A Bidirectional Multi-channel MAC Protocol for Improving TCP Performance on Multihop Wireless Ad Hoc Networks

Present by Lin Yu-Chen
11/18/2004

form：Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems

Outline

- Introduction
- Problem Definition and Motivation
- Proposed BI-MCMAC Protocol
- Simulations and Performance Evaluation
- Conclusions and Discussions

Introduction

- In a multihop wireless network, the limited radio transmission range of each node often requires packets to traverse multiple intermediate wireless nodes en route from the source to the destination.
- In addition to roles as a traffic source and a traffic sink, each node acts as a store-and-forward router for packet traffic originated from other sources.
- As TCP is used by many applications in the current Internet, it is important to design a MAC protocol that works well with TCP.

Problem Definition and Motivation

- The TCP protocol often suffers from performance problems in conventional single-channel multihop wireless ad hoc networks. The problems arise from hidden node and exposed node issues.
- The popular IEEE 802.11 MAC protocol attempts to solve the hidden node problem by requiring a RTS and CTS handshake between the sender and receiver before actual data frame transmission.

Problem Definition and Motivation

- TCP throughput often decreases dramatically with the number of hops traversed by a flow, even in a static multihop wireless environment, ignoring wireless channel errors and mobility-related link breakages. This trend is particularly evident when many packets are "in flight" at a time.
- The reason is link-layer packet delay/drops caused by *contention* between data packets traveling in the same direction and *collisions* between data packets and TCP acknowledgment (ACK) packets traveling in opposite directions
- The other performance problem is unfairness that can occur between different TCP flows. The unfairness is due to unfortunate interactions between TCP and MAC-layer timers, causing a "rich get richer" and "poor get poorer" phenomenon.

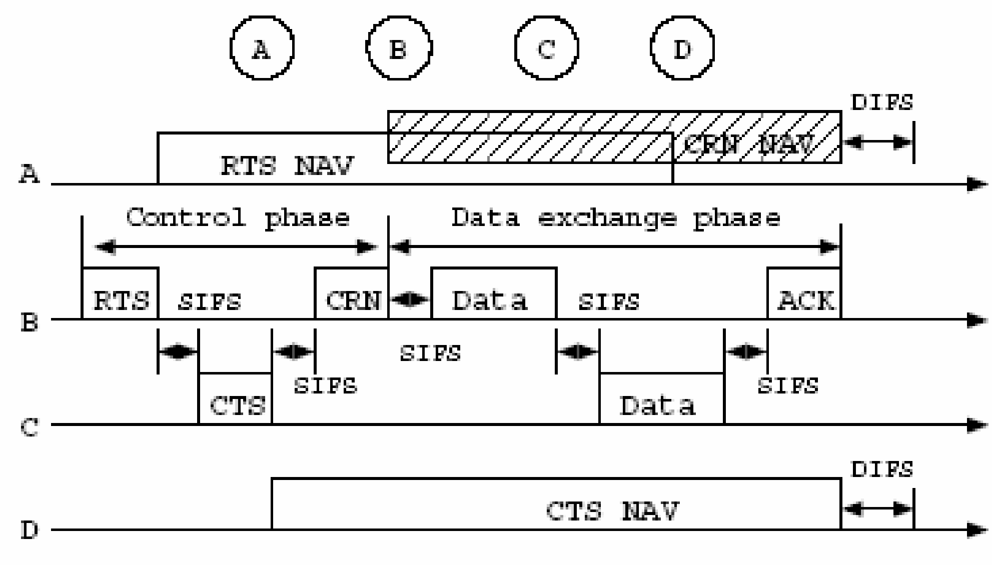
Problem Definition and Motivation

- It propose another method to improve TCP performance in multihop wireless networks use multiple channels in the physical layer.
- This novel bidirectional multi-channel MAC (Bi-MCMAC) protocol, which is designed with a special emphasis on bidirectional TCP traffic issues.
- It reduces the link-layer contention using two key ideas
  - *The protocol uses multiple transmission channels at the physical layer*.
  - *The protocol extends the RTS/CTS handshake to do bidirectional channel reservations*.

The BI-MCMAC Protocol

- In the Bi-MCMAC protocol, physical channels are divided into a control channel and several data channels.
- When a node has a data frame to send, it goes through two phases: the *control phase*, and the *data exchange phase*.

The BI-MCMAC Protocol



- When a sender wishes to send a frame, it sends an RTS frame containing its own free channel list and the estimated transmission duration.
- When the receiver receives the RTS frame, it checks its own local free channels and compares them with the proposed list in the RTS frame.

The BI-MCMAC Protocol

- What's different with other MAC protocol?
  - The receiver will checks to see if it has pending data to transmit to the sender. If so, it increases the channel reservation duration (from RTS) to include the time to transmit its own frame and sets the new value in the CTS frame.
  - When the sender receives the CTS, it extracts the selected channel number and new duration value. This information is then announced in a CRN frame.

The BI-MCMAC Protocol

- Missed reservation problem
  - When a node misses seeing control handshakes initiated by other nodes while it is busy transmitting or receiving a data frame.
- When the same node has new data to send after finishing a data transfer cycle
  - The node initiates a control handshake to a node already in a data exchange phase.
  - The node offers incorrect information, allowing a receiver to select a data channel that is actually in use for a nearby data transfer.

The BI-MCMAC Protocol

- The former case causes the initiating node to back off, since there is no response from the receiving node. The latter case is more severe as it causes a collision on the data channel.
- The probability of a data channel collision depends on the channel selection algorithm used in the channel negotiation procedure.
- In Bi-MAMAC, the receiver always selects the channel used for the last successful transmission, it hopes that if the node uses a previously successful channel, it is likely to succeed again.

Simulations and Performance Evaluation

Parameter Settings for the Simulations

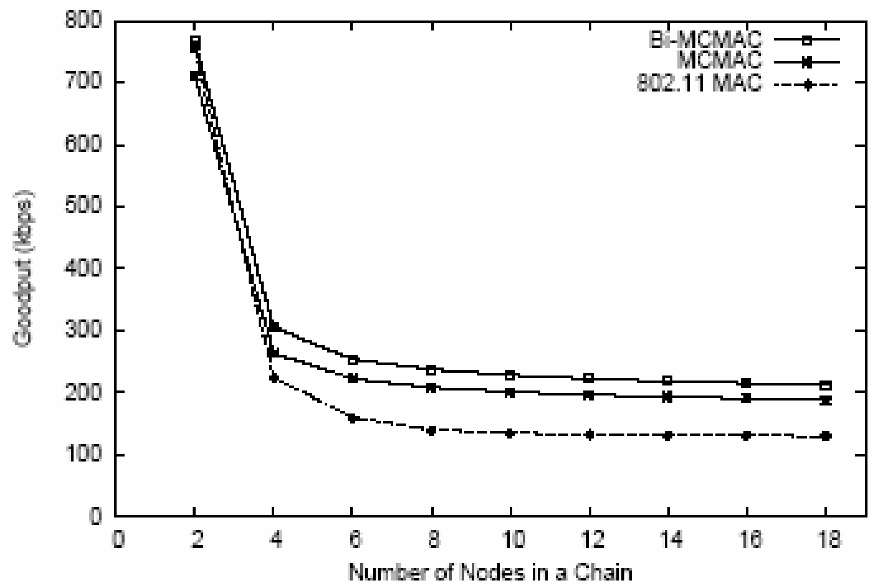
Parameter	Value
Speed	1 Mbps
Number of Channels	4
Channel Model	Two-ray ground
Transmission Range	250 meters
Carrier Sensing Range	250 meters
Ad Hoc Routing Protocol	AODV
Transport Protocol	TCP NewReno
TCP packet size	1000 bytes
TCP Delayed ACK	Enabled

- We have conducted extensive simulations using the ns-2 network simulator to evaluate the TCP performance of the proposed Bi-MCMAC protocol.
- Our performance study focused on *throughput*, *fairness*, and *transfer delay*.

Simulations and Performance Evaluation

- Throughput Experiments
  - The chain Topology
  - The grid topology
  - The random topology
- Fairness Experiments
- Transfer Delay Experiments

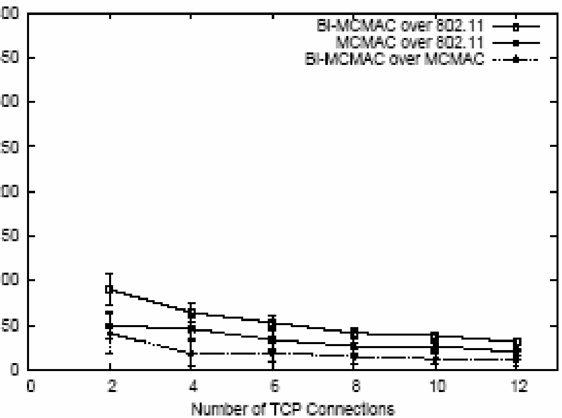
Throughput Experiments



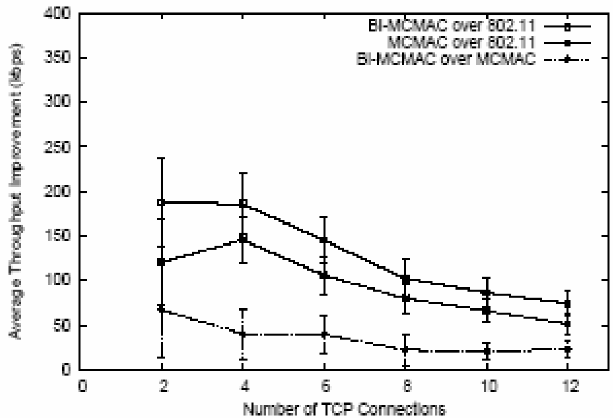
Throughput

- The Bi-MCMAC protocol has slightly better throughput than the 802.11 MAC, despite the additional CRN frame in the handshake. This means the savings of bidirectional reservation can outweigh the CRN overhead.
- the average throughput advantage of the Bi-MCMAC protocol on large chain networks (more than 6 nodes) is about 67.1%.

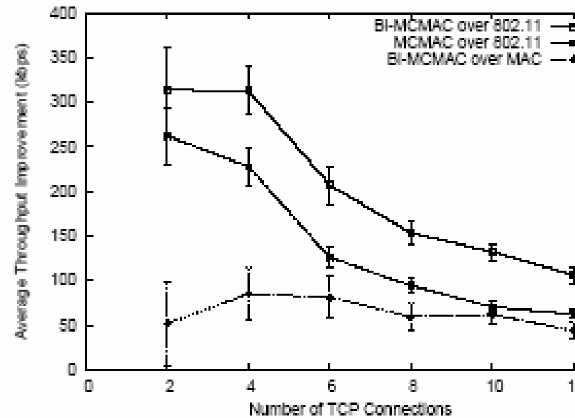
Throughput Experiments



Grid Topology



Sparse Random (500m x 500m)



Dense Random (250m x 250m

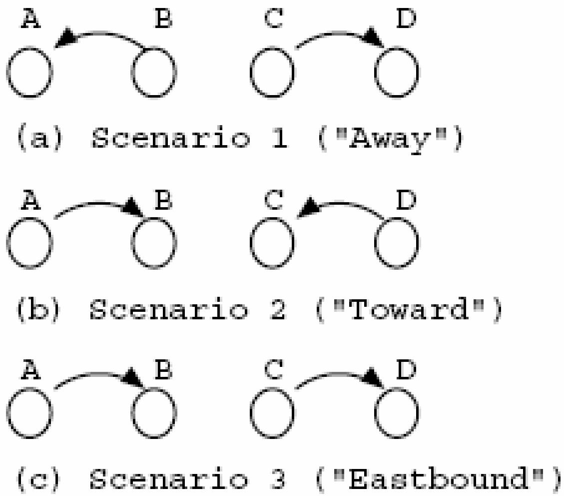
- The results indicate that performance improvements do exist between the protocols, in spite of the increased local contention.
- The multi-channel protocol resolves this contention by allowing more concurrent transmissions. The throughput gains also show that although the missed reservation problem exists, the gain outstrips the potential performance loss.

Fairness Experiments

- It study fairness in two ways.
  - In the strict sense: when two TCP connections traverse similar network paths and face similar competition, they should achieve similar throughput.
  - In the general sense: different flows should have similar throughput when they share the same physical channels, regardless of their local contention.
- Jain's Fairness Index (FI) is used to quantify the fairness. where Ti is the throughput of TCP connection i and N is the total number of connections.

$$FI = \frac{\left(\sum_{i=1}^{N} T_i\right)^2}{N \sum_{i=1}^{N} {T_i}^2}$$

Fairness Experiments



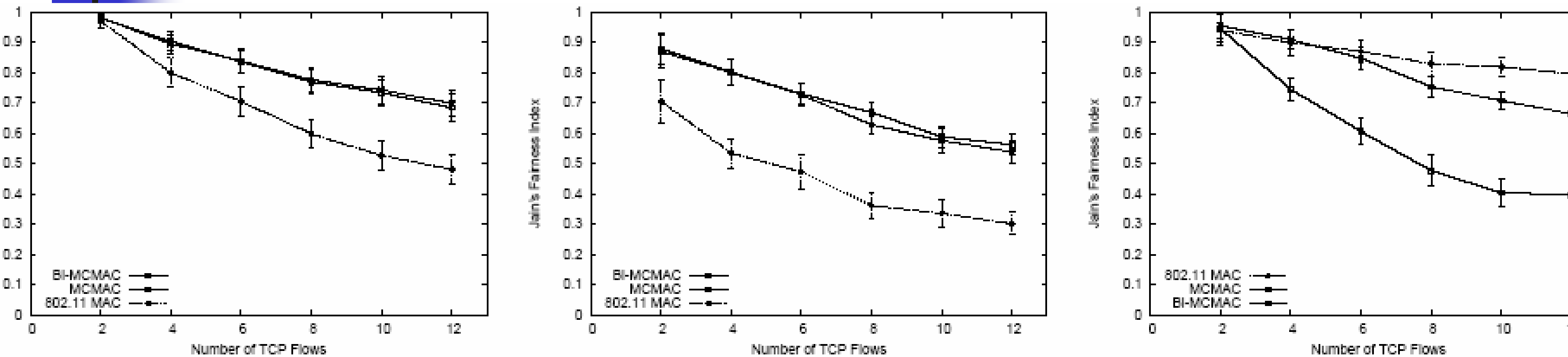
Fairness Testing Scenarios

FI Results for Strict Sense Fairness

	802.11	MCMAC	Bi-MCMAC
Away	0.59	1.00	1.00
Toward	0.81	1.00	1.00
Eastbound	0.50	1.00	1.00

- The multi-channel MAC protocols solve the fairness problem, with an FI of almost 1.00 for all scenarios. With two data channels, the only interference between two flows is on the control channel.
- Since the control channel is not congested in this case, perfect fairness is obtained.

Fairness Experiments



Grid Topology

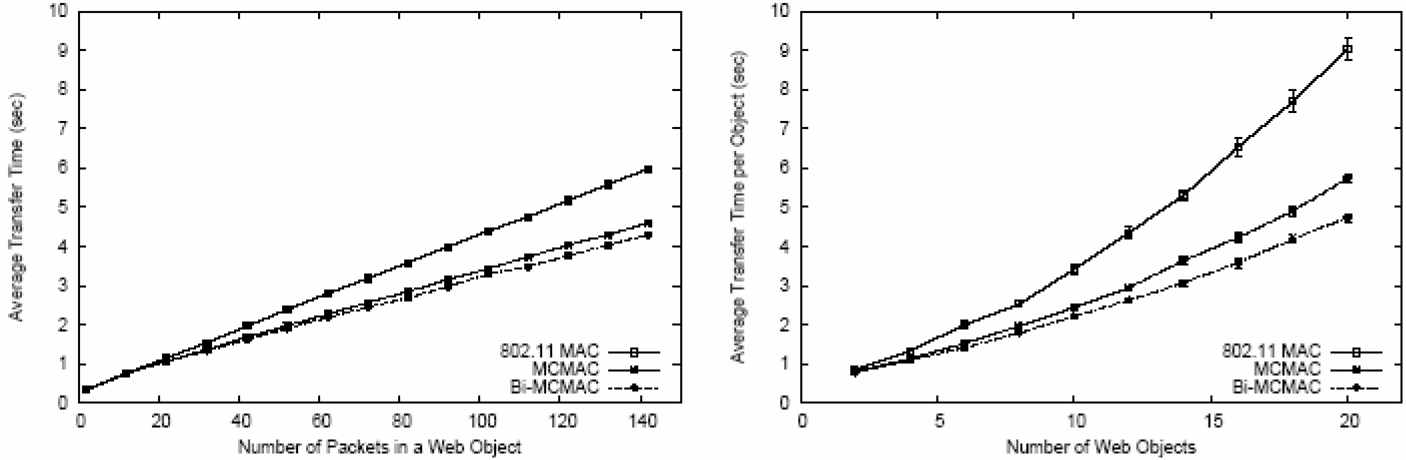
Sparse Random (500m x 500m)

Dense Random (250m x 250m

Fairness for the Grid and Random Topology

- When the number of flows increases, the contention between flows increases, leading to unfairness.
- Since the 250m x 250m topology is approximately a single cell topology, all 100 nodes share the same control channel. When the number of TCP flows is increased, the control channel becomes congested.

Transfer Delay Experiments



Single Web Object

Multiple Web Objects

Simulated Web Transfer Time in a Chain Network

- Unlike the throughput experiments that simulate FTP like data transfers, the transfer delay experiments simulate Web-like transfers.
- With many packets in flight, there are also many TCP ACKs returning in the opposite direction. The Bi-MCMAC protocol has the advantage of transferring a data packet and a TCP ACK in a single handshake, therefore reducing the total transfer time.

Conclusions and Discussions

- In the proposed protocol, after a successful RTS/CTS/CRN handshake, there is only one frame exchanged in each direction.
- Although the protocol design is explicitly targeted at TCP, which induces loosely-synchronized forward and backward traffic (data and ACKs), we expect that our Bi-MCMAC protocol will improve the overall performance for general bidirectional traffic (from different connections) in a multihop wireless network.