

# IP Help Session

CS168 Fall 2017

# How to run

- `./net2Inx [...].net` (AB.net, ABC.net, loop.net)
  - AB.net
    - Node A localhost
    - Node B 127.0.0.1
    - A <-> B
  - A.Inx :
    - Localhost 5000 (A)
    - 127.0.0.1 5001 192.168.0.1 192.168.0.2 (A <-> B)
    - cslab4g 5002 192.168.0.3 192.168.0.4 (A <-> C)
- `./node A.Inx; ./node B.Inx; [...]`
- See it run!

# In *node* REPL

- Interfaces : print interfaces
- Routes : print routes
- Send [IP] [Protocol] [Data] : send IP packet
  - If Protocol = 0, prints out packet on the receiving side!
- Up/down [interface] : enable/disables the specified interface
- Quit : quit cleanly!

Basic REPL, parsing .Inx files, and some debugging tools are given in C support code.

# Abstraction of Layers

- Interface
  - Abstraction of a network interface chip : think of “*ip a*”
- Link Layer
  - Abstraction such that “sendable to nodes without routing”
  - Underlying abstraction of UDP (physical host / port in [file],Inx) shouldn't be accessible to any upper layers
- IP Layer
  - Abstraction of network layer, handling IP packet sending, forwarding, parsing, TTL, ...
  - Relies on upper layer (here, RIP) to resolve routing, so IP doesn't need to know in detail
- RIP Layer
  - Abstraction of RIP running on top of IP
  - Main routing service tool for IP, Should register its handler to IP

# Milestone (By Monday, Oct. 9th)

- No coding needed!
- We will ask questions about your design, i.e.:
  - What is your general architecture? How will you divide the work/components?
  - What should happen when (...) ?
  - What threading/mutexes/data structures are you planning to use?
  - ...
- You can reach out to ask specific questions about your design beforehand!

# Final (By Monday, Oct 16th)

## - Requirements

- Assume Link Layer is 100% secure - no delay, no replication, no loss (maybe except when mtu is low)
- Will basically test functionality based on (AB.net, ABC.net, loop.net) and see whether up/down/send/prints work correctly

## - Capstone

- Fragmentation : MTU changing, Fragmentation & reFragmentation, Reassembly
- Traceroute : Shows shortest path to any reachable node, up/down changes this
- Multicast : Sending to multiple receivers with shared “multi\_ip”

# Questions?

- If you find any bug (or something is fishy) in our reference node, please let us know quickly so that we can fix it for you!

**Have fun coding!**