

Introduction to Sensor Networks

Sébastien Tixeul
tixeul@lri.fr

Outline

- Motivation
- Architecture
- Overview

Motivation

Sensor Networks

- **Definition:** Network of wireless nodes dedicated to a particular application
- **Purpose:** Acquire sensed data and transmit to a processing station
- **Application domains:** Military, Civilian, etc.

Motivation

- **Acquire data and feed a processing station**
- **Application domains:**
 - *Military:* risky area monitoring, intrusion detection, etc.
 - *Civilian:* fire detection, chemical facilities monitoring, etc.

Sensor vs. *ad hoc*

Sensors	<i>ad hoc</i>
Specific	Generic
Collaboration	Selfishness
Many-to-one	Any-to-any
No ID	ID
Energy	Throughput

Issues

- *ad hoc* deployment
- Unattended operation
- Untethered
- Dynamic changes

Issues

- *ad hoc* deployment
- Unattended operation
- Untethered
- Dynamic changes

No
infrastructure

Issues

- *ad hoc* deployment
- Unattended operation
- Untethered
- Dynamic changes

No human
intervention

Issues

- *ad hoc* deployment
- Unattended operation
- Untethered
- Dynamic changes

No
permanent power
supply

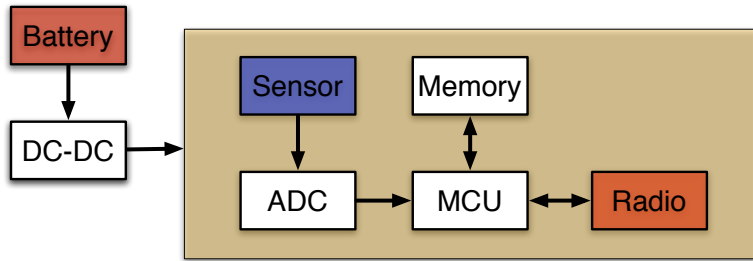
Issues

- *ad hoc* deployment
- Unattended operation
- Untethered
- Dynamic changes

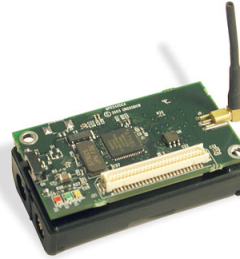
Changing
environment

Architecture

Sensor Node Architecture



Available Devices



- MicaZ (Crossbow)
- 2.94 GHz IEEE 802.15.4 Zigbee radio
- 128 KB program memory
- 512 KB data memory
- 8 mA draw

Available Devices



- Tmote Sky/invent (Moteiv)
- 2.94 GHz IEEE 802.15.4 Zigbee radio
- 8MHz processor
- 10 KB RAM
- 48 KB Flash

Available Devices



- Stargate (Crossbow)
- Wired Ethernet
- Wifi/Cellular via PCMCIA
- INTEL PXA 255
- Linux Kernel

Sensors

- **Exterioceptors:** information about the surroundings
- **Proprioceptors:** information about the internal workings

Sensors

	Measurand	Transduction
Physical	Pressure	Piezoresistive, capacitive
	Temperature	Thermistor, thermomechanical, thermocouple
	Humidity	Resistive, capacitive
	Flow	Pressure change, thermistor

Sensors

	Measurand	Transduction
Motion	Position	E-mag, GPS, contact
	Velocity	Doppler, Hall effect, optoelectronic
	Angular velocity	Optical encoder
	Acceleration	Piezoresistive, piezoelectric, optical fiber

Sensors

	Measurand	Transduction
Contact	Strain	Piezoresistive
	Force	Piezoelectric, piezoresistive
	Torque	Piezoresistive, optoelectronic
	Vibration	Piezoresistive, piezoelectric, optical fiber, sound, ultrasound

Sensors

	Measurand	Transduction
Presence	Tactile	Contact switch, capacitive
	Proximity	Hall effect, capacitive, magnetic, seismic, acoustic, RF
	Distance	E-mag (sonar, radar, lidar), magnetic, tunnelling
	Motion	E-mag, IR, acoustic, seismic

Sensors

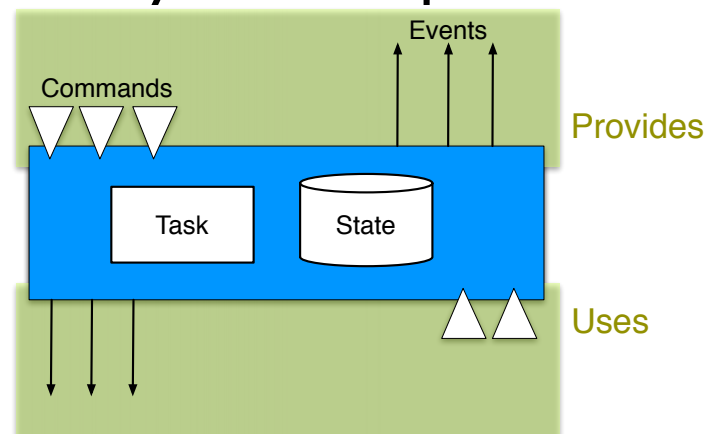
	Measurand	Transduction
Biochemical	Agents	biochemical transduction
Identification	Personal features	Vision
	Personal ID	Fingerprints, retinal scan, voice, heat plume, vision, motion analysis

Sensor Node Operating System

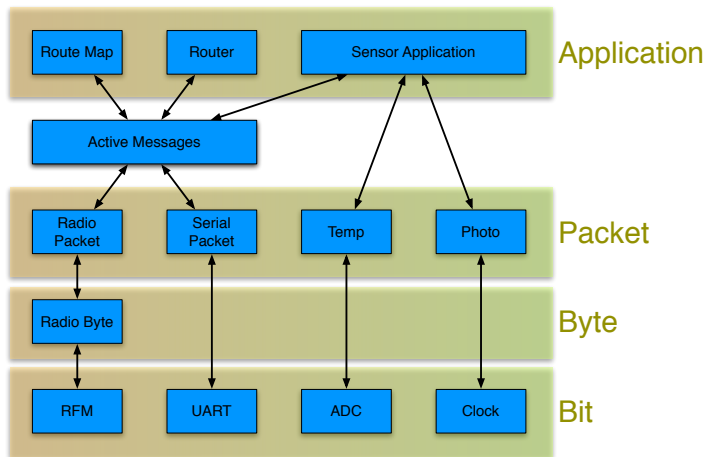
• TinyOS concepts

- Scheduler + Graph of components
- Component
- Constrained storage model
- Very Lean multithreading
- Efficient Layering

TinyOS Component



TinyOS Application



Programming TinyOS

- **TinyOS is written in NesC**
 - Applications are written as system components
- **Syntax for concurrency and storage model**
- **Compositional support**
 - Separation of definition and linkage

Simulating TinyOS

- **Target platform: TOSSIM**
 - Native instruction set
 - Event driven execution mapped to event driven simulator
 - Storage model mapped to virtual nodes
 - Radio and environmental models

Overview

Issues and Solutions

- Localization
- Routing
- Medium Access Control
- Applications

Localization

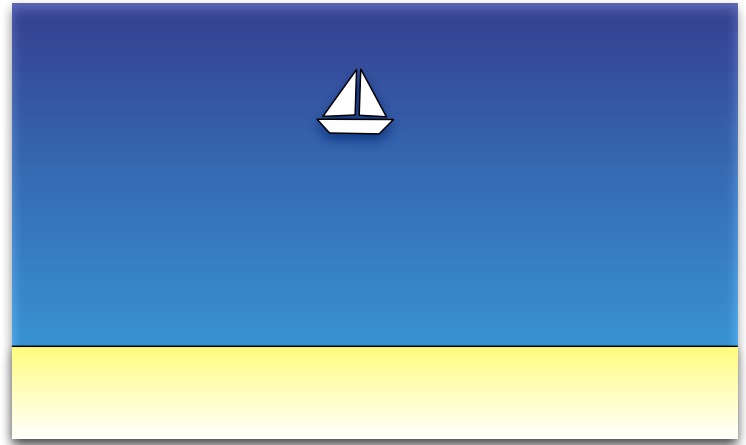
Localization

- **Fine-grained**

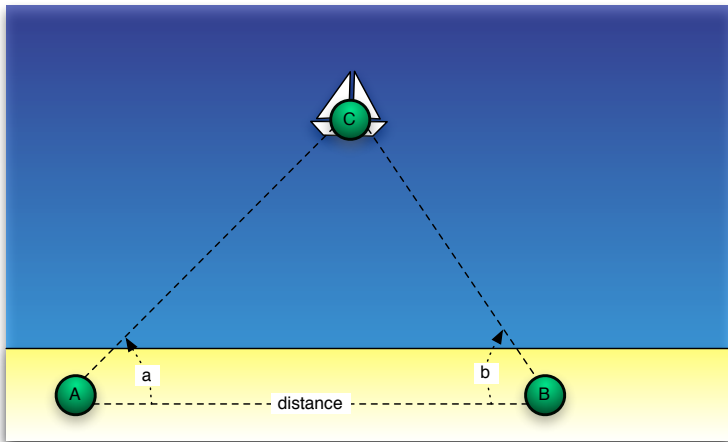
- Timing
- Signal strength
- Signal pattern matching
- Directionality

- **Coarse-grained**

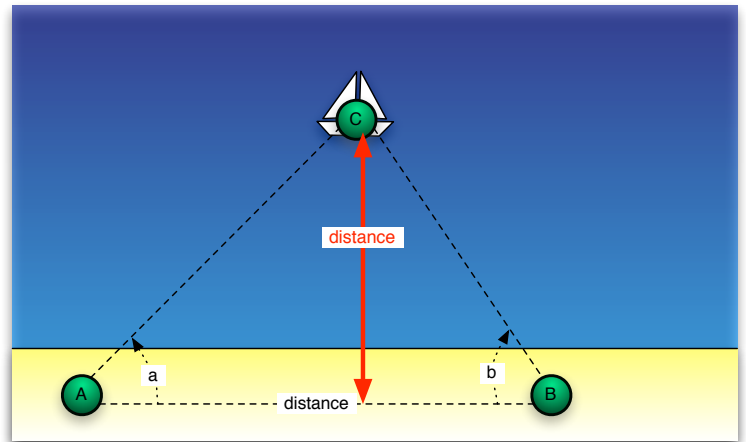
Triangulation



Triangulation



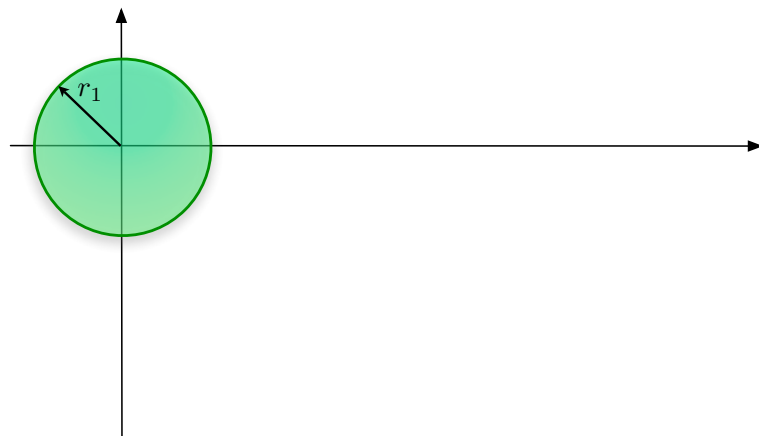
Triangulation



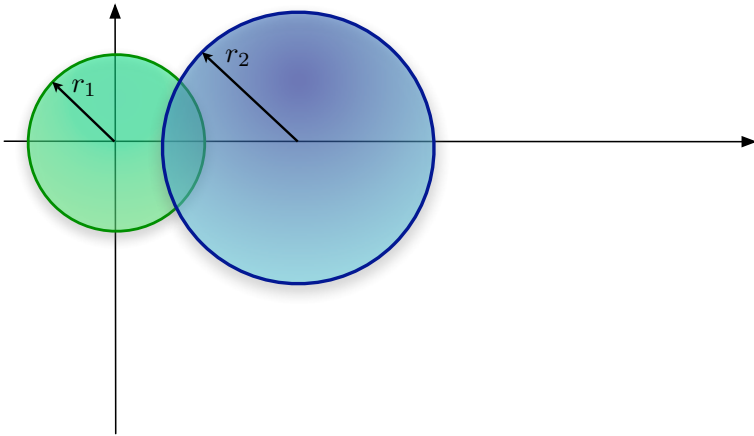
Trilateration



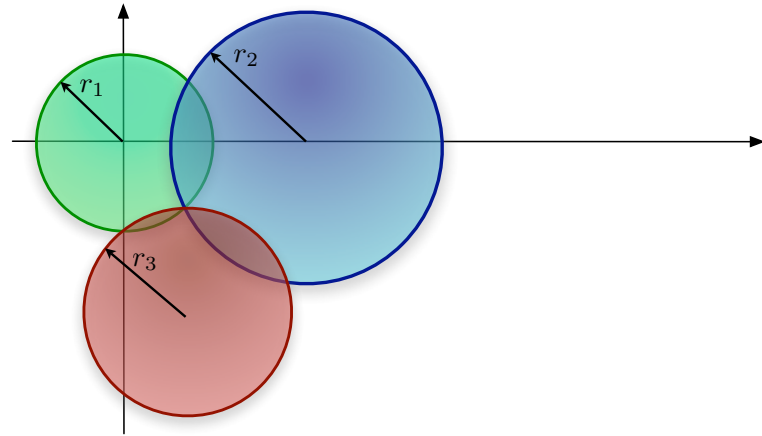
Trilateration



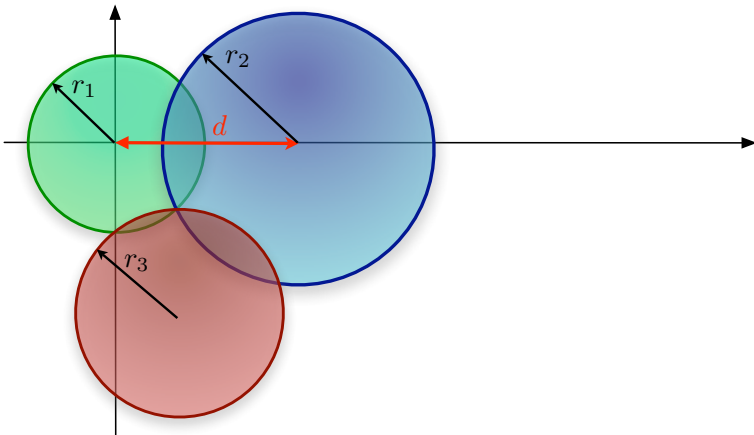
Trilateration



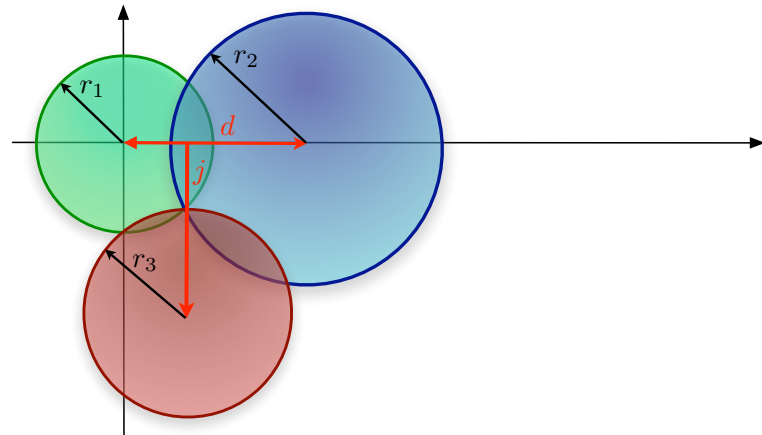
Trilateration



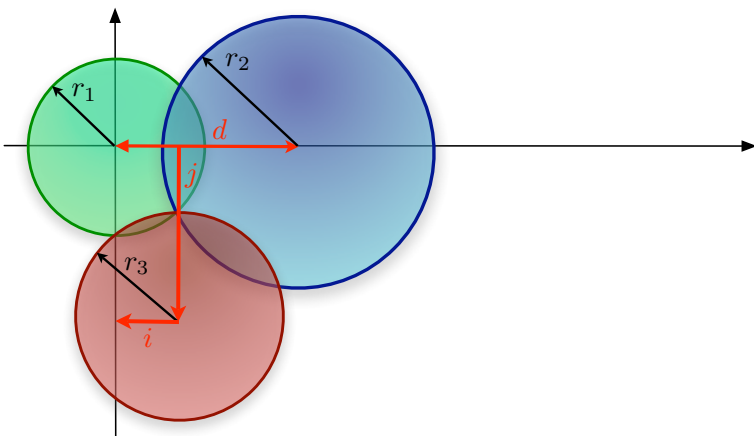
Trilateration



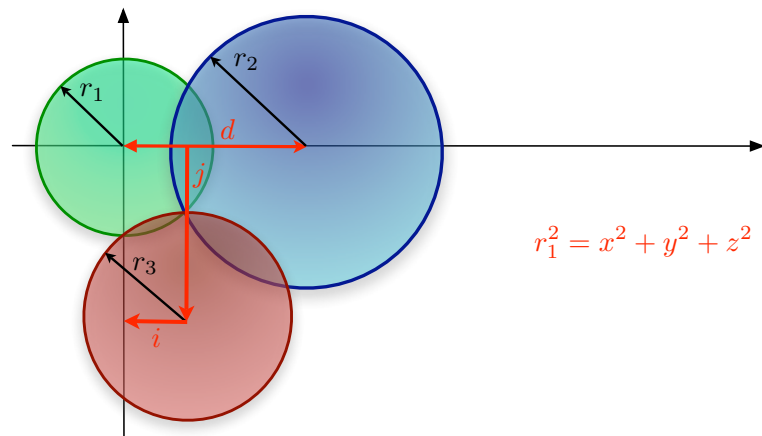
Trilateration



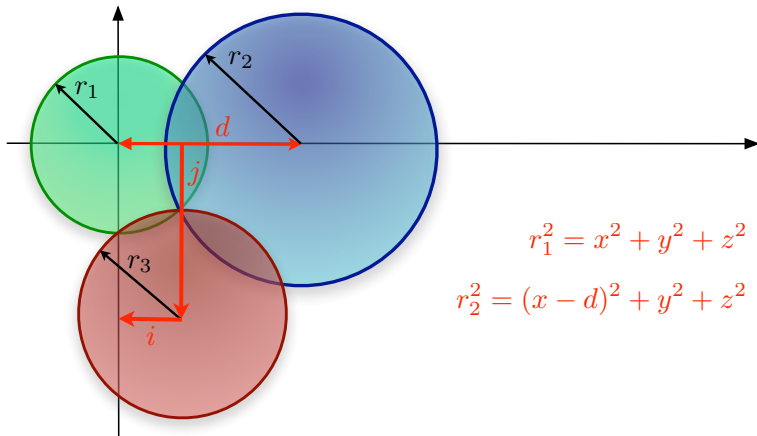
Trilateration



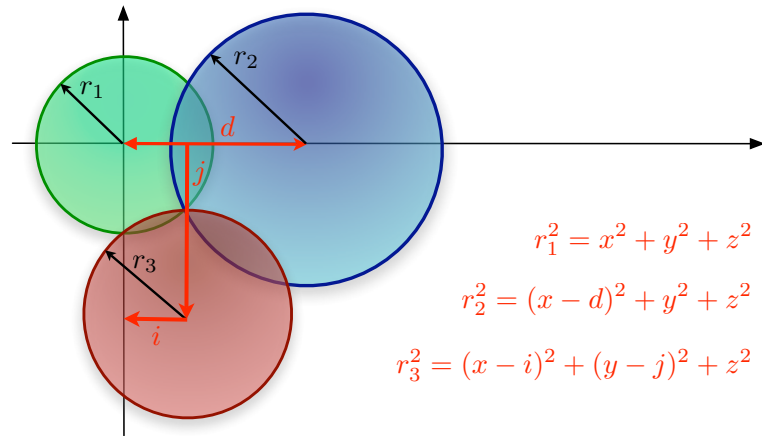
Trilateration



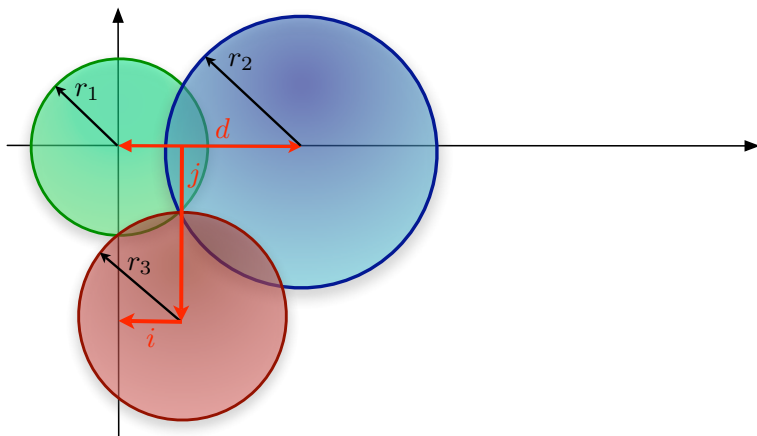
Trilateration



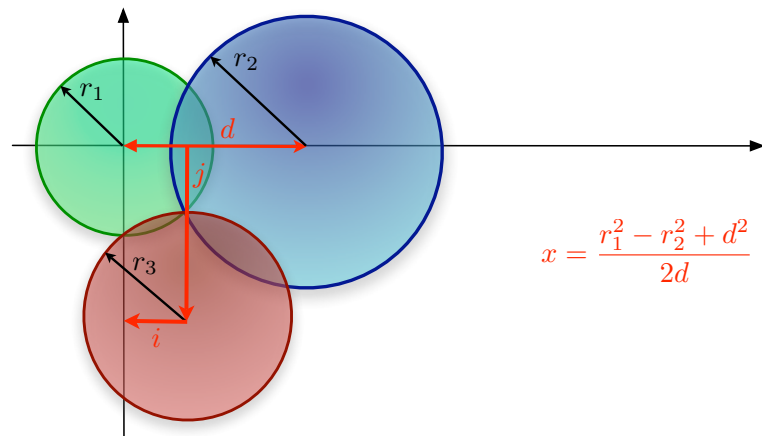
Trilateration



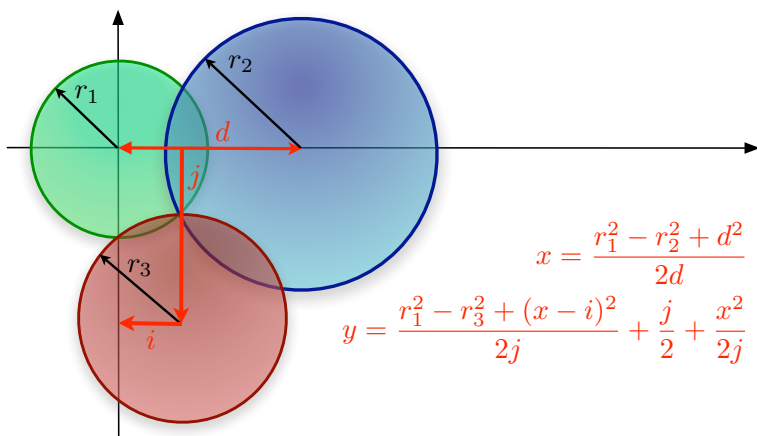
Trilateration



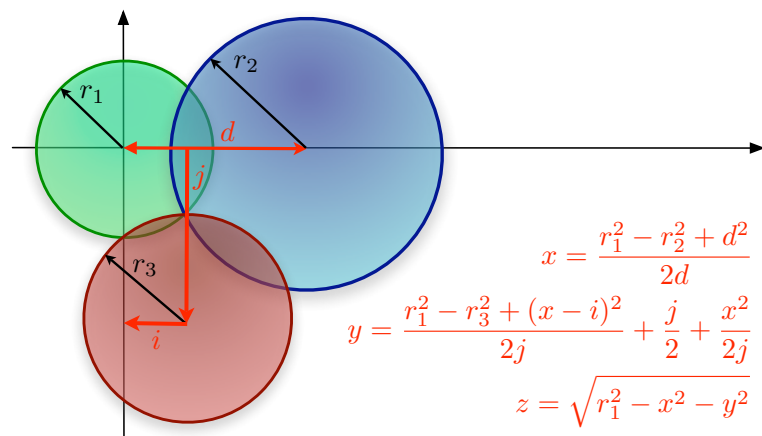
Trilateration



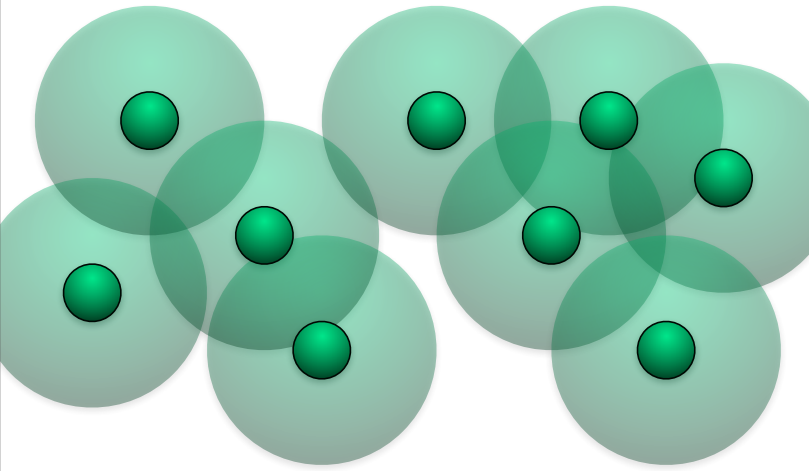
Trilateration



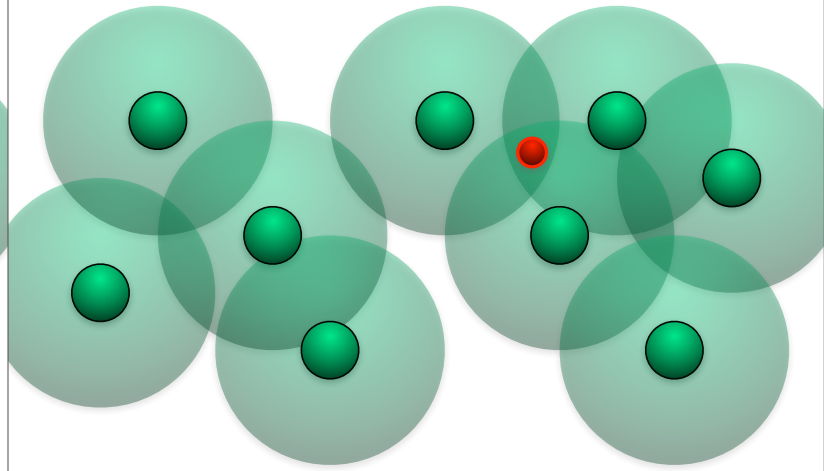
Trilateration



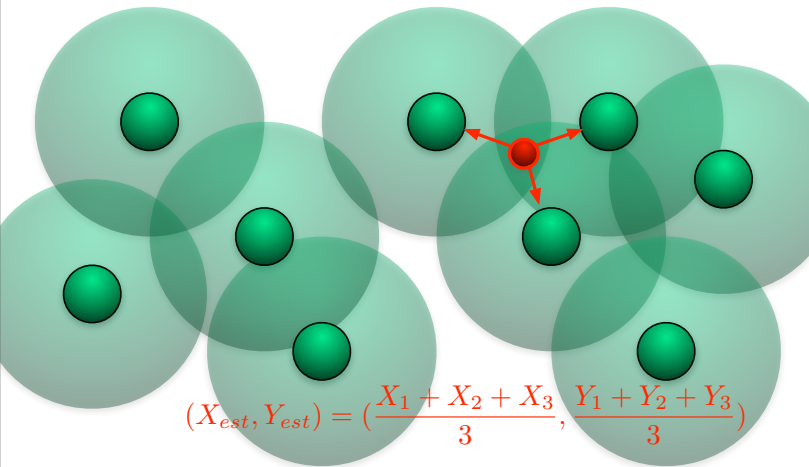
Centroid



Centroid



Centroid



Routing

Routing

- **Classical flooding**
 - *Implosion*
 - *Resource management*
- **Negotiation based protocols**
 - *SPIN*
 - *Directed Diffusion*

Negotiation Based Protocols

- **SPIN:** Sensor Protocols for Information via Negotiation
 - Information descriptors for negotiation prior to data transmission
 - Negotiation relates to available energy

SPIN

- **ADV**: new data is available and described
- **REQ**: Request to receive data
- **DATA**: actual data

SPIN-PP

S

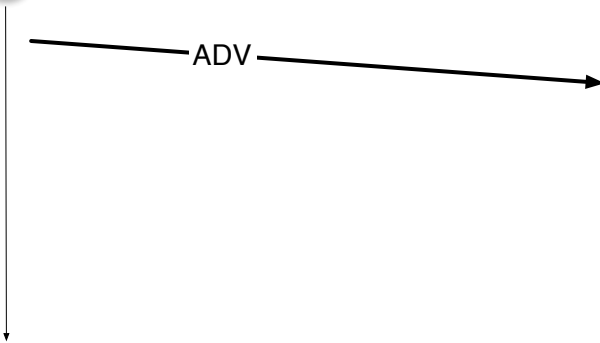
R



SPIN-PP

S

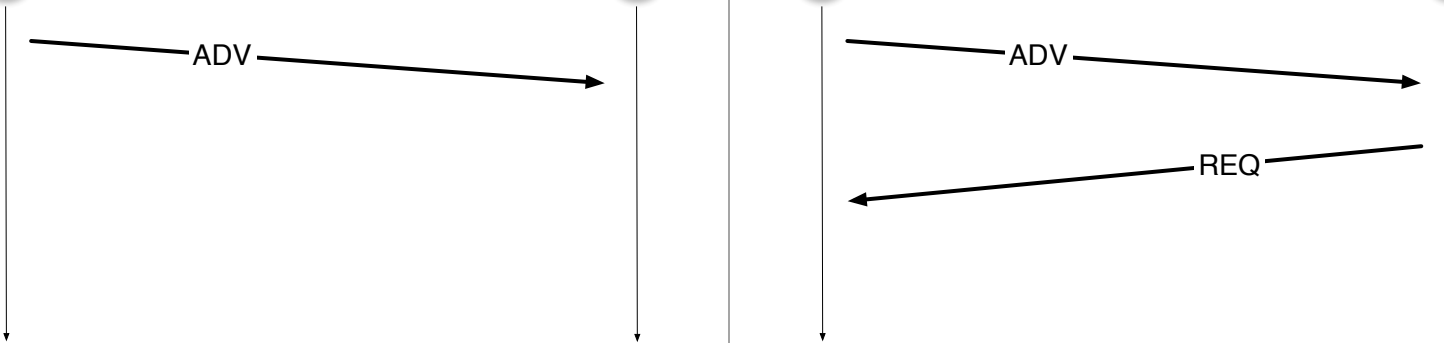
R



SPIN-PP

S

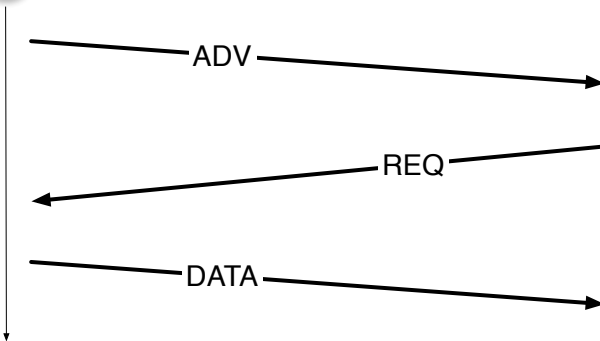
R



SPIN-PP

S

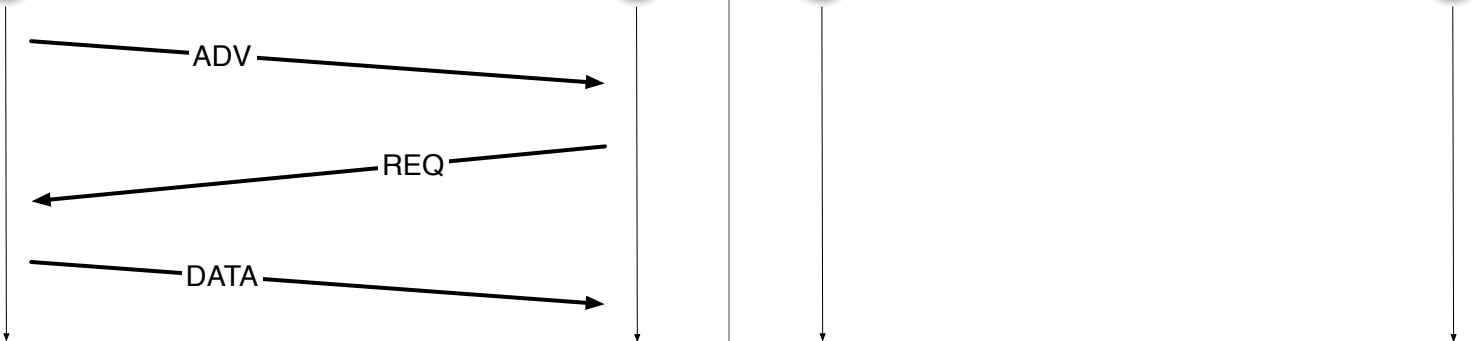
R



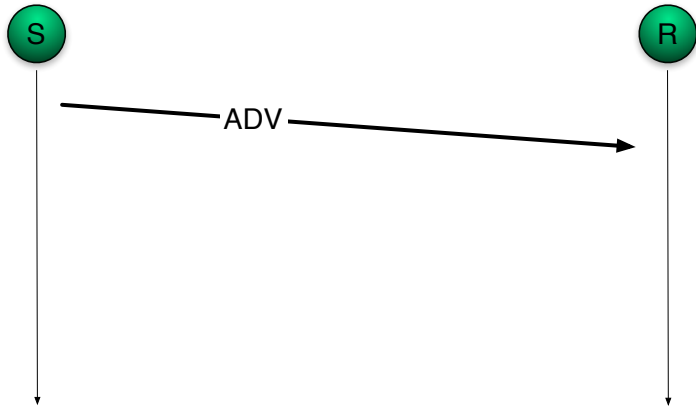
SPIN-EC

S

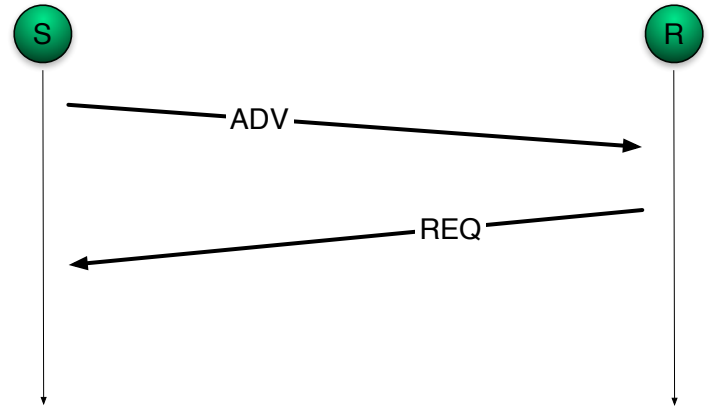
R



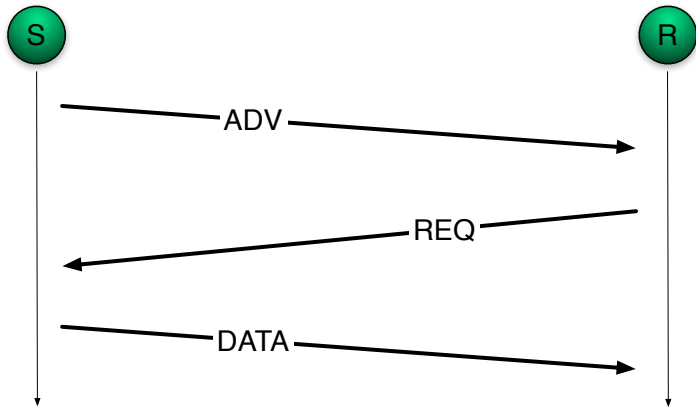
SPIN-EC



SPIN-EC



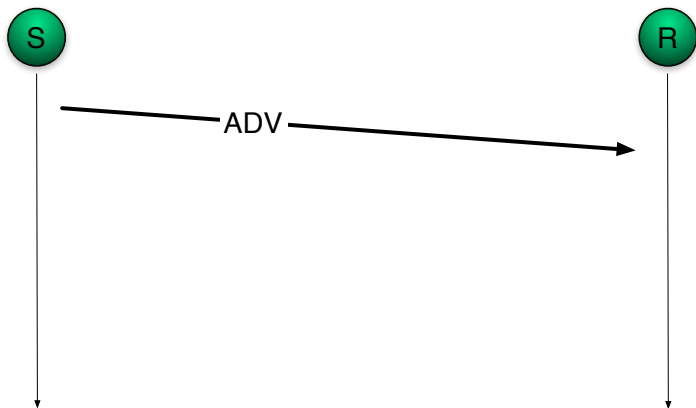
SPIN-EC



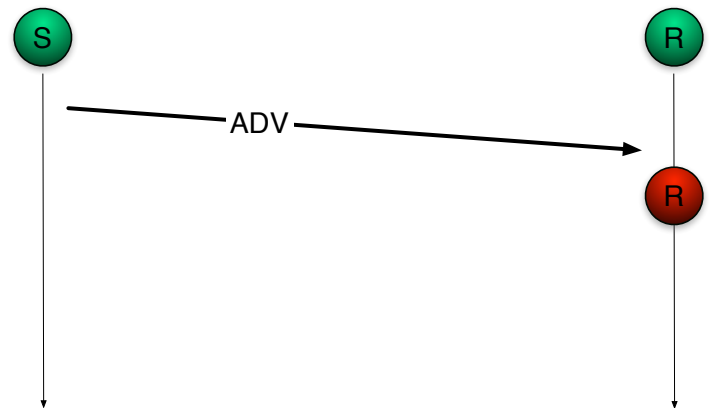
SPIN-EC



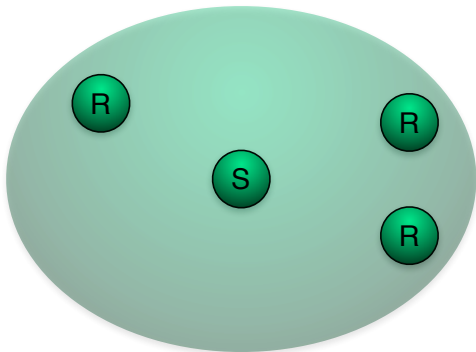
SPIN-EC



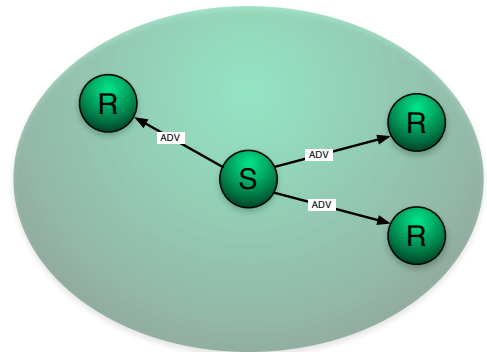
SPIN-EC



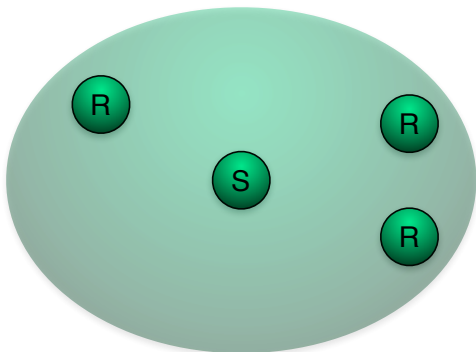
SPIN-BC



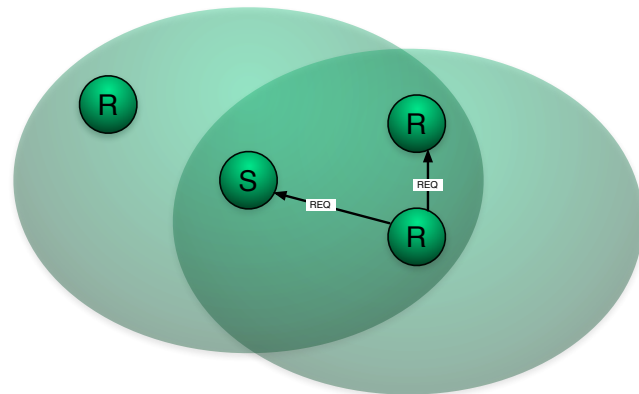
SPIN-BC



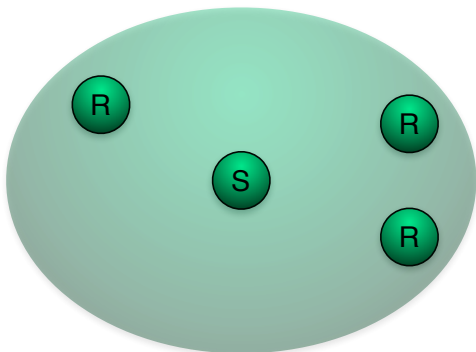
SPIN-BC



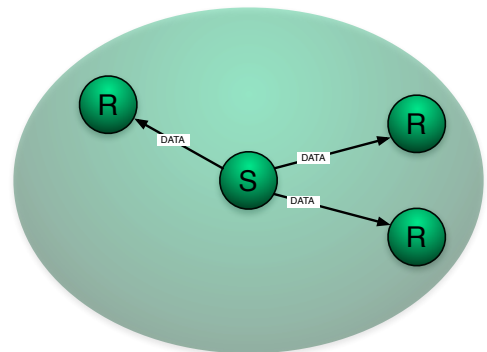
SPIN-BC



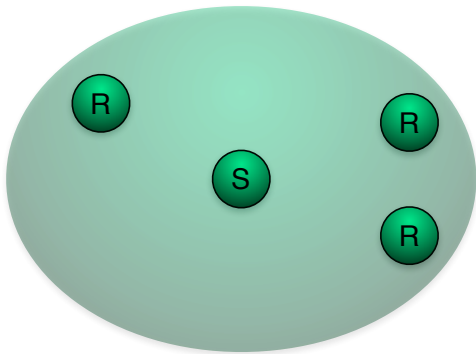
SPIN-BC



SPIN-BC



SPIN-BC



Directed Diffusion

- Destination-initiated (sink) reactive routing technique
- Tasks are described by attribute-value pairs (*interests*)
- All nodes maintain interest cache for each requested interest

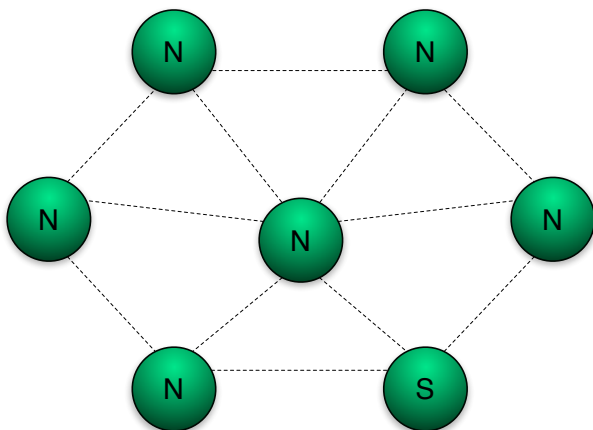
Interests

item name	value
type	four-legged animal
interval	20 ms
duration	10 s
rect	[-100, 100, 200, 400]

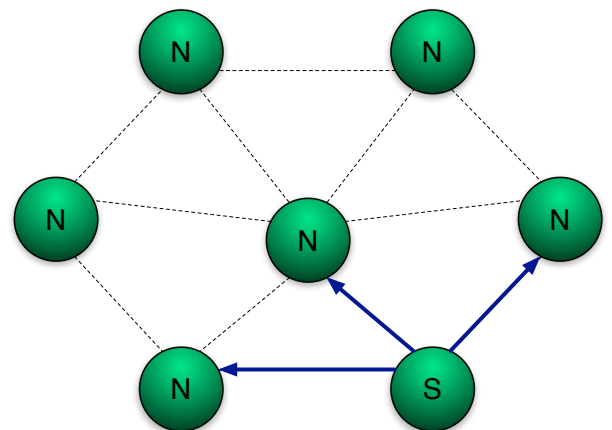
Returned Data

item name	value
type	four-legged animal
instance	[125, 220]
intensity	0.6
confidence	0.85
timestamp	01:20:40

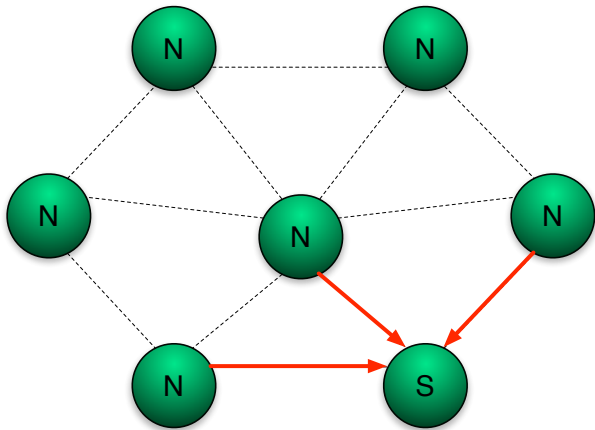
Directed Diffusion



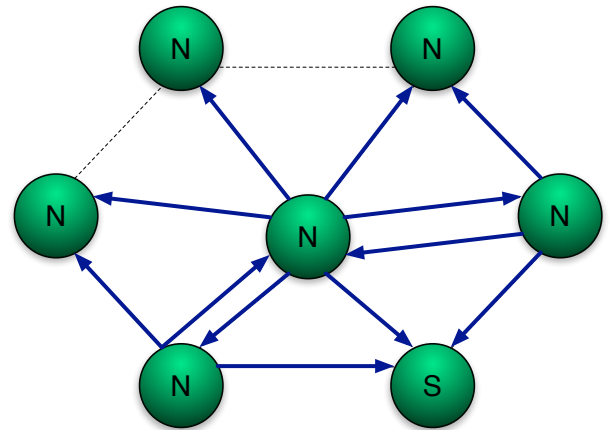
Directed Diffusion



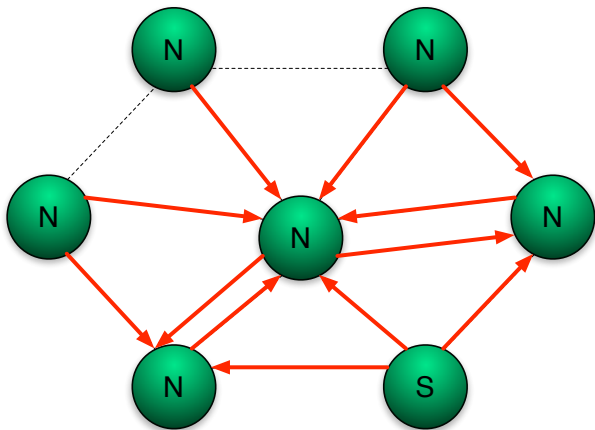
Directed Diffusion



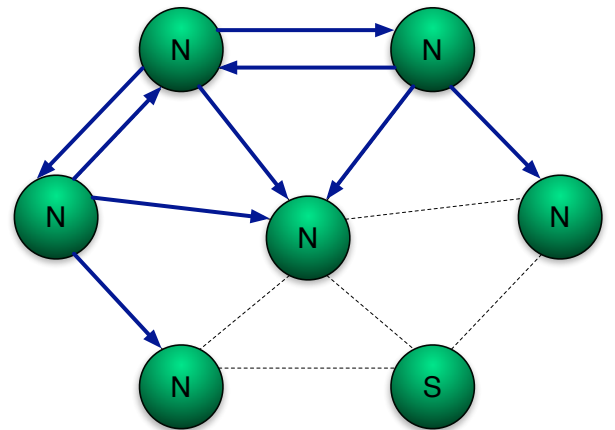
Directed Diffusion



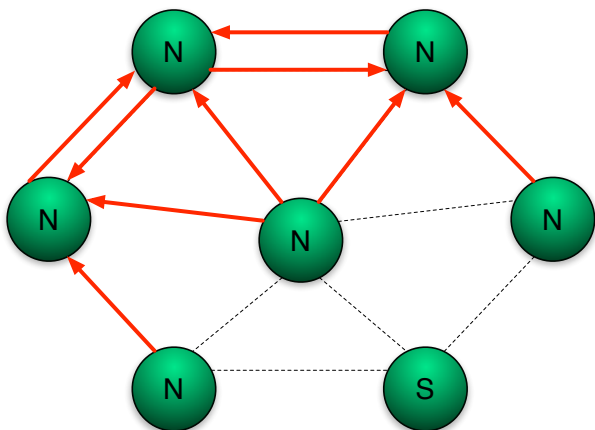
Directed Diffusion



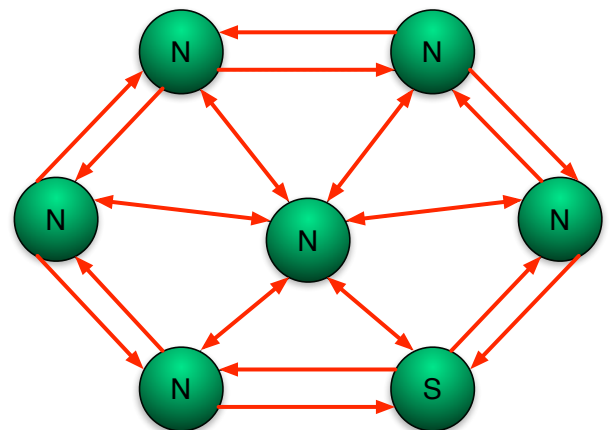
Directed Diffusion



Directed Diffusion



Directed Diffusion



Interest Cache

- Periodically purged
- No information about sink
- Gradient table
 - *rate per neighbor*
 - *timestamp*
 - *expiration*

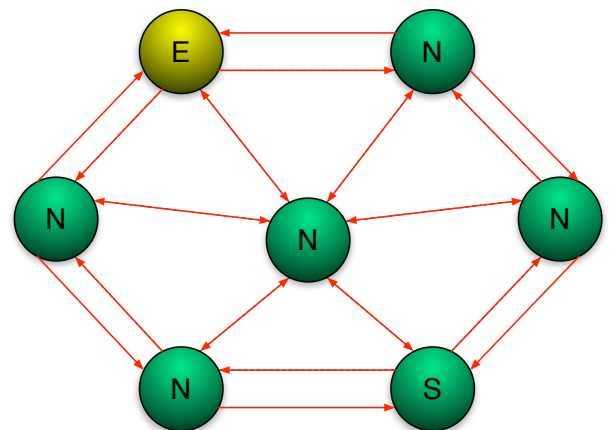
Interest Forwarding

- When new received, add to cache
- Simplest policy: rebroadcast interest
- No way of distinguishing new interests from repeated ones
- Set up (very low rate) gradients between all neighbors
- Must distinguish between neighbors

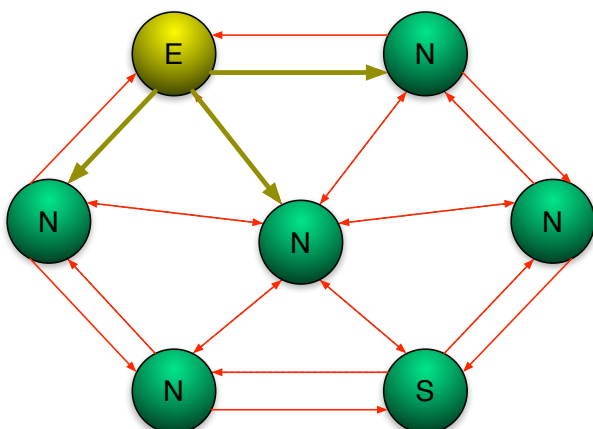
Message propagation

- A node matching an interest generates replies at desired rate
- When receiving a reply, lookup interest cache
- Forward along given route(s) if found, drop otherwise
- Loop prevention

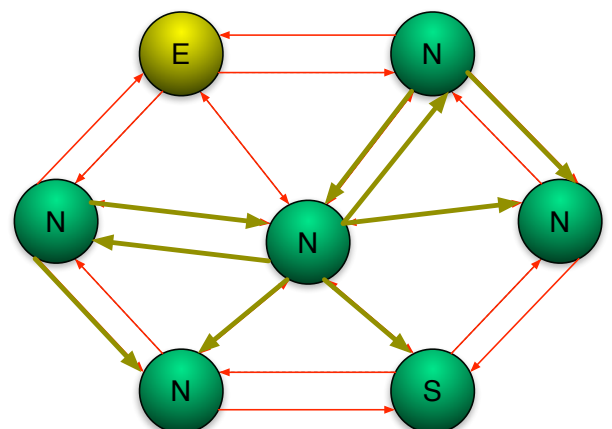
Directed Diffusion



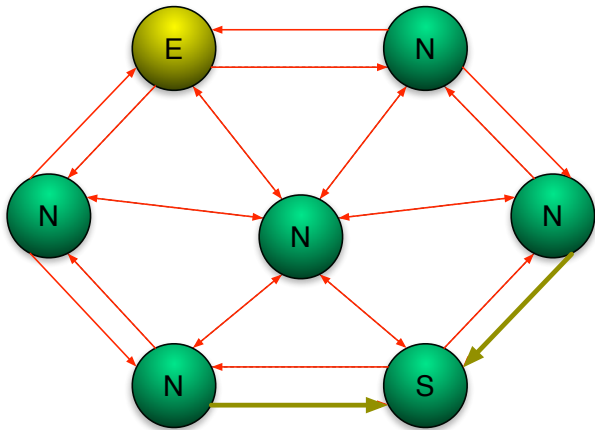
Directed Diffusion



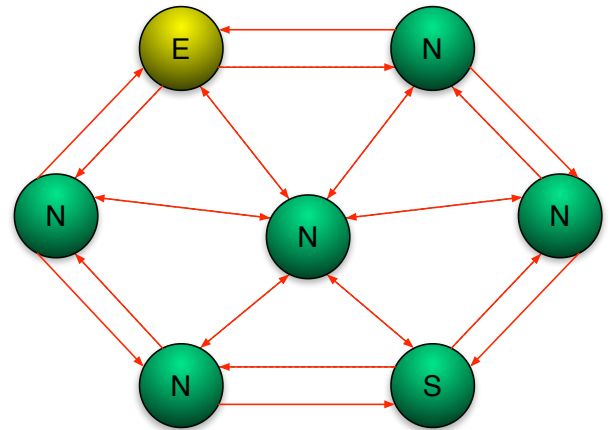
Directed Diffusion



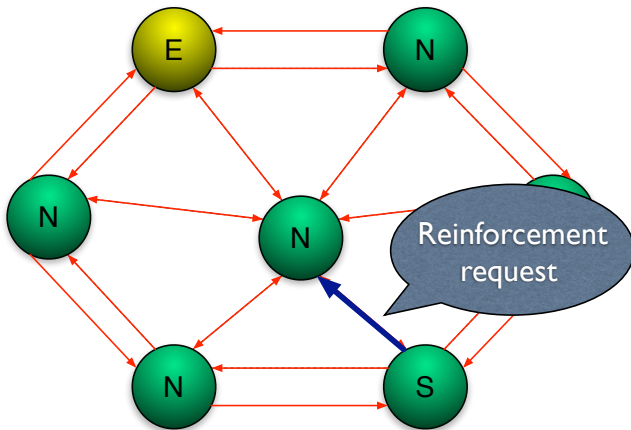
Directed Diffusion



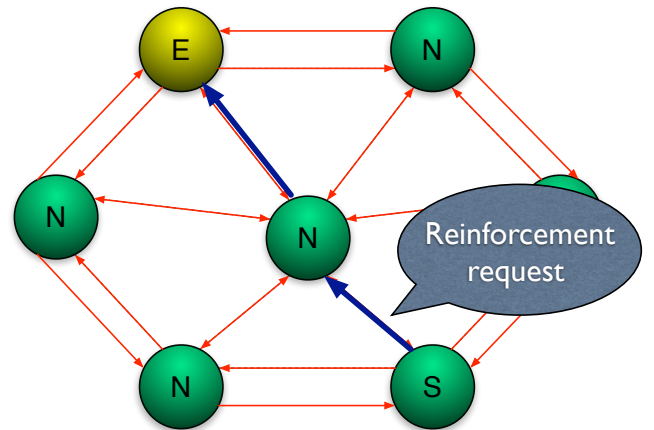
Directed Diffusion



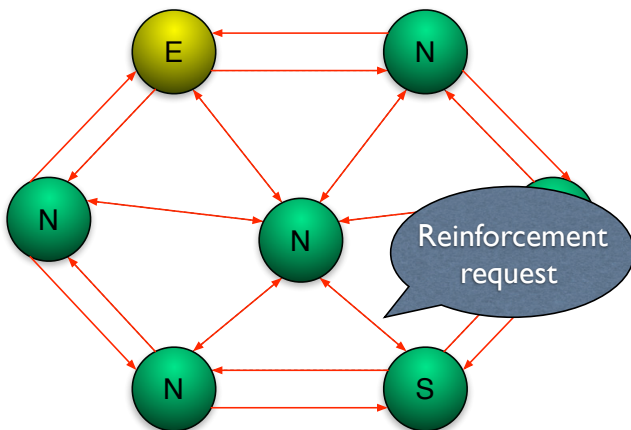
Directed Diffusion



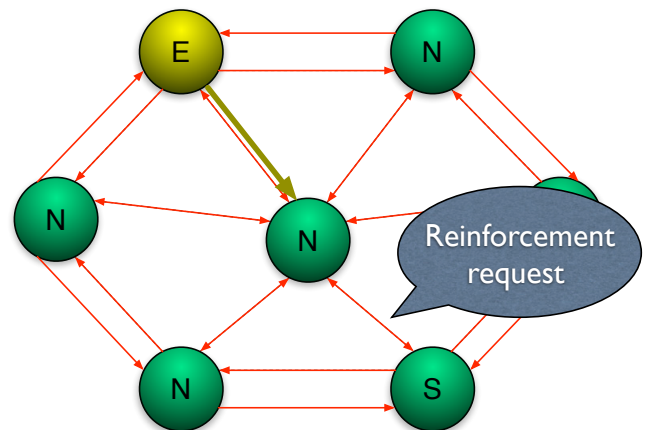
Directed Diffusion



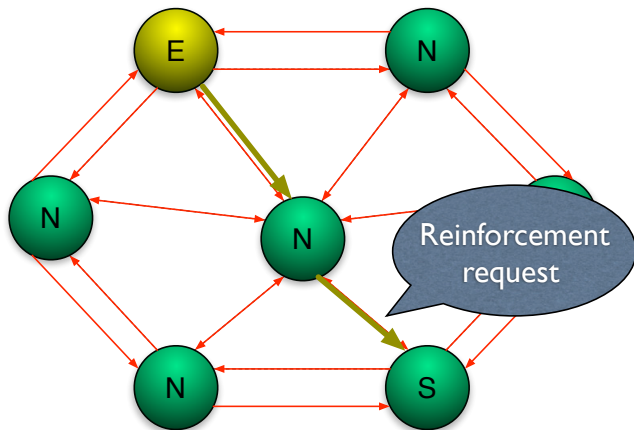
Directed Diffusion



Directed Diffusion



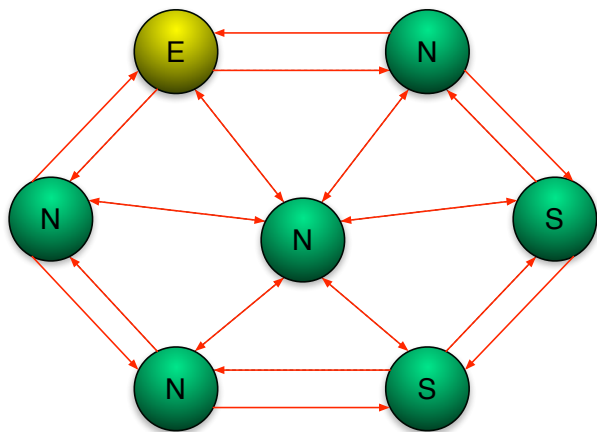
Directed Diffusion



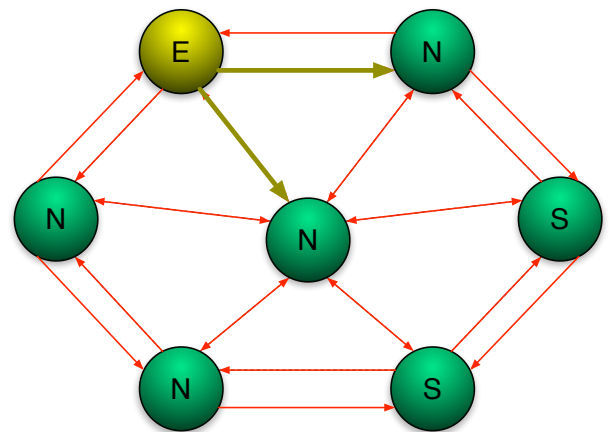
Reinforcement

- Sink can reissue the same request with a higher rate
- “Draw down” higher quality data from a particular neighbor
- Other nodes react when receiving
- “Outflow” increased, must reinforce another node to increase “inflow”

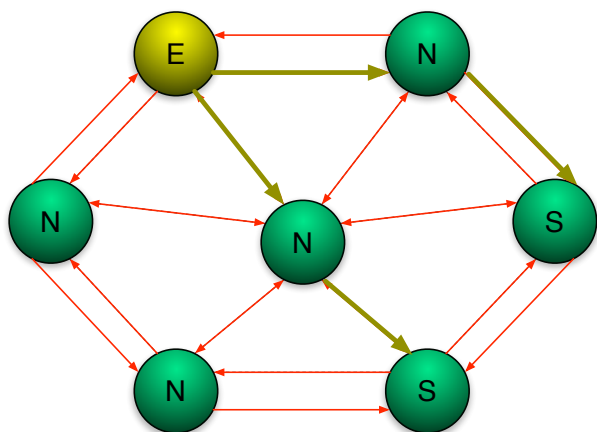
Directed Diffusion



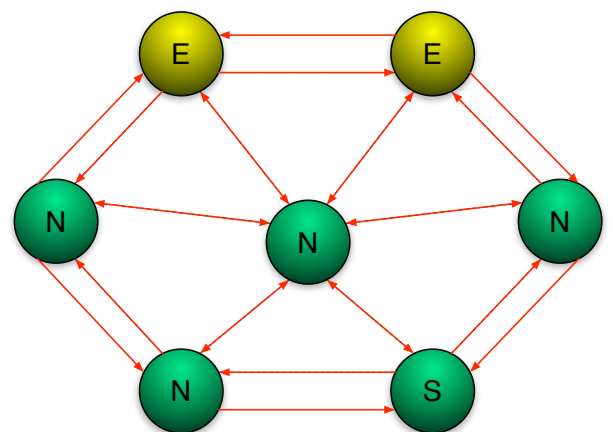
Directed Diffusion



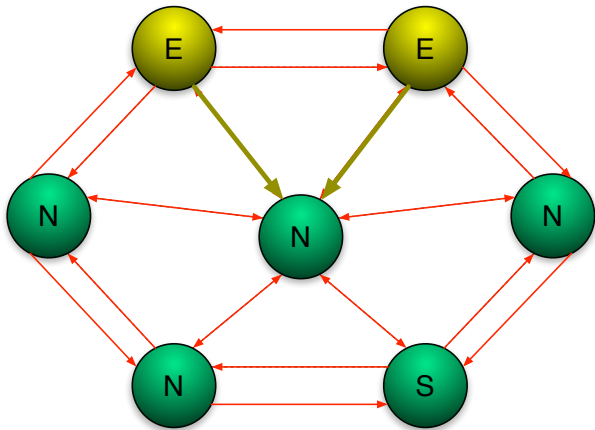
Directed Diffusion



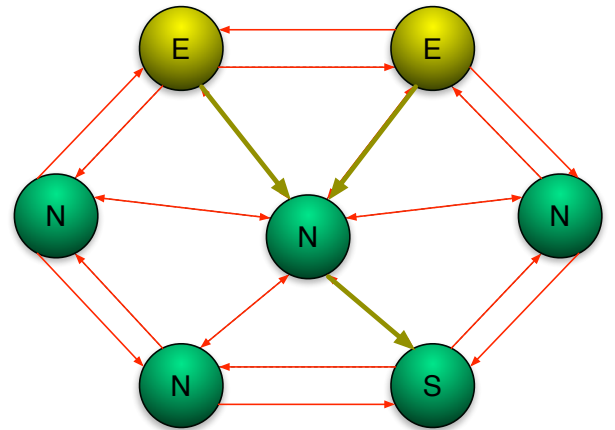
Directed Diffusion



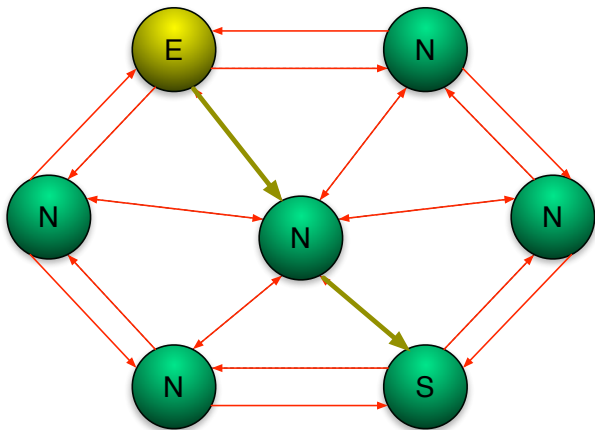
Directed Diffusion



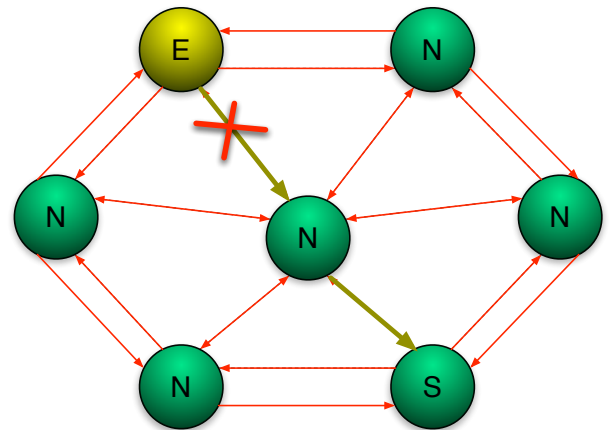
Directed Diffusion



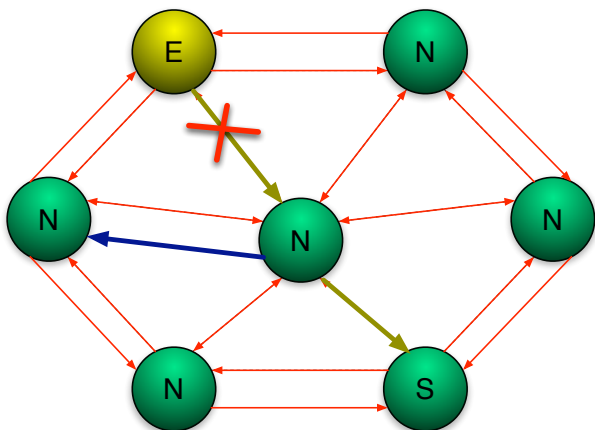
Directed Diffusion



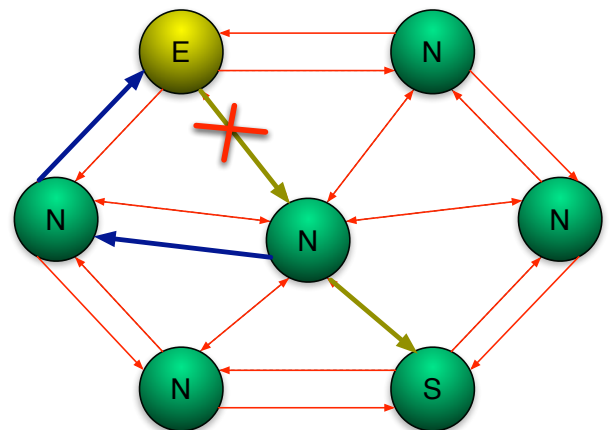
Directed Diffusion



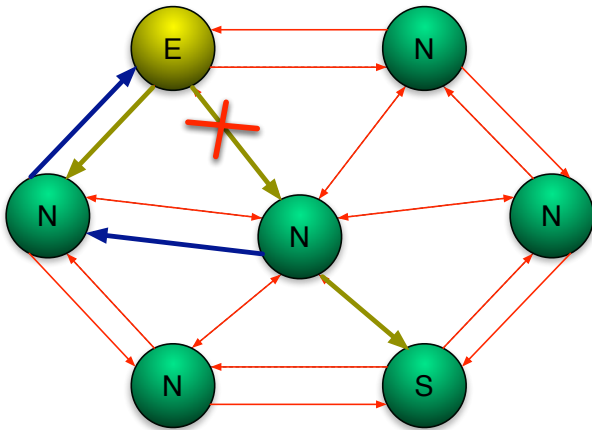
Directed Diffusion



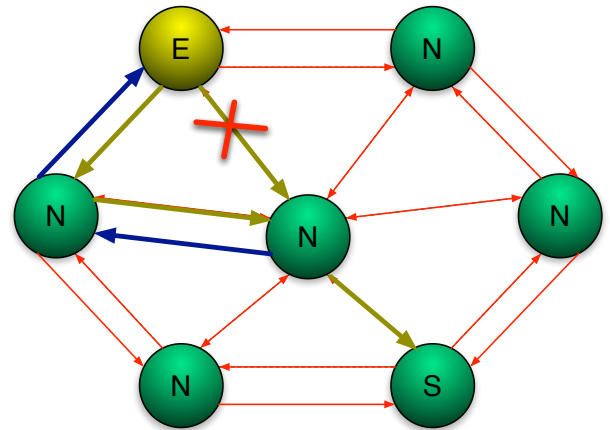
Directed Diffusion



Directed Diffusion



Directed Diffusion



Directed Diffusion

- **Local algorithm policies**

- *Propagating interests*
 - flood, cache information, GPS
- *Setting up gradients*
 - first heard neighbor, highest energy neighbor

Directed Diffusion

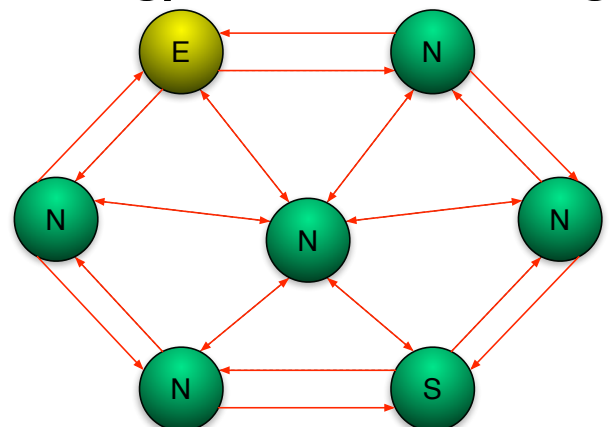
- **Local algorithm policies**

- *Data transmission*
 - single path, striped multi-path, multiple sources, etc.
- *Reinforcement*
 - observer losses, resources levels, etc.

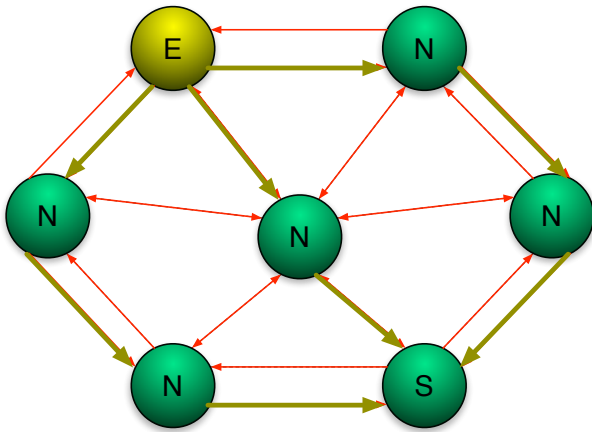
Energy Aware Routing

- Similar to Directed Diffusion
 - destination initiated
 - initial flooding to discover routes
 - several sub-optimal paths can be used (with a probabilistic distribution)

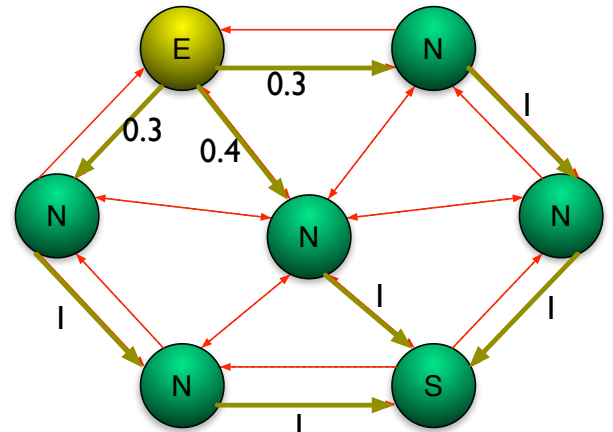
Energy Aware Routing



Energy Aware Routing



Energy Aware Routing



Medium Access Control

Classical Approaches

- **FDMA**: Frequency division multiple access
- **TDMA**: Time division multiple access
- **CDMA**: Code division multiple access
- **CSMA**: Carrier sense multiple access
 - **CD**: Collision detection
 - **CA**: Collision avoidance

Classical Approaches

One frequency available

- **FDMA**: Frequency division multiple access
- **TDMA**: Time division multiple access
- **CDMA**: Code division multiple access
- **CSMA**: Carrier sense multiple access
 - **CD**: Collision detection
 - **CA**: Collision avoidance

Classical Approaches

One frequency available

- **FDMA**: Frequency division multiple access
- **TDMA**: Time division multiple access
- **CDMA**: Code division multiple access
- **CSMA**: Carrier sense multiple access
 - **CD**: Collision detection
 - **CA**: Collision avoidance

One code available

Classical Approaches

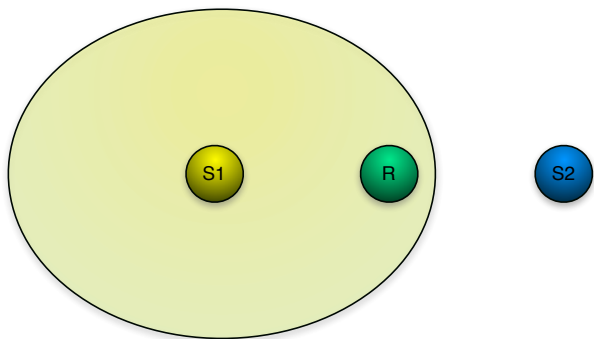
One frequency available

- **FDMA:** Frequency division multiple access
- **TDMA:** One code available
- **CDMA:** Code division multiple access
- **CSMA:** Can't listen while transmitting
 - **CD:** Collision detection
 - **CA:** Collision avoidance

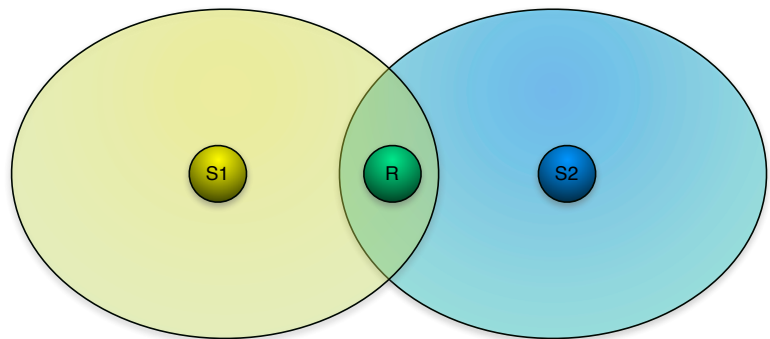
Hidden Terminal problem



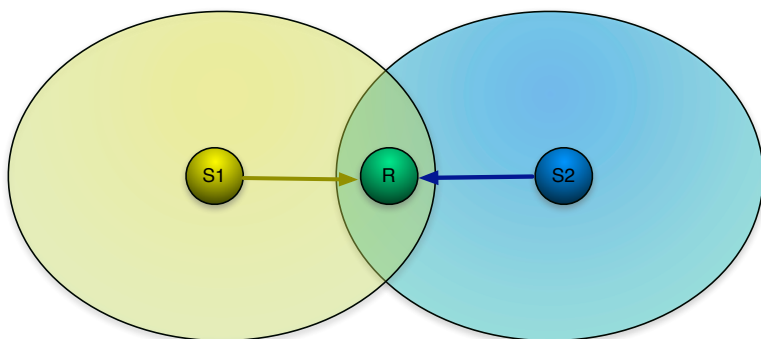
Hidden Terminal problem



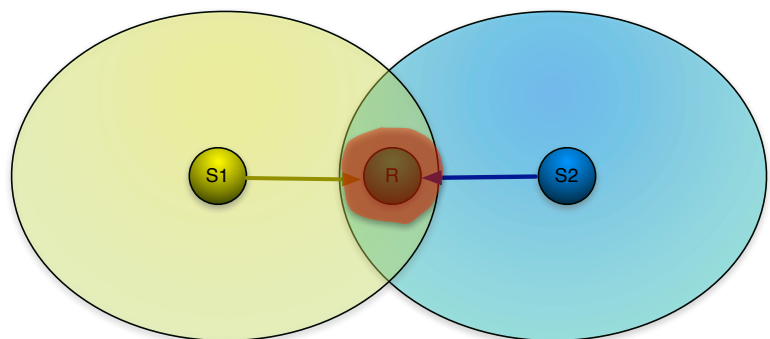
Hidden Terminal problem



Hidden Terminal problem



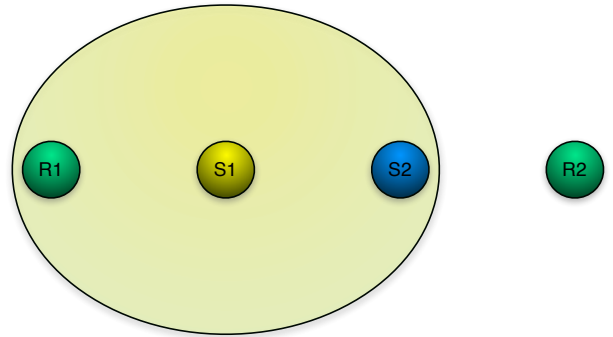
Hidden Terminal problem



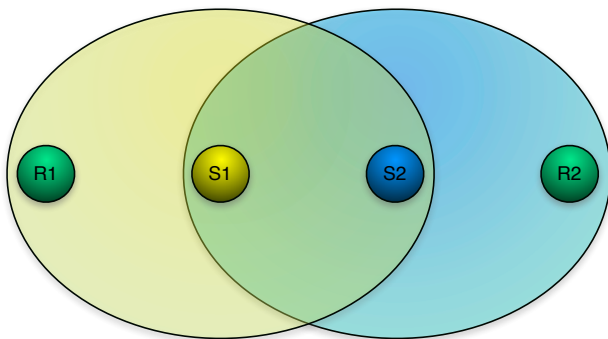
Exposed Terminal Problem



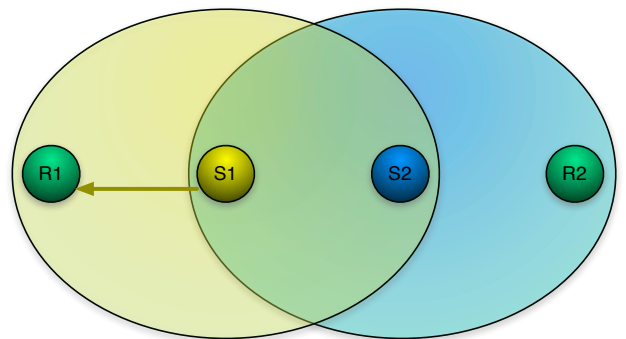
Exposed Terminal Problem



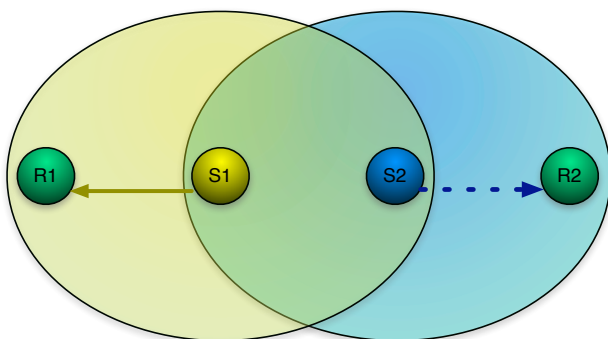
Exposed Terminal Problem



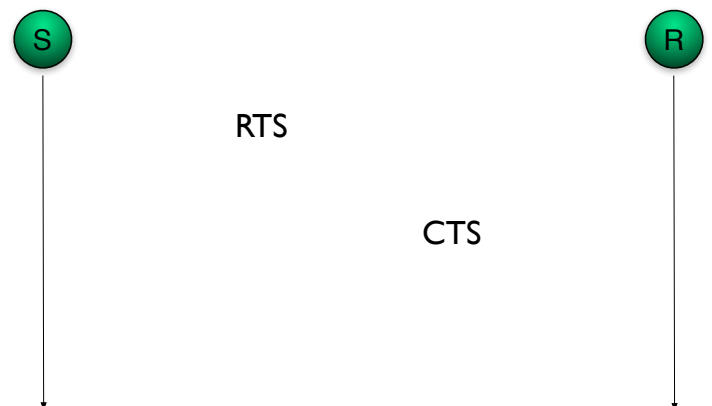
Exposed Terminal Problem



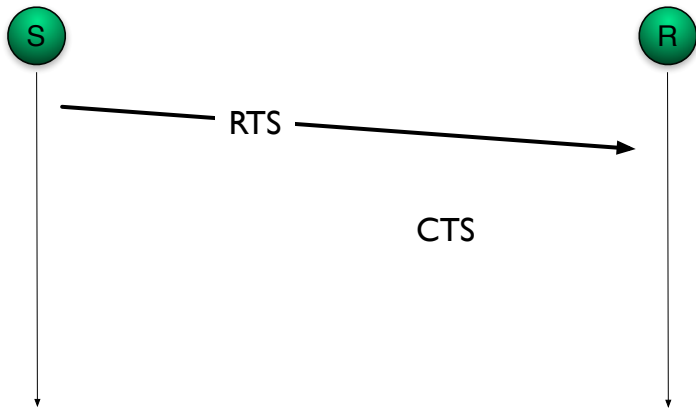
Exposed Terminal Problem



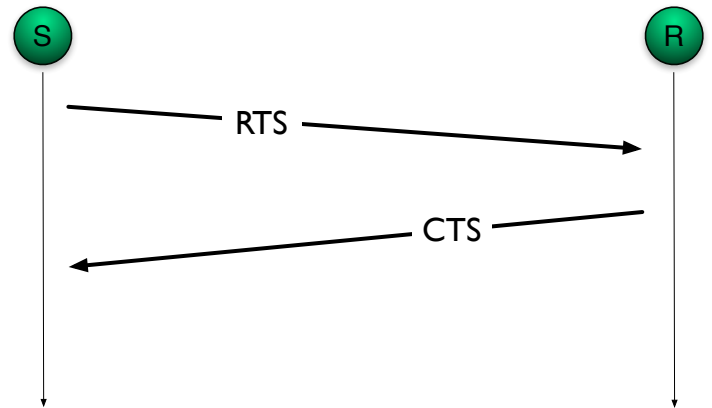
IEEE 802.11 RTS/CTS



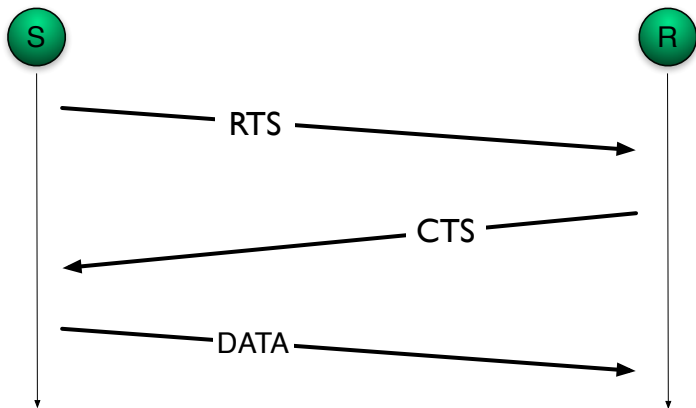
IEEE 802.11 RTS/CTS



IEEE 802.11 RTS/CTS



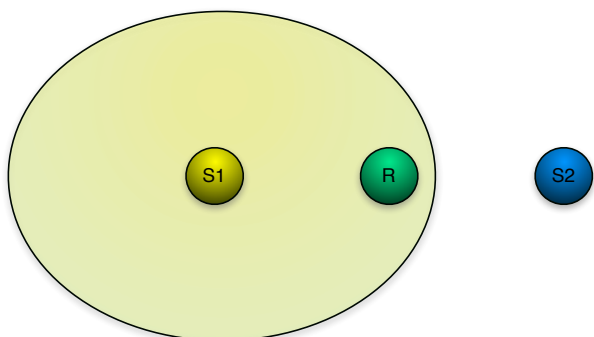
IEEE 802.11 RTS/CTS



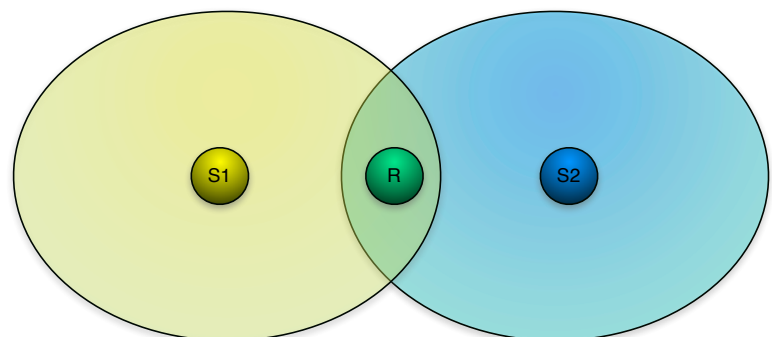
IEEE 802.11 RTS/CTS



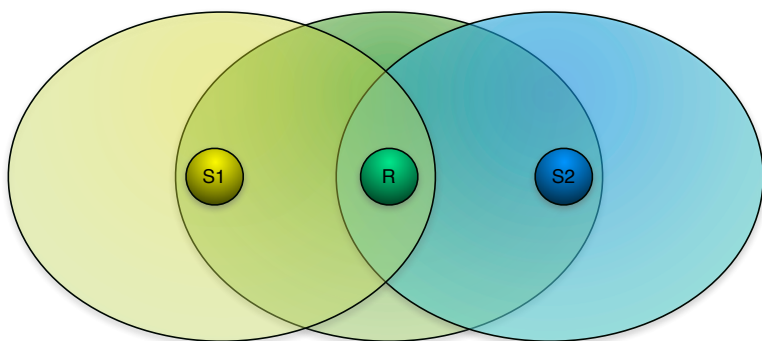
IEEE 802.11 RTS/CTS



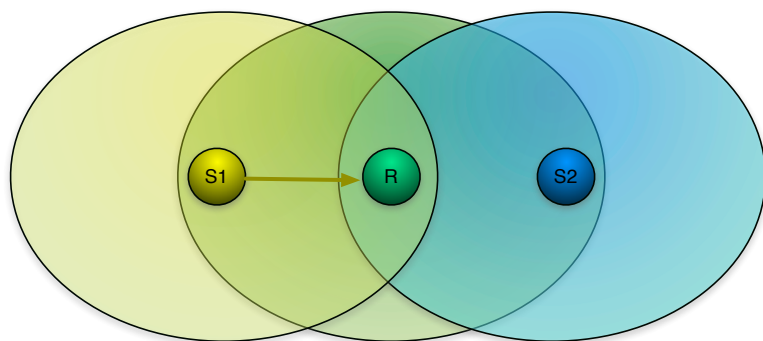
IEEE 802.11 RTS/CTS



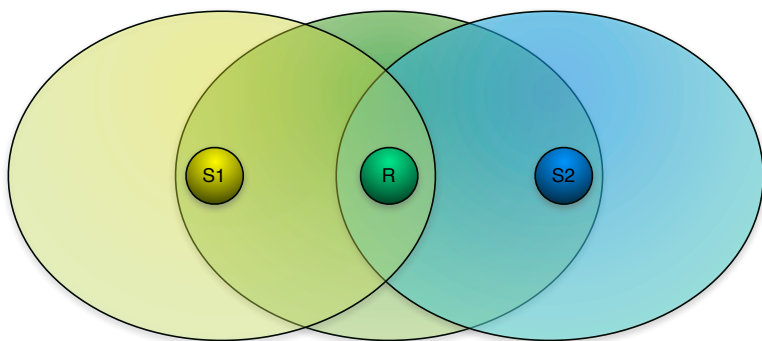
IEEE 802.11 RTS/CTS



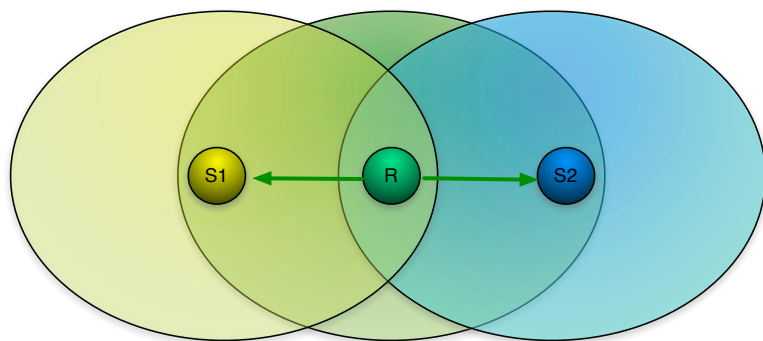
IEEE 802.11 RTS/CTS



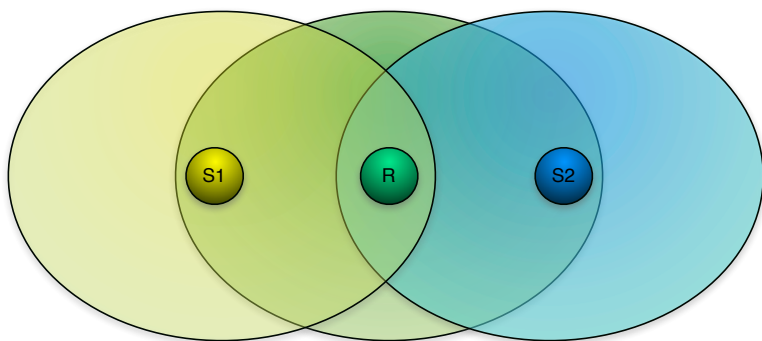
IEEE 802.11 RTS/CTS



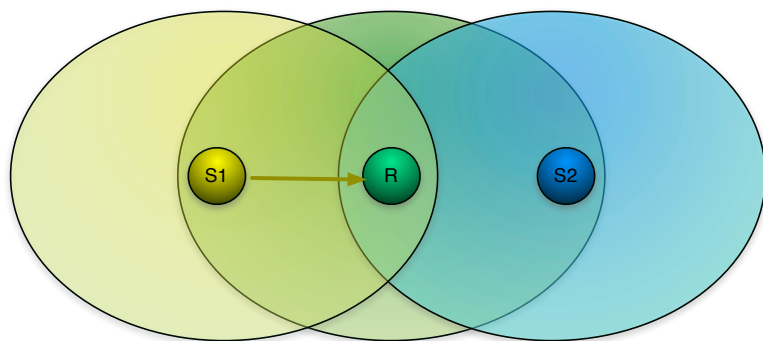
IEEE 802.11 RTS/CTS



IEEE 802.11 RTS/CTS



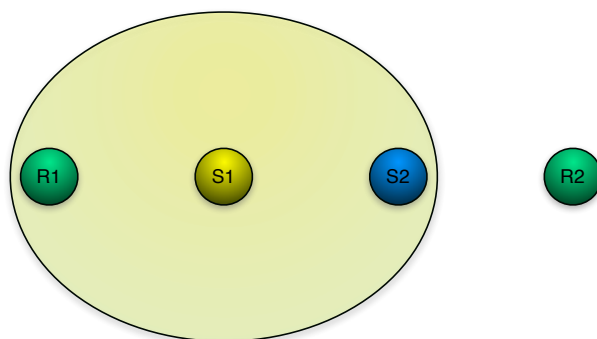
IEEE 802.11 RTS/CTS



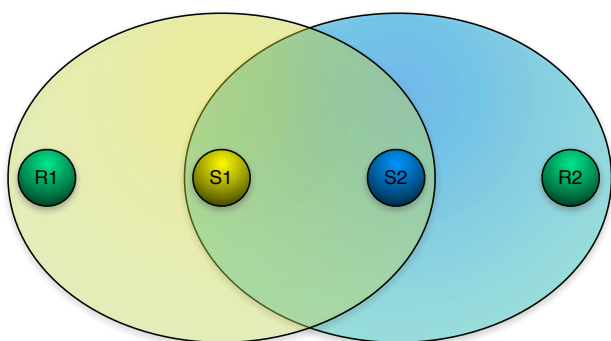
IEEE 802.11 RTS/CTS



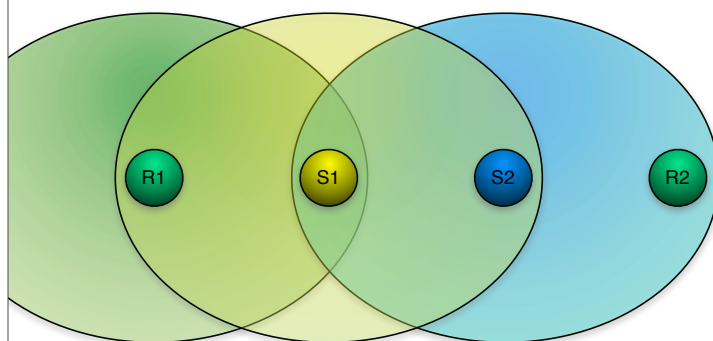
IEEE 802.11 RTS/CTS



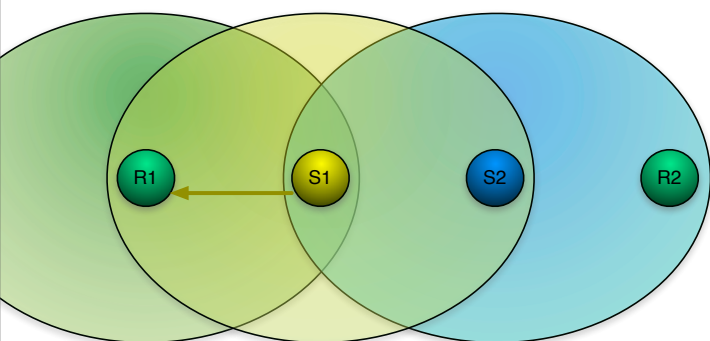
IEEE 802.11 RTS/CTS



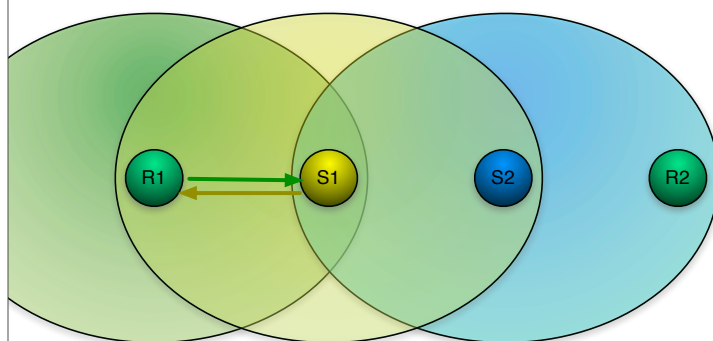
IEEE 802.11 RTS/CTS



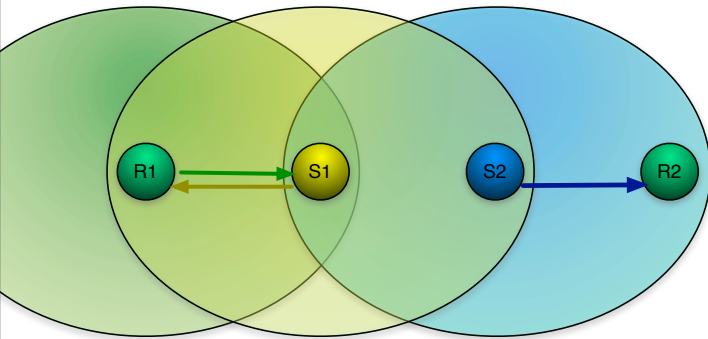
IEEE 802.11 RTS/CTS



IEEE 802.11 RTS/CTS



IEEE 802.11 RTS/CTS



Energy Consumption

	Idle	Receive	Transmit
[1]	1	1.05	1.4
[2]	1	2	2.5

[1] LAN MAN Standards Committee of the IEEE Computer Society, Wireless LAN medium access control (MAC) and physical layer (PHY) specification.

[2] Mark Stemm and Randy H. Katz, "Measuring and Reducing Energy Consumption of Network Interfaces in Hand-held Devices"

Duty Cycling

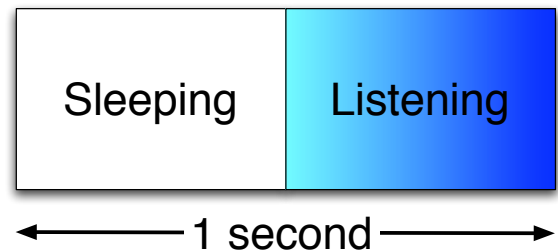
- **Reduces idle listening time**

- Sensors switch between sleep and active mode

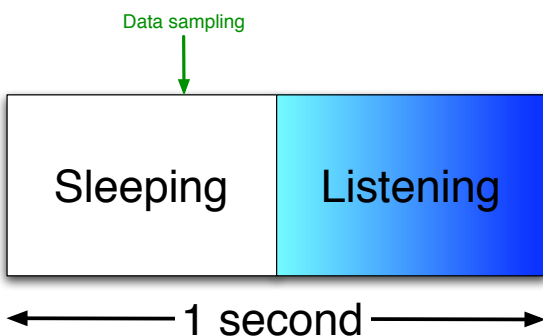
- **Suits low traffic networks**

- If data rate is very low, it is not necessary to keep sensors listening all the time
- Energy can be saved by turning off sensors

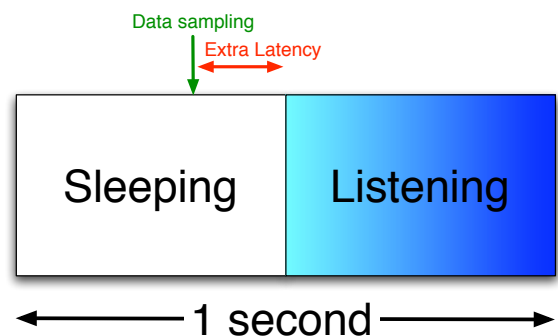
Duty Cycling



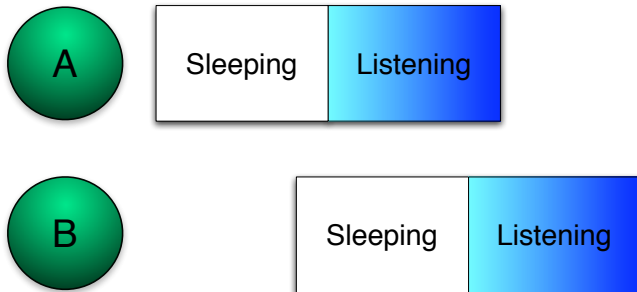
Duty Cycling



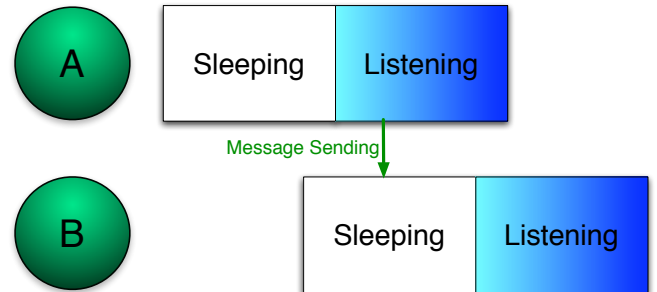
Duty Cycling



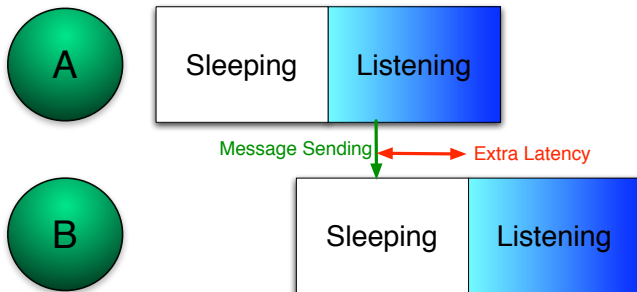
Duty Cycling



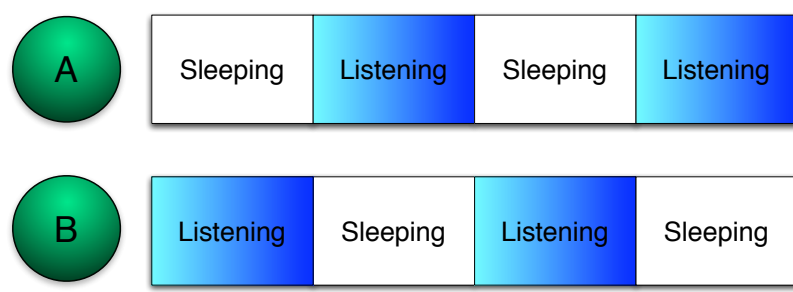
Duty Cycling



Duty Cycling



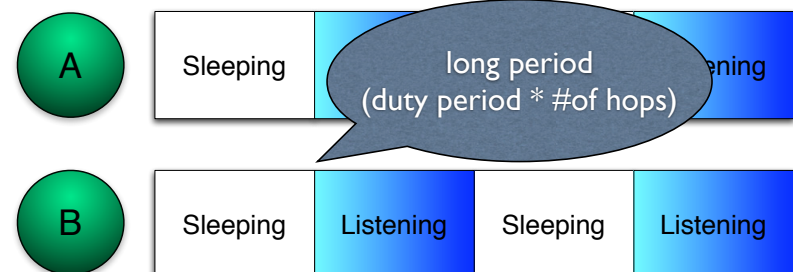
Duty Cycling



Duty Cycling



Duty Cycling



Time Synchronization

Time synchronization

- **Definition:** providing a common time scale for local clocks of nodes in the network
- Stamp event, duration between events, order events
- No global clock or shared memory

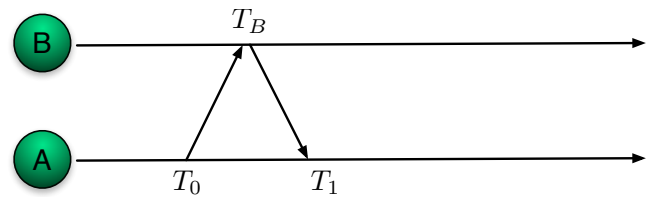
Time Synchronization

$$C_p(t) = a_p t + d_p$$

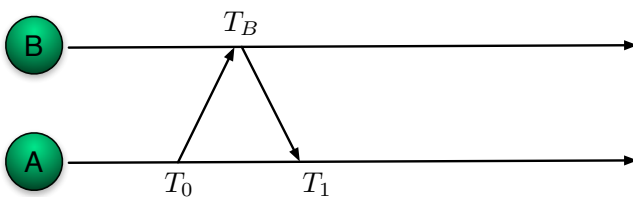
a_p : clock frequency

d_p : offset

Remote Clock Reading

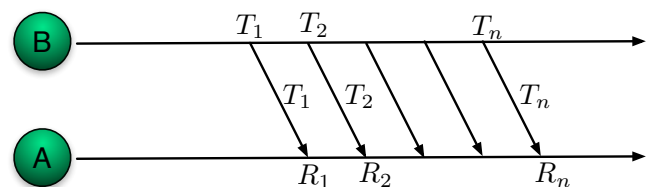


Remote Clock Reading

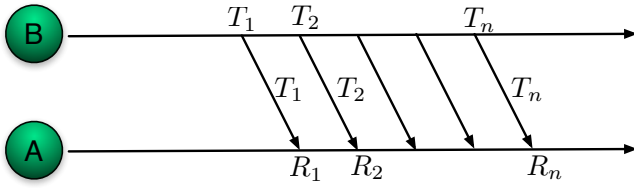


$$T_B(T_1) = T_B + \frac{T_1 - T_0}{2}$$

Time Transmission

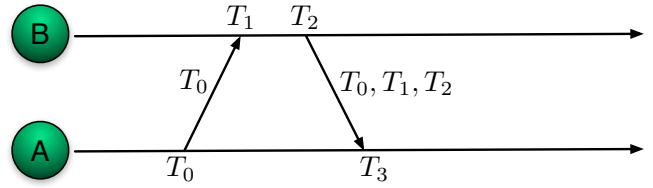


Time Transmission

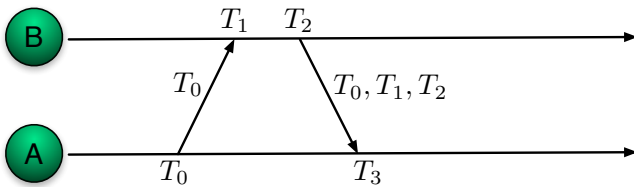


$$T_B(R_n) = R_n - \left(\frac{1}{n} \sum_{i=1}^n R_i - \frac{1}{n} \sum_{i=1}^n T_i \right) + d$$

Offset Delay Estimation



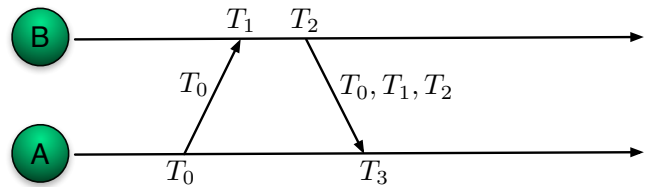
Offset Delay Estimation



$$\Delta = \frac{(T_1 - T_0) - (T_3 - T_2)}{2}$$

Drift

Offset Delay Estimation

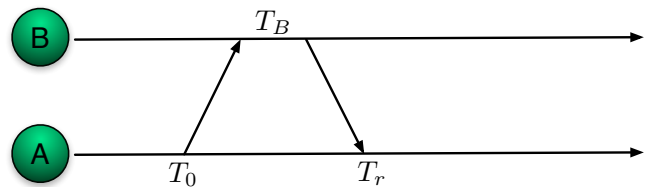
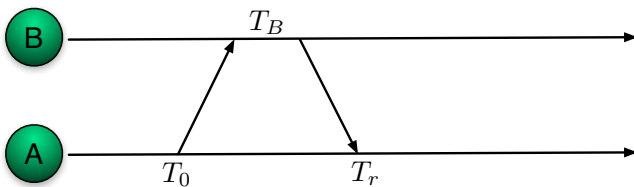


$$\Delta = \frac{(T_1 - T_0) - (T_3 - T_2)}{2} \quad d = \frac{(T_1 - T_0) + (T_3 - T_2)}{2}$$

Drift

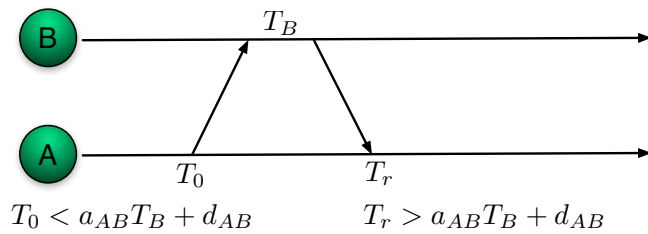
Offset

Set Valued Estimation



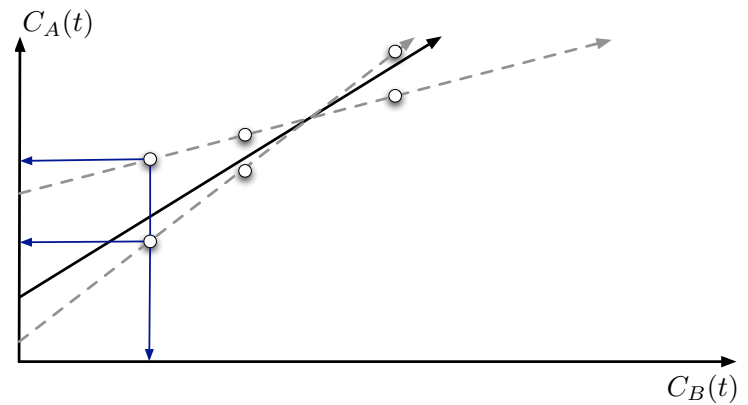
$$C_A(t) = a_{AB}C_B(t) + d_{AB}$$

Set Valued Estimation

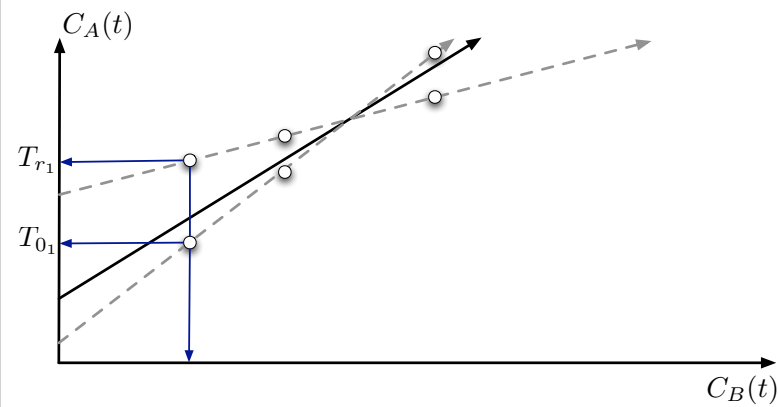


$$C_A(t) = a_{AB}C_B(t) + d_{AB}$$

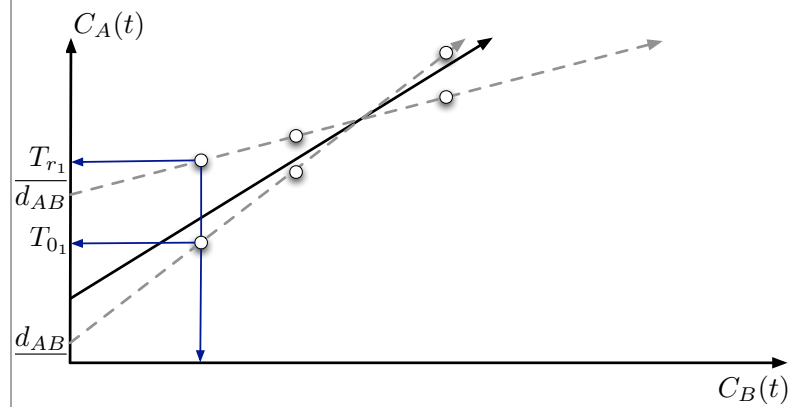
Set Valued Estimation



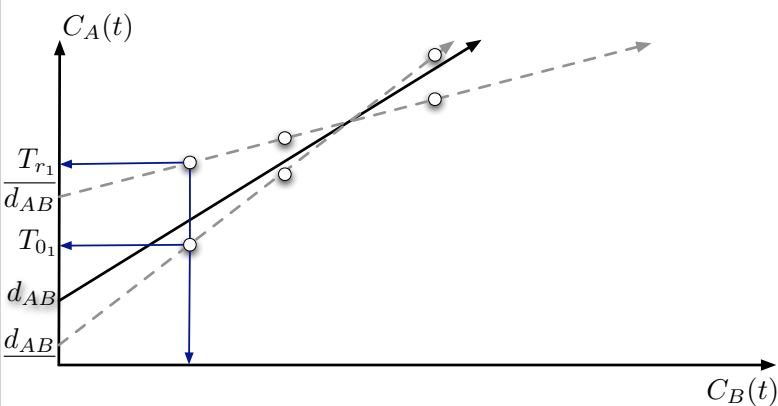
Set Valued Estimation



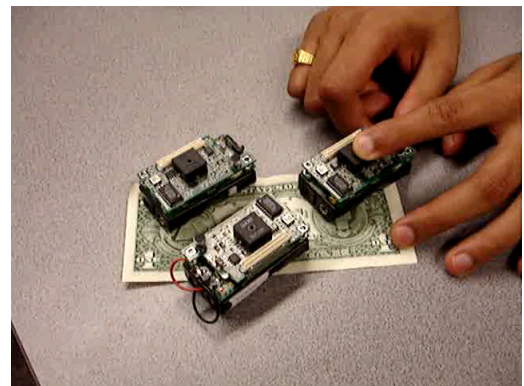
Set Valued Estimation



Set Valued Estimation

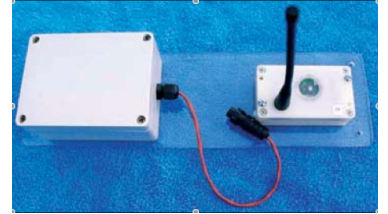


Time Synchronization

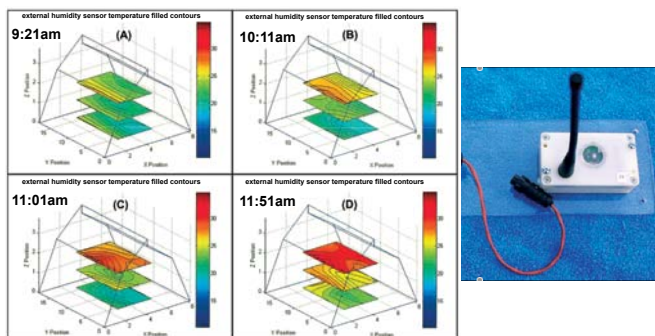


Applications

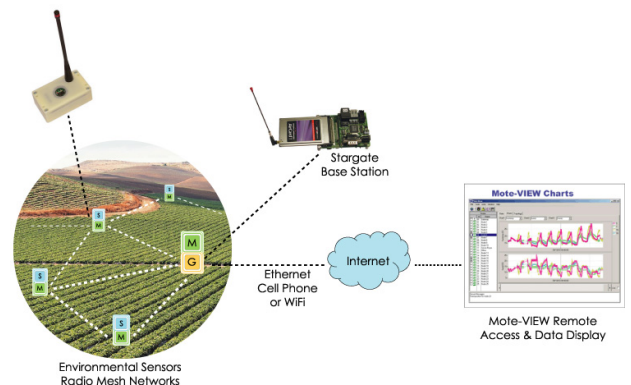
Habitat Monitoring



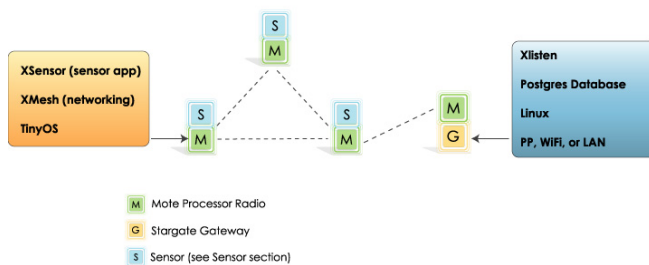
Habitat Monitoring



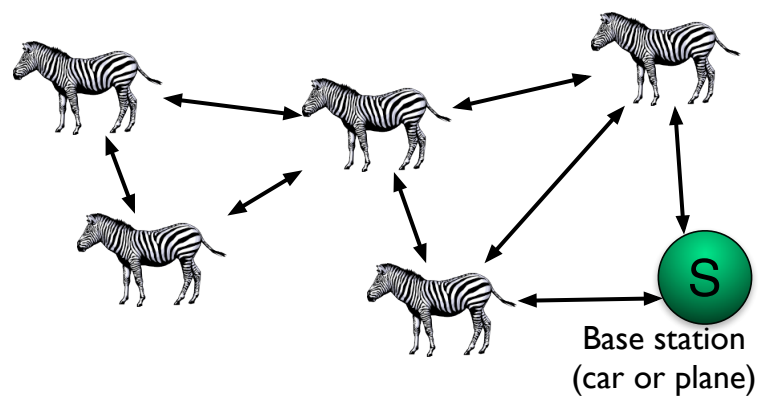
Environment Monitoring



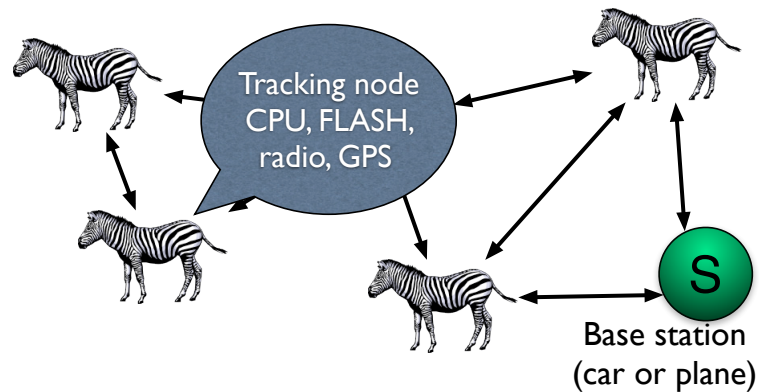
Environment Monitoring



Zebranet



Zebranet



Zebranet

Attribute	Zebranet	Sensors
Mobility	High	Low/static
Range	Miles	Meters
Frequency	Constant	Sporadic
Power	Hundreds of mW	Tens of mW

Zebranet



Zebranet

