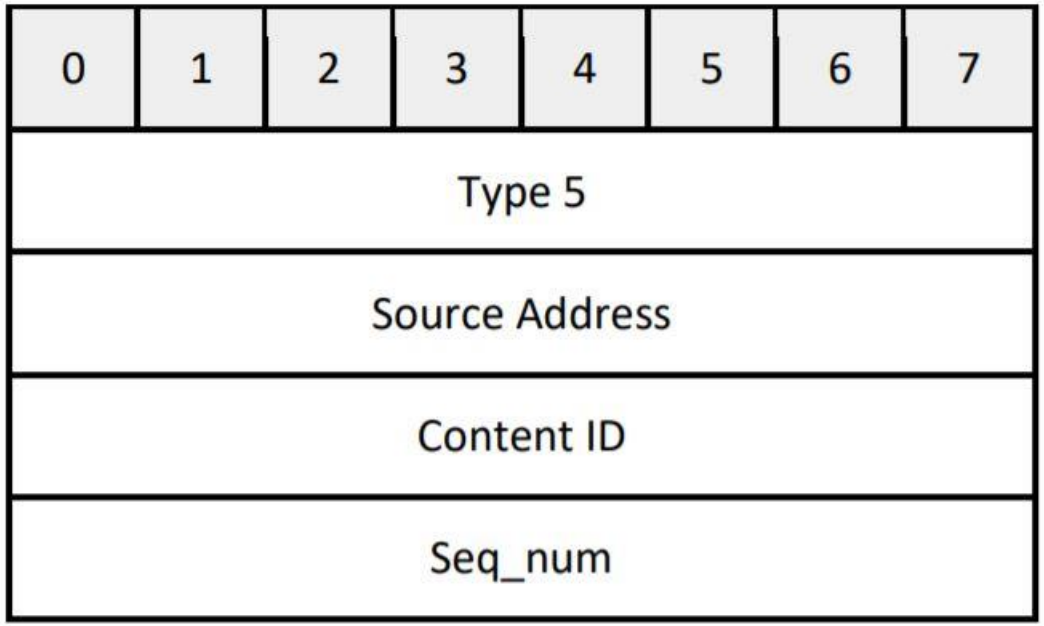


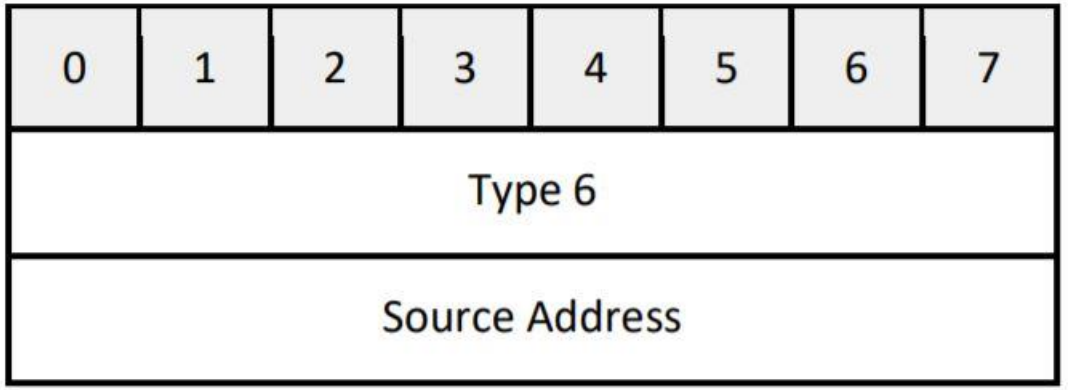
Update packet and Delete packet



Routing Update packet

PACKET STRUCTURE

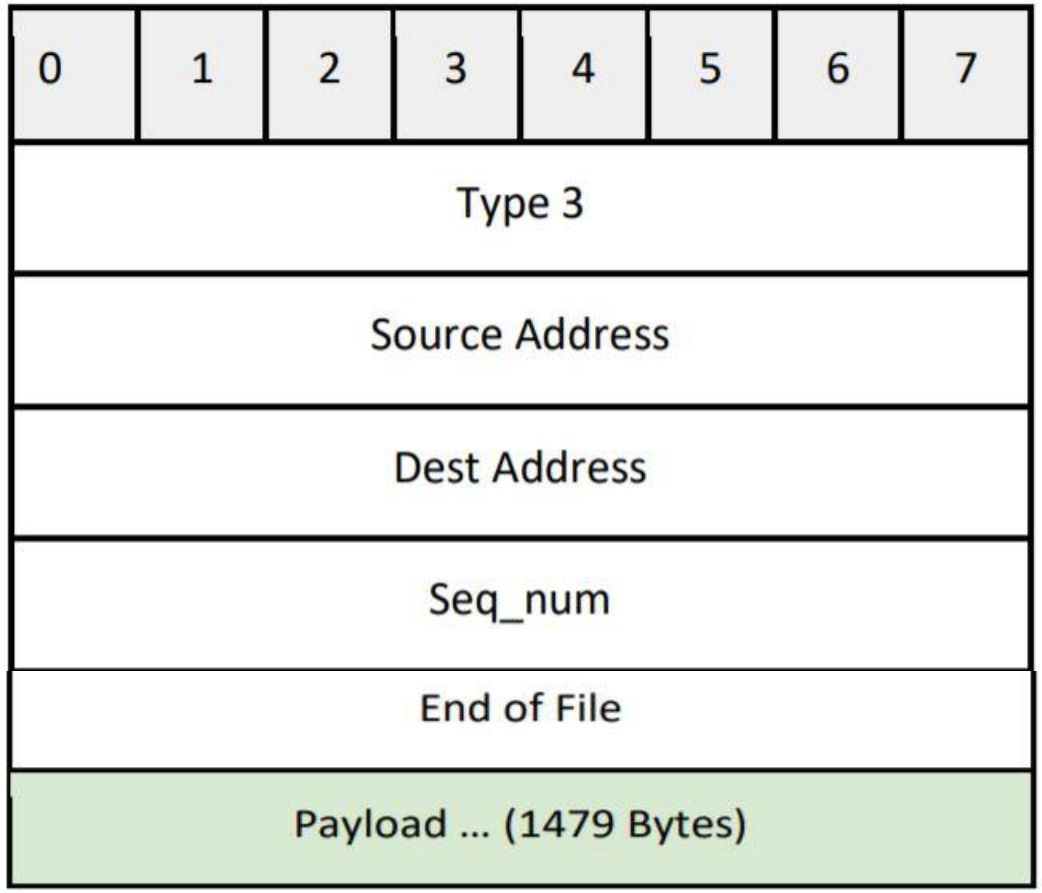




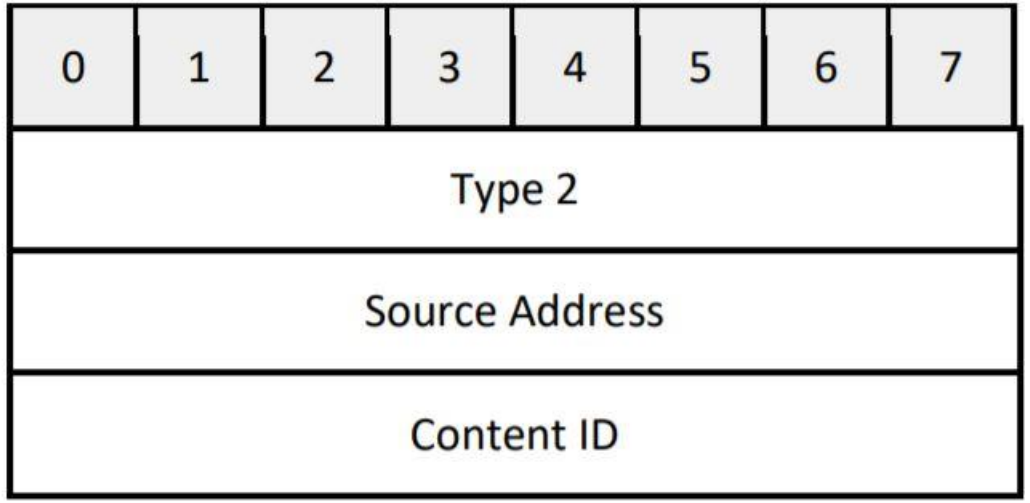
ACK packet

Hello packet

PACKET STRUCTURE

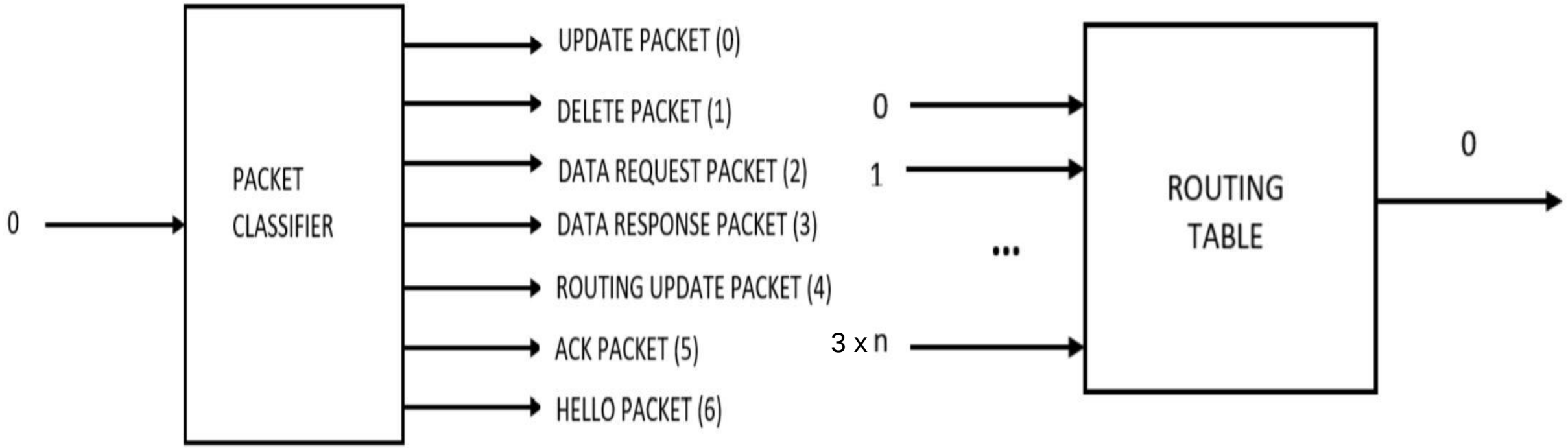


Data Response packet

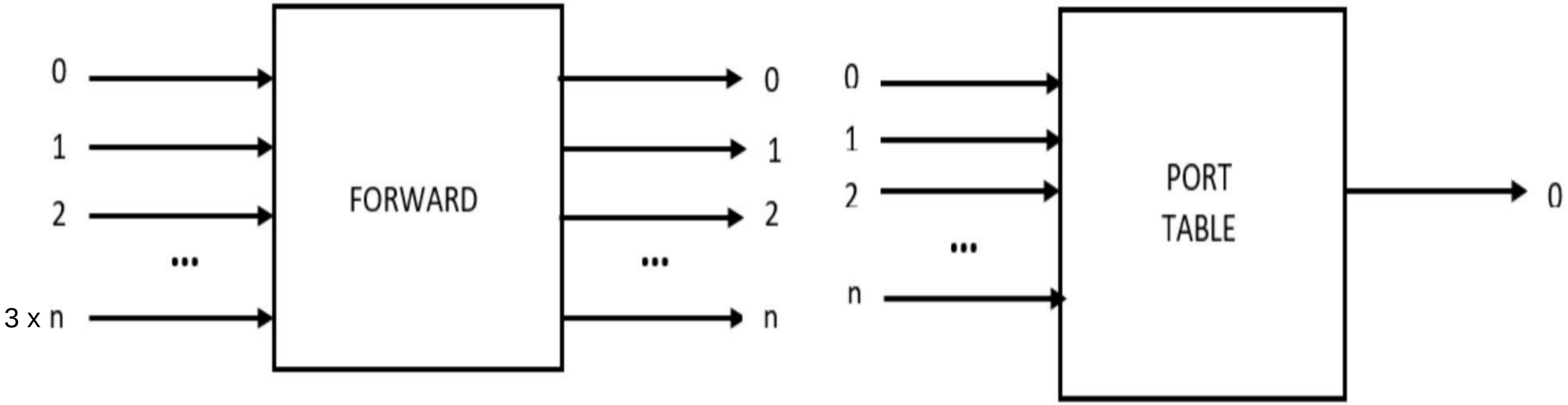


Data request packet

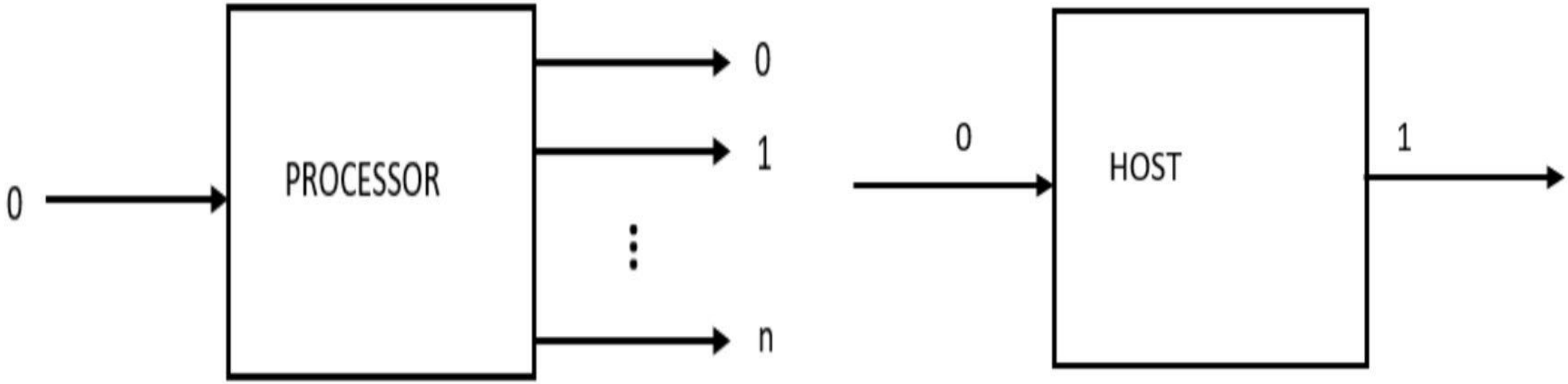
PACKET HANDLING ELEMENTS



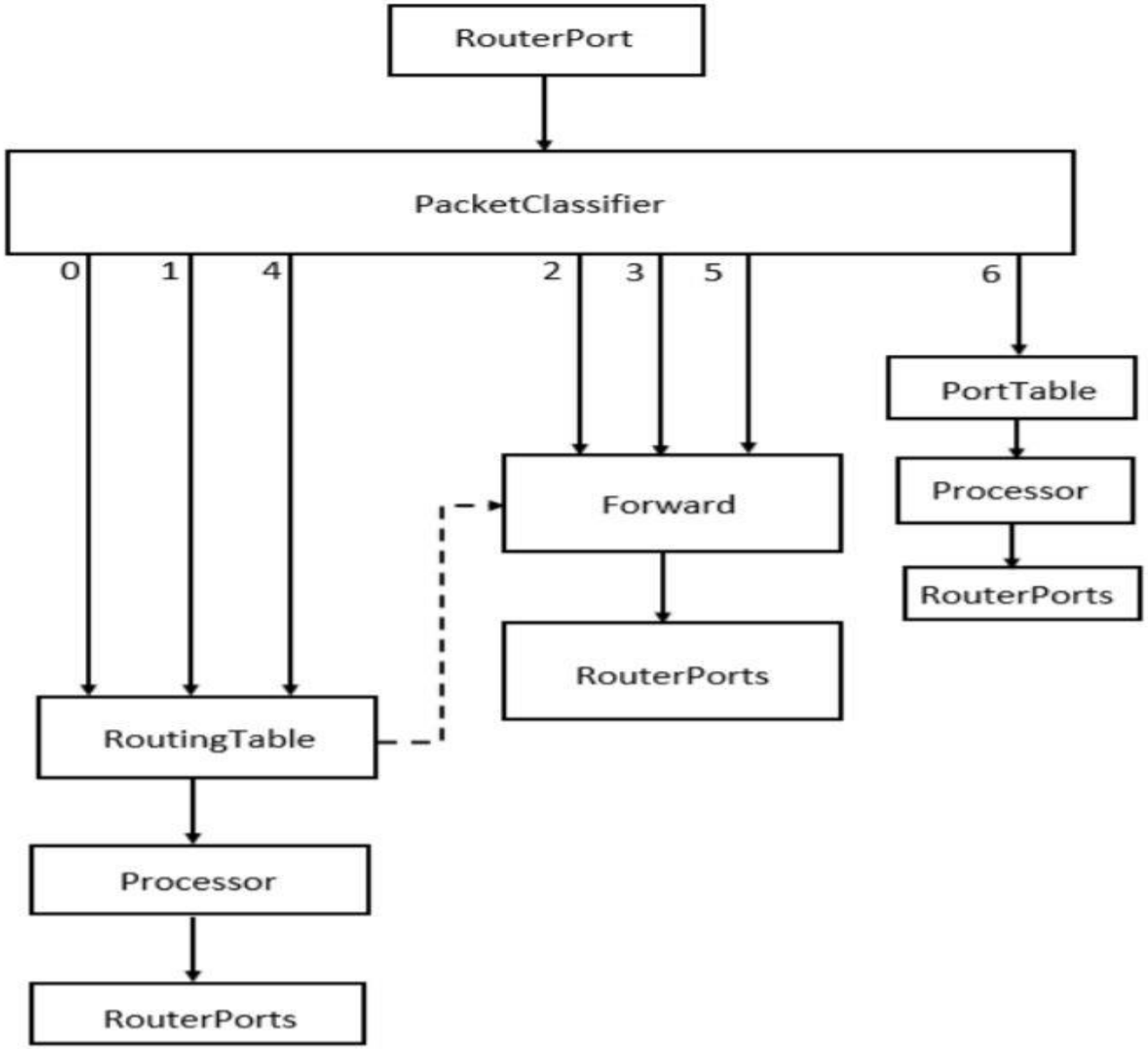
PACKET HANDLING ELEMENTS



PACKET HANDLING ELEMENTS

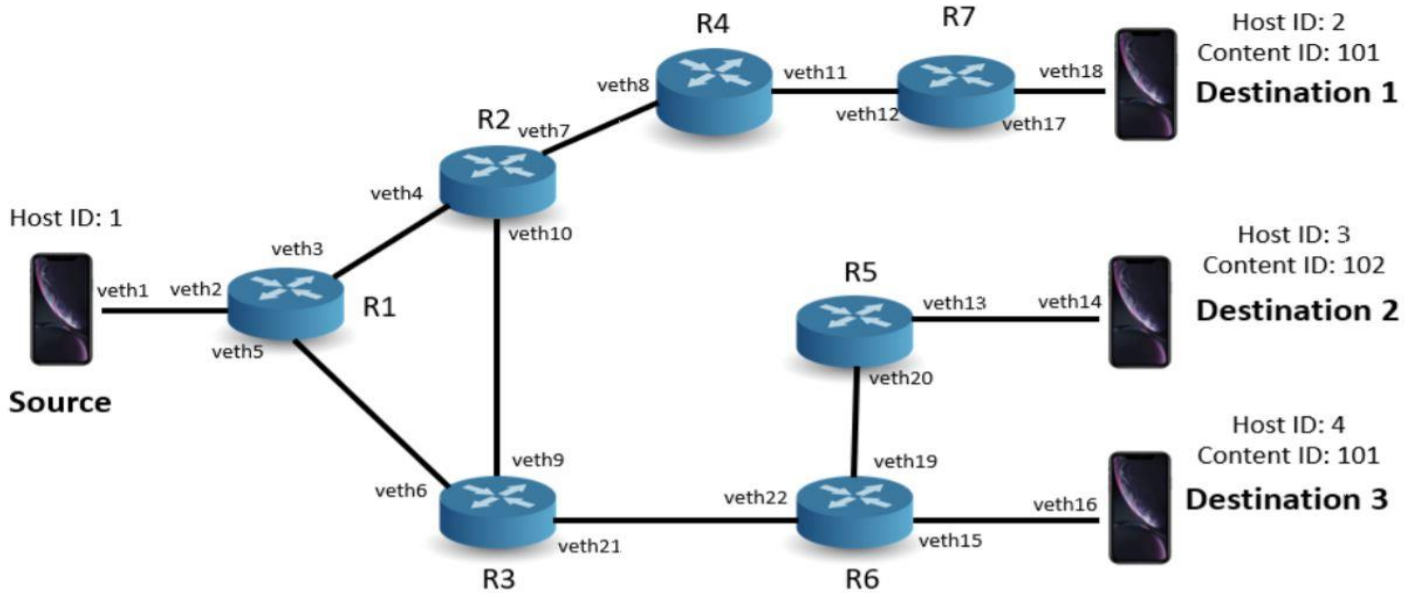


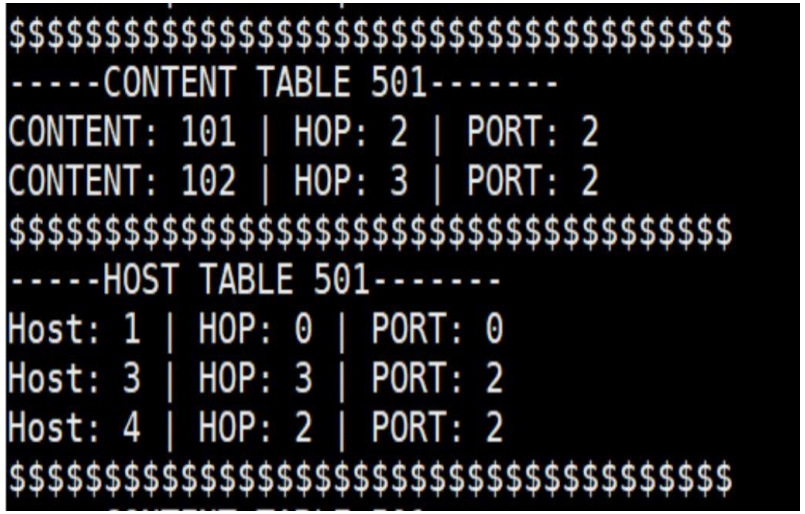
SYSTEM FLOWCHART



RESULTS

Source (Host ID)	Destination (Host ID)	End-to-end Latency (ms)
1	2	61
1	3	64
1	4	37





CONCLUSION

- Traditional network protocols are host centric rather than content centric.
- Designed and implemented the syntax, semantics and the algorithm for content based routing protocol.
- Used controlled flooding based routing with loop prevention mechanism to disseminate control information.
- Implemented the packets and router elements in click.
- Reduced end-to-end latency by 39%.

FUTURE WORK

- Analyze results for larger and complex topologies.
- Implement and analyze the results for mobile host devices.
- Expand the current implementation to support multicast.

THANK YOU 

QUESTIONS ?

BACKUP SLIDES

- For further information, please visit: https://github.com/mareeshissar/ECE544_Communication_Networks_II

Pseudo code: Routing Update

Router receives a type 3 packet:

num_ids = length

increment sequence number in packet by 1

for (i=1 to *length*){

if the CID/Host ID exists in the CID/Host routing table {

if ((hops for CID in table) > (hops for CID in pkt)||(hops for host in table) > (hops for host in pkt)){

Replace the entry in respective table with new entry for host or content

}

else {

Pop out that content or host from the packet

num_ids =num_ids - 1 (from the packet) //loop prevention

}

}

Pseudo code: Routing Update

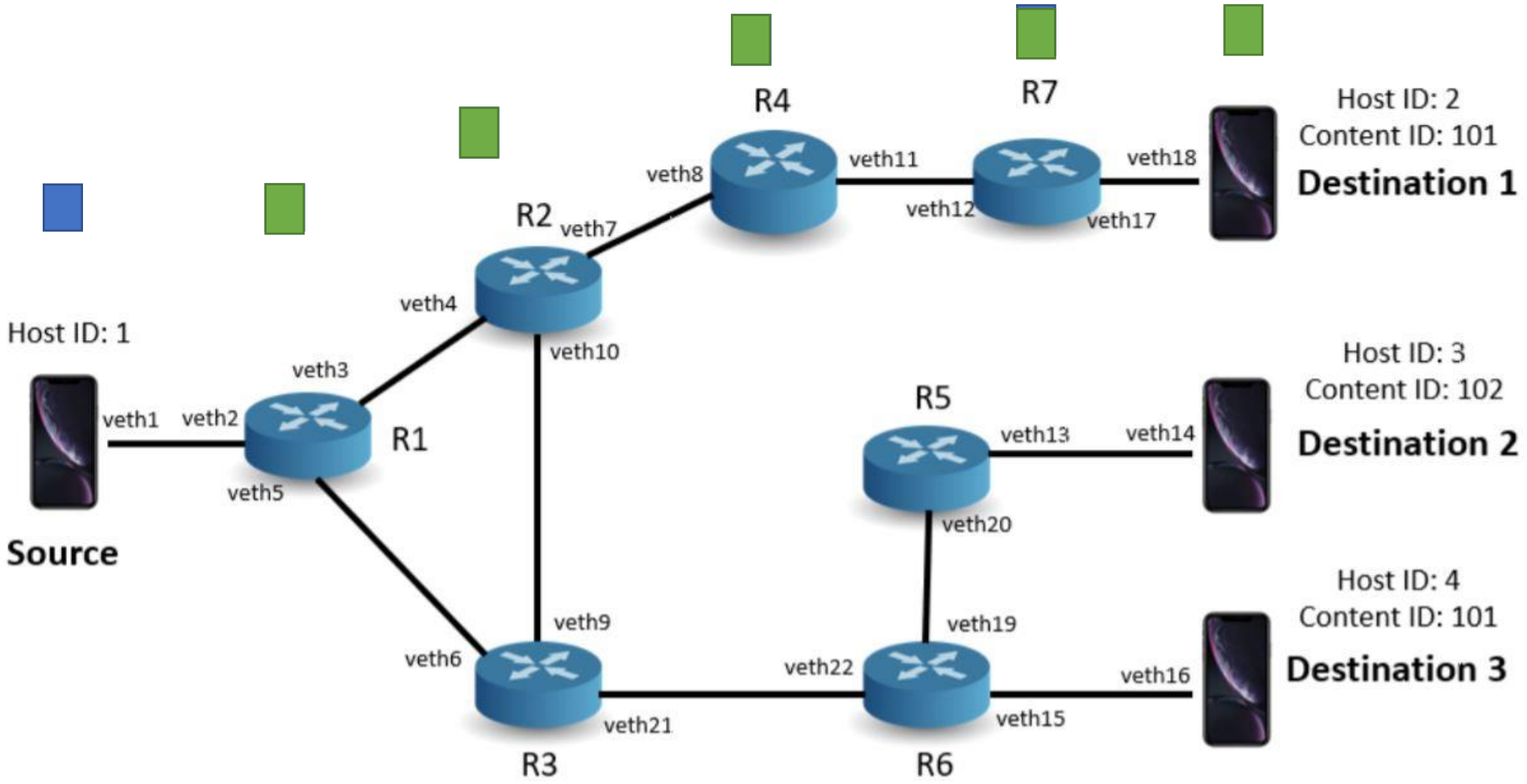
```
else {
        Add CID/Host ID with hop count along with port number to respective  table
        }
} //end of for loop

if (num_ids == 0){ // No IDs were updated
                Drop packet }//loop prevention
else {
                Forward the updated packet to all output ports
}
```

Pseudo code: Packet Forwarding

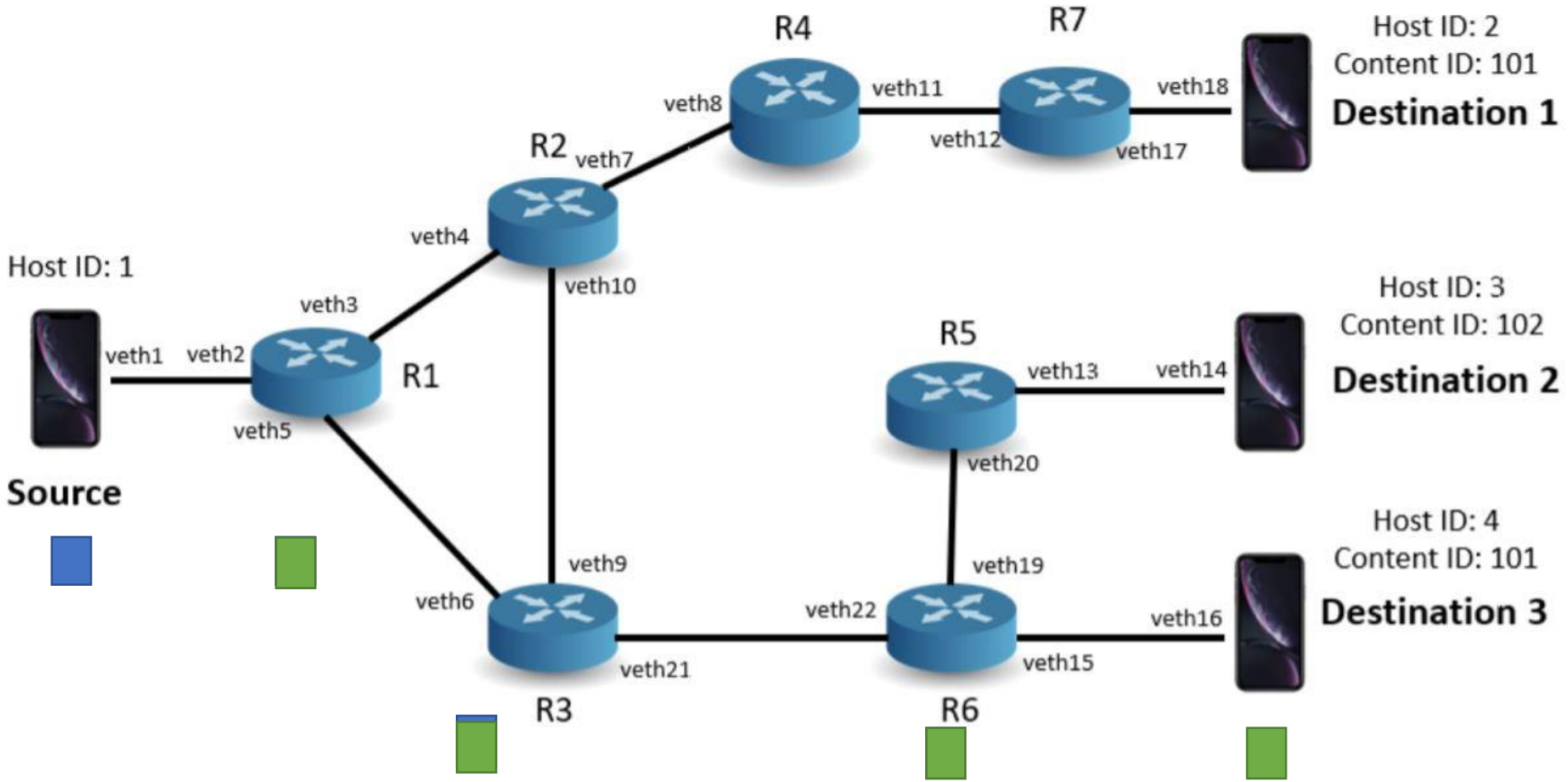
Date request/response packet received

```
if (Type == 2 || Type == 3|| Type == 5){ //data request packet type
      if (Dst address == Content ID){
            Lookup next hop in Content Table and forward packet}
      else {
            Lookup next hop in Host Table and forward packet}
        }
```

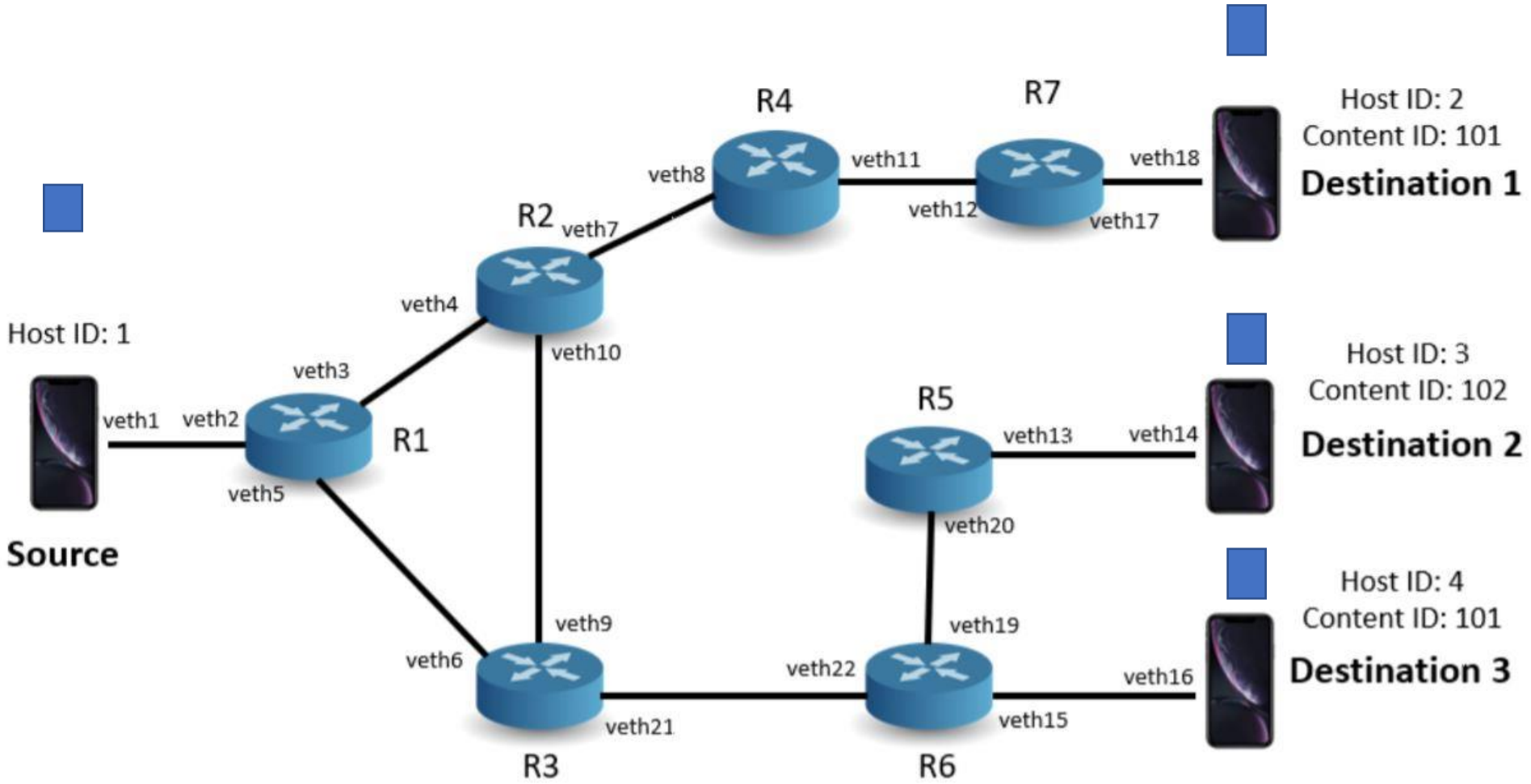
TRADITIONAL MECHANISM (UNICAST)



PROPOSED MECHANISM (ANYCAST)



BOOTSTRAP AND DISCOVERY



ROUTING UPDATE