

CS 249A  
Assignment 3 Review Session

Aditya Ramesh

# Overview

- Builds up on Assignment 2
- Run actual simulation on network
- Composed of multiple parts to implement
- Heavily uses activities and reactors
- Open-ended design
  - Implementation and output are under your control

# Error Handling

- Rep layer can throw exceptions back to client now
- Determine which exceptions should be caught at rep layer and which should propagate back
- Create exception hierarchy as discussed in Chapter 7

# Virtual Time Activities

- Activity Manager
  - Determines which activity to execute next
  - Add and remove activity objects
  - Sample code (ActivityImpl.cpp) has rough design for Activity and Activity Manager
- Virtual Time
  - Client controls simulation by setting current time
  - Simulation execute until reaches desired time (from client)

# Shipments

- Shipment
  - Source location, destination location
  - Number of packages
  - Customer can only accept, not transfer packets
    - Return shipments back or don't forward it to customers
  - Other locations can transfer shipments along path
    - Capacity constraints
    - Use routing algorithms to determine next location
    - Can assume network will not change once simulation starts

# Simulation

- Maintained through activities and reactors
- Once shipment attributes set, have reactor create activity to inject shipments into network
- Reactor at location should respond to received shipments and forward them
- Use timeout notifiee to indicate that segment is free

# Real-time Activity Manager

- Advance virtual time by having activity for real-time manager
- Advance virtual time with scaling factor
  - E.g 1 second of real-time is equivalent to one hour in network
- Make calls to sleep to slow-down simulation if needed

# Simulation Tests

- Create networks to show how changing attributes can result in smooth and heavily congested networks
- Create bottleneck network (100 source -> 10 terminal -> 1 terminal -> 1 destination) and vary package size
- Report results in README and submit as `experiment.cpp`

# Groups

- Implement 2 different routing algorithms (Dijkstra's, BFS, etc)
- Implement scheduled changes to fleet attributes
  - Client specifies time range and value
  - Activity scheduler track when changes occur and modify attributes appropriately
  - Assume that packages in segment have fixed values
- Additional attribute for Customer

# Extensions

- Fancier routing algorithms
- Congestion control mechanisms
- Detailed statistical information about network
- Non-network related ideas include database connectivity, inter-language operability, etc.
- Ask CAs about any potential ideas
- Get up to 10% extra credit

# Conclusion

- Attributes and reactor-based approach
- Much harder than first two assignments, so start as early as possible
- Very open-ended
  - README file has to be detailed
  - Give good documentation and explanation, particularly for test cases