Improving TCP performance in ad hoc networks using signal strength based link management

Fabius Klemm
Zhenqiang Ye
Srikanth V. Krishnamurthy
Satish K. Tripathi

Elsevier Ad Hoc Networks, 2005

Outline

- Introduction
- Reducing link failures to improve TCP performance
  - Reducing false link failures
  - Signal strength based link management methods
    - Proactive link management
    - Reactive link management
    - Modifications to AODV
    - Transport layer
- Simulations
- Conclusions

Introduction

- The objective is to mainly stem the degradation in TCP performance due to mobility
  - Mobility in ad hoc networks causes frequent link failures
- Node mobility and link layer contention are the two main reasons for packet losses
- Propose mechanisms to reduce the number of packet losses
  - These mechanisms are based on signal strength measurement at the physical layer

Reducing false link failures

- The IEEE 802.11 MAC protocol reports a link failure
  - If it cannot establish an RTS-CTS handshake with a neighbor within seven RTS attempts
- The idea is to double the number of retransmission attempts
  - If there is a high probability that the neighbor is still within transmission range
- In order to determine whether a node is still within range
  - When a node receives a packet from a neighbor, it measures the received signal strength

Reducing false link failures

- ns-2 uses the two-ray ground propagation model
- Using this propagation model, the distance d to the transmitter of a packet can be calculated as follow:

$$d = \sqrt[4]{\frac{P_t \cdot G_t \cdot G_r \cdot h_t^2 \cdot h_r^2}{P_r \cdot L}}$$

If different, it must include the value in options filed of the MAC protocol header

  - P_t : default transmission power
  - P_r : received signal strength
  - G_t and G_r : antenna gains of the transmitter and the receiver
  - h_t and h_r : antenna height of the transmitter and the receiver
  - L : system loss, which is set to 1 by default

Reducing false link failures

- The MAC protocol keeps a record of the distances to neighboring nodes in a neighbor table

- A table entry consists of five fileds
  - A neighbor ID
  - A distance d_1 to the neighbor, time t_1
  - A distance d_2 to the same neighbor, recent time t_2

- Thus, at any given time t, the current distance d_{est} as follow

$$d_{\text{est}} = d_2 + \frac{d_2 - d_1}{t_2 - t_1} \cdot (t - t_2)$$

$$\text{for } t_1 < t_2 < t \text{ and } d_1, d_2 \geqslant 0.$$

Proactive link management

- The idea of Proactive LM is to inform the routing protocol that a link is going to break before the link actually breaks
  - Tries to predict link breakage

- Proactive LM estimates the projected distance to a neighbor in the immediate future
  - For example, if the current time is t, the distance $d_{0.1}$ of a particular neighbor at (t + 0.1)s is

$$d_{0.1} = d_2 + \frac{d_2 - d_1}{t_2 - t_1} \times (t + 0.1 - t_2)$$

$$\text{for } t_1 < t_2 < t \text{ and } d_1, d_2 \geqslant 0.$$

Proactive link management

- Proactive LM informs the routing layer as soon as $d_{0.1}$ is estimated to be greater than the transmission range
- The routing protocol then informs the packet source, which stops sending packets and initiates a route discovery
- Proactive LM and Reactive LM are implemented at the MAC layer

Reactive link management

- Reactive LM temporarily increases the transmission range of a node to re-establish a broken link
  - Keeps a broken link alive with higher transmission power
  - Packets in transit can then traverse the re-established high power link

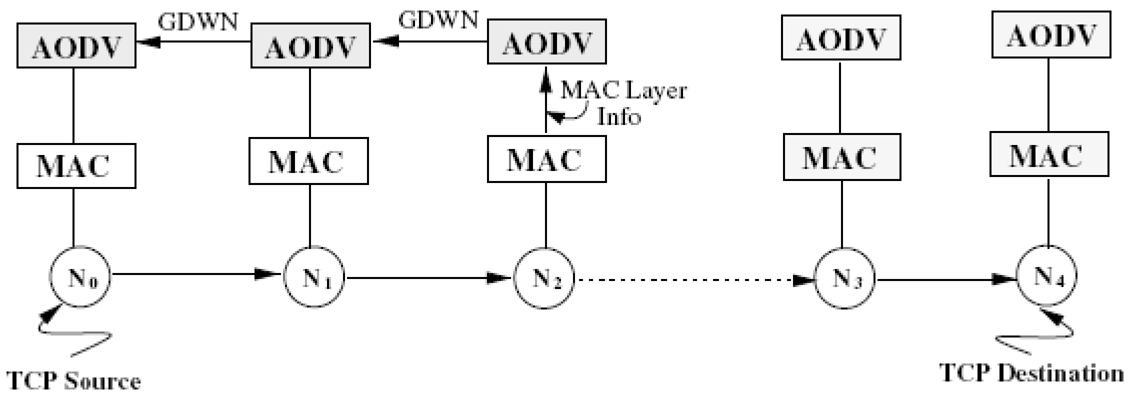
- When Reactive LM establishes the temporary high power link, it stimulates the routing protocol to begin a new route discovery

Modification to AODV

- In AODV, a route to a destination N_d in the routing table of a node N_x can be in either of two states
  - Up (N_x forwards packets to N_d)
    - If Nx receives a Route Request (RREQ) for Nd, it will respond with a Route Reply (RREP)
  - Down (N_x does not have a route for $N_{d)}$
    - If N_x receives a packet for N_d, it will drop the packet and respond with a Route Error (RERR)
    - If N_x wants to send packets to N_d, it will initiate a route discovery
    - If N_x receives a RREQ for N_d, it will broadcast the RREQ
- To add an additional route state
  - Going Down (GDWN) state

Modification to AODV

- The GDWN state has the following characteristics
  - If N_x receives a packet for N_d, it will forward the packet
  - If N_x receives an RREQ for N_d, in lieu of responding with a RREP it broadcasts the RREQ
  - If an application at N_x wants to send packets to N_d, the modified AODV will initiate a route discovery

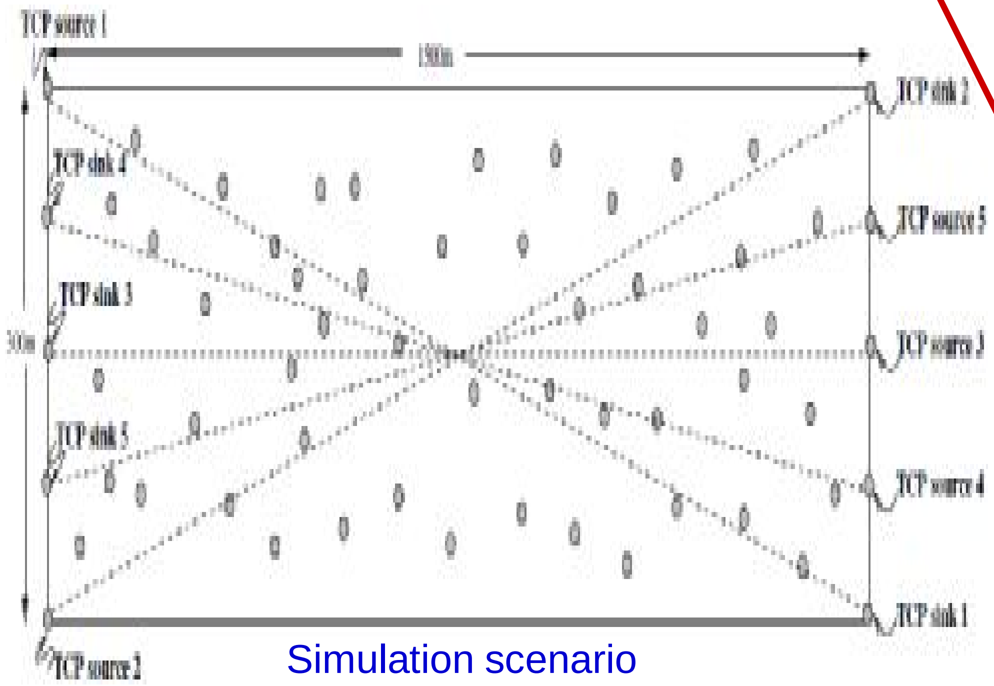


Transport layer

- TCP Tahoe, Reno and New Reno grow the congestion window until packet are dropped
  - Suffer from the “instability problem” due to the excessive growth of the congestion window
- TCP Vegas does not need packet losses to stop the growth of the congestion window
  - Uses Round Trip Time (RTT) estimations to control the size of congestion window
- To use TCP Vegas for all simulations with ns-2

Simulations

Interference range = 550 m



Simulation scenario

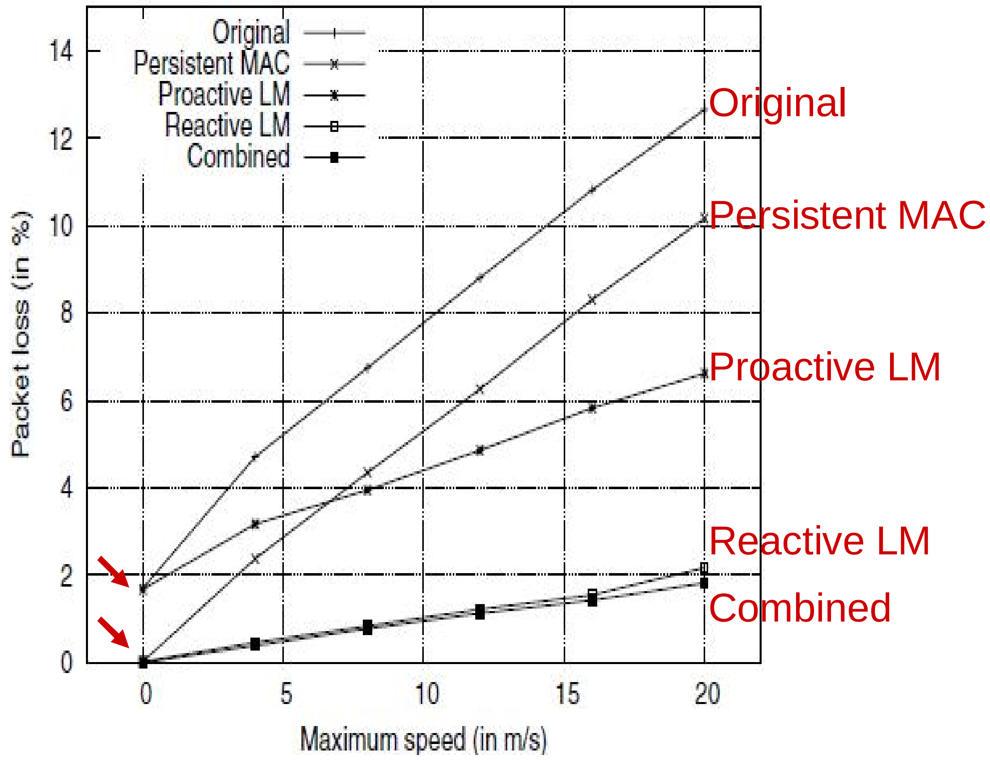
Summary of simulation parameters

Parameter	Value
Simulation time	600 s
Number of mobile nodes	50
Number of static nodes	2 in setup I, 6 in setup II, 10 in setup III
Default transmission range	250 m
Proactive LM time to link breakage	0.1 s
Reactive LM high power range	275 m
Traffic	
Type	FTP with infinite backlog over TCP
Packet size	1460 bytes
Number of TCP connections	1 in setup I, 3 in setup II, 5 in setup III
Movement	
Pause time	0 s
Maximum speed	0, 4, 8, 12, 16, and 20 m/s

Minimum speed = 10% of maximum speed

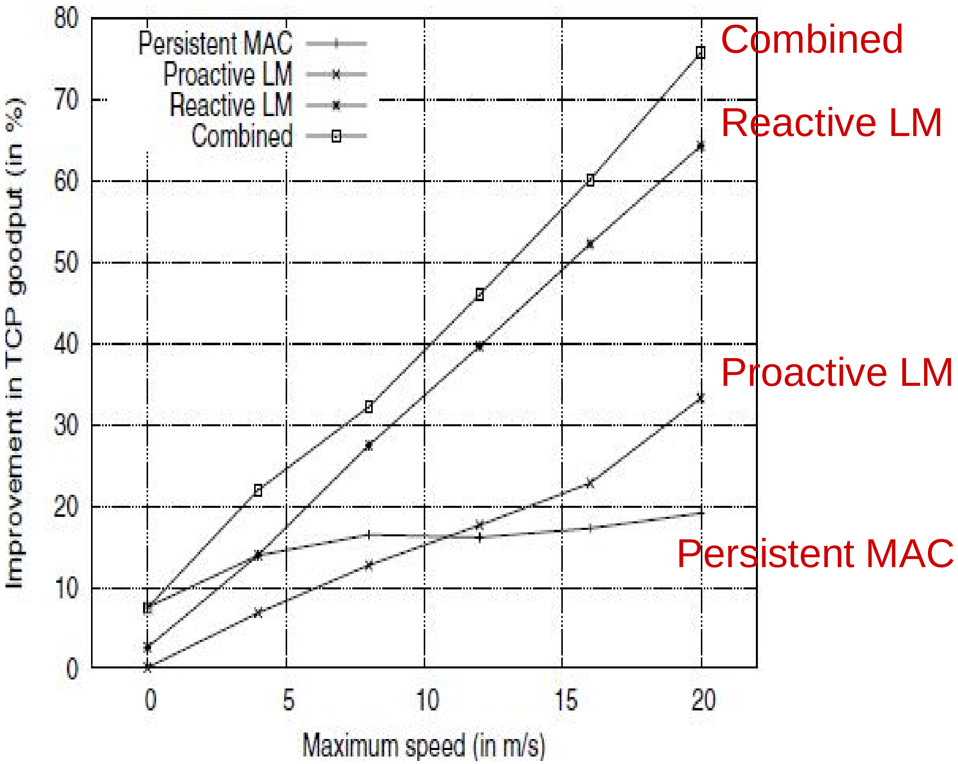
= 50% of maximum speed (highly mobile scenario)

Simulations

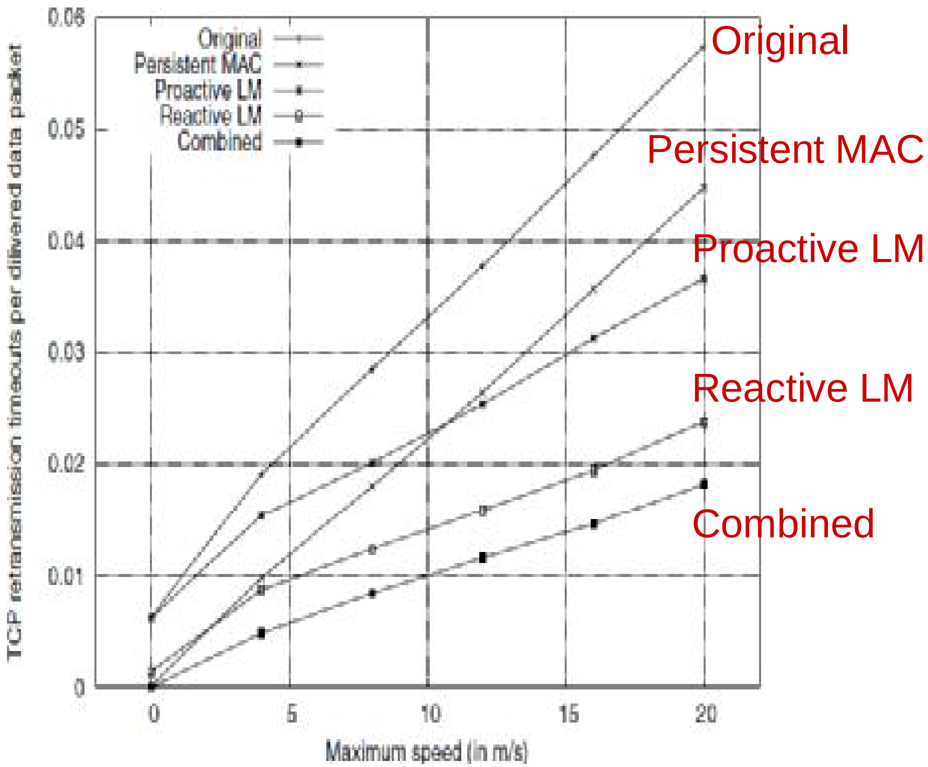


Performance of the various schemes with one TCP session

Simulations

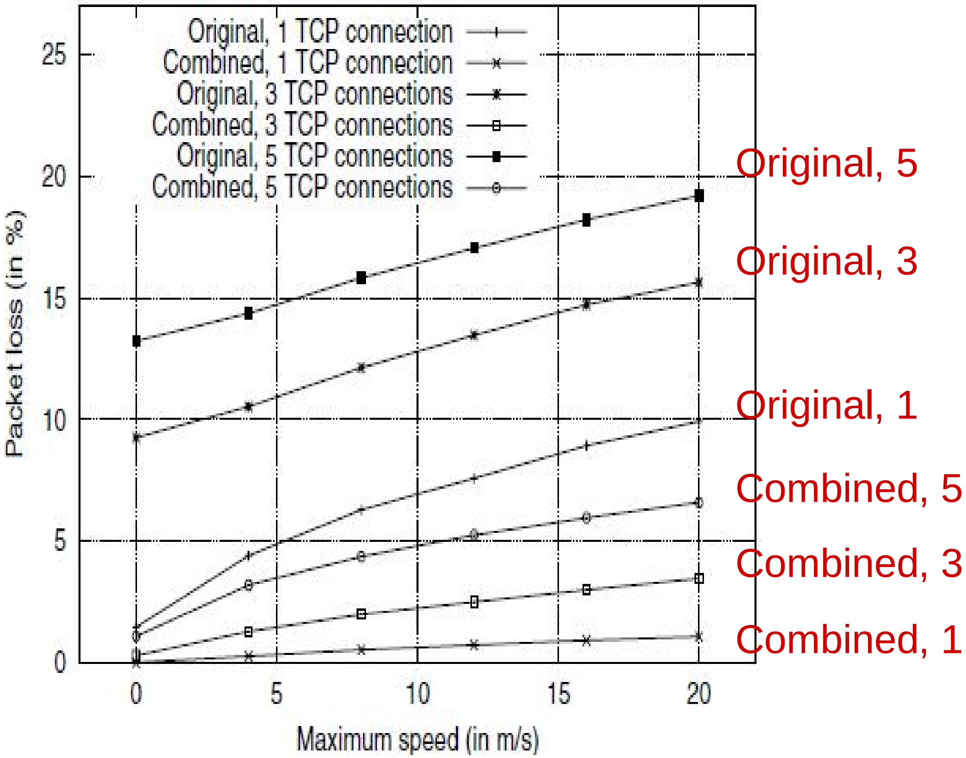


Improvement in TCP goodput versus maximum speed for one connection

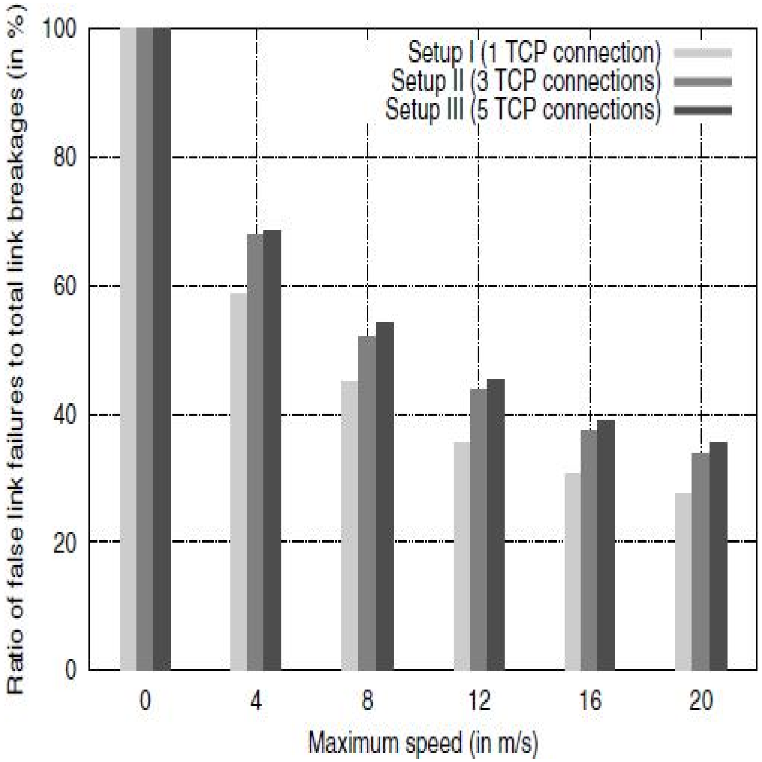


TCP re-transmission time-outs per delivered packet as a function of maximum speed

Simulations
Effects of traffic load

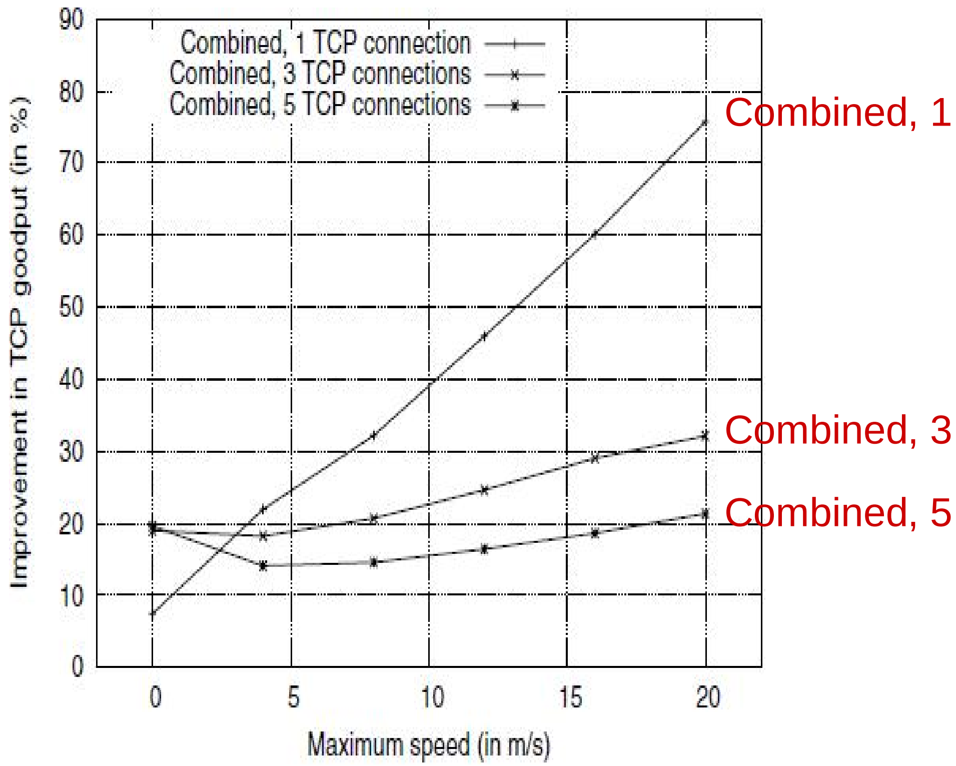


Comparison of packet losses with various nmber of TCP connections



Fraction of false link failures

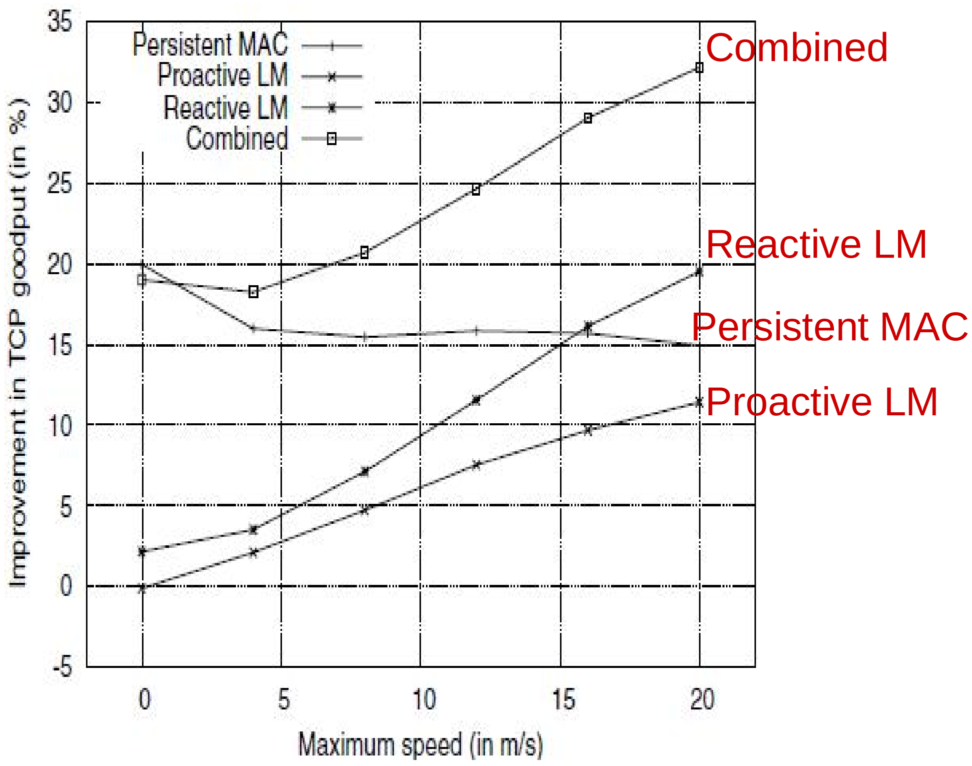
Simulations
Effects of traffic load



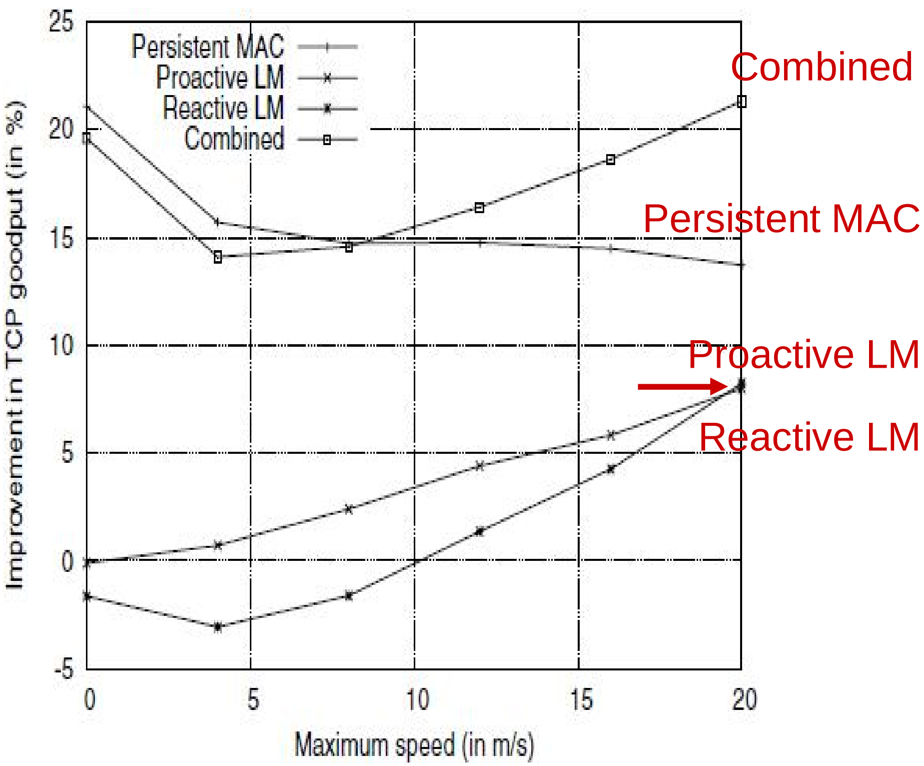
Improvement in TCP goodput versus maximum speed with various number of TCP connections

Simulations

Effects of traffic load



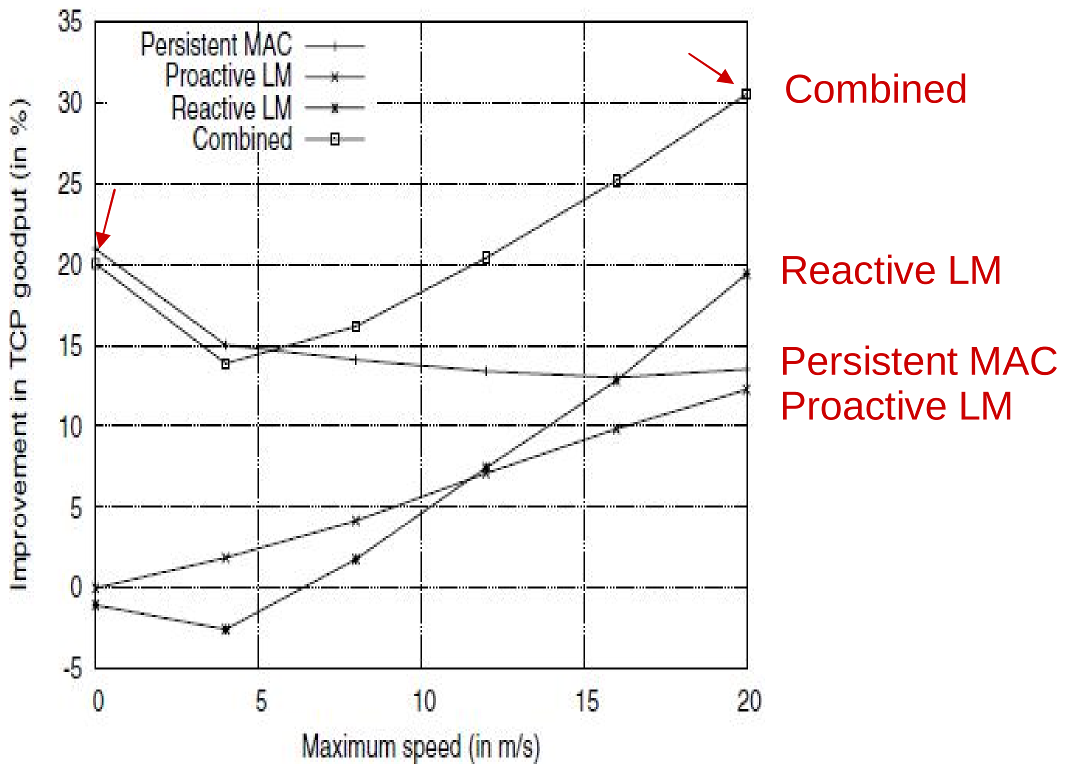
Improvement in TCP goodput versus maximum speed for three connections



Improvement in TCP goodput versus maximum speed for five connections

Simulations

Effects of node mobility



Improvement in TCP goodput versus maximum speed for five TCP connection in high mobility pattern (min_speed = 50% off the max_speed)

Conclusion

- The objective is to reduce the packet losses due to mobility and thereby improve the performance of TCP
- To propose a link management framework
  - Persistent MAC
  - Proactive Link Management
  - Reactive Link Management
- The simulation results show that, in high mobility, the combined scheme can improve the TCP goodput
  - The network is lightly loaded：up to 75%
  - The network is heavily loaded：14 – 30%