Lecture

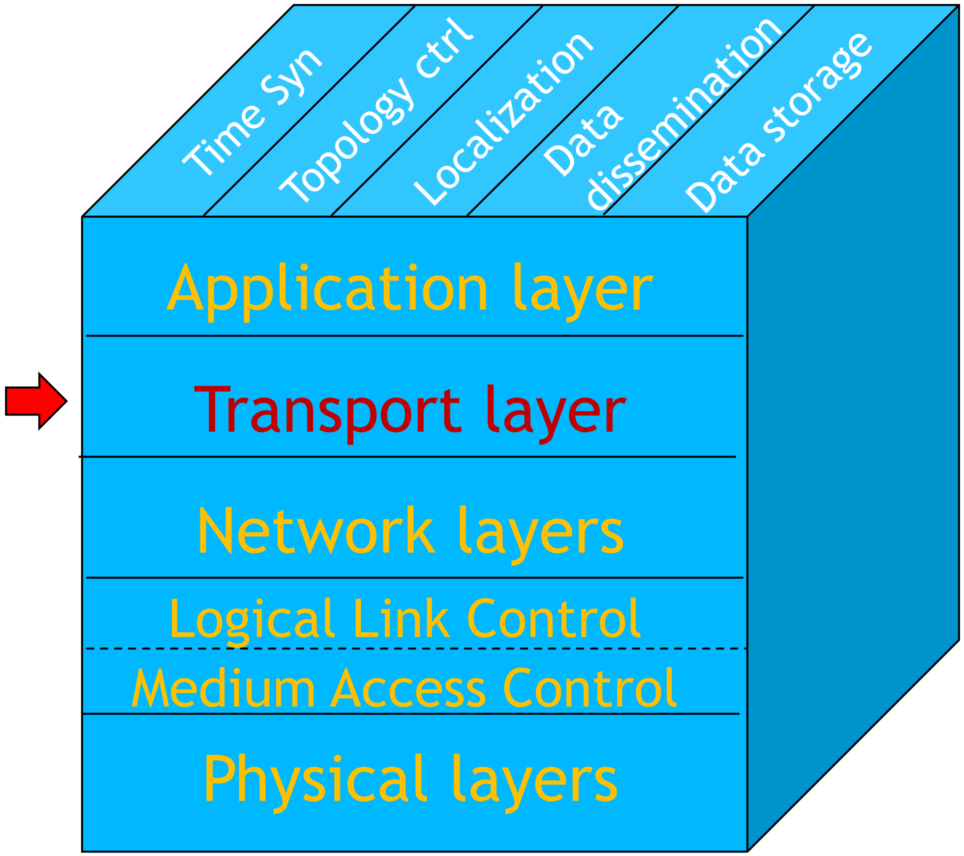
Transport Layer Protocols of WSN

Qi Zhang

(Email: qz@eng.au.dk; Edison 314)

AARHUS SCHOOL OF ENGINEERING

Relevant topics in WSN



AARHUS SCHOOL OF ENGINEERING

Outline

- **Reliable data transport basics**
- Single packets delivery
- Blocks of packets delivery
- Congestion control and rate control

Transport Protocol

- What is expected out of a transport protocol?
  - End-to-end **Reliability/loss recovery**,
  - End-to-end **congestion control**.

- Although the two tasks are supposed to be implemented in the same transport protocol, some transport schemes only focus on one of them (either reliability or congestion issues).
  - It is not a complete transport protocol if only one of them is achieved.

Motivation

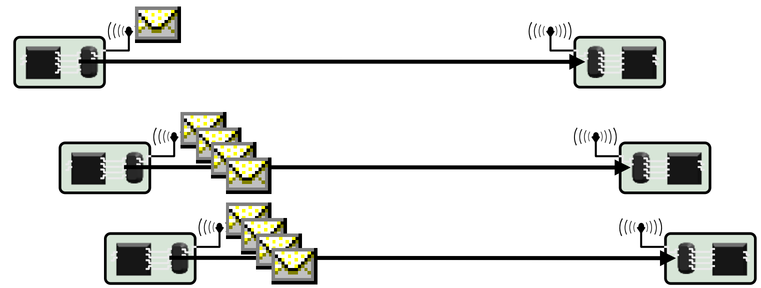
- Why can't we use the existing TCP protocols?
  - TCP has big packet overhead
    - 20 bytes header for congestion control
  - TCP requires perfect reliability and accept no loss.
  - TCP is not aware of the redundancy and correlation among the sensed data.
  - TCP connects two end nodes and the intermediate nodes just forward blocks of bits and don't care the content
  - etc.

Constraints in WSN for Reliable Data Delivery

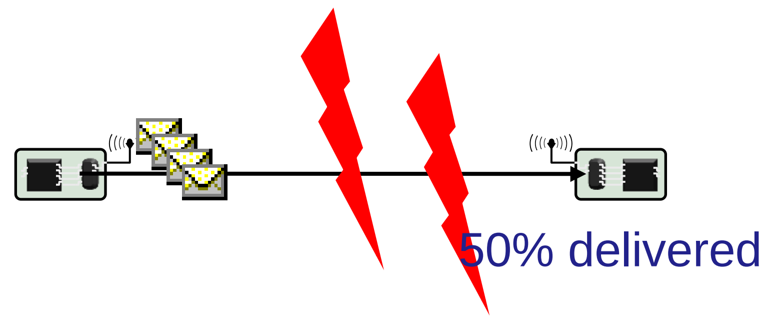
- Wireless sensor networks (WSN) have unique constraints for reliable data delivery
  - High error rate over a wireless channel
  - Limited energy of sensor node
    - Send as few packets as possible
    - Send with low power => high error rates
    - Avoid retransmissions
  - Limited computational resources in a sensor node
    - Only simple FEC schemes
    - No complicated algorithms (coding)
  - Limited memory
    - Store/cache as little data as possible
  - Limited reliability of individual nodes
  - etc

Reliable data transport - context

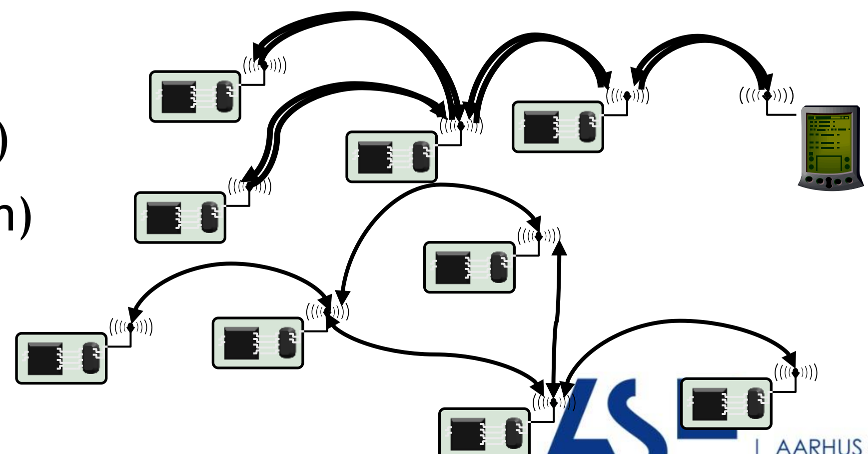
- Items to be delivered
  - Single packet
  - Block of packets
  - Stream of packets



- Level of guarantee
  - Guaranteed delivery
  - Stochastic delivery



- Involved entities
  - Sensor(s) to sink (Upstream)
  - Sink to sensors (Downstream)
  - Sensors to sensors



AARHUS SCHOOL OF ENGINEERING

Reliability

- Packet reliability
  - Applications are loss-sensitive and require the successful transmission of all packets or a certain percent
- Event reliability
  - Applications require only successful event detection, not successful transmission of each packet

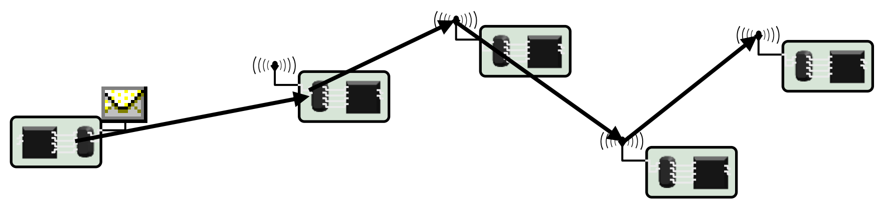
AARHUS SCHOOL OF ENGINEERING

Outline

- Reliable data transport basics
- **Single packets delivery**
- Blocks of packets delivery
- Congestion control and rate control

Single packet to single receiver over single path

- Single, multi-hop path is giving by some routing protocol



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

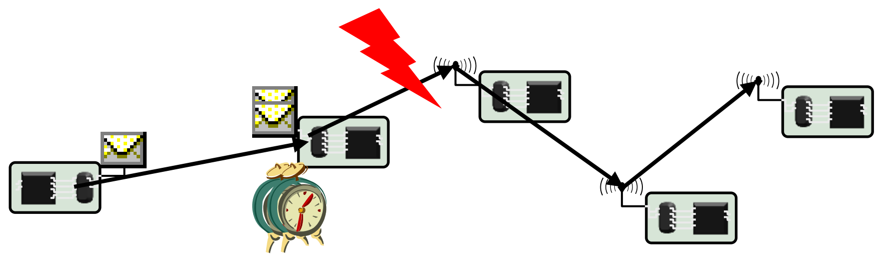
Detecting & signaling losses in single packet delivery

- Detecting loss of a single packet:
  Only positive acknowledgements (ACK) feasible
  - Negative ack (NACK) is not an option – receiver usually does not know a packet should have arrived, has no incentive to send a NACK

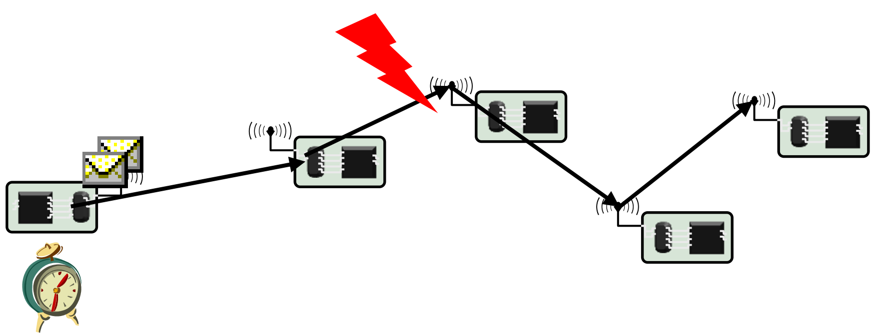
- Which node sends ACKs (avoiding retransmissions)?
  - At each intermediate node, at MAC/link level
    - Usually accompanied by link layer retransmissions
    - Usually, only a bounded number of attempts
  - At the destination node
    - Transport layer retransmissions
    - Problem: Timer selection

Carrying out retransmissions

- For link layer acknowledgements: Neighboring node



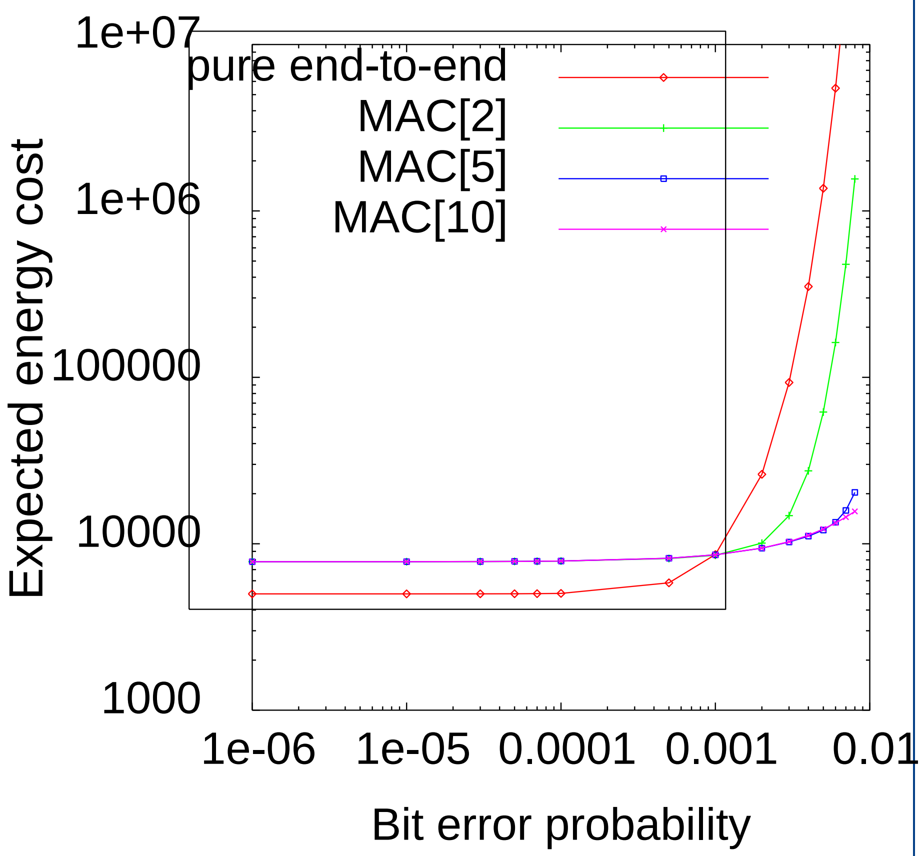
- For transport layer acknowledgements:
  - Source node for end-to-end retransmissions



AARHUS SCHOOL OF ENGINEERING

Tradeoff: End-to-end vs. link-layer retransmission

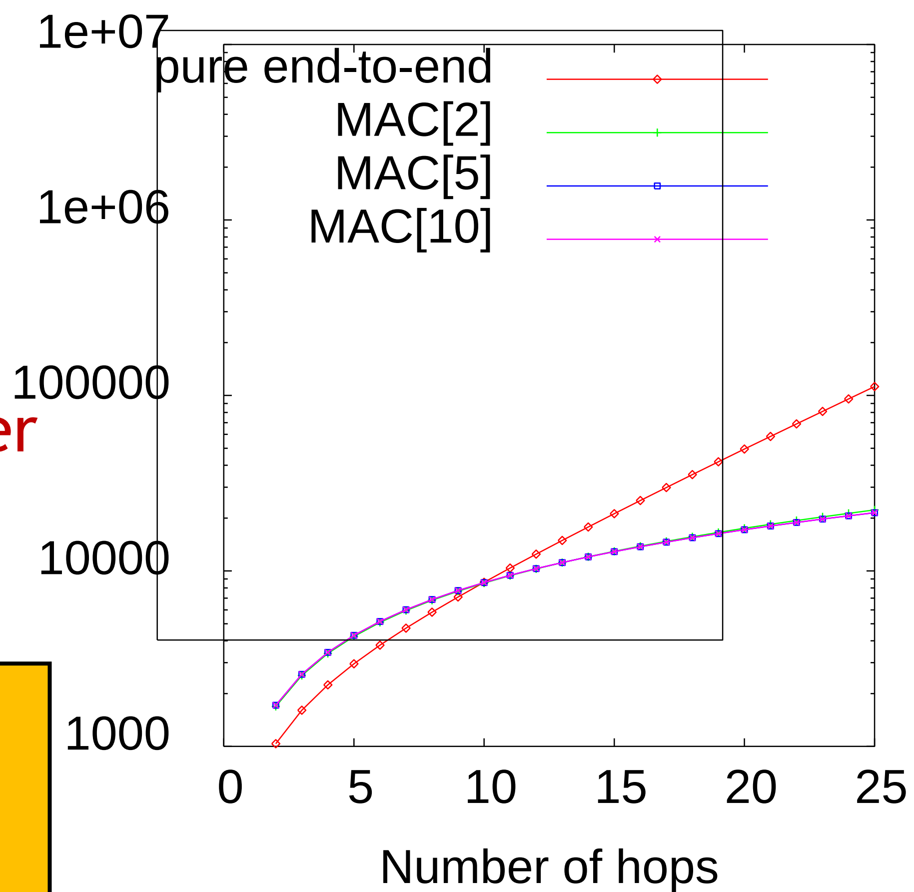
- Scenario: Single packet, *n* hops from source to destination, BSC channel
- Transport-layer, end-to-end retransmission: Always
- Link-layer retransmissions: varying the number of maximum attempts
  - Drop packet if not successful within that limit



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

Tradeoff: End-to-end vs. link-layer retransmission

- Same scenario, varying number of hops
  - Fixed BER=0.001 of BSC channel
- Conclusion: Use link-layer retransmissions only for longer routes



In both figures, difference between maximum link-layer retries schemes is small. Why?

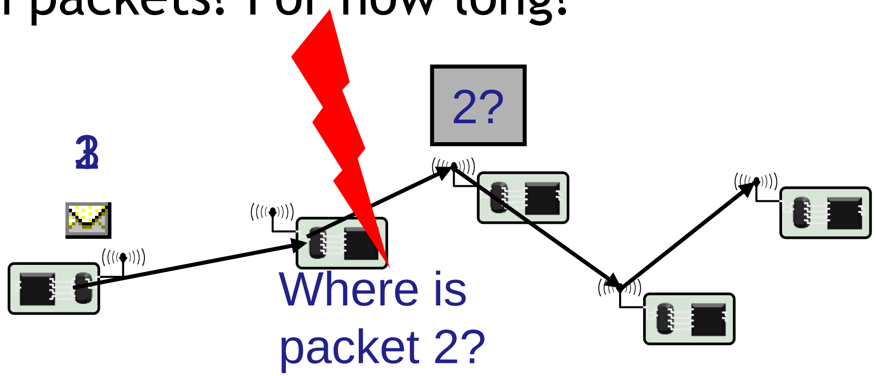
AARHUS SCHOOL OF ENGINEERING

Outline

- Reliable data transport basics
- Single packets delivery
- **Blocks of packets delivery**
  - **PSFQ**
- Congestion control and rate control

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



AARHUS SCHOOL OF ENGINEERING

PSFQ - Pump Slowly Fetch Quickly

- Goal: Distribute block of packets to from one sender to multiple receivers (*i.e. **sink to sensors***)
  - Designed for applications that losses are not tolerable, delay not critical
    - e.g., disseminating a program image from sink to sensors (i.e., re-tasking of a group of sensors)
  - Routing structure is assumed to be known
- Basic idea: minimize the loss recovery by localized recovery (hop-by-hop error recovery)
  - WSN usually operates in harsh radio environment and reply on multi-hop forwarding
  - Hop-by-hop recovery can eliminate error accumulation

AARHUS SCHOOL OF ENGINEERING

PSFQ - Pump Slowly Fetch Quickly

- Basic operation:
  - User node **pumps** data into network (slow injection of packets into the 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: message loss in wsn mainly occurs due to poor link quality triggered transmission errors,
      - i.e. packet is available => no congestion, only channel errors

- Pumping is slow, fetching is quick

PSFQ - Pump Slowly Fetch Quickly

- PSFQ has three functions:
  - **Message Relaying (PUMP Operation)**
  - **Relay initiated error recovery (FETCH Operation)**
  - Selective status reporting (REPORT Operation)

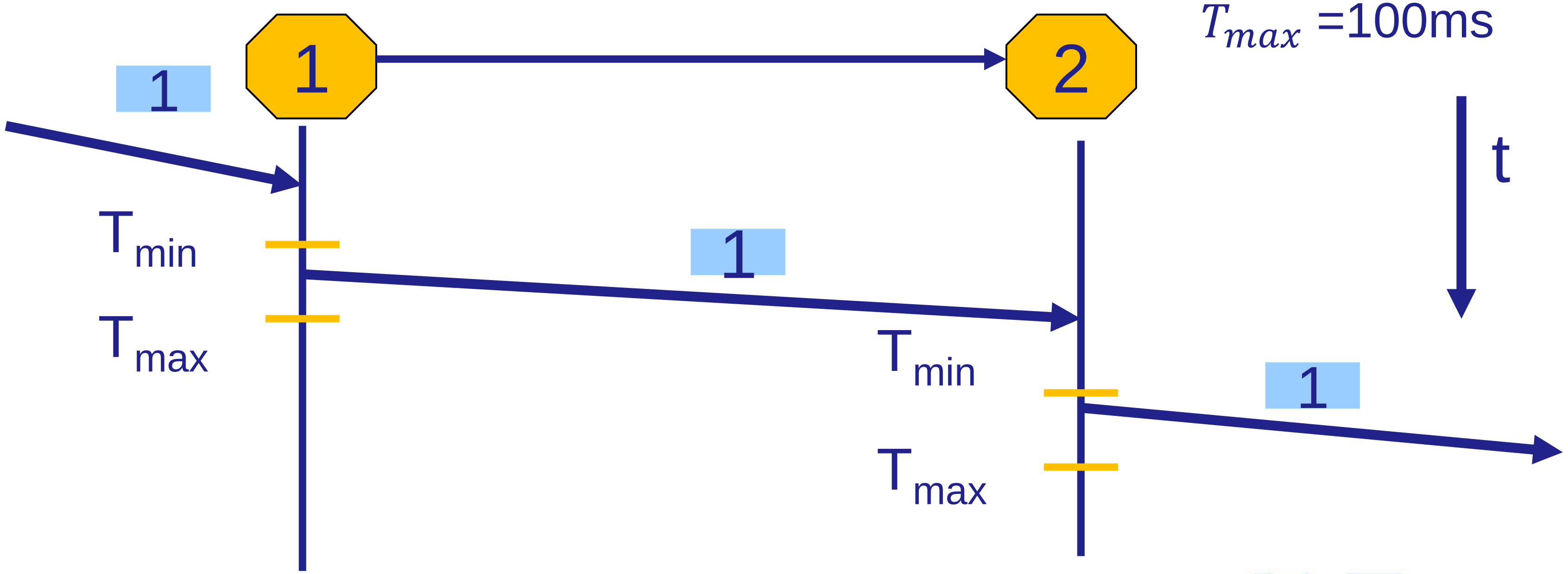
Pump Operation

- A user node broadcasts a packet to its neighbors every T_{min} until all the data fragments have been sent out
- Neighbors who receive the packet check against their local cache discarding any duplicates
- If it is just a new message the packet is buffered and the Time-To-Live (TTL) field in the header is decremented by 1
- If TTL > 0 after being decremented, and there is no gap in the packet sequence numbers, the packet is scheduled to be forwarded.
- However, the packets are delayed for a random period between T_{min} and T_{max} , and then forwarded.
  - Random delay:
    - Allows a downstream node to recover missing packets before the next packet arrives from an upstream node
    - Also allows reducing the number of redundant broadcasts of the same packet by neighbors

Pump Operation

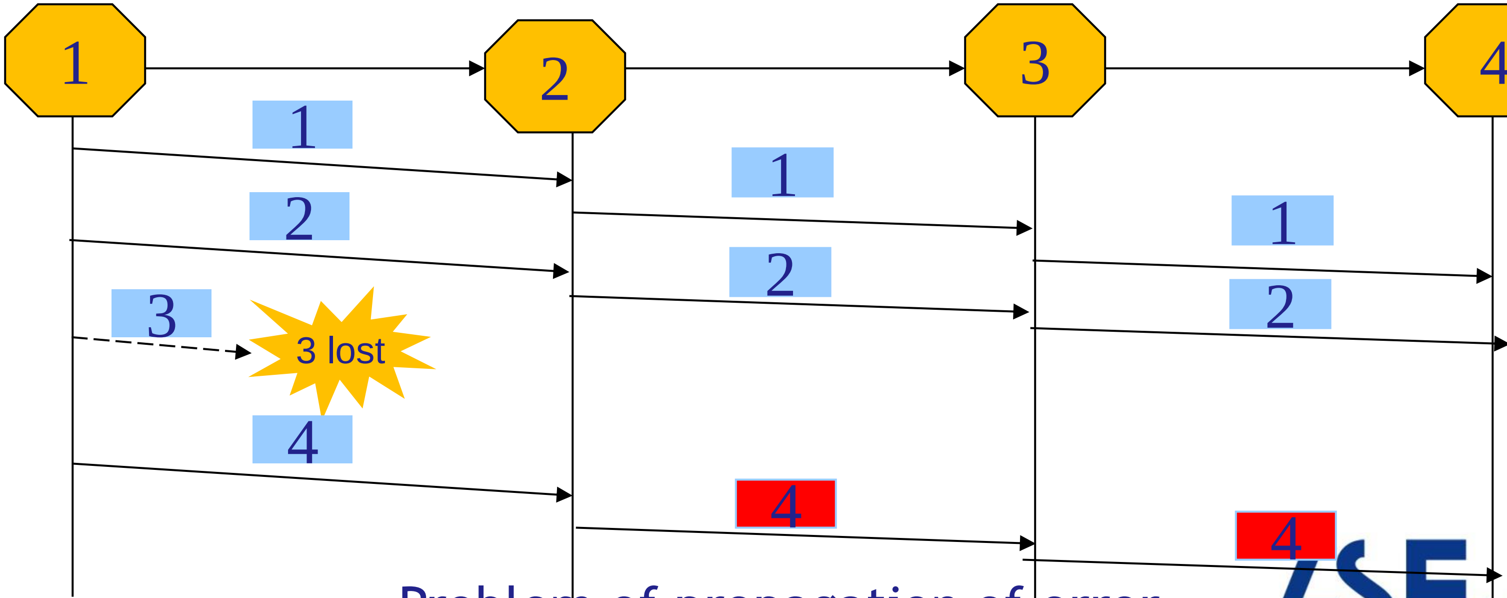
- If *not duplicate* and *in-order* and *TTL not 0*, cache and schedule for forwarding at time T_x
  - where (T_{min} < T_x < T_{max})

e.g., T_{min} =50ms
T_{max} =100ms



How to handle out-of-order packet?

- Aggressive hop-by-hop recovery in case of packet losses
  - Each intermediate node is required to create and maintain a data cache to be used for local loss recovery and in-sequence data delivery ONLY.
  - Avoid loss propagation, because propagation of loss event could cause a serious energy waste
  - i.e. node 2 should not relay msg #4 until it has recovered msg #3

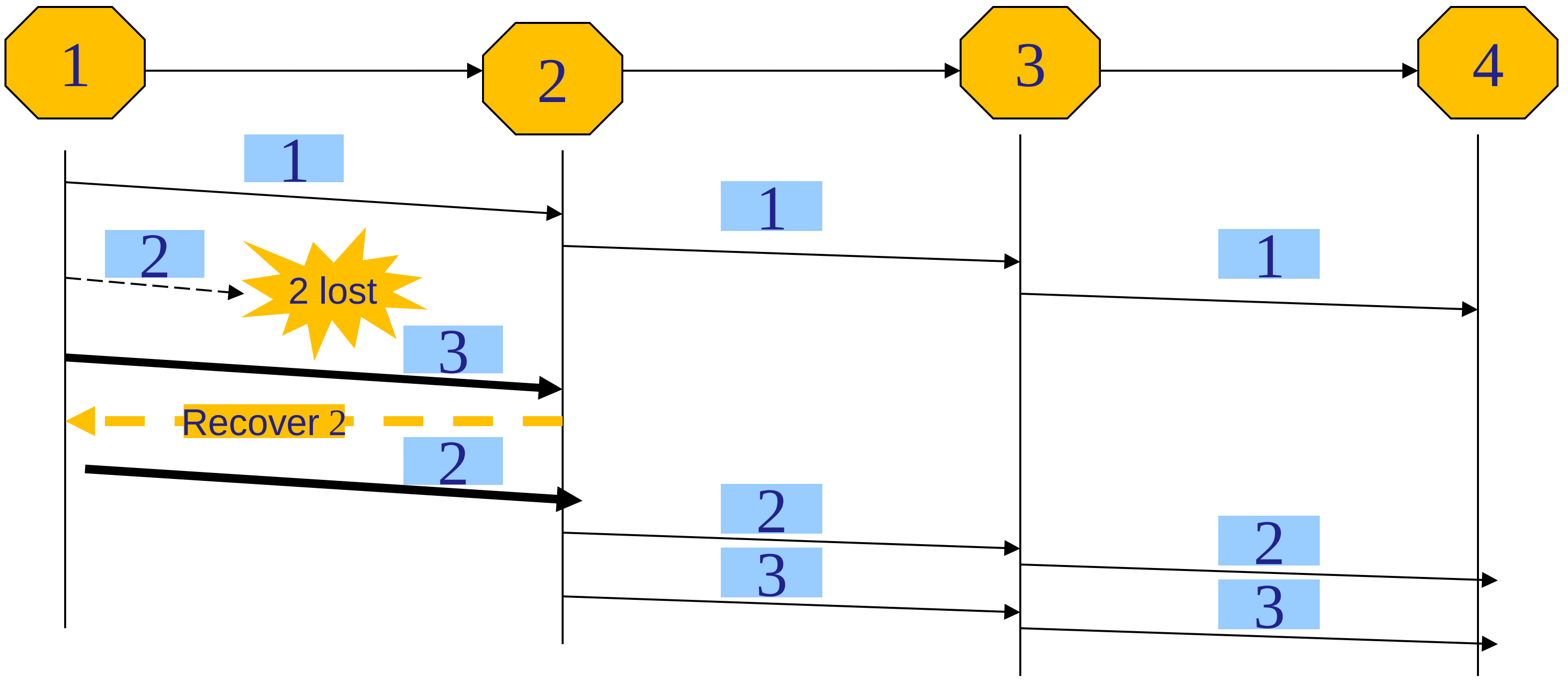


Problem of propagation of error

Fetch Operation

- A node goes into fetch mode when a sequence number gap is detected
- A node aggressively sends out NACK messages to its immediate upstream neighbors to request missing packets
- If no reply is heard or only a partial set of missing packets are recovered within a period T_r (~20ms) (T_r < T_{max} (~100ms)) then the node will resend the NACK every T_r interval until all packets are recovered.
- Since it is very likely that consecutive packets are lost because of fading conditions, PSFQ aggregates losses such that the fetch operation deal with a "window" of lost packets

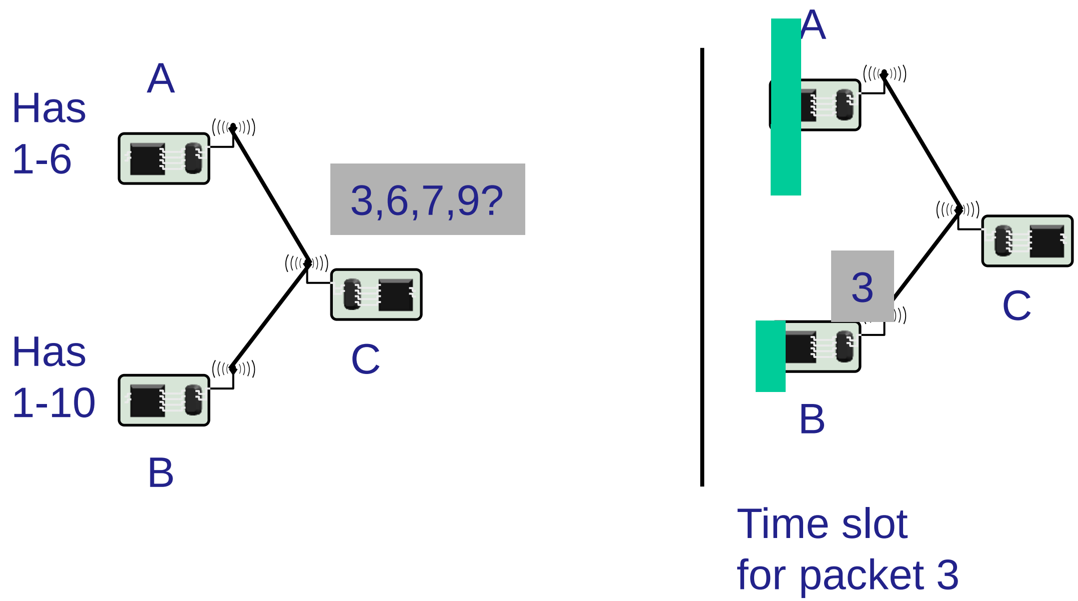
Recovery from Errors Fetch Quickly Mode



AARHUS SCHOOL OF ENGINEERING

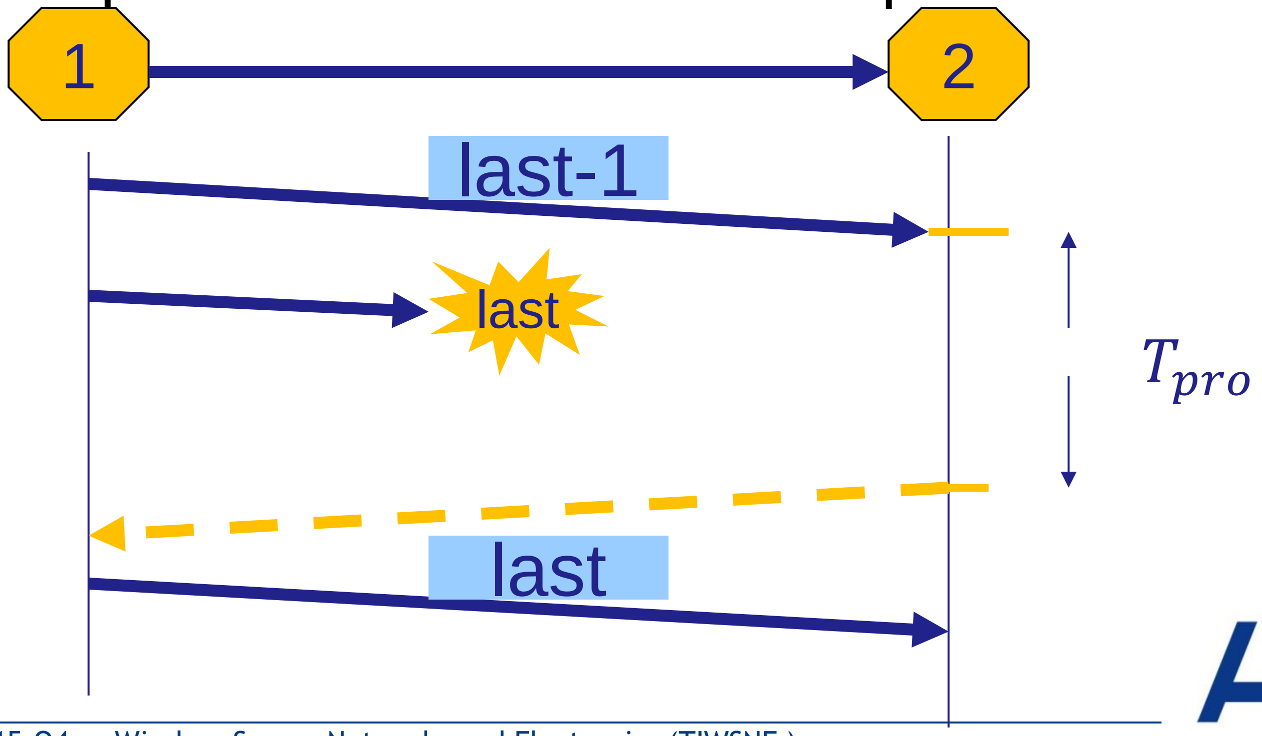
How to handle fetch requests (NACKs)?

- Fetch requests are broadcast, might arrive at multiple nodes
- Nodes receiving NACK might not have all the requested packets, PSFQ allows different segments of the loss window to be recovered from different neighbors
- A node that receives a NACK message checks the loss window field against its cache.
  - If found, to reduce redundant retransmission of the same segment, the segment is scheduled for transmission at random time between (0, T_r)
  - Neighbors will cancel retransmission if a reply to the same NACK is overheard.
- NACK messages usually should not be propagated to upstream neighbors to avoid message implosion


A
Has
1-6
3,6,7,9?
C
Has
1-10
B
A
3
C
B
Time slot
for packet 3

Proactive Fetch Mode

- To take care of situations such as when the last packet of a message is lost.
- The node sends a NACK for the next or remaining packets if the last packet has not been received and no new packet is delivered after a period of time T_{pro}



Proactive Fetch Mode

- Choice $T_{pro} = \alpha(S_{max} - S_{last})T_{max}$

  where

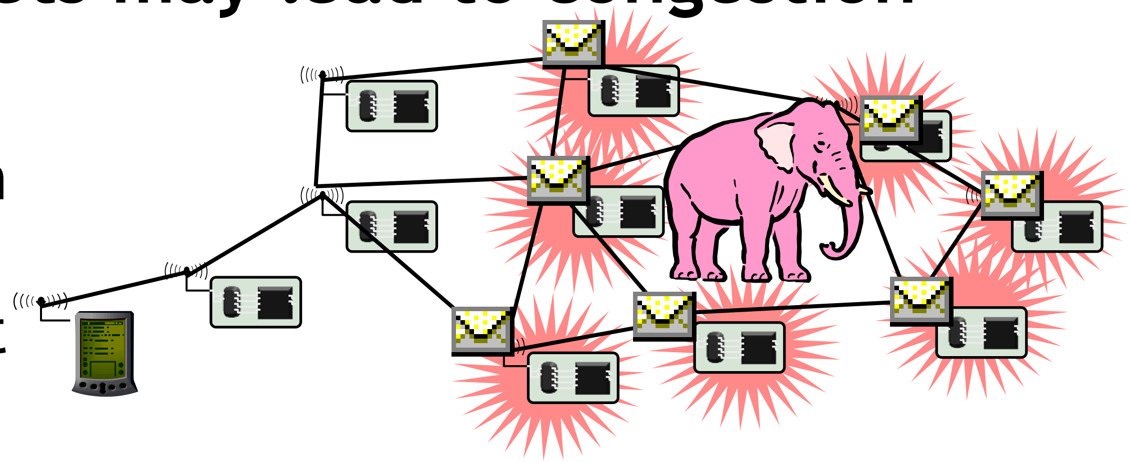
  - $\alpha \geq 1$ (scaling factor to adjust the delay in triggering the proactive fetch and is set to 1 usually)
  - S_{last} is the last highest sequence number packet received
  - S_{max} is the largest sequence number of the entire message

Outline

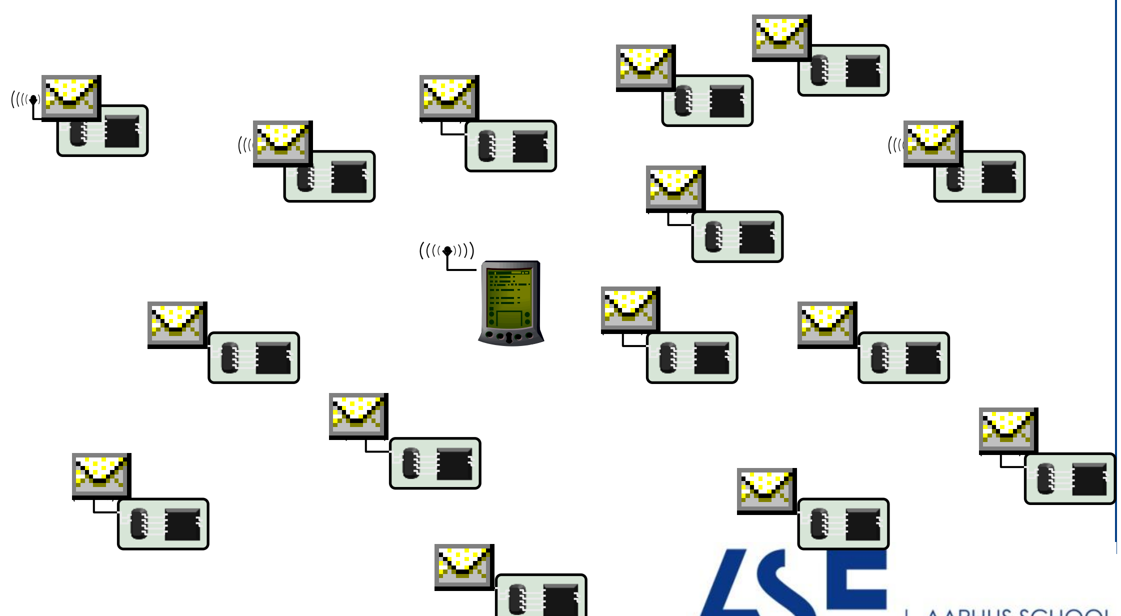
- Reliable data transport basics
- Single packets delivery
- Blocks of packets delivery
- **Congestion control and rate control**
  - **Congestion control basics**
  - CODA (Congestion Detection and Avoidance)
- Reliability & Congestion Control
  - ESRT (Event-to-Sink Reliable Transport protocol)

Streams of packets may lead to congestion

- When several sensors observe an event and try to periodically report it, congestion around event may occur



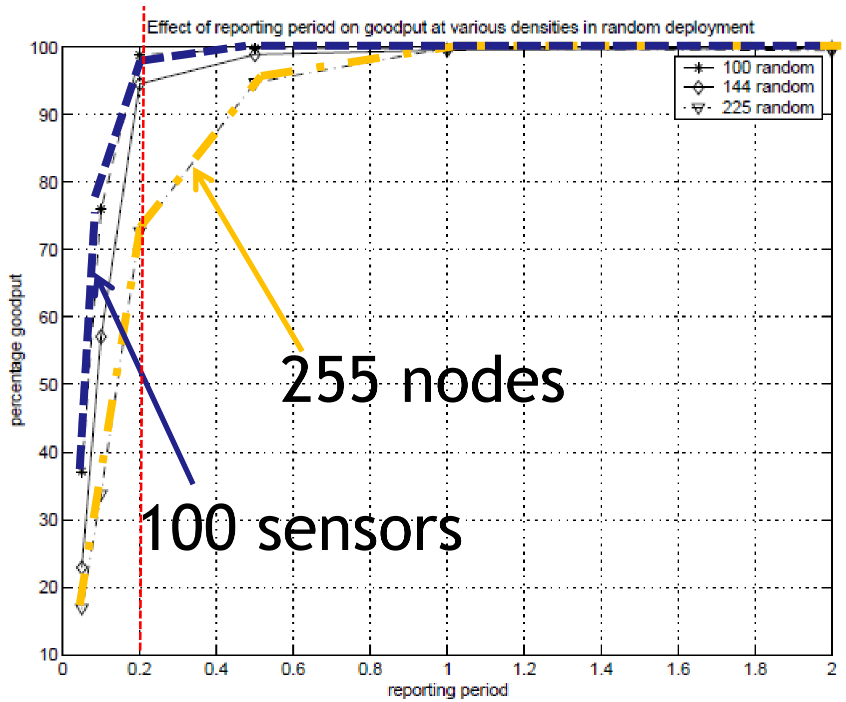
- When many sensors stream data to a sink, congestion around the sink may occur

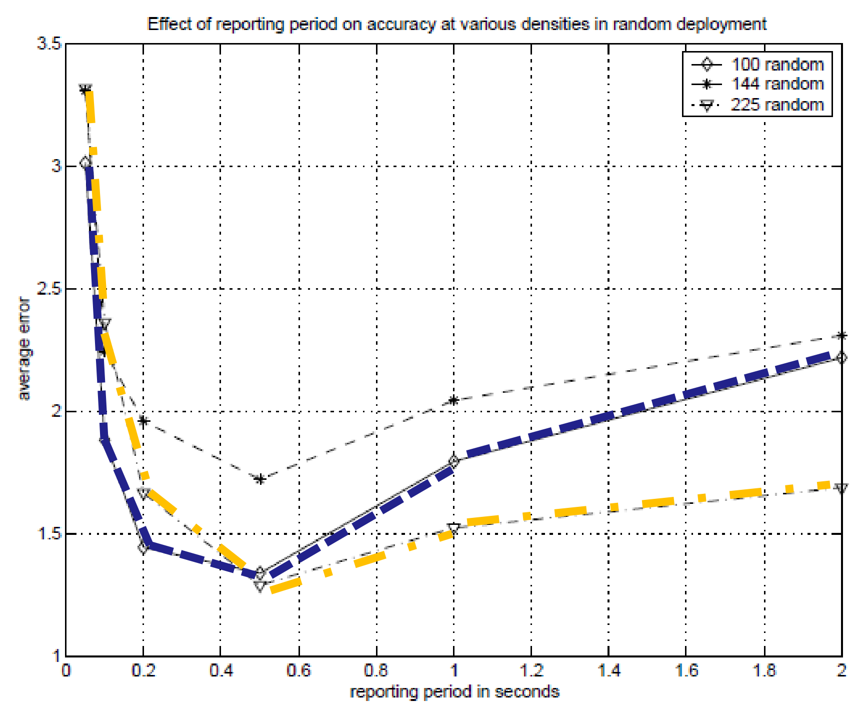


AARHUS SCHOOL OF ENGINEERING

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





Congestion Control

- Congestion detection
- Congestion Notification
  - Explicit message, e.g., suppression message (CODA)
  - piggypack, e.g., CN bit piggybacked in normal data packets (ESRT)
- Rate adjustment (Congestion Handling)
  - Typically, Additive Increase Multiplication Decrease (AIMD) scheme or its variants
  - Exact or accurate rate adjustment is possible if more congestion information is available.

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
- Measure channel utilization (channel sampling)
  - Need to interaction with MAC
    - CSMA-type MACs: high channel utilization, channel load = congestion; easy to detect
    - TDMA-type MACs: high channel utilization not problematic for throughput; congestion more difficult to detect

ASE | AARHUS SCHOOL OF ENGINEERING

Congestion handling

- Once congestion is (locally) detected, how to handle it?
- Option 1: Drop packets
  - No alternative ways when buffers overflow
  - Drop tail, random (early) drop (for TCP), …
  - Better: drop semantically less important packet
- Option 2: Rate control
  - Control sending rate of individual node
    - Rate of locally generated packets
    - Rate of remote packets to be forwarded, i.e., backpressure
  - Control how many nodes are sending
- Option 3: Aggregation, in-network processing

ASE | AARHUS SCHOOL OF ENGINEERING

Outline

- Reliable data transport basics
- Single packets delivery
- Blocks of packets delivery
- **Congestion control and rate control**
  - Congestion control basics
  - **CODA (Congestion Detection and Avoidance)**
- Reliability & Congestion Control
  - ESRT (Event-to-Sink Reliable Transport protocol)

AARHUS SCHOOL OF ENGINEERING

CODA

- Energy efficient congestion control.

- Three mechanisms are involved:
  - Congestion detection
  - Open-loop hop-by-hop backpressure
  - Closed-loop multi-source regulation

AARHUS SCHOOL OF ENGINEERING

Congestion detection in CODA

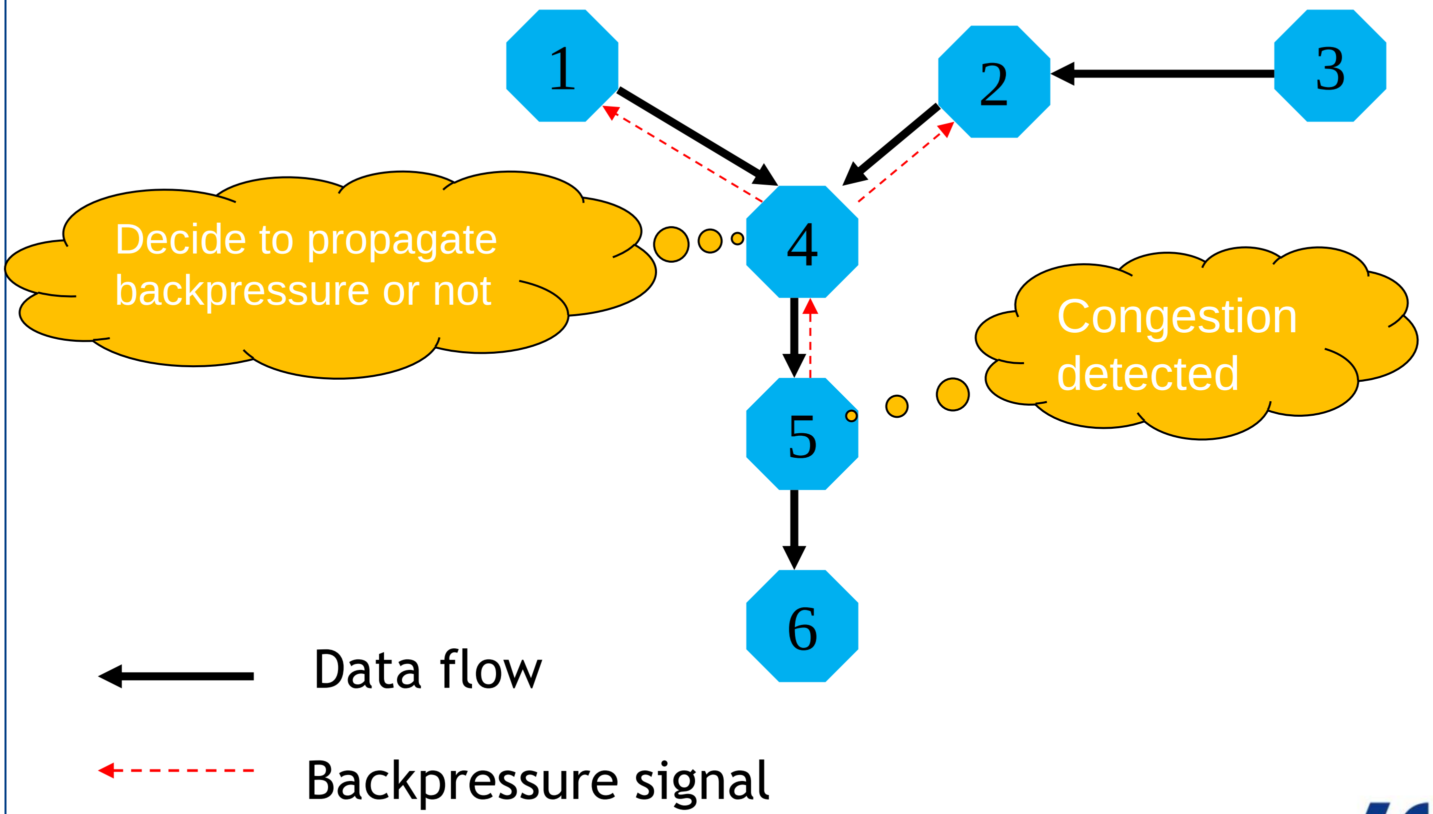
- There are multiple good indications of congestion:
  - A nearly overflowing queue
  - A measured channel load higher than a fraction of the optimum utilization
- Cost of monitoring queue size is nearly free but it provides only a bimodal indication
- Measure channel loading or acquire signal information for collision detection provides good indication, but incurs high energy cost
- **Key observation:**
  - with CSMA based MAC, no extra cost to listen and measure channel loading is needed when a node has packets in the buffer to transmit.
  - Channel loading measure will stop naturally when the buffer is cleared. It indicates with high probability that any congestion is mitigated.

Open-loop hop-by-hop backpressure

- Basic idea:
  - To do local congestion detection at each node with low cost.
  - Once congestion is detected, the node broadcasts a suppression message (backpressure signal) to its upstream neighbors
  - And also make local adjustments to prevent propagating the congestion downsteam
  - The node receiving backpressure signals will
    - Reduce sending rate or drop packets based on local congestion policy
    - Decide whether or not to further propagate the backpressure signal based on its local network condition
- Backpressure is a primary fast time scale control mechanism when congestion occurs

ASE | AARHUS SCHOOL OF ENGINEERING

Open-loop hop-by-hop backpressure



AARHUS SCHOOL OF ENGINEERING

Closed-loop multiple source regulation

- In WSN there is a need for congestion control of multiple sources from one sink.
  - i.e., sink plays 1 to N controller over multiple sources
- Basic idea:
  - When the source event rate r is less than some fraction η of the maximum theoretical throughput of the channel S_{max} $(i.e., r < \eta S_{max})$, the source regulate itself.
  - When $r \geq \eta S_{max}$, a source is more likely to contribute to congestion and closed-loop control is triggered (i.e. source triggers sink regulation).
    - The source sets a *regulate bit* in the data packets by which events are reported towards sink
    - If the sink receives a packet with regulate bit set, the sink has to send ACKs to regulate sources associated with a particular event. (e.g. 1 ACK per 100 data packets)

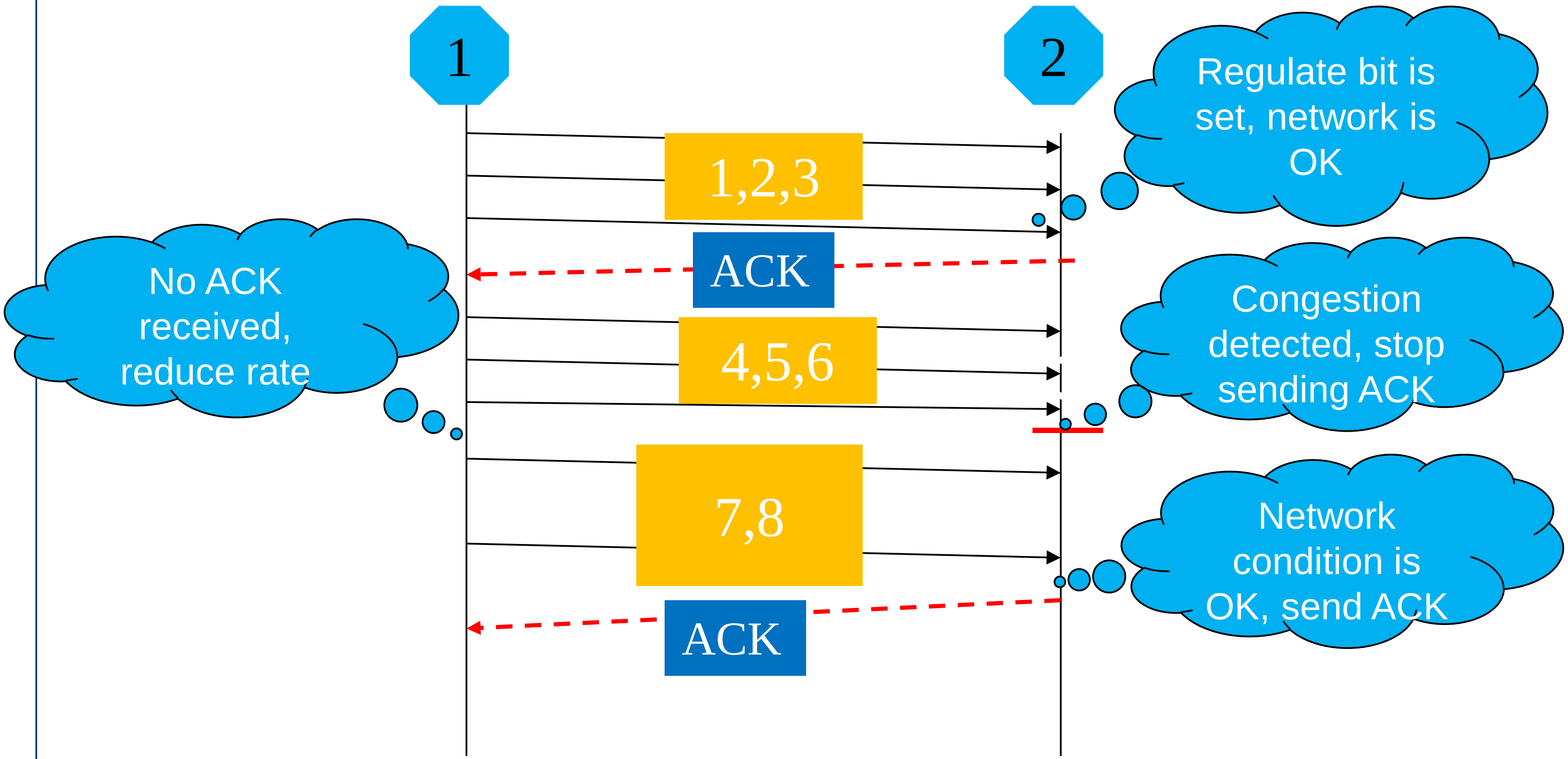
AARHUS SCHOOL OF ENGINEERING

Closed-loop multiple source regulation

- Cont..
  - When a source sets the *regulate bit*, it expects to receive an ACK every *N* packets (predefined value)
  - If a source receives an expected number of ACKs, it maintain its rate r
  - When congestion builds up, ACKs can be lost which can force sources to reduce their event rate r according to some rate decrease function.
  - Sink can also stop sending ACKs based on its own view of network conditions.

- The cost of closed-loop flow control is typically higher than simple open-loop control.

Closed-loop multiple source regulation



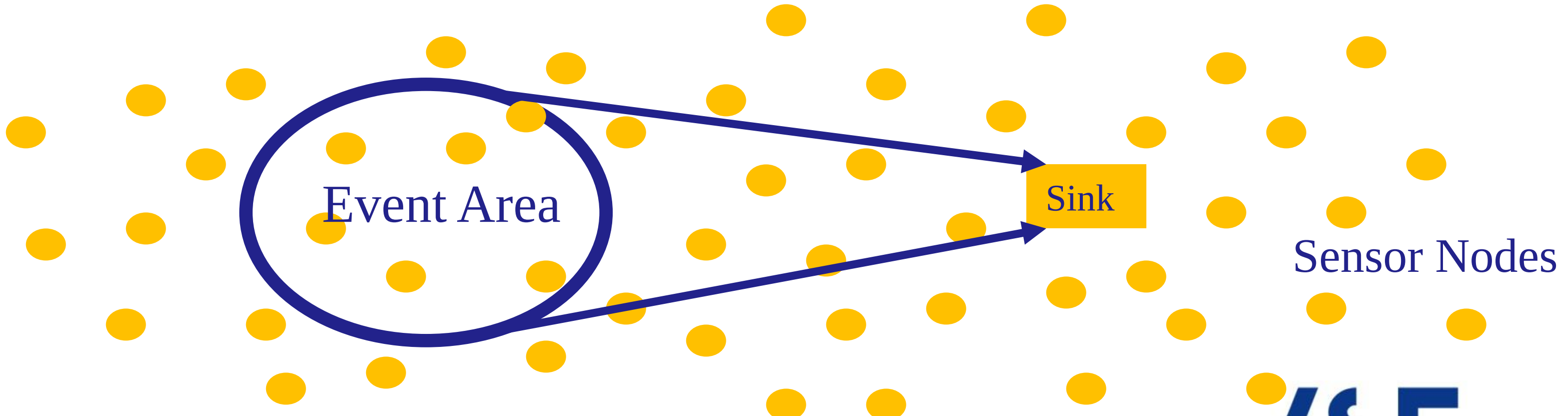
ASE | AARHUS SCHOOL OF ENGINEERING

Outline

- Reliable data transport basics
- Single packets delivery
- Blocks of packets delivery
- Congestion control and rate control
  - Congestion control basics
  - CODA (Congestion Detection and Avoidance)
- Reliability & Congestion Control
  - **ESRT (Event-to-Sink Reliable Transport protocol)**

Event-to-Sink Reliability Transport (ESRT)

- Sensor networks are 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. (*to guarantee event reliability*)
- Event-to-Sink collective reliability notion
- EXPLORE SPATIAL CORRELATION !!!!



Event-to-Sink Reliability Transport (ESRT)

- 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 sink
  - i.e., the distortion observed in the event estimation
- Definition 2: ***Desired Event Detection Reliability*** R is the number of packets required for reliable event detection,
  - i.e., the desired event estimation distortion level 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.

AARHUS SCHOOL OF ENGINEERING

Event-to-Sink Reliability Transport (ESRT)

- Definition 3: ***Reporting Frequency Rate***, f, of a sensor node
  - the number of packets sent out per unit time by a node.

- Definition 4: ***Transport Problem***
  - To configure the reporting frequency rate, f, of source nodes so as to achieve the *Desired event detection reliability*, R, at the sink with minimum resource utilization.

Event-to-Sink Reliability

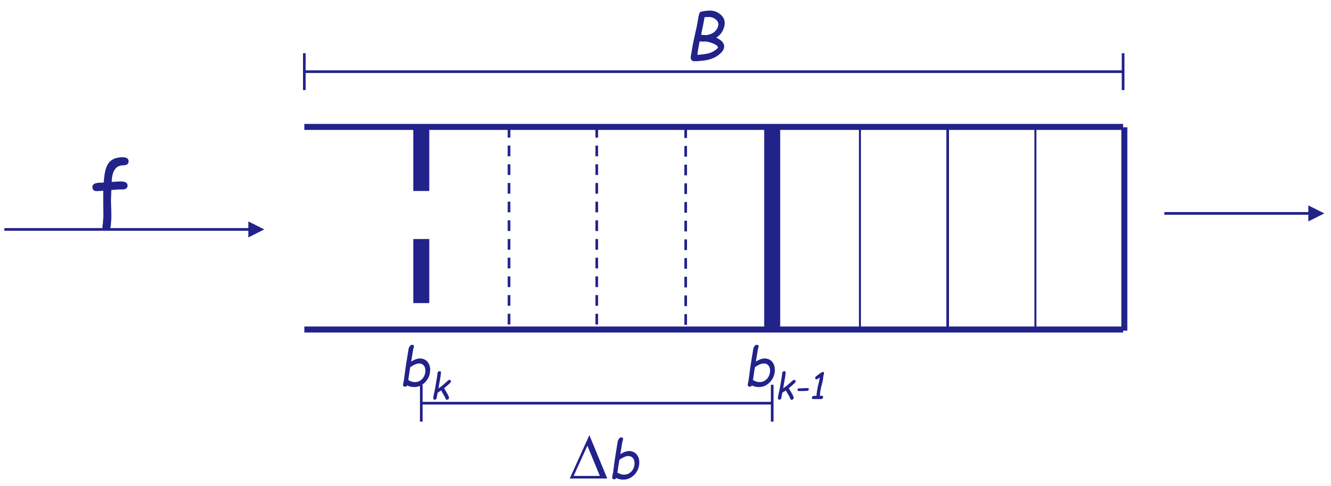
- Source (Sensor Nodes):
  - Send data with reporting frequency f
  - Monitor buffer level and notify congestion to the sink

- 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 the feedback from the sensor nodes
  - Updates f based on η_i and the relation between f and f_{max} (congestion) to achieve desired event reliability R
  - Broadcasts the new reporting rate to all sensor nodes directly with high power

AARHUS SCHOOL OF ENGINEERING

ESRT: Congestion Detection Mechanism

- ESRT uses buffer overflows to signal congestion



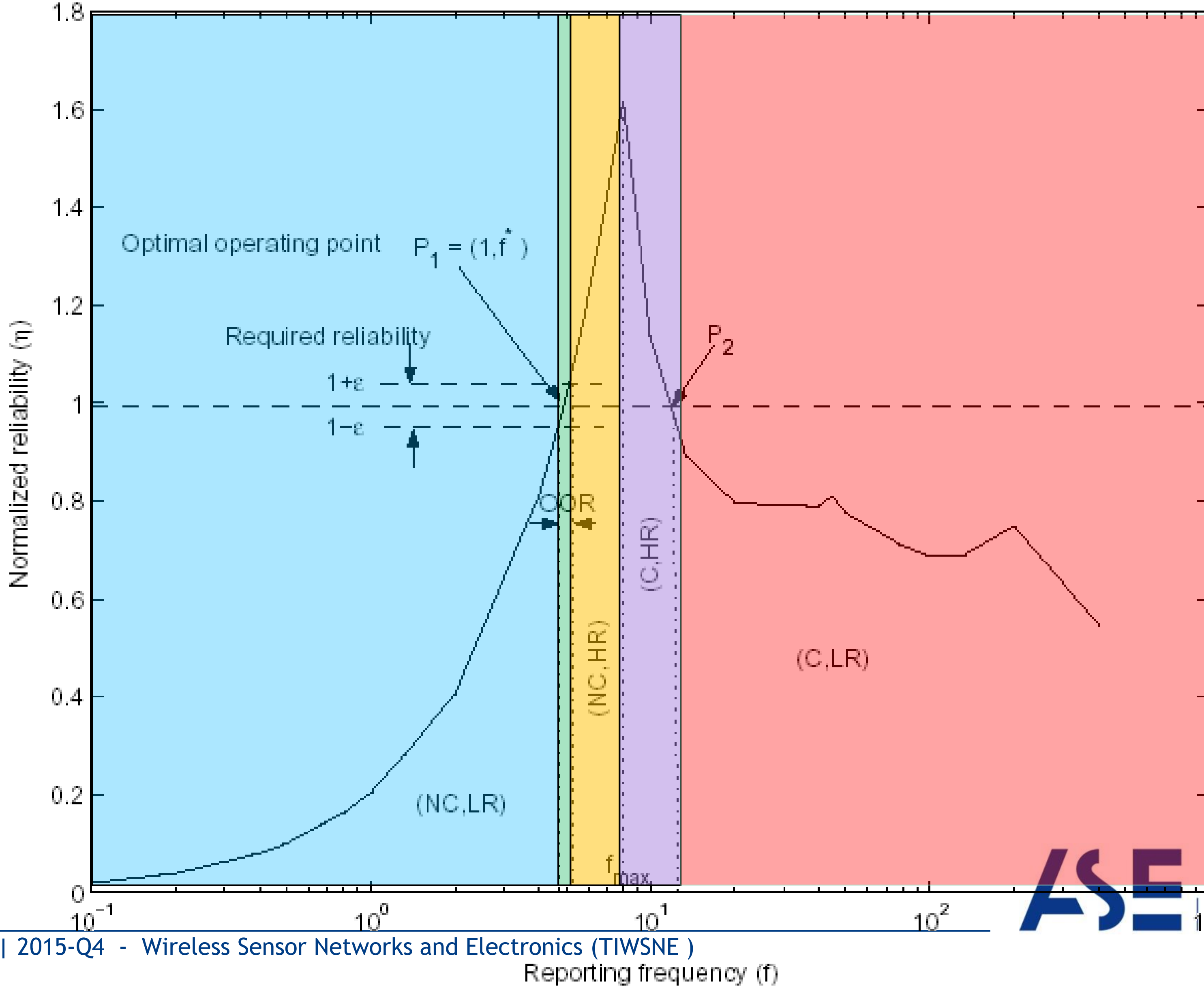
- b_k : Buffer fullness level at the end of reporting interval k
- Δb : Buffer length increment over past interval $\Delta b = b_k - b_{k-1}$
- B : Buffer size
- f : Reporting frequency

ESRT: Congestion Detection Mechanism

- Nodes mark Congestion Notification (CN) field in packet
  - At end of the interval *k th*, *if* $b_k + \Delta b > B$, the node infers that it will experience congestion in the next reporting interval
  - The node will set a special bit CN flag to 1 in the header, the sink can adjust the reporting frequency accordingly

Event ID	**CN (1 bit)**	Destination	Time Stamp	Payload	FEC

ESRT rate control tradeoff

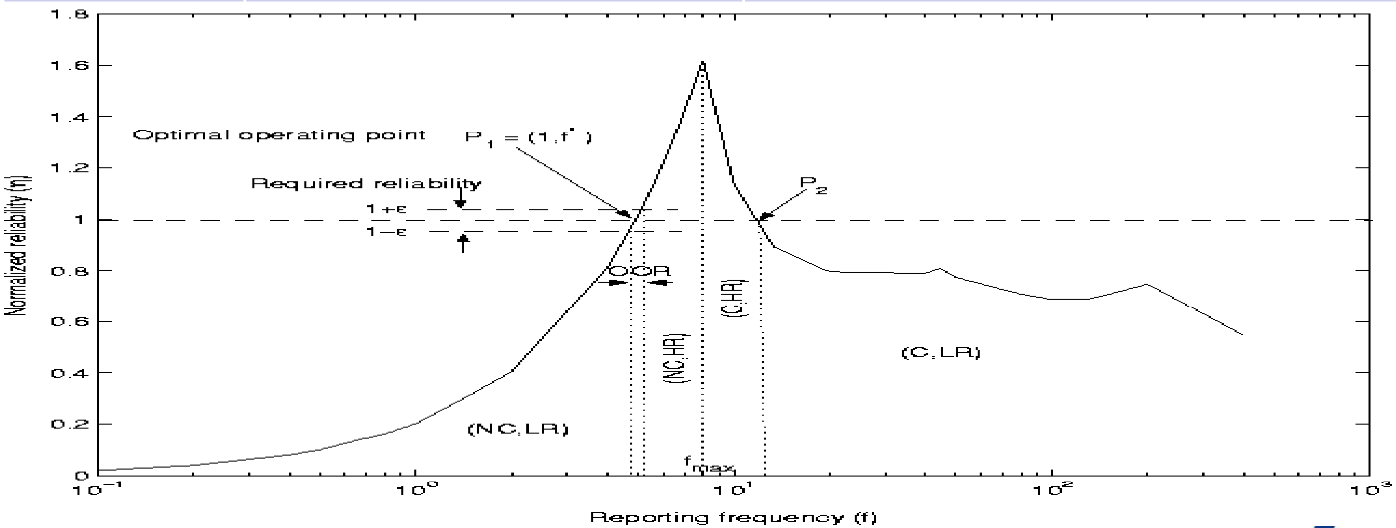


ESRT: Network status

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

ESRT: Frequency Update

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



AARHUS SCHOOL OF ENGINEERING

Design Guideline in Transport Layer Protocol

- Transport layer protocols should have two components
  - Congestion control
  - Reliability/Loss recovery
- Two approaches to achieve this
  - Design separate protocols, respectively, for congestion control and loss recovery
    - E.g., Joint use of CODA and PSFQ provide full functions required by transport protocol in WSNs
    - Flexible
  - Design a full-fledged protocol in an integrated way
    - Possibly optimize congestion control and loss recovery as they are often correlated.
    - Not well documented in literatures

ASE | AARHUS SCHOOL OF ENGINEERING

Summary

- Transport protocols have considerable impact on the services in wireless sensor networks
- Various facets – no "one size fits all" solution in sight
  - E.g., some protocols provide reliability in downstream and some are for upstream
  - E.g., some protocols are packet-driven and others are event-driven

PSFQ	**Hop-by-hop**	**Downstream**	**Packet reliability**
ESRT	**End-to-End**	**Upstream**	**Event reliability**