Institute for the Wireless Internet of Things at Northeastern University

EECE 5155
Wireless Sensor Networks (and The Internet of Things)

Prof. Francesco Restuccia

WSNs Dependability Aspects

- ***Coverage & deployment***
  - Is there a sufficient number of nodes such that an event can be detected at all?
  - Can such data be accurately measured?
  - How do they have to be deployed?
- ***Information accuracy***
  - Which of the measured data have to be transported where such that a desired accuracy is achieved?
  - How do we deal with inaccurate measurements?

Dependable Data Transport

- ***Dependable data transport***
  - Once it is clear which data should arrive where, how to make sure that it actually arrives?
  - How to deal with ***transmission errors*** and ***omission errors/congestion***?

Transport Layer

- Transport layer handles
  - Dependable (reliable) data transport
    - Deals with end-to-end packet losses
  - Flow Control
  - Congestion Control
    - Regulates transmission rates at the source to avoid losses due to congestion
  - Network Abstraction
    - TCP Sockets in Internet

- Classical solutions (TCP/UDP) cannot be used in sensor networks, why?

  - **Resource constraints!**

Dependability: Terminology

- *(Steady state) availability*
  - Probability that a system is operational at any given point in time
  - Assumption: system can fail and will repair itself
- *Reliability at time t*
  - Probability that system works correctly during the entire interval [0,t)
  - Assumption: It worked correctly at system start t=0
- *Responsiveness*
  - Probability of meeting a deadline
  - Even in presence of faults
- *Packet success probability*
  - Probability that a packet (correctly) reaches its destination
  - Related: packet error rate, packet loss rate
- *Bit error rate*
  - Probability of an incorrect bit
  - Channel model determines precise error patterns

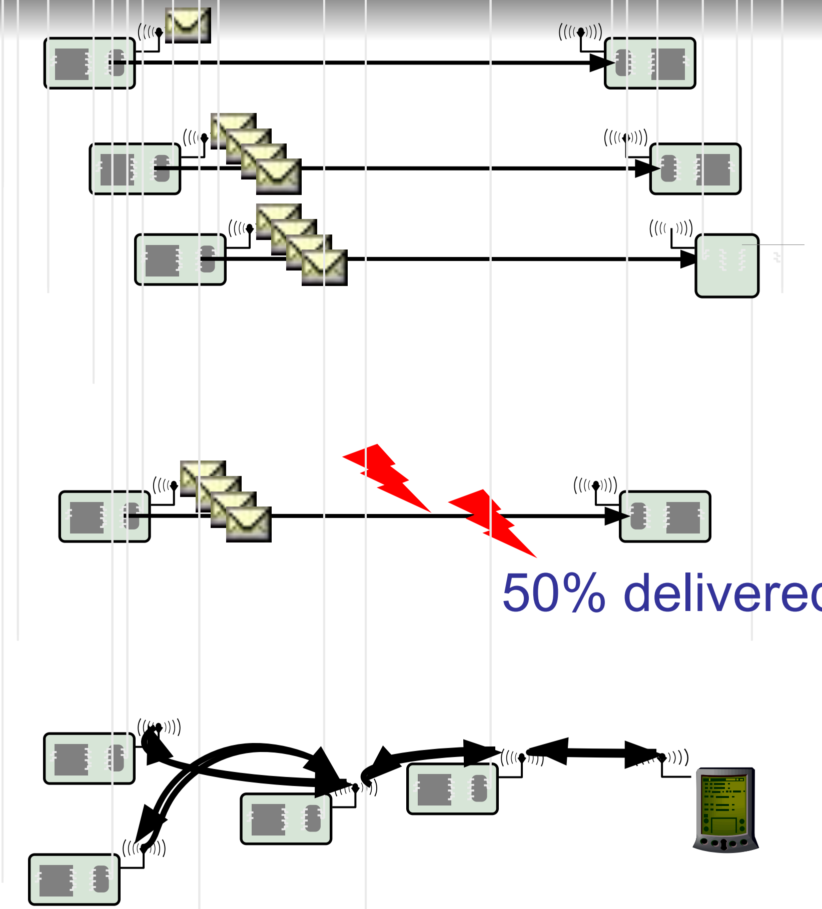
Dependability Constraints

- Wireless sensor networks (WSN) have **unique constraints** for dependable data delivery
  - Transmission errors over a wireless channel
  - Limited computational resources in a WSN node
  - Limited memory
  - Limited time (deadlines)
  - Limited dependability of individual nodes

- Standard mechanism: Redundancy
  - Redundancy in nodes, transmission
  - Forward and backward error recovery
  - Combinations of both are necessary

Dependable Data Transport

- Items to be delivered
  - Single packet
  - Block of packets
  - Stream of packets
- Level of guarantee
  - Guaranteed delivery
  - Stochastic delivery
- Involved entities
  - Sensor(s) to sink
  - Sink to sensors
  - Sensors to sensors



We Also Have Constraints...

- Energy
  - Send as few packets as possible
  - Send with low power -> high error rates
  - Avoid retransmissions
  - Balance energy consumption in network
- Processing power
  - Only simple FEC schemes
  - No complicated algorithms (coding)
- Memory
  - Store as little data as briefly as possible

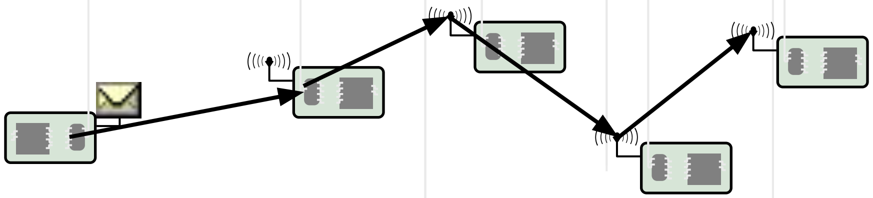
Delivering Single Packets

- What are the intended receivers?
  - A ***single receiver (unicast)***?
  - ***Multiple receivers (multicast)***?
    - In close vicinity? Spread out?
  - Mobile?

- Which routing structures are available?
  - Unicast routing along a ***single path***?
  - Routing with ***multiple paths*** between source/destination pairs?
  - No routing structure at all – rely on ***flooding/gossiping***?

Single Packet, Single Receiver, Single Path

- Single, multi-hop path is given by routing protocol



- Issues: Which node
  - Detects losses (using which indicators)?
  - Requests retransmissions?
  - Carries out retransmissions?

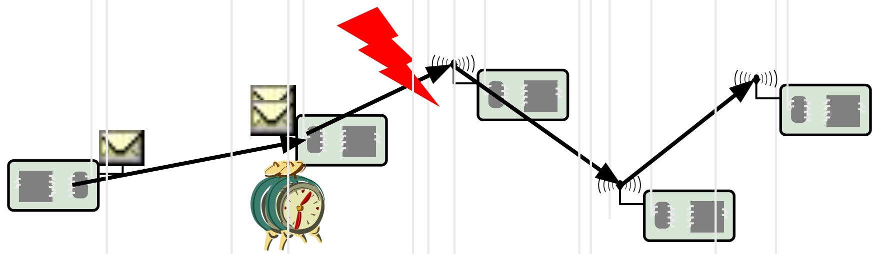
Single Packet, Single Receiver, Single Path

- Detecting loss of a *single packet*: Acknowledgements (ACK)

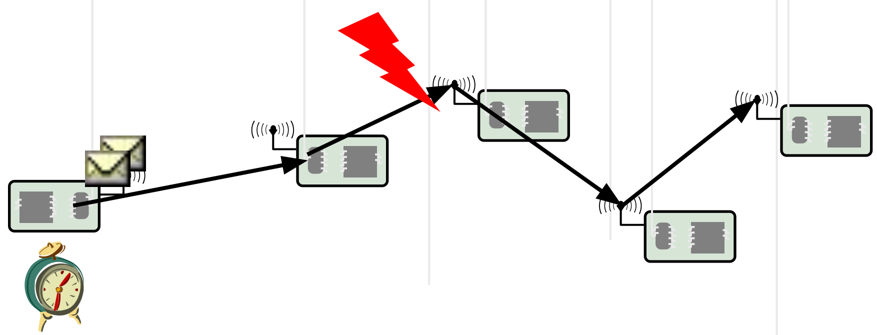
- Which node sends ACKs?
  - At each intermediate node, at MAC/link level
    - Usually accompanied by link layer retransmissions
    - Usually, only a bounded number of attempts
  - At the destination node
    - Transport layer retransmissions
    - Problem: Timer selection

Retransmissions

- For link layer acknowledgements: Neighboring node



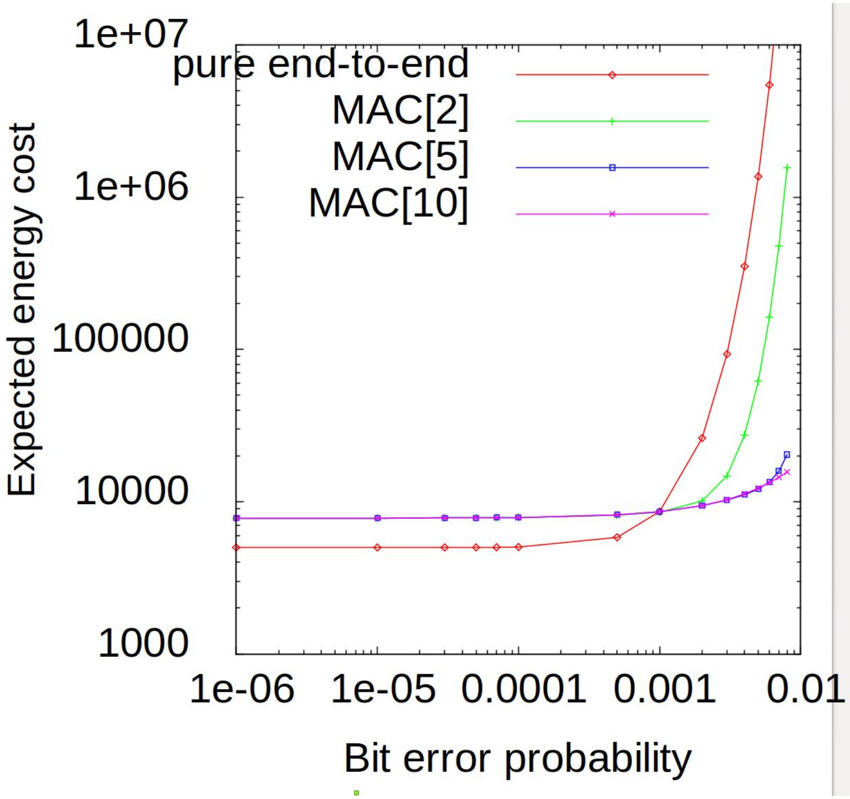
- For transport layer acknowledgements:
  - Source node -> end-to-end retransmissions



End-to-end vs. Link-layer Retransmission

- Scenario: Single packet, n hops from source to destination, BSC channel
- Transport-layer, end-to-end retransmission: Always
- Link-layer retransmissions: Vary number of maximum attempts
  - Drop packet if not successful within that limit

-> For good channels, use end-to-end scheme; else local retransmit



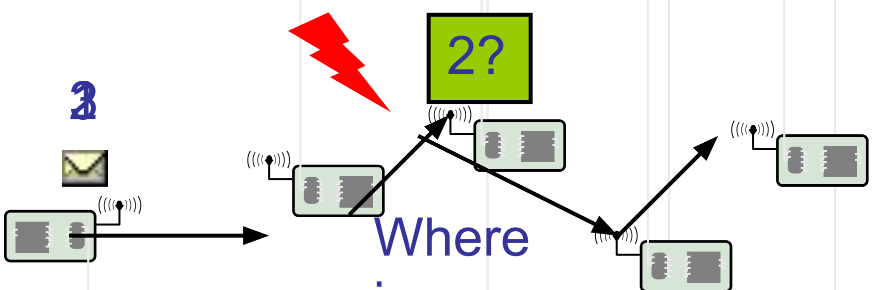
Example Schemes: HHR and HHRA

- *Hop-by-hop reliability (HHR)*
  - Idea: Locally improve probability of packet transmission, but do not use packet retransmission
  - Instead, simply repeat packet a few times – a repetition code
  - Choose number of repetitions per node such that resulting end-to-end delivery probability matches requirements
- *Hop-by-hop reliability with Acknowledgements (HHRA)*
  - Node sends a number of packets, but pauses after each packet to wait for acknowledgement
  - If received, abort further packet transmissions

Delivering Blocks of Packets

- Goal: Deliver large amounts of data
  - E.g., code update, large observations
  - Split data into several packets (reduce packet error rate)
  - Transfer this *block* of packets

- Main difference to single packet delivery: Gaps in sequence number can be detected and exploited
  - For example, by intermediate nodes sending NACKs



Where is packet 2?

- To answer NACK locally, intermediate nodes must cache packets
- Which packets? For how long?

Pump Slowly Fetch Quickly (PSFQ)

Pump-slowly, fetch-quickly (PSFQ): a reliable transport protocol for sensor networks
Wan, C.-Y.   Campbell, A.T.   Krishnamurthy, L.
Journal of Selected Areas in Communications, April 2005

- Goal: Distribute block of packets from one sender to multiple receivers (sink to sensors)
  - E.g., code update -> losses are not tolerable, delay not critical
  - Routing structure is assumed to be known

Pump Slowly Fetch Quickly (PSFQ)

- Basic operation
  - Source ***pumps*** data into network
    - Using broadcast, *large inter-packet gap time*
  - Intermediate nodes store packets, forward if in-sequence
  - Out-of-sequence: buffer, request missing packet(s) – ***fetch*** operation (a NACK)
    - Previous node resends missing packet -> **local recovery**
    - Assumption: packet is available <- no congestion, only channel errors

  -> Pumping is slow, fetching is quick

PSFQ Protocol Details

- How big an inter-packet gap?
  - Big enough to accommodate at least one, better several fetch operations
  - Probability that next packet arrives when the previous one has not yet been repaired should be small

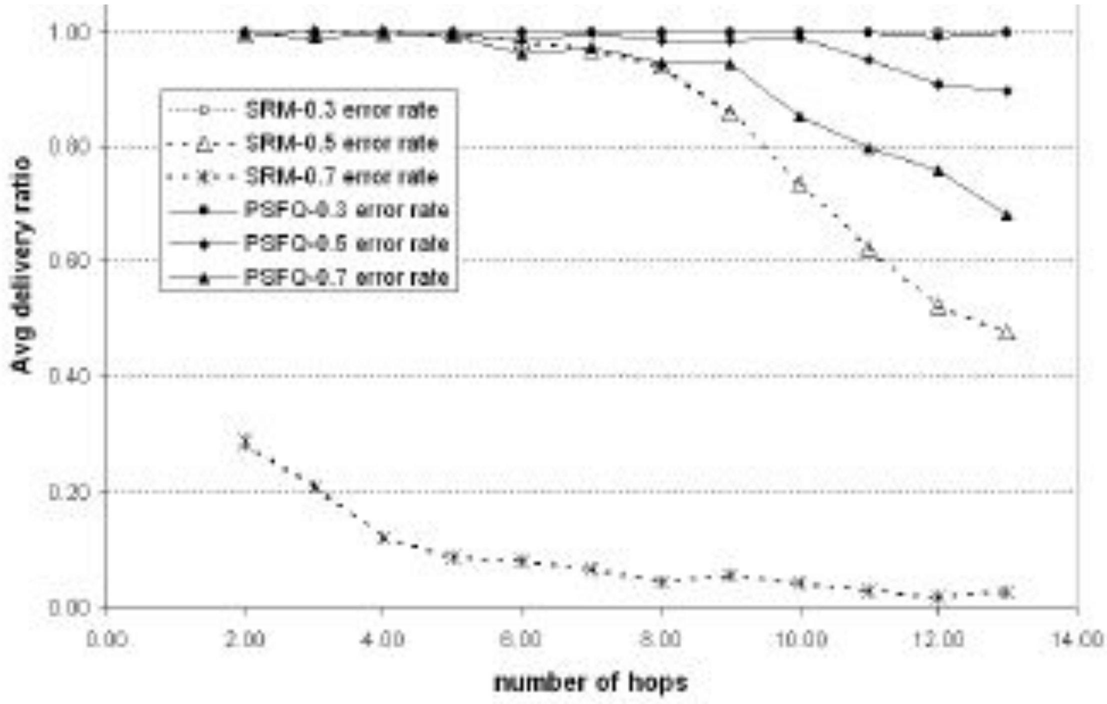
- Handle out-of-order packet?
  - Do not forward, fill the gap first by fetching -> avoid loss propagation

PSFQ protocol details (2)

- How to handle fetch requests (NACKs)?
  - Fetch request are broadcast, might arrive at multiple nodes
  - Nodes receiving NACK might themselves not have all requested packets
  - Use a **slotted resend mechanism** for requested packets – each one corresponds to a time slot

PSFQ Performance

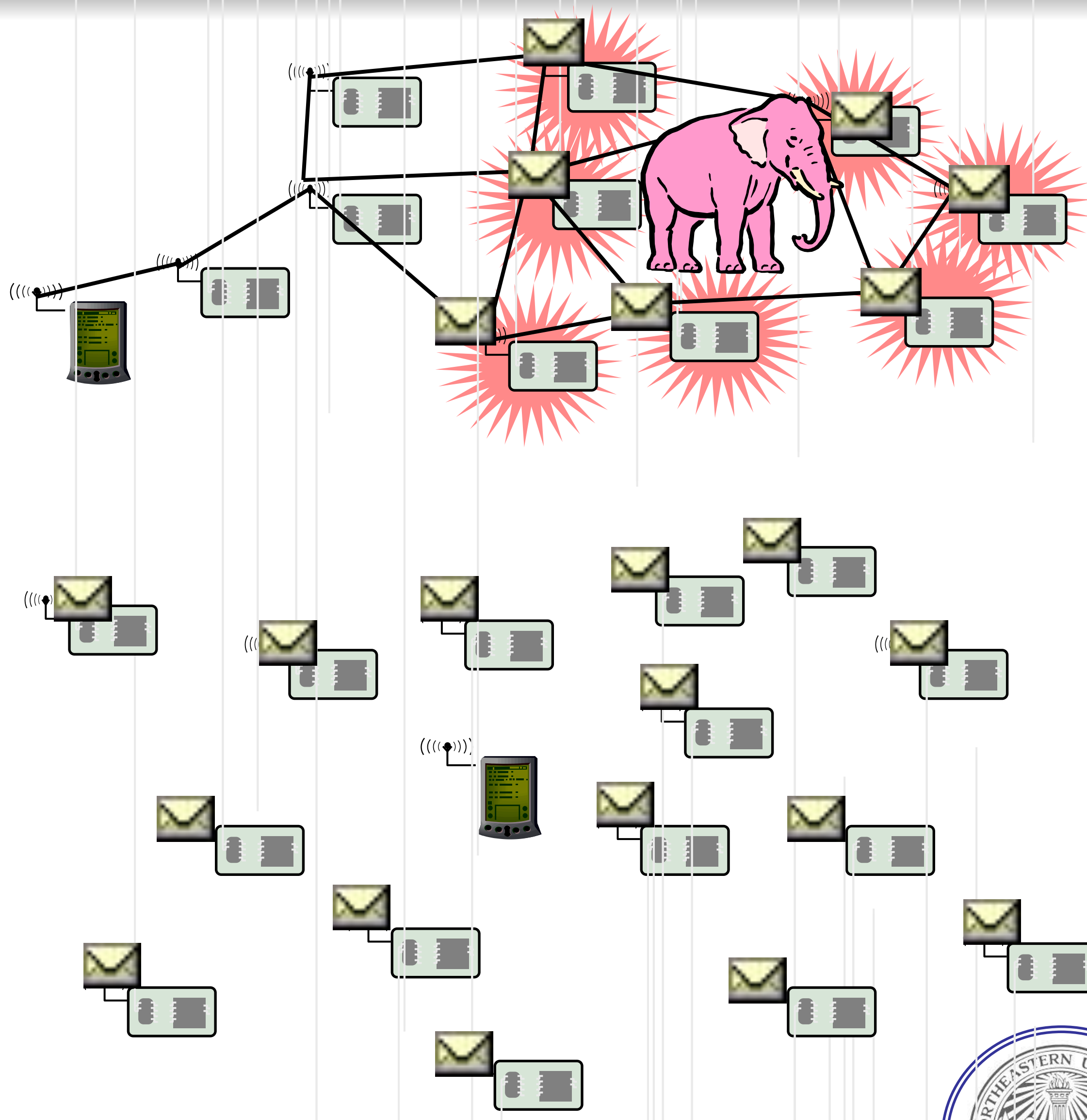
- Comparison case: Scalable Reliable Multicast (SRM)
  - Provides similar service
  - Main difference: in-sequence not enforced
- PSFQ works up to higher error rates



Congestion in WSNs

Streams of Packets

- When several sensors observe an event and try to periodically report it, congestion around event may occur
- When many sensors stream data to a sink, congestion around the sink may occur



Consequences of congestion

- Congestion can have surprising consequences
- More frequently reporting readings can reduce goodput and accuracy
  - Owing to increased packet loss
- Using more nodes can reduce network lifetime

Detecting Congestion

- ***Locally*** detect congestion
  - Intuition: Node is congested if its buffer fills up
  - Rule: “Congested = buffer level above threshold” is **overly simplistic**
  - Need to take **growth rate** into account as well
    - Occupancy not a good indicator when packets can be lost in the channel
- **Problematic:** Interaction with MAC
  - CSMA-type MACs: high channel utilization = congestion; easy to detect
  - TDMA-type MACs: high channel utilization not problematic for throughput; congestion more difficult to detect

Congestion Handling

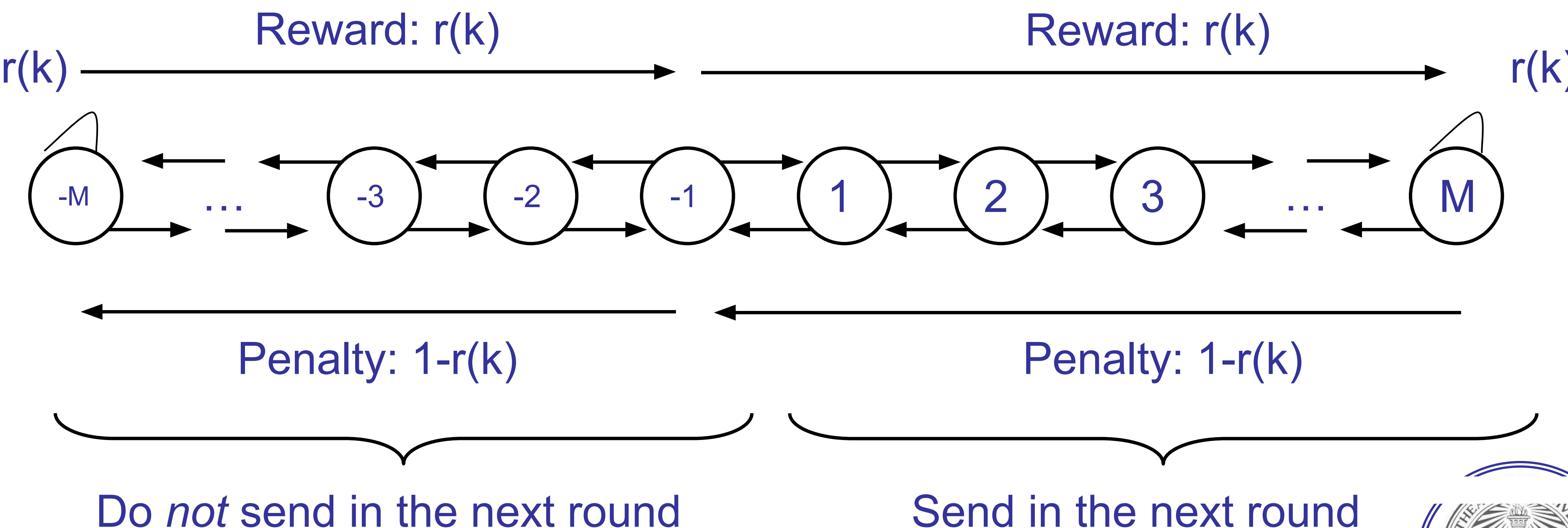
- Once congestion is (locally) detected, how to handle it?
- **Option 1:** Drop packets
  - No alternative when buffers overflow
  - Better: drop semantically less important packet
- **Option 2:** Control sending rate of individual node
  - Rate of locally generated packets
  - Rate of remote packets to be forwarded -> ***backpressure***
- **Option 3:** Control how many nodes are sending
- **Option 4:** Aggregation, in-network processing

Control How Many Nodes Are Sending

- **Scenario:** Nodes send at a given rate, cannot be controlled
- **Option:** Turn on or off nodes to avoid congestion, achieve desired target number of packets **k*** per round
  - If *total* number of nodes N known, easy: Simply send probability k*/N to all nodes; each node sends with this probability
- What to do if number of nodes N not known?

Gur Game

- N nodes, unaware of each other; 1 referee
- Referee, in each round:
  - Counts number k of packets (assumption: no packet loss)
  - Determines ***reward probability r(k)***, sends r(k) to all nodes
- Each player: rewards itself with probability r(k), penalizes with probability 1-r(k)
  - Rewards/penalties: Moves in finite state machine



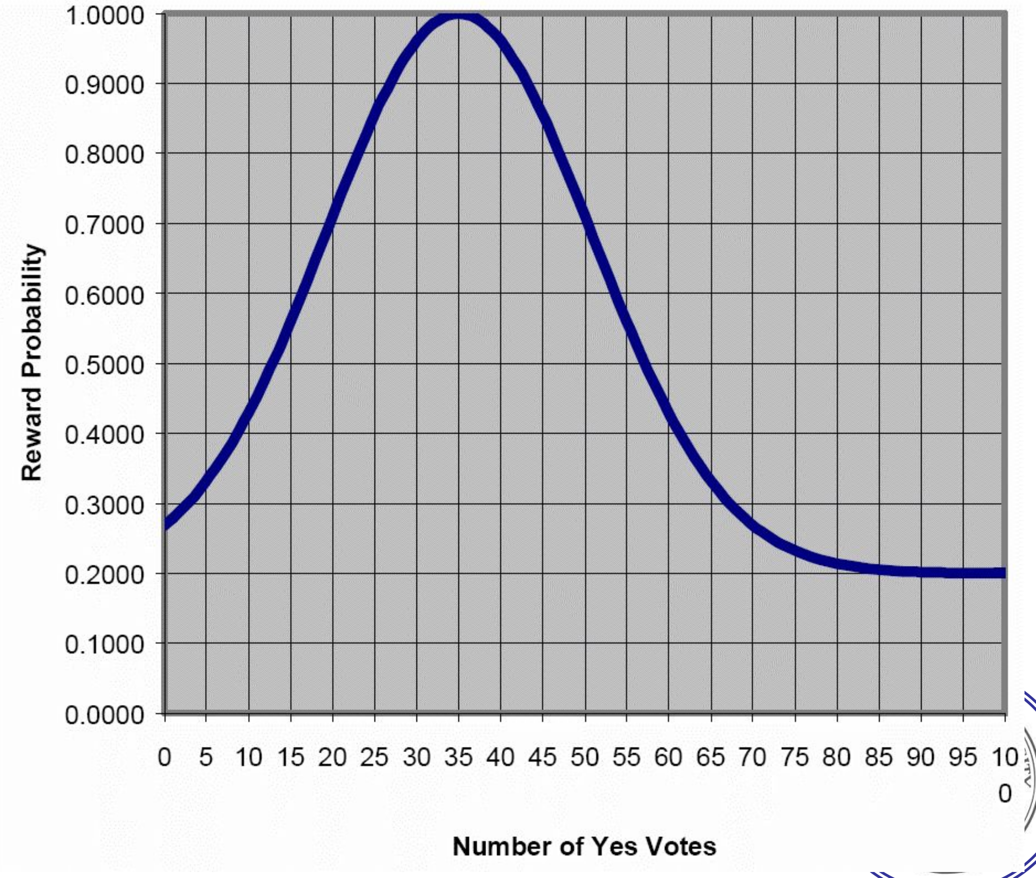
Gur Game: How To Choose r(k)?

- Intuition
  - When received number of packets **k** is close to **k***, the right number of nodes are sending
  - Thus, the right mixture of send/not send states is present

  -> Nodes should stay on the side where they are

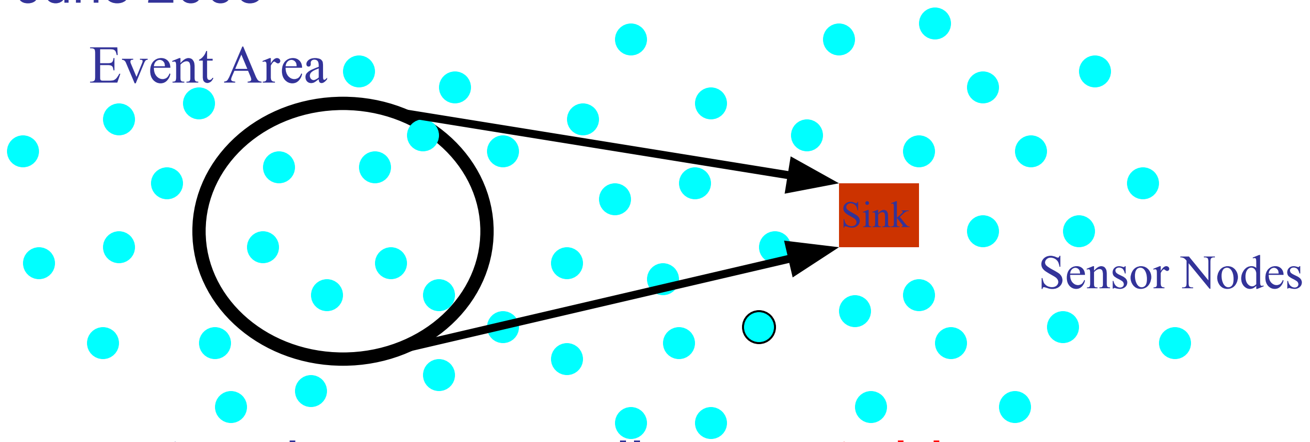
  -> Rewards should be high
- Formally
  - Reward function is maximal at k*
  - Example: See figure
  - **What are the problems c**

the Gur Game?



Event-to-Sink Reliability

O. B. Akan and I. F. Akyildiz, “Event-to-Sink Reliability”, IEEE/ACM Transactions on Networking, Oct. 2005 - Shorter version in Proc. of ACM MobiHoc’03, June 2003



- Sensor networks are usually event-driven
- Multiple correlated data flows from **event to sink**
- GOAL: To reliably detect/estimate event features based on the collective reports of several sensor nodes observing the event

→ Event-to-Sink collective reliability notion

→ EXPLORE SPATIAL CORRELATION !!!!

Event-to-Sink Reliability

- Sink decides about event features every i time units (decision intervals)
- DEFINITION 1: Observed Event Detection Reliability
  r_i is the number of data packets received in decision interval i at the sink

Event-to-Sink Reliability (2)

- DEFINITION 2: Desired Event Detection Reliability

  R is the number of packets required for reliable event detection→ Application specific and is known a-priori at the sink!!

  (If r_i > R, then the event is reliably detected. Else, appropriate actions must be taken to achieve R)

Event-to-Sink Reliability (3)

DEFINITION 3: Reporting Frequency Rate

f is the number of packets sent out per unit time by that node

DEFINITION 4: TRANSPORT PROBLEM

To configure the reporting frequency rate, **f**, of source nodes so as to achieve the required event detection reliability, **R**, at the sink with minimum resource utilization.

ESRT: Protocol Overview

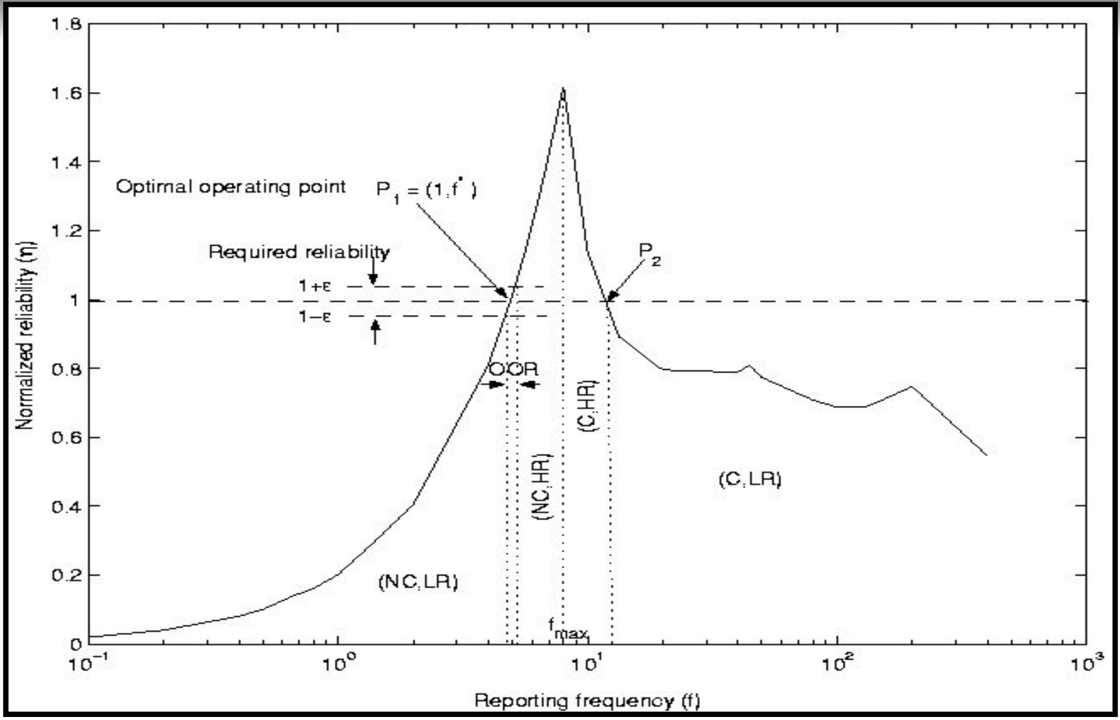
- Determine reporting frequency f to achieve desired reliability R with minimum resource utilization

- Source (Sensor Nodes):
  - Send data with reporting frequency f
  - Monitor buffer level and notify congestion to the sink

ESRT: Protocol Overview

- Sink:
  - Measures the observed event reliability r_i at the end of decision interval i
  - Normalized Reliability → $\eta_i = r_i / R$
  - Performs congestion decision based on feedback from sensor nodes (to determine $f \gtrless f_{max}$)
  - Updates f based on η_i and $f \gtrless f_{max}$ (congestion) to achieve desired event reliability R
  - Broadcasts the new reporting rate to all sensor nodes

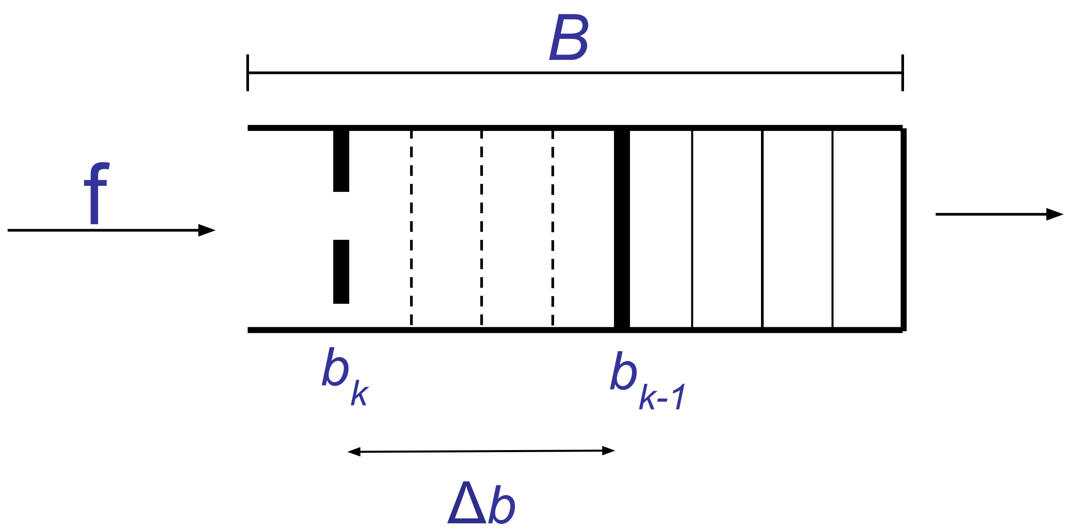
Network States



State	Description	Condition
(NC,LR)	(No Congestion, Low reliability)	$f < f_{max}$ and $\eta < 1 - \varepsilon$
(NC,HR)	(No Congestion, High reliability)	$f \leq f_{max}$ and $\eta > 1+ \varepsilon$
(C,HR)	(Congestion, High reliability)	$f > f_{max}$ and $\eta > 1$
(C,LR)	(Congestion, Low reliability)	$f > f_{max}$ and $\eta \leq 1$
OOR	Optimal Operating Region	$f < f_{max}$ and $1- \varepsilon <= \eta <= 1+ \varepsilon$

ESRT: Congestion Detection Mechanism

- Use local buffer level monitoring in sensor nodes



b_k : Buffer level at the end of reporting interval k

Δb : Buffer length increment

B : Buffer size

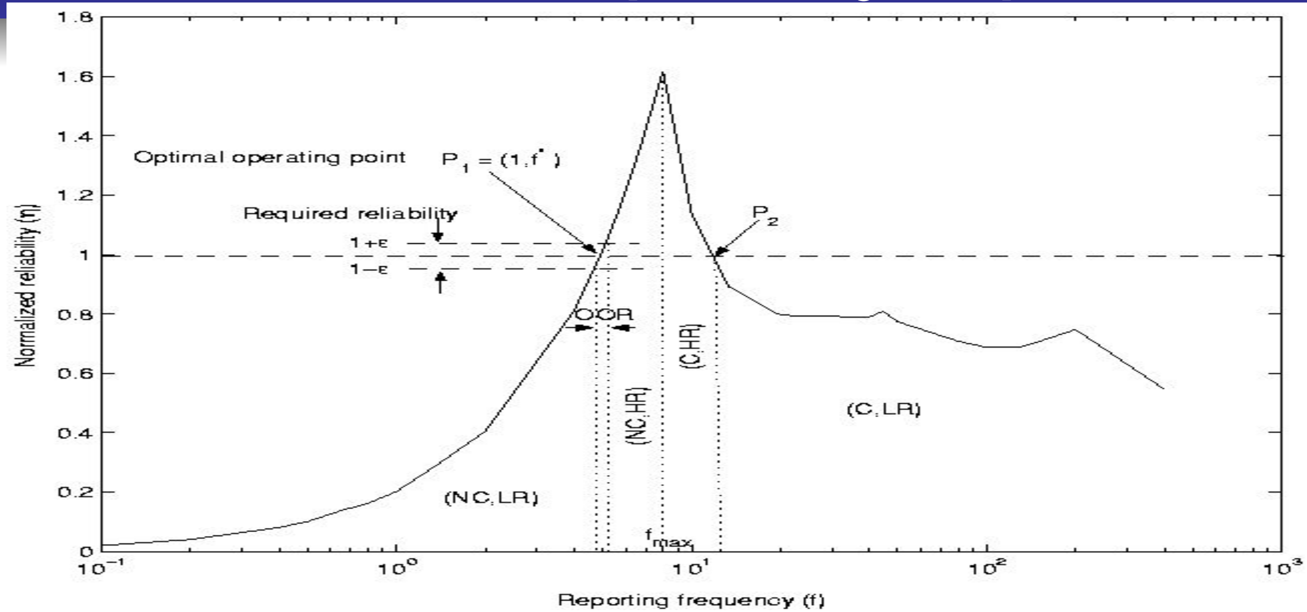
f : Reporting frequency

ESRT: Congestion Detection Mechanism

Mark Congestion Notification (*CN*) field in packet if congested, i.e., $b_k + \Delta b > B$

Event ID	CN (1 bit)	Desti-nation	Time Stamp	Payload	FEC

ESRT: Frequency Update



State	Frequency Update	Comments
(NC,LR)	$f_{i+1} = f_i / \eta_i$	Multiplicative increase, achieve desired reliability asap
(NC,HR)	$f_{i+1} = f_i/2\,(1 + 1/\eta_i)$	Conservative decrease, no compromise on reliability
(C,HR)	$f_{i+1} = f_i / \eta_i$	Aggressive decrease to state (NC,HR)
(C,LR)	$f_{i+1} = f_i^{(\eta_i/k)}$	Exponential decrease, relieve congestion asap
OOR	$f_{i+1} = f_i$	Unchanged

Performance Evaluation

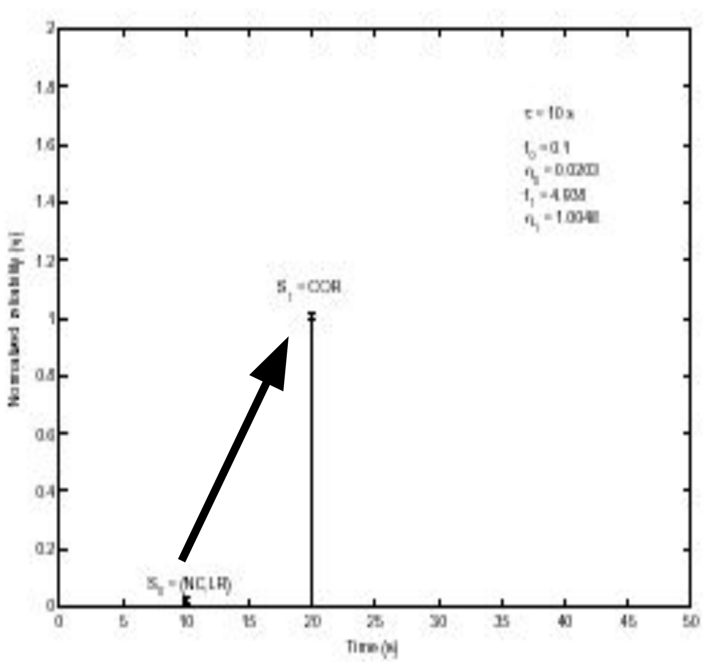
- Starting from no congestion, high reliability and with linear reliability behavior when the network is not congested, the network state remains unchanged until ESRT converges

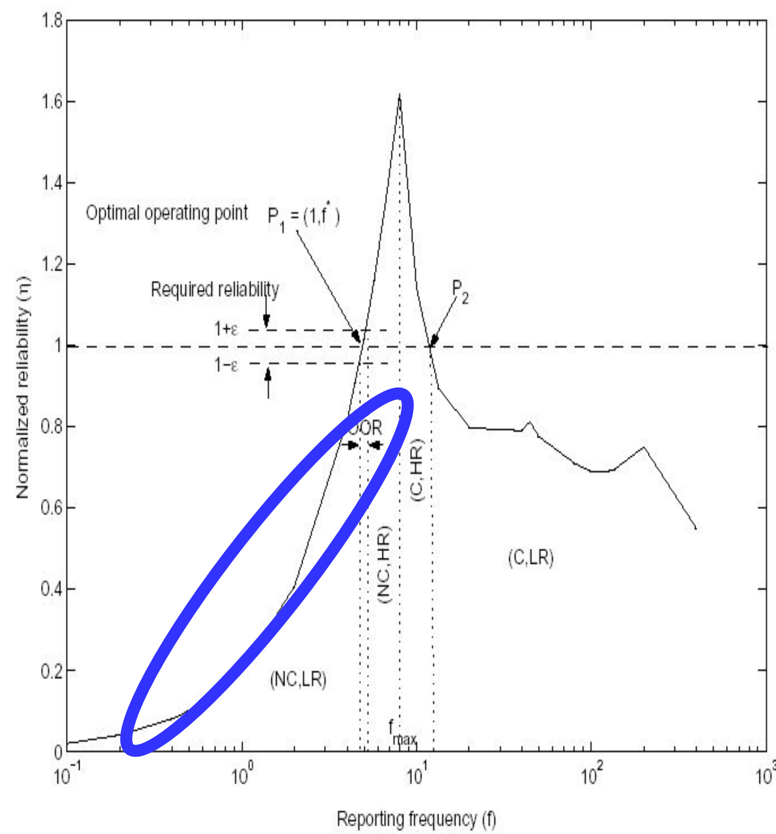
- Starting from no congestion, high reliability, and with linear reliability behavior when the network is congested, ESRT converges to optimum operating range

- With linear reliability behavior when the network is not congested, the network state transition from congestion, high reliability to no congestion, low reliability.

Performance Results

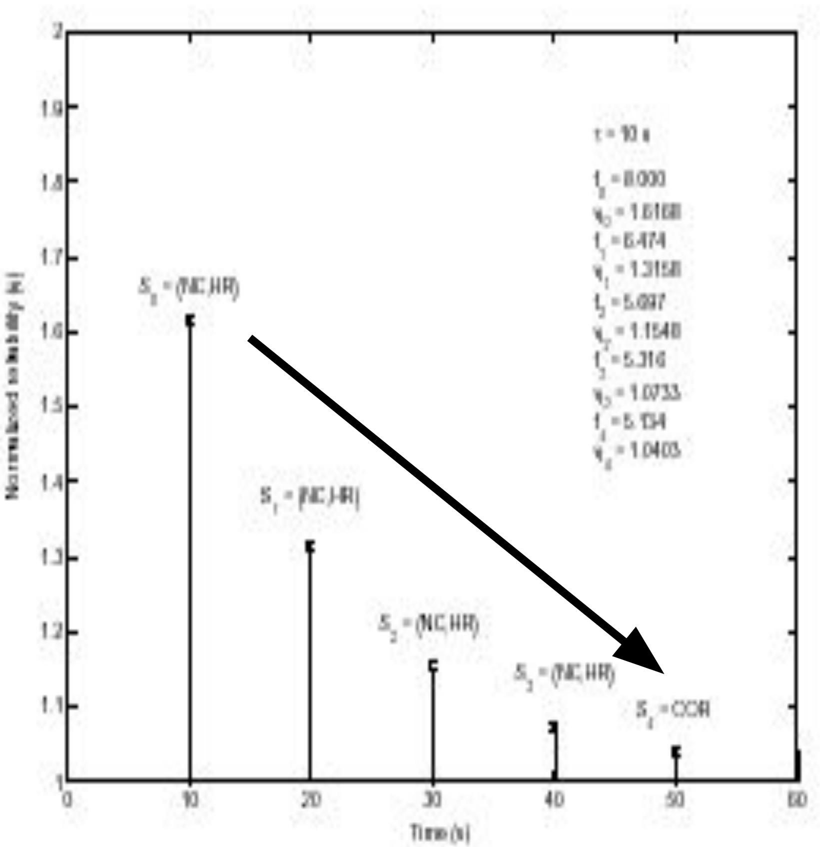
Starting with no congestion and low reliability:

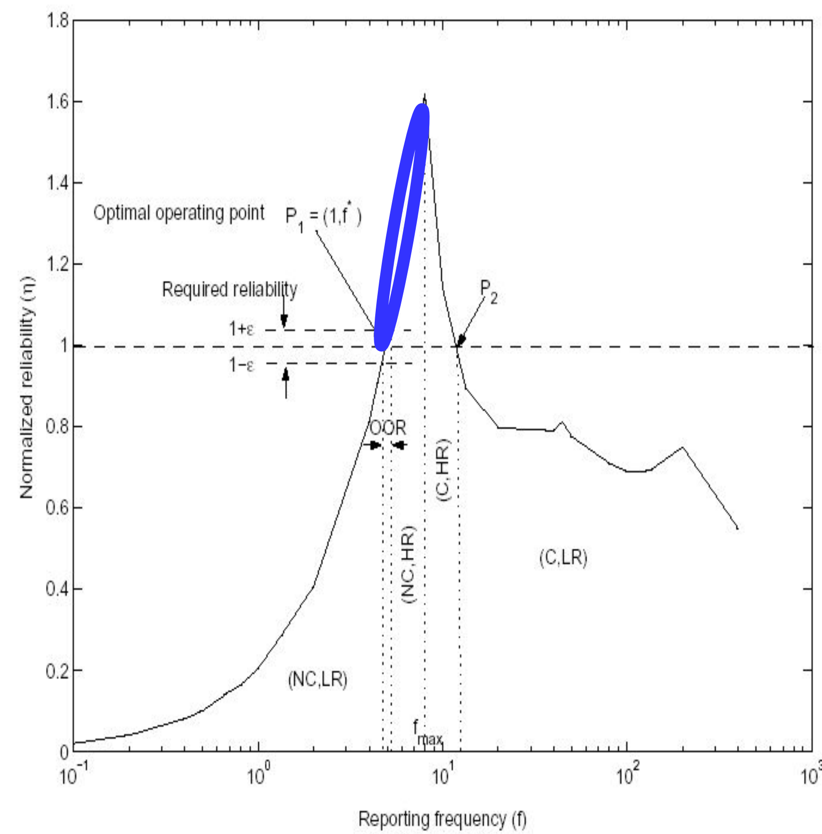




Performance Results

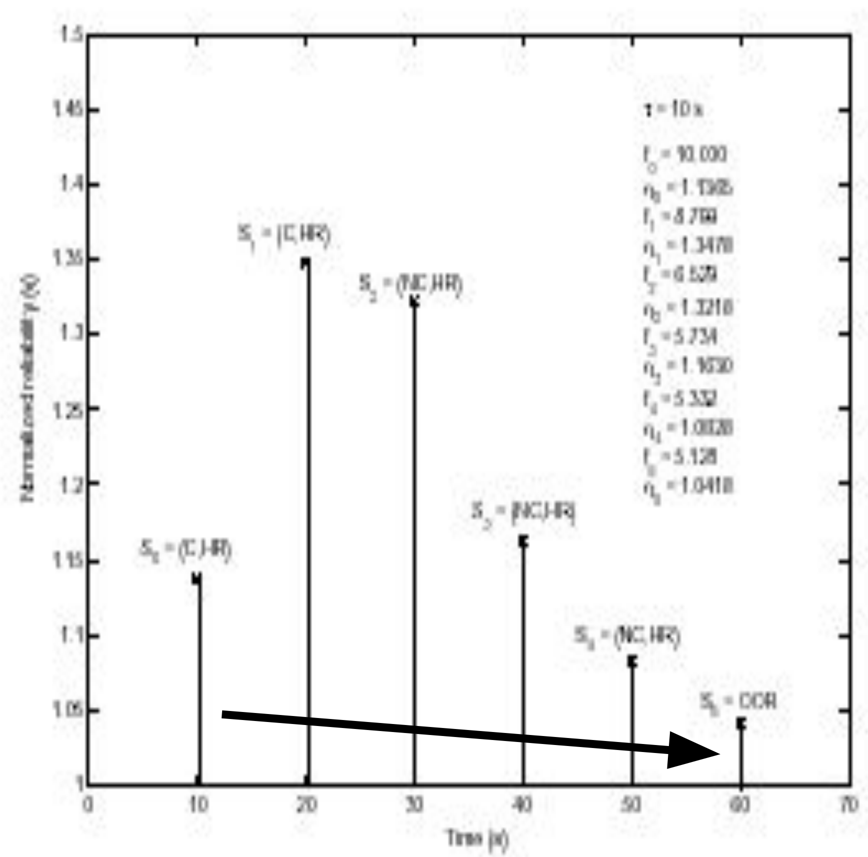
Starting with no congestion and high reliability:

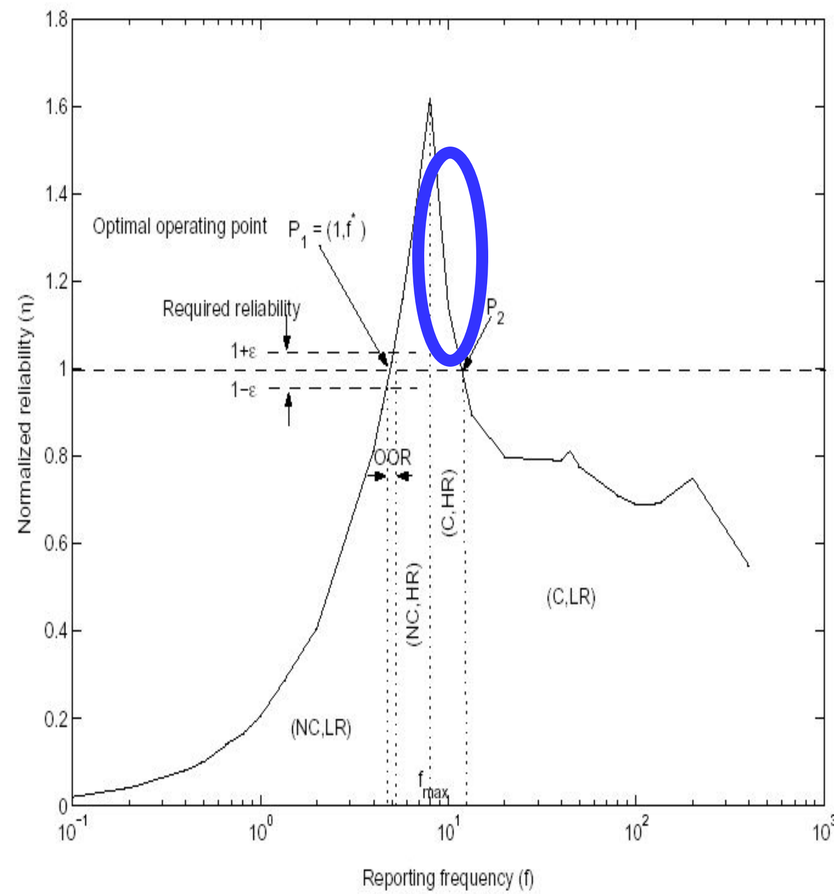




Performance Results

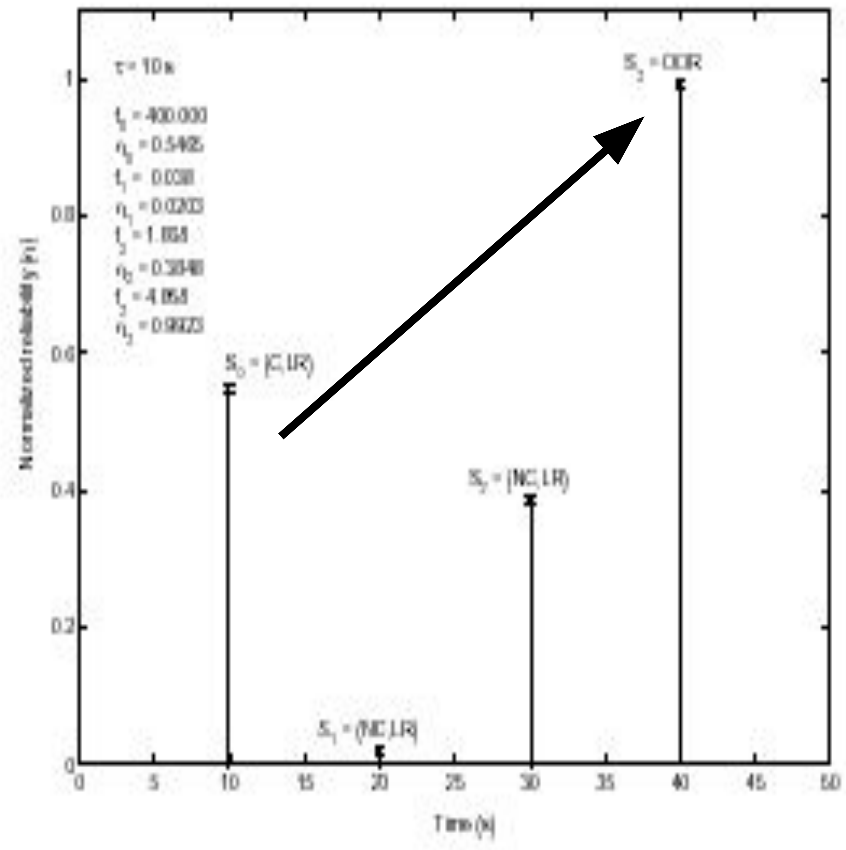
Starting with congestion and high reliability:

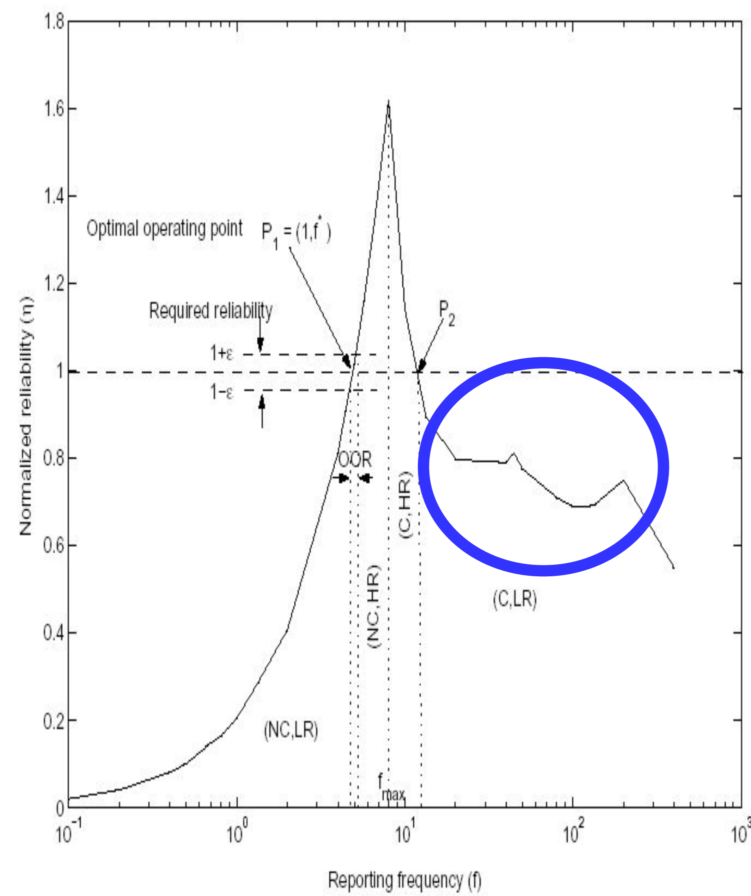




Performance Results

Starting with congestion and low reliability:





Conclusion

- Transport protocols have considerable impact on the service rendered by a wireless sensor networks
- Various facets – no "one size fits all" solution in sight
- Still there are relatively unexplored areas

- Items not covered
  - Relation to coverage issues
  - TCP in WSN? Gateways?
  - Aggregation? In-network processing?

Conclusion

- What are the pros and cons of ESRT?

Think-Share!