Alleviating the effects of mobility on TCP Performance
Signal Strength based Link Management

Fabius Klemm*, Srikanth Krishnamurthy+
and Satish Tripathi+,
* EPFL Lasaunne ,+ University of California, Riverside

Paper Presented by Dr.Nitin Vaidya, UIUC

Presentation at UCLA on June 6th , 2002

Outline

- **Motivation for Research**

- **Using Lower Layer Support to improve TCP performance**

    - **Link Failures – True and False**

    - **Signal Strength based methods to help improve TCP goodput**

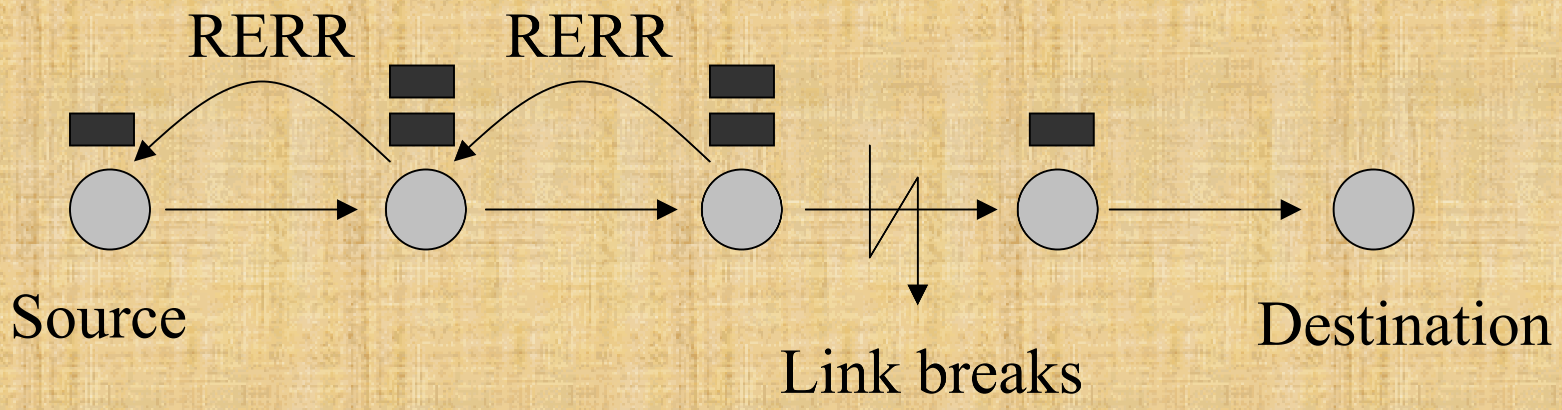
    - **Preliminary experiments and results**

Motivation

- TCP is unable to differentiate between true and false link failures – former due to mobility, latter due to congestion.
- Implement link layer mechanisms that can help:
  - Anticipate real link failures by signal strength measurements – preemptively initiate route discovery.
  - Reactively increase power level for transmission upon the detection of a real link failure to salvage TCP packets in transit.
- Requires mechanisms for differentiating between true and false link failures.

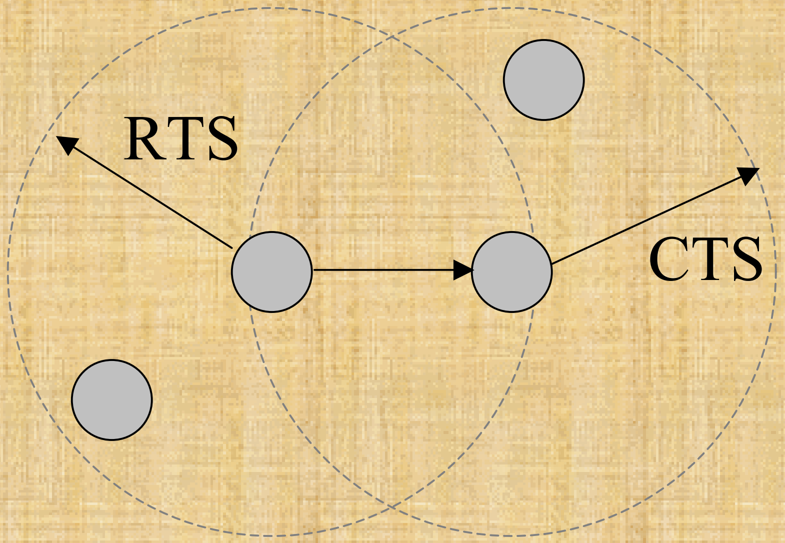
Background -- AODV

- Ad hoc On-Demand Distance-Vector
- Route discovered by queries. RERR message sent upon discovery of a link failure.



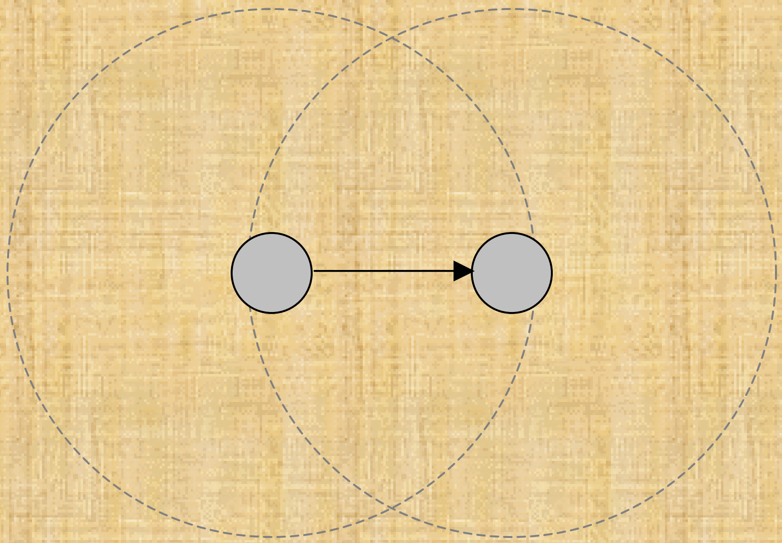
Revisiting the IEEE 802.11 MAC protocol

- RTS – CTS – DATA – ACK
- Solves the hidden and exposed terminal problem in most cases.



False Link Failure Reports

- Neighbor within reach
- Mac Layer cannot establish RTS/CTS Handshake
- Mac Layer reports link break to upper layers

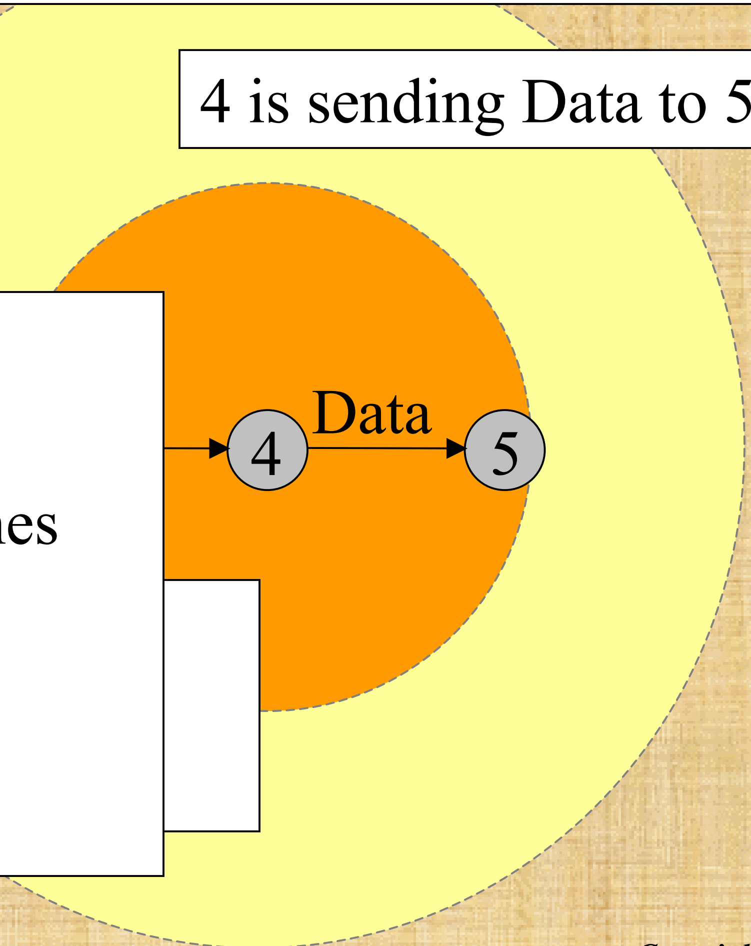


How come?

- Transmission Range: 250 m
- Interference Range: 550 m

4 is sending Data to 5

1 is sends an RTS to 2



Node 1 gives up after seven times

→ False link layer report

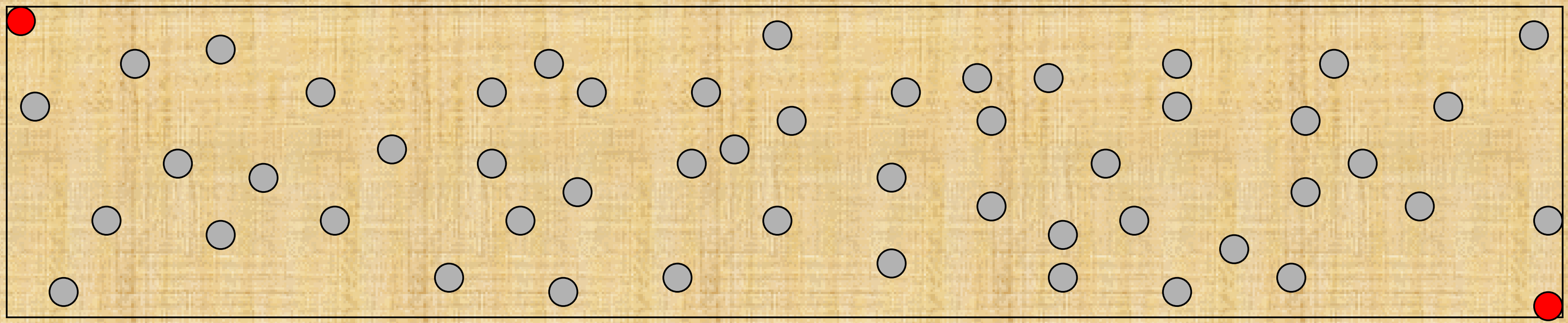
Objective

- In this preliminary work, we consider sparse scenarios and use a rather naïve approach to differentiating between true and false link failures.
- In reality, more sophisticated techniques might be needed.

Simulation Scenario

- 50 mobile nodes + 2 static nodes
- 1 TCP connection

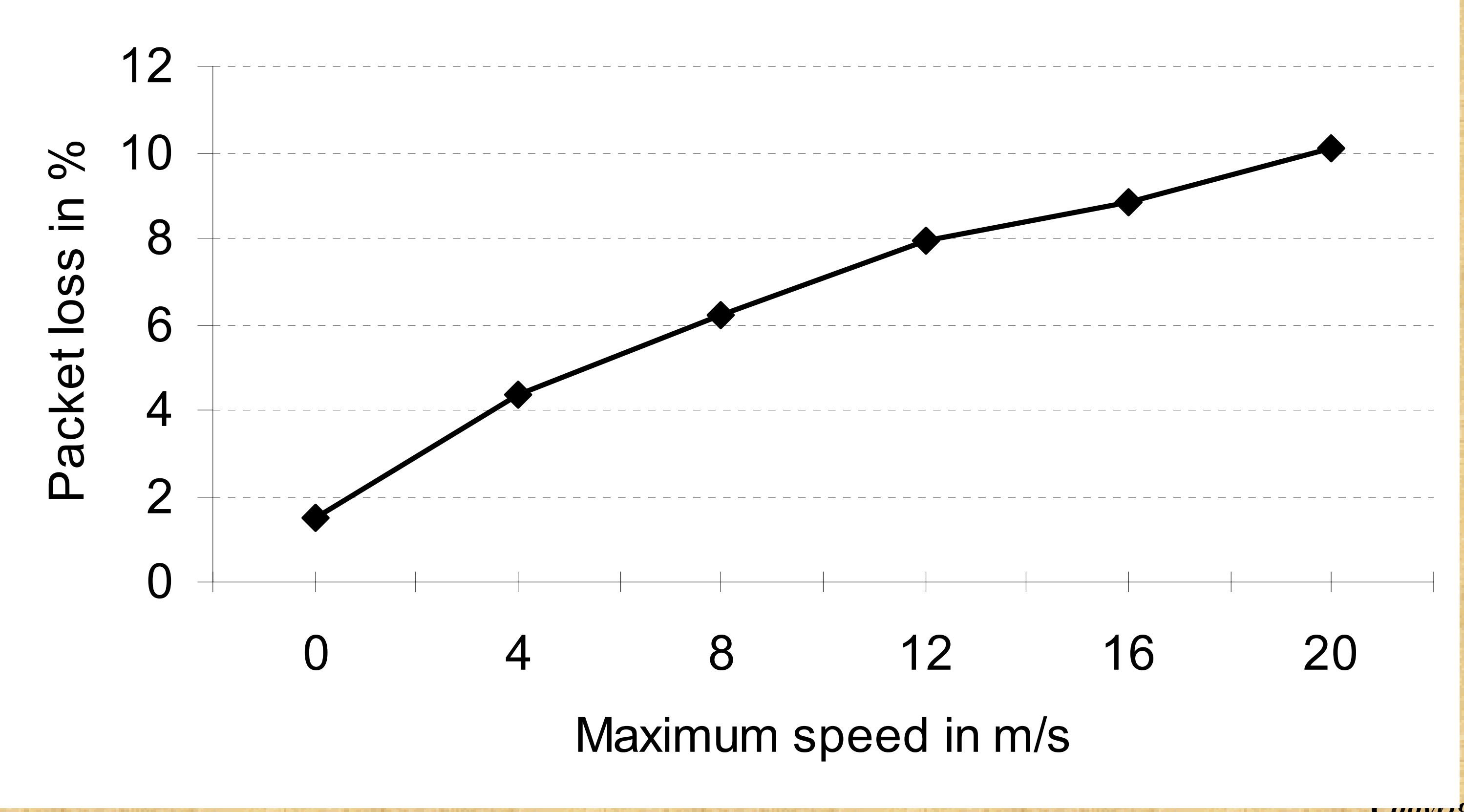
300 x 1500 meters



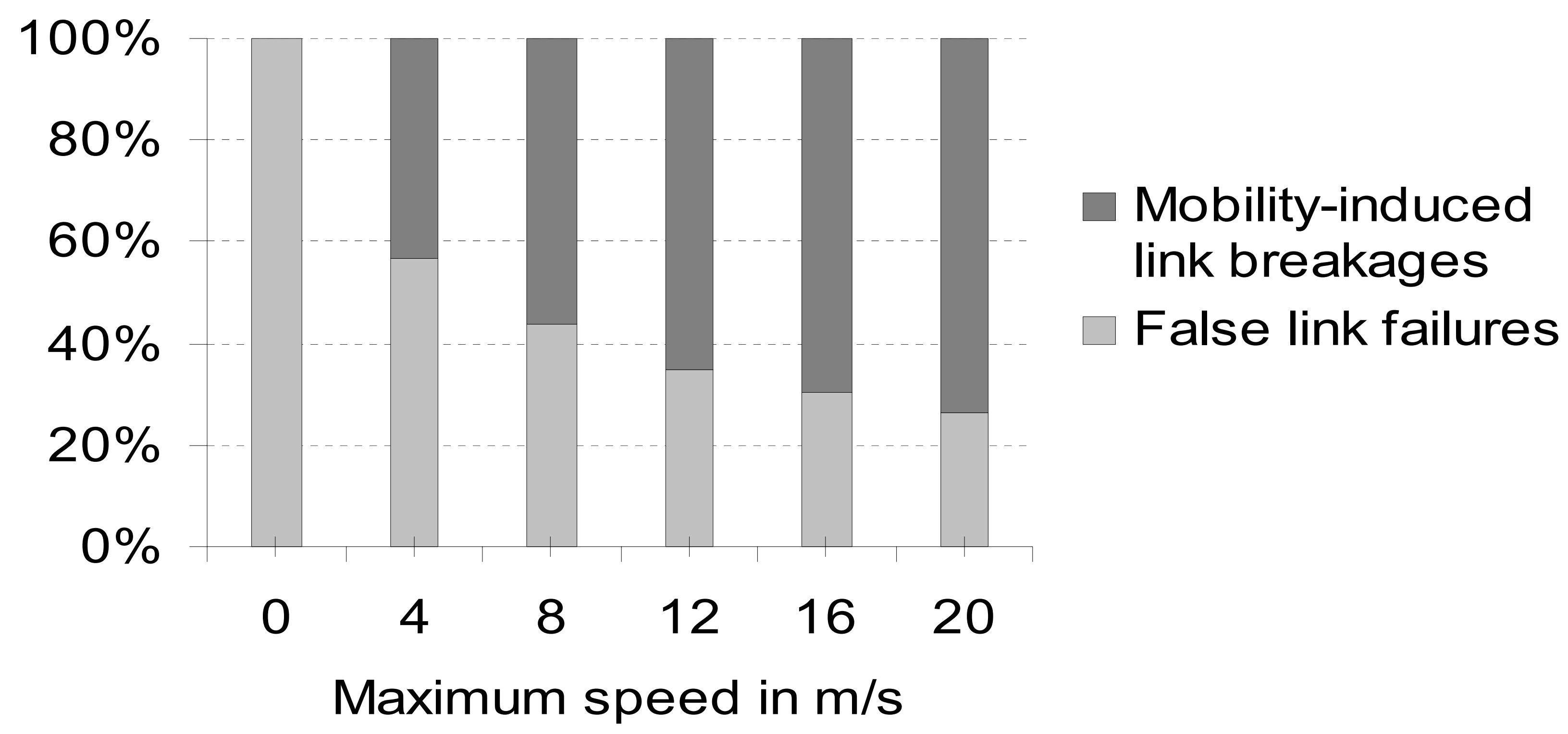
- Distance between TCP source and sink:
  - about 1530 m or 8.8 hops in average

Problems to be solved

- Reduce packet losses due to **mobility** (correct link breaks)
- Reduce packet losses due to **false link failure reports**



Reasons for packet loss



Low Mobility: False link failures dominate
High Mobility: Correct link failures dominate

Tackling False Link Failures

Each node maintains a Mac layer neighbor table:

Neighbor ID	Timestamp 1	Distance 1	Timestamp 2	Distance 2
3	4.200	200	4.205	201
…				
…				

- **Node computes distance from signal strength – simple model is assumed wherein the attenuation is inversely proportional to the square of the distance.**
- **The time stamps correspond to the last two instances when the node heard the neighbor.**

Tackling False Link Failures

Persistent Mac

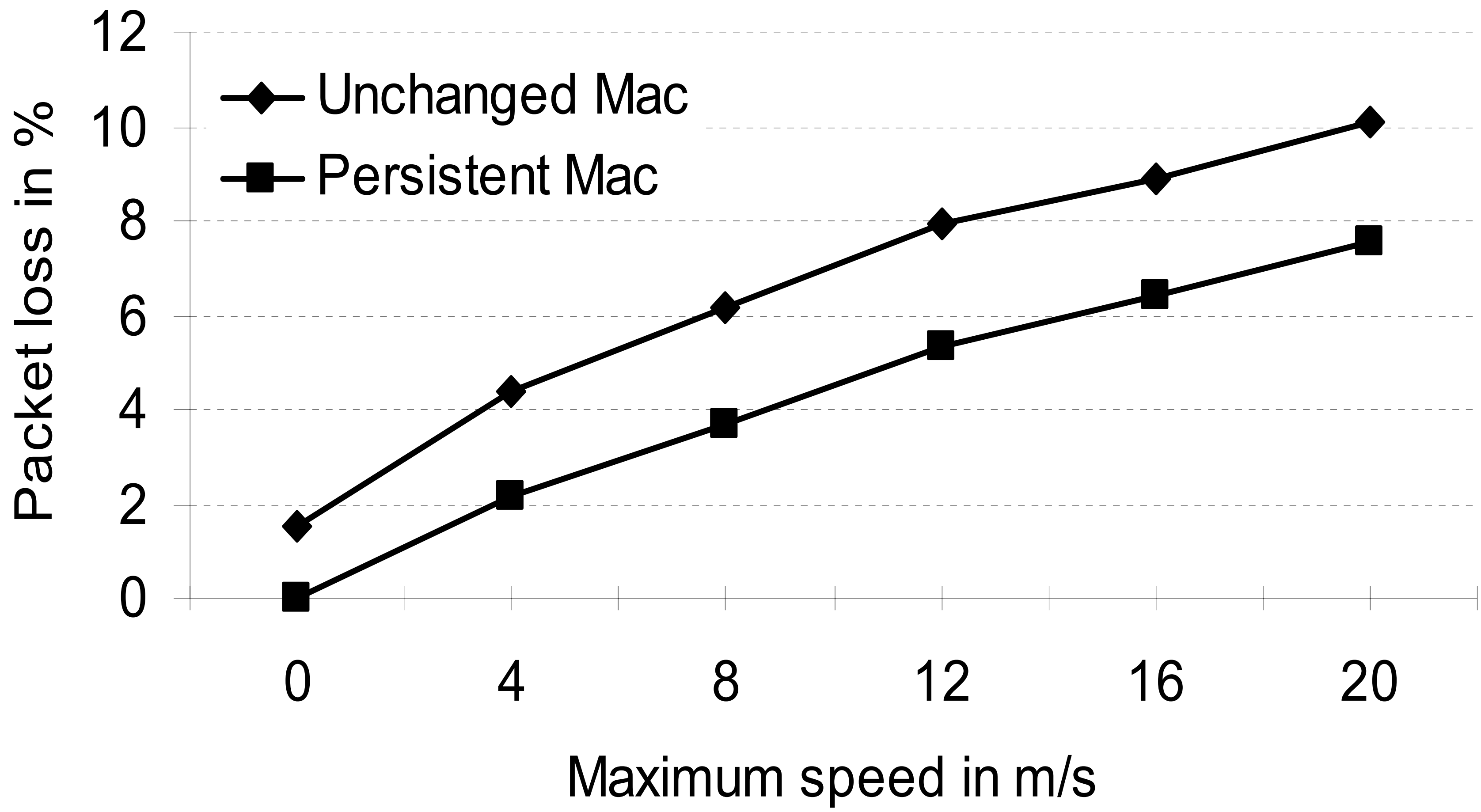
–A Node sends RTS packets more than seven times if neighbor is likely to be within transmission range.

– Simple naïve approach.

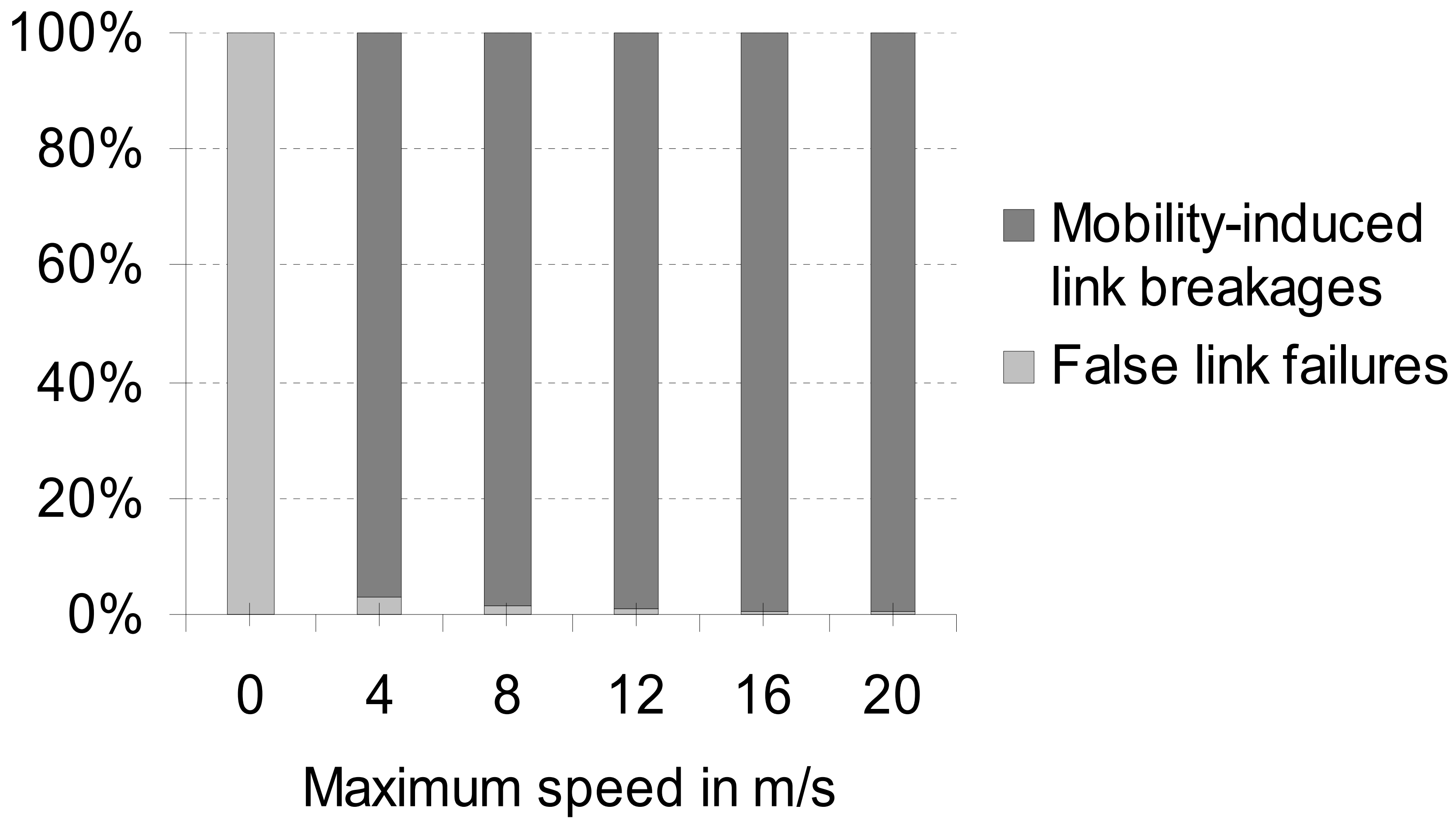
– Seems to work in the sparse scenarios considered.

– More sophistication may be needed in dense scenarios.

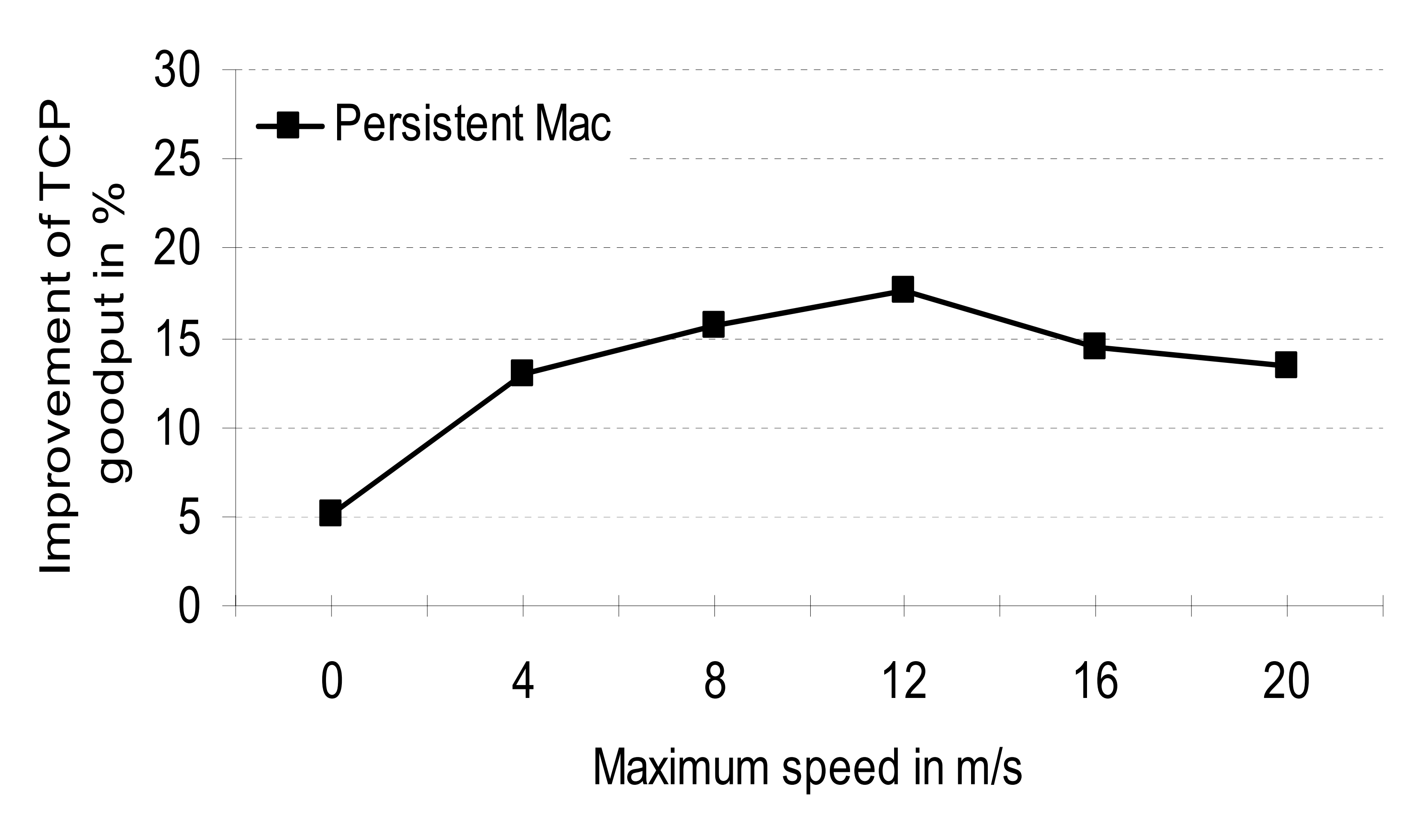
Persistent Mac – Packet Loss



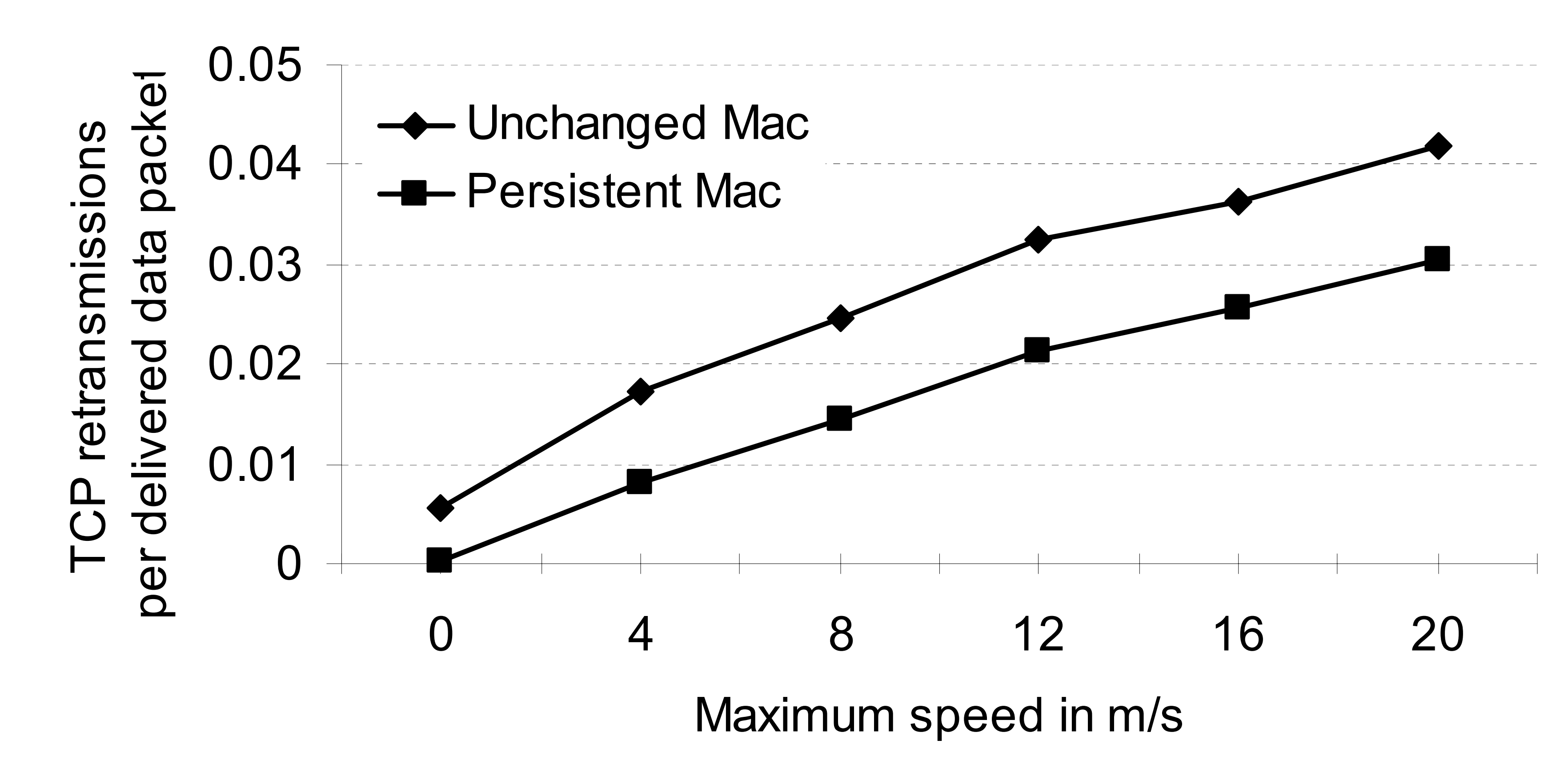
Persistent Mac – Link Breakages



Persistent Mac - Goodput

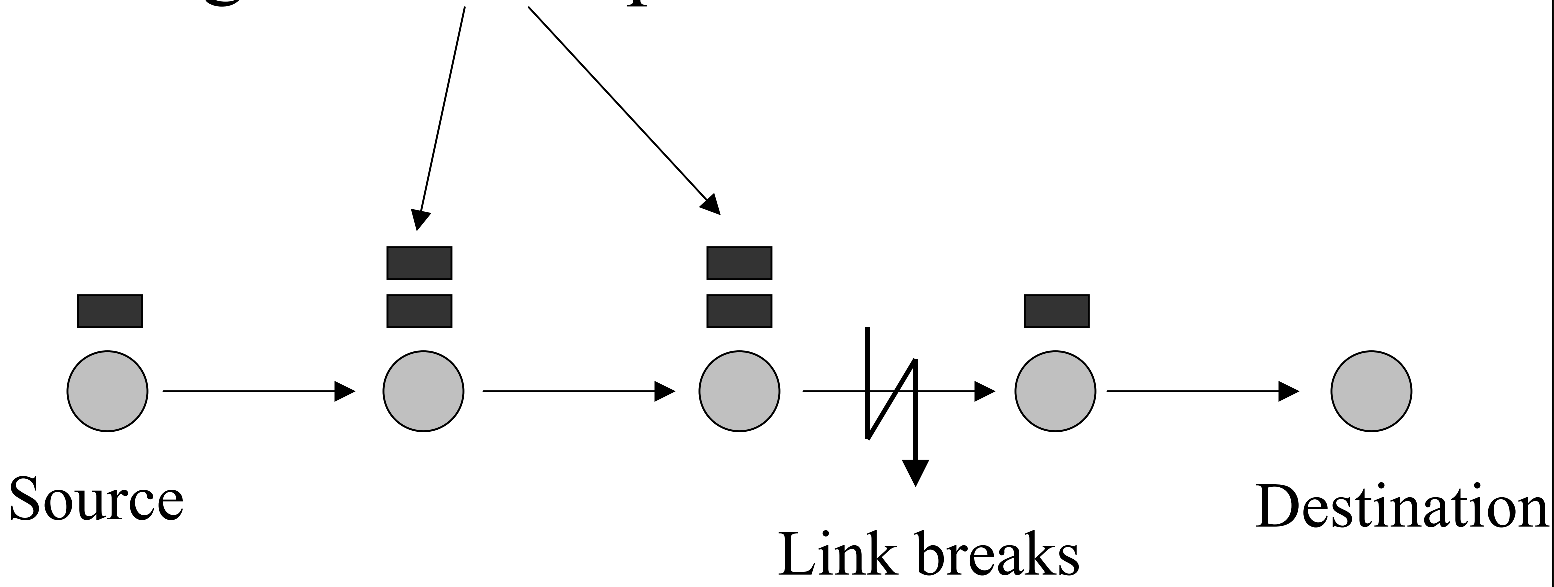


Persistent Mac – TCP Retransmissions



Lost packets due to Mobility

- Salvage in transit packets:



Salvage Packets

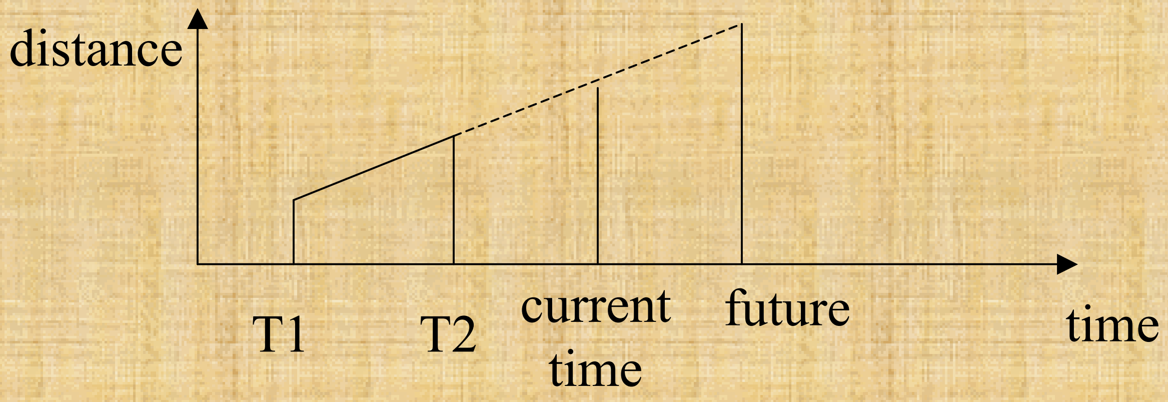
- Two Approaches:

1. *Proactive:* Predict link breakage and stimulate the source to preemptively initiate a route discovery.

2. *Reactive:* Re-establish a broken link with a temporary higher transmission power level.

Mac Layer: Proactive

- Nodes use neighbor table to predict node movement in the future:
  - –Simple prediction: Assume linear node movement



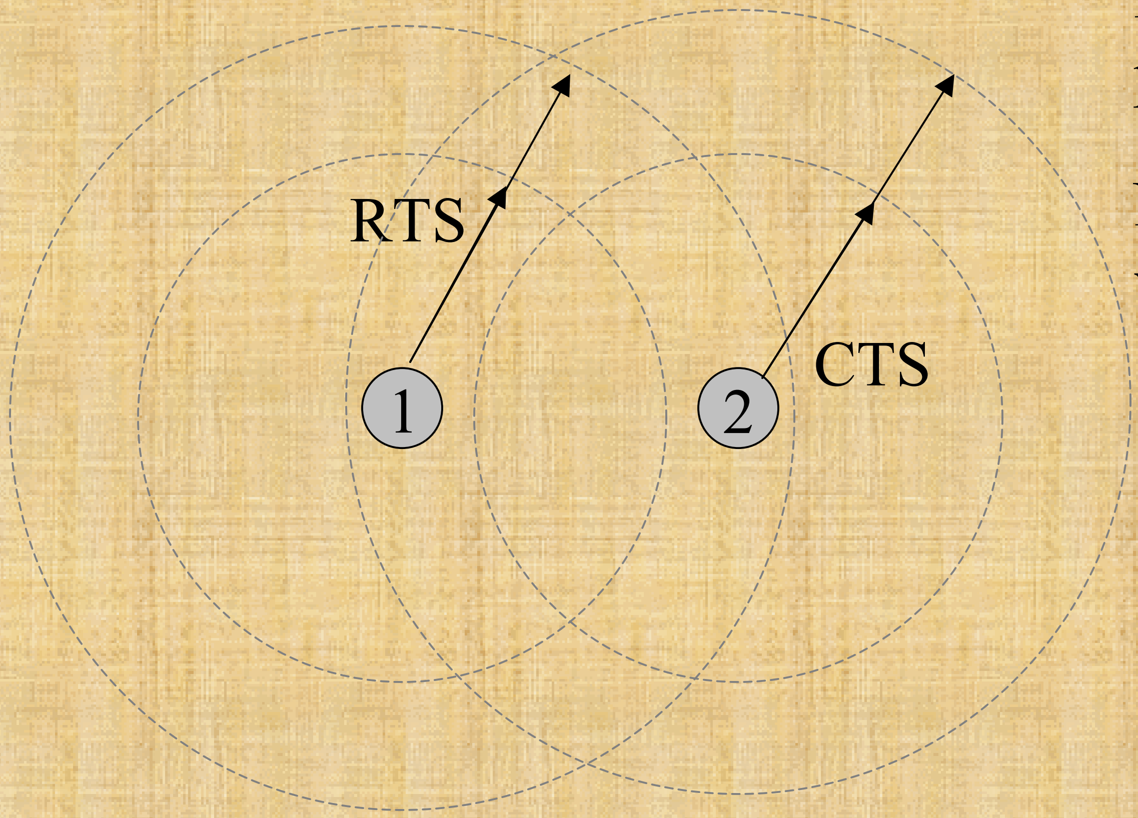
- Mac layer informs routing layer when next hop is almost out of range

Mac Layer: Reactive

- Node raises transmission power temporarily if it cannot establish an RTS/CTS handshake

Reactive Mac

Node 2 moves out of range of Node 1

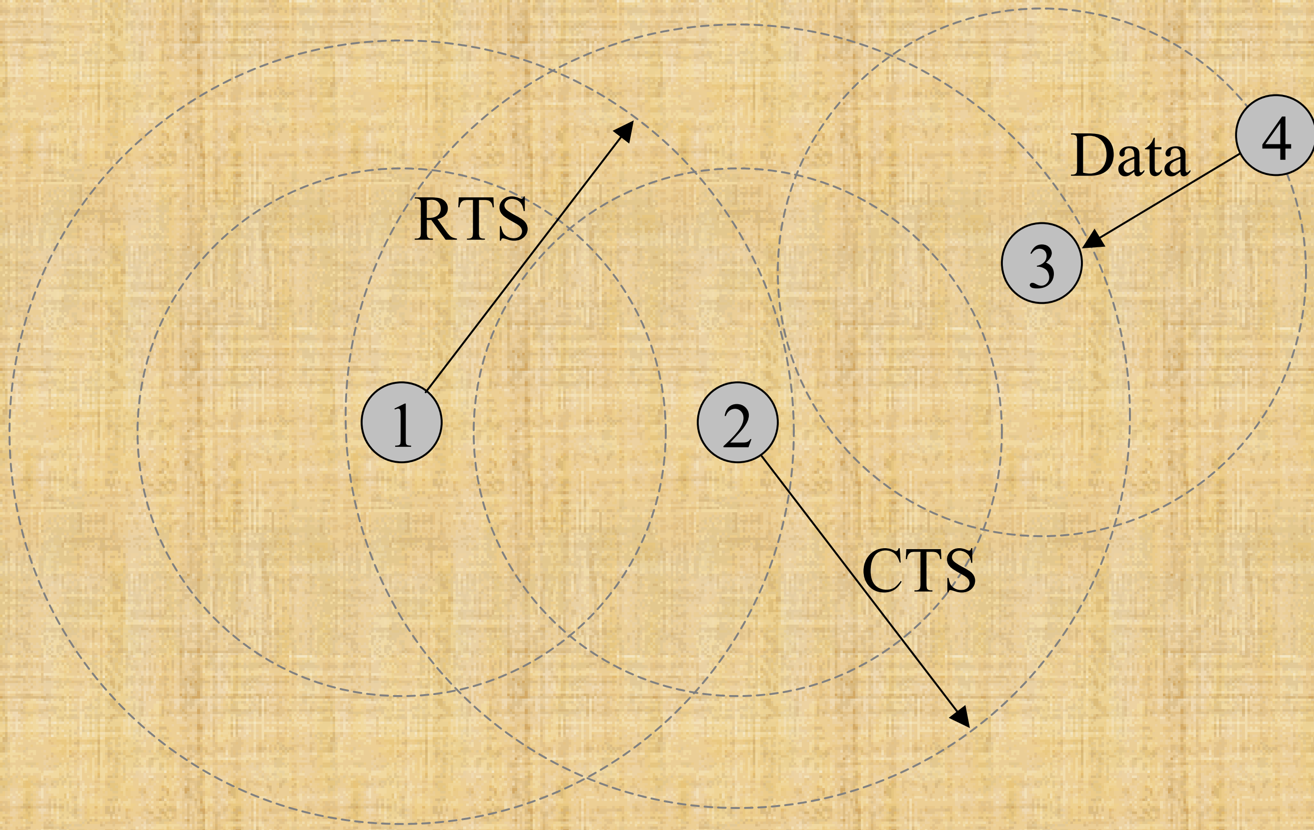


RTS – Frame contains power value

Node 2 sends CTS with the same power

Same for Data and ACK

Problems!

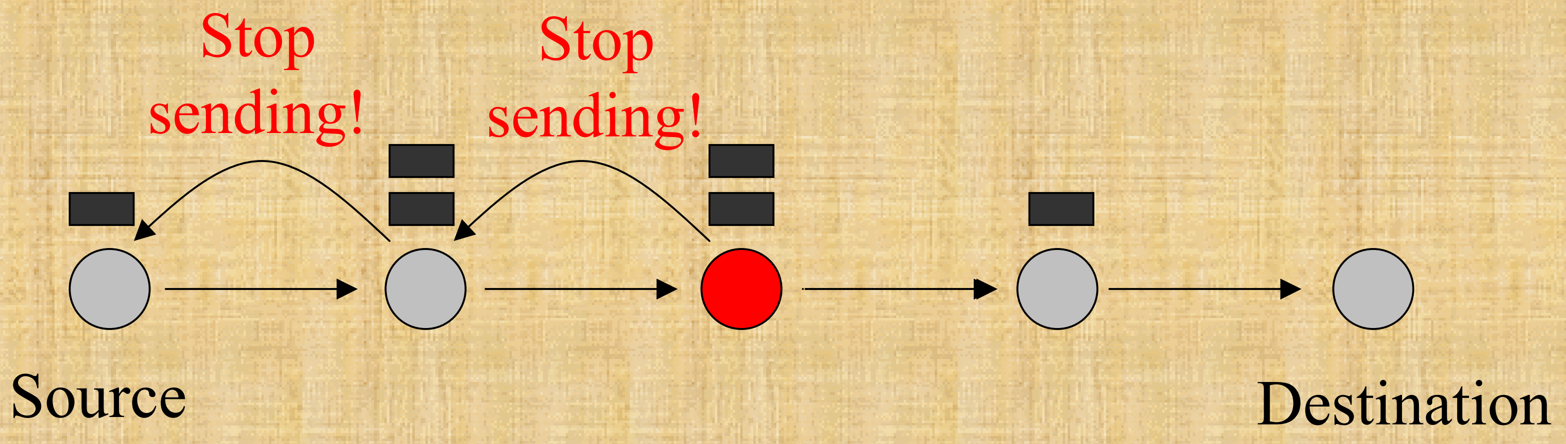


- Node 1 establishes a high power link
- Node 3 is receiving Data from Node 4

Node 2 does not know about the data transfer
The high power CTS collides with the Data at Node 3

Salvaging Packets

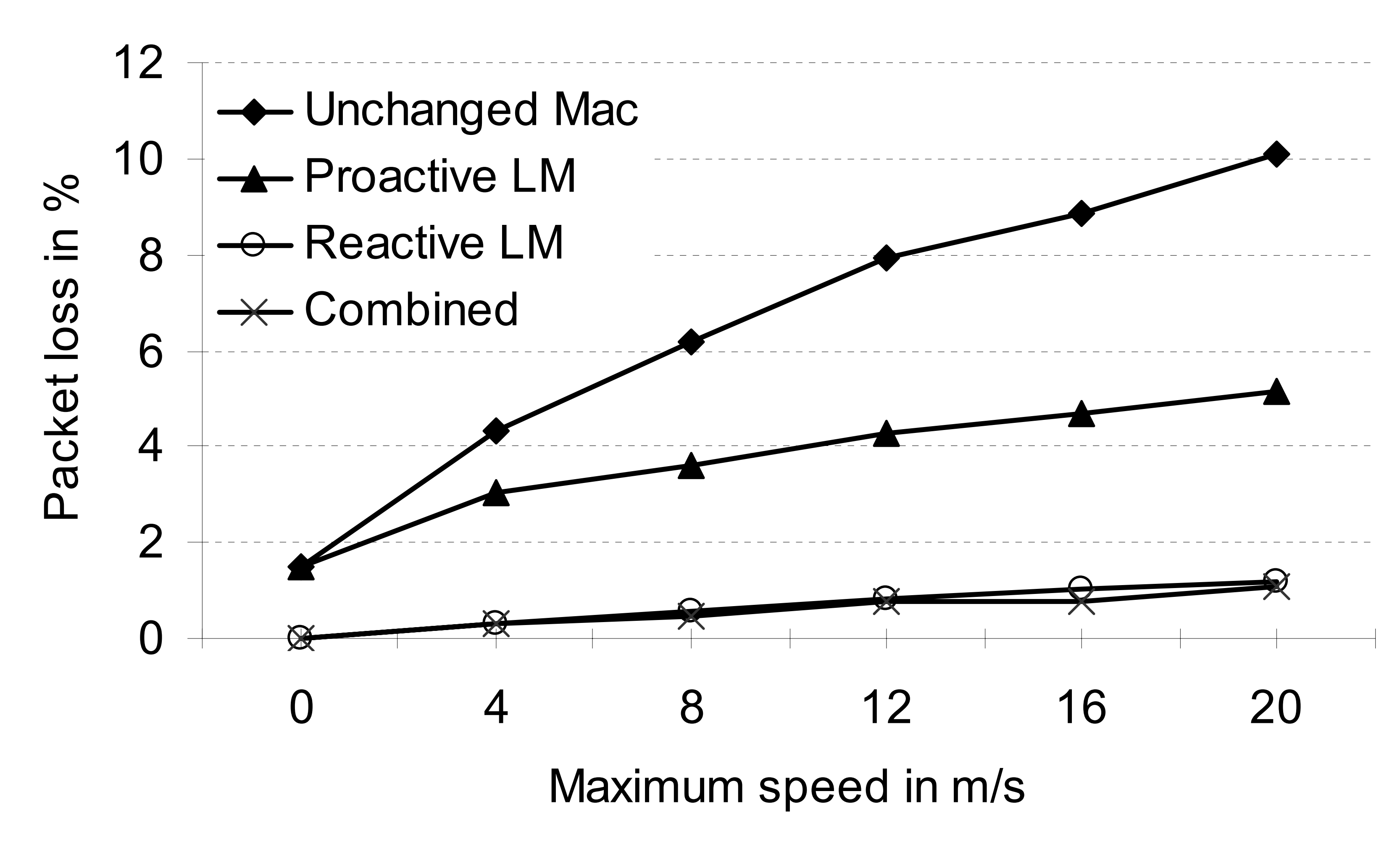
- Routing layer informs source to stop sending
- But: Intermediate nodes keep forwarding packets



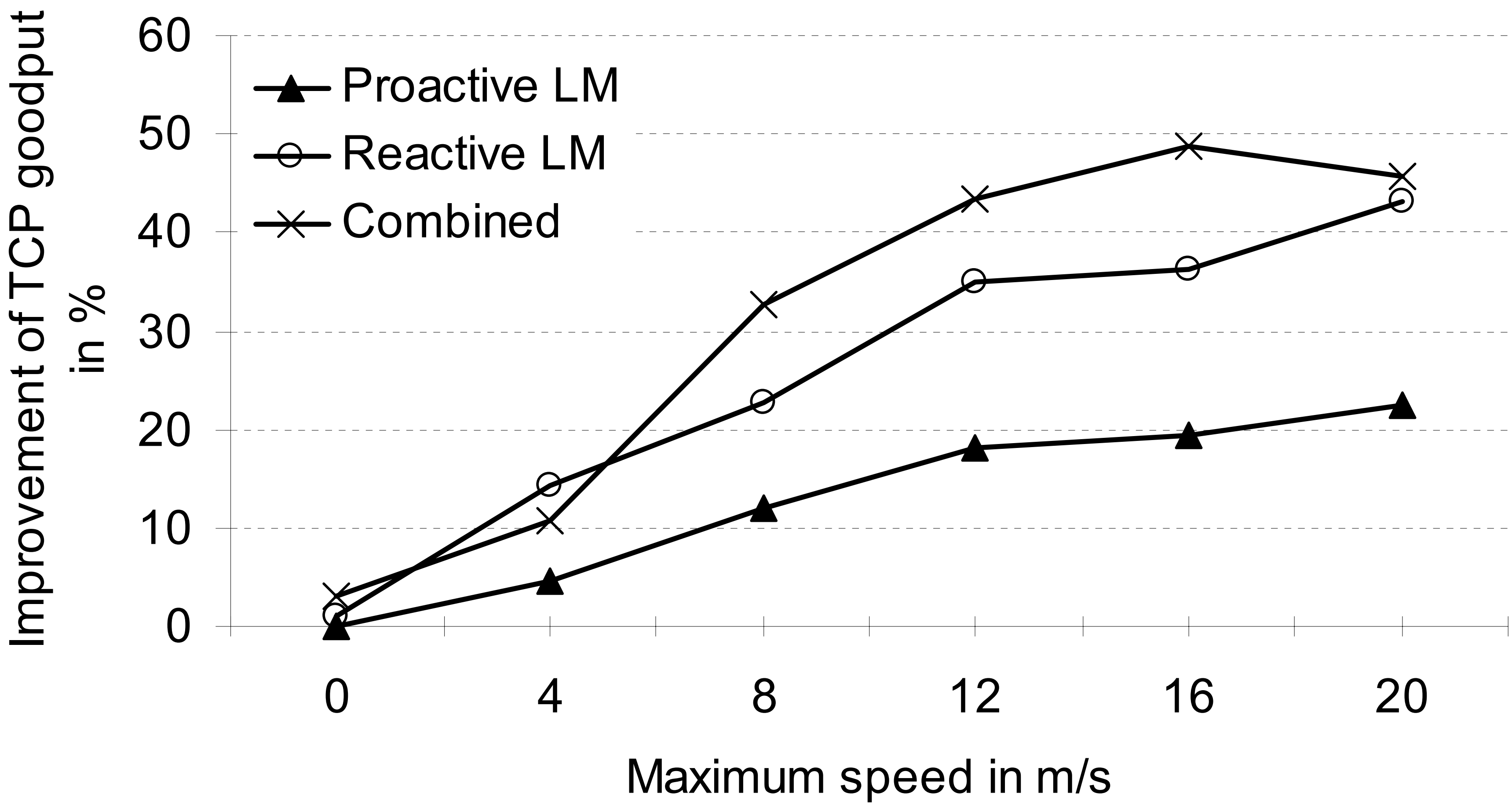
Salvaging Packets

- Routing Layer
  - Three route states:
    - Down: no route
    - Up: route ok, answer route requests
    - Going Down: "weak route", use only to salvage packets, do not answer route requests! → NEW!

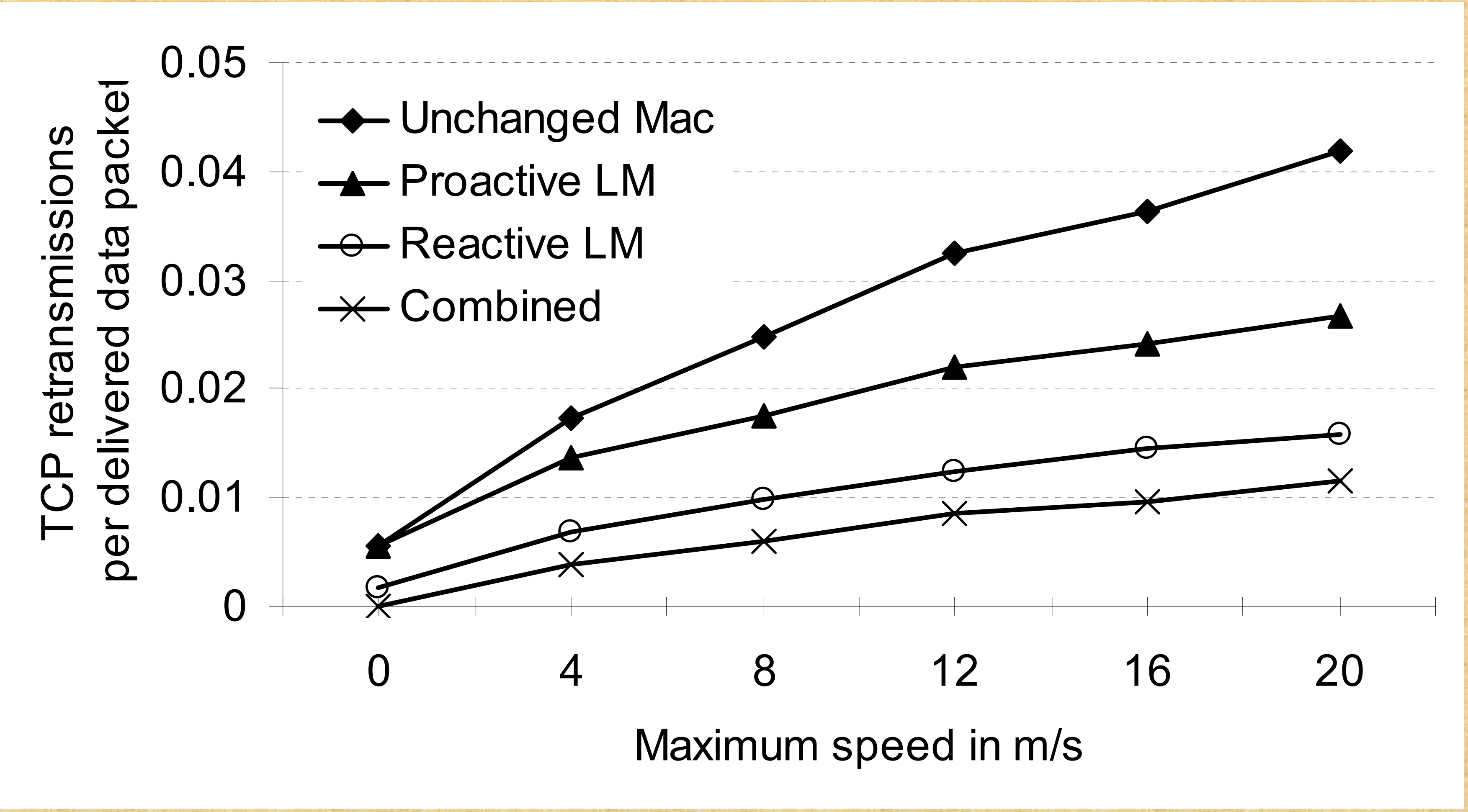
Results – Packet Loss



Results - Goodput



Results – Retransmissions per transmitted packet



Conclusions & Future Work

- The methods proposed seem to improve TCP performance by as much as 40 % in the scenarios considered.
- The reactive scheme might cause problems in highly congested scenarios – especially when the network is dense.
- More sophisticated methods may be needed to clearly differentiate between link failures due to mobility and congestion.
- A node might need to more intelligently decide upon when to increase its transmission power level.

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Thank You