TRANSPORT PROTOCOLS & CONGESTION CONTROL IN WIRELESS SENSOR NETWORKS

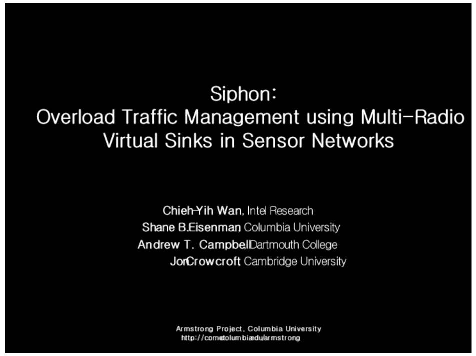
Journée thématique RESCOM
"IP et réseaux de capteurs"

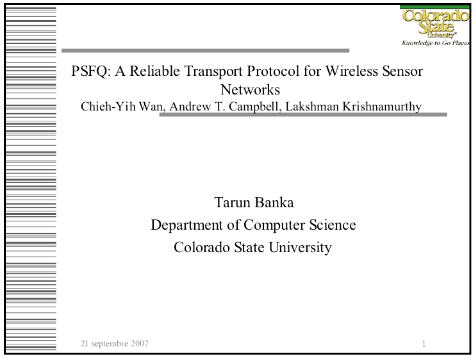
December 10th, 2008

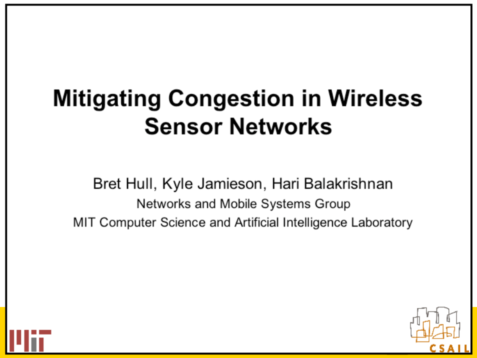
C. Pham
http://www.univ-pau.fr/~cpham
University of Pau, France

Acknowlegments

- These slides borrow some materials from le following presentations
  - Siphon: Overload Traffic Management using Multi-Radio Virtual Sinks in Sensor Networks, by Chieh-Yih Wan et al.
  - PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks, presented by Tarun Banka
  - Mitigating Congestion in Wireless Sensor Networks, by Hull et al.







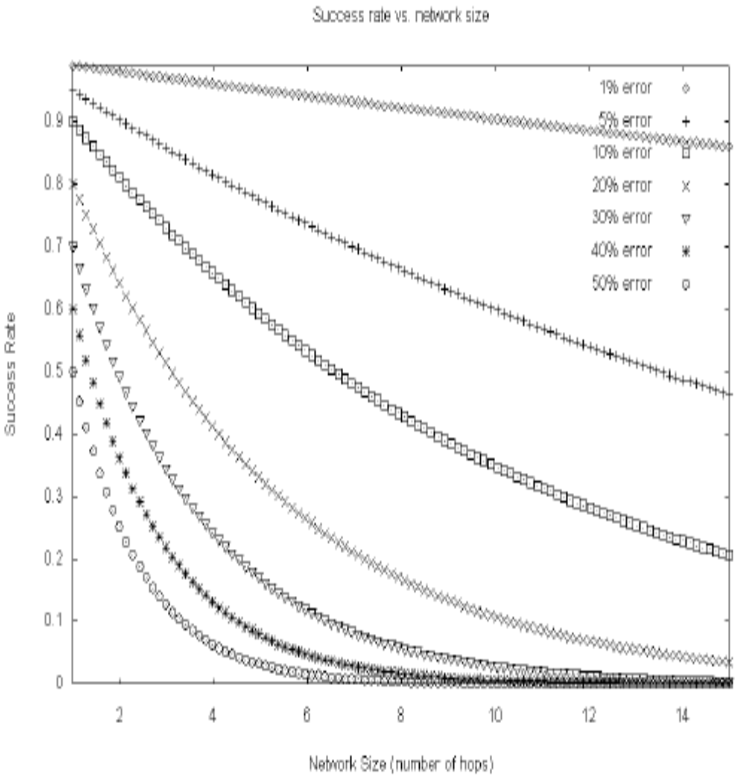
Transport layer for WSN

- Higher semantic than packet level
  - Multipoint communication
  - Data aggregation, data dissemination
- Reliability/loss recovery
- Congestion control
  - Congestion detection
  - Fairness issues

TCP or UDP?

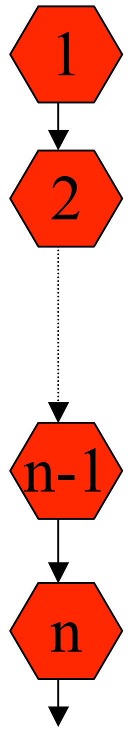
- TCP
  - Connection-oriented, 3-way handshake
  - Assumes segment losses results from congestion
  - E2E reliability
  - Congestion control mechanism
  - Fairness as a function of RTT
- UDP
  - No reliability
  - No flow control nor congestion control

Probability of successful delivery using E2E Model



$(1\text{-}p)^n$

Prob. to detect loss in NACK system.



$(1\text{-}p)$

$(1\text{-}p)^{n-1}$

$(1\text{-}p)^n$

p is the error rate of wireless link between two hops

Back in time?

- reliability in a hop-by-hop rather than end-to-end manner at either the MAC or transport layer
- best to avoid congestion entirely, or have packet losses occur close to the source. Back pressure is a useful technique
- Looks like protocols in the old time !
  - X.25
  - Frame Relay, ATM

TCP or specialized approach?

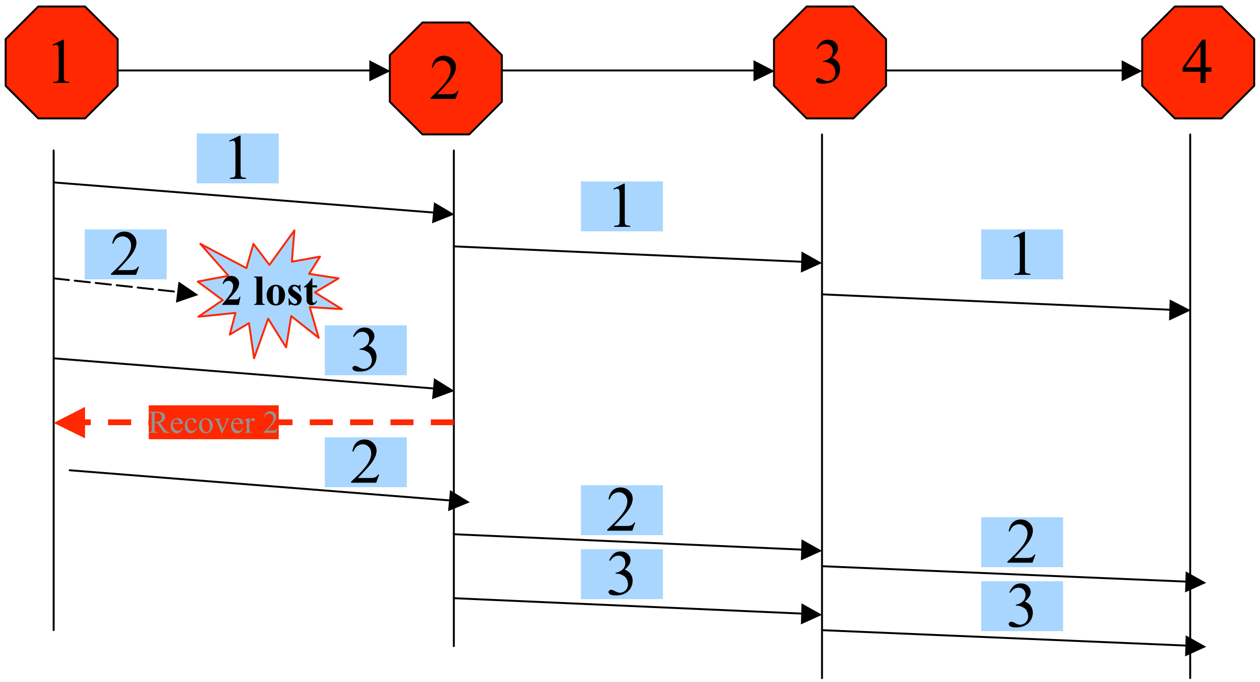
- TCP with appropriate modifications is better than UDP if standardized protocols are to be used.
  - Header compression
  - Help of link & MAC layers (cross-layering), segment caching,...
- Specialized approaches
  - Allow for a specific preference between reliability and congestion control
  - Application awareness is also possible: event reliability ≠ packet reliability

Ex: PSFQ
Pump Slowly and Fetch Quickly

- Inject packets (pump) in a controlled manner
- Recover (fetch) from losses locally (cache)
- Minimum signaling involved for Loss Detection and Recovery
- Operate correctly in high error prone environment

Based from a slide from PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks, presented by Tarun Banka

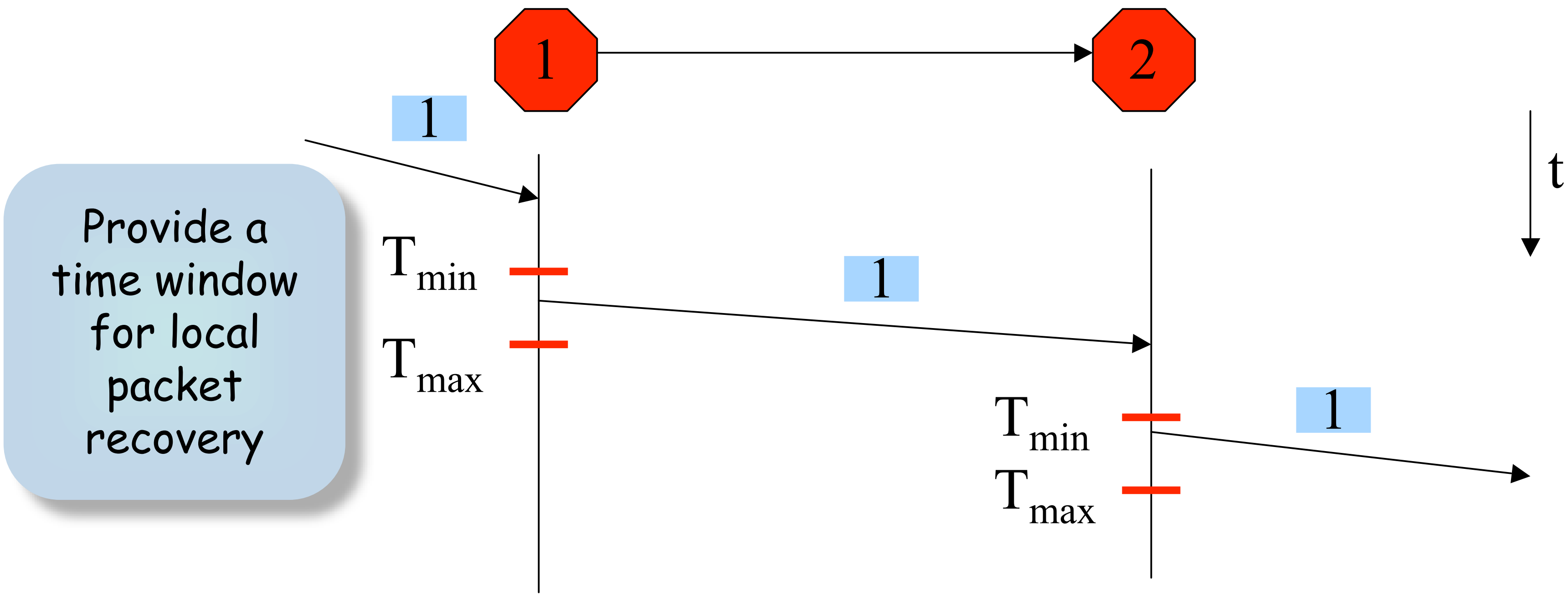
Recovering from Errors "Store and Forward"



No wastage of the Error Recovery control messages

From PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks, presented by Tarun Banka

PSFQ Pump Schedule

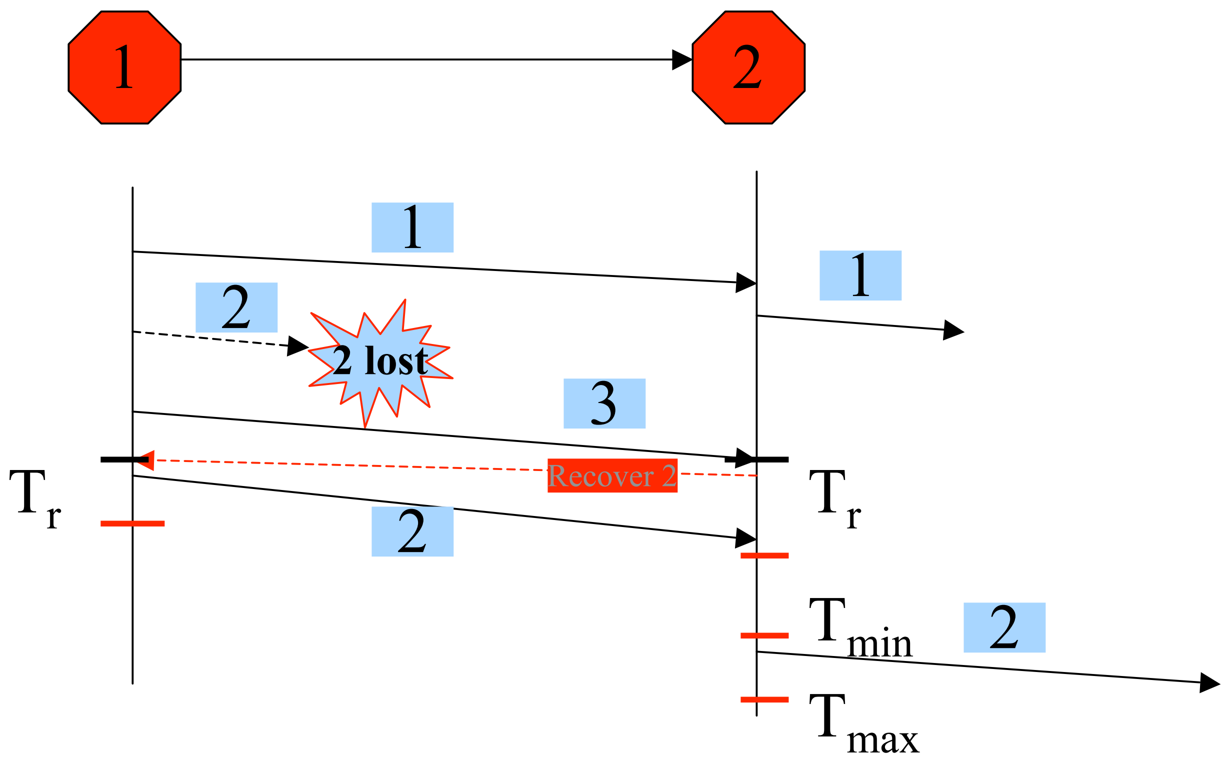


If not duplicate and in-order and TTL not 0

Cache and Schedule for Forwarding at time t (T_{min}<t<T_{max})

From PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks, presented by Tarun Banka

"Fetch Quickly" Operation



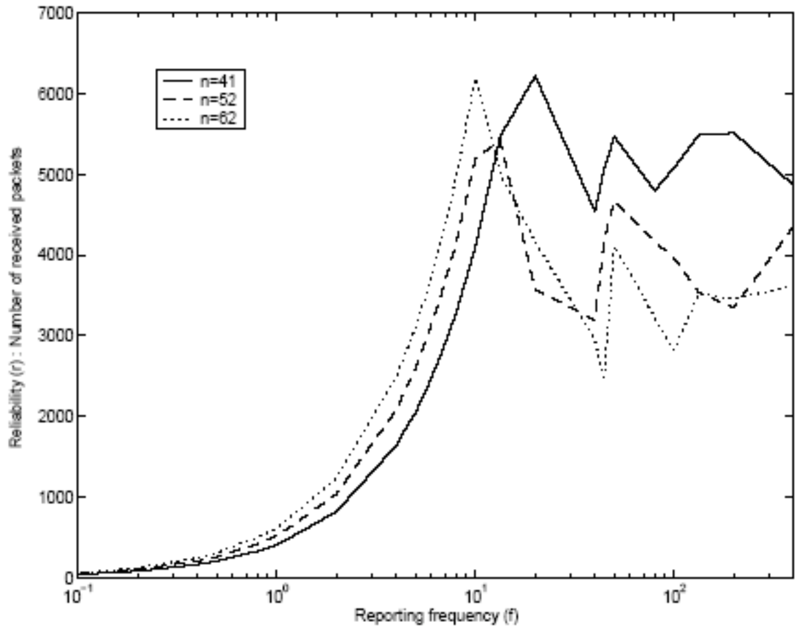
From PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks, presented by Tarun Banka

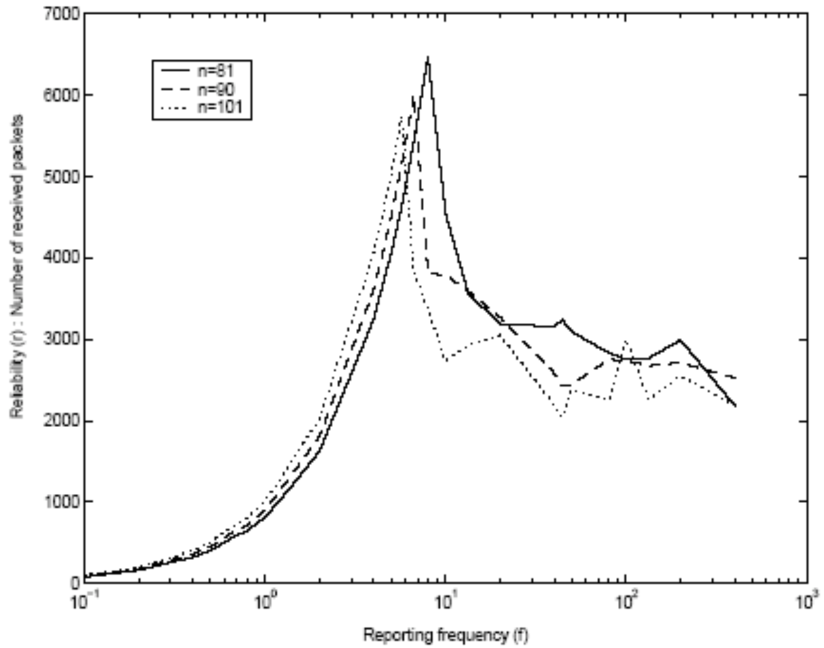
Ex: ESRT
Event-to-Sink Reliable Transport

- Places interest on events, not individual pieces of data
- Application-driven: Application defines what its desired event reporting rate should be
- Runs mainly on the sink
- Main goal: Adjust reporting rate of sources to achieve optimal reliability requirements → event reliability

Y. Sankarasubramanian, O. B. Akan, and I. F. Akyildiz, "ESRT: Event-to-Sink Reliable Transport," in Proceedings of ACM MobiHoc'03

Reliability vs Reporting frequency





- Initially, reliability increases linearly with reporting frequency
- There is an optimal reporting frequency (f_{max}), after which congestion occurs
- F_{max} decreases when the # of nodes increases

Characteristic Regions

η: normalized reliability indicator
ε: protocol parameter

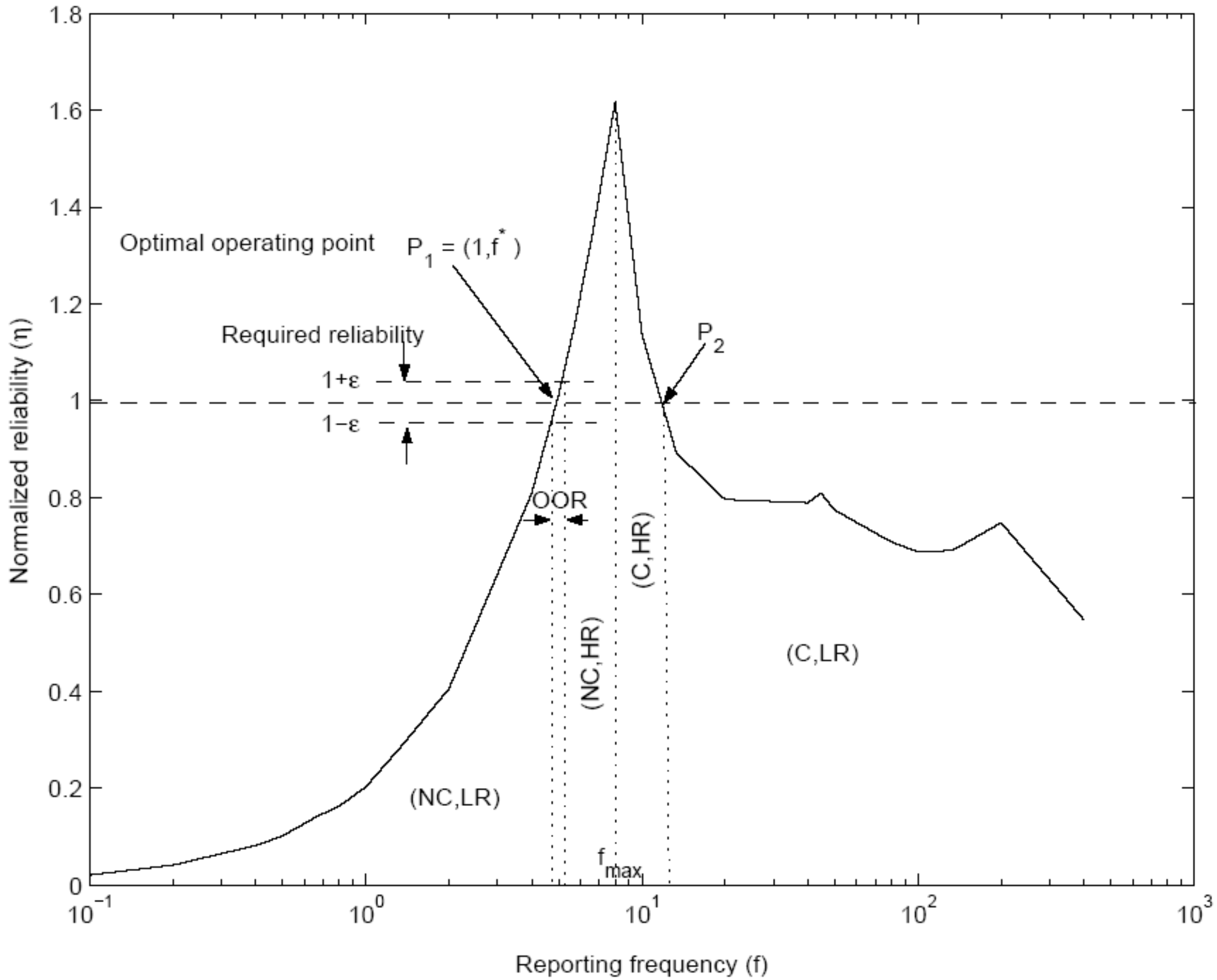
(NC,LR): No congestion, Low reliability
f < fmax, $\eta < 1-\varepsilon$

(NC, HR): No congestion High reliability
f <= fmax, $\eta < 1+\varepsilon$

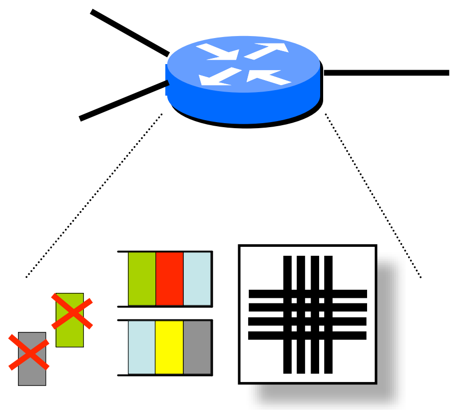
(C, HR): Congestion, High reliability
f > fmax, $\eta > 1$

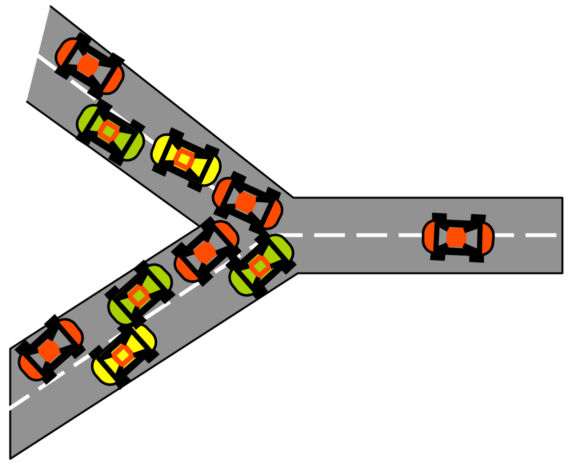
(C, LR): Congestion, Low reliability
f < fmax, $\eta <= 1$

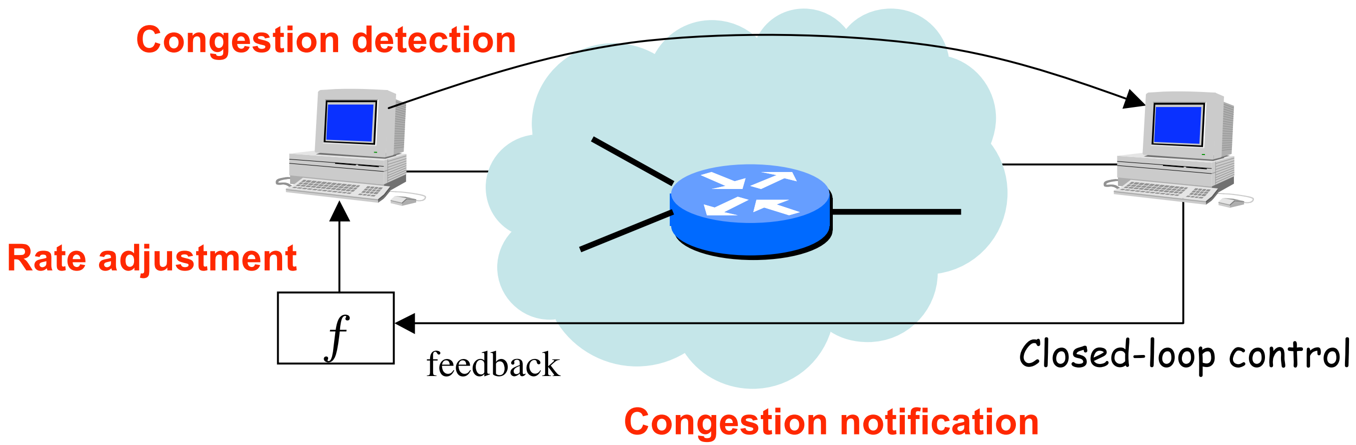
OOR: Optimal Operating Region
f < fmax,
$1-\varepsilon <= \eta <= 1+\varepsilon$



Congestion Control

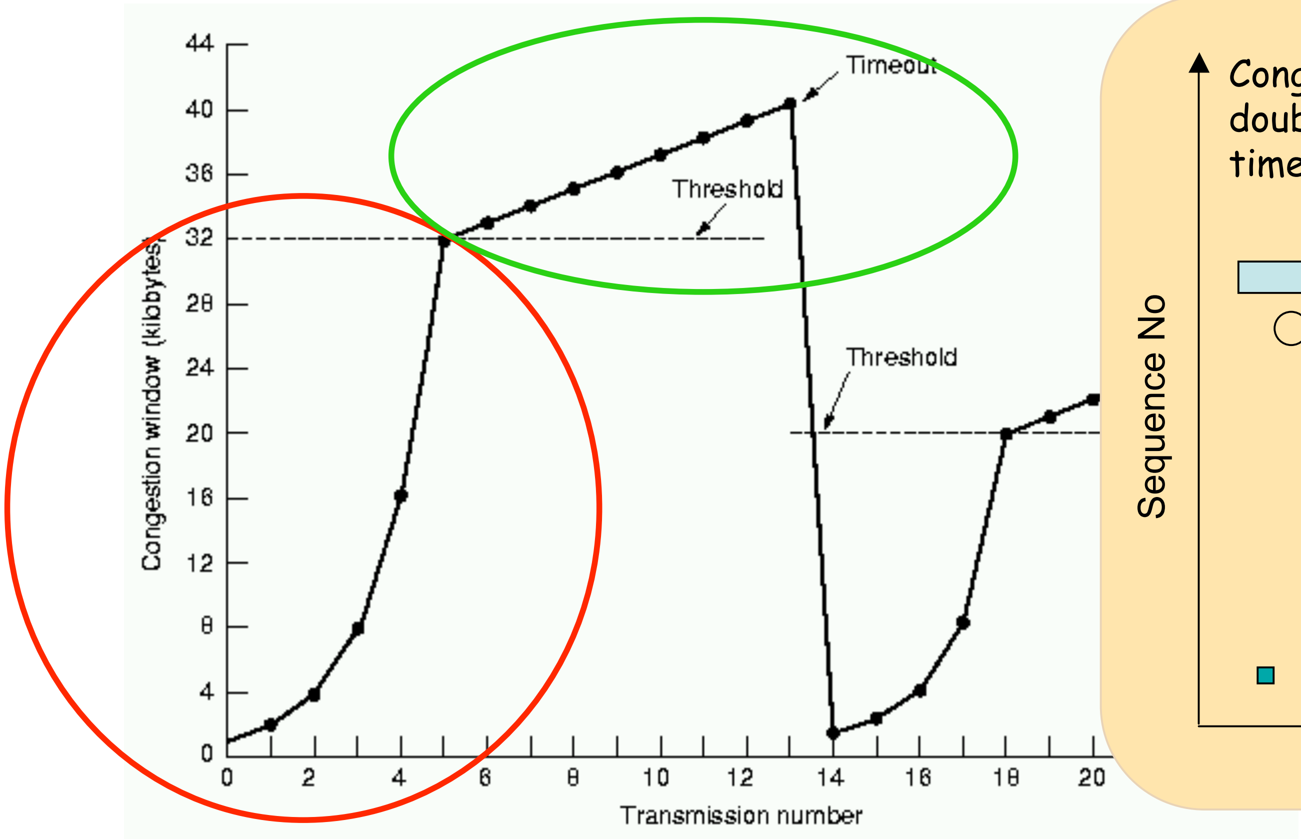


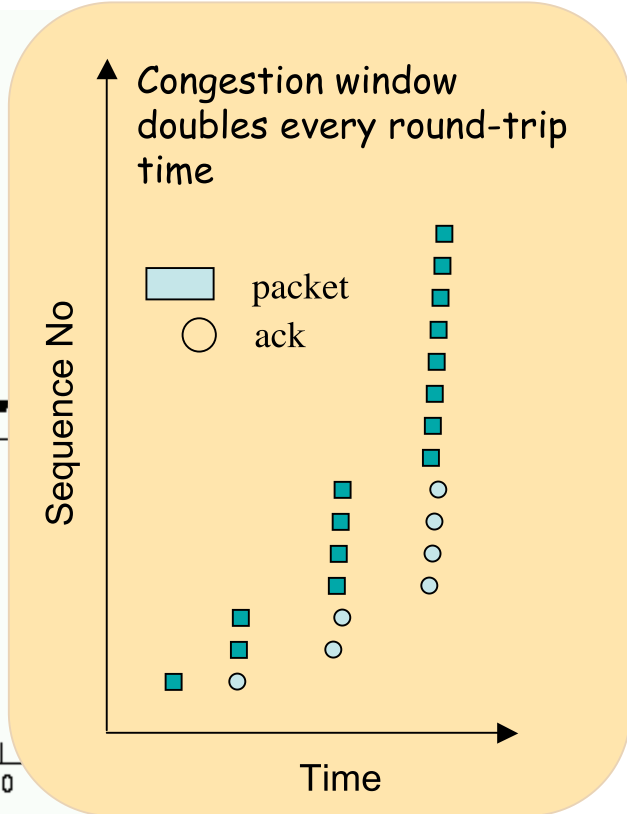




Feedback should be frequent, but not too much otherwise there will be oscillations
Can not control the behavior with a time granularity less than the feedback period

TCP congestion control





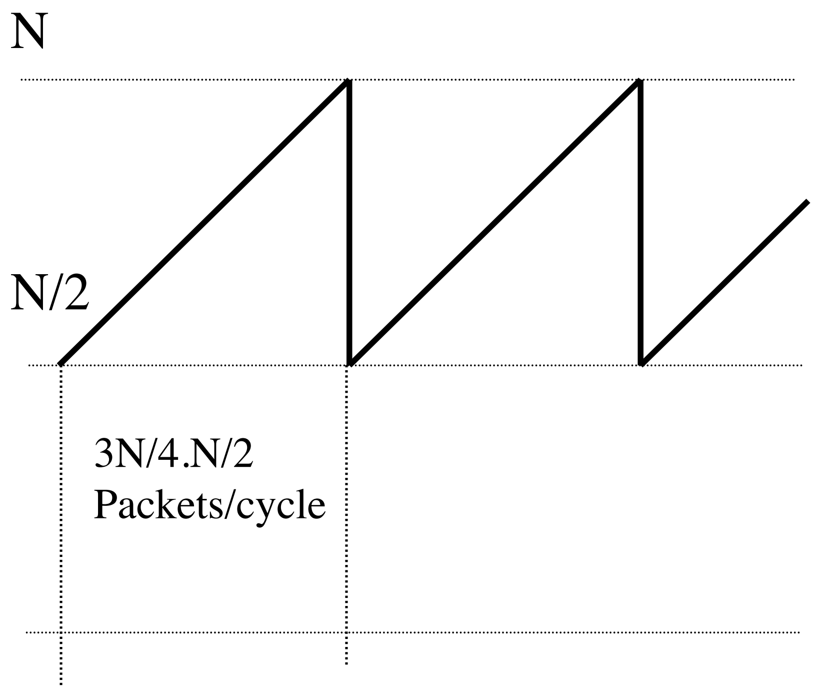
cwnd grows exponentially (slow start), then linearly (*congestion avoidance*) with 1 more segment per RTT

If loss, divides threshold by 2 (multiplicative decrease) and restart with cwnd=1 packet

TCP in steady state

TCP behavior in steady state

Isolated packet losses trigger the fast recovery procedure instead of the slow-start.



- The TCP steady-state behavior is referred to as the Additive Increase- Multiplicative Decrease process

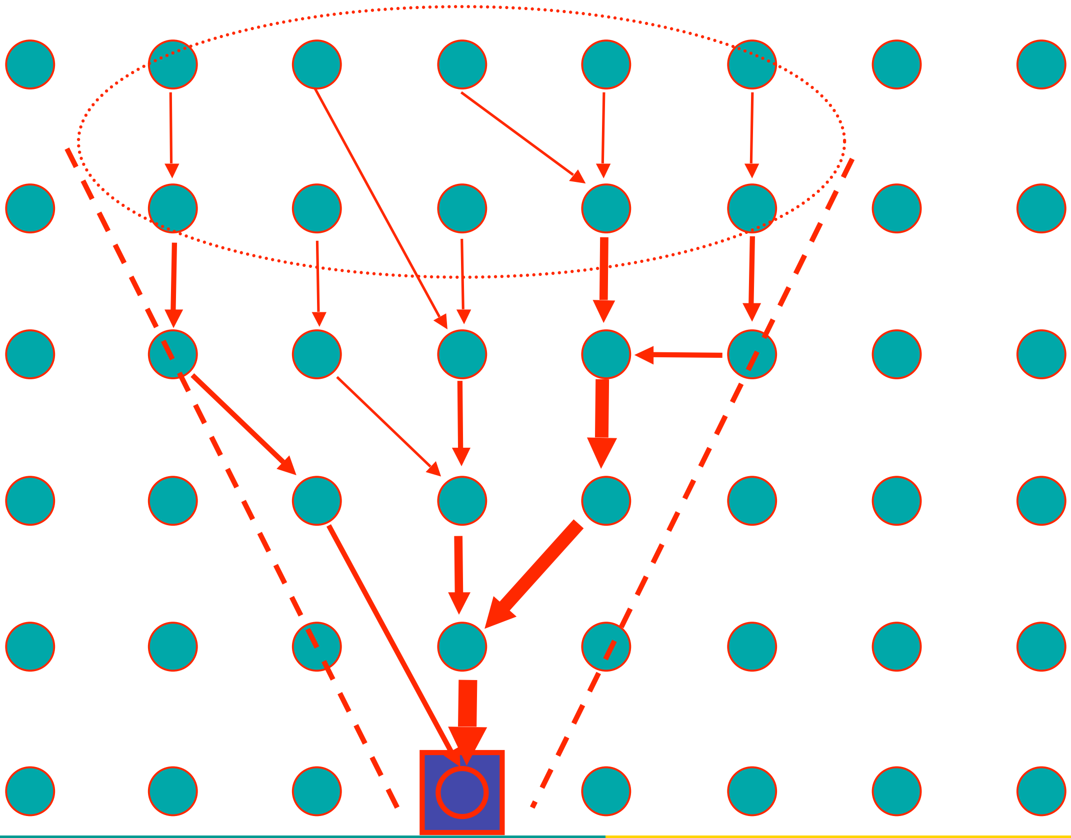
no loss:
cwnd = cwnd + 1
loss:
cwnd = cwnd*0.5

Congestion in wireless env.

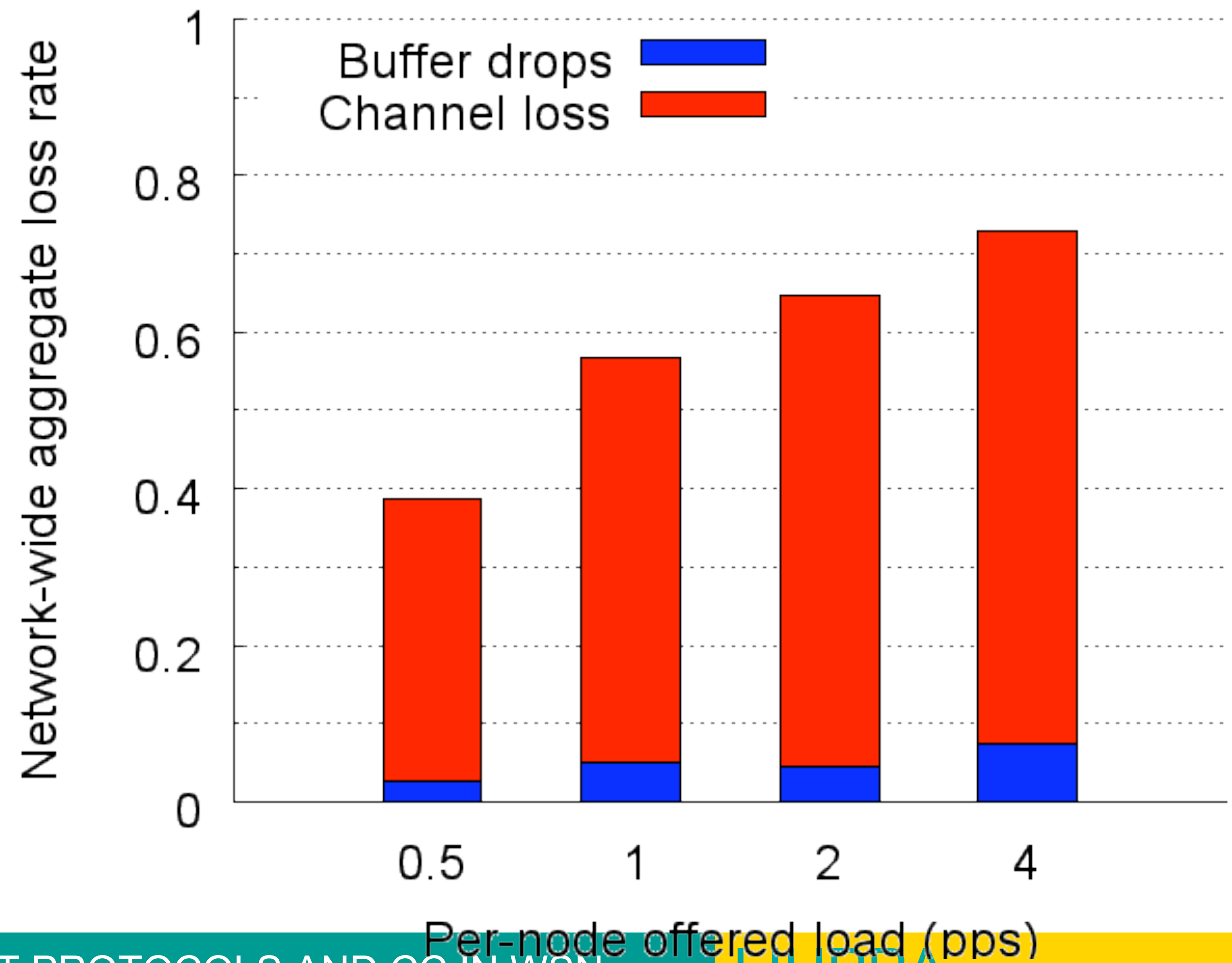
- Very lossy environments
- High interferences
- Difficult to distinguish congestions from node failures or bad channel quality
- Input queue occupancy is not a good indicator of congestion level !!

Funneling Effect

- Many-to-one traffic pattern causes congestion in the routing funnel



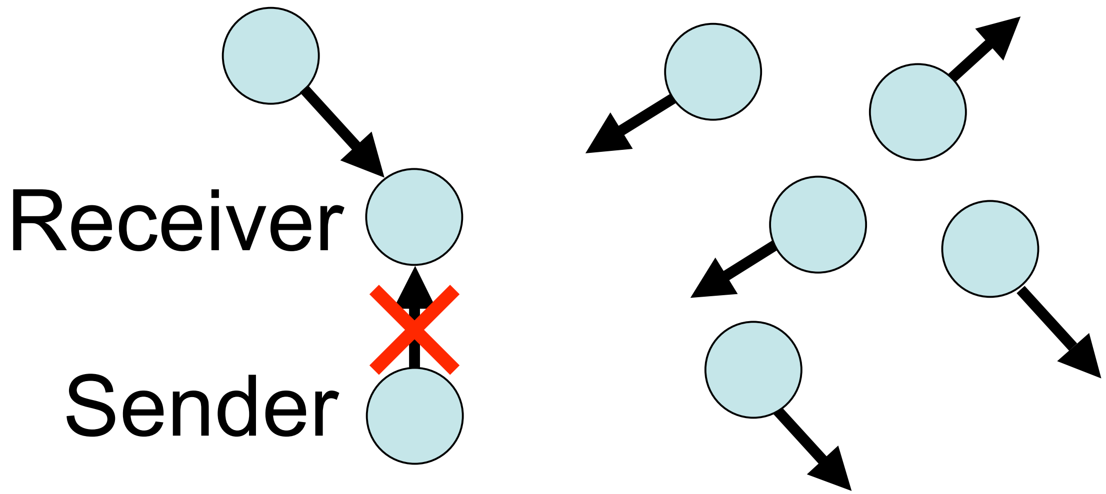
Congestion dramatically degrades channel quality



From "Mitigating Congestion in Wireless Sensor Networks", by Hull et al.

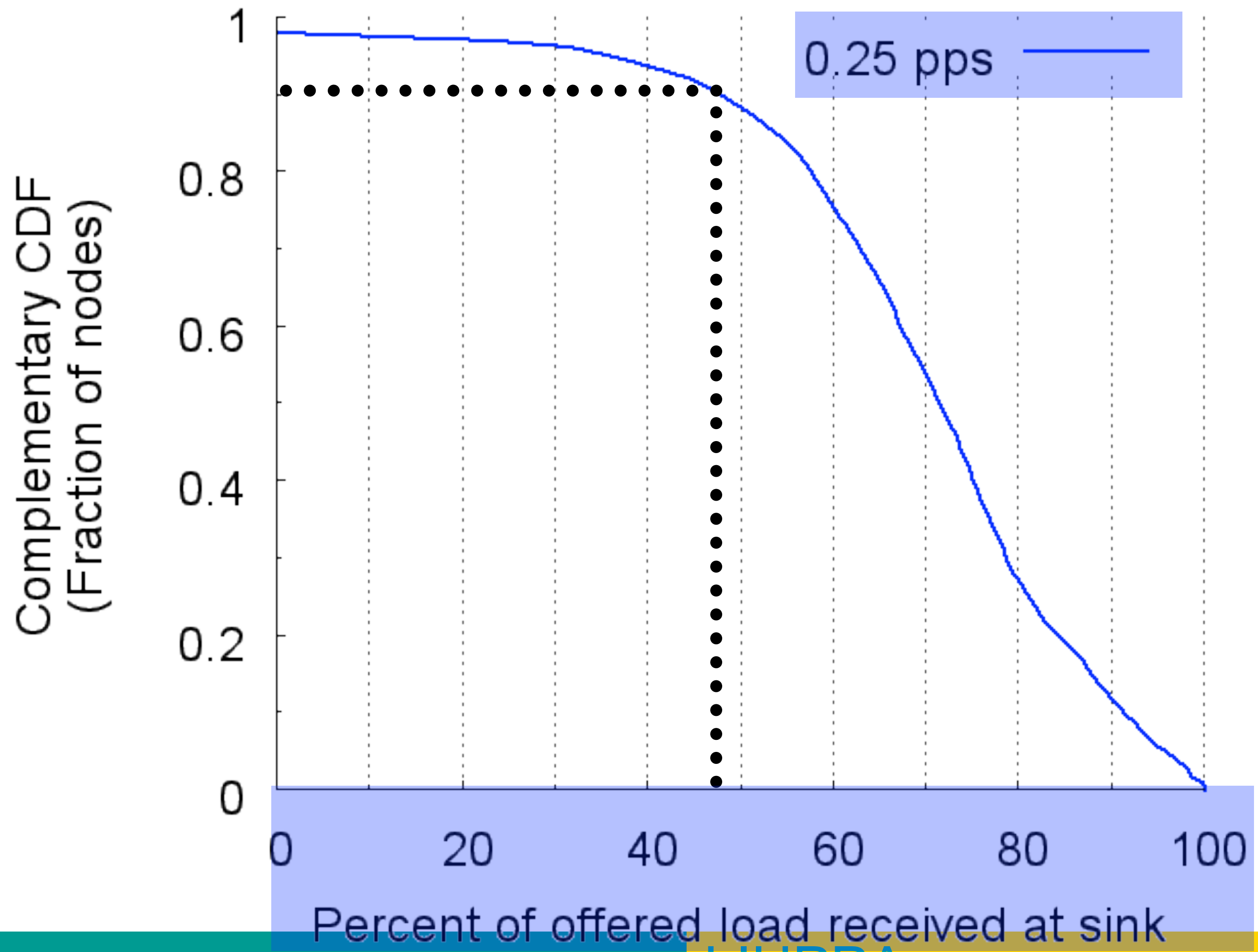
Why does channel quality degrade?

- Wireless is a shared medium
  - Hidden terminal collisions
  - Many far-away transmissions corrupt packets



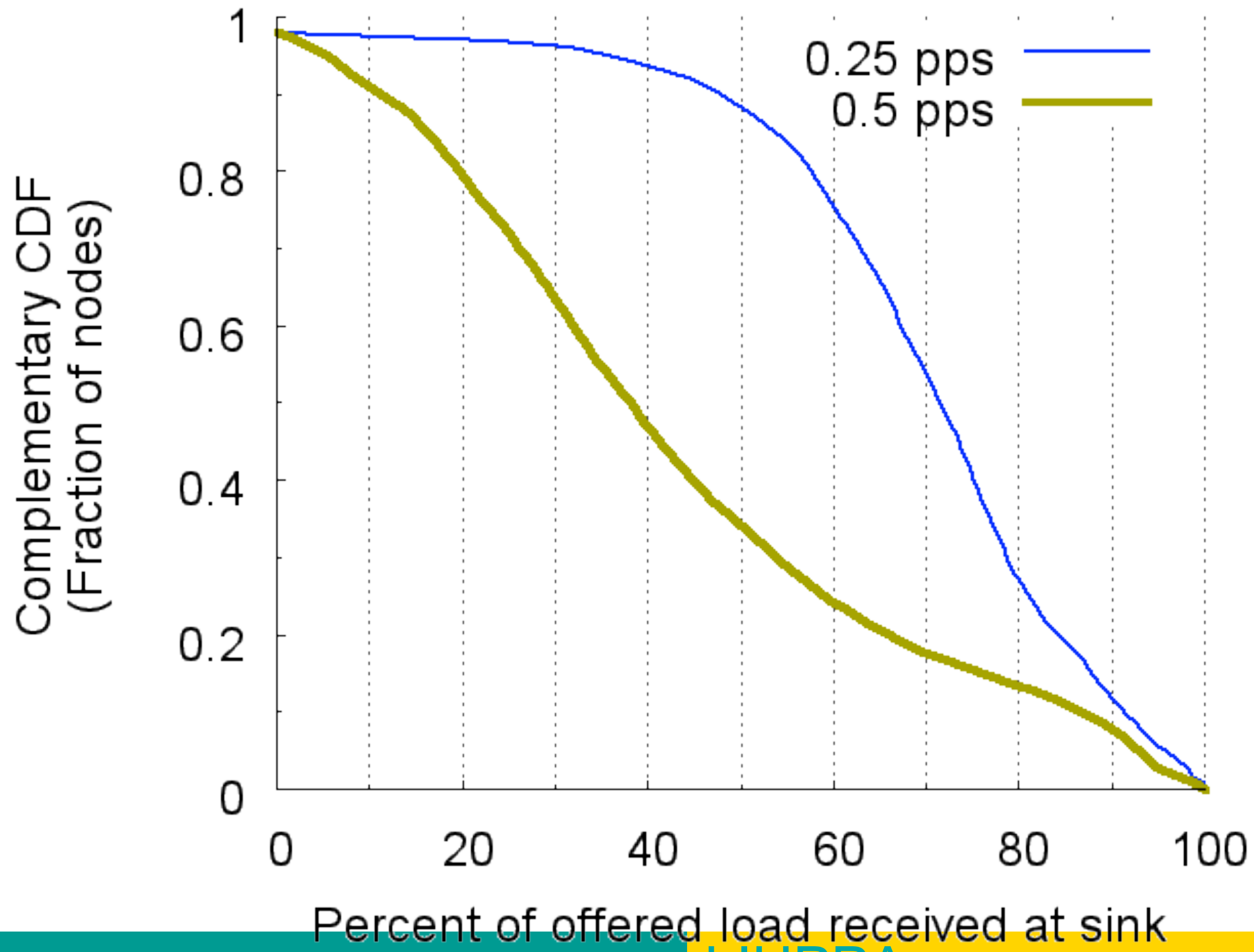
From "Mitigating Congestion in Wireless Sensor Networks", by Hull et al.

Per-node throughput distribution



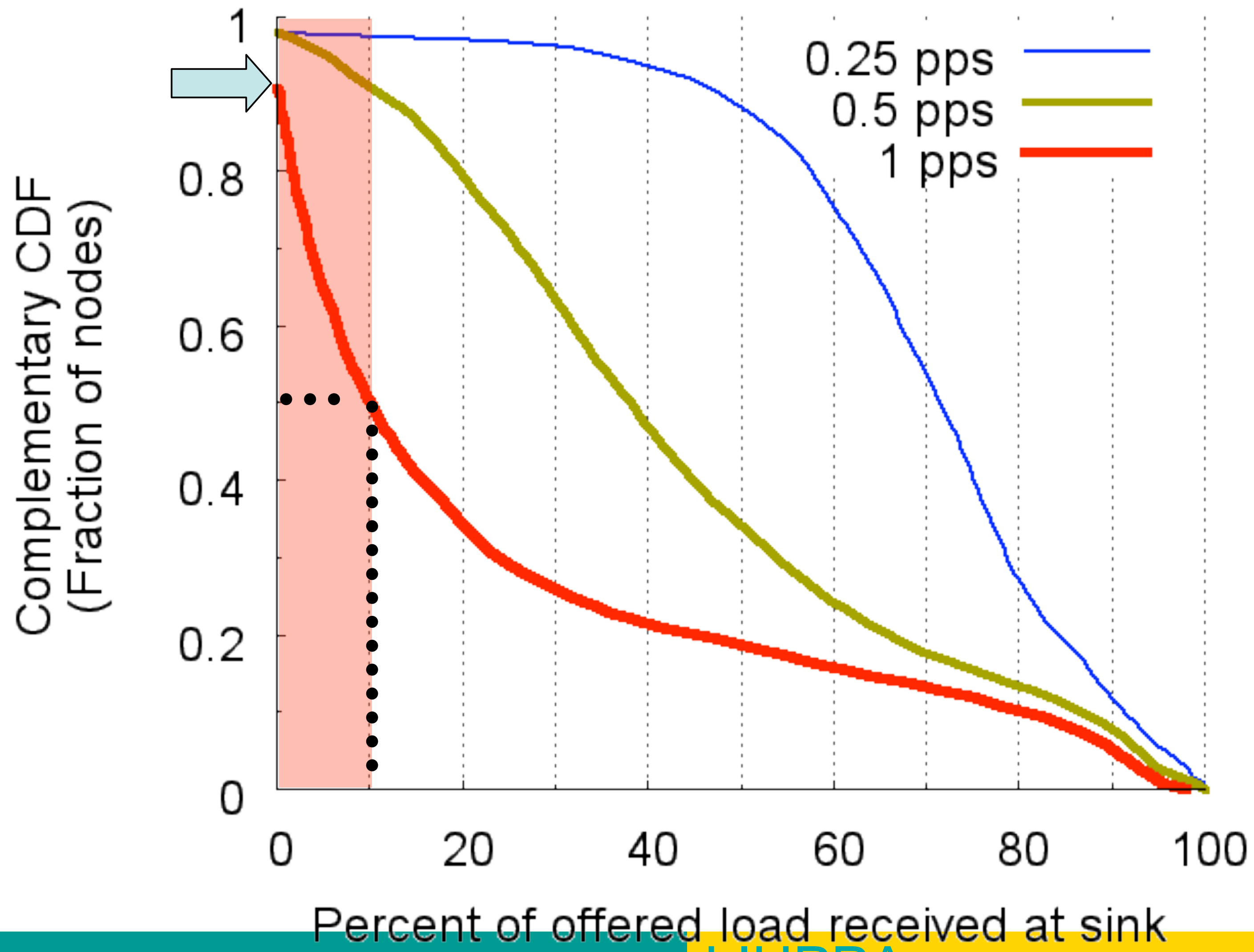
From "Mitigating Congestion in Wireless Sensor Networks", by Hull et al.

Per-node throughput distribution



From "Mitigating Congestion in Wireless Sensor Networks", by Hull et al.

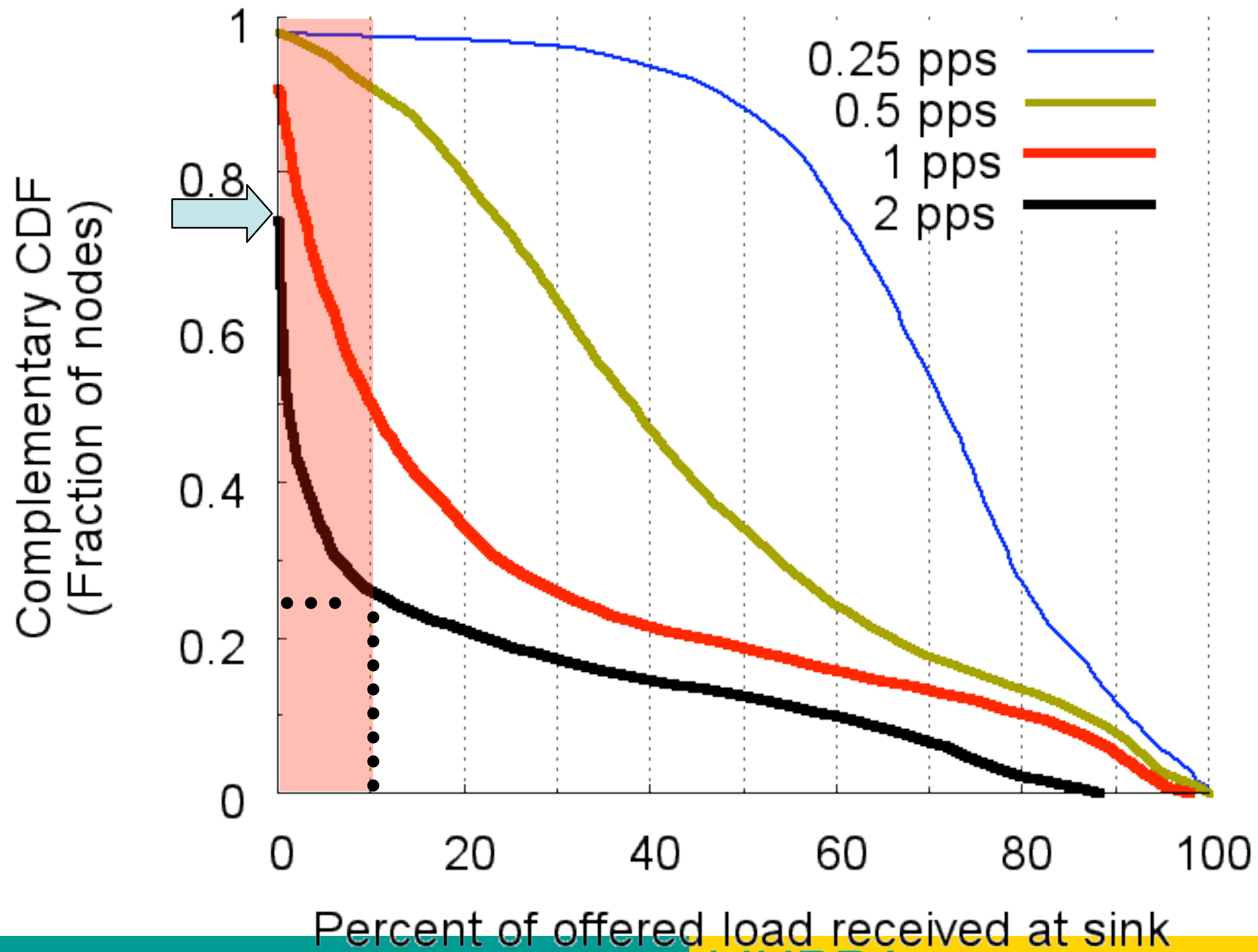
Per-node throughput distribution



From "Mitigating Congestion in Wireless Sensor Networks", by Hull et al.

Per-node throughput distribution

From "Mitigating Congestion in Wireless Sensor Networks", by Hull et al.



TCP Westwood example

- Enhance Congestion Control Using Eligible Rate Estimate (ERE)
- ERE is computed at the sender by sampling and exponentially averaging an estimate of the instantaneous bandwidth share used by the connection
- Bandwidth samples are determined from ACK inter-arrival times and info in ACKs

From "TCP Westwood: Enhanced Congestion Control for Large Leaky Pipes", *M. Gerla, G.Pau, M. Y. Sanadidi, and R.Wang, 2001*

TCPW's estimations

<u>BE Sampling</u>: $S_k = d_k / (t_k - t_{k-1})$

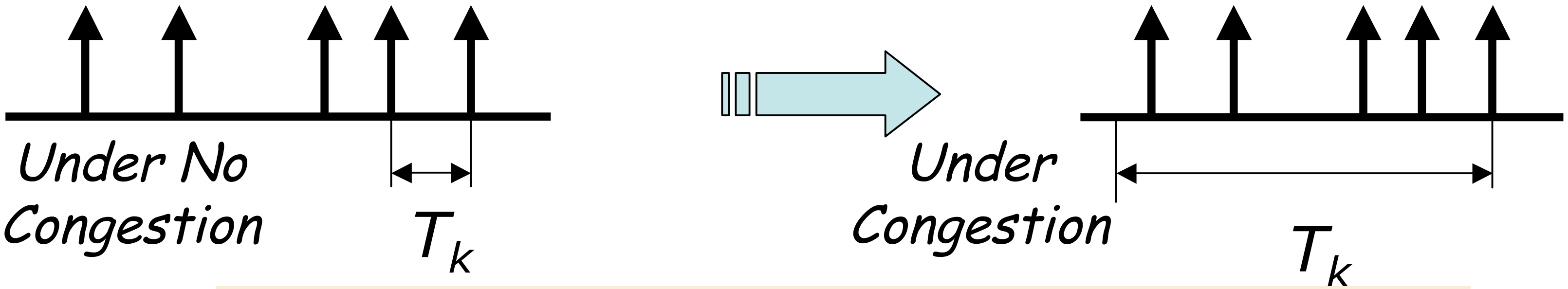
Packet *pair*,

effective under random loss,
overestimates under congestion

<u>RE Sampling</u>: $S_k = \dfrac{\sum_{t_j > t_k - RTT} d_j}{RTT}$

Packet *train*,

fair estimate under congestion,
underestimates under random loss



$$b_k = d_k / (t_k - t_{k-1})$$ sample

$$BE_k = \alpha_k BE_{k-1} + (1 - \alpha_k)\left(\frac{b_k + b_{k-1}}{2}\right)$$ exponential filter

$$\alpha_k = \frac{2\tau - \Delta t_k}{2\tau + \Delta t_k}$$ filter gain

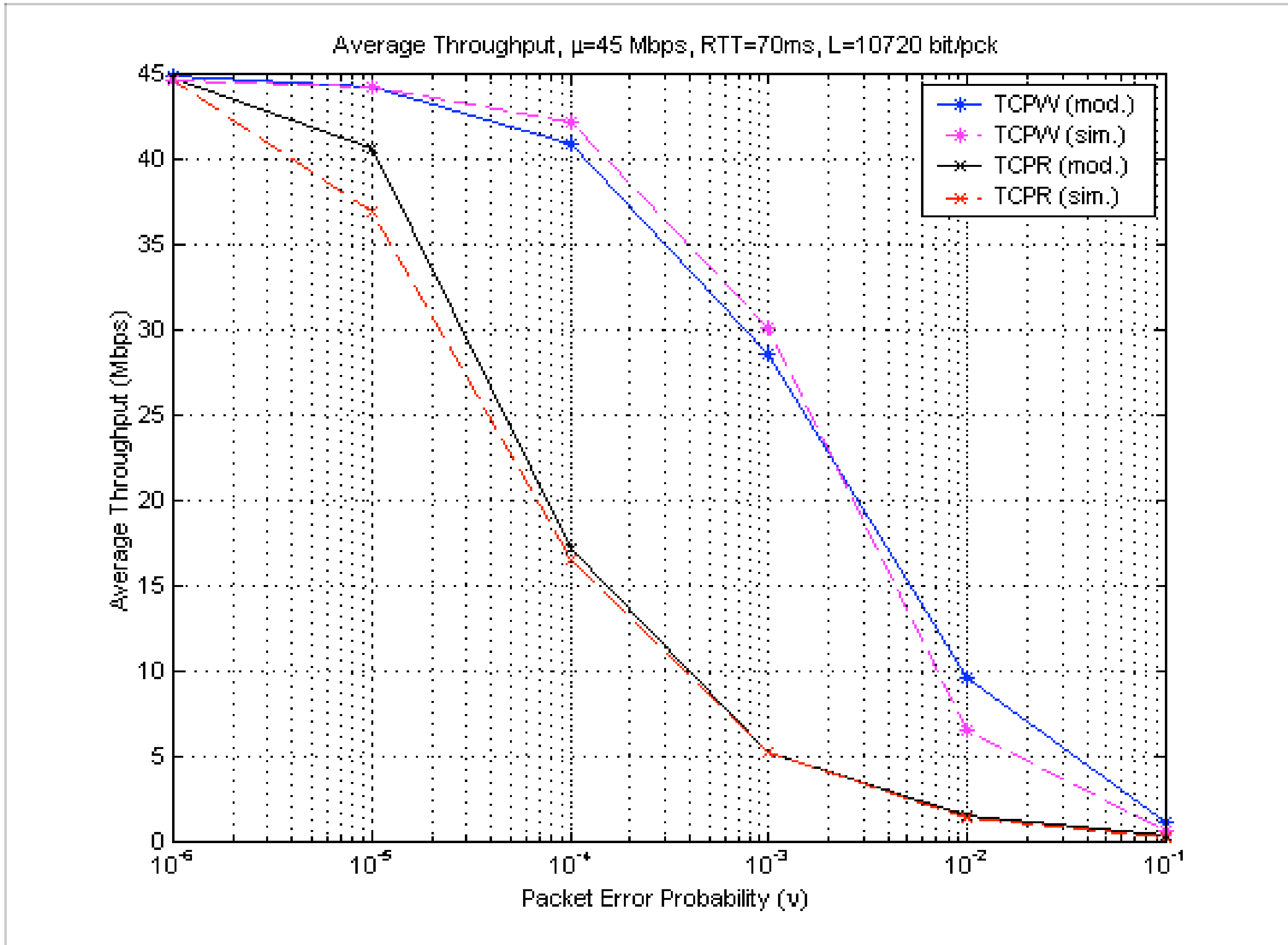
TCPW algorithm

- When three duplicate ACKs are detected:
  - set ssthresh=ERE*RTT
    (instead of ssthresh=cwnd/2 as in Reno)
  - if (cwnd > ssthresh) set cwnd=ssthresh

- When a TIMEOUT expires:
  - set ssthresh=ERE*RTT; set cwnd=1 ;
    (instead of ssthresh=cwnd/2 as in Reno)

From "TCP Westwood: Enhanced Congestion Control for Large Leaky Pipes", *M. Gerla, G.Pau, M. Y. Sanadidi, and R.Wang, 2001*

TCPW Throughput Gain
(Analysis Validated By Simulation)



45Mb/s link

70 msec RTT

Router Buffer Size=294

(=Pipe Size)

From "TCP Westwood: Enhanced Congestion Control for Large Leaky Pipes", *M. Gerla, G.Pau, M. Y. Sanadidi, and R.Wang, 2001*

WSN vs Internet

- Small fraction of time dealing with impulses, but data of greatest importance!
- Sensors have limited resources
  - Simplicity in congestion detection and control algorithms
  - Great interest of in-network processing: hop-by-hop CC more efficient than E2E?
- WSN are collaborative in nature. Fairness issues less important?

CC scenario in WSN

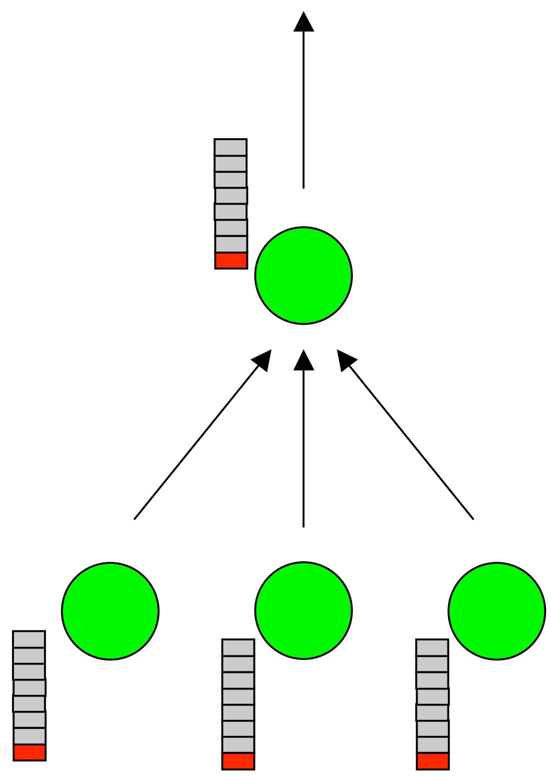
- Densely deployed sensors
  - Persistent hotspots
  - Congestion occur near the sources
- Sparsely deployed sensors, low rate
  - Transient hotspots
  - Congestion anywhere but likely far from the sources, towards the sink
- Sparsely deployed sensors, high rate
  - Both persistent and transient hotspots
  - Hotspot distributed throughout the network

Some ideas for CC in WSN

- Congestion detection
  - Monitor output buffer/queue size
  - Monitor channel busy time, estimate channel's load
  - Monitor the inter-packet arrival time (data, ctrl)
- Congestion notification
  - Explicit congestion notification in packet header, then broadcast (but then energy-consuming!)
- Congestion control
  - Dynamic reporting rate depending on congestion level
  - In-network data reduction techniques (agressive aggregation) on congestion

Detecting congestion?

- **Queue occupancy-based congestion detection**
  - Each node has an output packet queue
  - Monitor instantaneous output queue occupancy
  - If queue occupancy exceeds α, indicate *local congestion*



Queue occupancy not enough!

- ESRT uses only buffer occupancy
- CODA uses
  - Channel sampling: sample channel at appropriate time to detect congestion
  - Report Rate from sources: Fidelity measurement – observed over a long period

C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, “CODA: Congestion detection and avoidance in sensor networks,” in Proceedings of ACM Sensys’03

Channel sampling

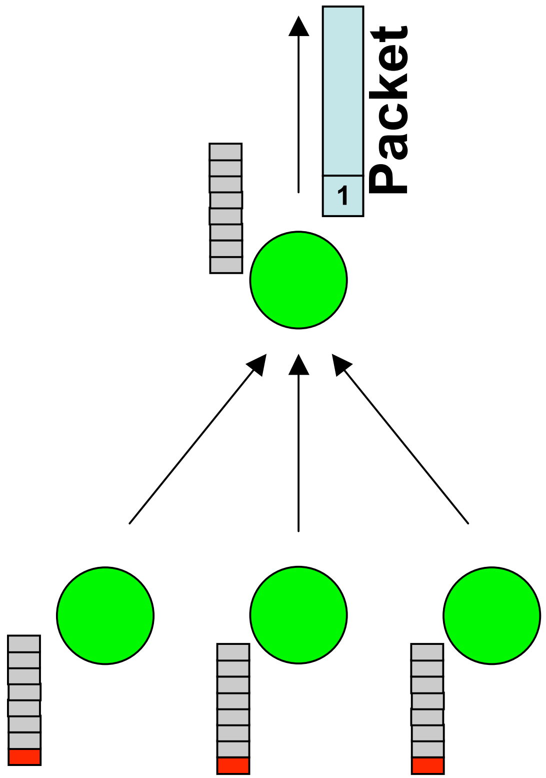
- Channel status (busy/idle) measured for N consecutive sensing epochs of length E with a predefined sampling rate ➔ Φ_n : # of busy(idle) / epoch
- $\Phi_{n+1} = \alpha\, \Phi_n + (1-\alpha)\, \Phi_n$ (EWMA)
- Experimental validation for
  - $N \in \{2,3,4,5\}$
  - $E \in \{100ms, 200ms, 300ms\}$
  - $\alpha \in \{0.75, 0.80, 0.85, 0.95\}$

CODA overview

- Combination of backpressure (fast time scale) with closed-loop congestion control.
- Backpressure targets "local" congestion, whereas closed-loop regulation targets persistent congestion.
- Backpressure is cheaper/simpler since it's open loop.
- Congestion control requires a feedback loop -> Uses ACK from sink to self-clock

Detect then backpressure!

- Open-loop
- Hop-by-hop backpressure
  - Every packet header has a congestion bit
  - If locally congested, set congestion bit
  - Snoop downstream traffic of parent

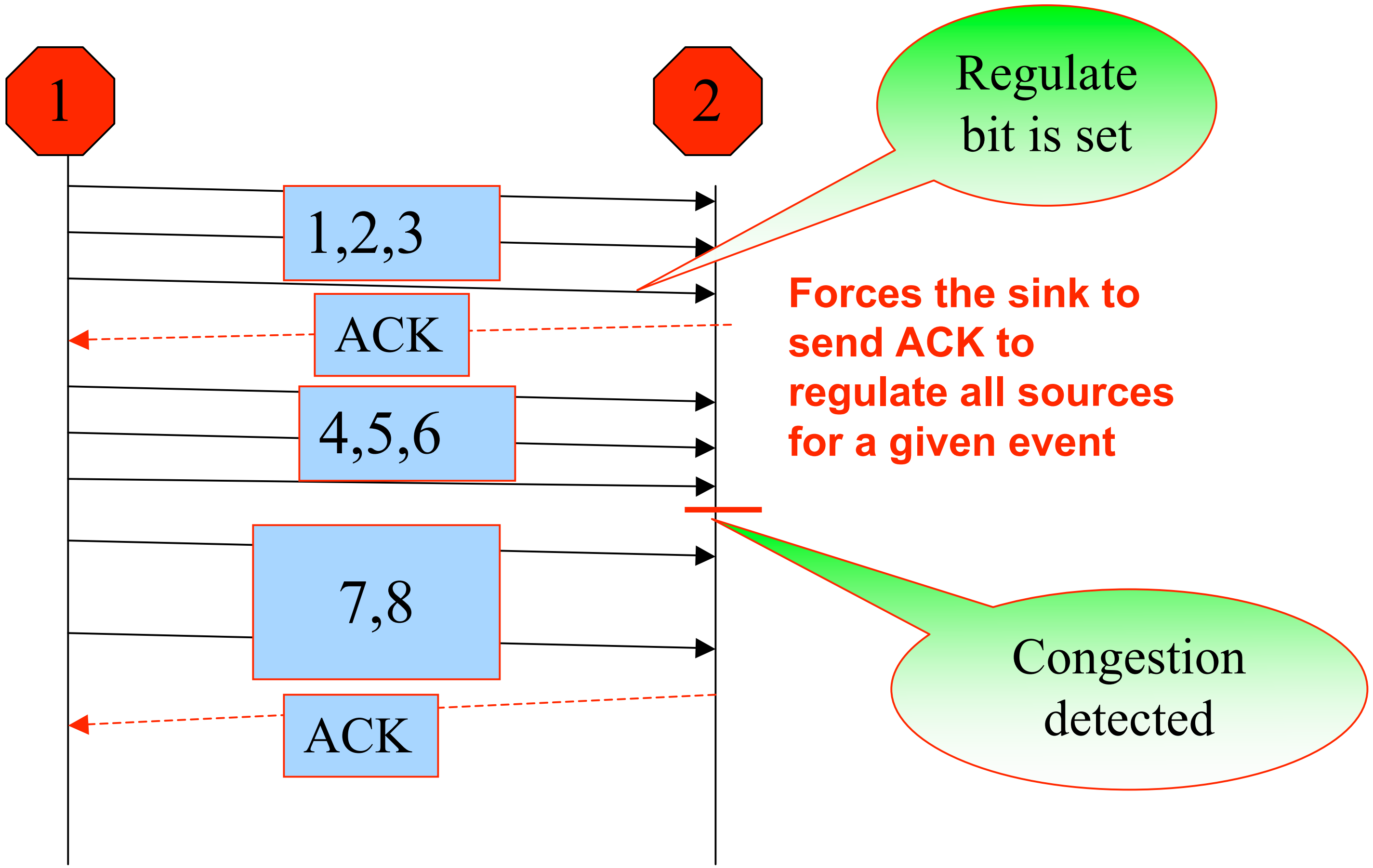


C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, “CODA: Congestion detection and avoidance in sensor networks,” in Proceedings of ACM Sensys’03

Closed Loop Multi-Source Regulation

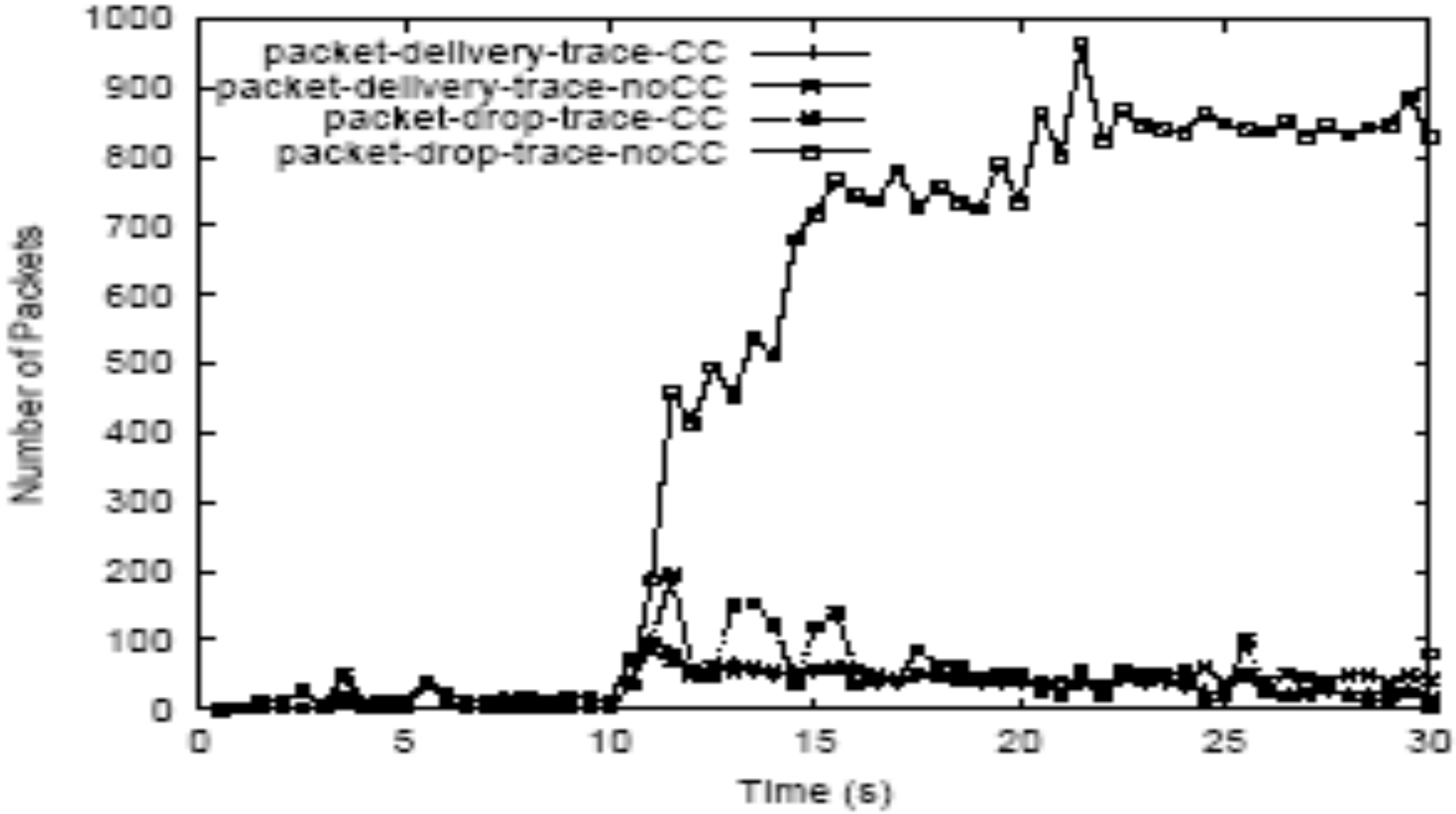
When the source event rate exceeds a given threshold, it set a « regulate bit »

Event rate is decreased if the number of ACK over a predefined period decreases.



Forces the sink to send ACK to regulate all sources for a given event

Simulation Results (Dense Source , High Rate)



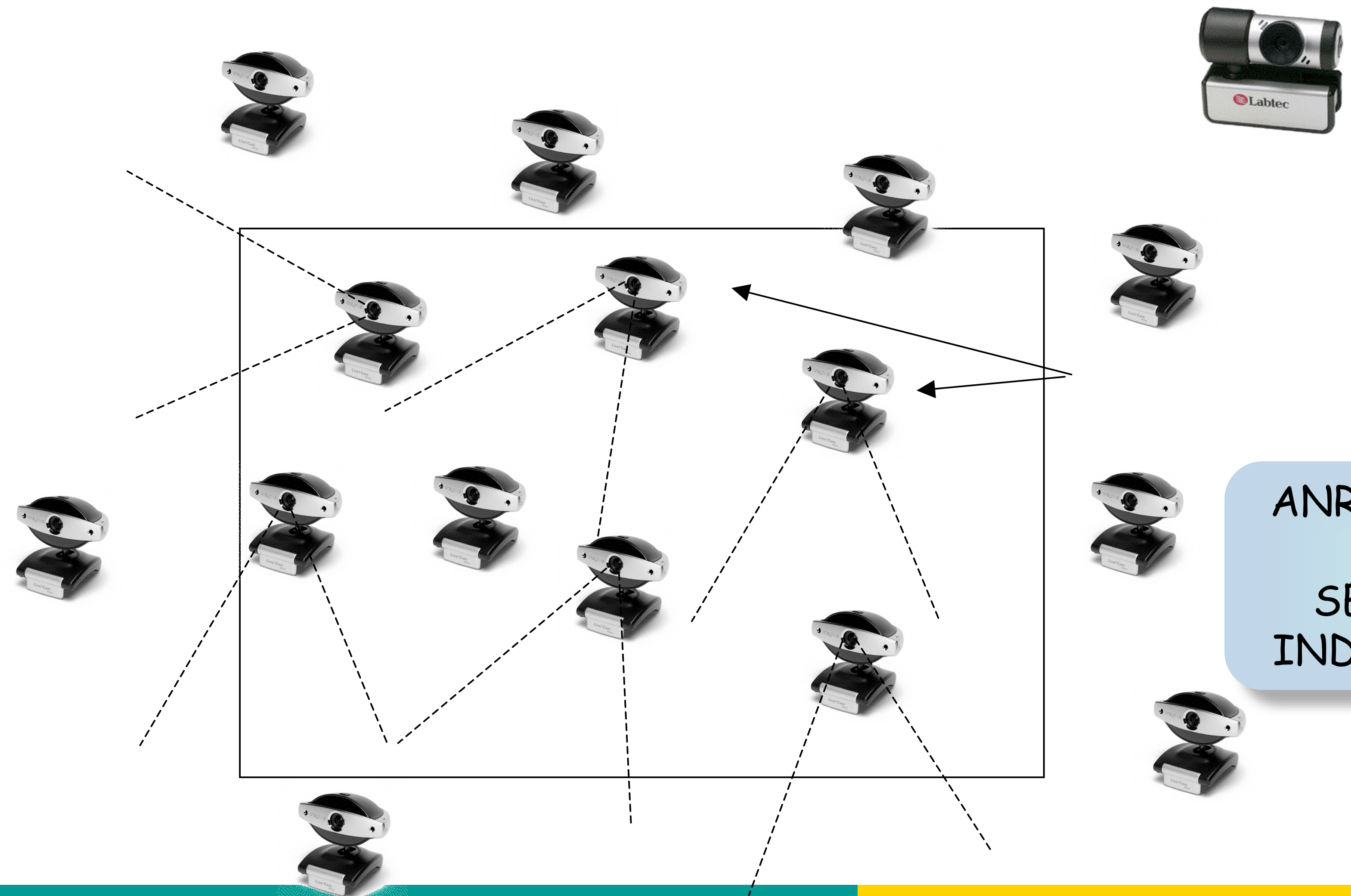
- Random Network Topologies with network size from 30 to 120 nodes
- 2Mbps IEEE 802.11 MAC (RTS/CTS are disabled)
- Fixed Work load, 6 Sources and 3 Sinks

Ex: PCCP

- Uses mean packet inter-arrival time t_a and mean packet service time t_s at the MAC layer.
- Both values are computed using EWMA process
- $d=t_s/t_a$, le congestion degree
  - d>1, experienced congestion
  - d<1, incoming rate below outgoing rate

C. Wang, B. Li, K. Sohraby, M. Daneshmand, Y. Hu, "Upstream Congestion Control in Wireless Sensor Networks Through Cross-Layer Optimization" in IEEE JSAC, 20(4), May 2007

Wireless Video Sensor Network (WVSN)



ANR PROJECT

SEEK FOR INDUSTRIAL!

Which CC for WMSN? (1)

- WSN: scalar data
- Wireless Multimedia Sensor Networks add video, audio for
  - Enlarging the view
    - Field of View of single camera is limited
    - Multiples camera overcome occlusion effects
  - Enhancing the view
    - Can help disambiguate cluttered situations
  - Enabling multi-resolution views

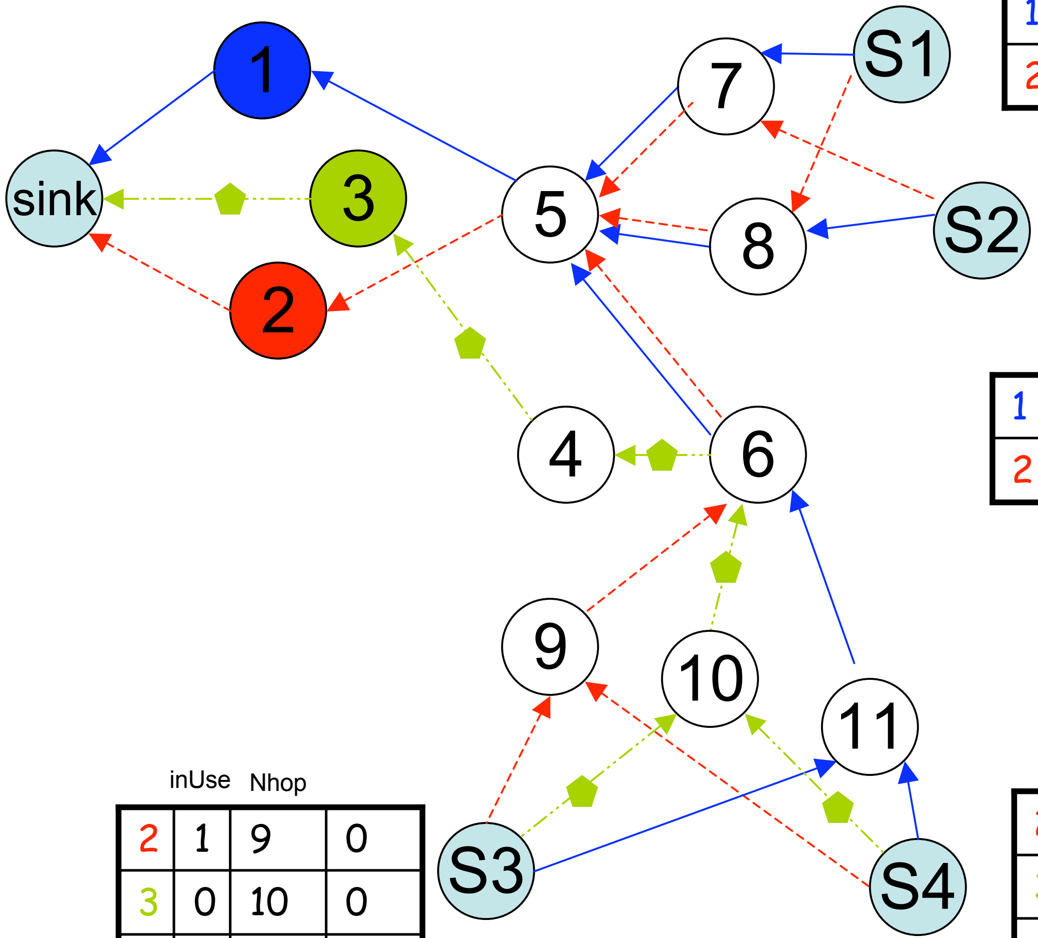
Which CC for WMSN? (2)

- Reliability should be enforced at the packet level
  - Some packets are more important than others in most of video coding schemes
- Collaborative in-network processing
  - Reduce asap the amount of (redundant) raw streams to the sink

Lightweight Load-Balancing

- Keep sending rate, thus video quality, constant: surveillance & critical applications
- Suppose
  - path diversity: path-id
  - Congestion notifications from network: CN(node-id, path-id)
- Load balance in video traffic on multiple paths

Path diversity



	inUse	Nhop	
1	1	7	0
2	0	8	0

	inUse	Nhop	
1	1	8	0
2	0	7	0

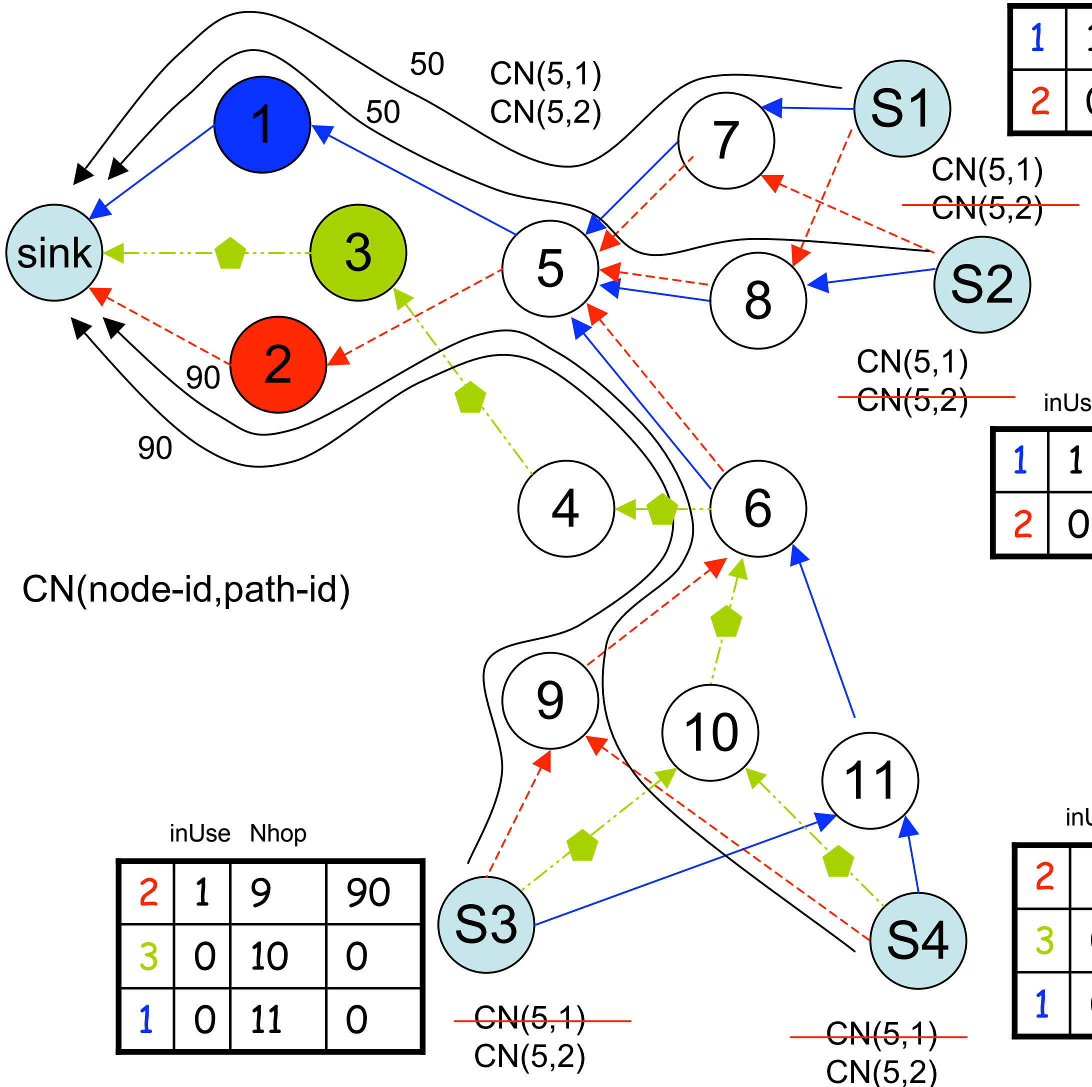
	inUse	Nhop	
2	1	9	0
3	0	10	0
1	0	11	0

	inUse	Nhop	
2	1	9	0
3	0	10	0
1	0	11	0

Load-balancing modes

- Mode 0
  - no load-balancing
- Mode 1
  - uses all available paths from the beginning
- Mode 2
  - starts with 1 path, for each CN(nid,pid) adds a new path
- Mode 3
  - starts wih 1 path, for each CN(nid,pid) balance uniformly trafic load of path pid on all available paths (including path pid)

Node 5 is congested



inUse Nhop (S1)

1	1	7	50
2	0	8	0

inUse Nhop (S2)

1	1	8	50
2	0	7	0

inUse Nhop (S3)

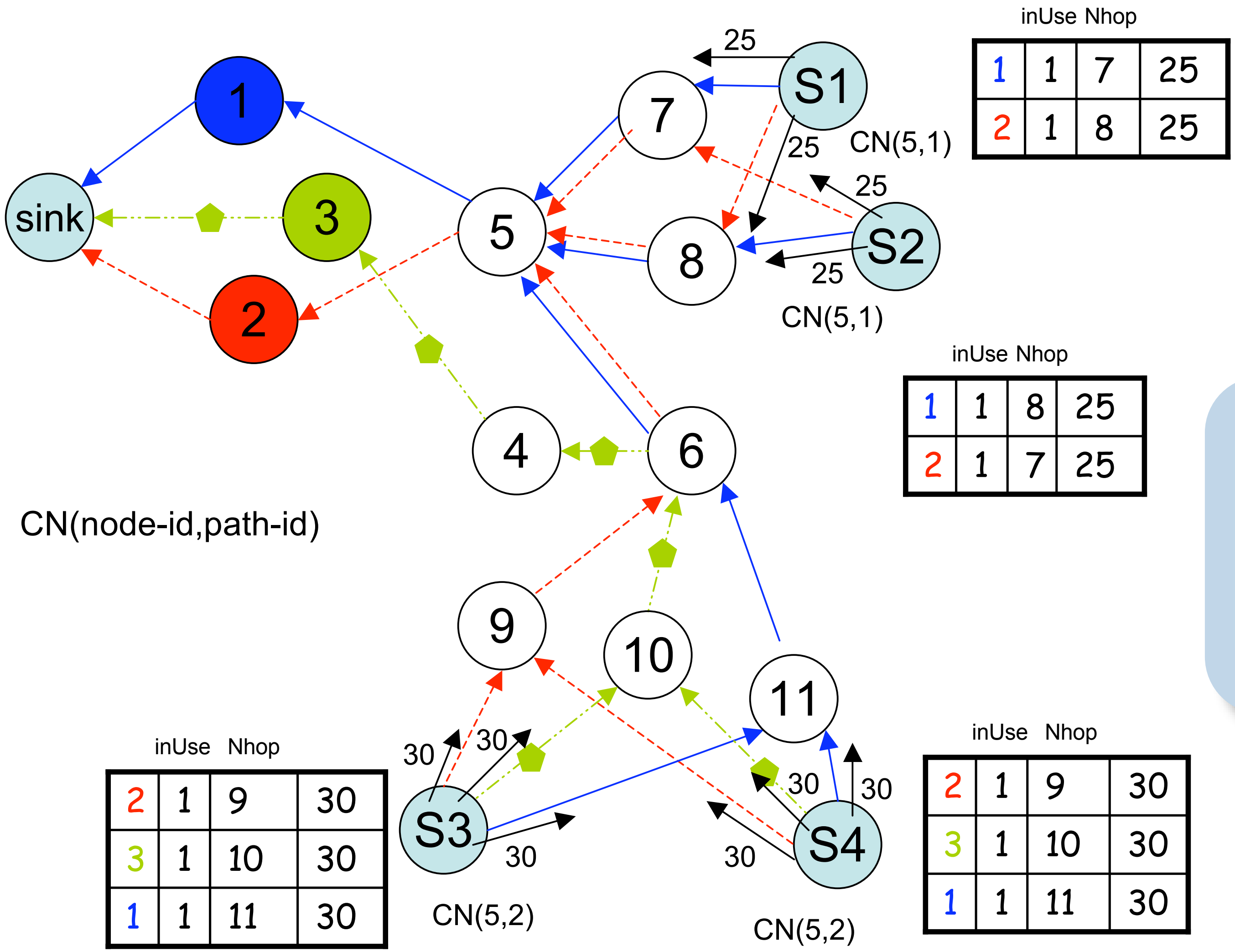
2	1	9	90
3	0	10	0
1	0	11	0

inUse Nhop (S4)

2	1	9	90
3	0	10	0
1	0	11	0

Only active paths are concerned

Node 2 becomes congested



1	1	7	25
2	1	8	25

1	1	8	25
2	1	7	25

2	1	9	30
3	1	10	30
1	1	11	30

2	1	9	30
3	1	10	30
1	1	11	30

Load balance excess trafic of available paths

Some results (1)

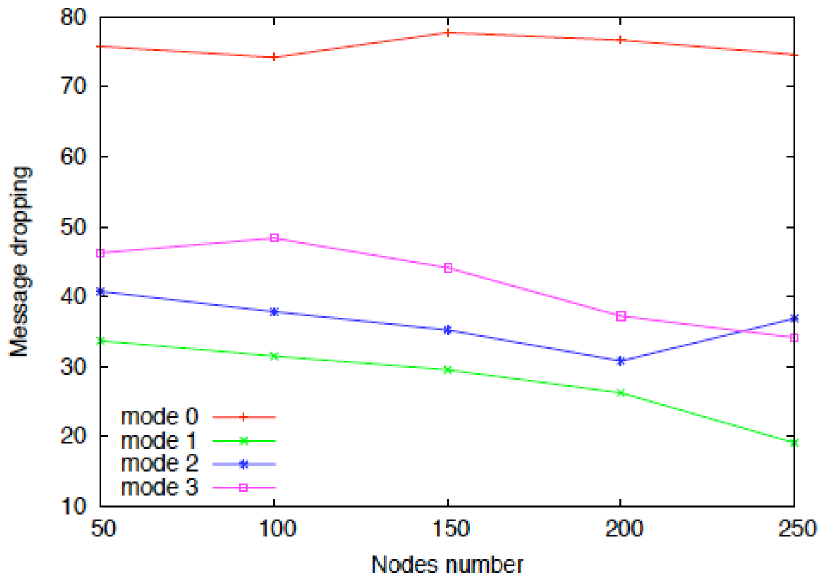


Fig. 4. Message dropping rate at sensor queues

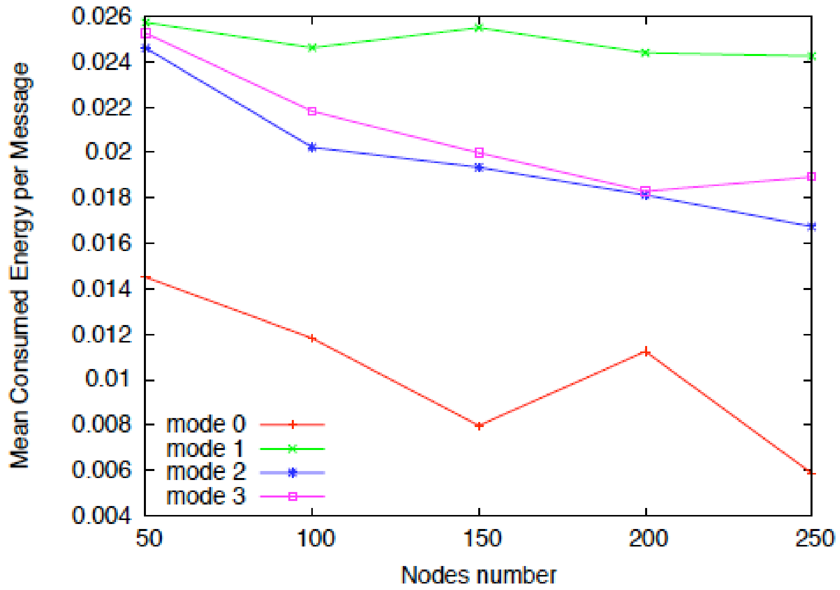


Fig. 7. Mean consumed energy per received packet

Some results (2)

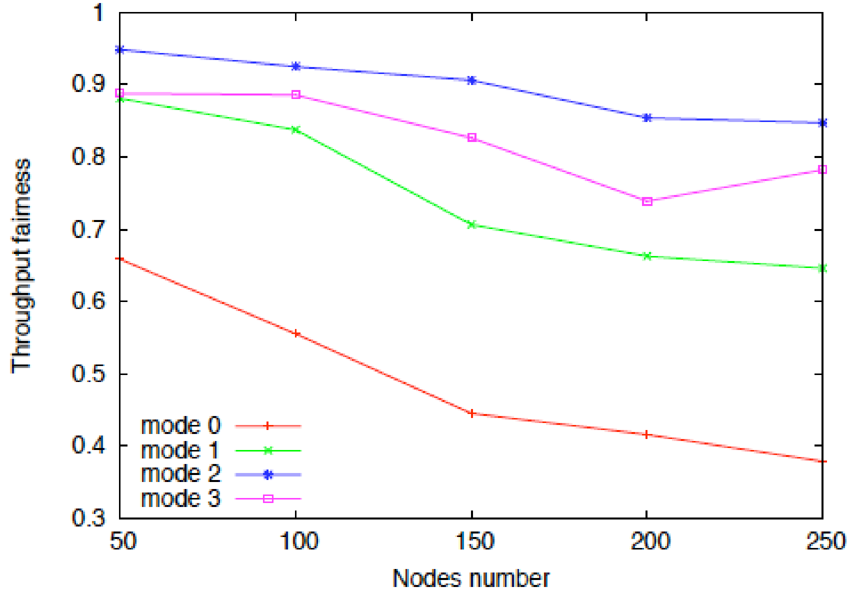


Fig. 5. Rate fairness among sources

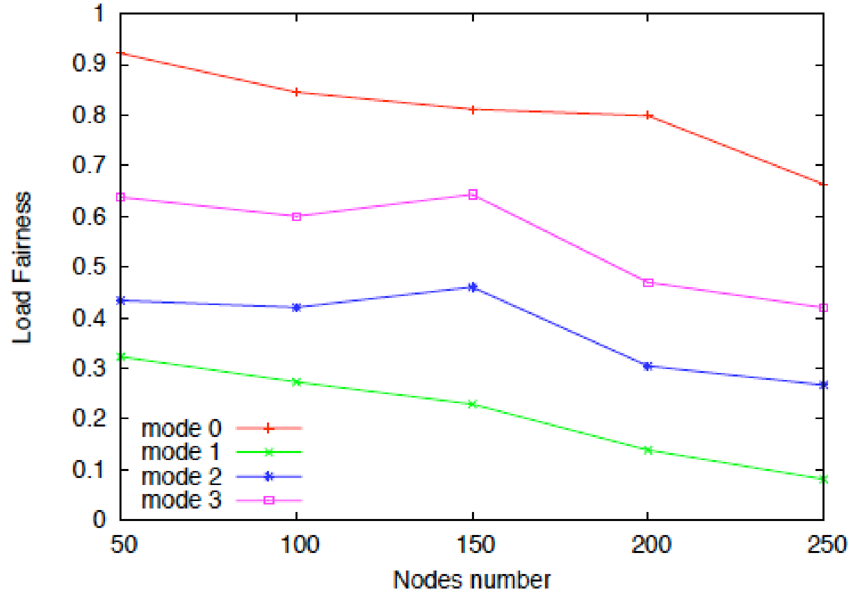


Fig. 6. Load fairness among active sensors

Conclusions

- Transport protocols are essentially application-aware
- Efficient congestion detection mechanisms are still a hot-topic work, but once again maybe application-specific
- Case of video flows challenging, especially for critical applications