Daniel Erenrich
Emil Ibrishimov
Nathan Watson
Robert Shimizu

M1
ARCHITECTURE AND TASKS

CS 143 Network Simulation Project

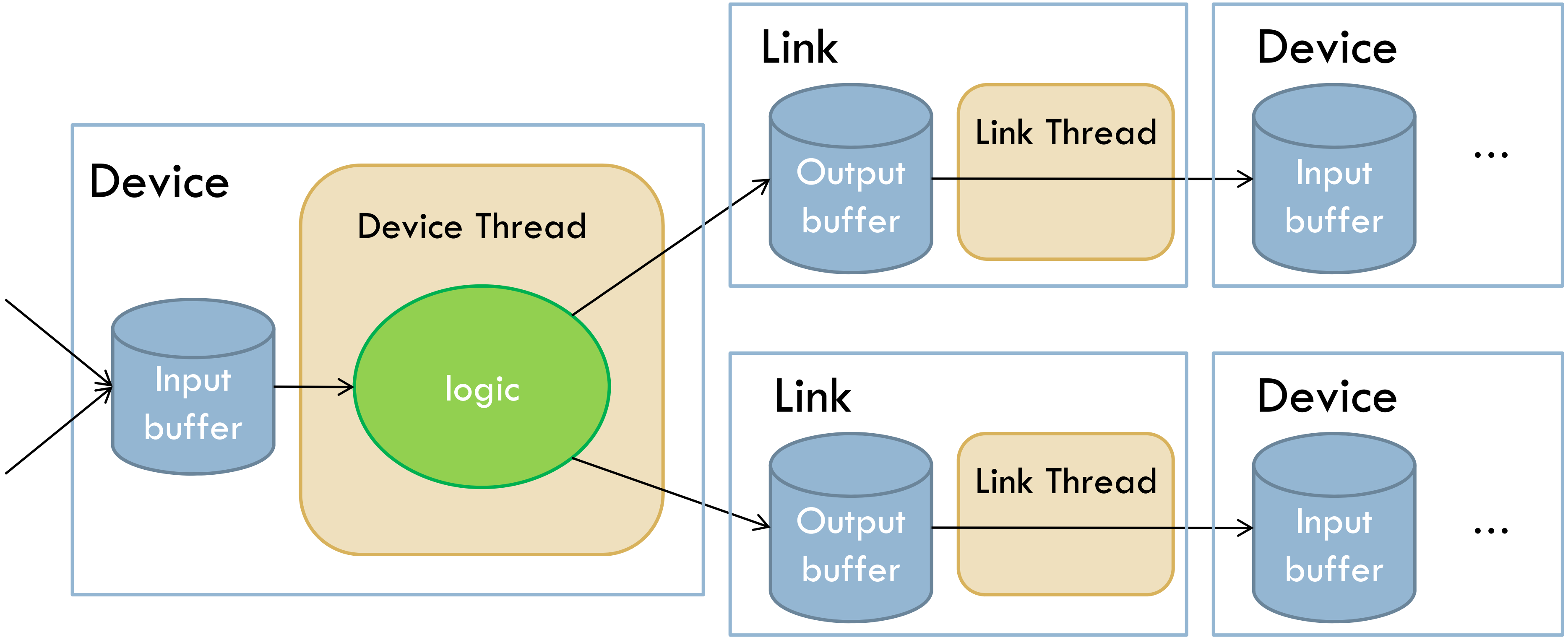
Goals

- Keep the design simple but robust
- Required features
  - Bellman-Ford routing
  - Reno TCP and FAST TCP congestion control
- Additional features
  - Low-delay “lossless” connections (error correction, no ACKs, out of order)
  - Malicious behavior: DNS spoofing, bandwidth stealing

Architecture

- Simplifications
  - Ports are unnecessary – multiple programs on the same host can be modeled as multiple hosts behind a single bottleneck
- Simulation details
  - Two types of objects: devices and links
  - Simulate in real time (maybe support accelerated time)
  - Each object on the network has its own thread

Network Structure



Project modules

Controller
- Terminal, Network Builder
- Owned by Robert

Skeleton
- Abstract Device, Link
- Owned by Emil

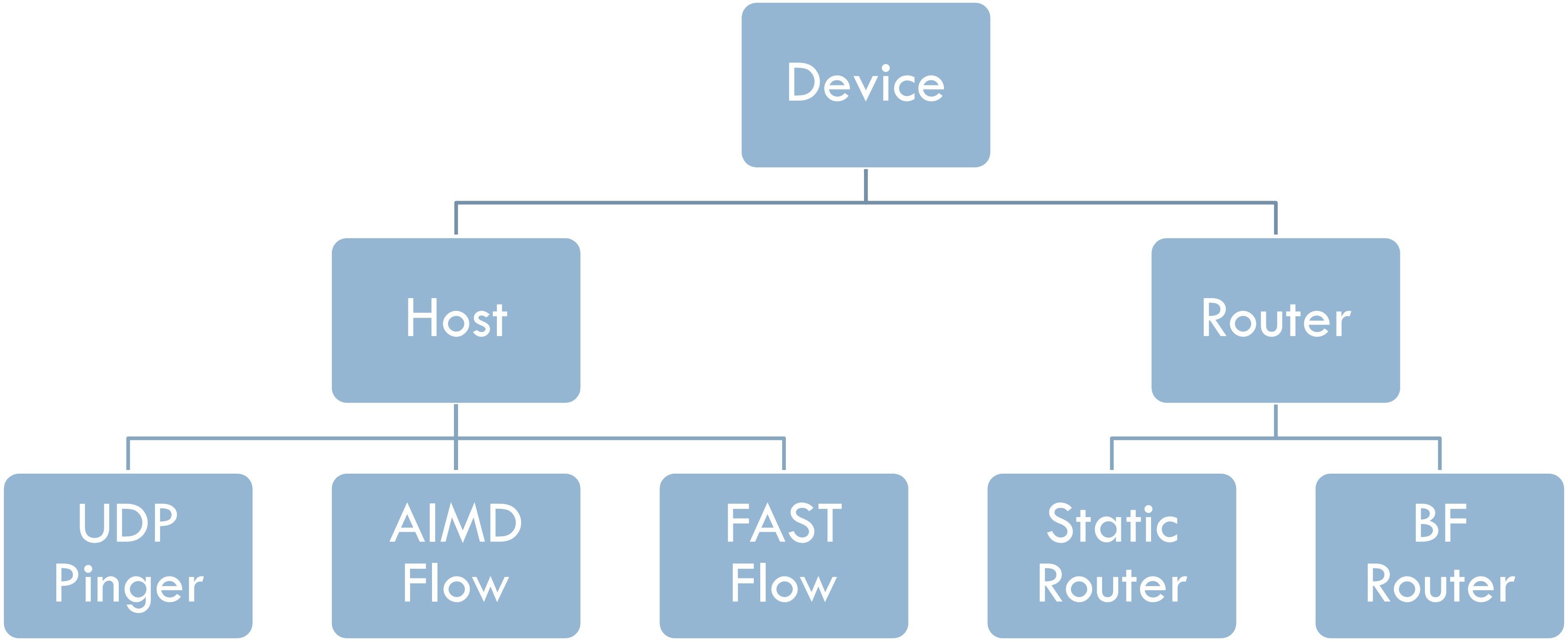
Routers
- Implement routing protocols
- Owned by Daniel

Hosts
- Implement transport protocols
- Owned by Nathan

Device

- Any object that can receive or send information on the network
- How it works
  - Input buffer stores all incoming packets
  - A thread processes packets from the buffer when ready
- Interface
  - abstract process() – implemented by subclass to deal with received packets
  - receive() – called by incoming links when a packet arrives

Devices



Device
Host
Router
UDP Pinger
AIMD Flow
FAST Flow
Static Router
BF Router

Link

- Transmits packets between devices
- How it works
  - Output buffer stores all packets waiting to be transmitted
  - A thread processes packets from the buffer when ready
- Interface
  - send() – called by Device that wants to transmit a packet through the link

Plan for M2

- Develop the Skeleton
- Create initial version of the controller terminal
- Implement UDP Pinger and Static Router
- Test