

PhD defense, June 16, 2005
**Addressing the Challenges for TCP over
Multihop Wireless Networks**

Ruy de Oliveira
Universität Bern

Outline

- > Motivation
- > Multihop networks
- > Transmission Control Protocol (TCP)
- > TCP challenges and proposed solutions
- > ***Packet loss discrimination using Fuzzy Logic***
- > ***Smart acknowledgment strategy***
- > Conclusions and outlook

Motivation

- > More and more Internet traffic are foreseen in future wireless communications based on IEEE 802.11 standard

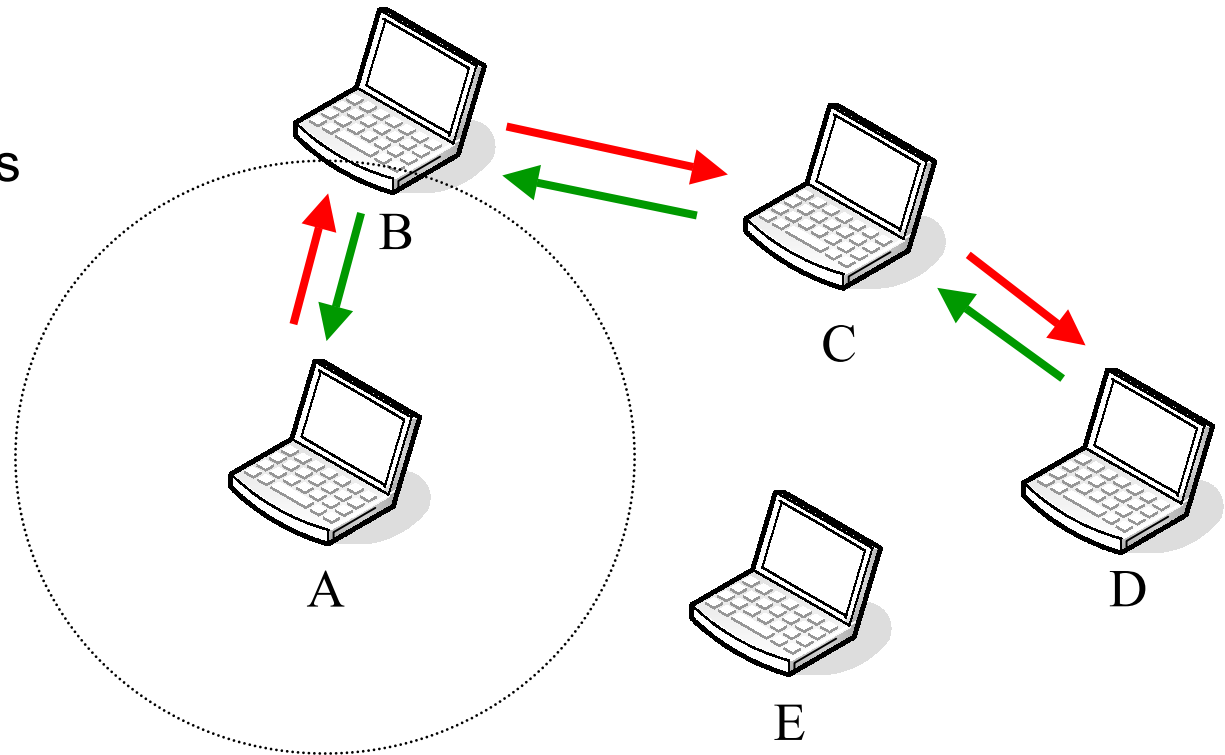
- > Wireless communications are quite different from the wired world
 - Wireless medium are inherently noisy
 - Bandwidth in wireless channels is a scarce resource

- > The widespread use of TCP in the Internet motivates its extension to wireless

Multihop Wireless Networks

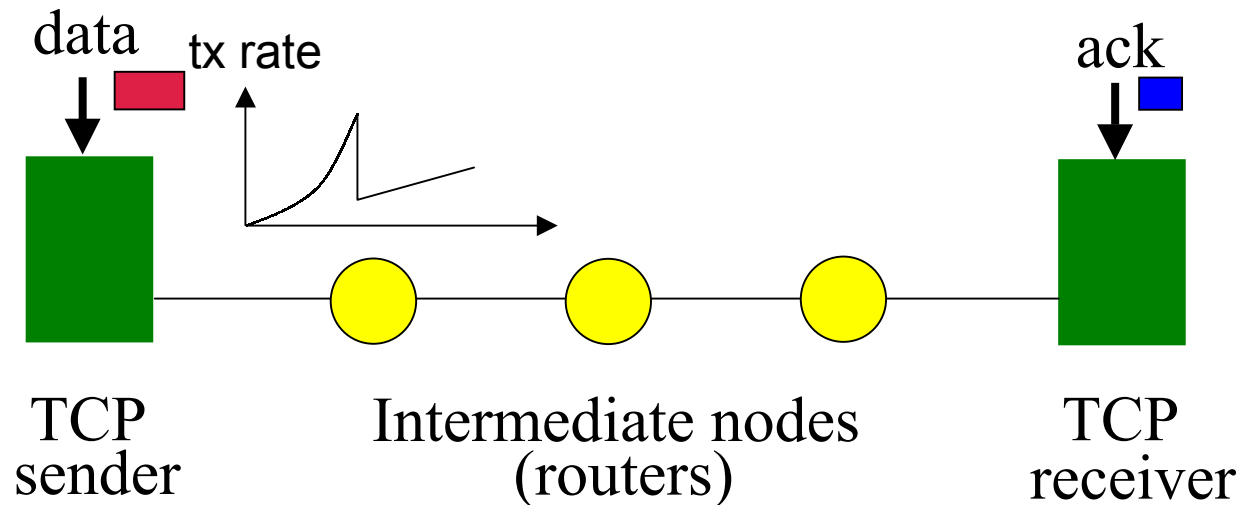
Multihop ad hoc network

- > No fixed infrastructure
- > Communication range is increased by multiple links
- > Nodes forward data to each other
- > Nodes may move
- > Low energy resources



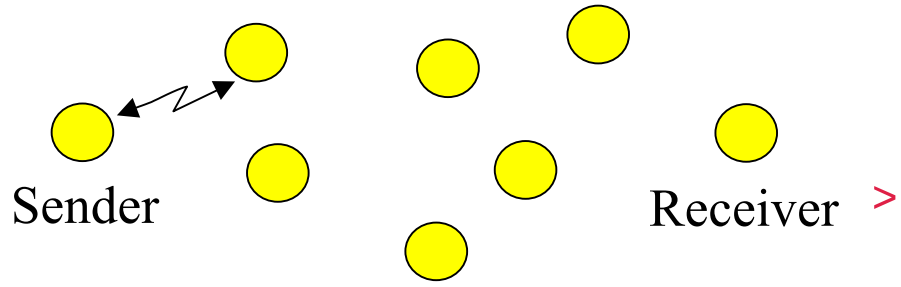
Transmission Control Protocol (TCP)

- > Provides reliable data delivery (bidirectional flow)
- > Works on an end-to-end basis
- > Continuously probes the network for resources
- > Retransmission: - *retransmit timeout (timer expiration)*
- *fast retransmit (3 duplicate ACKs)*

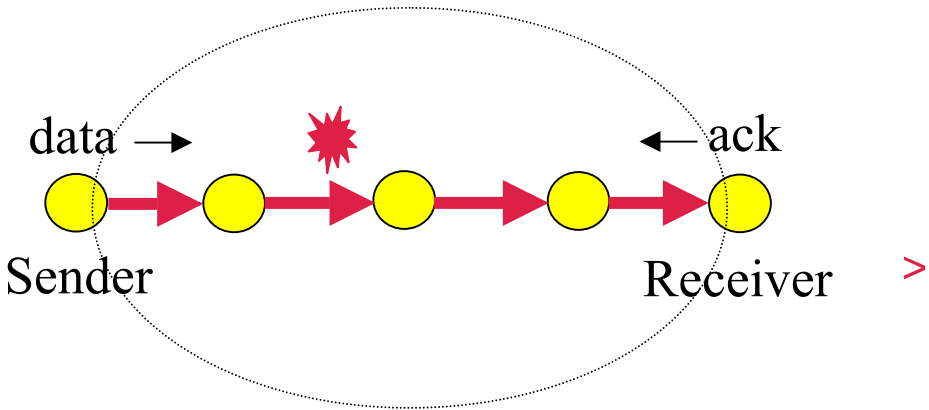


Challenges and Solutions for TCP

Wireless network



Lossy channel
↓
Packet loss discrimination



Traffic redundancy
↓
Smart ACK management

First Main Contribution: Packet Loss Discrimination

- > Purpose:
 - To distinguish between congestion and medium induced errors

- > How:
 - Evaluating packet delay variations (Round-Trip Time) inside the network

- > Advantages:
 - Very lightweight (only a few logic operations)
 - Independent of explicit signaling from the network (end-to-end)

- > Assumptions:
 - Fixed packet size
 - Conventional lower layer protocols
 - Known RTT distributions as a function of the # of hops

Design Rationale

Why RTT measurements?

- > TCP relies naturally on round-trip time measurements for taking decisions
- > It is intuitive that such measurements may be useful in inferring the internal network state

Difficulties

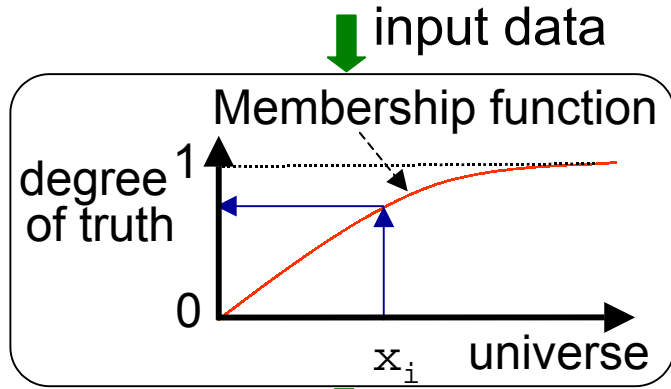
- > RTT measurements may contain overlapping values under certain conditions

Then

- > Appropriate tool may perform good reasoning on the measured data → Fuzzy logic

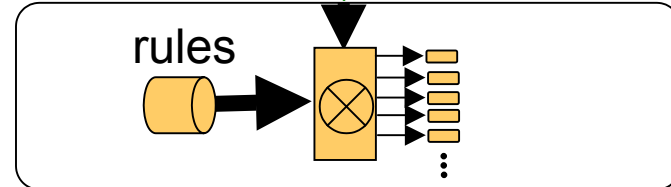
Fuzzy Logic (FL)

Fuzzification



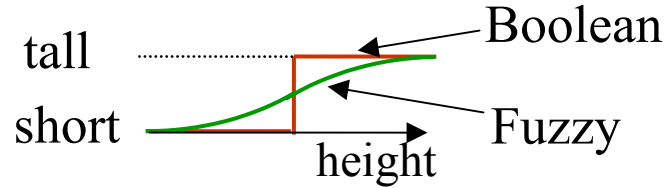
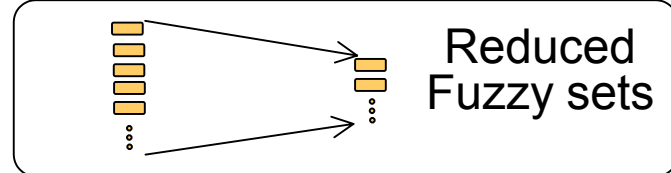
> It uses membership functions and rules to reason about the data

Inference



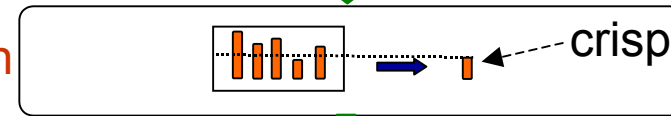
> FL avoids rigid boundaries
ex. "people height":
is 1.79m short and 1.80m tall?

Aggregation



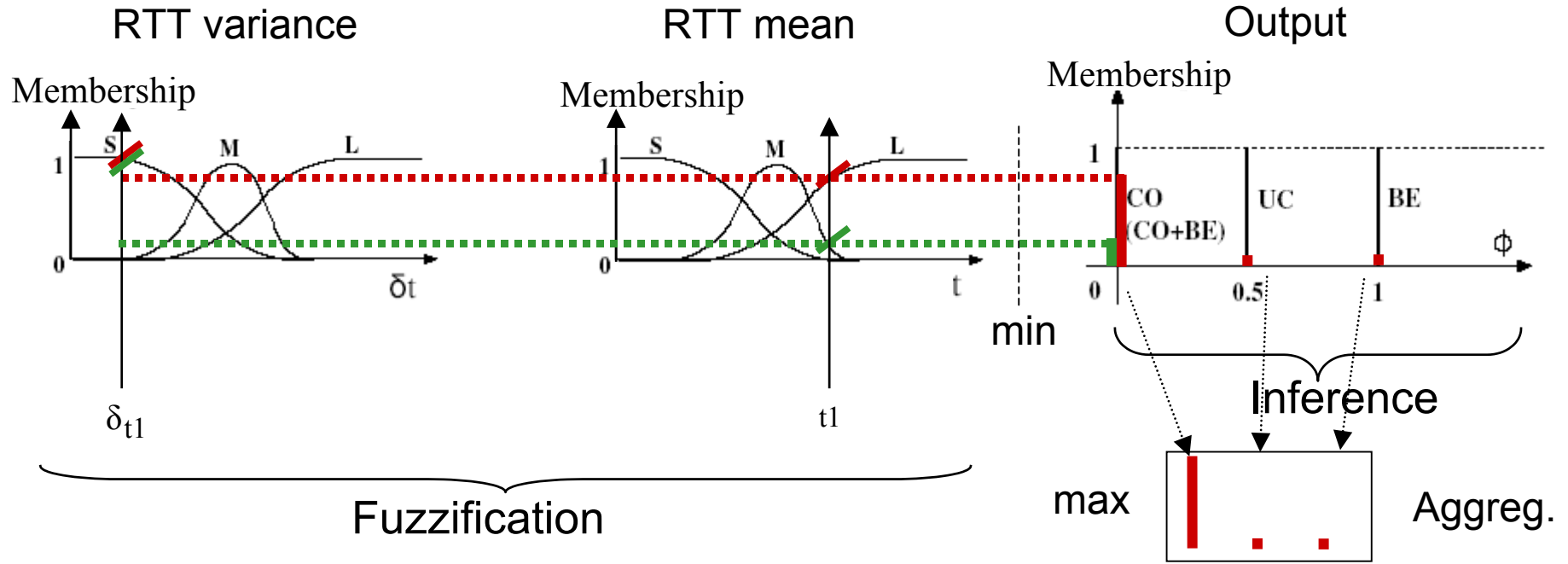
> FL is proper for pattern recognition tasks

Defuzzification



output data

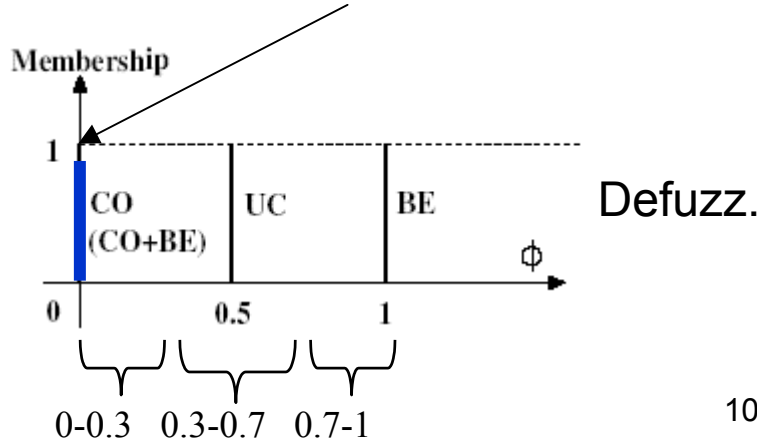
The Inference Procedure of the Fuzzy Engine



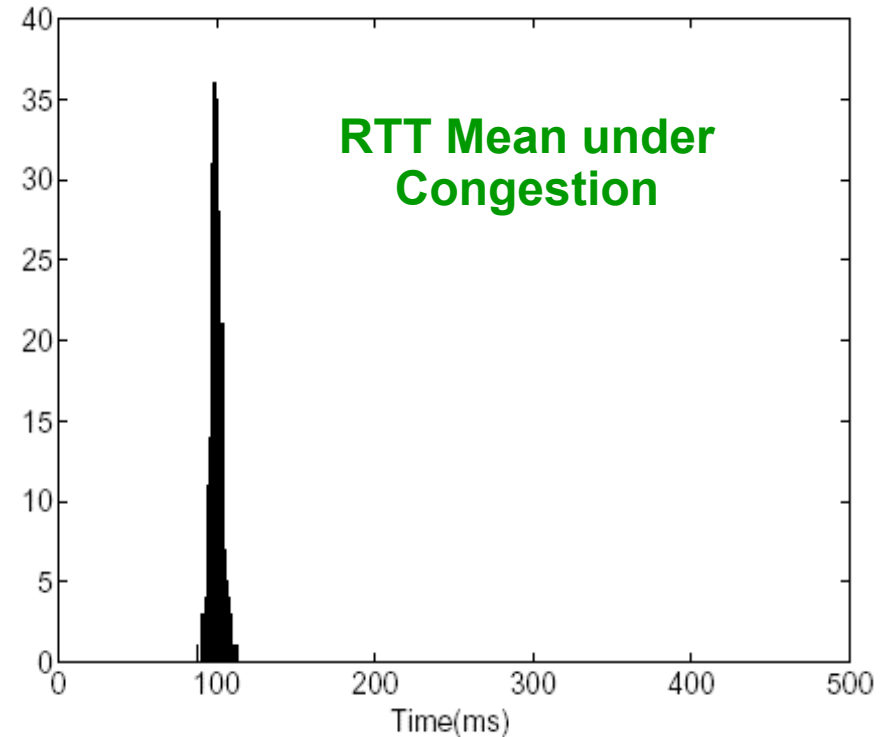
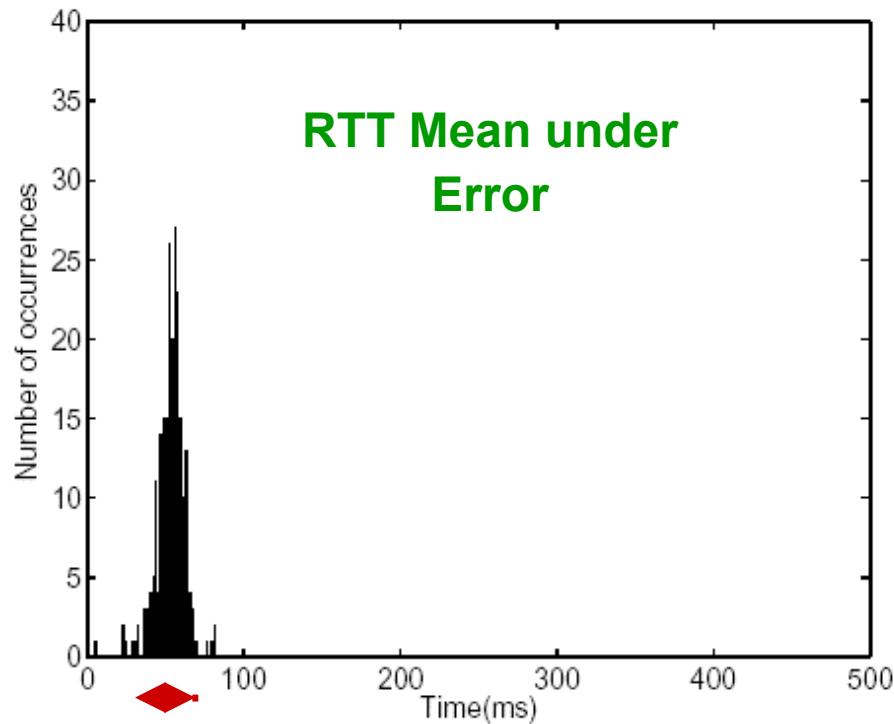
Fuzzy rules output Φ

Var(δ) \ mean (t)	S	M	L
S	BE	CO	CO
M	BE	UC	CO
L	BE	BE	CO

S = Small
M = Medium
L = Large
CO = Congestior
UC = Uncertain
BE = Bit Error



Settings of the Membership Functions for RTT mean by Simulations (3-hop)



Being conservative: $RTT > 60 \rightarrow$ congestion
 $RTT < 40 \rightarrow$ medium error

Fuzzy settings: Gaussian curve center is 50 and width is 20

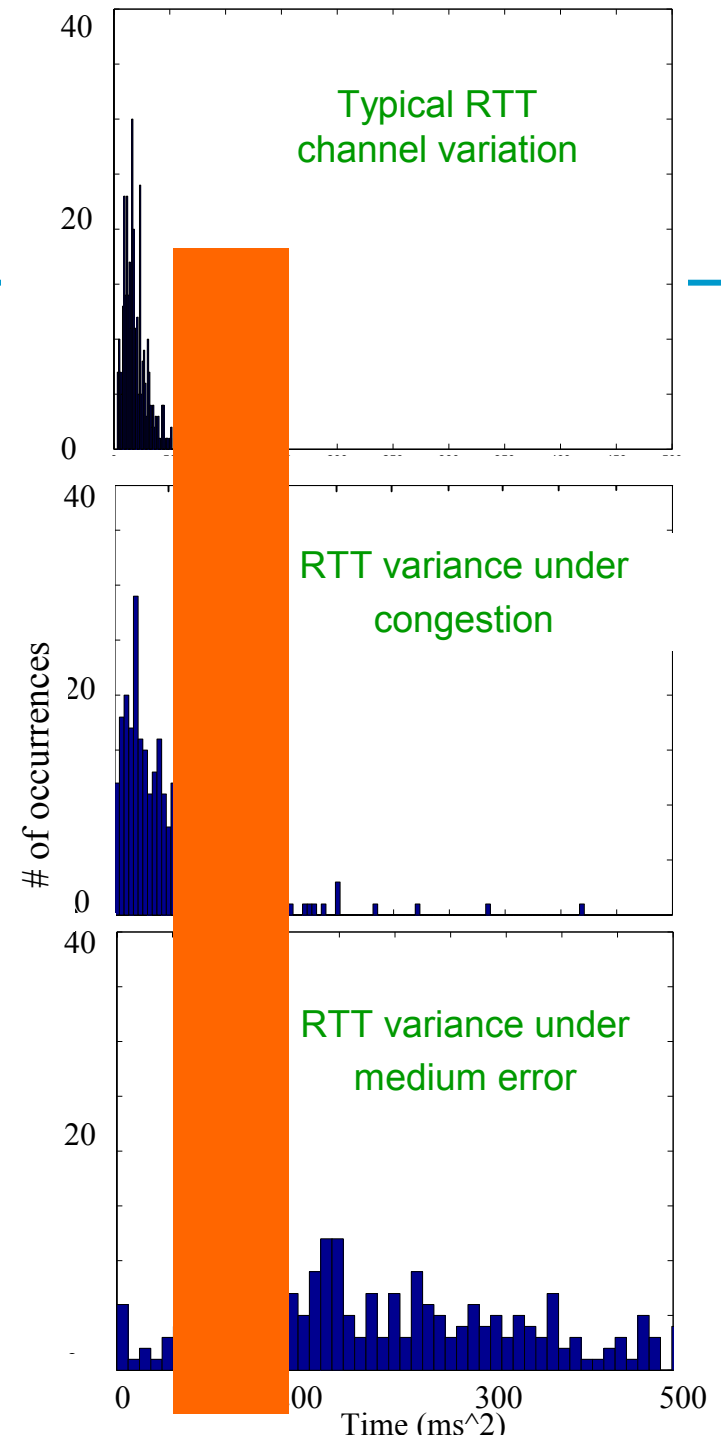
Settings of the Membership Functions for RTT var. (3-hop)

- > Congestion and medium error have distinct distributions for RTT variance
- > There is an overlapping area between 50-150 (ms^2)
- > Below 50 the values are clearly low and above 150 clearly high



Fuzzy settings:

- > Gaussian curve: center is 100 and width is 50.
- > Universe of discourse is 500



Fuzzy Engine Features

- > **Fuzzy Input**
 - RTT mean (t) and variance (δ)

- > **Fuzzy Rules**

Fuzzy rules output Φ

δ, t	S	M	L
S	BE	CO	CO
M	BE	UC	CO
L	BE	BE	CO

- > **Settings of the membership functions**
 - ✓ mean (t): 50 +/- 20 ms
 - ✓ Variance (δ): 100 +/- 50 ms²
 - ✓ Universe of discourse is 500 ms²
 - ✓ Output ranges:
 - CO: 0.0-0.3
 - UC: 0.3-0.7
 - BE: 0.7-1.0

Fuzzy engine output correctness

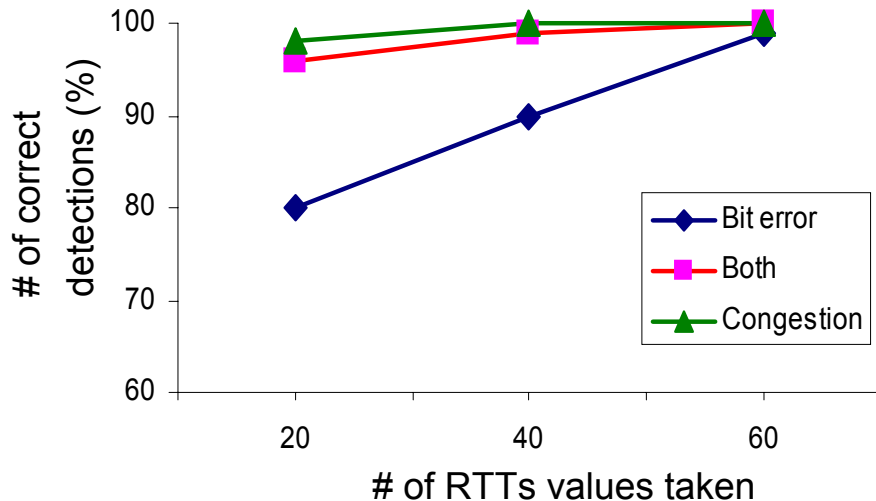
Results

- Congestion was detected in over 98% of the cases
- Bit error requires more values

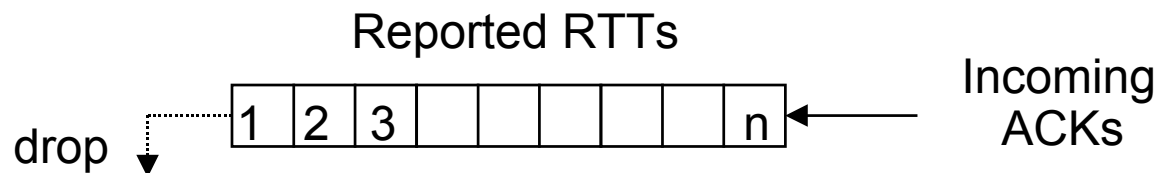
scenario

- Chain topology with 3 hops
- 3 distinct runs of 100 seconds
 - Packet error rate of 10%
 - 1 competing flow for cong.
 - Both (PER=5% + 1 c. flow)

Correct detection (%) vs. # of RTTs per sampling
Worst case

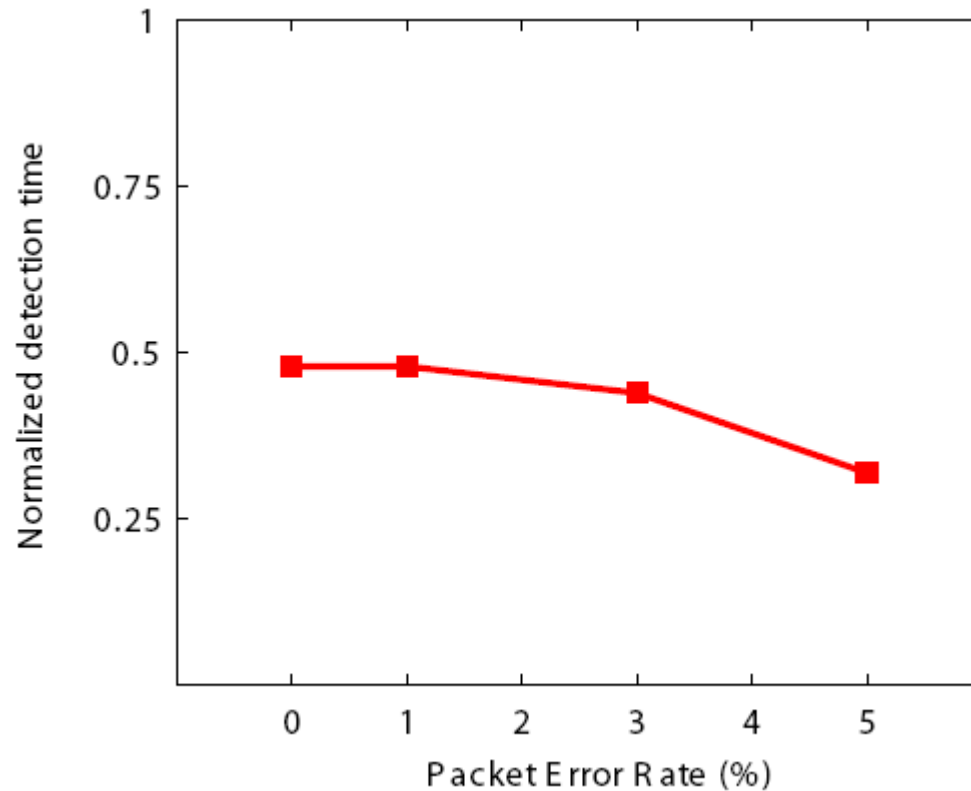


- **History of RTTs**



Fuzzy engine reaction time to abrupt congestion

Detection time vs. error rate



- > Scenario: an unconstrained channel is suddenly overloaded
- > The fuzzy engine detects incipient congestion in time

Summary of the Fuzzy Engine based Approach

- > If correctly adjusted it may provide accurate results
- > Most effective in steady state conditions
- > It is very computational inexpensive
- > It cannot always obtain a conclusive result
- > Tradeoff between speed and accuracy is always an issue
- > RTT measurements tend to become unfeasible for large number of hops

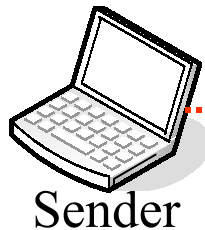
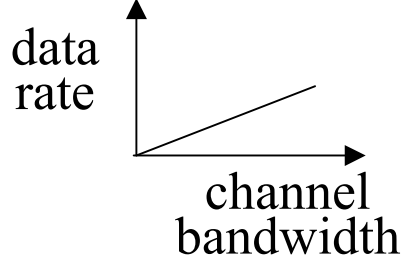
Future work

- > Different scenarios
- > More elaborated fuzzy engines
- > Adaptive algorithms for setting the fuzzy engine parameters
- > Integration with the “error recovery” mechanism

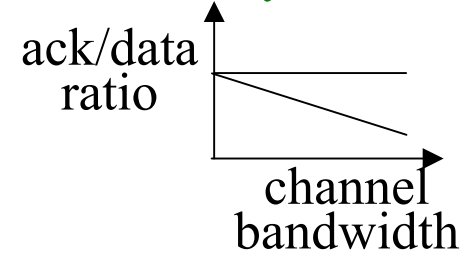
Second Main Contribution: A Smart TCP Acknowledgment Approach

Smart Acknowledgment

Congestion control



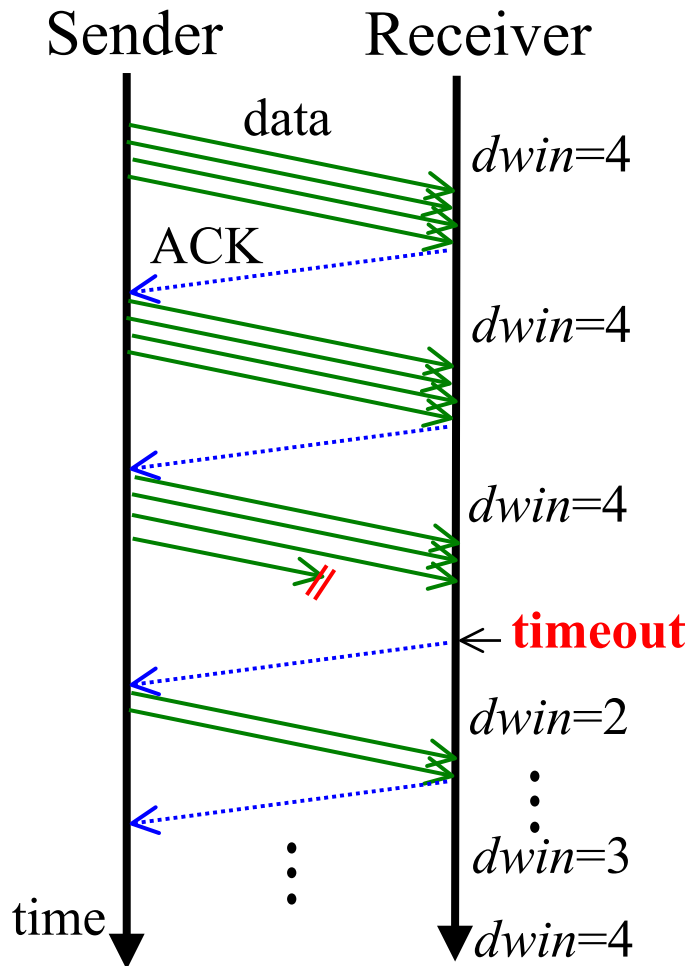
Traffic redundancy control



Dynamic Adaptive Acknowledgements (DAA)

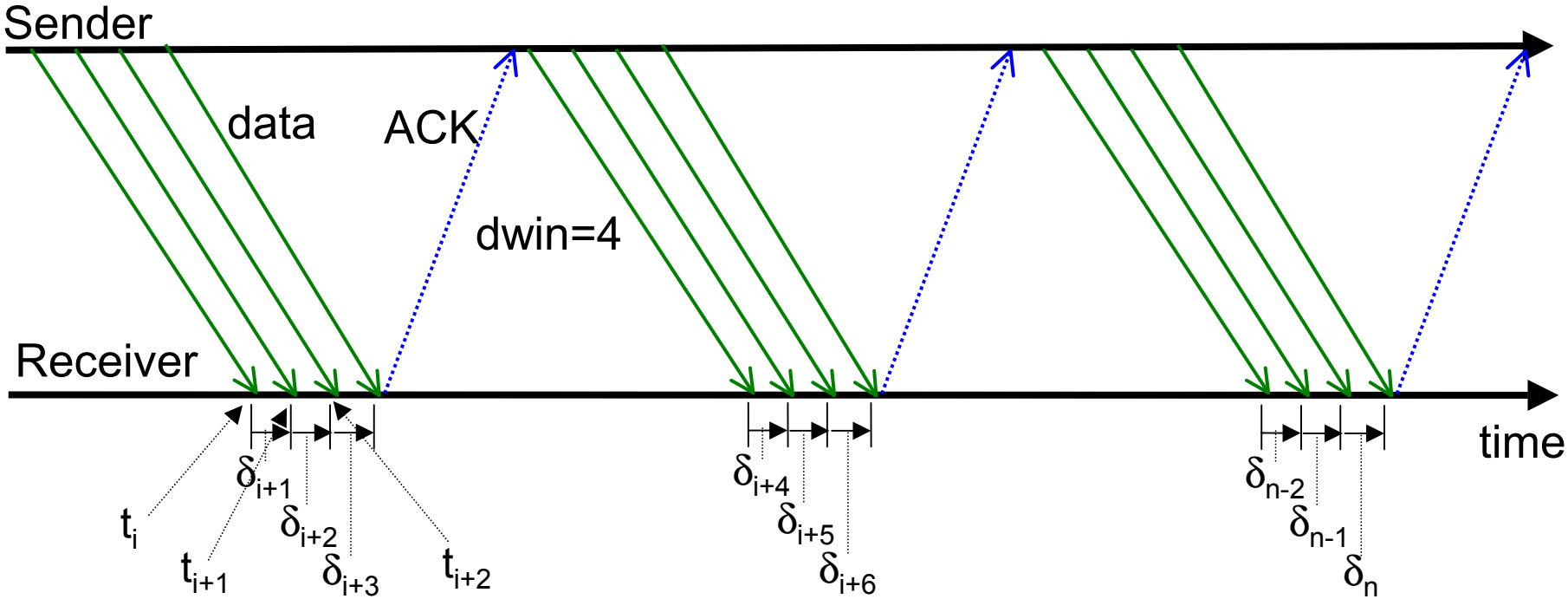
- Less ACKs leave more bandwidth for data packets
- Avoid useless retransmissions ! save energy
- Self-adjustment depending on channel conditions

DAA Mechanisms at the Receiver



- > Receiver combines up to 4 ACKs ($dwin = 4$)
- > Out-of-order packet or timeout triggers immediate ACK ! $dwin := 2$
- > $dwin$ is increased by 1 (by $\mu < 1$ if in startup phase)
 - for each correctly received data packet (limit $dwin = 4$)
- > Timeout interval T is adapted based on data packet inter-arrival time δ

Packet Inter-arrival computation



At $t_i, t_{i+1}, t_{i+2}, \dots \rightarrow$ start timer and increment delack counter (ack_count)

Low-pass filter:

$$\bar{\delta}_i = (1-\alpha) \bar{\delta}_{i-1} + \alpha \delta_i$$

$$T_i = (2+k) \bar{\delta}_i$$

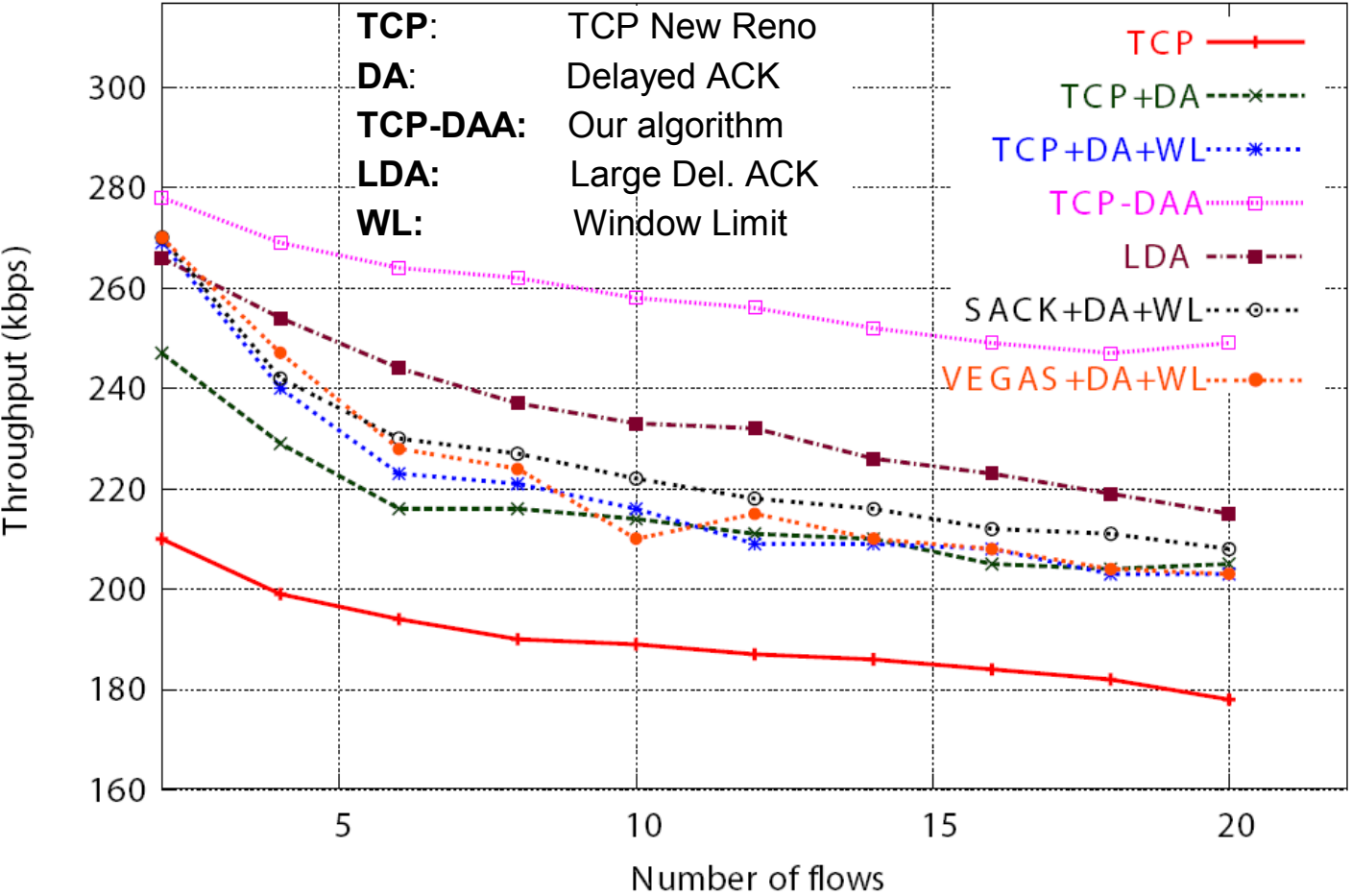
- > $\bar{\delta}_i$ = smoothed pkt inter-arrival
- > $\bar{\delta}_{i-1}$ = sampled pkt inter-arrival
- > α = smoothing factor
- > k = timeout tolerance factor
- > T_i = timeout interval at arrival of pkt i

Adjustments at the DAA Sender

- > Fivefold retransmission timeout interval ($\text{newRTO} = 5 * \text{RTO}$)
- > Fast retransmission for 2 duplicate ACKs
- > Maximum congestion window = 4
(enough to as many as 10 hops)

Throughput

TCP throughput vs. network load (hops: 5)

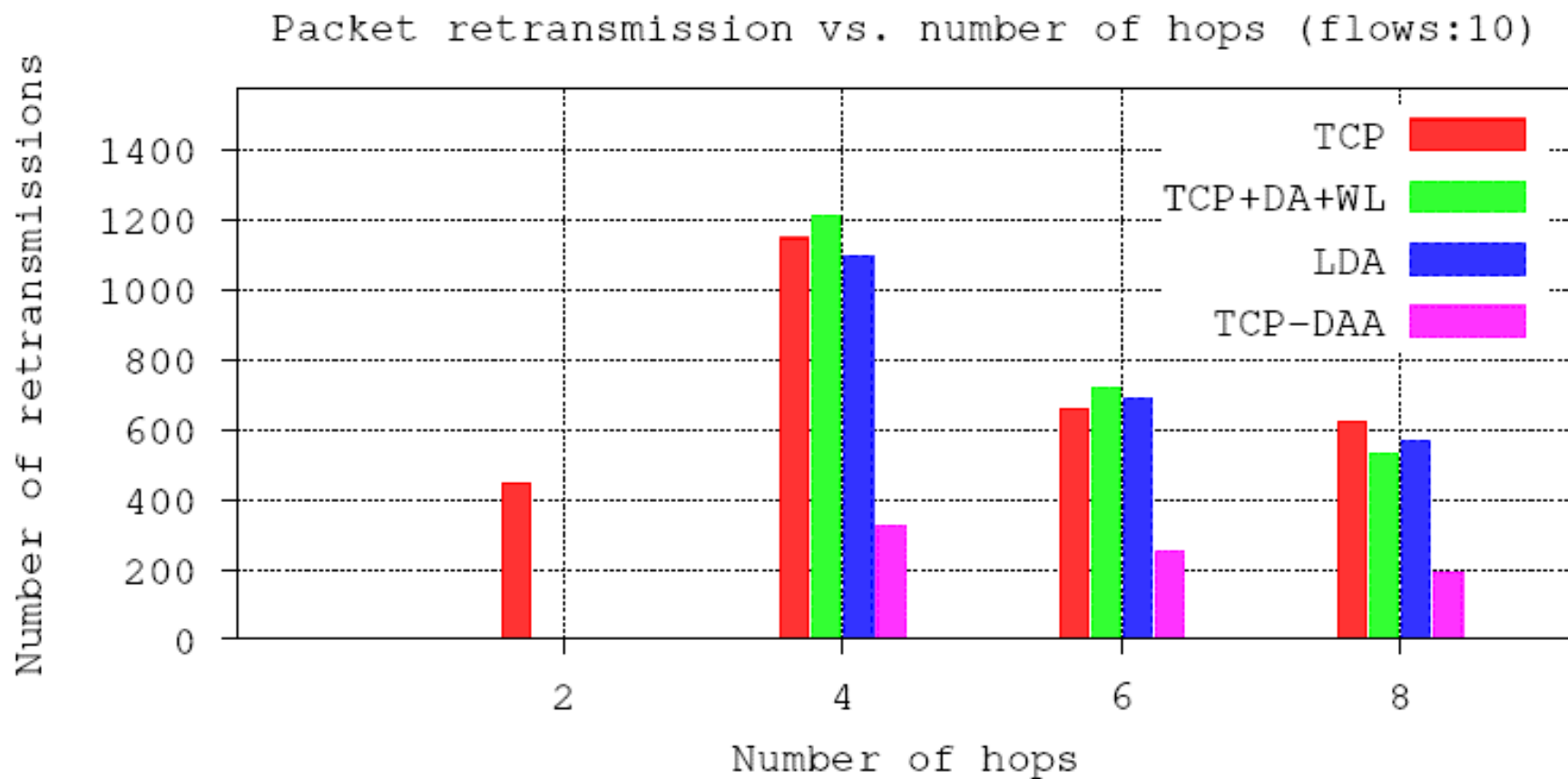


Simulation scenario

- chain topology
- loss due to collisions by MAC protocol only
- duration: 300 seconds

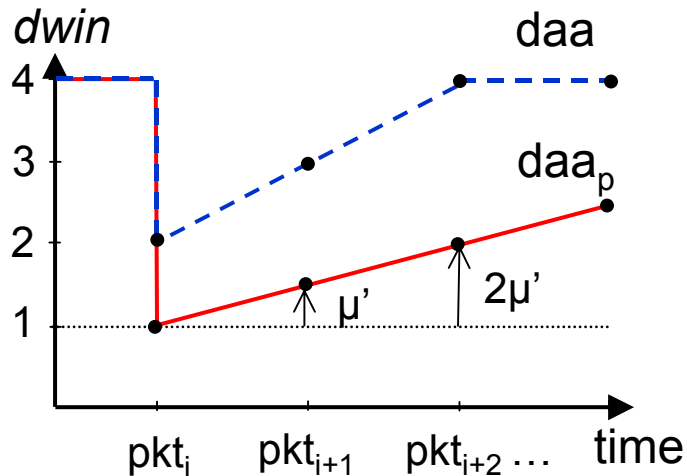
Retransmissions

- > Lower traffic overhead leads to fewer retransmissions
- > Energy consumption benefits are considerable



A Modified dwin strategy for Robustness in Highly noisy Environments (DAAp)

- > *dwin* decreases more and increases slower than basic algorithm
 - > Sender regular RTO is doubled
- (W: cumulative value of cwnd)

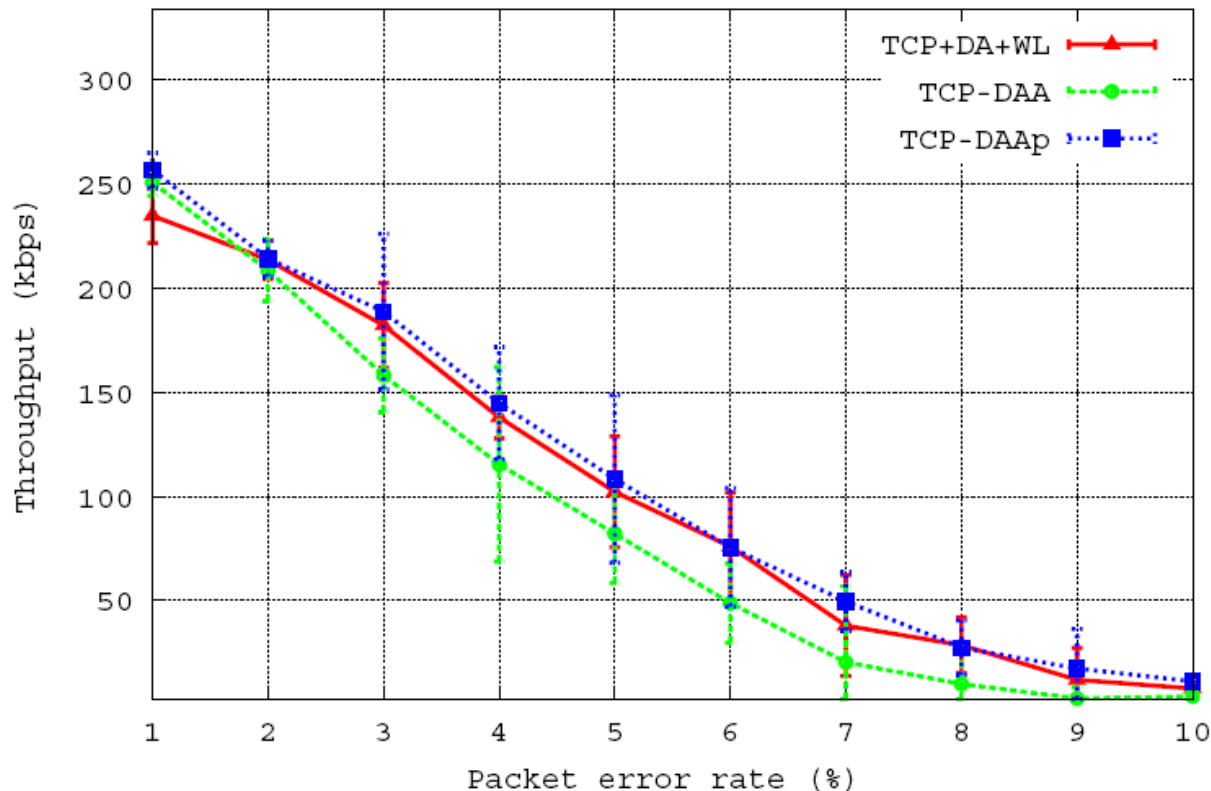


- >
$$cwnd_i = cwnd_{i-1} + \frac{1}{cwnd_{i-1}}$$
- >
$$W = cwnd_0 + \sum_{i=1}^n \frac{1}{cwnd_i}$$
- > for $W=4$ (max cwnd) $\rightarrow n=7 \rightarrow \mu' = 0.3$
- >
$$dwin_i = \begin{cases} dwin_{i-1} + \mu, & \text{if in startup} \\ dwin_{i-1} + \mu', & \text{otherwise} \end{cases}$$

Throughput under external disturbances

Comparison with adjusted TCP NewReno

TCP throughput vs. error rate (hops: 5, flows: 1)



- > DAAp prevents the sender from missing ACKs
- > DAAp sender reacts faster than DAA
- > Switching mechanism is useful

Summary of the Smart Acknowledgment Approach

- > The proposed mechanism improves throughput and energy consumption in a variety of scenarios
- > It outperforms related approaches in many situations
- > It is proper for environments under moderate loss rates

Future work

- > A tailored sender side algorithm
- > Optimization of the configuration parameters

Conclusions and outlook

- > The IEEE 802.11 is feasible for short-range networks
- > Existing protocols can be fine tuned to enhance 802.11 performance
- > Complex modeling are mostly unfeasible here
- > Proactive and reactive approaches should be combined
- > End-to-end approaches facilitate deployment

- > *An alternative MAC protocol for long-range ad hoc networks is very much needed*
- > *TCP will have to be adjusted to such a new protocol*
- > *Cross layer design seems to be the way*