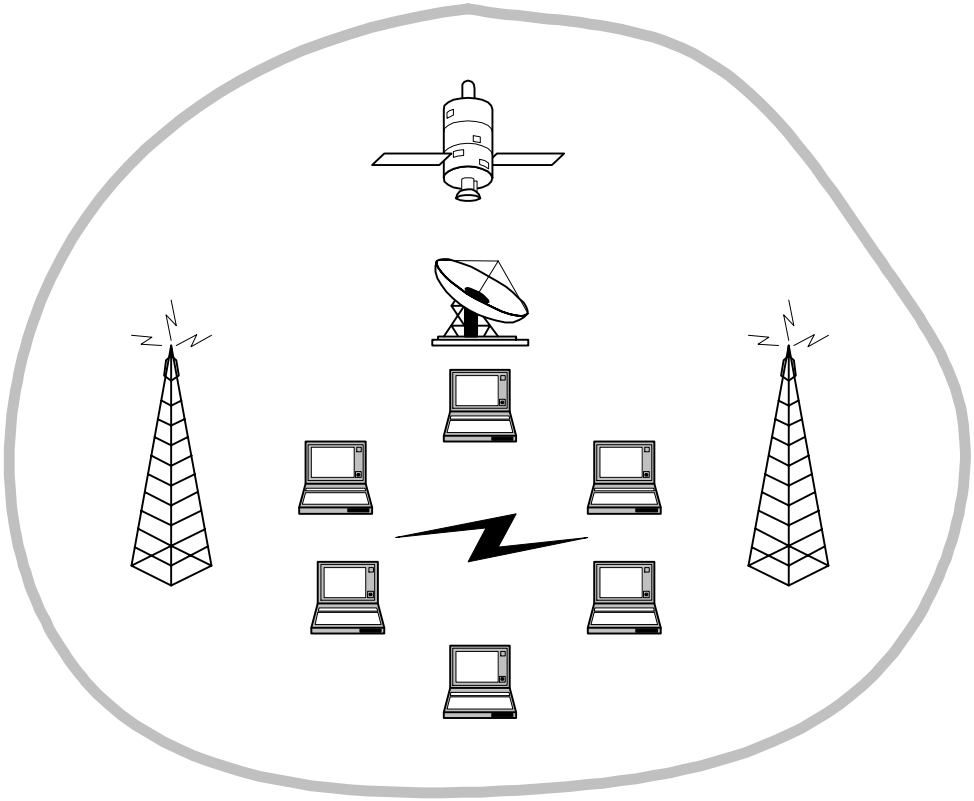
Self-Organizing Networks

Suman Banerjee

SDAG Meeting

April 21, 1999



What is it ?

- Independent nodes
  - Geographically distributed and maybe mobile
- Node availability erratic
  - Arbitrary joins and leaves
  - No stable nodes
- Limited resources
  - Power, Computation, Bandwidth, etc.
- Usually communication is wireless

Provide all network services to applications

Why need new protocols ?

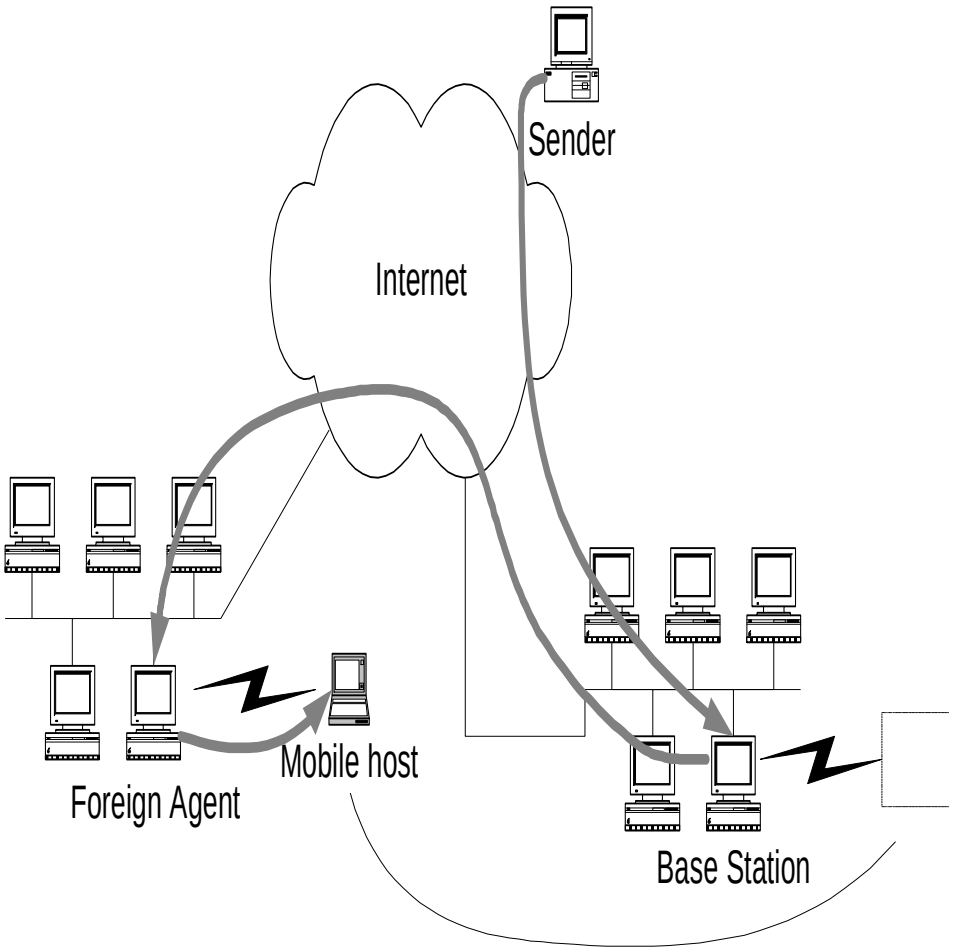
- Current Internet protocols have different assumptions
  - Dedicated backbone routers
  - Very high availability of backbone routers
    - Near zero downtime
    - Very rare changes in backbone topology
      - Protocols like BGP are feasible
  - Backbone routers more powerful with
      - Dedicated route processors
      - Substantial memory and caches

Wired vs Wireless

- Wired backbone
  - Low BER
  - Higher bandwidths
  - Point to point channels
    - No interference
  - Usually not mobile
    - e.g. Internet backbone
- Wireless backbone
  - High and fluctuating BER
  - Lower bandwidths
  - Broadcast channels
    - Interference among nodes
    - Hidden / exposed terminals
  - Mostly mobile
    - Dynamically changing neighborhood

Protocols for wired backbones cannot be applied to self-organizing networks

Mobile IP



- Add mobility feature to Internet
- Mobile host (MH) register with base station on home network
- Base station forwards packets to foreign agent
- Foreign agent forwards to MH
- IP tunneling to optimize

Why not Mobile IP ?

- Basic routing infrastructure same as before
- Mobile hosts are dynamic, but connect directly to the stable Internet
  - Base station and Foreign agent are stable
- Not meant for multi-hop dynamic connectivity

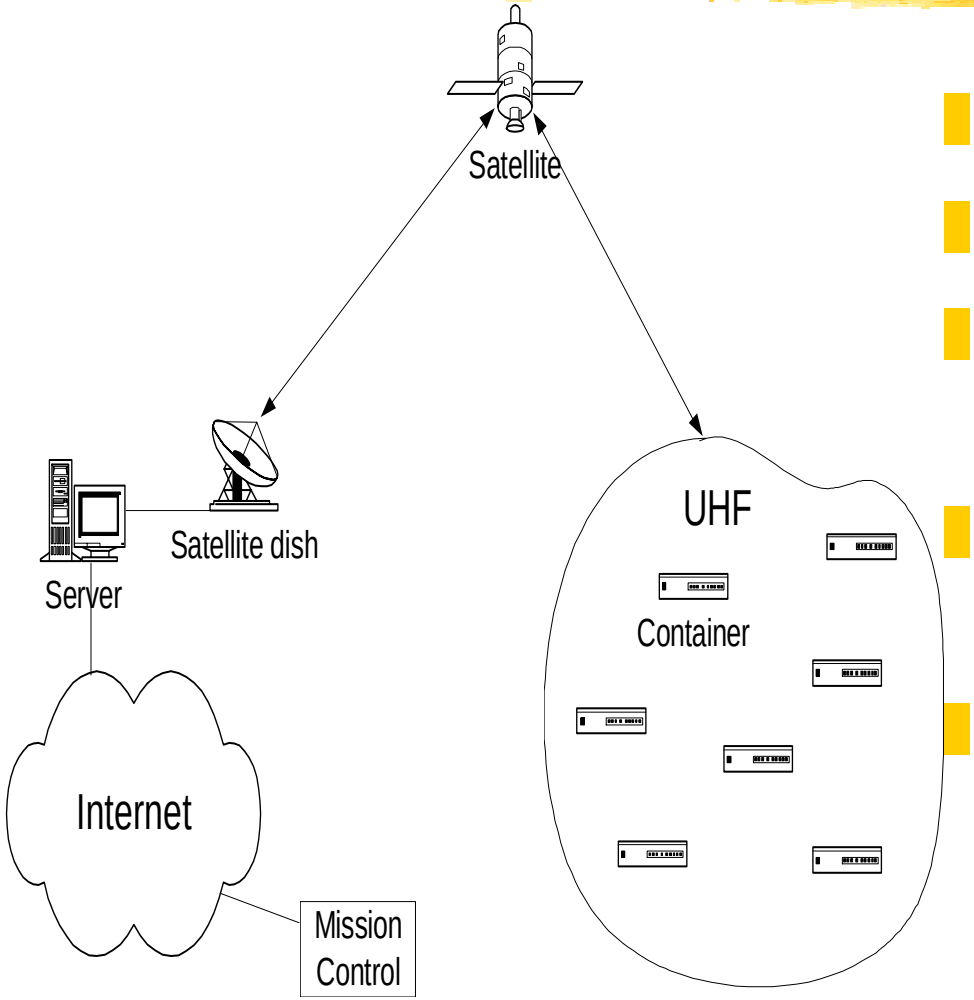
Roadmap

- Introduction and motivation
- Applications
- Design goals
- Research directions
- Some approaches
  - Some related work
- Sample Architecture
- Final questions and summary

Applications - I

- Sensor networks
  - Nodes have sensors
  - Missions
    - Aggregate and report regularly
    - Triggered reports of "critical events"
    - Disseminate commands from mission control
  - Nodes rarely have direct human interaction
    - i.e. no applications like telnet, web browsing etc.

Example - Container shipment

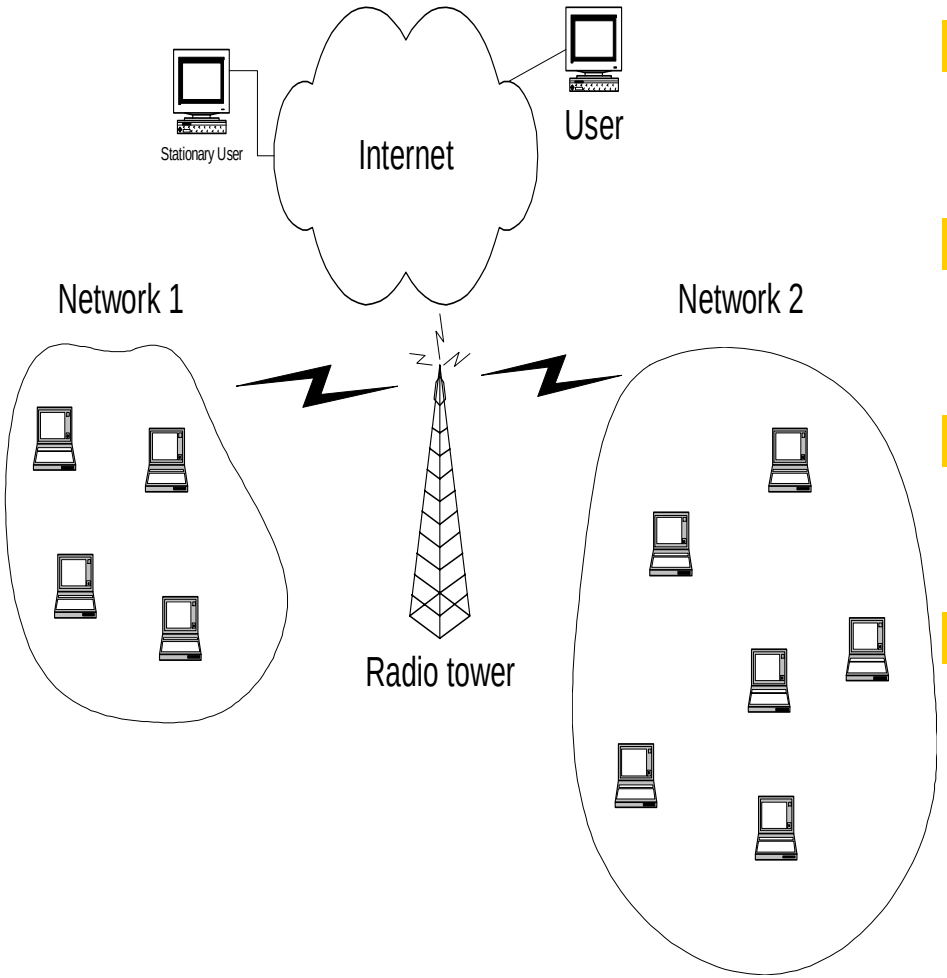


- Containers are in motion
- Report to mission control
- Communicate mutually using UHF
- Satellite communication is expensive - power
- Organize into networks to preserve satellite bandwidth and power

Applications - II

- Palmtop-connectivity networks
  - Nodes are handheld PDAs or equivalent
  - End-user applications run on nodes
  - Users communicate and exchange info
  - Capability to interact across the Internet
  - Examples
    - Combat teams
    - Exploratory expeditions
  - Can extend to home networking applications

Example - Combat teams



- Interactive users with PDAs
- Exchange info mutually and across Internet
- Power conservation needed
- Organize into networks to minimize transmission to radio tower

Design goals

- Conserve power and bandwidth
  - Load balance routing tasks
  - Minimize communication
- Robustness in presence of joins and leaves
  - Prevent network partitioning
  - Quick incorporation of new nodes
  - Fault tolerance with backups
- Scalability
  - Hierarchical arrangement

Research directions

- Application services
- Addressing and naming schemes
- Routing protocols

- Related Issues
  - Load balancing strategies
  - Synchronization for communication
  - Initialization procedures
  - Merging and splitting networks

Roadmap

- Introduction and motivation
- Applications
- Design goals
- Research directions
- Some approaches
  - Some related work
- Sample Architecture
- Final questions and summary

Application services

- Services needed from any capable unit
  - Primarily for sensor networks
  - Do not care which exact unit
  - Similar to web service
    - A specific web page from any site (any mirror or original)
- Need addressability to services and physical locations

Features like multicast, anycast, etc.

What kind of addressing ?

- Service-based addressing
  - All units that provide a specific service
- Geographical addressing
  - All units in a specific geographical location
- Combination of both

Addressing and Naming

- Need to distinguish between
  - Name
    - Identifies the unit uniquely
  - Address
    - Contains routing information
- Address may change, name should not
- Not true in current Internet
  - Acceptable, since addresses rarely change

What kind of names ?

- Basically two kinds
  - Personal name - to uniquely identify a unit
    - e.g. Sensor unit 743
  - Group name - to identify a set of units
    - To map service-based addressing
    - e.g. Radioactive sensor
- Group names can be hierarchical
    - Like domain names in the Internet
    - Name trees and name graphs
- Applications aware of names, not addresses

Name-address translation

- Must be distributed
  - Scalability
- Some kind of distributed naming service
  - Should handle frequent joins and leaves
  - Fault tolerance, since a critical service
- Spread across a hierarchy
  - To permit inter-network addressing and routing

Routing do's and don'ts

- Must be distributed
- Care about routes to destinations only
- Avoid / eliminate stale routes
- Backup routes, is it a good idea ?
- Converge to optimal routes
- Should not maintain global state
  - Minimize non-local state
- Should not participate in frequent updates

Routing protocols

- Depends on traffic characteristics
- Generally two kinds of methods are used
  - Proactive
    - e.g. Link state routing
    - High overhead to maintain routes
  - Reactive / on-demand
    - Adds delays to packet forwarding
    - Partly avoidable by route caching for frequent paths

More on routing protocols

- What kind of metrics should be used ?
  - Number of hops need not be critical
  - Minimize the power consumption
- Hop-by-hop error checking
  - More computation at “routers”
- Medium is inherently broadcast
  - So can add multicast features more easily

Existing research

- IETF group - MANET ("nearly defunct now")
  - Mostly dealt with routing issues
  - Various protocols suggested
    - Ad-hoc On-demand Distance Vector algorithm (AODV)
    - Dynamic Source Routing (DSR)
    - Zone Routing Protocol (ZRP)
    - Core-Extraction Distributed Ad-hoc Routing (CEDAR)
  - Main aim - to interoperate with IP
    - Run TCP/UDP/IP stacks on the end nodes (laptops)

Example - AODV

- Ad-hoc On-demand Distance Vector algorithm [Perkins, Royer 98]
- On-demand route computation
- Broadcasts Route-Request on failure
- Route-Reply unicasted back by
  - Destination
  - Intermediate node with fresh enough info
- Triggered Route-Replies to other senders
- Defined to work on top of IP, ICMP etc.

Example - ZRP

- Zone Routing Protocol [Haas, Pearlman 98]
- Defines “zones” using proximity
- Combines proactive and reactive schemes
  - Proactive within a zone
  - Reactive between zones
- Zone discovery needed for inter-zone routes
  - Through a query - reply mechanism

Example - Others

- Dynamic Source Routing [Broch et al 98]
  - Similar to AODV except
    - DSR packets carry routing info
    - AODV nodes contain the routing info
  - Also defined on top of IP, ICMP etc.
- Core-Extraction Distributed Ad-hoc Routing [Sivakumar et al 98]
  - Maintains a "core" - more stable, high bandwidth
  - Link state propagated towards core
  - CEDAR considers QoS issues considered

Load balancing

- Main reason - conserve power
- Nodes should time-share router operations
  - Packet forwarding
  - Route computation
- However all nodes cannot act as routers
  - e.g. border nodes
- Need "hand-off" mechanism - transfer state
  - How to inform others of change ?
  - When to hand-off ?

Synchronization

- Can be a link layer mechanism
- Nodes cannot listen for messages at all times
  - To conserve power
  - Need synchronized message exchange
- Clocks should be “synchronized”
  - External means e.g. GPS, not always possible
- Some kind of Sync messages
  - Frequency depends on clock drift rates
  - Nodes exchange Syncs even at idle periods

More on synchronization

- Extent of synchronization may vary
  - Do not need clocks exactly matched
- In a slotted medium, enough to synchronize so that can distinguish adjacent slots
- Synchronization trades off with listening delays

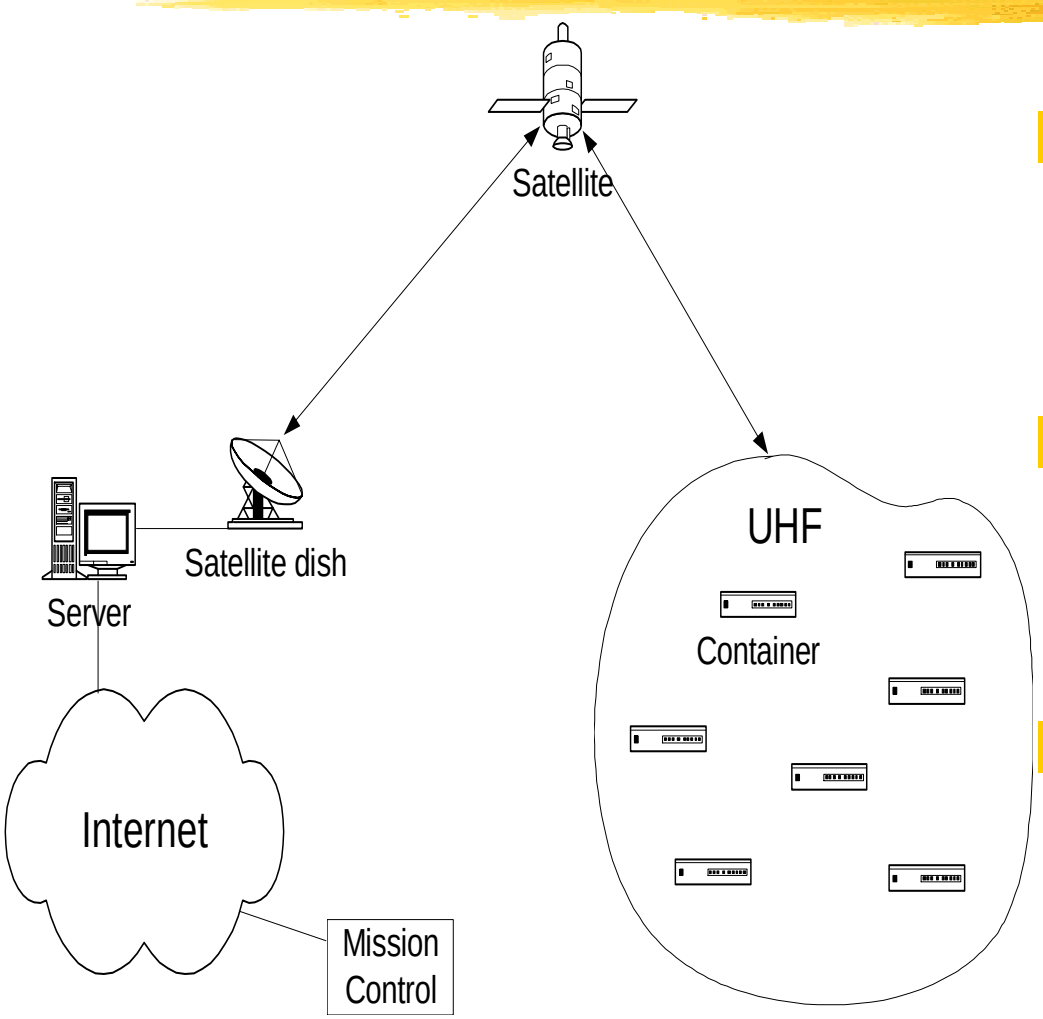
Initialization

- Invoked after powered on, or after battery is recharged after a power loss
- First job - detect network in vicinity
  - Synchronizing to that network is not easy
  - Join that network
- If cannot find neighbors, start own network
  - Comprise of only one node (itself)
  - Others, that arrive later, can join it

Merge and split networks

- Two networks that come close spatially
  - Messages of one interfere with other
  - Merging the two networks is a solution
    - Can be more efficient
    - Might be constrained with the "carrying-capacity" of a network - efficiency might fall above a threshold
- Some nodes move away
  - Reverse of above - split the network
- Or, some combination of above

Sample Architecture



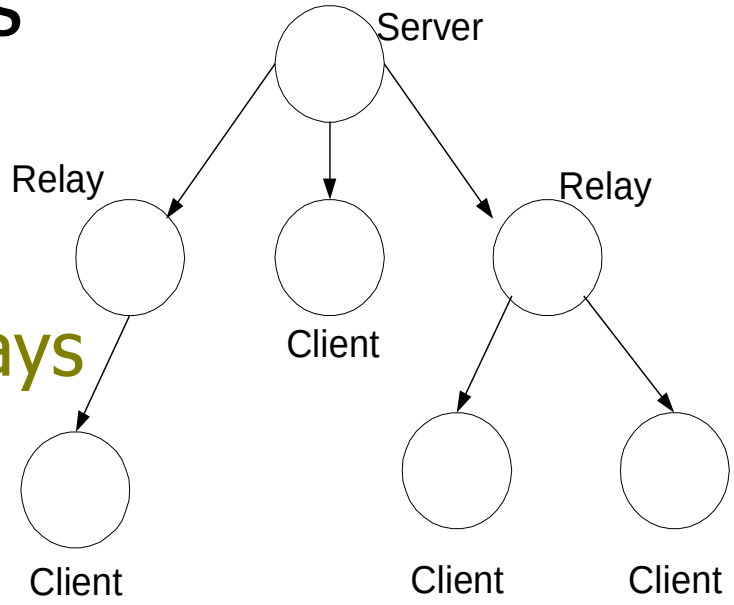
- Container shipment problem
- Describe protocols for containers only
- Communication is via UHF

Sample architecture (cont'd)

- Physical architecture
  - Nodes contain processor, memory, interface for UHF and satellite
    - No caches, no virtual memory / paging system
  - Interfaces are dumb
    - Processor needed to match message destination
      - This can be done on-the-fly by the processor
  - UHF ~ 16 Kbps, Satellite ~ 20 bps !

Sample architecture (cont'd)

- Nodes arrange into trees
- Three kinds of nodes
  - Root node - server
  - Intermediate nodes - relays
  - Leaf nodes - clients
- Master-slave operations
  - Parents arbitrate control of media for children
    - Allotment may be static or dynamic
    - Guarantees one node in charge of media at a time



Sample architecture (cont'd)

- Server node initiates a duty cycle
- Duty cycle comprises of multiple slots
- Slots are pre-allotted for -
  - Sync and control info transfer
  - New members to join
  - Data messages from parent to children
  - Data messages from children to parent
  - Relays to perform their own duty cycles

Related architecture - I

- Architecture proposed for sensor networks
  [Rockwell Science Center and UCLA]
- Have developed prototypes - AWAIRS I
- Basic concepts similar
  - TDMA-like slotted duty cycles
- Define topology learning scheme
  - Uses pre-assigned slots and check what one can hear

Related architecture - II

- Bluetooth architecture
  - www.bluetooth.com
  - Founded by Ericsson, Toshiba, Intel, IBM, Nokia
- "Ad-hoc, …, unconscious, automatic connections between devices"
  - e.g. handheld PCs, mobile phones, etc.
- Carry data and voice at upto ~ 720 Kbps
- Mostly patented

Final questions

- How to extend this to certain applications
  - Home networking
  - Ubiquitous Internet connectivity
- What are good platforms for development
  - Active networking ?
- Is RTX-CTX a good approach

Summary

- Self-organizing networks are vastly different from existing networks
- Need specialized technology for its specific demands

- Many application opportunities exist
- Others would come up as this technology becomes more viable