Ripple: A Distributed Medium Access Protocol for Multi-hop Wireless Mesh Networks

Ray-Guang Cheng, Cun-Yi Wang, and Li-Hung Liao

Department of Electronic Engineering,
National Taiwan University of Science and Technology,
Taipei, Taiwan, R.O.C
crg@mail.ntust.edu.tw

Abstract—**Wireless mesh network, a new wireless broadband access technology, is currently attracting significant attention. This work proposes a distributed medium access protocol, named *Ripple*, for wireless mesh networks (WMNs) under tree topology. In contrast to existing random-access approaches, *Ripple* uses a controlled-access approach to protect nodes from unintentional packet collisions and maximize the spatial reuse. The performance of *Ripple* under an error-free channel was investigated and the accuracy of the analysis was verified by simulation. Simulation results also indicated that *Ripple* achieved throughput, stability, and QoS enhancement than that of 802.11 DCF under a highly loaded situation in both chain and tree topologies.**

Keywords- medium access control, wireless mesh networks

I. INTRODUCTION

In a multi-hop wireless network, communication between two nodes is carried out through a number of intermediate nodes via relaying packets from one node to another. In the past few years, many researchers have focused on issues of 'mobile ad hoc networks (MANETs)', in which relaying nodes are in general mobile, and communication is performed between arbitrary pair of nodes within the same network. Recently, an increasing number of multi-hop wireless deployments and proprietary commercial solutions have focused on a class of networks termed 'wireless mesh networks (WMNs)' [1-4]. One of the potential applications for WMN is to provide high-speed wireless backhaul links that offers low-cost public access services in outdoor environment. To forego costly wired infrastructure, a WMN adopting a tree topology with a single entry point to the wired Internet will be considered herein [1]. Unlike MANET, the WMN serves as an access network that employs multi-hop wireless links provided by non-mobile nodes to relay traffic to and from wired Internet [2]. The non-mobile nodes (also referred as transit access points [1], wireless routers [2], or mesh points and mesh access points [3]) forms a wireless backbone and provides multi-hop connectivity between nomadic users and the entry point(s) (also known as mesh portals [3]) to the wired Internet [2]. In such an environment, power consumption is not a primary concern since relaying nodes are fixed and wire-powered. Due to the lack of a centralized coordinator, each relaying node should be operated in a fully distributed manner which results in inevitable packet collisions and may degrades the network throughput. Hence, one of the main challenges of WMN is the provisioning of a proper medium access control (MAC) protocol that can coordinate the channel access among neighboring nodes base on limited information exchange.

Many researches has been studied the limitation of adopting 802.11 MAC in wireless ad hoc or mesh environment. Currently, IEEE 802.11 MAC uses the virtual carrier sensing with ready-to-send/clear-to-send (RTS/CTS) handshake to alleviate packet collisions due to hidden node problem. However, it may not be applicable in WMN. Jangeun and Sichitiu indicated that RTS/CTS does not correctly solve hidden terminal problem in a mesh network [4]. Xu and Saadawi [5] found that RTS/CTS scheduling along a chain can cause serious TCP fairness problems and backoff inefficiencies. Li *et. al.* [6] found that RTS/CTS does not efficiently schedule transmissions and fails to achieve good schedule in a multi-hop chain. Xu *et. al.* [7] found that 802.11 MAC tends to either sacrifice spatial reuse or allow excessive interference.

Several techniques were proposed to enhance the network utilization. Jain, *et. al.* [8] proposed a multi-channel MAC protocol to mitigate the exposed node problem. Acharya and Misra [9] proposed a MACA-P method which adopts spatial-reuse technique to improve channel utilization. They also proposed a data-driven cut-through medium access (DCMA) method to reserve channel form the next forwarding node and thus, reduce the chance of packet collisions. Raguin, *et. al.* [10] adopted the concept of DCMA and proposed a queue-driven cut-through medium access (QCMA) method for a multiple-queue environment. In QCMA, each node may select the highest priority packet from its queue and thus, a certain degree of QoS can be supported. To sum up, existing approaches enhance the network utilization by reducing hidden and expose nodes, adopting spatial-reuse, or utilizing cut-through techniques. However, existing approaches still adopted a random-access mechanism with exponential backoff and thus, they all suffer from the same backoff inefficiencies and fairness problems as that of 802.11 DCF. Inspiring by the IEEE 802.4 token bus protocol, Ergen, *et. al.* [11] proposed a wireless token ring protocol (WTRP) to eliminate the

inefficiencies. WTRP aimed at guaranteeing the quality of service (QoS) of stations rather than the throughput enhancement. Therefore, the spatial reuse was not applicable.

In [12], we proposed a distributed medium access protocol, *Ripple* to enhance the throughput of WMN through spatial reuse. Different to existing random-access-based protocols, *Ripple* adopts a token-passing mechanism to prevent nodes from unintentional packet collisions backoff and thus, enhances the network utilization. This paper further extends our work in [12] to a tree topology. The rest of this paper is organized as follows. The proposed *Ripple* protocol is described in Section II, and its key parameters and their effect on the system performance are discussed. Section III presents the simulation results. Conclusions are finally drawn in Section IV.

II. *RIPPLE* PROTOCOL

Li *et al.* proved that a node in the chain topology may attain an optimal utilization of 1/3 by applying spatial-reuse [6]. They also predicted that, data frames could be forwarded hop-by-hop without interfering with each other if each node can properly schedule its frame transmission interval. Under this optimal condition, the frame-forwarding resembles ripples of water moving apart from a central location, which is referred as "ripple phenomenon" herein. However, the optimal condition is hard to be realized in WMN because the lack of a central coordinator and the presence of hidden nodes and expose nodes [2] in WMN. The objective of this work is to propose a wireless token-passing protocol, named *Ripple*, to coordinate frame transmission for nodes in WMN and thus, enable the ripple phenomenon in the chain and tree topology.

Before going into details, a generalized system model of a WMN adopting a tree topology is considered herein, as depicted in Fig. 1. Each node in Fig. 1 is a wireless router which can generate traffic by itself and forward traffic for other wireless routers. In the rest of this paper, a node that connects the WMN to a wired Internet gateway is denoted as a root node (i.e., Node A_1); a node that has more than one child nodes is denoted as a cross node (i.e., Node C_1), and a node that has no child node is denoted as a leaf node (i.e., Nodes B_j, D_k, E_m) for easy reference. Without loss of generality, the WMN can accommodate one root node and more than one cross nodes and leaf nodes. In the tree topology, the downlink transmission refers to packets transmission along the direction from a parent node to its child node(s). In contrast, the uplink transmission refers to packets transmission along the direction from a child node to its parent node. To simplify the description, the following assumptions are first made:

- Nodes are stationary [6]. That is, the location of each node is fixed. Radios of nodes that are not neighbors do not interfere with each other [6].
- Two separate high-speed wireless channels are allocated for downlink and uplink transmissions, respectively. The time required to transmit a data packet is fixed.

Ripple uses six types of frames and four kinds of operational states. The frame format used in *Ripple* includes:

- **DATA frame:** DATA frame carries user information. The time required to transmit the DATA frame is a constant.
- **NULL frame:** NULL frame is a DATA frame but carries no information.
- **RTS frame:** a node, which has the right to send a DATA frame, will send an RTS frame.
- **CTS frame:** the target node that receives an RTS frame will responses a CTS frame.
- **Ready-to-receive (RTR) frame:** a node, which has the right to receive a DATA frame, will send an RTR frame to the sender if the expected RTS frame is not received.
- **ACK frame:** a node receives a DATA frame correctly will response an ACK frame.

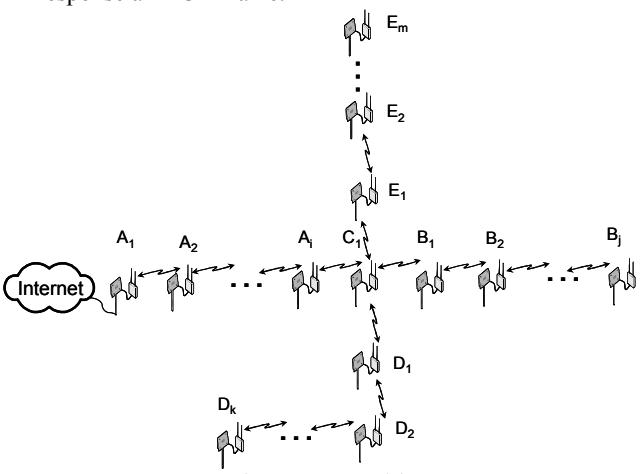


Fig. 1. System model

The frame format of RTS, CTS, DATA, and ACK frames are the same as that defined in 802.11 MAC. The frame format of an RTR frame resembles that of a CTS frame. The inter-frame-spaces (IFSs) of DATA, RTS, CTS, and ACK frames are all set to be the short-IFS (SIFS) as that defined in 802.11. The IFS for RTR frame will be defined later. The four operational states are:

- **Transmit (TX) state:** a node which is ready to send a DATA frame will enter this state.
- **Receive (RX) state:** a node which is ready to receive a DATA frame will enter this state.
- **Listen state:** a node which is a hidden node (i.e., CTS frame is overheard) or an exposed node (i.e., RTS frame is overheard) or both will enter this state.
- **Idle state:** a node which has interrupted by unexpected conditions during TX, RX, and Listen states will return to this state. The idle state is also the initial state for all nodes.

Among them, TX, RX, and Listen are normal states while Idle is the only abnormal state.

The downlink packet transmissions of *Ripple* is described below. The result is then extended to accommodate uplink packet transmissions. In *Ripple*, the operation of each node depends only on its state machine. The state transition of each node is triggered by the transmission or reception of a frame. *Ripple* adopts a controlled access mechanism to eliminate the backoff inefficiencies. Similar to the token-passing protocol, *Ripple* utilizes a special frame as a "token" to authorize a node for sending DATA frame. A tagged node that has the token at hand is eligible to perform RTS/CTS handshake with the next

forwarding node. The RTR frame is designed to regenerate the token if the RTS frame is lost. That is, the next forwarding node may actively send an RTR frame to the tagged node and request for a DATA frame after RTS timeout. Hence, the IFS of RTR frame, denoted by IFS_{RTR}, is set to be

$$IFS_{RTR} = SIFS + T_{RTS} + SIFS = 2SIFS + T_{RTS}, \quad (1)$$

where T_{RTS} is the time required by a node to transmit an RTS frame.

In the downlink transmission, *Ripple* utilizes RTS or RTR frame as the token. That is, a tagged node that receives an RTS or an RTR frame has the right to transmit a DATA packet to the next forwarding node (i.e., the child node of the tagged node). Note that a cross node may use any pre-defined scheduling algorithm for selecting a child node to be the next forwarding node. Figure 2 depicts the finite state machine of *Ripple* for downlink packet transmissions. Except for the root node, leaf nodes, and child nodes of cross node, each node is normally circulated among TX, Listen, and RX states in turn. Hence, each node will have the equal chance to utilize the shared wireless medium in the downlink data transfer. However, the root node is normally circuited among TX, Listen, Idle states because it receives the downlink packets from the Internet gateway through the wire. Similarly, the leaf nodes are normally circuited among Idle, Listen, RX states because they do not have child nodes for forwarding downlink packets. Consider a tagged node running at TX state, the tagged node sends a DATA (or NULL) frame to the next forwarding node and expects to receive an ACK frame. Note that the tagged node may not have to forward the same DATA frame, which is received from its parent node, to the next forwarding node. That is, each node may schedule its packet transmission based on the QoS requirement of packets stored in its queues. By overhearing the RTS/CTS handshake or RTR frame broadcasted by its neighboring nodes, the tagged node will move to Listen state. At Listen state, the tagged node has to keep silent for network-allocation-vector (NAV), which is specified in the duration field of the overheard frame header. The tagged node will expect to receive an RTS frame from its parent node after the expiry of the NAV. In case that the RTS frame is lost, the tagged node may actively send an RTR frame to its parent node to request for transmission after RTS timeout. That is, *Ripple* utilizes RTR frame to survive from RTS transmission failure. In order to exploit spatial reuse, *Ripple* also utilizes an RTR frame to trigger new ripples for nodes at Idle state. Note that child nodes of a cross node are not allowed to transmit the RTR frame to the cross node in order to prevent from the collision of RTR frames at cross node. The tagged node moves to RX state when it is ready to receive a DATA frame. At RX state, the tagged node will receive a DATA (or NULL) frame from its parent node and then response an ACK frame. With the RTS token at hand, the tagged node has to initiate RTS/CTS handshake with the next forwarding node. The tagged node should wait for SIFS and move to TX state again. Note that, during any state, the tagged node is forced to enter Idle state whenever unexpected errors occur.

Figure 3 illustrates the state transitions and the timing diagram for a chain of nodes. Initially, all nodes in WMN use 802.11 DCF to communicate asynchronously among themselves to elect a supernode, which possesses the only RTS token in WMN, and then move to Idle mode. In Fig. 3, Node *n+3* is elected as the supernode at time t_0. Similar to 802.11 DCF, Node *n+3* performs RTS/CTS handshake with its downstream node (i.e., Node *n+4*) before transmitting a DATA frame (i.e., a NULL frame is transmitted if all of its queues are empty). With RTS token at hand, Node *n+4* can sends a DATA frame to its downstream node after performing RTS/CTS handshake. The process continues such that DATA frames are forwarded hop-by-hop without interruption, activating the ripple phenomenon. *Ripple* utilizes RTS/CTS handshake for triggering neighboring nodes' state transition; preventing hidden node and exposed node problems [3], and passing RTS token to the downstream node. For example, at time t_0, Nodes *n+2* and *n+5* are forced to enter Listen state by overhearing RTS and CTS frames, respectively, and have to keep silence for NAV. Figure 3 demonstrates that *Ripple* utilizes RTR frame to activate a new ripple if its upstream node is in Idle state (e.g., Node *n*+2 at time t_1) or re-generate an interrupted ripple if its upstream node is in TX state (e.g., Node *n*+5 at time t_2). The left-hand side of Fig. 3 further illustrates the behavior of nodes observed at time t_3. It showed that *Ripple* enables the ripple phenomenon such that two nodes distanced from a spatial-reuse distance of three nodes can transmit simultaneously without affecting each other.

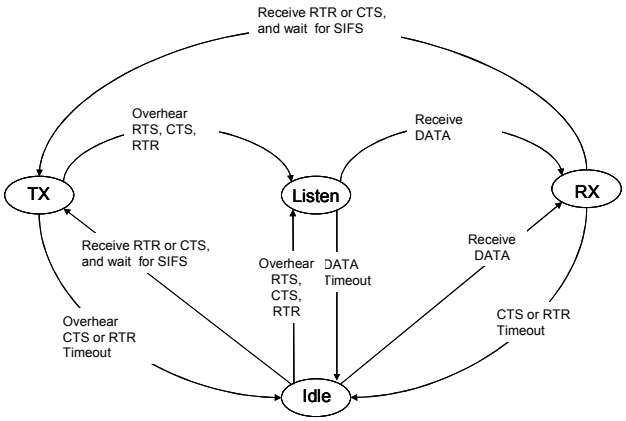


Fig. 2. Finite state machine: Downlink transmission

The message sequence chart detailing the operation of *Ripple* in the presence of a cross node is shown in Fig. 4. For simplicity, a tree with only two branches was demonstrated (i.e., $i = 3$, $j = 3$, $k = 5$ and $m = 0$ in Fig. 1), as illustrated at the top of Fig. 4. In the example, the cross node (i.e., Node C_1) utilized a Round-Robin method to choose the next forwarding node. The propagation delay between two adjacent nodes was not depicted. In the figure, the solid lines refer to message exchange between two communication parties while the dash lines refer to messages overheard by neighbors of the communication parties. It can be found in Fig. 4 that the RTR frame can be used (e.g., by Node A_2) to generate a new ripple or (e.g., by Node C_1) to resume an interrupted ripple from RTS transmission failure. As demonstrated, the two branches (i.e., one for node Bs and the other for node Ds) are independently operated thanks to spatial reuse. Note that Node B_1 and D_1 (i.e., the child nodes of the cross node) are prohibited to send RTR frame to Node C_1. However, both Node B_2 and D_2 are free to trigger a new ripple if the channel is sensed idle for IFS_{RTR} after

the expiry of the NAV at Listen state. The figure also illustrates that nodes which are separated from a spatial-reuse distance of three hops (e.g., Nodes A_2, D_1, and D_4) can be simultaneously transmitted. That is, the optimal channel utilization of 1/3 is achieved [6]. The result can be easily extended to accommodate a WMN adopting different scheduling algorithm, different values of propagation delay (i.e., the operation of *Ripple* may not be affected by the variation of propagation delay because the location of each node is fixed), or variable packet lengths (i.e., local synchronization among neighboring nodes can still be guaranteed through the overheard NAV). The implementation details are not addressed herein due to space limitation.

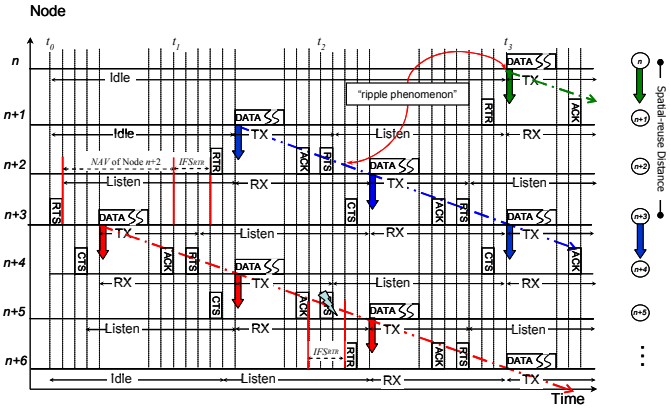


Fig. 3. State transitions and timing diagram: Downlink transmission

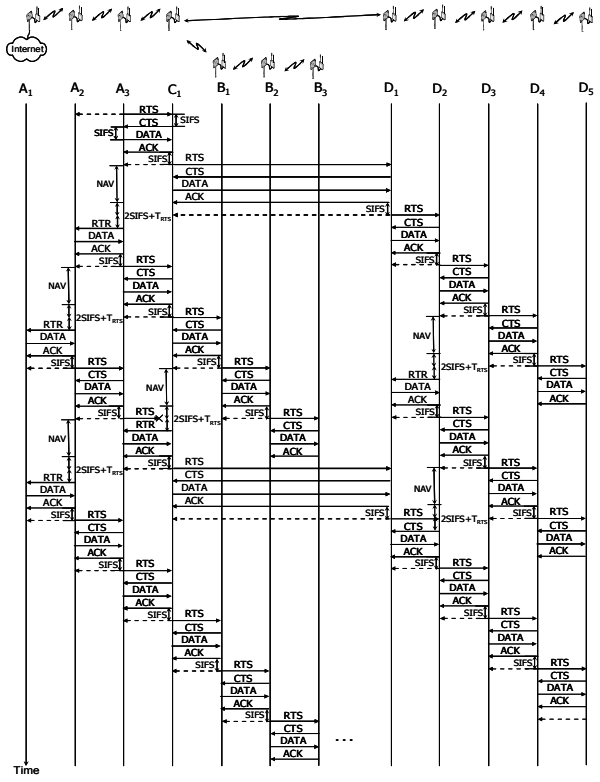


Fig. 4. Message sequence chart: Downlink transmission

The downlink packet transmission procedure can be slightly modified to support the uplink packet transmission. In the uplink transmission, the roles of sender and receiver are interchanged. Different to the downlink transmission, a tagged node that overhears the transmission of a DATA or an ACK frame (i.e., the tagged node is an exposed node) owns the token. Hence, the tagged node is eligible to send an RTS frame to the next forwarding node at the expiry of the overheard NAV. In the uplink transmission, the default next forwarding node is the parent node of the tagged node. The only exception is the child nodes of a cross node, which are not allowed to send the RTS frame even if it owns a token. Otherwise, RTS packets would be collided at the cross node. Details of uplink packet transmission procedure are neglected herein due to the space limitation.

For a tree topology, the performance of each nodes (i.e., the root, leaf nodes, child nodes of cross node are excluded), in *Ripple* under an error-free channel can be easily derived. Denote $p_{i,j}$ as the state transition probability from state i to state j and P_i as the state probability of a tagged node, where $i, j \in$ {TX, RX, Listen, Idle}, respectively. In an error-free channel, $p_{TX,Idle} = p_{RX,Idle} = p_{Listen,Idle} = 0$ and $p_{TX,Listen} = p_{Listen,RX} = p_{RX,TX} = 1$. Thus, it can be shown that $P_{TX} = P_{RX} = P_{Listen} = 1/3$. In other words, a node attains the optimal utilization of 1/3 under spatial-reuse and each node has an equal chance to access the wireless channel. The performance of *Ripple* under noisy channel is rather complicated, which is left as our future work.

III Simulation Results

The simulations in this work were performed using *NS2*, and each sample was obtained by averaging outcomes collected within 1500 seconds. In the simulation, both the transmission range and interference range were assumed to be one-hop radius. The lengths of RTS, CTS, RTR, and ACK frames were 44, 38, 38, and 38 bytes, respectively, and the link capacity was 1 Mbps [6]. A chain of homogeneous nodes that separated by equal distance was first studied. The maximum attainable throughput of the chain under an error-free wireless channel using both *Ripple* and standard 802.11 DCF were investigated. In the following, we will demonstrate the simulation results for the network topology depicted in Fig. 1. Specifically, a chain topology (i.e., $m = k = 0$) with chain length $(i+j+1)$; and, a symmetric tree topology with two branches (i.e., $i = j = k$, $m = 0$, and the angle subtended by two adjacent branches is 120 degrees) and three branches (i.e., $i = j = k = m$, and the angle subtended by two adjacent branches is 90 degrees), both with tree depth $(2i+1)$ were simulated.

Figure 5(a) shows the effective chain throughput (or end-to-end throughput) for various chain lengths and different DATA frame sizes, where the source and sink node were located at two ends of the chain in the simulation. In this example, the traffic source was always backlogged (i.e., offered load = 0.4 Mbps) and the effective chain throughput (or, end-to-end throughput) excluding the control overhead was evaluated at the sink node. The accuracy of the analysis was first verified via simulation. For a chain with only two nodes, it is found that 802.11 DCF attained the maximum throughput of about 0.85 Mbps for 1000-byte frames, because there was no packet collision. However, the throughput of 802.11 DCF for chains with more than three nodes dramatically decreased to 0.1 Mbps, which is far less than one-third of 0.28 Mbps under

spatial-reuse [3]. Conversely, *Ripple* always attained the optimal throughput of 0.28 Mbps, which was irrelative to the chain length. Figures 5(b) illustrates the effective chain throughput of *Ripple* and 802.11 MAC, respectively, under various offered-load conditions for a chain length of 8 nodes. 802.11 DCF attained a maximum throughput of about 0.2 Mbps but dropped below 0.1 Mbps as a result of excess collision under high traffic-load. In contrast, *Ripple* provides a stable throughput even under high traffic-load. Figure 5(c) shows the aggregated throughput of a symmetric tree topology for various tree depths. In this example, the source is located at the root node and the sink nodes are located at the end leaf of each branch. The aggregated throughput is an aggregation of the effective throughput for the two or three branches. Similar to the chain topology, *Ripple* attained a stable throughput while the performance of 802.11 MAC was severely degraded for long tree depth.

III. CONCLUSIONS

In [12], we proposed a distributed medium access control protocol, called *Ripple*, for WMNs. *Ripple* includes features of 802.11 DCF but avoids its backoff inefficiencies and excess collisions. Therefore, it enhances the throughput of WMNs in chain topology. This paper further extends our work in [12] to accommodate a tree topology. The major extension we made is that, in downlink (uplink) transmission, the child nodes of the cross node are prohibited to send RTR (RTS) to prevent from packet collisions. Simulation results showed that *Ripple* achieved throughput and stability enhancement in both chain and tree topologies than that of 802.11 DCF under a highly loaded situation.

REFERENCES

[1] V. Gambiroza, B. Sadeghi, and E. W. Knightly, "End-to-end performance and fairness in multihop wireless backhaul networks," *Proc. of ACM MobiCom*, Sept. 2004.

[2] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: Commodity multihop ad hoc networks," *IEEE Communications Magazine*, pp. 123-131, March 2005.

[3] L. Yang, "Issues for mesh media access coordination component in 11s," *IEEE 802.11-04/0968R13*, January 2005.

[4] J. Jangeun and M. L. Sichitiu, "The nomial capacity of wireless mesh networks," *IEEE Wireless Communications*, pp. 8-14, Oct. 2003.

[5] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?" *IEEE Communications Magazine*, P130-137, June 2001.

[6] J. Li, C. Blake, D. S. De Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," *Proc. of ACM MobiCom*, pp. 61-69, July 2001.

[7] K. Xu, M. Gerla, and S. Bae, "How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks?" *Proc. of IEEE Globecom*, pp. 72 -76, 2002.

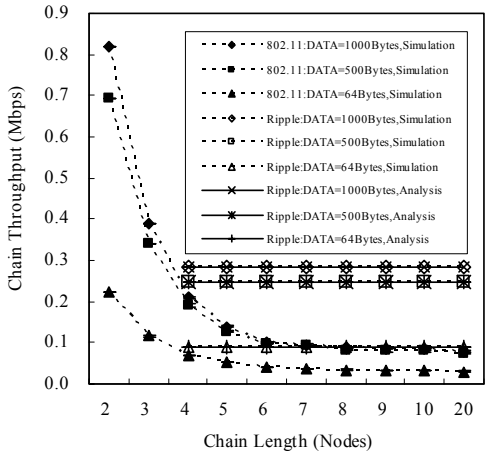
[8] N. Jain, S. R. Das, and A. Nasipuri, "A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks," *Proc. of IEEE ICCCN*, October 2001.

[9] A. Acharya and A. Misra, "High-performance architecture for IP-based multihop 802.11 networks," *IEEE Wireless Communications*, pp.22-28, Oct. 2003.

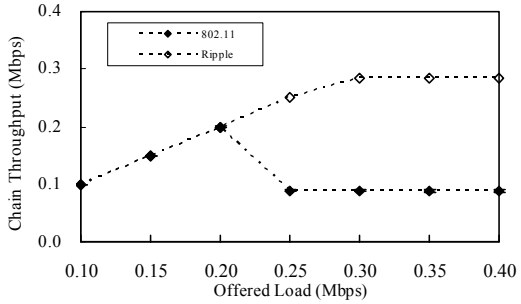
[10] D. Raguin, M. Kubisch, H. Karl, and A. Woltz, "Queue-driven cut-through medium access in wireless ad hoc networks," *Proc. of IEEE WCNC*, pp.1909-1914, 2004.

[11] M. Ergen, D. Lee, R. Sengupta, and P. Varaiya, "WTRP – Wireless token ring protocol," *IEEE Trans. Vehicular Technology*, pp. 1863-1881, vol. 53, no. 6, Nov. 2004.

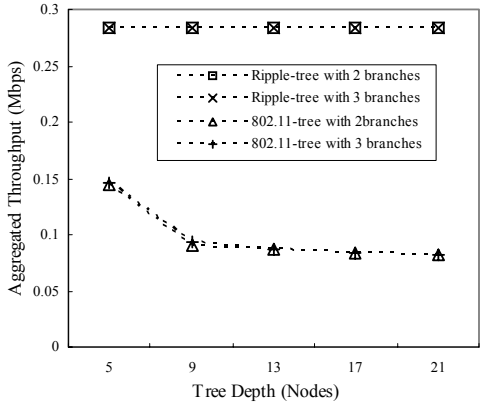
[12] R. G. Cheng, C. Y. Wang, L. H. Liao and J. S. Yang, "*Ripple*: A wireless token-passing protocol for multi-hop wireless mesh networks," *IEEE Communications Letters,* vol. 10, no. 2, Frb. 2006.



(a) Chain throughput for various chain lengths and different DATA frame size



(b) Chain throughput for various offered load



(c) Aggregated throughput for different tree depth and branches

Fig. 5. Performance evaluation