

CS 525M – Mobile and Ubiquitous Computing Seminar

RTS/CTS-Induced Congestion in Ad Hoc Wireless LANs

Saikat Ray, Jeffrey B. Carruthers and David Starobinski

In Proc. IEEE WCNC 2003

Presented by Mingzhe Li

Introduction

- RTS/CTS-Induced Congestion in Ad Hoc WLAN
 - 802.11 ad hoc network uses Multiple Access with Collision Avoidance for Wireless (MACAW)
 - RTS/CTS used to avoid packet collisions
 - RTS/CTS cause “*blocking problem*”
 - *False blocking problem*
 - *Pseudo-Deadlock*
- Solution – RTS Validation
 - Reduces false blocking problem
 - Increases throughput
 - Reduces delay

Outline

- Introduction
- RTS/CTS-Induced Congestion
- RTS Validation
- Simulations
- Summary

RTS/CTS

- Four Way Handshake
 - Request-to-Send (RTS)
 - Clear-to-Send (CTS)
 - Data
 - Acknowledgement (ACK)
- Network Allocation Vector (NAV)
 - Any node heard RTS/CTS is prohibited from transmitting
 - NAV period is set to the *duration* field in RTS/CTS
- Backoff and retransmission
 - RTS/DATA get no CTS/ACK
 - Exponential backoff
 - Retransmissions

RTS/CTS

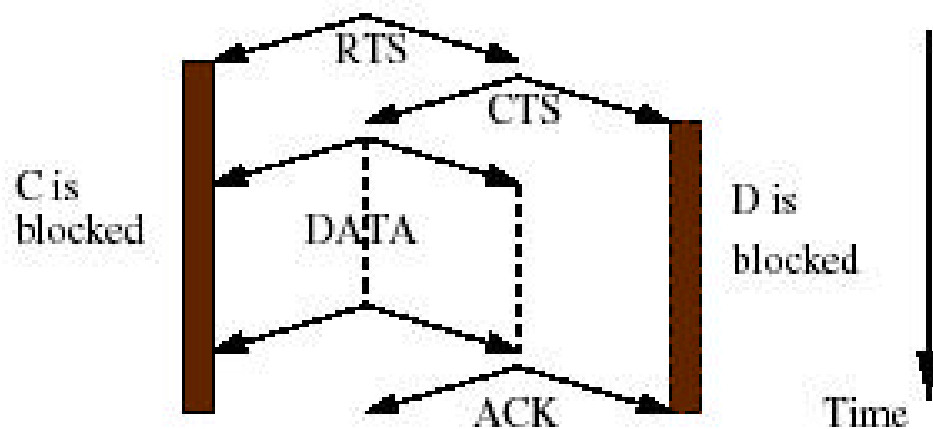
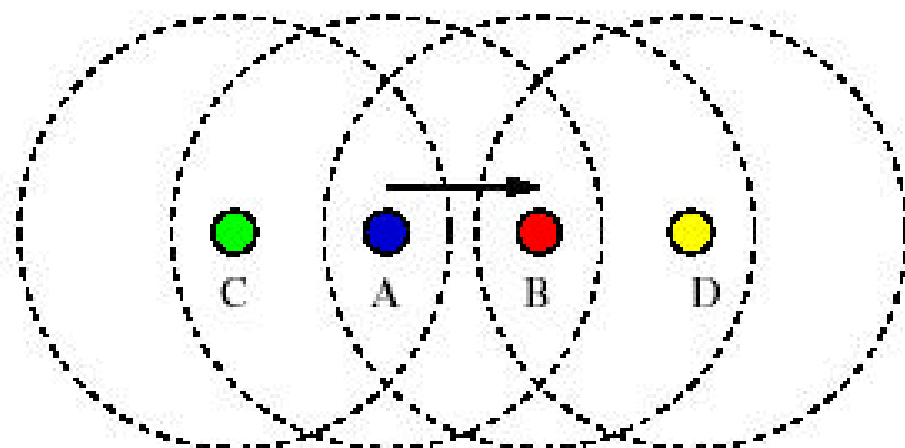


Fig. 2. IEEE 802.11 MAC. The lower half depicts the time-line. The dark bars below node *C* and *D* indicates their NAV.

Blocking Problem

- Blocking Problem
 - A node received RTS and set NAV
 - Blocked node will not response to RTS
- Effects
 - Avoid collisions
 - Exponential backoff
 - Under-utilizes the network

Blocking Problem

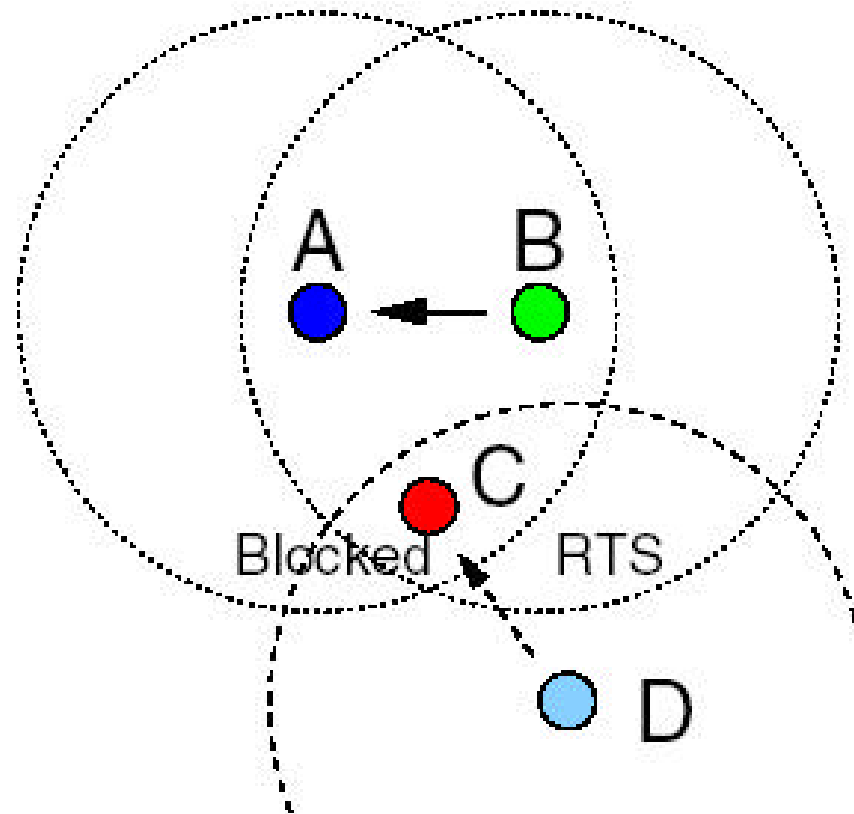


Fig. 3. Blocking Problem. Node *C* is blocked due to the communication between node *A* and node *B*. Therefore, node *D* does not get any response to the *RTS* packets it sends and enters backoff.

False Blocking Problem

- False Blocking Problem
 - A node could be blocked even if no other node is actually transmitting (No DATA packet)
 - May propagate through the network
 - Affects network performance severely
- Pseudo-Deadlock
 - False clocking takes place along a “circular” path
 - Packet drop after certain number of retries
 - Affect higher layer protocols (TCP/UDP)
 - Significantly affect the network performance

False Blocking Problem and its propagation

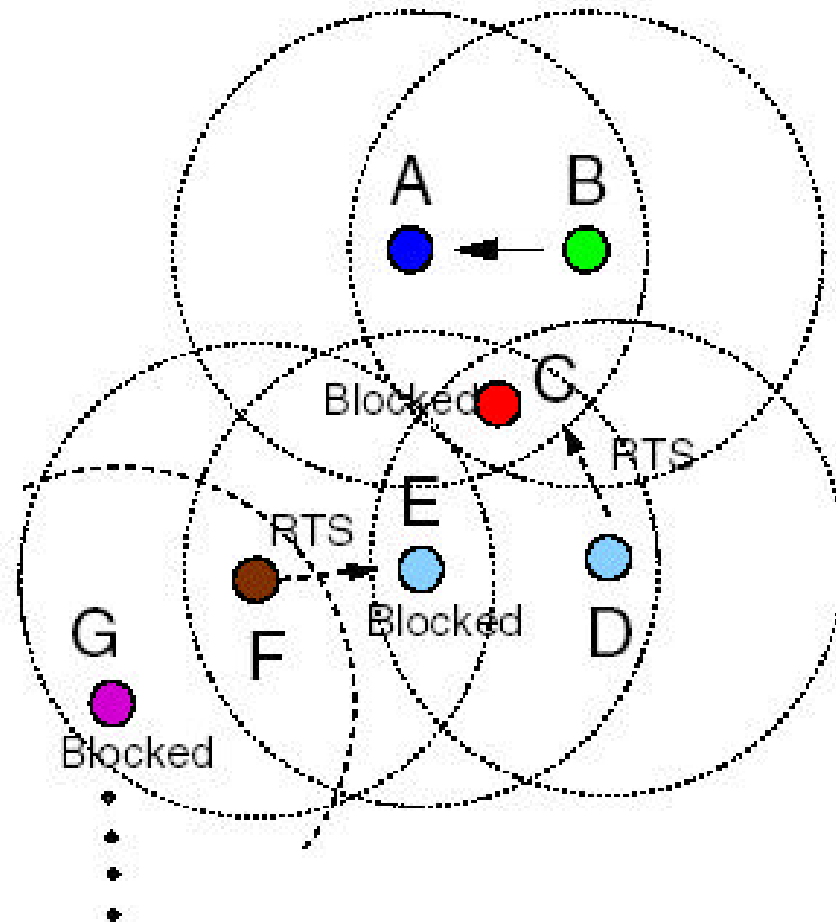
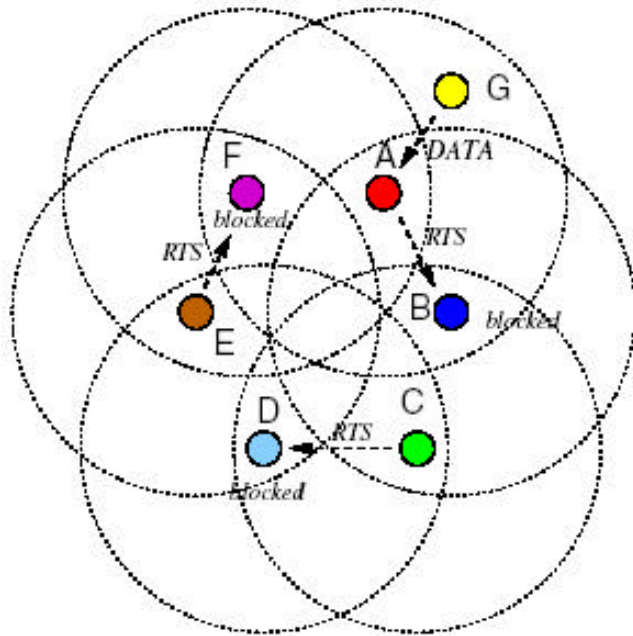
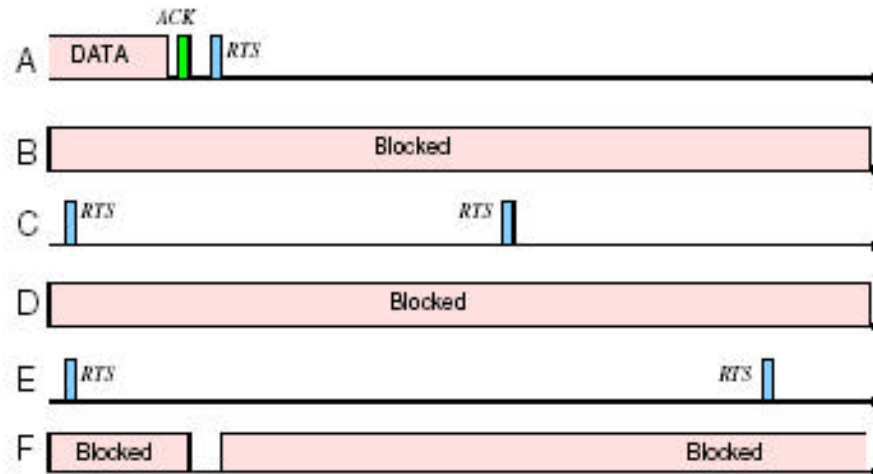


Fig. 4. False blocking problem. Node *E* is unnecessarily blocked because of node *D*'s *RTS*. Therefore, node *F* does not get any response to its *RTS*, which in turn blocks node *G* and so on.

Pseudo-Deadlock



(a) Topology.



(b) Timing.

Fig. 5. Pseudo-Deadlock. Initially node *A* receives a packet from node *G*. This transmission creates the sequence of blocked and deferring nodes $\{F, E, D, C\}$. After *G*'s transmission is over, node *A* transmits an *RTS* packet to node *B*. However, node *C*'s *RTS* sent to node *D* blocks node *B* while node *A* receives packet from node *G*, therefore, node *A* does not get response to its *RTS*. Now, every node tries to transmit to a blocked node and a deadlock occurs.

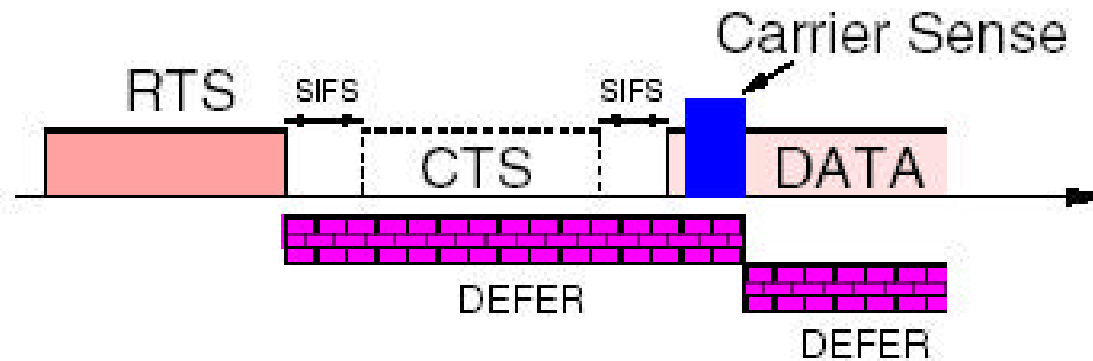
Outline

- Introduction
- RTS/CTS-Induced Congestion
- RTS Validation
- Simulations
- Summary

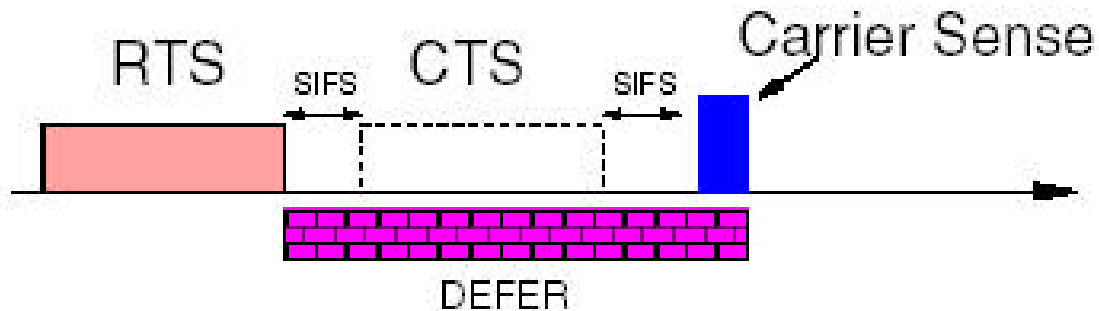
RTS Validation

- Detect false blocking problem
 - False blocking node won't receive DATA packets after RTS
- RTS Validation
 - On receive a RTS not destined itself
 - Defer for *RTS_Defer_Time*
 - Sensing the network after the defer for *CCA_Time*, while continuing deferring
 - If network is idle (No DATA), stop deferring
 - Else continue to defer for a total deferral time equals *Requested_Defer_Time*
 - *RTS_Defer_Time*: as small as possible, but should be longer enough to detect DATA packet
 - *CCA_Time*: Clear-Channel Assessment Time
 - *Requested_Defer_Time*: Deferral time requested by RTS
 - Backward Compatible to regular RTS/CTS mechanism

RTS Validation



- (a) The node senses busy channel following *RTS* Defer Time and therefore continues deferral.



- (b) The node assesses idle channel following *RTS* Defer Time and therefore defers no longer.

Outline

- Introduction
- RTS/CTS-Induced Congestion
- RTS Validation
- Simulations
- Summary

Simulation Model

- Simulation Setup
 - MATLAB based simulator
 - *Footprint*: transmission range
 - No interference outside footprint *
 - 10 nodes in each footprint
 - Packet size: 2300 bytes *
 - 1-Mbps capacity
 - 1 hop traffic
 - Successful RTS/CTS guarantee a collision free DATA transmission *
 - No error simulated *
- Load = packets per second per footprint

Simulation Model

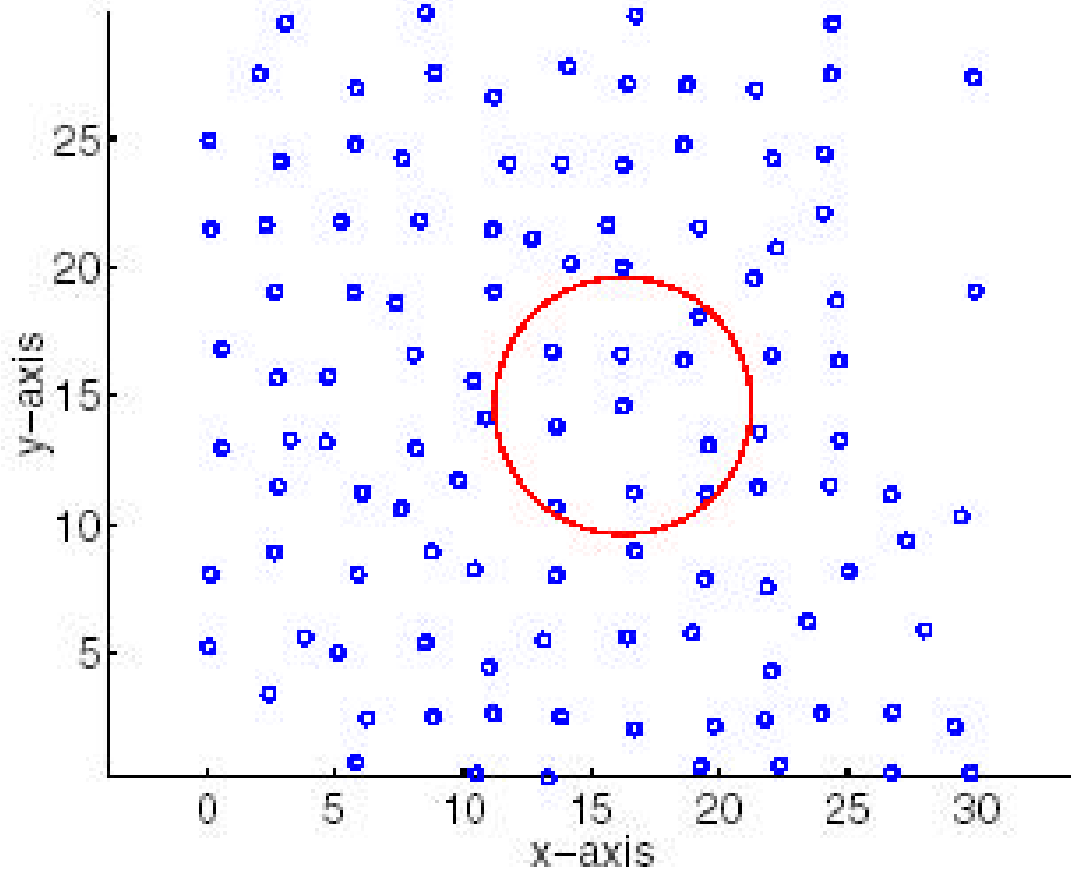


Fig. 7. The network used for the simulation. The circle represents the footprint of the node at the center of the circle. There are 10 nodes per footprint on average.

Simulation Results

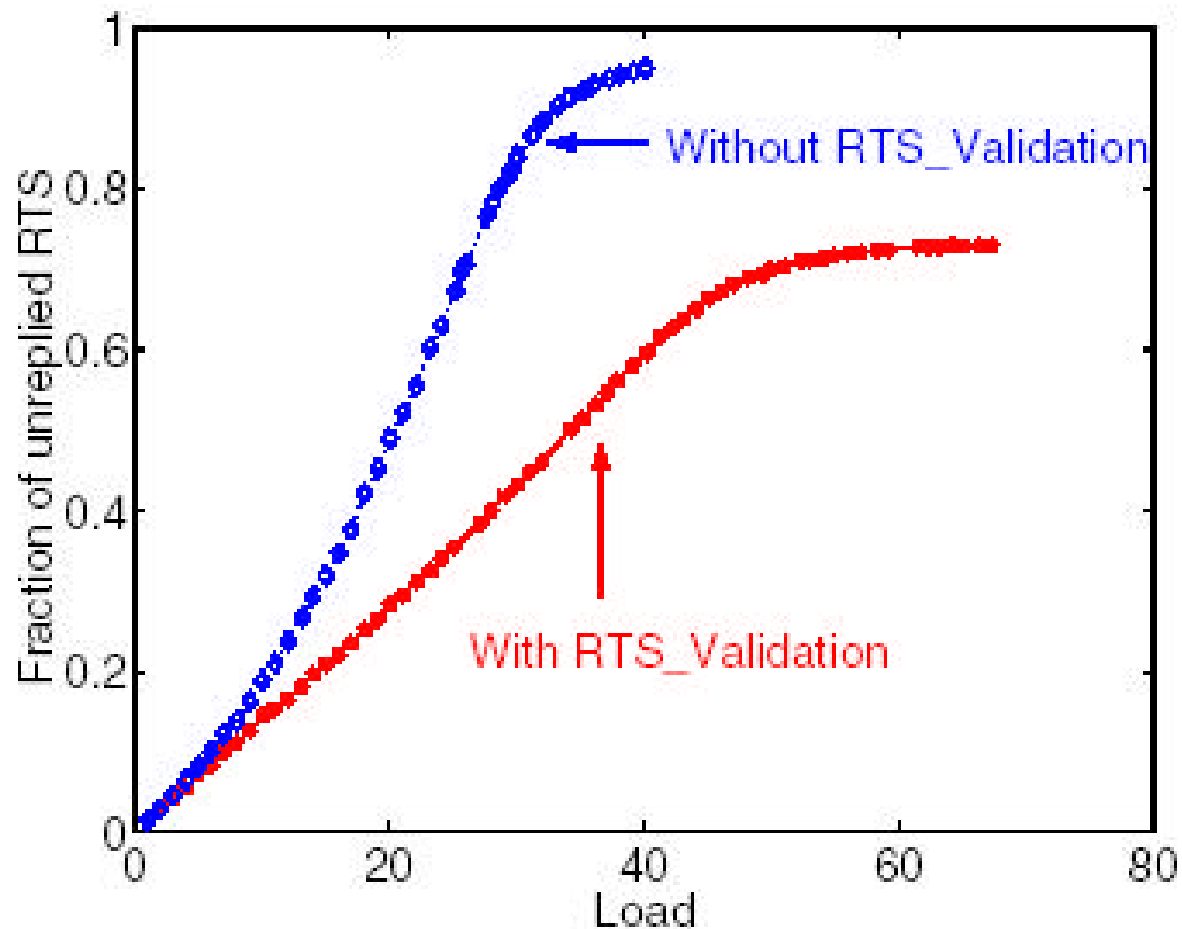


Fig. 8. Blocking: Comparison between the standard and the *RTS* Validation approach. Only successfully received *RTS* packets are counted in this figure.

Simulation Results

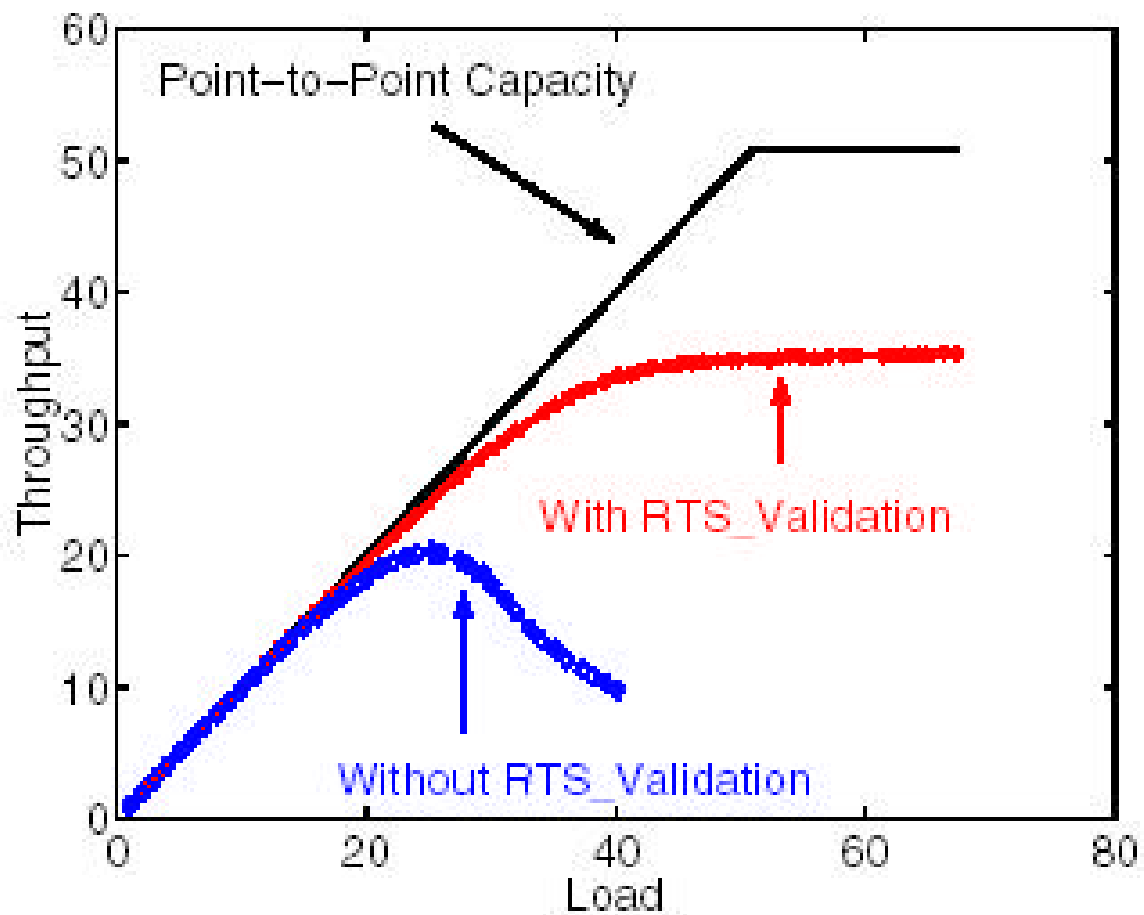


Fig. 9. Throughput: Comparison between the standard and the *RTS* Validation approach.

Simulation Results

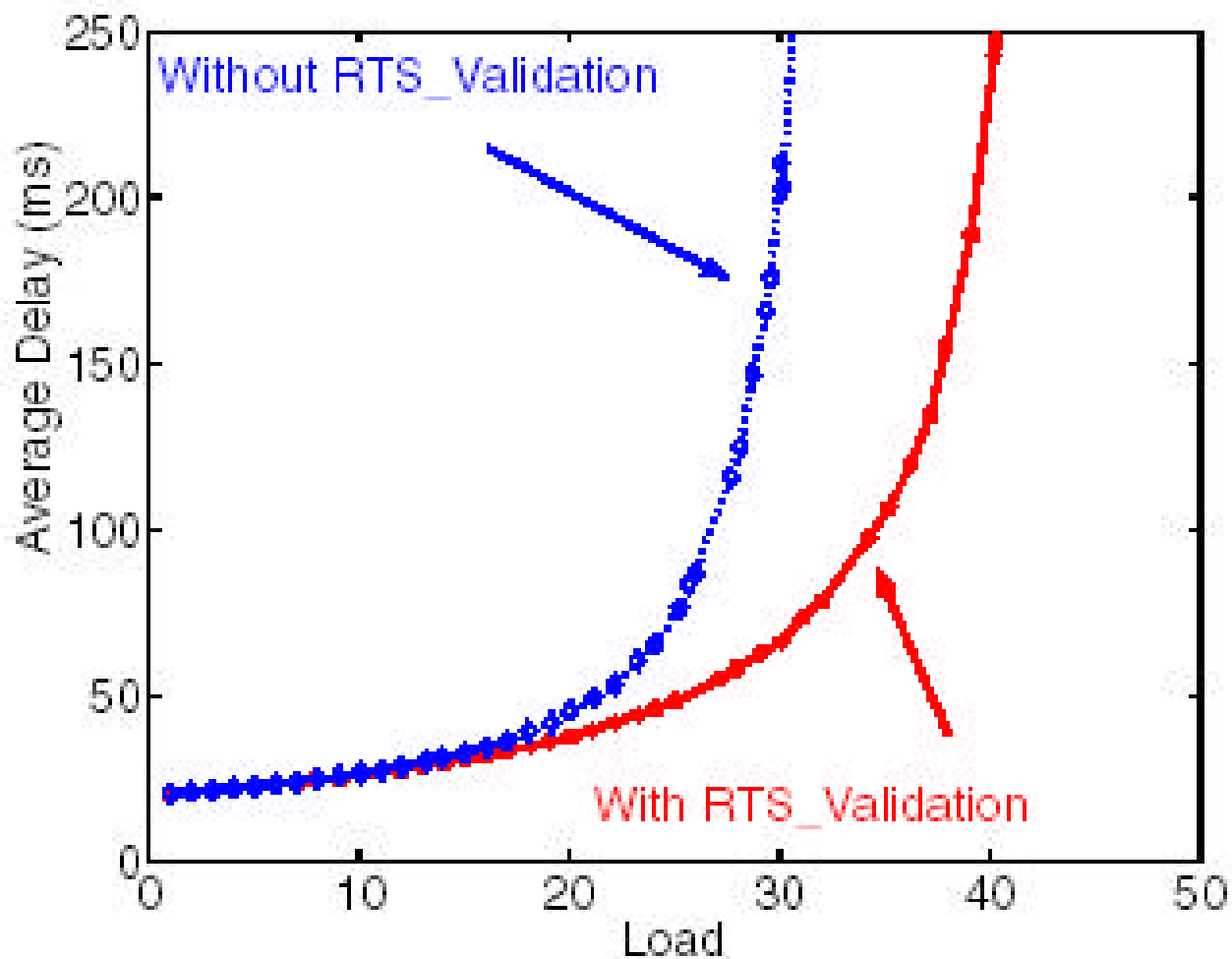


Fig. 10. Delay: Comparison between the standard and the *RTS* Validation approach.

Outline

- Introduction
- RTS/CTS-Induced Congestion
- RTS Validation
- Simulations
- Summary

Summary

- Summary
 - RTS/CTS-Induced Congestion: caused by *“False blocking problem”* and *“Pseudo-Deadlock”*
 - RTS validation
 - Stabilizes the network throughput
 - Increases the peak throughput (60%)
 - Reduce average delay (Packet delivery time)

Thanks!

Questions?