A Smart TCP Acknowledgment Approach for Multihop Wireless Networks

Presented by Chun-Chieh Chang

IEEE Transactions on Mobile Computing 2007

Outline

- Introduction
- Related Work
- Dynamic Adaptive Acknowledgment
- Simulation
- Conclusion

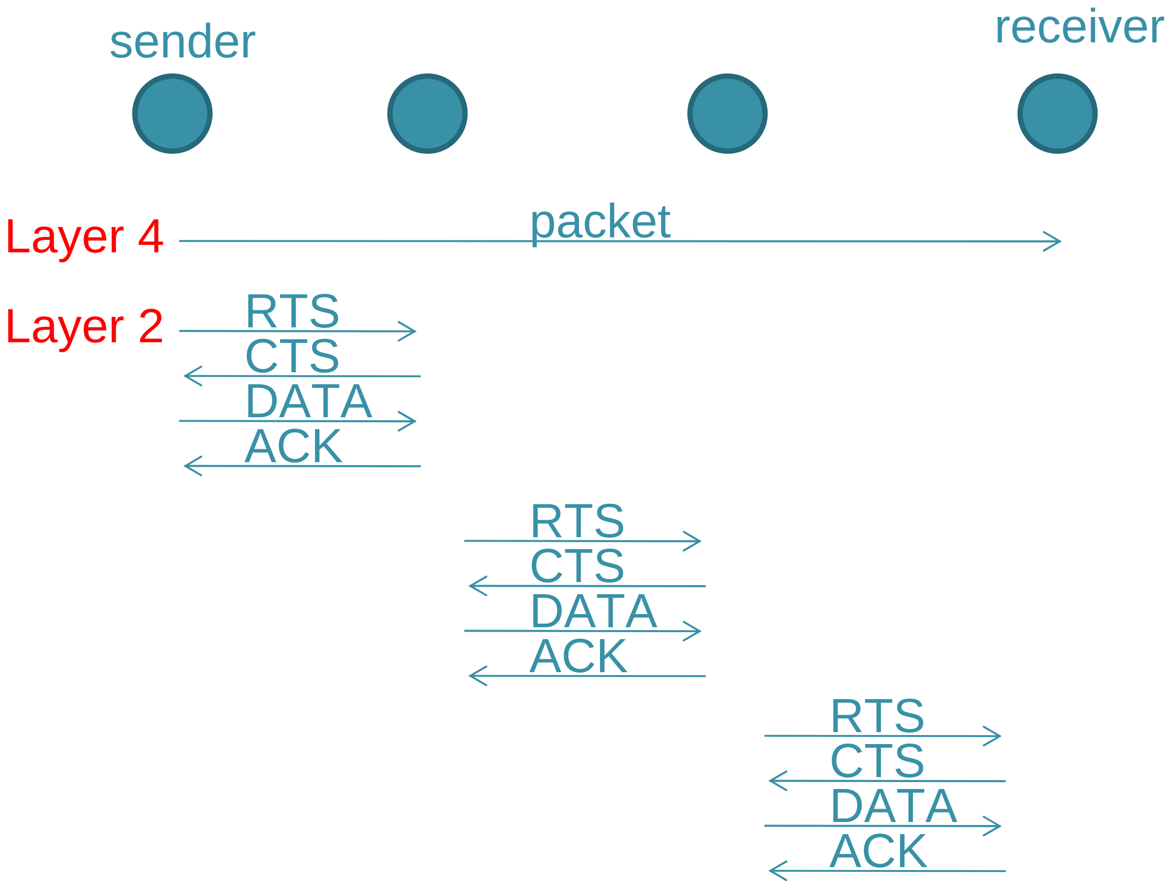
Introduction

- Goal:
  - Provide reliable end-to-end data transmission in multihop wireless networks

- TCP has been successful in wired networks due to:
  - Providing reliability on an end-to-end basis
  - Reacting dynamically to network condition

- But, TCP faces severe performance degradation over multihop wireless networks due to the 802.11 MAC protocol

TCP over 802.11 MAC Protocol



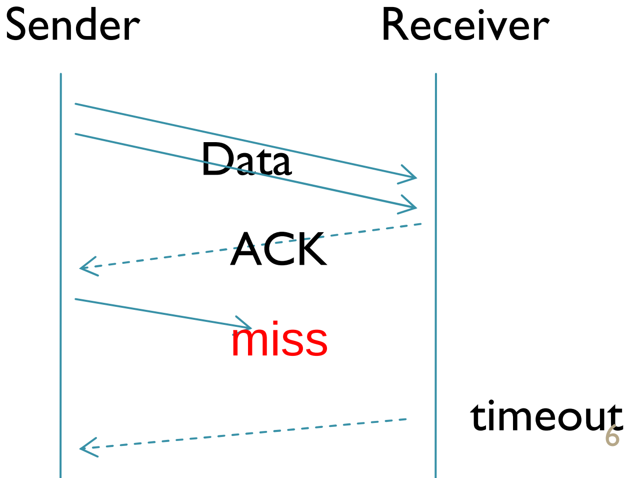
Problem

- The main problem of TCP over 802.11:
  - Excessive number of medium accesses by TCP
    - ACK packets
    - TCP retransmission policy

Related Work

- Prevent the TCP sender from slowing down in case of losses by the wireless medium

- Improve bandwidth utilization by reducing the number of medium access requests
  - fixed timeout interval



Dynamic Adaptive Acknowledgment

- Reducing the number of medium accesses
- Main ideas (on receiver)
  - delayed ACK
  - adaptively computing the timeout interval

- Environment
  - Far below 10 hops
  - Not exceed 100 nodes

Delayed ACK

cwnd: congestion window

- Delayed window (dwin)
  - Limits the maximum number of ACKs to be delayed

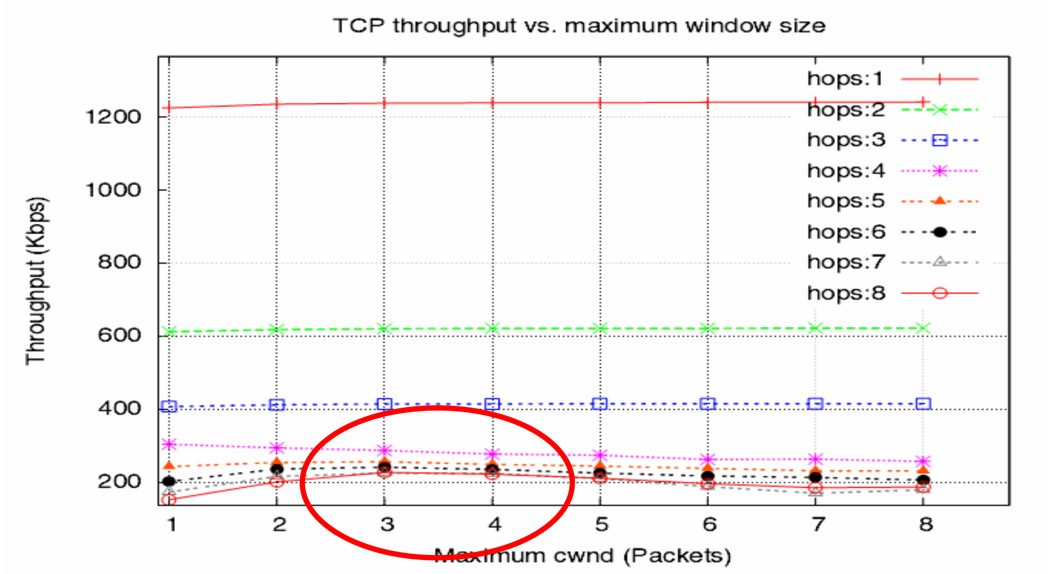


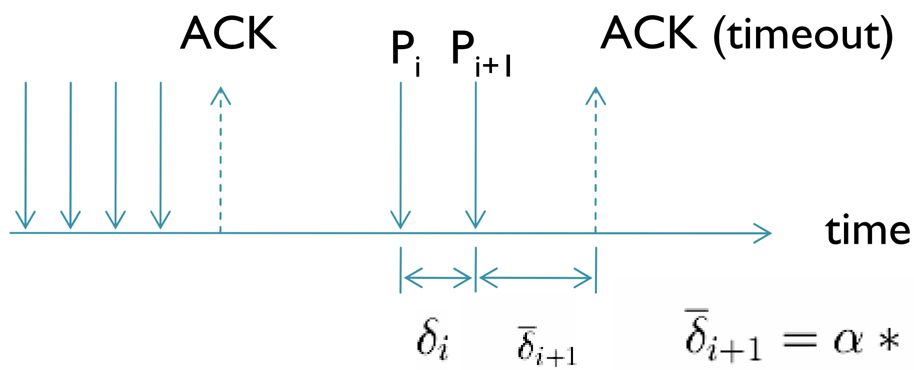
Fig. 2. The optimal limit for the sender congestion window

Max dwin size = Max cwnd size

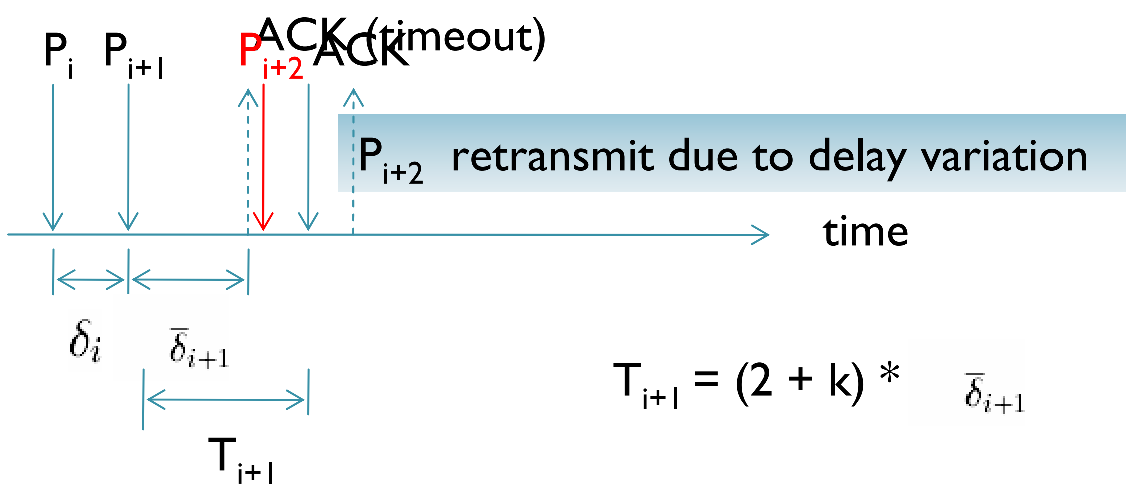
Dynamic Delayed Window

- dwin is initialized to one and increases gradually for each received data packet
- dwin size is four packets
  - under normal conditions
- dwin size is reduced when facing losses to avoid sender retransmission
- dwin size is reduced to two packets under:
  - the received packet is out-of-order
  - the received packet is filling in a gap in the receiver's buffer
  - the timer expires

Timeout Interval



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

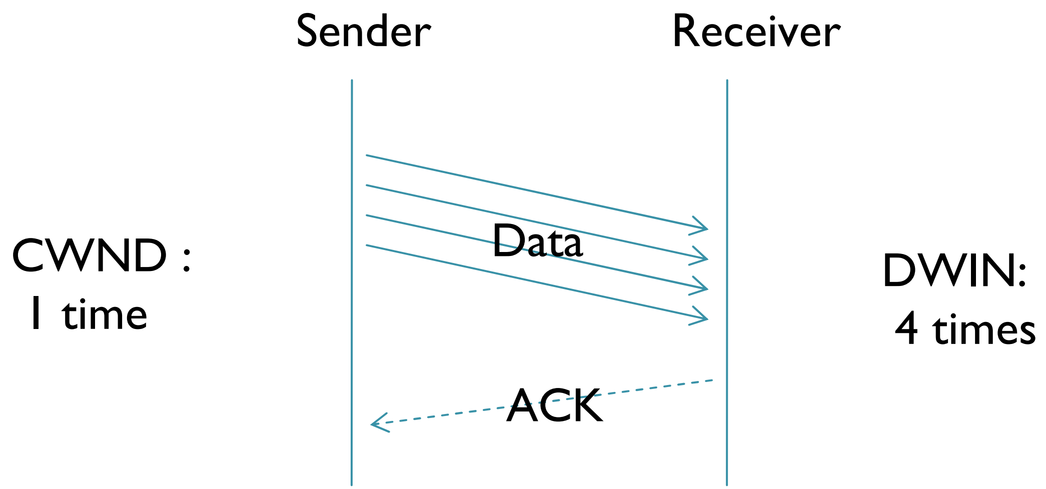


$T_{i+1} = (2 + k) * \bar{\delta}_{i+1}$

K: tolerant factor

CWND Growth

- Assumption:
  - the rate of increasing dwin is equal to the rate of increasing cwnd



- Thus, dwin is greater than cwnd, especially in startup (the cwnd is small)

CWND Growth (cont.)

- The dwin growth is governed by the following

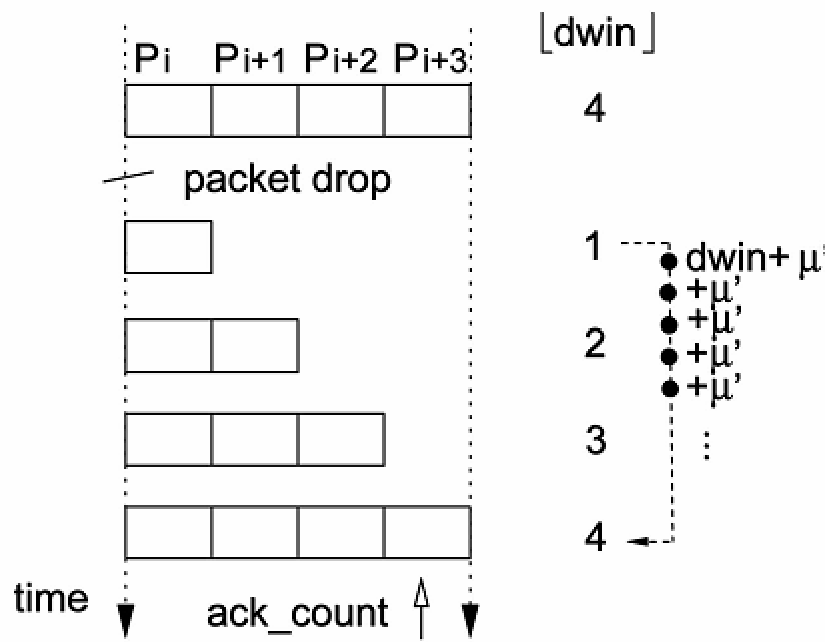
$$dwin = \begin{cases} dwin + \mu, & \text{if } maxdwin = false \\ dwin + 1, & \text{otherwise} \end{cases} \qquad 0 < \mu < 1$$

- Initially, the variable *maxdwin* is set to *false*
- When dwin first reaches its maximum value, the *maxdwin* is set to *true*

An Improved Delaying Window Strategy for High Loss Scenarios

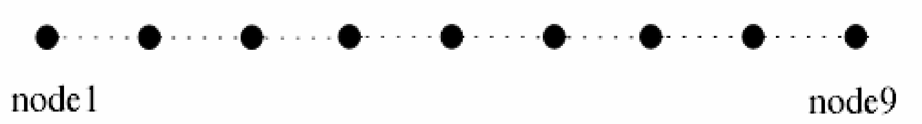
- If the channel if facing <u>constant losses</u>
  - Reduce the delaying window (dwin) to one
  - And, the dwin should be enlarged by less than one

$$dwin = \begin{cases} dwin + \mu, & \text{if in startup} \\ dwin + \mu', & \text{otherwise.} \end{cases}$$



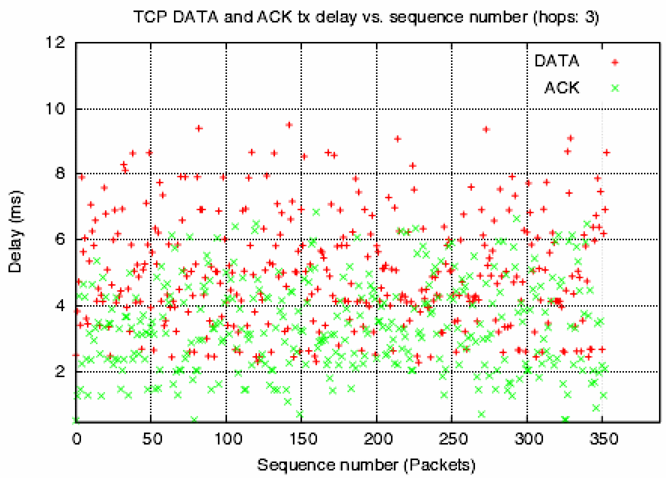
Simulation

- Tool: ns2
- Scenarios:

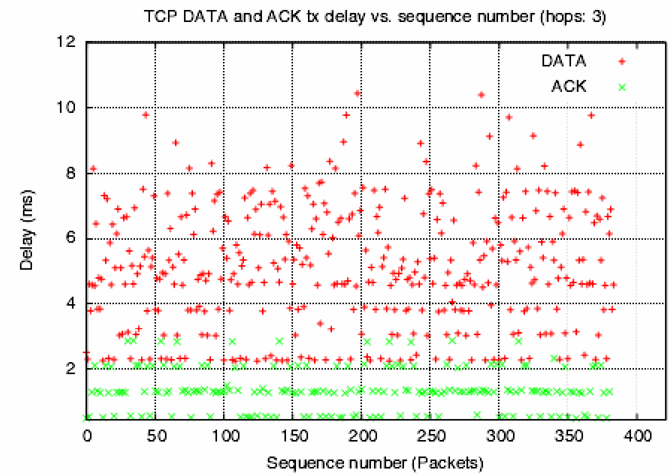


(a) chain topology

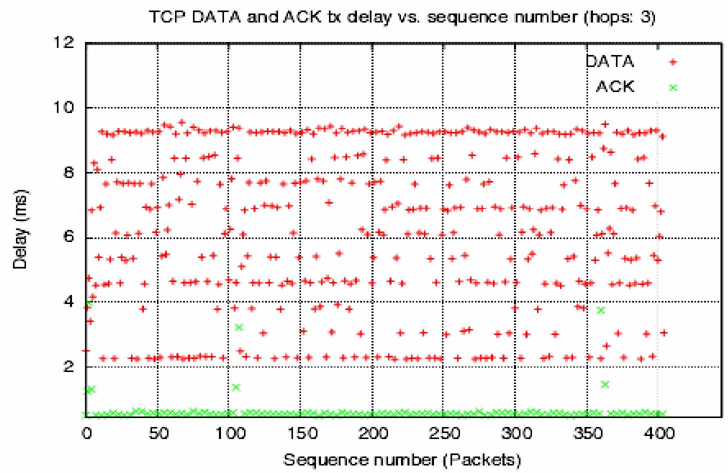
DATA and ACK transmission delay in a typical wireless channel



(a) No delayed ACK



(b) Standard DA



(c) TCP-DAA

Throughput

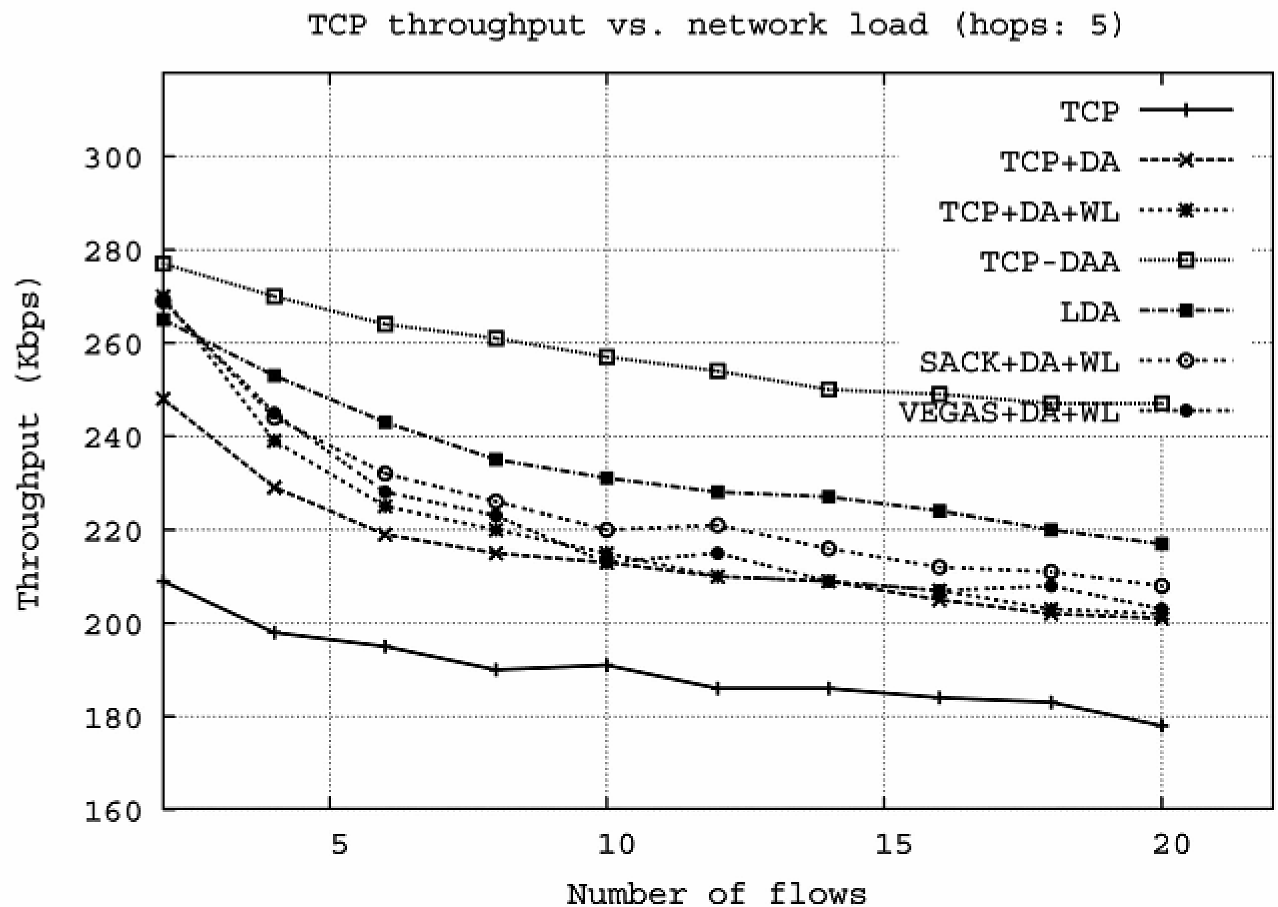


Fig. 8. Aggregate throughput in the chain topology.

Conclusion

- TCP-DAA aims to minimize collisions resulting from mutual interference between data and ACK packets by transmitting as few ACKs as possible.

- The key ideas are to delay ACKs and to adaptively compute the timeout interval.

- The simulation showed that this algorithm can outperform conventional TCP.