

# Introduction to Sensor Networks

Maria Potop-Butucaru, Franck Petit, Sébastien Tixeuil  
Université Pierre & Marie Curie - Paris 6, France  
[Firstname.Lastname@lip6.fr](mailto:Firstname.Lastname@lip6.fr)

1

## 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, Environment, Wildlife, etc.

4

## COMMON-Sense Net (CSN)



Tumkur, Karnataka

<http://commonsense.epfl.ch/>

6-2

## Outline

- Motivation
- Architecture
- Overview

2

## Applications

## Motivation

3

## COMMON-Sense Net (CSN)

Tumkur, Karnataka

<http://commonsense.epfl.ch/>

6-1

## COMMON-Sense Net (CSN)



Tumkur, Karnataka

<http://commonsense.epfl.ch/>

6-3

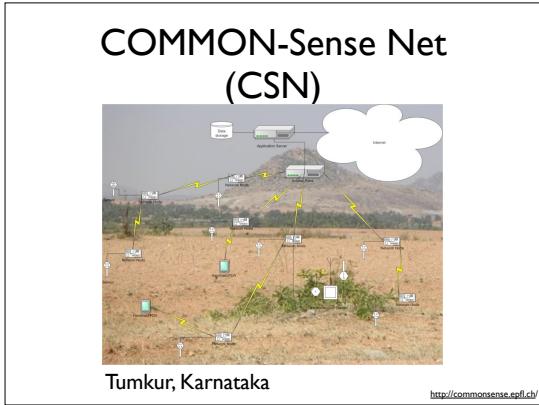
## COMMON-Sense Net (CSN)



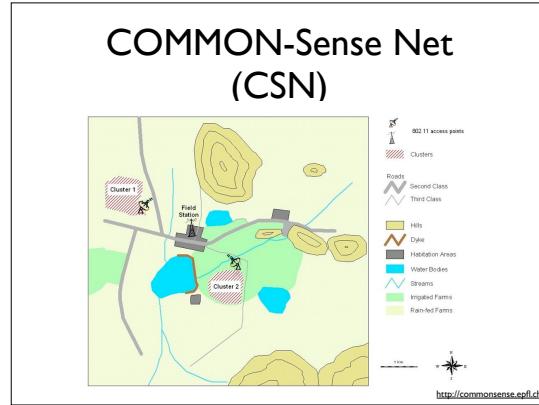
Tumkur, Karnataka

<http://commonsense.epfl.ch/>

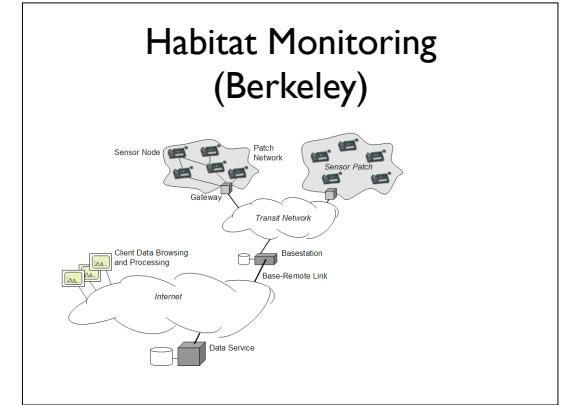
6-4



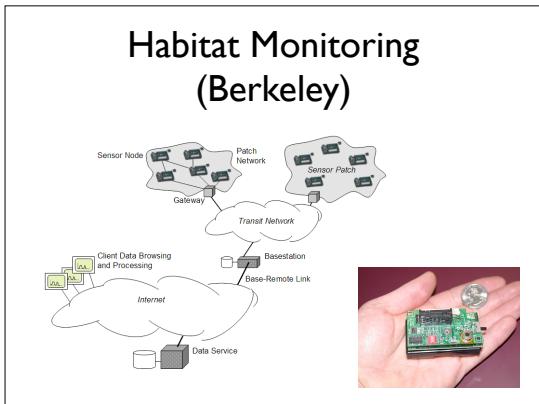
6-5



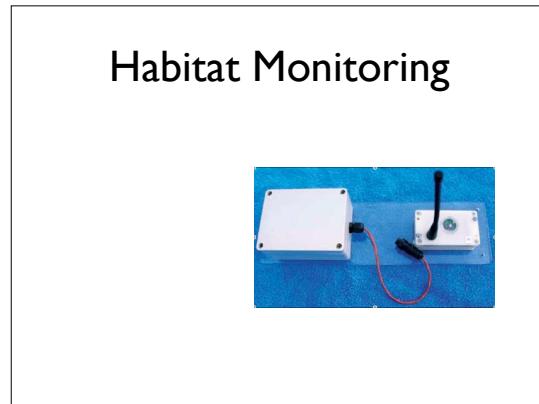
7



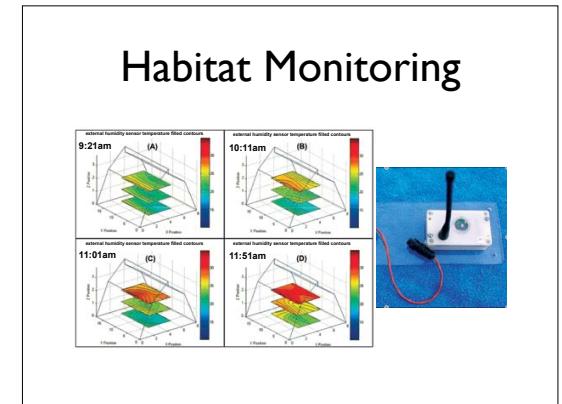
8-1



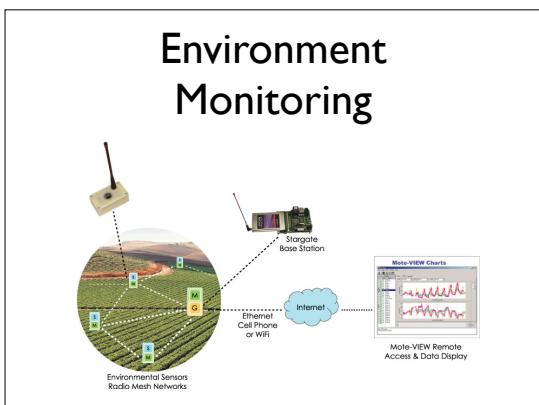
8-2



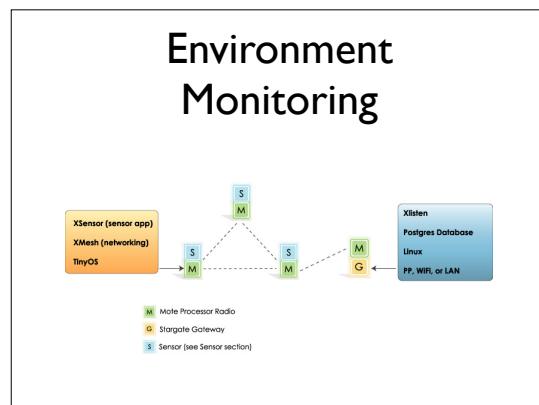
9-1



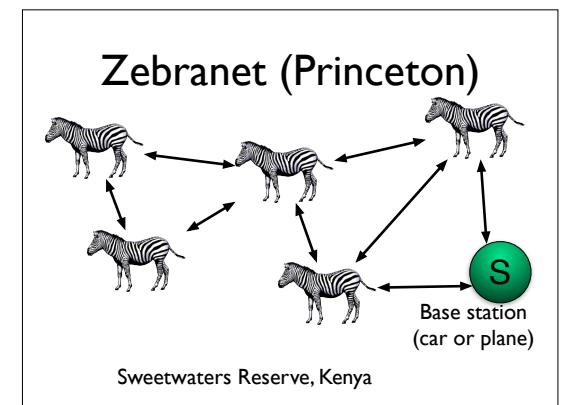
9-2



10-1

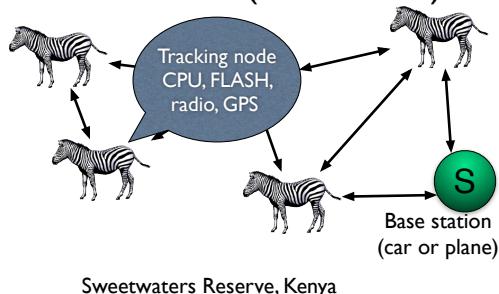


10-2



11-1

## Zebranet (Princeton)



11-2

## Zebranet

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

12

## Zebranet



13-1

## Zebranet



Sweetwaters Reserve, Kenya

13-2

## Motivation

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

14

## Sensor vs. *ad hoc*

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

15

## Issues

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

## Issues

No infrastructure

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

16-1

16-2

## Issues

No human intervention

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

16-3

## Issues

- *ad hoc* deployment
  - Unattended
  - Untethered
  - Dynamic changes
- No permanent power supply

16-4

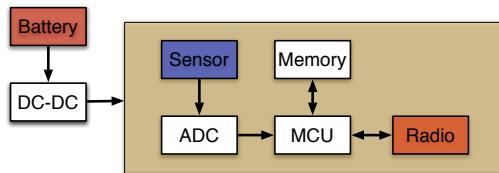
## Issues

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

Architecture

17

## Sensor Node Architecture



18

## Available Devices



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

19

## Available Devices



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

20

## Available Devices



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

21

## Available Devices



- GreenNet (ST Micro)
- 802.15.4 radio
- Solar energy harvesting

22

## Sensors

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

23

## Sensors

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

24

## Sensors

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

25

## Sensors

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

26

## Sensors

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

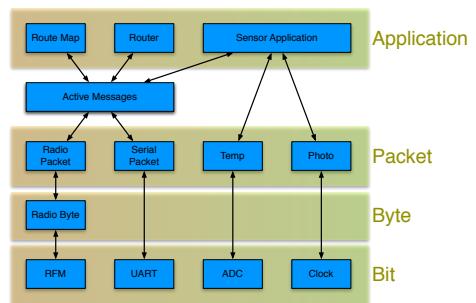
27

## Sensors

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

28

## TinyOS Application



30

## 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

31

## 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

32

## Other OSes for WSN

- MagnetOS
  - Virtual machines, byte code
- Mantis
  - Pure multithread
- Contiki
  - Dynamic linking of binaries
  - Event/Thread hybrid

33

## Overview

34

## Issues and Solutions

- Localization
- Routing
- Medium Access Control
- Applications

35

## Localization

36

## Localization

- Fine-grained
  - Timing
  - Signal strength
  - Signal pattern matching
  - Directionality
- Coarse-grained

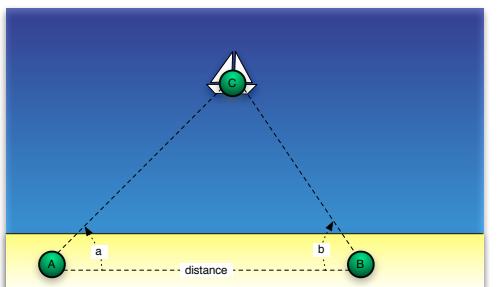
37

## Triangulation



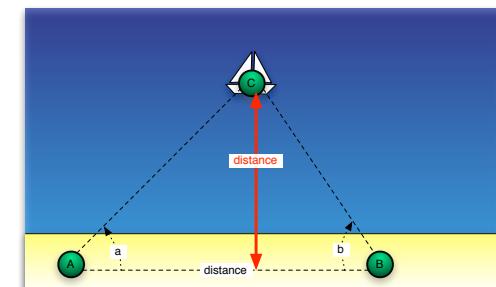
38-1

## Triangulation



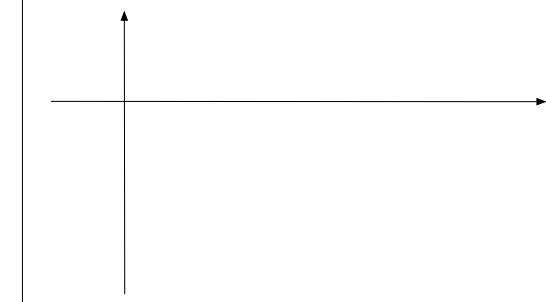
38-2

## Triangulation



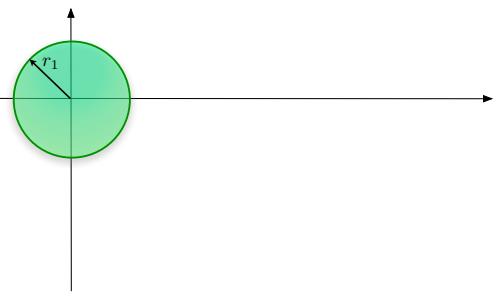
38-3

## Trilateration



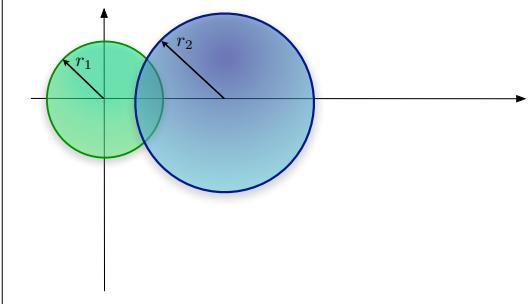
39-1

## Trilateration



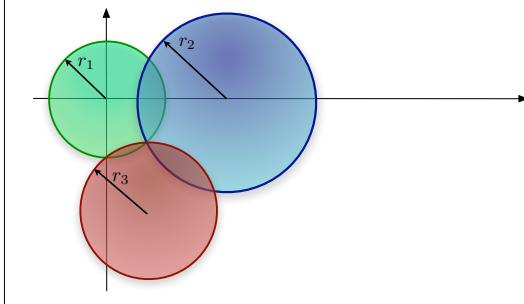
39-2

## Trilateration



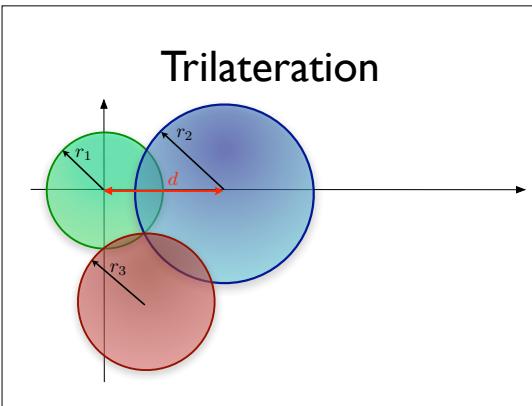
39-3

## Trilateration



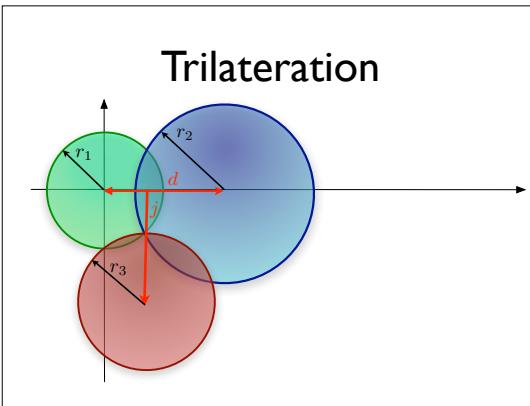
39-4

## Trilateration



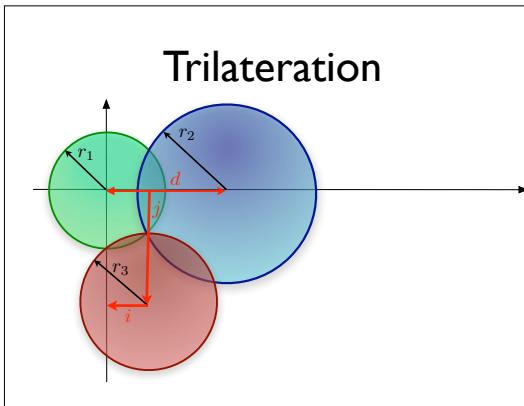
39-5

## Trilateration



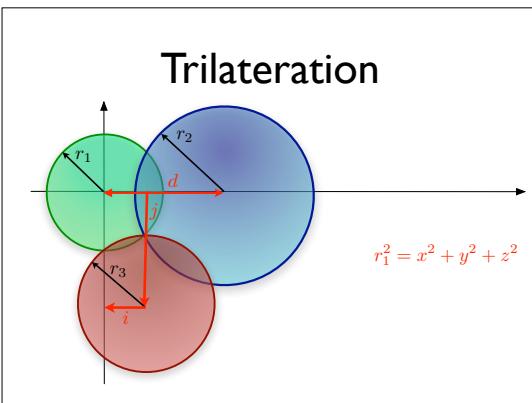
39-6

## Trilateration



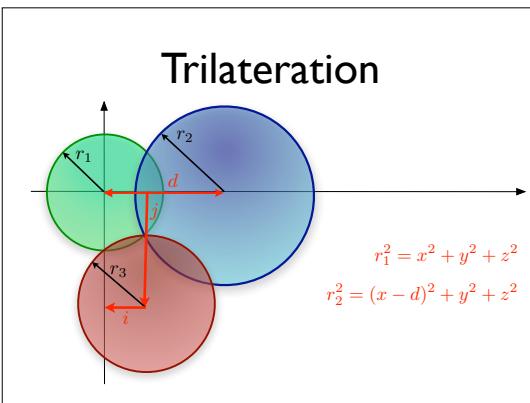
39-7

## Trilateration



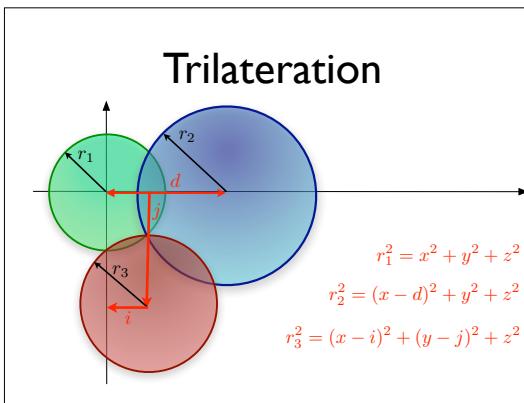
39-8

## Trilateration

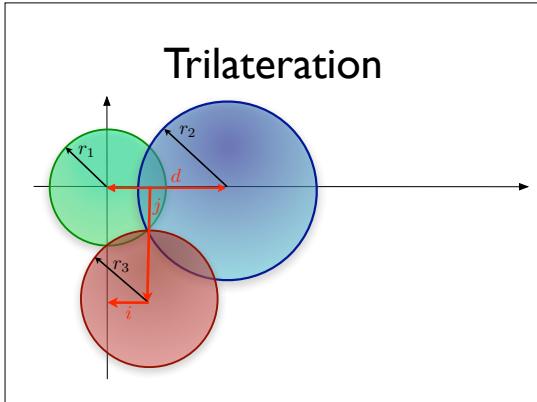


39-9

## Trilateration



39-10



### Trilateration

$$x = \frac{r_1^2 - r_2^2 + d^2}{2d}$$

39-12

### Trilateration

$$x = \frac{r_1^2 - r_2^2 + d^2}{2d}$$

$$y = \frac{r_1^2 - r_3^2 + (x - i)^2}{2j} + \frac{j}{2} + \frac{x^2}{2j}$$

39-13

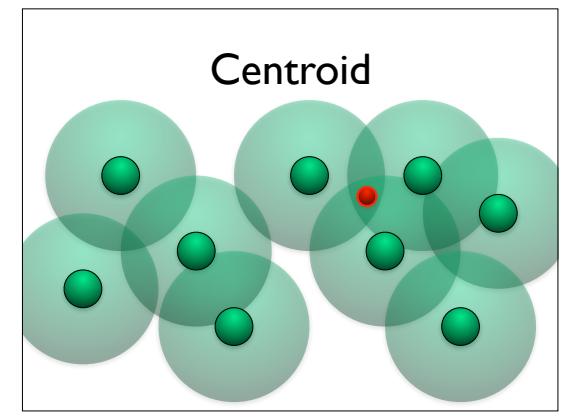
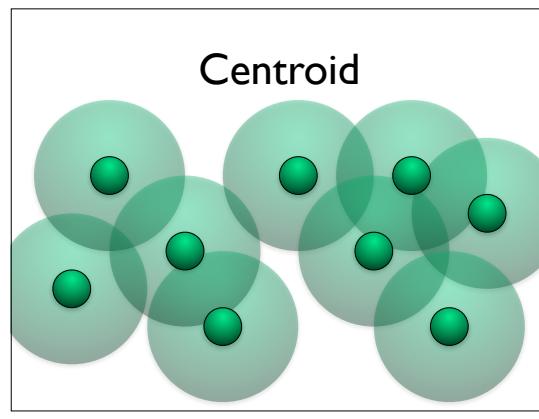
### Trilateration

$$x = \frac{r_1^2 - r_2^2 + d^2}{2d}$$

$$y = \frac{r_1^2 - r_3^2 + (x - i)^2}{2j} + \frac{j}{2} + \frac{x^2}{2j}$$

$$z = \sqrt{r_1^2 - x^2 - y^2}$$

39-14



### Centroid

$$(X_{est}, Y_{est}) = \left( \frac{X_1 + X_2 + X_3}{3}, \frac{Y_1 + Y_2 + Y_3}{3} \right)$$

40-3

### Routing

41

- ### Routing
- Classical flooding
    - *Impllosion*
    - Resource management
  - Negotiation based protocols
    - SPIN
    - Directed Diffusion
- 42

## Negotiation Based Protocols

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

43

## SPIN

- **ADV:** *advertize* that new data is available and described
- **REQ:** *request* to receive data
- **DATA:** *actual data*

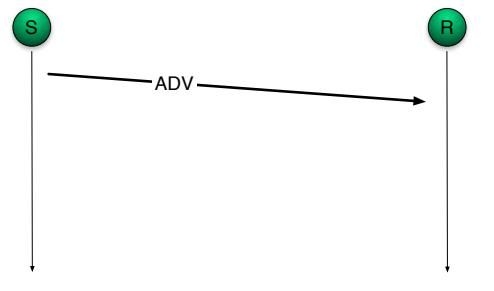
44

## SPIN-PP



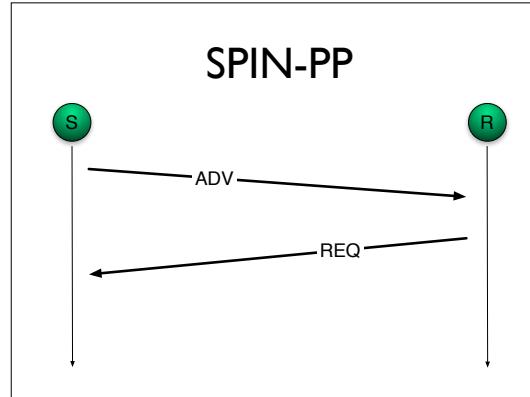
45-1

## SPIN-PP



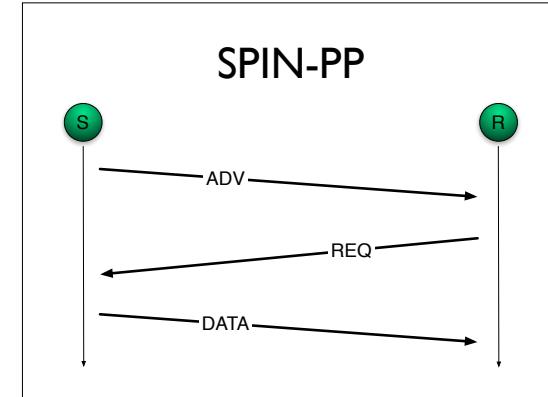
45-2

## SPIN-PP



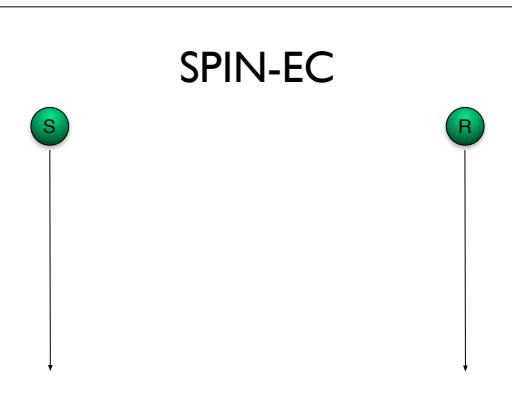
45-3

## SPIN-PP



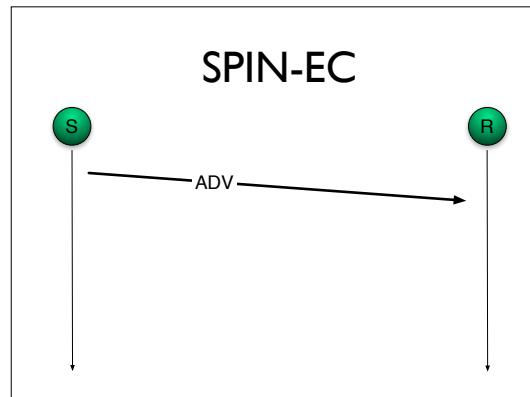
45-4

## SPIN-EC



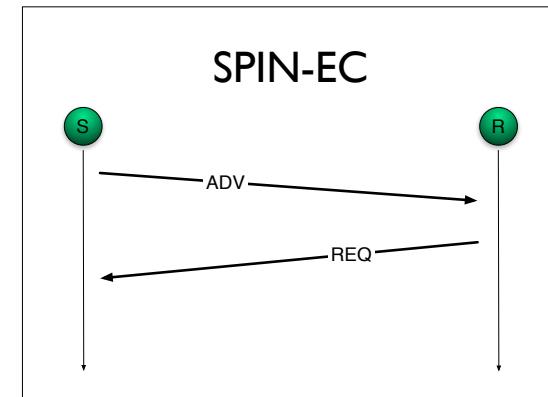
46-1

## SPIN-EC



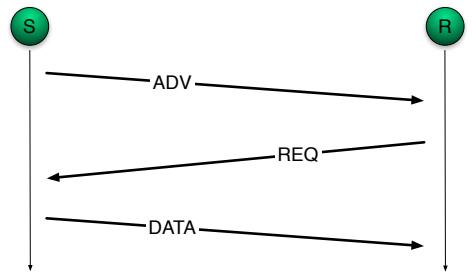
46-2

## SPIN-EC



46-3

### SPIN-EC



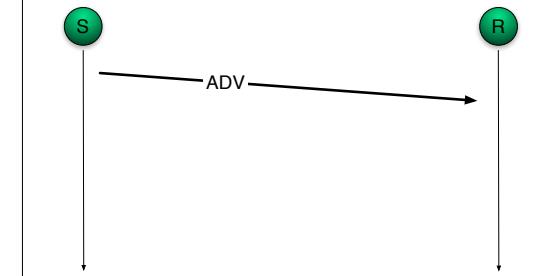
46-4

### SPIN-EC



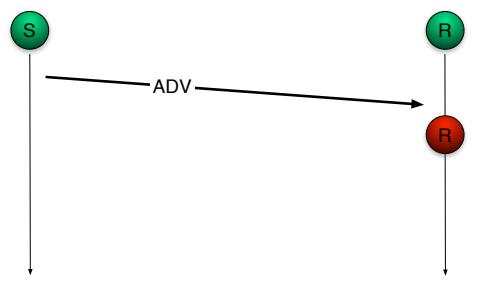
47-1

### SPIN-EC



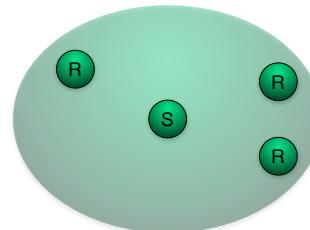
47-2

### SPIN-EC



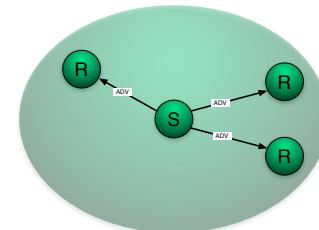
47-3

### SPIN-BC, SPIN-RL



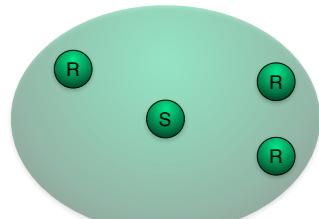
48-1

### SPIN-BC, SPIN-RL



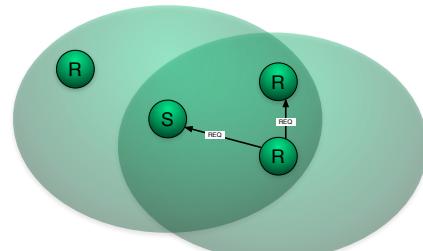
48-2

### SPIN-BC, SPIN-RL



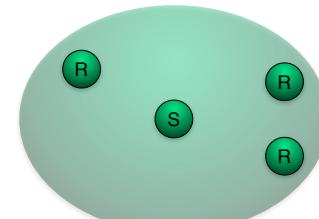
48-3

### SPIN-BC, SPIN-RL



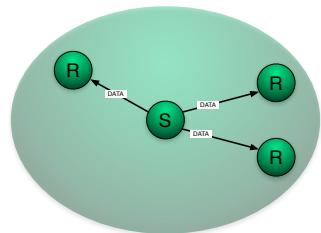
48-4

### SPIN-BC, SPIN-RL



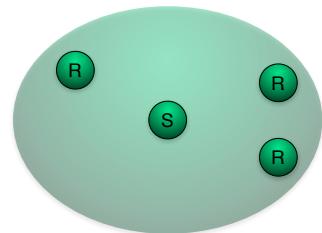
48-5

## SPIN-BC, SPIN-RL



48-6

## SPIN-BC, SPIN-RL



48-7

## Directed Diffusion

- Destination-initiated (sink) reactive routing technique
- Data is named by an attribute-value pair
- Sensing tasks are initiated in order to match events *interests*
- All nodes maintain interest cache for each requested interest

49

## Interests

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

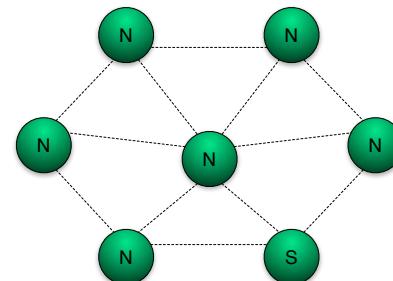
50

## Returned Data

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

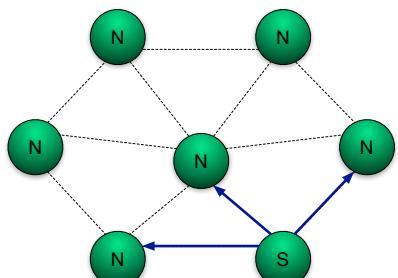
51

## Directed Diffusion



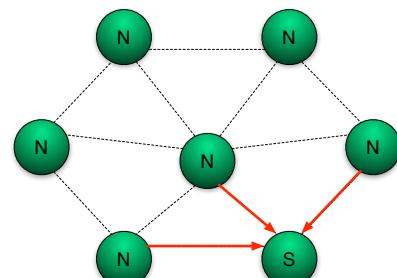
52-1

## Directed Diffusion



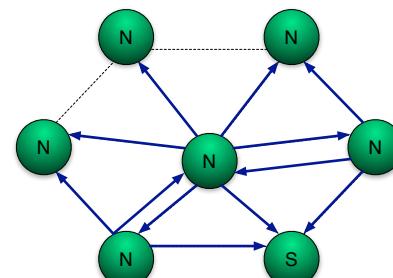
52-2

## Directed Diffusion



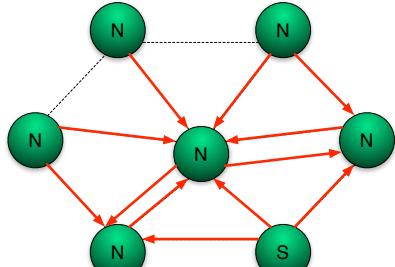
52-3

## Directed Diffusion



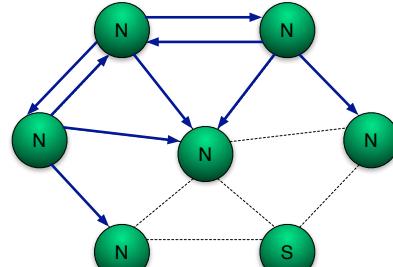
52-4

## Directed Diffusion



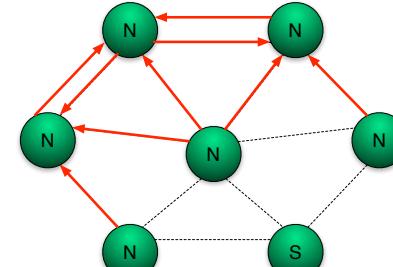
52-5

## Directed Diffusion



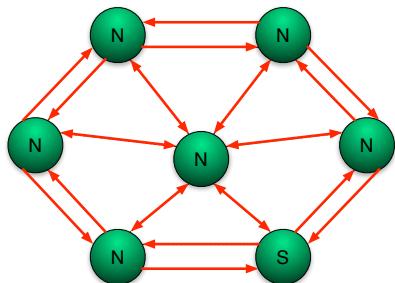
52-6

## Directed Diffusion



52-7

## Directed Diffusion



52-8

## Interest Cache

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

53

## Interest Forwarding

- When new interest/task is 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

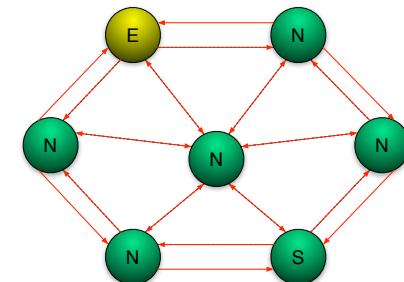
54

## 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

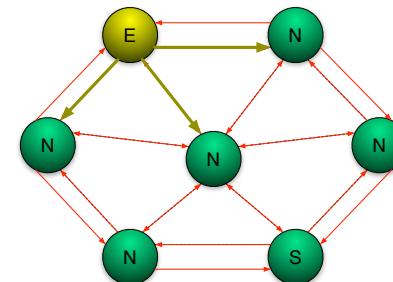
55

## Directed Diffusion



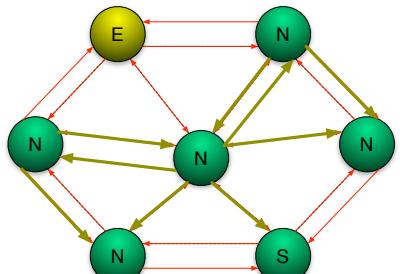
56-1

## Directed Diffusion



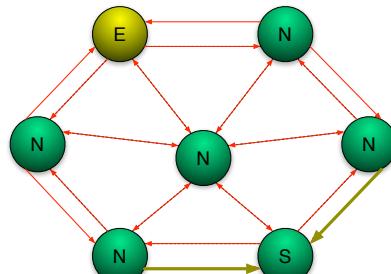
56-2

### Directed Diffusion



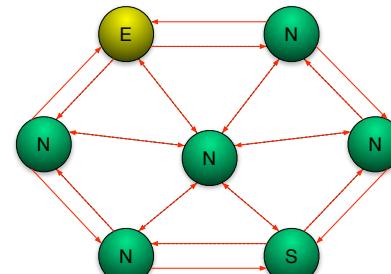
56-3

### Directed Diffusion



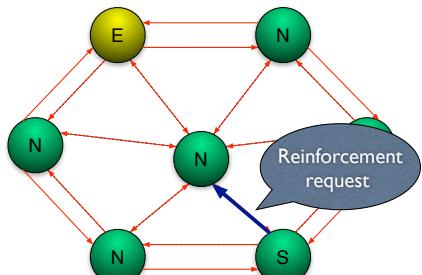
56-4

### Directed Diffusion



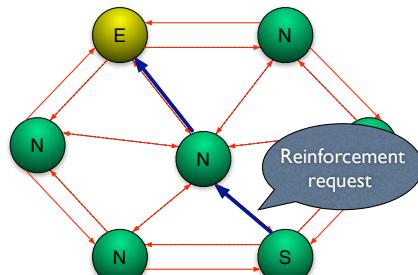
56-5

### Directed Diffusion



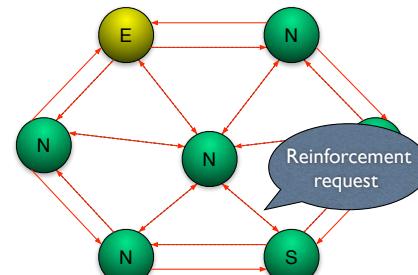
56-6

### Directed Diffusion



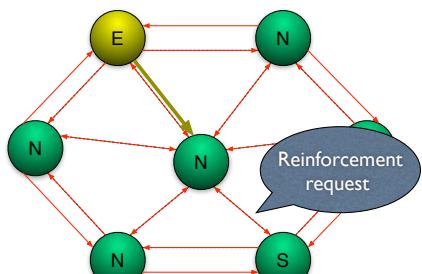
56-7

### Directed Diffusion



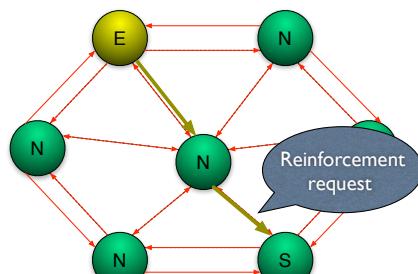
56-8

### Directed Diffusion



56-9

### Directed Diffusion



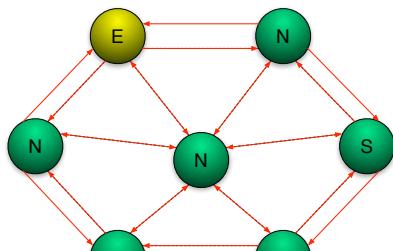
56-10

### 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”

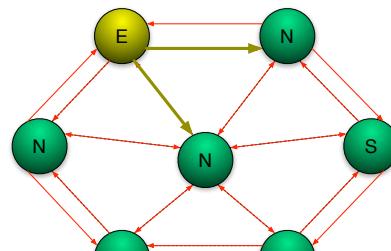
57

### Directed Diffusion



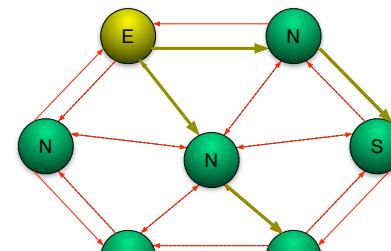
58-1

### Directed Diffusion



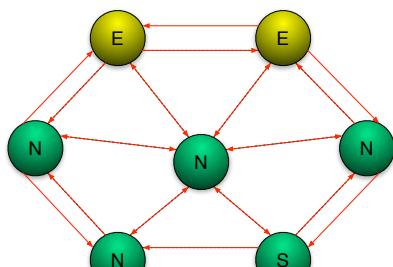
58-2

### Directed Diffusion



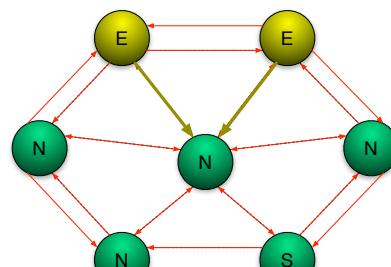
58-3

### Directed Diffusion



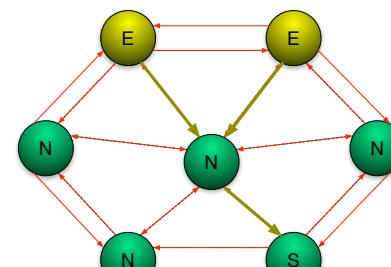
59-1

### Directed Diffusion



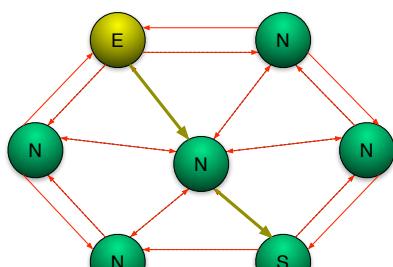
59-2

### Directed Diffusion



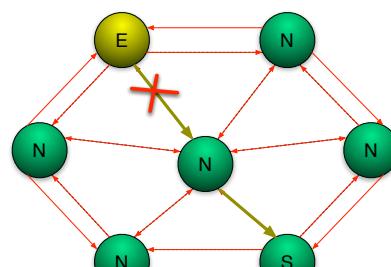
59-3

### Directed Diffusion



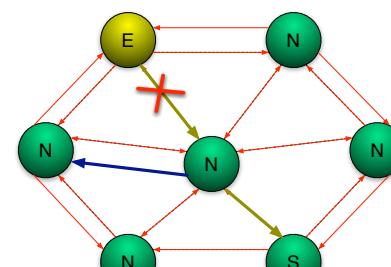
60-1

### Directed Diffusion



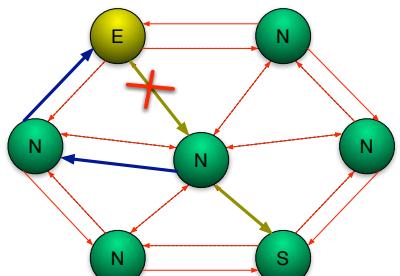
60-2

### Directed Diffusion



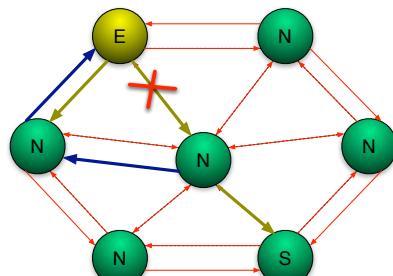
60-3

## Directed Diffusion



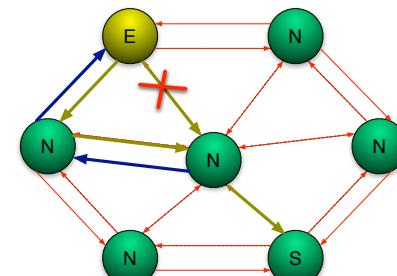
60-4

## Directed Diffusion



60-5

## Directed Diffusion



60-6

## Directed Diffusion

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

61

## Directed Diffusion

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

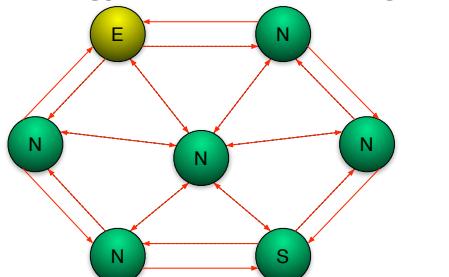
62

## 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)

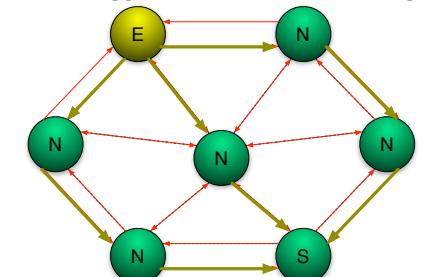
63

## Energy Aware Routing



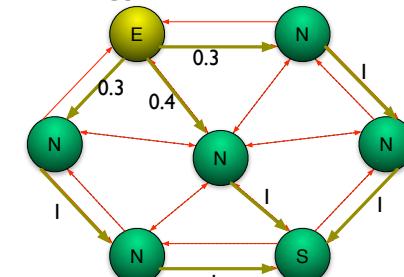
64-1

## Energy Aware Routing



64-2

## Energy Aware Routing



64-3

## Medium Access Control

65

### 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

- 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

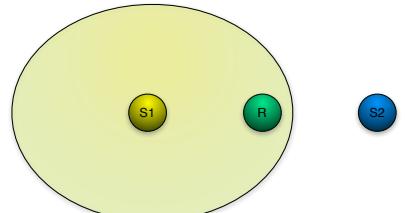
66-3

### 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

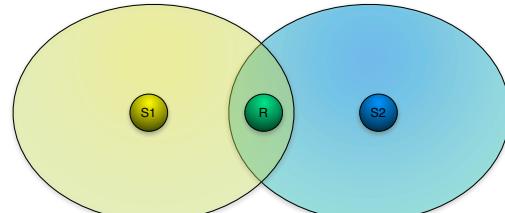
66-4

### Hidden Terminal problem



67-2

### Hidden Terminal problem



67-3

### 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

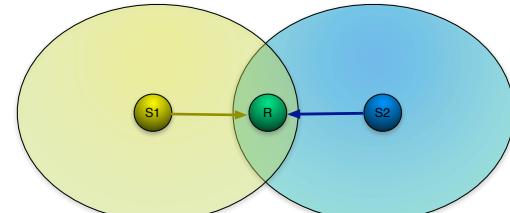
66-2

### Hidden Terminal problem

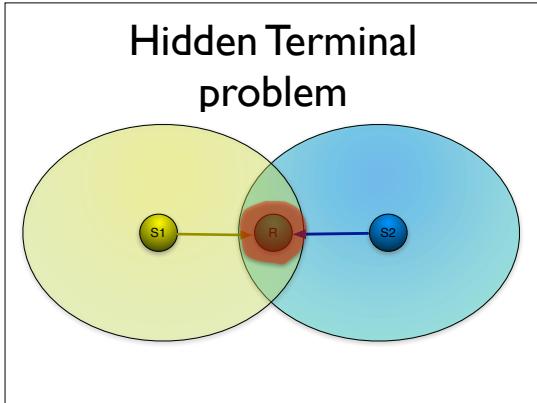


67-1

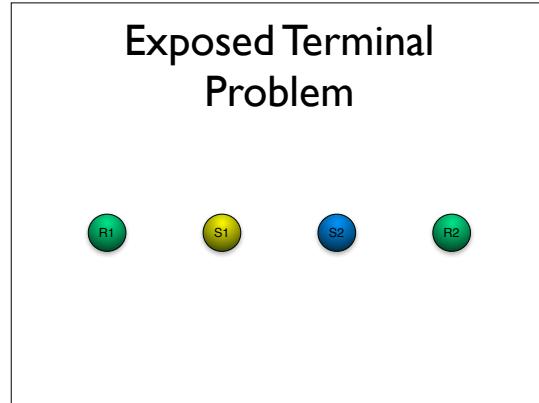
### Hidden Terminal problem



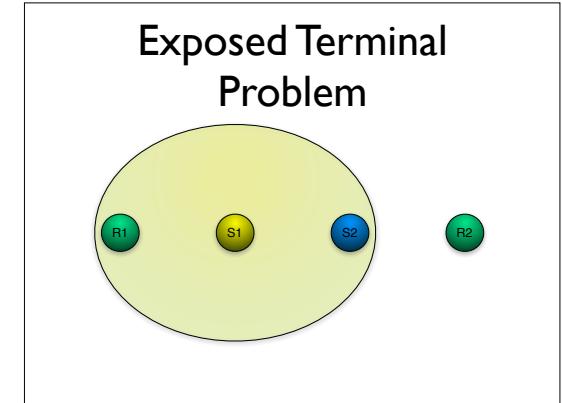
67-4



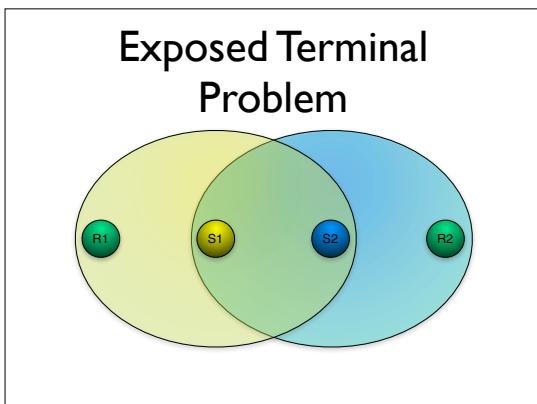
67-5



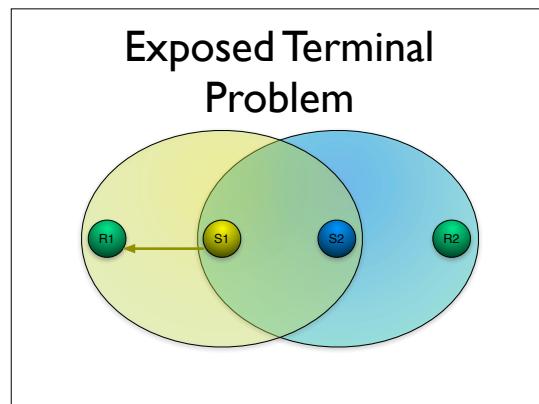
68-1



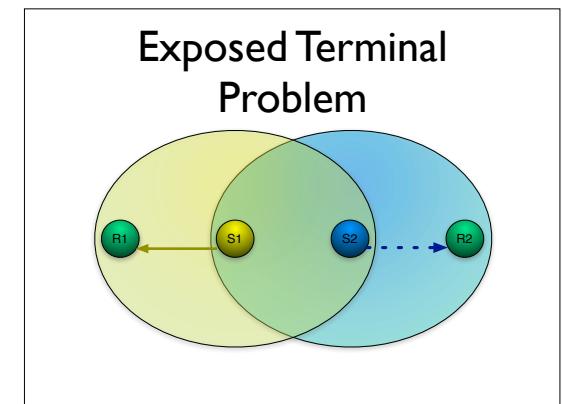
68-2



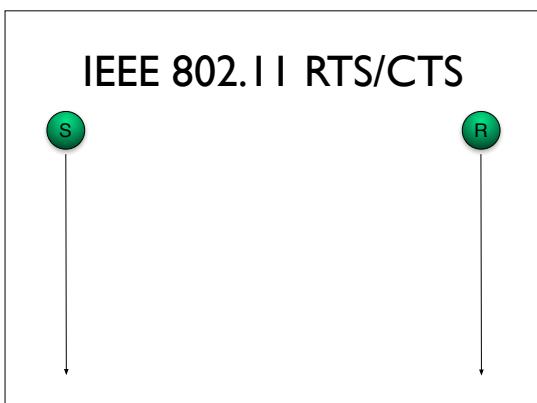
68-3



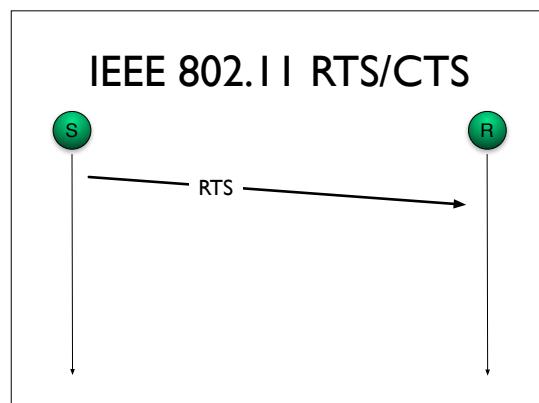
68-4



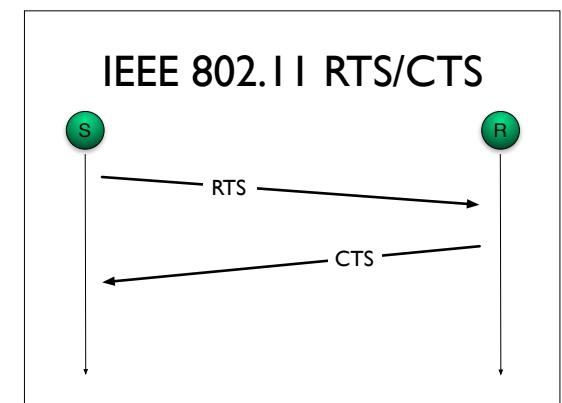
68-5



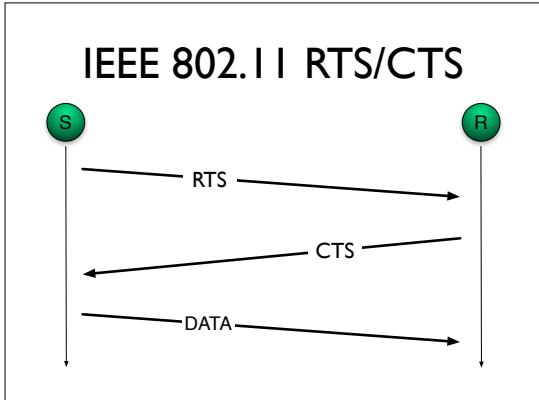
69-1



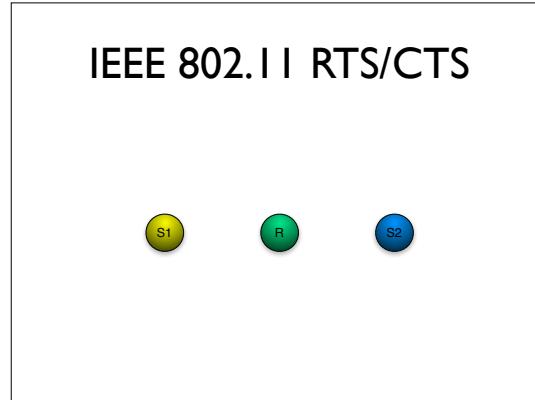
69-2



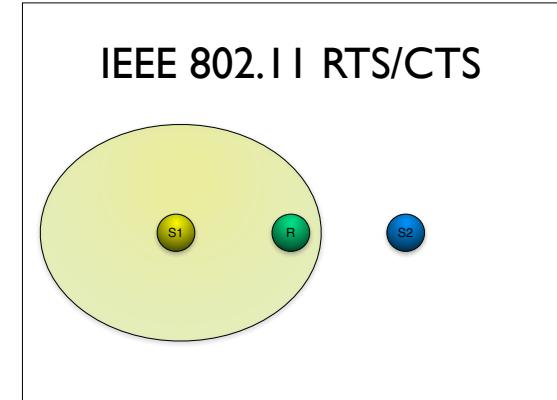
69-3



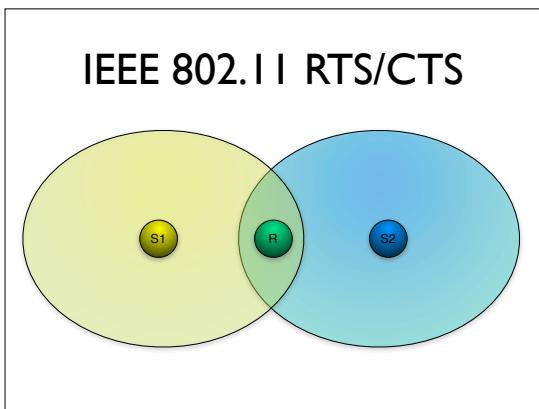
69-4



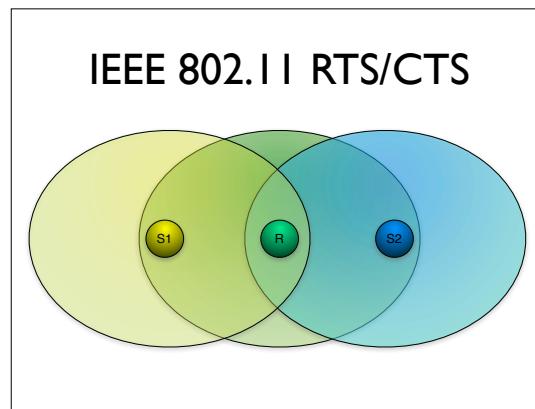
70-1



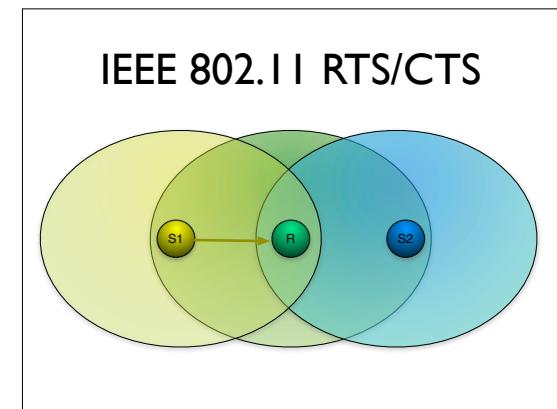
70-2



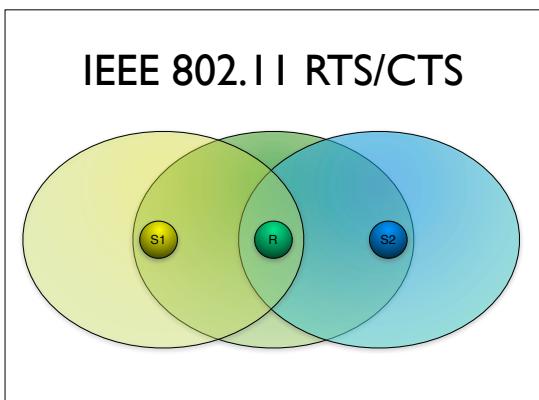
70-3



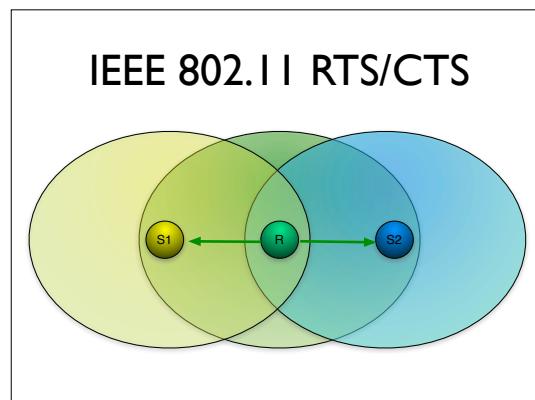
70-4



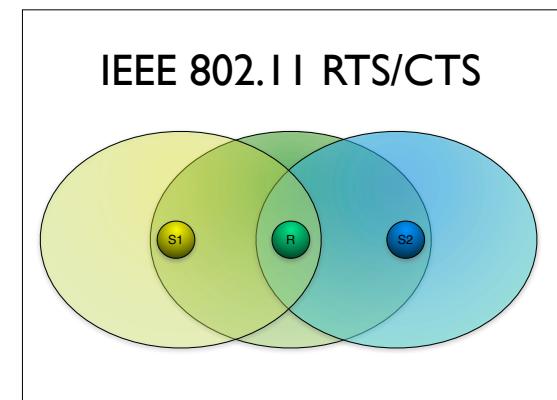
70-5



70-6

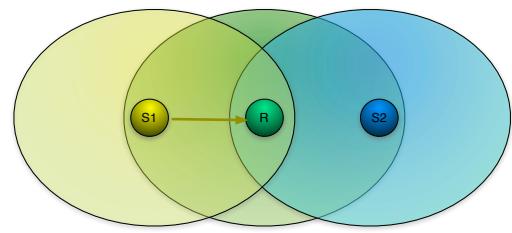


70-7



70-8

### IEEE 802.11 RTS/CTS



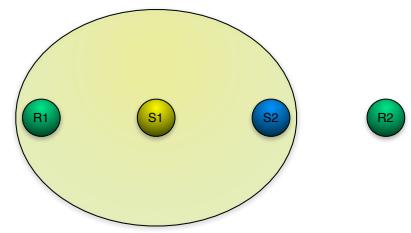
70-9

### IEEE 802.11 RTS/CTS



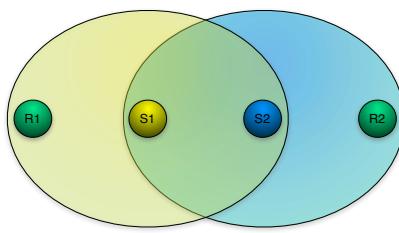
71-1

### IEEE 802.11 RTS/CTS



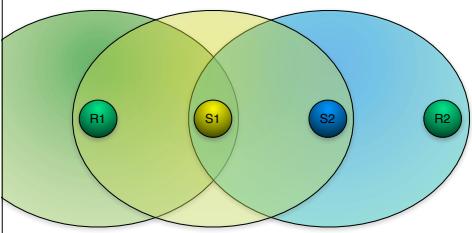
71-2

### IEEE 802.11 RTS/CTS



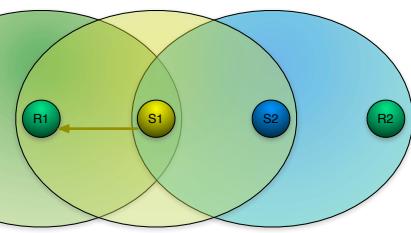
71-3

### IEEE 802.11 RTS/CTS



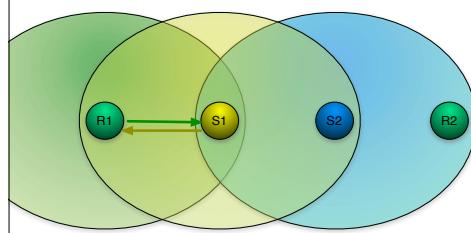
71-4

### IEEE 802.11 RTS/CTS



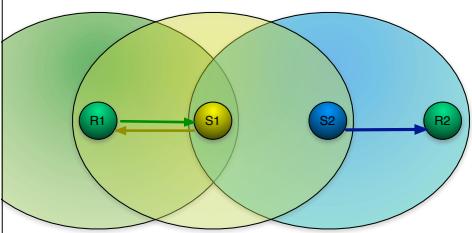
71-5

### IEEE 802.11 RTS/CTS



71-6

### IEEE 802.11 RTS/CTS



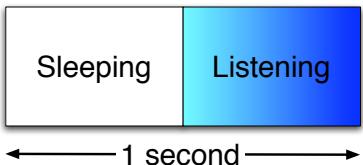
71-7

### 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

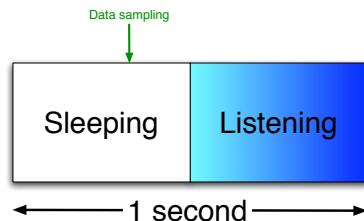
72

## Duty Cycling



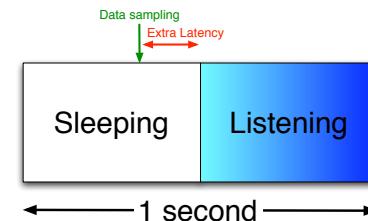
73-1

## Duty Cycling



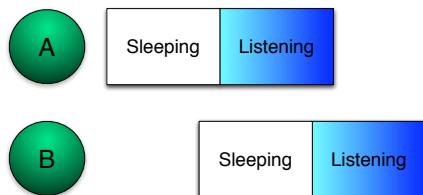
73-2

## Duty Cycling



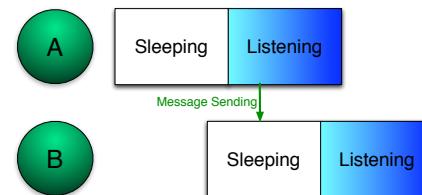
73-3

## Duty Cycling



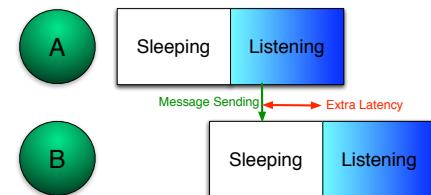
74-1

## Duty Cycling



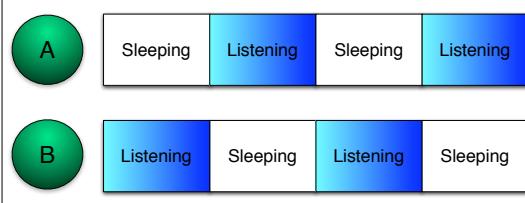
74-2

## Duty Cycling



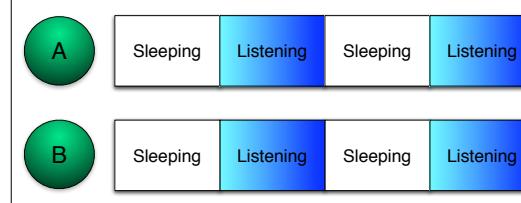
74-3

## Duty Cycling



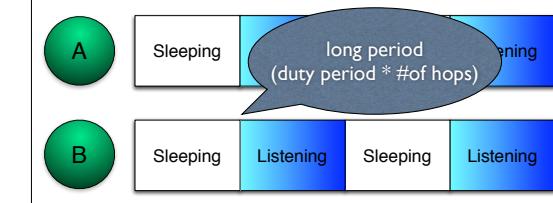
75

## Duty Cycling



76-1

## Duty Cycling

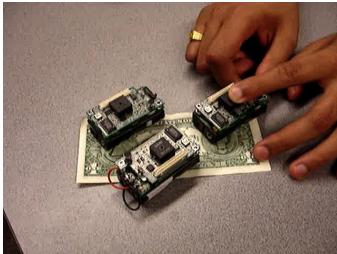


76-2

## Time Synchronization

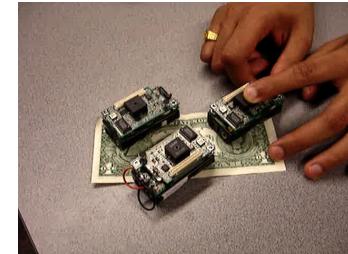
77

## Time Synchronization



78-1

## Time Synchronization



78-2

## 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

79

## Time Synchronization

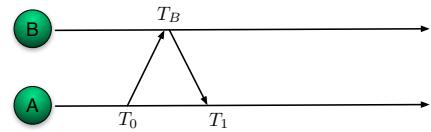
$$C_p(t) = a_p t + d_p$$

$a_p$  : clock frequency

$d_p$  : offset

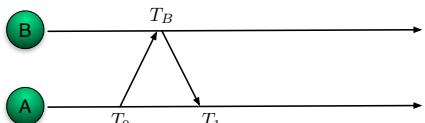
80

## Remote Clock Reading



81-1

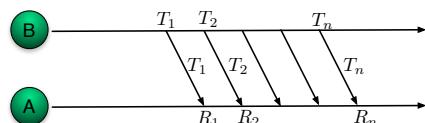
## Remote Clock Reading



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

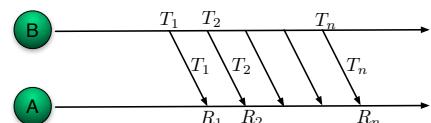
81-2

## Time Transmission



82-1

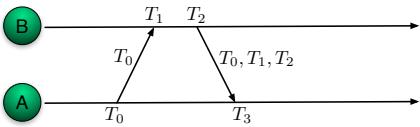
## Time Transmission



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

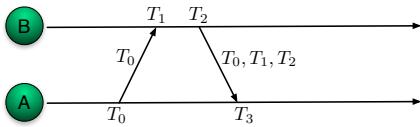
82-2

## Offset Delay Estimation



83-1

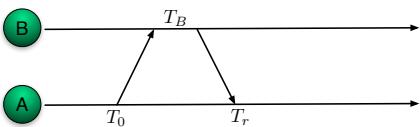
## Offset Delay Estimation



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

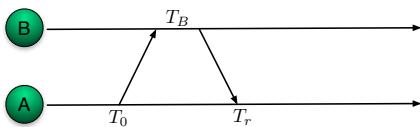
Drift

## Set Valued Estimation



84-1

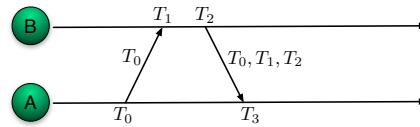
## Set Valued Estimation



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

84-2

## Offset Delay Estimation



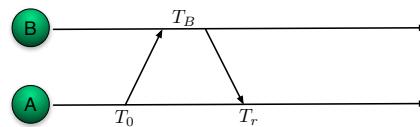
$$\Delta = \frac{(T_1 - T_0) - (T_3 - T_2)}{2} \quad d = \frac{(T_1 - T_0) + (T_3 - T_2)}{2}$$

Drift

Offset

83-3

## Set Valued Estimation

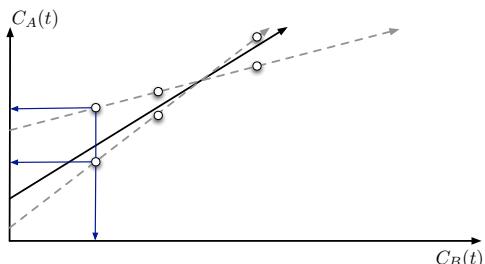


$$T_0 < a_{AB}T_B + d_{AB} \quad T_r > a_{AB}T_B + d_{AB}$$

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

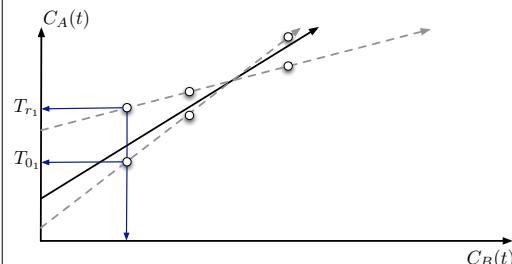
84-3

## Set Valued Estimation



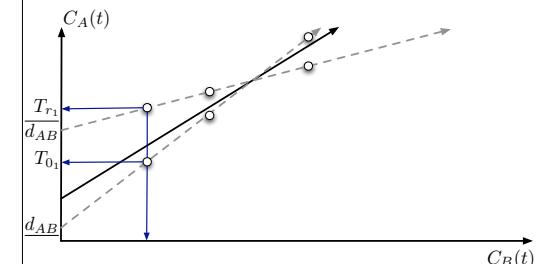
85-1

## Set Valued Estimation



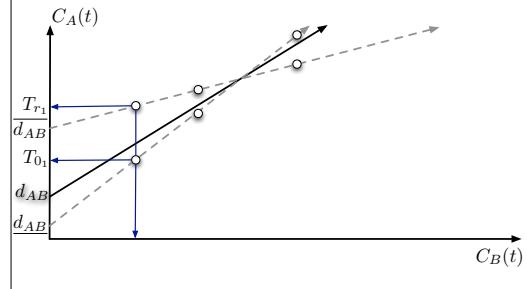
85-2

## Set Valued Estimation



85-3

## Set Valued Estimation



85-4

## Conclusion

86

## Sensor Networks

- Driven by applications
- Connexion between Computer Science and Biology, Environment, Rescue, etc.
- Hard problems yet to be solved

87