

15-441: Recitation 6

PROJECT 2 INTRO

Project 2: IRC Routing Daemon

- ⦿ Focus: not really IRC anymore
 - All about routing: link-state
- ⦿ You **NEED** a project partner
 - You may **NOT** work alone

Finding a Project Partner

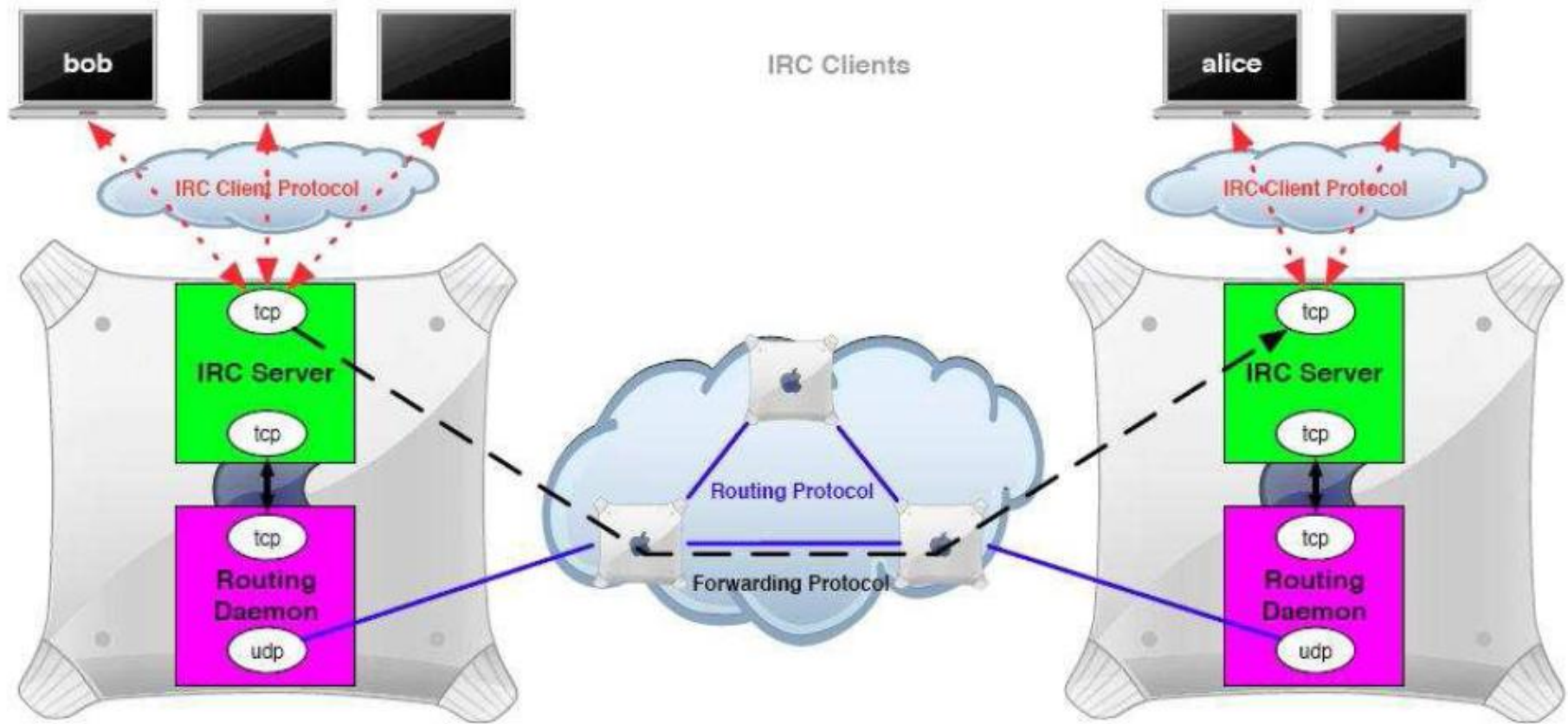
- ◎ This is *very* important, you will spend hours and hours with this person
- ◎ Start looking *soon*, inquire:
 - Class schedules
 - Additional constraints: research/work/life
 - Hopes & desires: striving for 100%? 80%?
 - Skills: someone good at routing logic and someone good at coding?

Post a Personal!

- ⦿ Talk to students
 - After class and in recitation
- ⦿ Additionally, mail the bboard:

Hi, my name is George. My class schedule is XXXXX. I have a job and I work from X-Y on days A and B....

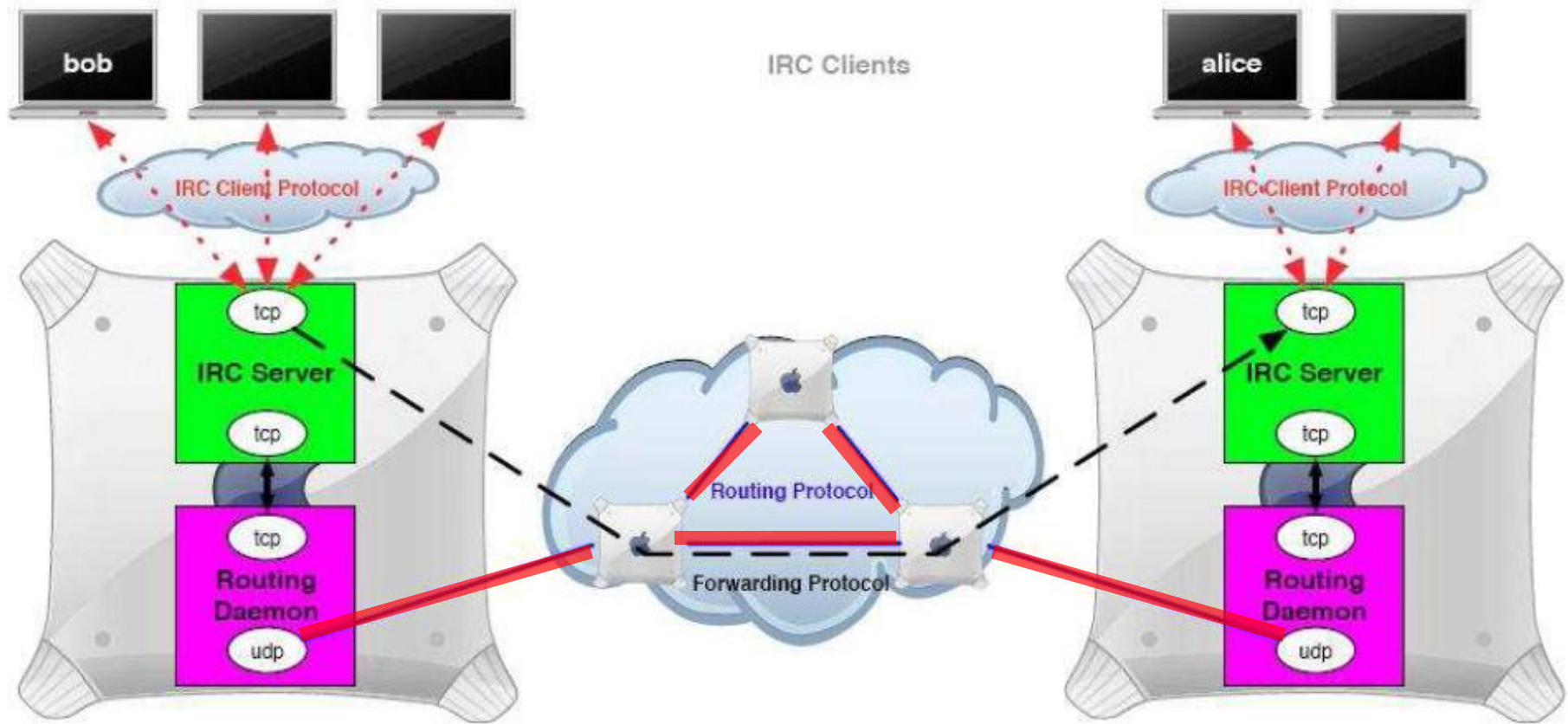
IRC Servers and Routing



New Definitions

- ◉ **Node** – IRC server + Routing Daemon
- ◉ **nodeID** – unique node identifier
- ◉ **Neighbor** – node's A and B are neighbors if there is a virtual link between them
- ◉ **Forwarding port** – same as IRC port
- ◉ **Local port** – TCP port for server $\leftrightarrow$ daemon
- ◉ **Routing port** – UDP port on daemon
- ◉ **OSPF** – link-state protocol similar to the one you'll be implementing
- ◉ **Routing table** – used to store paths

Inter-Daemon Communication



Link-state Routing Protocols

- ⦿ Protocols with a *global* view of the network
 - Every daemon knows about every daemon
 - Every daemon knows connectivity graph
- ⦿ Routing table constructed using shortest path tree
- ⦿ Flooding is used to propagate routing information

LSA: Link State Announcement

Version (1), TTL (1), type (2)
sender nodeID (4)
sequence number (4)
num link entries (4)
num user entries (4)
num channel entries (4)
Link entries (variable)
User entries (variable)
Channel entries (variable)

Version: always 1

TTL: initial(32) value--; every hop

Type: advertisement(0), ACK(1)

sender nodeID: original sender

sequence number: ++ per LSA

num _____: # of entries

Link entries: nodeID of neighbors

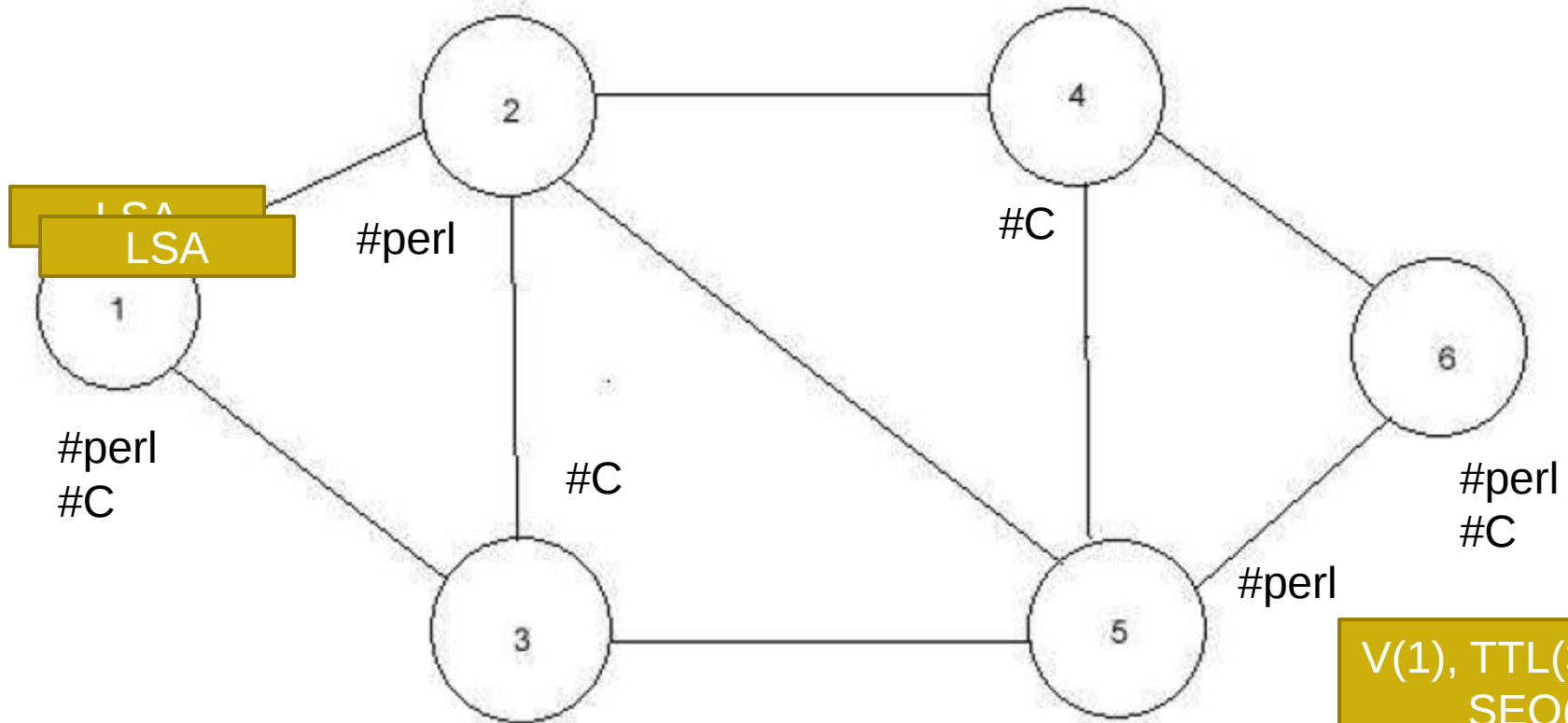
User entries: NICKs on server

Channel entries: channels on server

LSAs: When to Generate

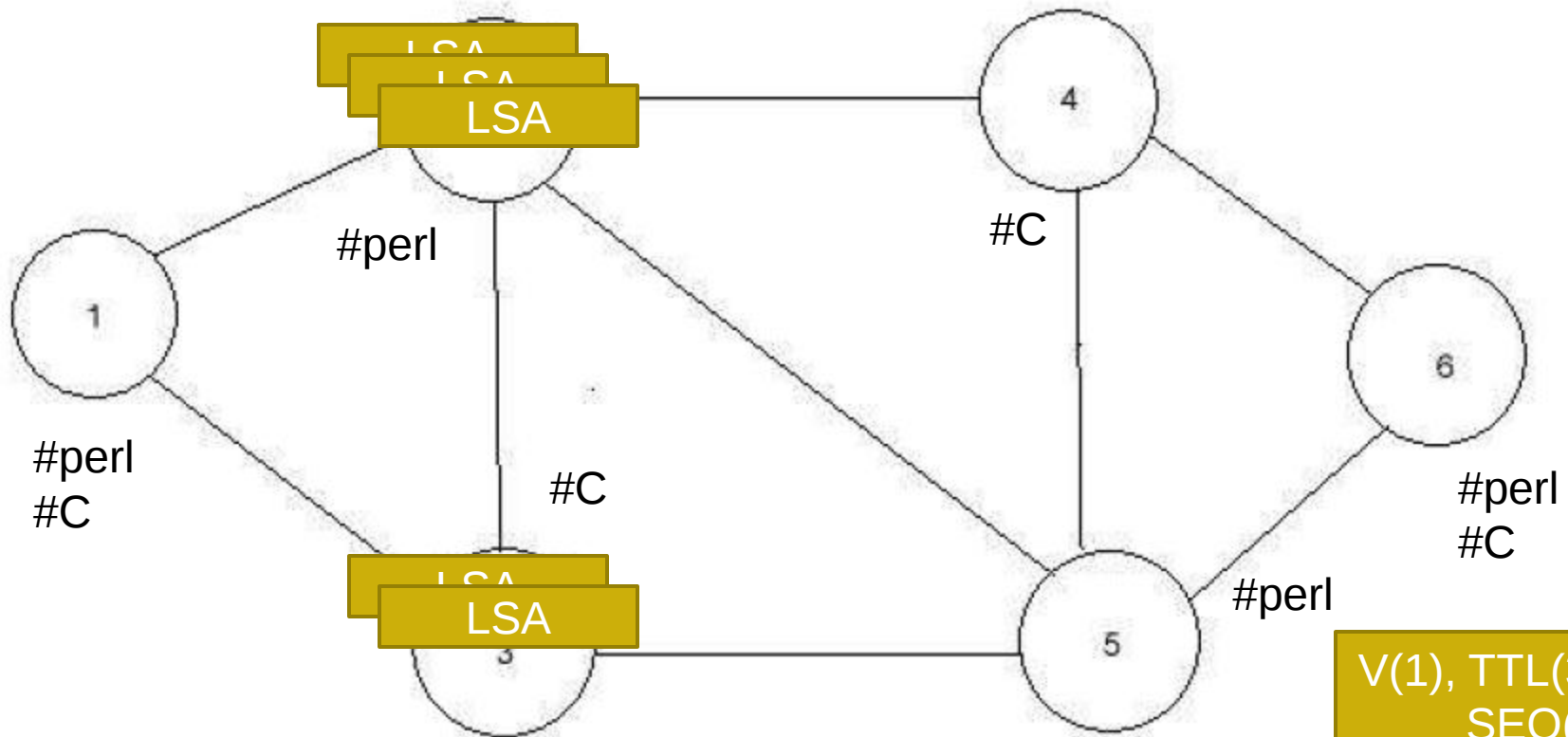
- ⦿ Generate LSAs periodically
 - Specified interval
- ⦿ Generate LSAs on-demand:
 - User joins or quits
 - First user on server to join a channel
 - Neighbor is detected as down (timeout)

LSA Propagation from nodeID 1



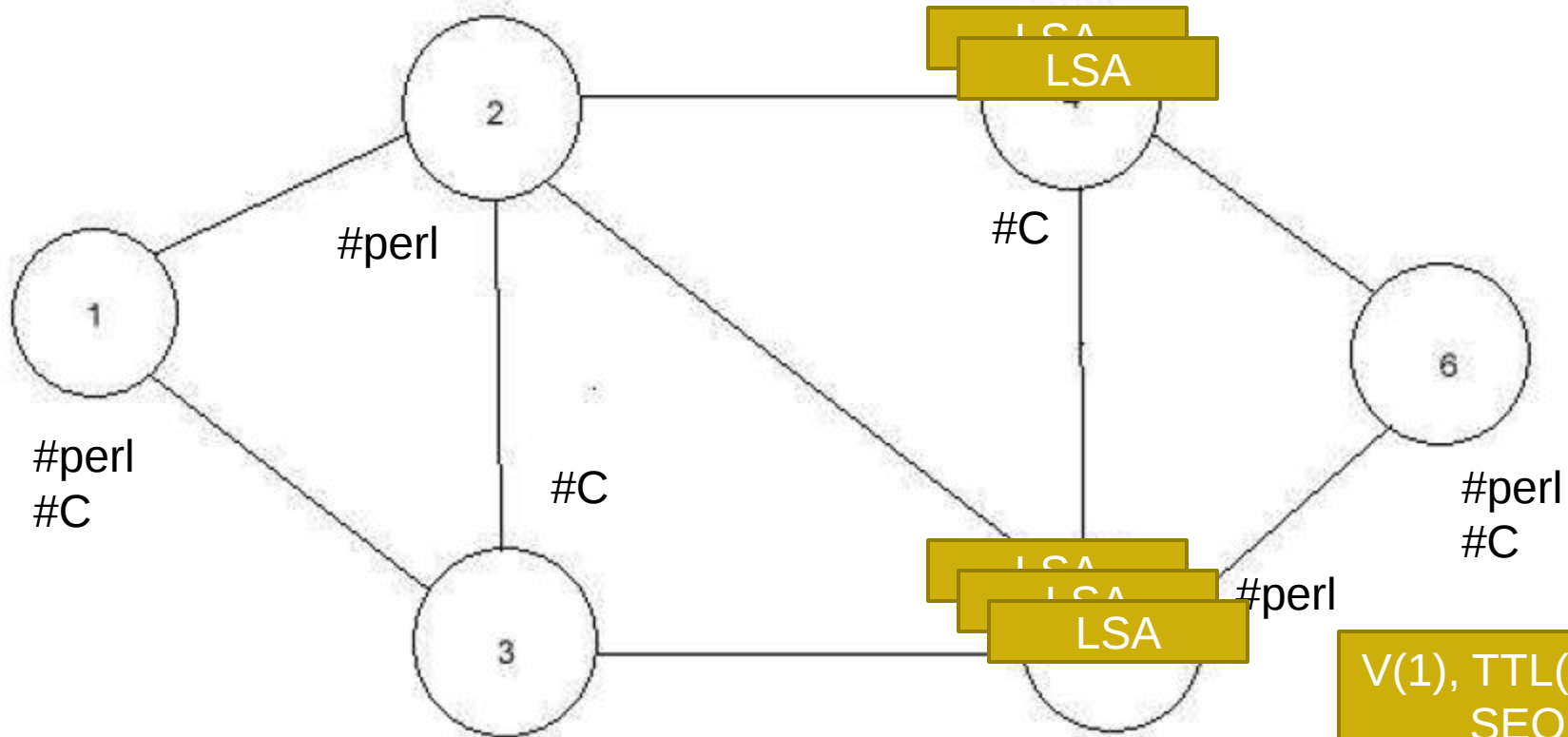
V(1), TTL(32), T(0)
SEQ(50)
snodeID(1)
L(2), U(0), C(1)
2
3
#linux

LSA Propagation from nodeID 1



V(1), TTL(32), T(0)
SEQ(50)
snodeID(1)
L(2), U(0), C(1)
2
3
#linux

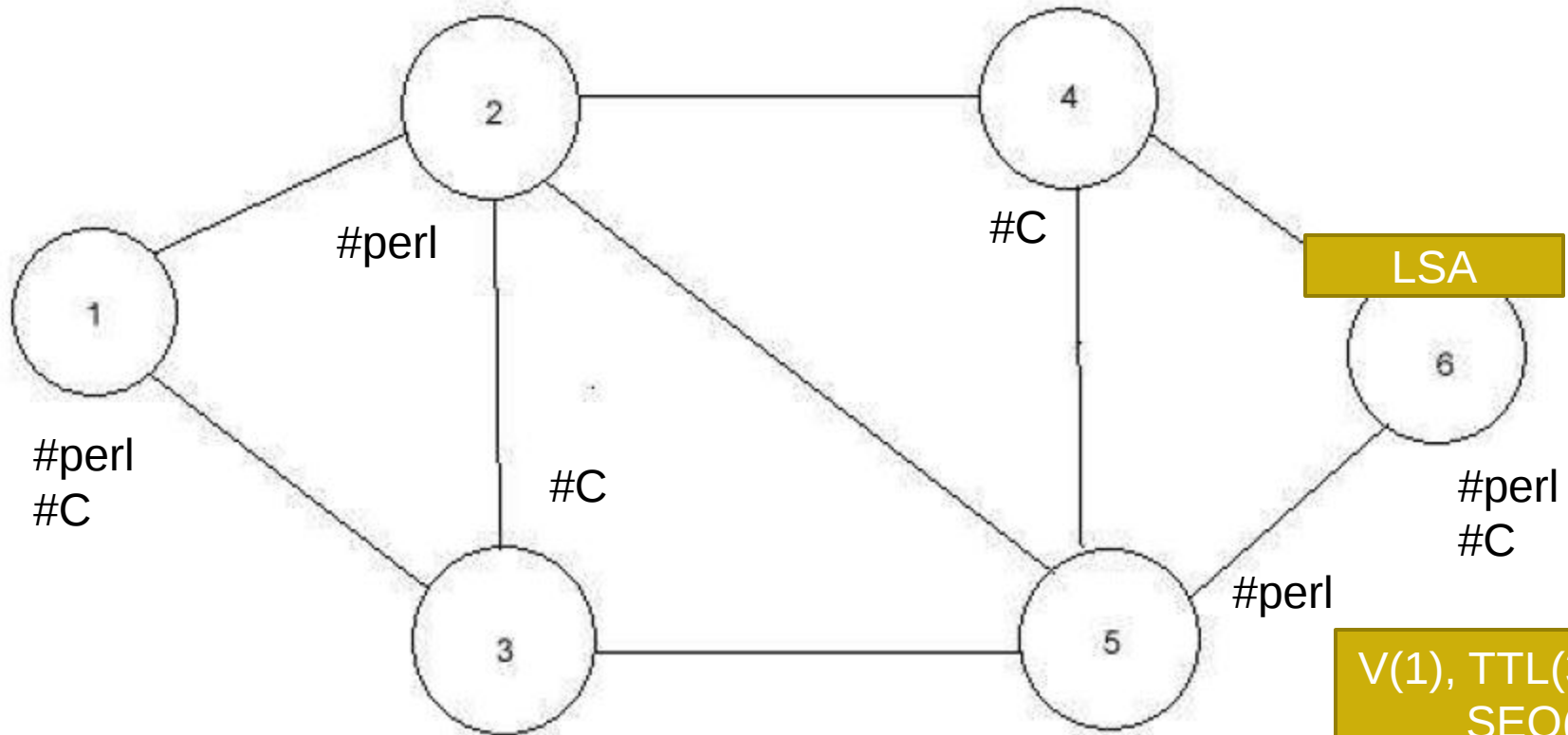
LSA Propagation from nodeID 1



Assume LSA from 2 got to node 5 before the LSA from 3

V(1), TTL(32), T(0)
SEQ(50)
snodeID(1)
L(2), U(0), C(1)
2
3
#linux

LSA Propagation from nodeID 1



Assume LSA from 4 got to node 6 before the LSA from 5

V(1), TTL(32), T(0)
SEQ(50)
snodeID(1)
L(2), U(0), C(1)
2
3
#linux

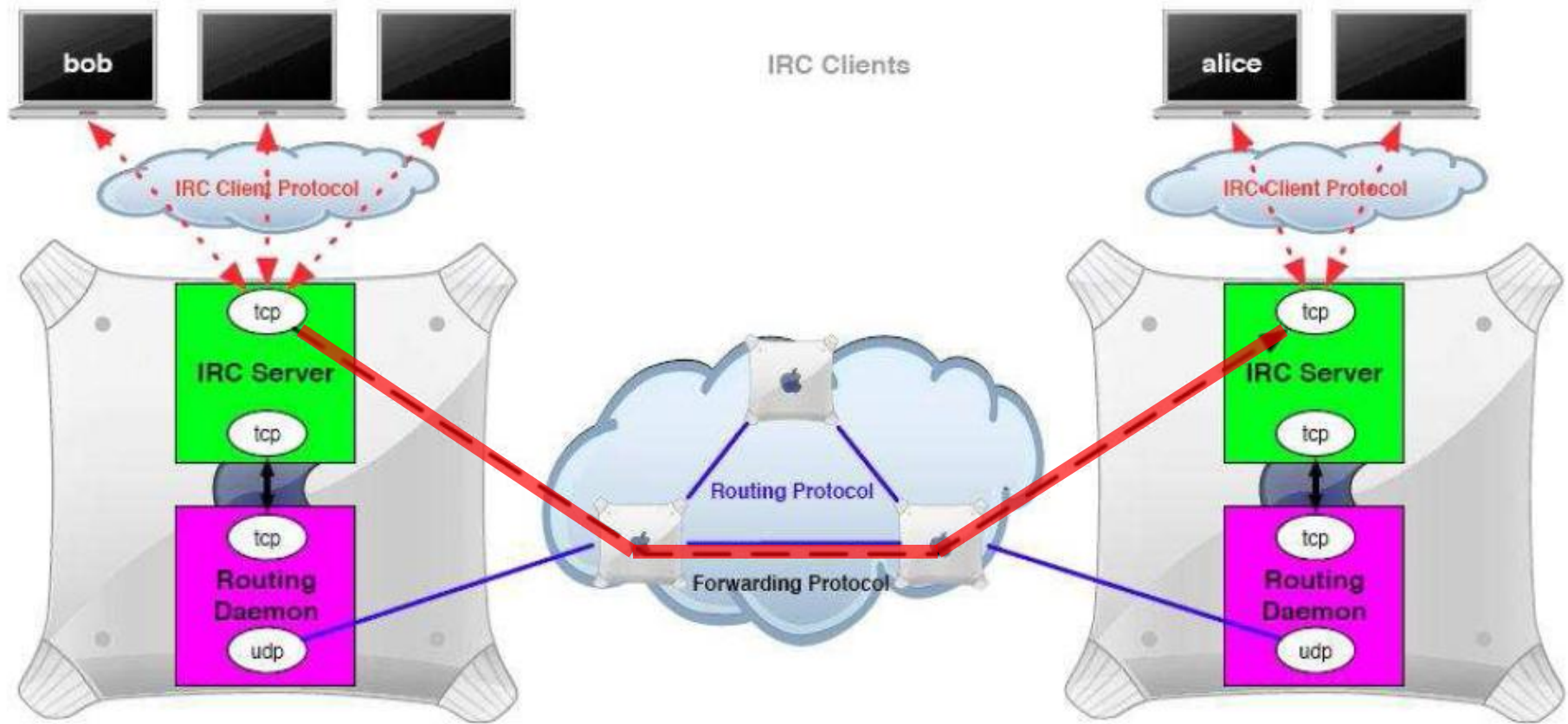
Details

- ⦿ Daemon $\leftrightarrow$ Daemon: UDP
- ⦿ Use acks to make reliable
- ⦿ Don't panic, sample UDP code going to be released

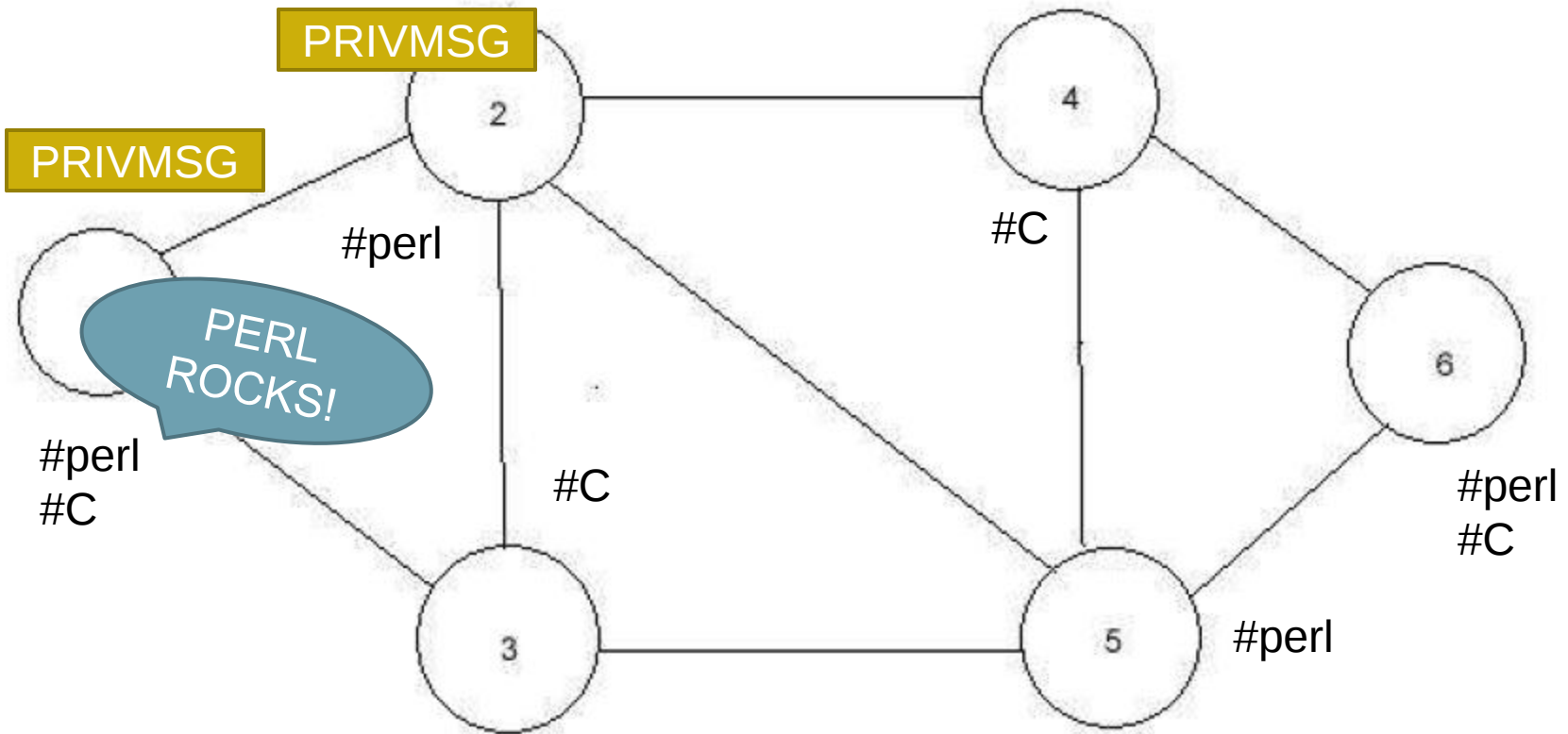
Computing Routing Tables

- ⦿ At every node:
 - Compute shortest path from every node to every node
 - Create next hops table
- ⦿ We will see in the following slides...

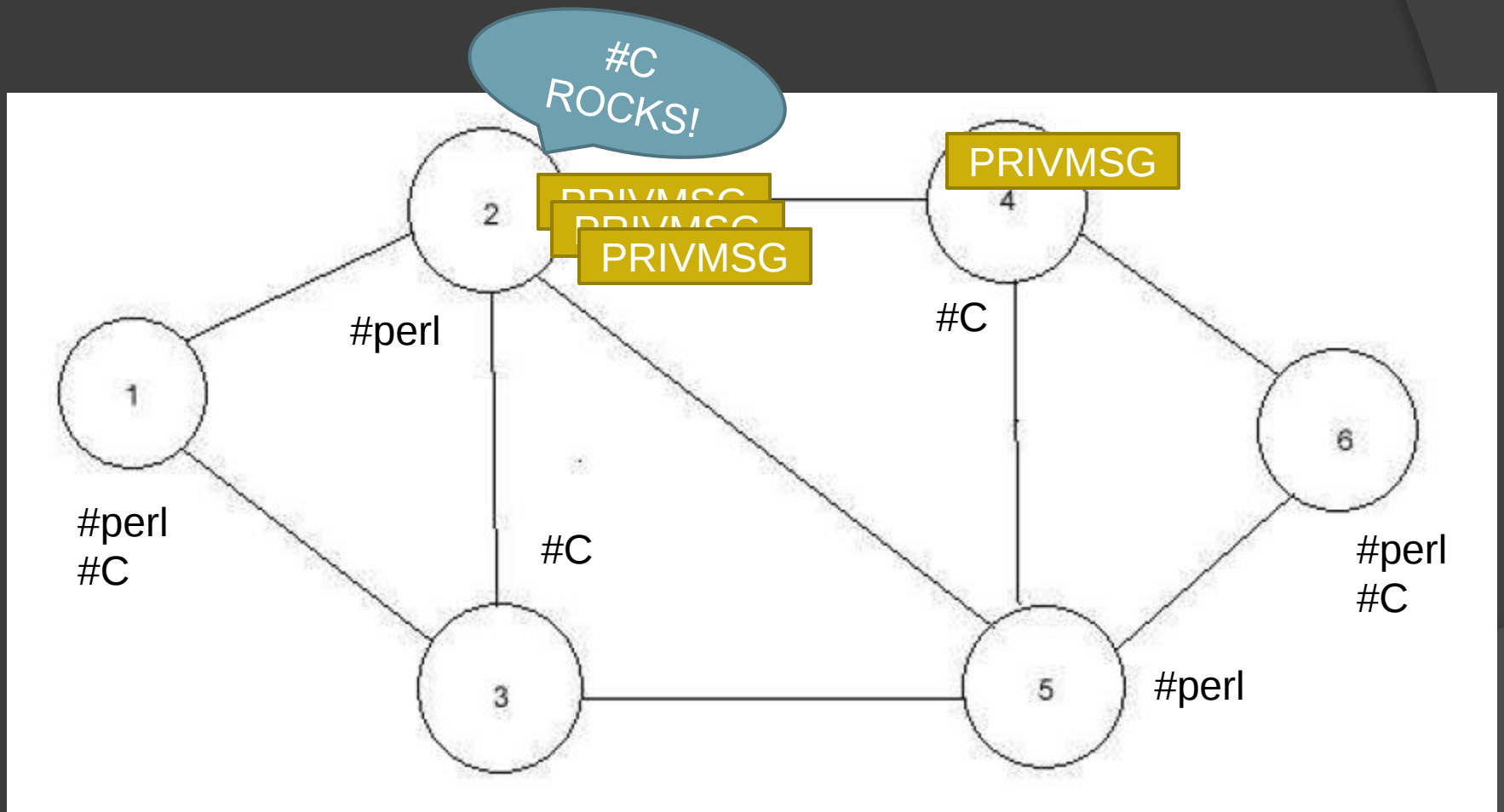
Inter-Server Communication

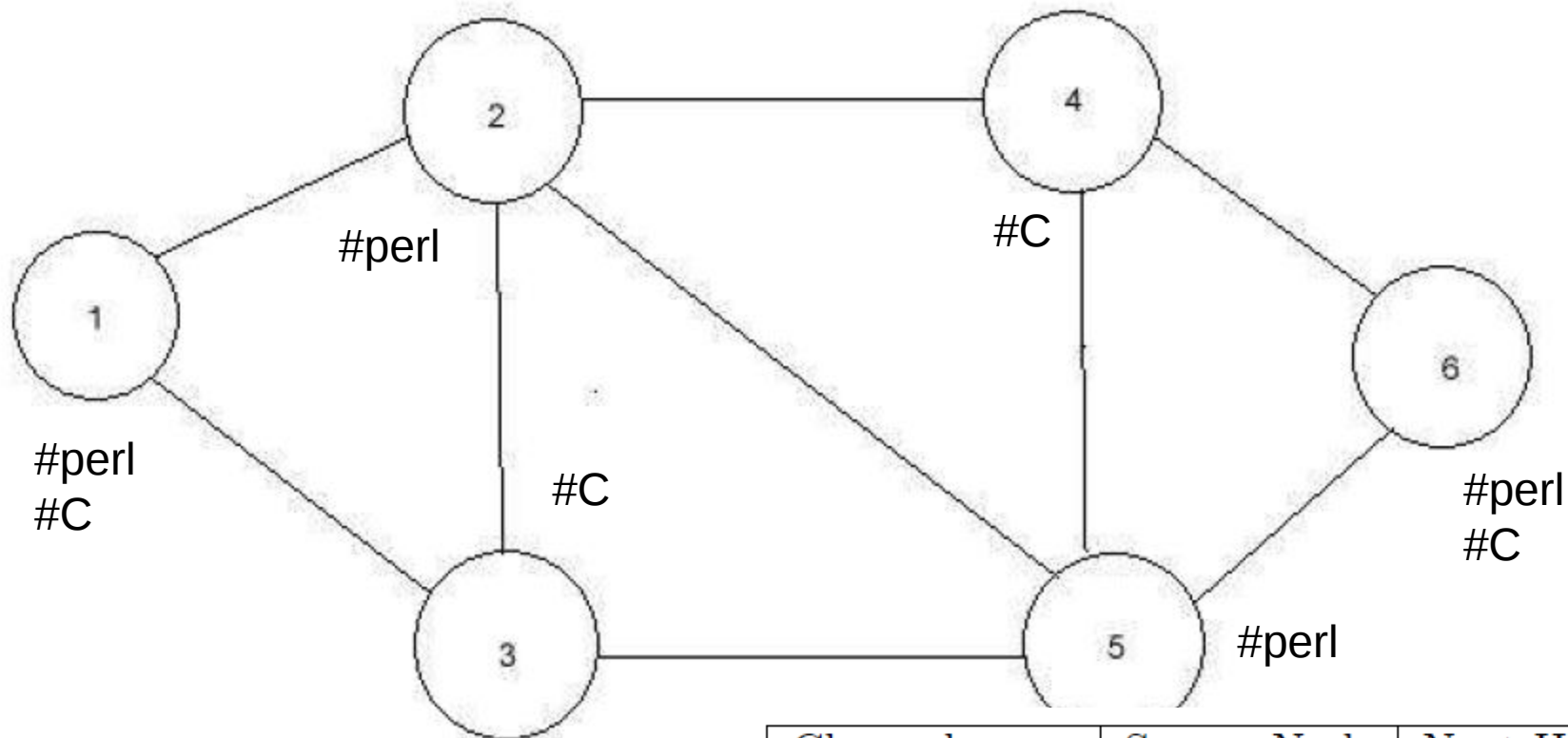


Routing: Posting to #perl



Understanding the Routing

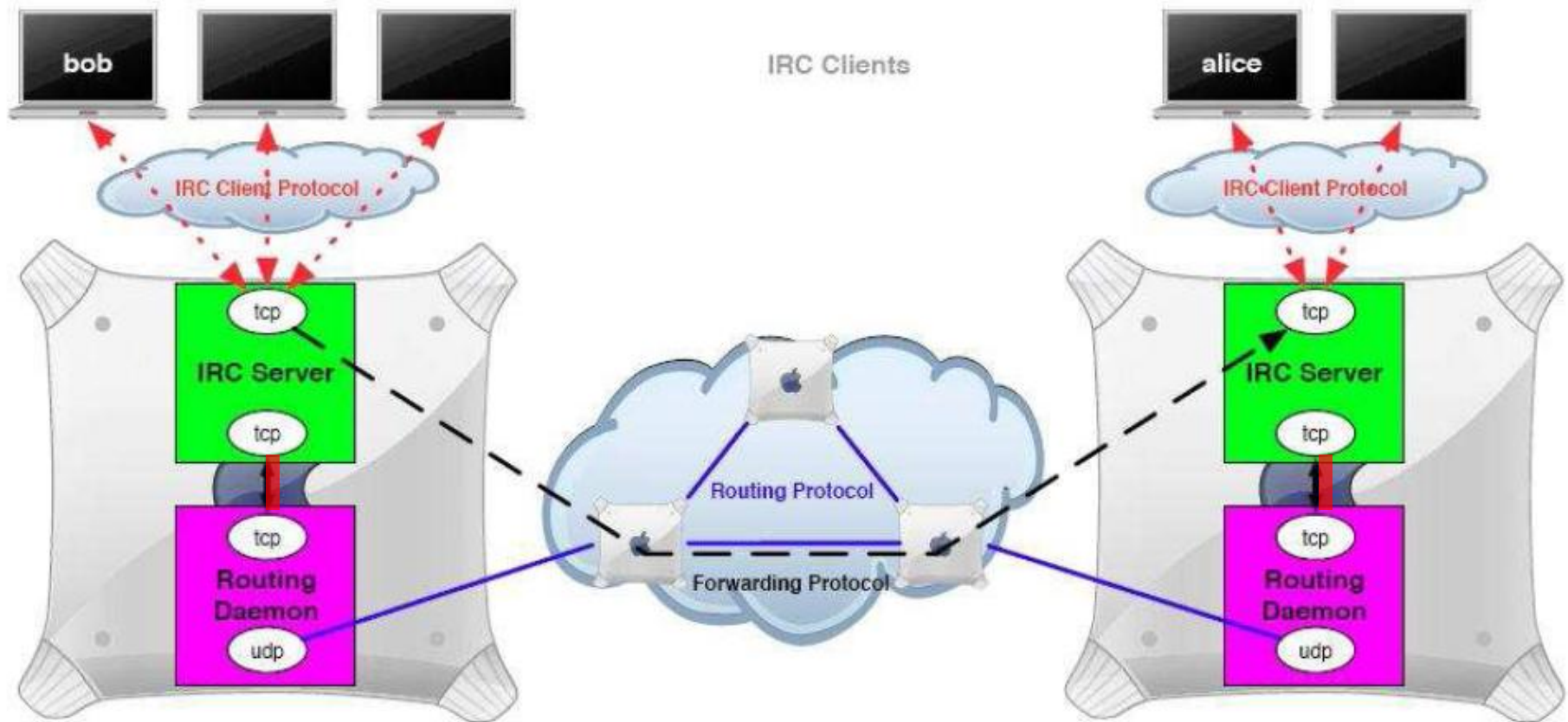




Node 1

Channel name	Source Node	Next-Hops
#perl	1	2
#perl	2	None
#perl	5	None
#perl	6	None
#c	1	2,3
#c	3	None
#c	4	None
#c	6	None

Server/Daemon Communication



Server $\leftrightarrow$ Daemon Comm.

- ⦿ Server and Daemon need to communicate
- ⦿ Server notifies daemon of updates
- ⦿ Networking details:
 - Server $\leftrightarrow$ Daemon: TCP

Server and Daemon Cmds

⦿ **Commands from server to daemon:**

- ADDUSER: adds a user
- ADDCHAN: adds a channel
- REMOVEUSER: deletes a user
- REMOVECHAN: deletes a channel
- NEXTHOP: nodeID of next hop for target
- NEXTHOPS: next hop, given a target
- USERTABLE: user routing table
- CHANTABLE: channel routing table

Announcements

- The rest of the details are going to be in the handout

Questions?