

Supporting a Pseudo-TDMA Access Scheme in Mesh Wireless Networks

Ilenia Tinnirello¹ and Pierluigi Gallo¹

¹ DEIM, Università di Palermo, Italy

`ilenia.tinnirello@unipa.it`, `pierluigi.gallo@unipa.it`

Abstract. Wireless mesh networks appear a promising solution for providing ubiquitous low-cost wireless access, but cannot rely on simple CSMA access protocols because of the critical inefficiencies that arise in topologies with hidden nodes. To overcome these limitations, some important protocol extensions based on synchronization and reservation mechanisms have been ratified.

In this paper we show that an alternative approach to the standardization of new features and signaling messages for mesh networks can be the utilization of programmable nodes able to execute different MAC protocols programmed on the fly. Signaling messages are used only for disseminating the new protocol among the nodes. The scheme, that we call pseudo-TDMA, can be optimized as a function of the node density in the network. Apart from the numerical evaluations, we also run some experiments by exploiting our prototype of wireless programmable node called Wireless MAC Processor.

Keywords: wireless mesh networks, synchronization, pseudo-TDMA, random access

1 Introduction

In recent years, Wireless Mesh Networks (WMNs) based on the IEEE 802.11 technology have gained enormous popularity due to the possibility to provide ubiquitous wireless access to end users with reduced infrastructure costs. However, the original 802.11 standard lacks of several functionalities for effectively managing multi-hop ad-hoc networks, while the legacy DCF channel access protocol has shown significant shortcomings in these network topologies where severe collision rates may arise because of hidden nodes [1].

In order to improve the network transport efficiency and guarantee self-organization, self-configuration, easy installation and maintenance of mesh networks, after an initial proliferation of proprietary incompatible solutions and research proposals, the 802.11s task group has worked on the standardization of new network management functionalities and channel access optimizations. For example, a key component of these enhancements, called MCCA (Multi-user Controlled Channel Access), enables nodes to reserve channel access intervals in advance for avoiding conflicts with contending nodes within a two hops distance.

This new feature requires a synchronization mechanism between neighbor nodes and new signaling messages for the set up of the channel reservations.

Obviously, despite the efforts of the standardization group, several brilliant solutions proposed by the research community have not been included in the new ratified functionalities, especially because some solutions are tailored to work in niche network scenarios. However, because of the heterogeneity of mesh network deployments (in terms of scale, node mobility, traffic types, distance to the gateways, etc.) and available hardware (single or multiple radio interfaces, clock stability, directional or omnidirectional antennas, etc.), these networks represent an interesting case in which the traditional concept of *one-for-all* networking solution exhibits clear limitations and new networking paradigms based on node programmability should be considered.

In this paper, after a brief review of a programmable node architecture, called Wireless MAC Processor (WMP), recently proposed for building customized MAC protocols, we focus on the analysis of MAC protocol extensions devised to work in unsynchronized multi-hop networks without signaling overheads. Thanks to the availability of a WMP prototype, we experimentally validate our approach in a simple multi-hop network topology supporting network-level reprogramming. Simulation results have been considered for evaluating the protocol effectiveness in more complex scenarios.

2 Related Work

2.1 The Wireless MAC Processor

The Wireless MAC Processor architecture allows to abstract the heterogeneous hardware capability of the nodes into a set of actions that can be performed on the hardware (starting a frame transmission, detecting the medium activity, freezing/activating a timer, etc.), a set of events triggered by the hardware, and a set of conditions that can be verified on the state of the hardware internal components. The set of hardware actions, events and conditions represent the node API that cannot be modified by the user.

Generic MAC protocols are executed by the MAC Engine, that is an executor of high-level state machines composed on the basis of the hardware abstractions. Indeed, the definition of the medium access control logic in terms of extended finite state machine (XFSM) permits to conveniently control the hardware. In [2] it is shown that completely different medium access operations (including a TDMA, CSMA, multi-channel schemes, and so on) can be defined by exploiting this simple programming model and an API of about ten actions, ten events and ten conditions.

A MAC program is coded into a table of transitions between logical protocol states that is loaded in a memory space on the hardware. Starting from an initial (default) state, the MAC engine fetches the table entry corresponding to the state, and loops until a triggering event associated to that state occurs. It then evaluates the associated conditions on the configuration registers, and

if this is the case, it triggers the associated action and register status updates (if any), executes the state transition, and fetches the new table entry for such destination state.

Since a MAC program is basically a list of labels specifying the events, actions and conditions associated to each state transition, by defining a common set of labels for the API (i.e. a machine language), the MAC program can be transported over data frames from one node to another. In [3] it has been shown that a basic version of DCF can be coded into 500 bytes only. By adding a simple header for controlling the loading and activation of the new state machine on the card, code mobility can be easily supported [3] in the so called MAClets (in analogy to the JAVA applets).

The MAC engine does not need to know to which MAC program a new fetched state belongs, so that a code switching is achieved by moving to a state in a different transition table and by updating the platform configuration registers (e.g. the operating channel, the transmission power, etc.). The definition of code switching transitions are logically independent of the MAC program definition. Therefore, rather than adding them to the MAC program, the architecture allows to program the switching transitions into a second-level state machine (meta state machine), whose states represent the MAC program under execution.

2.2 MCCA access mechanism

Mesh networks are characterized by local views of the channel sensed by each node, which have a strong impact on the performance of CSMA protocols both in terms of throughput degradation and in terms of fairness. Assuming that transmission and carrier sense ranges coincide, a transmitting station forces its 1-hop neighbors to be in a frozen state, which in turns will increase the channel access probability of the 2-hop neighbors (not hearing the transmission). As widely documented in literature, the overlapping of frame transmissions originated by stations at a 2-hop distance can lead to severe collisions or to the starvation of some traffic flows with a consequent advantage for some others [4].

To mitigate this problem, the recently introduced Mesh Coordinated Channel Access (MCCA) [5] tries to pre-allocate channel holding times to different groups of nodes for avoiding simultaneous transmissions by 2-hop nodes. MCCA reservations are then propagated to 2-hop neighbors for preventing conflicting allocations by hidden nodes. MCCA access rules provides transmission grants in terms of transmission opportunities (i.e. channel holding times), called MC-CAOPs, that are allocated within a multiple of beacon intervals (i.e. a DTIM interval). Each allocation is expressed in terms of: i) *MCCA periodicity*, that is the number of transmission opportunities provided to a given station (with equally space temporal intervals from one opportunity to the next one), ii) *offset* from the starting of the DTIM interval, and iii) *duration* specifying the channel holding time of each transmission opportunity. MCCA allocations are advertised by the transmitter and receiver nodes to their neighbors, which in turn re-broadcast the advertisements to reach the nodes at a distance of 2-hops.

Although the scheme is effective in mitigating hidden node problems and supporting bandwidth reservations in flat topologies, it depends on node synchronization and extra signaling. Commercial wireless cards are equipped with low-quality oscillators, with clock skews ranging from one to one hundred $\mu s/s$. The tradeoff between successful reservations and additional signaling overheads might vary as a function of the node density, source rates and burstiness, and traffic paths. Moreover, as discussed in [6], it still suffers of collisions due to acknowledgement transmissions.

3 Pseudo-TDMA for wireless mesh networks

A critical aspect of MAC protocol schemes for mesh networks is providing synchronization between different nodes. Indeed, network-wide synchronization can be obtained by means of out-of-band signaling employing GPS devices (but this solution has additional costs and does not work in indoor environments) or by the native 802.11 time synchronization function (TSF) which has been shown to have some scalability problems [7].

To avoid to rely on a global synchronization function, alternative coordination mechanisms among the nodes can employ traffic rate limitations, multi-channel solutions [8], token-based channel grants [9], traffic aggregation [10], and network coding [11]. Although these solutions have not been included in 802.11s, they can be easily supported by nodes based on the WMP architecture. To prove such a feasibility and quantify the achievable performance benefits, we focus on a scheme, that we call pseudo-TDMA, that limits the channel access rate at each node. The scheme supports the allocation of different channel holding times to groups of non interfering stations without explicit negotiation among adjacent nodes.

3.1 Pseudo-TDMA access mechanism

Pseudo-TDMA transmissions are performed after a successful random access phase. When a node successfully transmits a packet and receives the relevant acknowledgement (i.e. the intended receiver have not experienced any interfering signal during the whole frame reception process), it assumes to periodically perform subsequent channel accesses for the same traffic flow at regular time intervals. In other words, the first contention acts as a reservation phase after which the channel holding time of the first packet transmission (i.e. the *pseudo-slot*) is considered allocated for the next allocation intervals (i.e. the *pseudo-frames*). In case of successful pseudo-slot allocation, if no other station is trying to reserve a channel holding time, collisions cannot occur. Conversely, when new reservations are performed or reservations are still in progress, pseudo-slots can be affected by collisions. Carrier sense is still used before accessing each pseudo-slot, but no additional backoff is required. When the medium is sensed busy in an allocated pseudo-slot, the transmission is not performed. In such a case or in case of

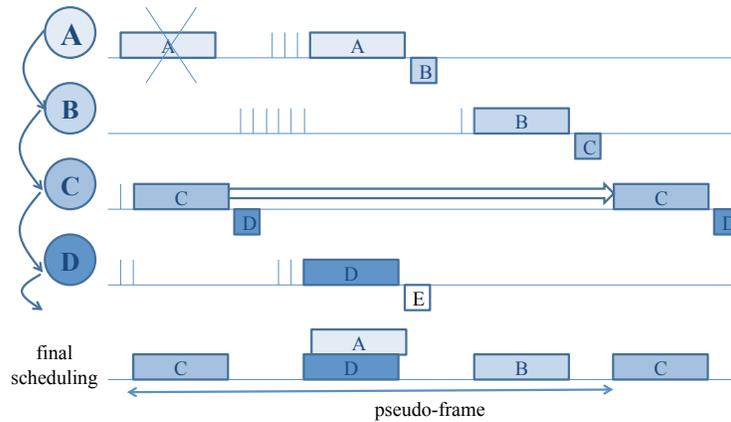


Fig. 1. An example of Pseudo-TDMA access operations in a chain of nodes.

collision, the station keeps the slot or tries a new reservation according to a random probability. Note that pseudo-slot allocations are performed on a per-link basis: a given node transporting two different traffic flows towards two different receivers has to allocate two different pseudo-slots, because the interfering conditions depend on the specific receiver location.

Figure 1 shows an illustrative example of pseudo-TDMA access operations in a network topology given by a chain of nodes. Nodes are labeled from A (first node) to E (last receiver node, not indicated in the figure), while traffic flows are set unidirectionally from two adjacent nodes (from A to B, from B to C, and so on). Each node hears only its neighbors (e.g. B hears only A and C). During the initial random access phase, nodes A and C transmit simultaneously under the assumption that transmission range and carrier sense range coincide. Since node D does not experience any interfering signals, it acknowledges node C transmission. Therefore, node C suspends the random access as indicated by the white arrow, waiting for the next pseudo-slot. As soon as the other nodes perform their first successful transmission, the channel access sequence is repeated periodically according to the final schedule of channel access grants.

As evident from the figure, the scheme basically works by trying to randomly find a successful scheduling of transmissions that can be performed sequentially or in parallel by multiple nodes (C, D+A, B) and repeat such a scheduling over time. Whenever the time interval between successive pseudo-slot allocations (i.e. the pseudo-frame) is large enough to accommodate all the interfering transmissions that cannot be performed simultaneously, after an initial random phase, the frame transmissions occur at regular time intervals without collisions. Note that simultaneous transmissions do not need to be perfectly synchronized (e.g. A and D transmissions), since pseudo-slots are automatically spaced of the time interval required for avoiding interference with 1-hop and 2-hops neighbors.

3.2 Numerical results

Before testing the pseudo-TDMA performance over the WMP, we performed some simulations for evaluating the scheme performance and scalability in general topologies with a large number of network nodes. Simulations have been performed in MATLAB, where we implemented the generation of random topologies, the setting of random traffic flows, and the tracking of channel access operations under standard DCF or pseudo-TDMA access rules. We quantified the per-node throughput results, as well as the channel access fairness, for different pseudo-frame intervals, in order to study the scheme effectiveness in improving starvation while keeping a good channel utilization. Indeed, for very large pseudo-frame intervals, we can easily imagine that all the nodes can successfully access the channel, but with a very poor throughput performance. Conversely, when the pseudo-frame interval is too short, it might happen that the random access phase is never concluded and some stations are prevented from accessing the channel.

The transient behavior of the scheme is shown in figure 2 for a random topology of 30 nodes uniformly distributed over an area of $300\text{ m} \times 300\text{ m}$ with a transmission range (equal to the carrier sense range) set to 100 m . Each node has a greedy traffic source towards a given neighbor node (randomly extracted among the available ones). The figure labels each node with a different identifier and plots the relevant transmission intervals with a different color over a simulation time of 0.15 s . For comparison, the figure also shows the transmission intervals of the nodes under legacy DCF.

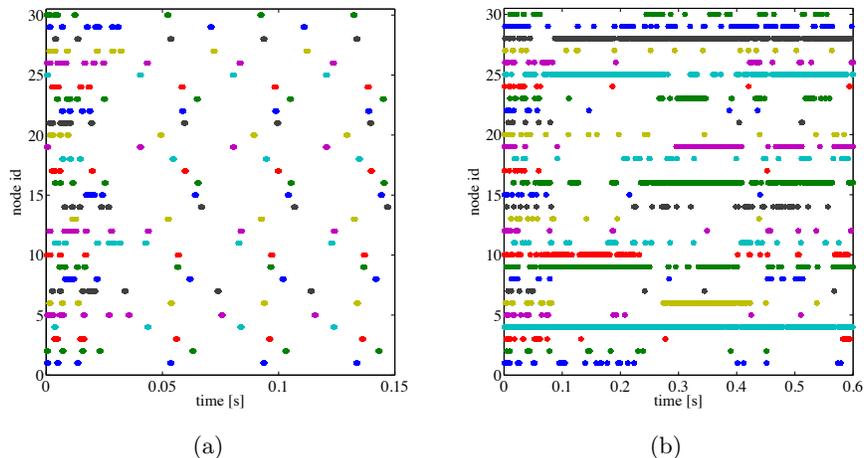


Fig. 2. Channel access intervals under pseudo-TDMA (a) and DCF random access (b).

Simulations have been carried out by considering an 802.11b PHY, with a data rate set to 11 Mbps. Under these settings, for a packet payload of 1000 bytes, the time required for transmitting a frame and the acknowledgement is about 1.3 *ms*. For a fully connected topology with 30 nodes, a perfect TDMA access would require a frame of about 39 *ms*. Obviously, such a time can be lower when parallel transmissions are possible, provided that a perfect synchronization is available at all the nodes and a central scheduler (aware of interference conflicts) notifies the slot allocations to each node. In figure 2, the pseudo-TDMA scheme has been run with a pseudo-frame of 50 *ms*. Although such a time is higher than the time required in the fully connected topology, it is large enough to find a final successful scheduling by the end of the first pseudo-frame. All the stations succeed in accessing the channel, without the starvation effects evident for legacy DCF (e.g. most of the time the channel is used only by stations 28, 25, 16, 10, 9 and 4).

Table 1. Simulation parameters

Common			
# nodes	30		
topology	uniformly distributed nodes over a square		
traffic model	saturated sources		
dst selection	randomly chosen among neighbors		
pkt duration	1 <i>ms</i>		
Pseudo-TDMA		DCF random access	
# of pseudo-TDMA slots per frame	30	CW_{min}	7
		CW_{max}	1023

We run experiments for 20 different topologies with similar characteristics (30 nodes, deployment area of $300\ m \times 300\ m$, transmission range set to $100\ m$) and averaged the aggregated and per-node throughput performance. Figure 3-(a) and (b) plots the total number of successfully transmitted packets and collided packets in the network for pseudo-TDMA with different pseudo-frame intervals (a) and DCF with different contention windows (b) in a simulation run of 1 *s*. For pseudo-TDMA the figure adopts a logarithmic scale, while the pseudo-frame interval is measured as a multiple of the channel holding time required for transmitting and acknowledging a packet (i.e. multiple of pseudo-slots). For a pseudo-frame higher than 30 pseudo-slots the fairness index is about one, while the throughput obviously degrades because of longer intervals in which the channel remains idle. Note also that in our network topology nodes have a limited distance (in hops) and therefore parallel (non-interfering) transmissions are not likely to occur.

If we compare the pseudo-TDMA performance with legacy DCF we can immediately observe that even with short pseudo-frame intervals pseudo-TDMA is more fair. Conversely, DCF can achieve higher aggregated throughput, but

this throughput is shared by a few nodes with several flows suffering starvation. Even reducing the DCF access rate (using higher collision windows) the fairness performance does not improve significantly.

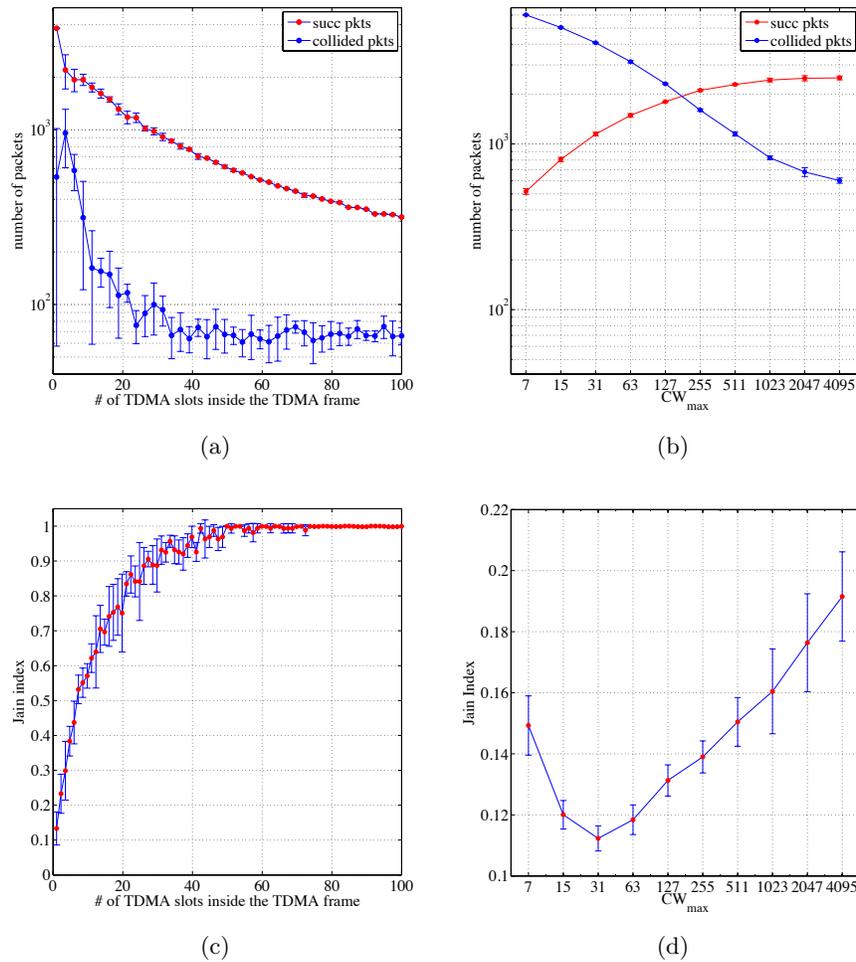


Fig. 3. Comparison between pseudo-TDMA and DCF: successful and collided packets (a), (b) and fairness (Jain) index (c), (d).

Finally, figure 4 compares successful and collided packets in (a), and fairness index in (b) for different graph depths obtained varying the coverage radius of each node range from 80 *m* to 300 *m*. In figure, the number of pseudo-TDMA slots in a pseudo-TDMA frame is computed as the next integer after the average degree of the square of the connectivity graph (G^2). The average degree of G^2

can be proficiently used to dimension the pseudo-TDMA frame because nodes have enough independent resources in terms of pseudo-TDMA slots to avoid collisions with neighbors up to two hops. Under such conditions, figure 4-(a) shows that pseudo-TDMA has an aggregate throughput comparable to DCF when the graph is highly connected. When the radius coverage is smaller, the aggregate number of successfully transmitted packets is lower than the DCF one but the Jain index is notably higher. Pseudo-TDMA always guarantees the best fairness and the lowest collision rates, which also implies energy savings and reduced airtime waste.

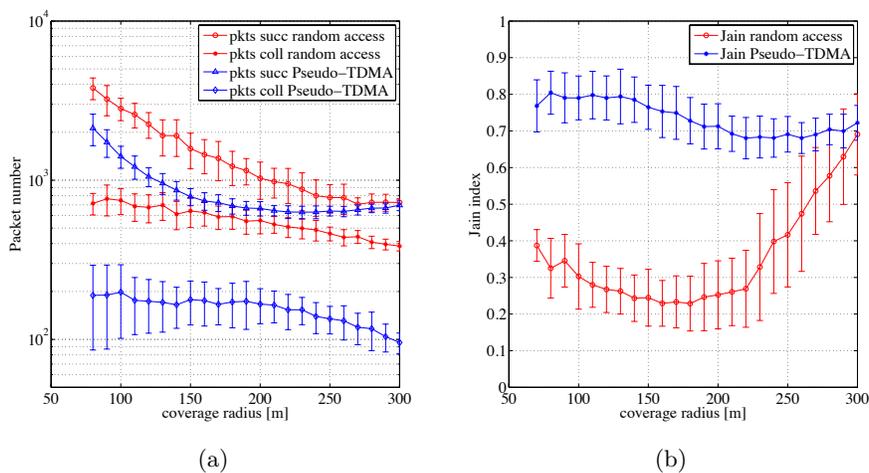


Fig. 4. Performance comparison between pseudo-TDMA (number of slots is the next integer after the average degree of G^2) and random access: pkts successful or collided (a), Jain index (b).

4 Experimental Results

In order to experimentally validate the proposed scheme in network topologies non fully connected, we designed a MAC protocol state machine on the basis of the WMP API. The state machine has been obtained by slightly modifying the one presented in [2], which in turns integrates a few modifications to the basic DCF transmission state machine. Figure 5 shows the resulting state machine. After the first ACK reception, the transition to the TX state can be performed from the WAIT PSEUDO FRAME state at the expiration of the frame interval timer. Since the medium state is verified before performing such a transition, in case of medium busy the machine remains to the WAIT PSEUDO FRAME state or switch to a new random access phase (BACKOFF state) with probability

modulation of node A packets. This phenomenon, that is known in literature as *capture*, is clearly evident from figure 6-(c), where node A throughput is slightly lower than the throughput obtained when node C is off. Note that in case of visible contending nodes this throughput would have been about one half of the throughput obtained with a single contending node (namely, about 6 Mbps for a packet size of 1500 bytes and a data rate of 11Mbps). In case of pseudo-TDMA with a pseudo-frame set to 4ms (able to accommodate the transmission of two packets and acknowledgements in every pseudo-frame interval), after a transient phase, the two stations equally share the available bandwidth when they are simultaneously active, with a throughput equal to $12000\text{bit}/4\text{ms}=3\text{Mbps}$. Note that, as described in [3], the switching from DCF to pseudo-TDMA could be automatically programmed into a meta-state machine in case of high collision rates.

5 Conclusions

In this paper we propose a simple extension of legacy DCF devised to work in mesh networks without requiring node synchronization and reservation messages. The basic idea of the protocol is combining random access and regular scheduling of packet transmissions, in order to repeat the sequence of channel accesses which result successful. Provided that the scheduling interval is large enough, the scheme is able to allocate one packet transmission to each contending node without signaling messages, while preventing flow starvation (as verified via simulation). Despite its simplicity the scheme cannot be supported in legacy 802.11 cards. However, thanks to the availability of a card able to execute generic MAC protocols programmed in terms of state machines (the so called Wireless MAC Processor), we implemented very easily the scheme and run some experiments in a simple network topology. More interesting, whenever the mesh nodes implement the WMP architecture, such a scheme can be dynamically programmed (or reconfigured, e.g. changing the pseudo-frame interval) by simply flooding a data packet transporting the protocol state machine in the whole network [3]. The experimental validation of our approach is performed in a simple multi-hop network topology supporting network-level reprogramming, further experiments will be run in the CREW testbed [17].

Acknowledgments

This work has been supported in part by the EU projects FP7-257263 (FLAVIA) and FP7-258301 (CABIN-CREW).

References

1. Xu, S., and Saadawi, T.: Revealing the problems with 802.11 medium access control protocol in multi-hop wireless ad hoc networks. *Computer Networks* 38, no. 4, pp. 531-548, 2002.

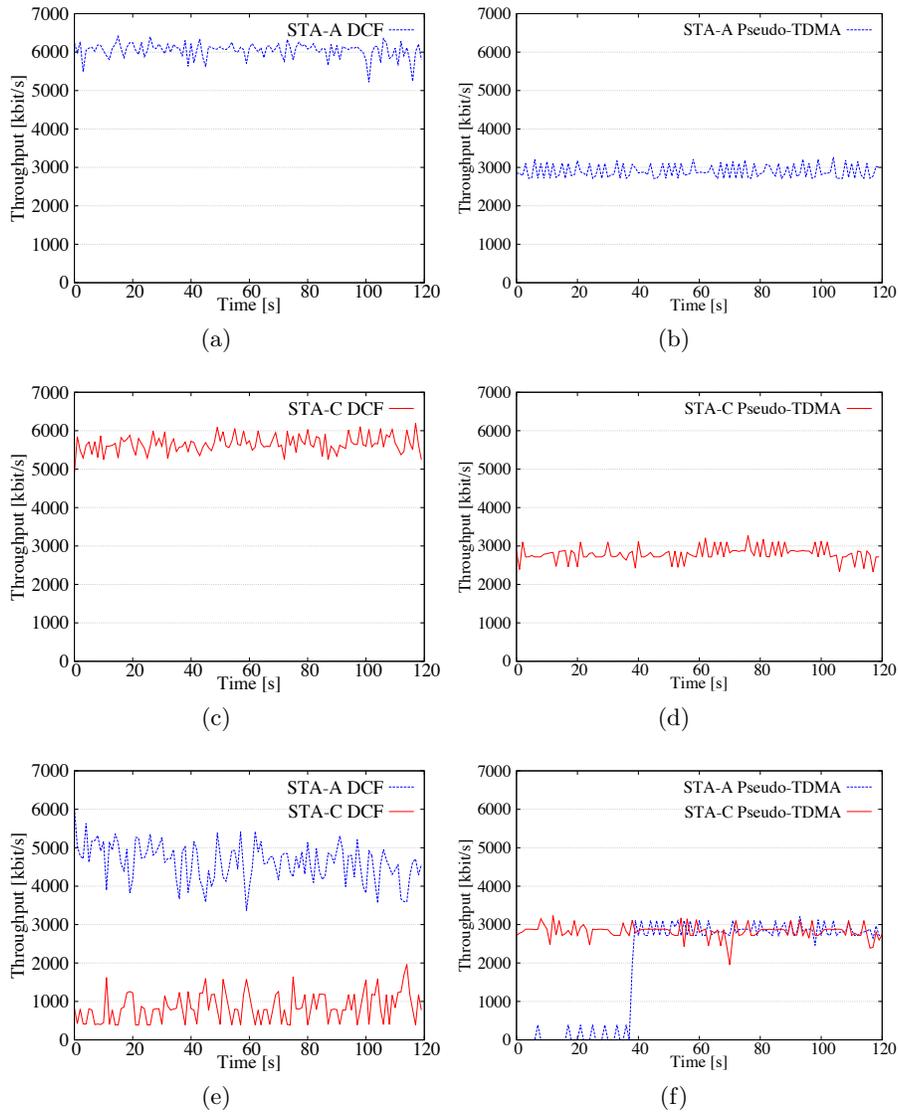


Fig. 6. Throughput performance of a chain of nodes when the node at the edge of the chain transmit to the middle node under DCF (a),(b),(c) and pseudo-TDMA (d),(e),(f).

2. Tinnirello, I., Bianchi, G., Gallo, P., Garlisi, D., Giuliano, F. and Gringoli, F.: Wireless mac processors: Programming mac protocols on commodity hardware. In INFOCOM, 2012 Proceedings IEEE, pp. 1269-1277. IEEE, 2012.
3. Bianchi, G., Gallo, P., Garlisi, D., Giuliano, F., Gringoli, F., and Tinnirello, I.: MAClets: active MAC protocols over hard-coded devices. In Proceedings of the 8th international conference on Emerging networking experiments and technologies,

- pp. 229-240. ACM, 2012.
4. Lyakhov, A., Pustogarov, I., Safonov, A., and Yakimov, M.: Starvation effect study in IEEE 802.11 mesh networks. In Proc. of Third IEEE International Workshop on Enabling Technologies and Standards for Wireless Mesh Networking (MeshTech'09), Macao SAR, P.R. China, 2009
 5. IEEE Standard for Information technology *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. 6 February 2012
 6. Krasilov, A., Lyakhov, A., and Safonov, A.: Interference, even with MCCA channel access method in IEEE 802.11s mesh networks. In Mobile Adhoc and Sensor Systems (MASS), 2011, pp. 752-757. IEEE, 2011.
 7. Huang, L., and Lai, T.H.: On the scalability of IEEE 802.11 ad hoc networks. In Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing, pp. 173-182. ACM, 2002.
 8. Shi, J., Salonidis, T., and Knightly, E.W.: Starvation mitigation through multi-channel coordination in CSMA multi-hop wireless networks. In Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, pp. 214-225. ACM, 2006.
 9. Tinnirello, I., Scalia, L., and Campocchia, F.: Improving IEEE 802.11 performance in chain topologies through distributed polling and network coding. In IEEE ICC'09, pp. 1-6. IEEE, 2009.
 10. Tao, Z., Teo, K.H., and Zhang, J.: Aggregation and concatenation in IEEE 802.16j mobile multihop relay (MMR) networks. In Mobile WiMAX Symposium, 2007. IEEE, pp. 85-90. IEEE, 2007.
 11. Katti, S., Rahul, H., Hu, W., Katabi, D., Mdard, M. and Crowcroft, J.: XORs in the air: practical wireless network coding. In ACM SIGCOMM Computer Communication Review, vol. 36, no. 4, pp. 243-254. ACM, 2006.
 12. Xu, S., and Saadawi, T.: Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?. Communications Magazine, IEEE 39, no. 6, pp 130-137, 2001.
 13. Akyildiz, I. F., Wang, W., and Wang, W.: Wireless mesh networks: a survey. Computer networks 47, no. 4 (2005): 445-487.
 14. He, Y., Yuan, R., Sun, J., and Gong, W.: Semi-Random Backoff: Towards resource reservation for channel access in wireless LANs. In 17th ICNP 2009 pp. 21-30. IEEE, 2009.
 15. Koutsonikolas, D., Salonidis, T., Lundgren, H., LeGuyadec, P., Hu, Y.C., and Sheriff, I.: TDM MAC protocol design and implementation for wireless mesh networks. In Proceedings of the 2008 ACM CoNEXT Conference, p. 28. ACM, 2008.
 16. Djukic, P., and Mohapatra, P.: Soft-TDMAC: A software TDMA-based MAC over commodity 802.11 hardware. In INFOCOM 2009, pp. 1836-1844. IEEE, 2009.
 17. CREW, Cognitive Radio Experimentation World, <http://www.crew-project.eu/>