# CMPE 257: Wireless and Mobile Networking

GloMoSim Overview

- Class Administrative Issues
  - Accounts on the server.
  - Mailing List

- T.A hours

# GloMoSim Overview

- **Installation.**
  - OS (Freebsd, linux, solaris, windows)
  - Requires parsec binaries.
  - Download glomosim-2.0.3 from the projects page.
  - Unzip and untar the files
    - "tar –zxvf *filename"* or "gunzip *filename* | tar xvf – "
  - Edit makefile in glomosim/main directory
    - #Parsec
    - ## echo Using Parsec in ${PCC_DIRECTORY}
    - ## ${PCC_DIRECTORY}/bin/parsecc $*
    - ---- Lines to be added
    - PARDIR = "path to the parsec directory"
    - PAR = $(PARDIR)/bin/pcc
    - PARFLAG = -O2 -g $(COMFLAG) -clock longlong -pcc_directory $(PARDIR)
    - ---- Delete the other "PARFLAG line"
      PARFLAG = $(COMFLAG) -clock longlong
  - Type make from glomosim/main.
    - Execute /bin/glomosim config.in

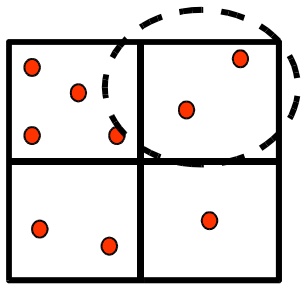
# GloMoSim Details

- Discrete-event simulator based on parsec
- Parsec is essentially C with extensions
  - parallel executing ENTITIES
  - Entities exchange messages using the **send** and **receive** primitives
- Some extra type declarations such as *clocktype, addr_t*

# GlomoSim Model

- All nodes are aggregated into PARSEC aggregation entities for:
  - Scalability
  - ease of neighborhood calculations Current release supports only 1 partition

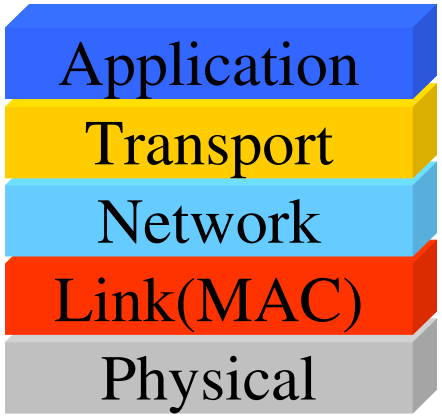
- Node state is maintained in a global data structure but simulation code for one node does not access other nodes' state



GlomoPartition

# GlomoSim Model (cont'd)

- Follows a layered approach for a network protocol architecture




| |
|---|
| FTP, TELNET, HTTP, CBR |
| TCP, UDP |
| AODV, DSR, LAR, ODMRP, ZRP |
| MACA, CSMA, 802.11, TSMA |
| Free space, Two ray, Rayleigh, Ricean |

- Uses *one* entity for all the communication layers for ease of inter-layer communication
- Neighboring layers exchange messages by fixed APIs

# Protocol Layer Functions

- Each layer has 3 major functions
- Initialization Function

```
void NetworkLar1Init(GlomoNode *node,
                     GlomoNetworkLar1** lar1,
                     const GlomoNodeInput *nodeInput);
```

- Finalization Function – statistics

```
void NetworkLar1Finalize(GlomoNode *node);
```

- Message Dispatcher Function

```
void NetworkLar1HandleProtocolPacket(
                     GlomoNode* node, Message* msg);
```

- Data structures:
  - GlomoNode – State of a node (handle of a node)
  - Message – Message or packet
  - GlomoNetworkLar1 – Protocol Specific information (e.g. statistics variables)
  - GlomoNodeInput – configuration options from command line

- Read the manuals (they are relatively small!):
  - PARSEC **http://pcl.cs.ucla.edu/projects/parsec/**
  - GlomoSim **http://pcl.cs.ucla.edu/projects/glomosim/**

- Read the source code:
  - You may need to modify it
  - You may need to watch the flow of packets from one layer to another

- You can subscribe to the mailing list by registering
  - http://pcl.cs.ucla.edu/projects/glomosim/obtaining_glomosim.html