Computer Communications 25 (2002) 1058–1065

computer communications

www.elsevier.com/locate/comcom

TRAP: a high performance protocol for wireless local area networks

P. Nicopolitidis[a], G.I. Papadimitriou[a], M.S. Obaidat[b,*], A.S. Pomportsis[a]

[a]Department of Informatics, Aristotle University, Box 888, 54006 Thessaloniki, Greece
[b]Department of Computer Science, Monmouth University, West Long Branch, NJ 07764, USA

Received 1 August 2001; revised 30 October 2001; accepted 28 November 2001

Abstract

The fixed number of available random addresses for randomly addressed polling (RAP) poses a significant problem in terms of scalability in case of many active stations. In such cases, the protocol's performance is significantly degraded. In this paper, we propose a TDMA-based randomly addressed polling (TRAP) protocol. The protocol employs a variable-length TDMA-based contention stage with the length based on the number of active stations. At the beginning of each polling cycle, the base station invites all active mobile stations to register their intention to transmit via transmission of a short pulse. The base station uses the aggregate received pulse in order to obtain an estimate of the number of contending stations and schedules the contention stage to comprise an adequate number of time slots for these stations to successfully register their intention to transmit. Then it transmits a READY message carrying the number of time slots P. Each mobile station calculates a random address in the interval $[0 \dots P-1]$, transmits its registration request in the respective time slot and then the base station polls according to the received random addresses. Simulation results are presented that reveal the superiority of TRAP against the RAP protocol in case of medium and high offered loads. Furthermore, the implementation of the proposed protocol is much simpler than that of CDMA-based versions of RAP, since no extra hardware is needed for the orthogonal reception of the random addresses.

Keywords: Randomly addressed polling; Wireless LANs; Medium access control protocol; Performance evaluation; TDMA

1. Introduction

There are fundamental differences between wireless and wired LANs that pose difficulties in the design of medium access control (MAC) protocols for wireless LANs (WLANs) [1–4]. Wireless LANs, as the name suggests, utilize wireless transmission for information exchange. The wireless medium is characterized by bit error rates (BER) having an order of magnitude even up to 10 times the order of magnitude of a LAN cable's BER. The primary reason for the increased BER is atmospheric noise [5], physical obstructions found in the signal's path, multipath propagation, interference from other systems and terminal mobility. Furthermore, in wireless LANs errors occur in bursts, whereas in traditional wired systems errors appear randomly. Finally, a fully connected topology between the nodes of a wireless LAN cannot be assumed. Rather, the logical topology of a wireless LAN tends to change as users move from one position to another. As a result, wireless LANs are characterized by unreliable links between nodes resulting in bursts of errors and dynamically changing network topologies.

MAC protocols can be roughly divided into three categories: Fixed assignment (e.g. TDMA, FDMA), random access (e.g. ALOHA, CSMA/CD, CSMA/CA) and demand assignment protocols (e.g. token passing, polling). Fixed assignment protocols exhibit high performance when the traffic of each station is stable and the network topology remains unchanged. However, they fail to adapt to changes in network topologies and traffic and thus exhibit low performance when used over the wireless medium or under bursty traffic conditions. Random access protocols, on the other hand, operate efficiently both without topology knowledge and under changing traffic characteristics. Nevertheless, their disadvantage is their non-deterministic behavior, a fact that causes problems in supporting QoS guarantees. Demand assignment protocols try to combine the advantages of fixed and random access protocols. However, the token-based approach is generally thought to be inefficient. This is due to the fact that in a wireless LAN token losses are much more likely to appear due to the increased BER of the wireless medium. Furthermore, in a token passing network, the token holder needs accurate information about its neighbors and thus of the network topology. Polling, on the other hand, is a more appealing MAC option for a wireless LAN since it offers centralized

* Corresponding author. Tel.: +1-732-571-4482; fax: +1-732-263-5202.
E-mail address: obaidat@monmouth.edu (M.S. Obaidat).

0140-3664/02/$ - see front matter
PII: S0140-3664(02)00020-8

supervision of the network nodes. However, constant monitoring of all nodes is required, which is not feasible in the harsh fading environment of a wireless LAN.

An effort to alleviate the above-mentioned problem of polling protocols in a wireless environment is made by the randomly addressed polling (RAP) protocol [6–9]. RAP is a polling protocol designed to work, not with all the nodes contained in a cell, but only with the active ones seeking uplink communication. The RAP protocol assumes an infrastructure cellular topology. Within each cell, multiple mobile nodes exist that, when active, compete for access to the wireless medium. RAP employs a contention scheme using a fixed number of random numbers, known as random addresses, that are used to resolve contention among mobile stations.

However, the fixed number of random addresses poses a significant problem for RAP in cases of many contending stations and in such cases the protocol's performance is significantly degraded. In this paper, we propose a mechanism that estimates the number of active stations at the beginning of each polling cycle. According to this proposal, at the beginning of each polling cycle, all active mobile stations register their intention to transmit via transmission of a short pulse. All active stations' pulses are added at the base station which uses the aggregate received pulse to estimate the number of active stations. Based on this estimate, the base station then schedules a TDMA-based contention stage to comprise an adequate number of slots for the active stations to successfully register their intention to transmit and then commences polling.

The rest of this paper is organized as follows. In Section 2, the operation of the RAP protocol is presented. Section 3 introduces TDMA-based randomly addressed polling (TRAP) protocol, our proposal for a variable-length TRAP with a contention stage with a length based on the number of active stations. Section 4 details the simulation environment used to compare the relative performance of TRAP to that of RAP and presents simulation results that reveal TRAPs superiority in cases of medium and high loads. Finally, concluding remarks are presented in Section 5.

2. The RAP protocol

RAP is a protocol designed for infrastructure WLANs. In RAP, the cell's base station initiates a contention period in order for active stations to inform their intention to transmit packets. For each polling cycle, contention is resolved by assigning addresses only to the active stations within the cell at the beginning of the cycle. To this end, all active mobile stations generate a random number and transmit it simultaneously to the base station using a form of orthogonal transmission, such as CDMA or FDMA. The number transmitted by each station identifies this station during the current cycle and is known as the station's random address. For a RAP WLAN consisting of M active mobile stations under the coverage of a base station, the stages of the protocol, are outlined later.

- *Contention invitation stage*: Whenever the base station is ready to collect packets from the mobile stations, it transmits a *READY* message, which may be piggybacked in a previous downlink transmission.
- *Contention stage*: Each active mobile station generates a random number R, ranging from 0 to $P-1$. All active stations transmit their random numbers simultaneously to the base station using a form of orthogonal transmission, such as CDMA or FDMA. The number transmitted by each station identifies this station during the current cycle and is known as its random address. To combat the medium's fading characteristics, a station may transmit its generated random address up to q times in a single contention stage. When an error-free transmission is assumed, $q=1$ suffices. Optionally, the contention stage may be repeated L times. Each time, each active station generates and transmits a (possibly different) random address, as described earlier.
- *Polling stage*: Suppose that at the lth stage ($1 \leq l \leq L$) the base station receives the largest number of distinct addresses and these are, in ascending order, $R_1, R_2, \ldots, R_n$. The base station polls the mobile stations using those numbers. When the base station polls mobile stations with R_k, stations that transmitted R_k as their random address at the lth stage transmit packets to the base station. Obviously, if two or more stations have transmitted the same random address at the lth stage, a collision would occur. If $n=M$, however, no collision occurs.
- If the base station successfully receives a packet from a mobile station, it sends a positive acknowledgment (ACK). ACK packets are transmitted right before polling the next mobile station. If a mobile station receives an ACK, it assumes correct delivery of its packet, otherwise, it waits for the current polling cycle to complete and retries during the next cycle.

RAP uses a fixed number of random addresses P with values of P around 5 suggested [6]. This limitation stems from the requirement for orthogonal transmission of the random addresses. If CDMA or FDMA is used for transmission, the maximum number of available random addresses is limited, due to either receiver complexity or spectrum shortage, respectively.

A station is said to be active, if it has a packet to transmit. The operation of RAP constitutes a number of polling cycles. When a collision between two or more stations occurs, these stations keep the collided packets and compete for access to the medium in the next polling cycle. Newly active stations are usually not allowed to compete with those having collided packets [7]. The collision resolution cycle (CRC) is defined as the period of time that elapses in order for all the active stations at the beginning of the CRC to

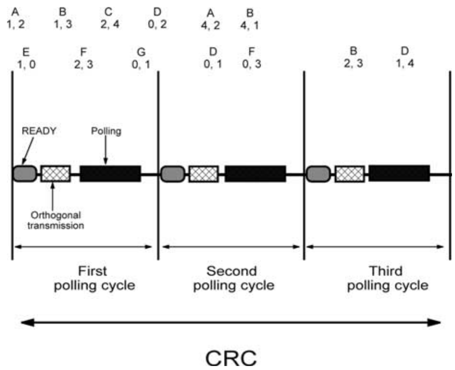

Fig. 1. A RAP CRC finished in three polling cycles.

transmit their packets. In order to keep newly active stations from entering the competition, the READY message at the beginning of a CRC can have a different form from that at the beginning of a polling cycle commencing inside a CRC. However, the prohibition of newly active stations to compete with those having collided packets is not compulsory [6].

To better understand the RAP protocol, we present an example. Fig. 1 shows an example with $M = 7$ active stations and $P = 5$ available random addresses. We assume that $L = 2$, thus at the beginning of the CRC, all seven stations transmit two random addresses to the base station. As we can see, the maximum number of distinct random addresses is received at the second stage, thus the base station polls according to the received numbers at this stage. Stations C, E and G manage to transmit their packets without a collision, while A, D, B and F proceed to the next polling cycle. At this cycle, the base station polls according to the numbers of the second stage and thus (assuming that no newly active stations are allowed to join repolling) A and F manage to transmit their packet, while B and D collide. During the third polling cycle, B and D transmit their packets. After the completion of the CRC, another one begins and all active stations join the new CRC.

3. The TDMA-based RAP protocol

The RAP protocol has been analyzed in Ref. [7]. For a small number M of active stations, compared to the number of random addresses P, the protocol performs well. However, the fixed number of used random addresses makes RAP protocol inefficient in cases of many active stations. This is because a small number of available random addresses provides very little space for the contention to be resolved, as for values of N with $P \leq M$, the selection of the same random address by more than one station becomes very likely. As a result, the probability of a successful transmission is lowered, which leads to the decreased throughput and increased delay.

In order to combat this phenomenon, we propose TRAP, which employs a variable-length TDMA-based contention stage, which lifts the requirement for a fixed number of random addresses. The TDMA-based contention stage comprises a variable number of equally-sized slots, with each slot corresponding to a random address. However, a mechanism for estimating the number of active stations is needed in order for the base station to select the appropriate number of slots (equivalently, random addresses) in the TDMA contention stage. To this end, at the beginning of each polling cycle, all active mobile stations register their intention to transmit via transmission of a short pulse. All active stations' pulses add at the base station, which uses the aggregate received pulse to estimate the number of active stations. The time slots will obviously be of fixed length, thus a mobile station that generates a random address R, $0 \leq R < P$, will transmit its random address at slot R. Based on this approach, the proposed protocol works as follows:

- *Active stations estimation*: At the at the beginning of each polling cycle, the base station sends an *ESTIMATE* message in order to receive active stations' pulses. After the base station estimates the number of active stations M based on the aggregate received pulse, it schedules the TDMA-based contention stage to comprise an adequate number of random addresses $P = kM$, where k is a positive integer, for the active stations to compete for medium access with few collisions.
- *Contention invitation stage*: The base station announces it is ready to collect packets from the mobile stations by transmitting a READY message, containing the number of random addresses P to be used in this polling cycle.
- *Contention stage*: Each active mobile station generates a random number R, ranging from 0 to $P - 1$. Active stations transmit their random numbers at the appropriate slot of the TDMA-based contention stage. As in RAP, stations can generate addresses up to q times in a single contention stage and the contention stage may be repeated L times, with each active station generating a random address for each stage. Obviously, if two or more mobiles select the same random address, their random address transmissions collide and are not received at the base station. Thus, the random addresses received correctly at the base station are always distinct, with each number identifying a single active station.
- *Polling stage*: Suppose that at the lth stage ($1 \leq l \leq L$) the base station received the largest number of random addresses and these are, in ascending order, $R_1, R_2, \ldots, R_n$. The base station polls the mobile stations using those numbers. When the base station polls mobile stations with R_k, the station that transmitted R_k as its random address at the lth stage transmits a data packet to the base station.
- If the base station successfully receives a packet from a mobile station, it sends a positive ACK. ACK packets are transmitted right before polling the next mobile station. If a mobile station receives an ACK, it assumes correct delivery of its packet, otherwise, it waits for the current polling cycle to complete and retries during the next cycle.

Under the assumption of all mobile random address transmissions reaching the base station, the protocol is collision free among data packets. This is because the transmission of the same random address by two or more stations occurs in the same time slot resulting in a collision of the control packets and the address not being polled. Thus the data packets do not collide. This is an advantage of TRAP against the original RAP protocol. Due to the CDMA nature of the contention stage of RAP, when the base station polls a random address that was selected by more than one mobile, the corresponding mobiles' packets will collide. This feature helps preserve bandwidth, since data packets are usually much larger than control packets. Also, it has found use in other WLAN MAC protocols as well, such as IEEE 802.11 and MACAW [10].

However, the obvious advantage of our proposed protocol is in terms of scalability. Since the number of random addresses can now vary according to the number of active stations, the protocol will not degrade in cases of a large number of competing stations. Simulation results that are presented in Section 4.2 reveal that the heuristic estimator $P = kM$ for the number of random addresses is sufficient, since the performance of the protocol at medium and high loads is significantly better than that of RAP. Furthermore, the implementation of TRAP protocol is much simpler than that of CDMA-based versions of RAP, since no extra hardware is needed for the orthogonal reception of the random addresses.

4. Performance evaluation

4.1. Simulation environment

In order to compare the performance of TRAP against RAP, we used a discrete event simulator coded in C. The simulator models N mobile stations, the base station and the wireless links as separate entities. Each mobile station uses a buffer to store the arriving packets. The buffer length is assumed to be equal to Q packets. Any packets arriving to find the buffer full are dropped. Regarding the aggregate network offered load, the simulator models packet arrivals at the mobile stations with packet inter-arrival times being exponentially distributed. The arrival rate is the same among all mobile stations. Each simulation run is carried out until R packet transmissions successfully take place.

As far as modeling of the wireless environment is concerned, the condition of the wireless link between any two stations was modeled using a finite state machine with two states. Such structures can efficiently approximate the bursty-error behavior of a wireless channel [5] and are widely used in WLANs modeling [3,4]. The channel model comprises two states:

- State G, denotes that the wireless link is in a relatively 'clean' condition and is characterized by a small BER, which is given by the parameter *GOOD_BER*.
- State B, denotes that the wireless link is in a condition characterized by increased *BER*, which is given by the parameter *BAD_BER*.

We assume that the background noise is the same for all stations and thus the principle of reciprocity stands for the condition of any wireless link. Therefore, for any two stations A and B, the BER of the link from A to B and the BER of the link from B to A are the same. The time spent by a link in states G and B are exponentially distributed, but with different average values, given by the parameters *TIME_GOOD* and *TIME_BAD*, respectively. The status of a link probabilistically changes between the two states. When a link is in state G and its status is about to change, the link transits to stage B. When a link is in state B and its status is about to change, the link transits to stage G. By changing the model's parameter values, the protocols can be simulated for a variety of physical environments.

In the process of delivering our simulation results, we made the following assumptions:

1. No data traffic is exchanged between the base station and the mobiles. Upon polled, a mobile station can initiate a data packet transmission with any other mobile as its destination. These assumptions limit the role of the base station to be only the means of executing the polling algorithms. They were made to measure the performance increase of TRAP over RAP due to the proposed contention stage. Furthermore, the ability for a mobile station to transmit to any other mobile station within its cell is clearly a more realistic assumption, especially in the absence of backbone traffic to/from the WLAN.
2. We did not include the effect of adding a physical layer preamble in our simulations.
3. No error correction is used and we did not account for the possibility of packet capturing. Whenever two packets collide, they are assumed lost.
4. In the original RAP protocol newly active stations are prohibited from joining repolling, as this is suggested in most of Refs. [7–9]. This means that the end of a CRC for RAP is accompanied by the overhead of a *READY* message followed by an orthogonal address transmission period. On the other hand, since we expect the dynamic nature of the number of random addresses of TRAP to provide enough space for efficient transmission of random addresses, we simulated TRAP to allow newly active stations to join repolling.
5. Inside a RAP CRC, upon dropping of a packet by an active station having two or more packets in its buffer, the station continues to compete in the next polling cycle of the CRC with the next packet in its buffer.

We employed the following broadly used performance metrics in order to compare the protocols:

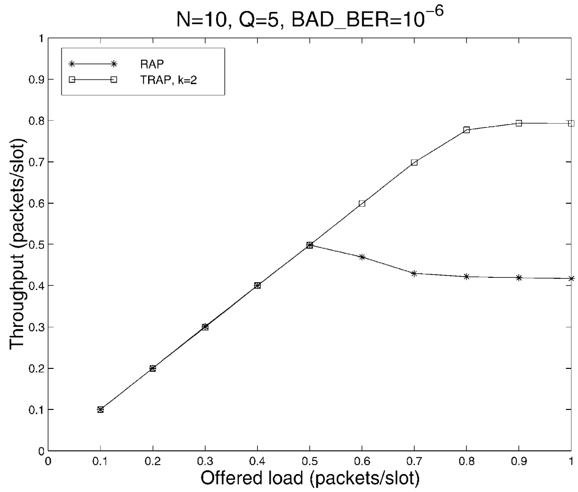

Fig. 2. The throughput versus offered load characteristics of RAP and TRAP when applied to network N_1.

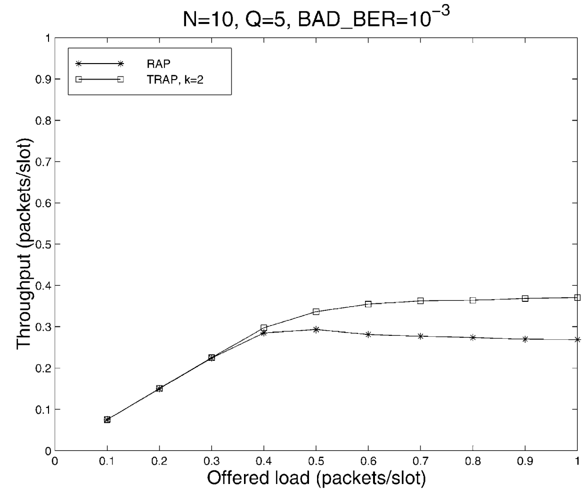

Fig. 4. The throughput versus offered load characteristics of RAP and TRAP when applied to network N_2.

1. The throughput versus offered load characteristic.
2. The delay versus throughput characteristic.

The number of mobile stations N under the coverage of the base station, the buffer size Q and the parameter *BAD_BER* were taken as follows:

1. Network N_1: $N = 10$, $Q = 5$, $BAD\_BER = 10^{-6}$;
2. Network N_2: $N = 10$, $Q = 5$, $BAD\_BER = 10^{-3}$;
3. Network N_3: $N = 50$, $Q = 5$, $BAD\_BER = 10^{-6}$;
4. Network N_4: $N = 50$, $Q = 5$, $BAD\_BER = 10^{-3}$.

All other parameters remain constant for all simulation results and are shown later: $R = 1\,000\,000$, $GOOD\_BER = 10^{-10}$, $TIME\_GOOD = 30$ s, $TIME\_BAD = 10$ s, $L = 2$, $P_{RAP} = 5$, $k = 2$, $RETRY\_LIMIT = 3$.

The variable *RETRY_LIMIT* sets the maximum number of retransmission attempts per data packet. If the number of retransmissions of a data packet exceeds this value (due to either collisions or channel errors) the packet is dropped.

At the MAC layer, the size of all control packets for the protocols is set to 160 bits, the data packet size is set to 6400 bits and the overhead for the orthogonal transmission of the random addresses in RAP is set to five times the size of the poll packet, as in Ref. [7]. The wireless medium bit rate was set to 1 Mbps. The propagation delay between any two stations was set to 0.05 ms.

4.2. Simulation results

The throughput versus offered load characteristics of the compared protocols when applied to networks N_1, N_2, N_3 and N_4 are shown in Figs. 2, 4, 6 and 8, respectively, while the delay versus throughput characteristics when applied to networks N_1, N_2, N_3 and N_4 are shown in

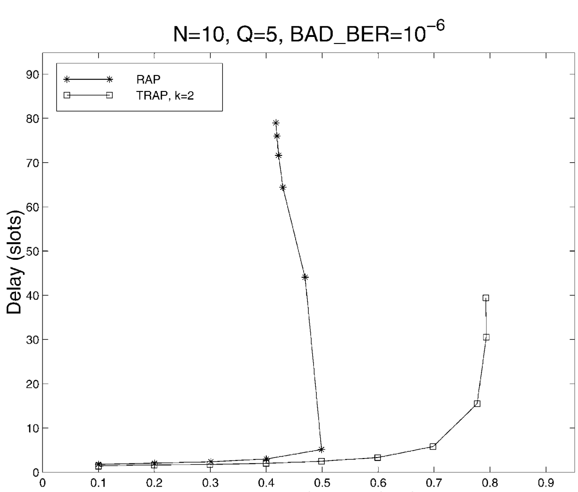

Fig. 3. The delay versus throughput characteristics of RAP and TRAP when applied to network N_1.

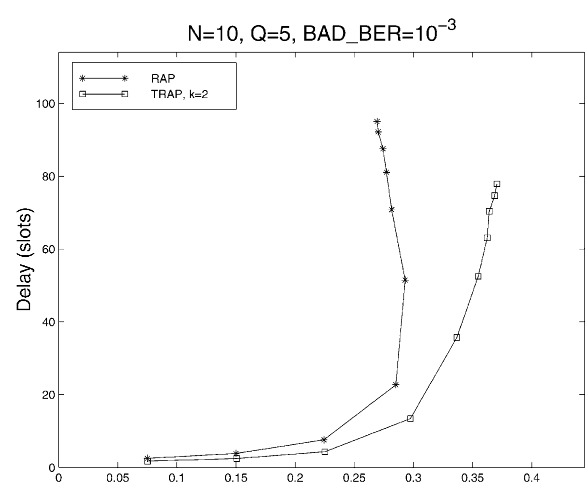

Fig. 5. The delay versus throughput characteristics of RAP and TRAP when applied to network N_2.

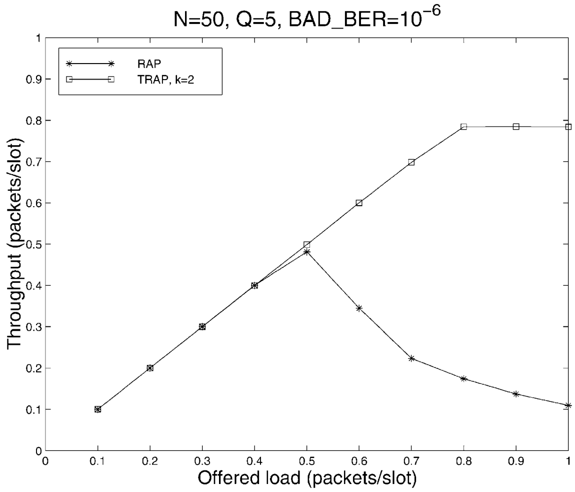

Fig. 6. The throughput versus offered load characteristics of RAP and TRAP when applied to network N_3.

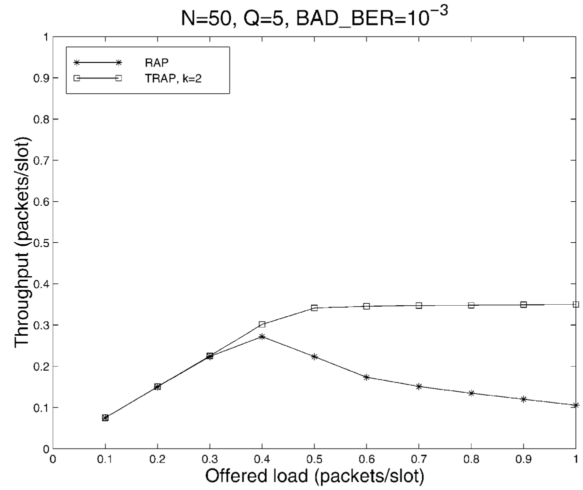

Fig. 8. The throughput versus offered load characteristics of RAP and TRAP when applied to network N_4.

Figs. 3, 5, 7 and 9, respectively. In these figures, "slot" corresponds to the transmission time of a data packet. From these graphs, it is obvious that the performance TRAP is superior to that of RAP in cases of medium and high load conditions. From the figures, we observe that for a WLAN of $N = 10$ mobile stations:

- Under relatively 'clean' wireless links ($BAD\_BER = 10^{-6}$), TRAP reaches a throughput gain over RAP, ranging from about 26% at medium loads (0.6 packets/slot) to about 90% at high loads (1 packet/slot) (Fig. 2).
- Under error-prone wireless links ($BAD\_BER = 10^{-3}$), TRAP reaches a throughput gain over RAP, ranging from about 26% at medium loads (0.6 packets/slot) to about 37% at high loads (1 packet/slot) (Fig. 4).

This superiority at high loads becomes clearer when the number of stations is high ($N = 50$). In that case:

- Under relatively 'clean' wireless links ($BAD\_BER = 10^{-6}$), TRAP reaches a throughput gain over RAP, ranging from about 73% at medium loads (0.6 packets/slot) to about 600% at high loads (1 packet/slot) (Fig. 6).
- Under error-prone wireless links ($BAD\_BER = 10^{-3}$), TRAP reaches a throughput gain over RAP, ranging from about 100% at medium loads (0.6 packets/slot) to about 250% at high loads (1 packet/slot) (Fig. 8).

The throughput of RAP for high loads in the case of $N = 50$ is less than that for the same loads for $N = 10$. This fact can be explained in terms of the number of active stations at the beginning of a CRC. Since in RAP the number of random addresses P is fixed, the protocol degrades in situations with $N > P$ due to the increased number of collisions.

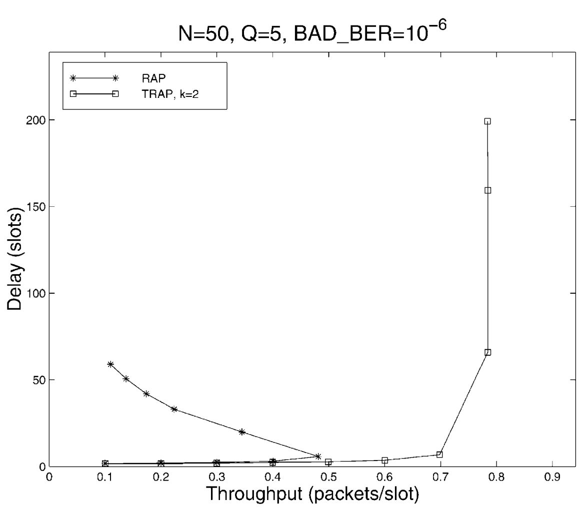

Fig. 7. The delay versus throughput characteristics of RAP and TRAP when applied to network N_3.

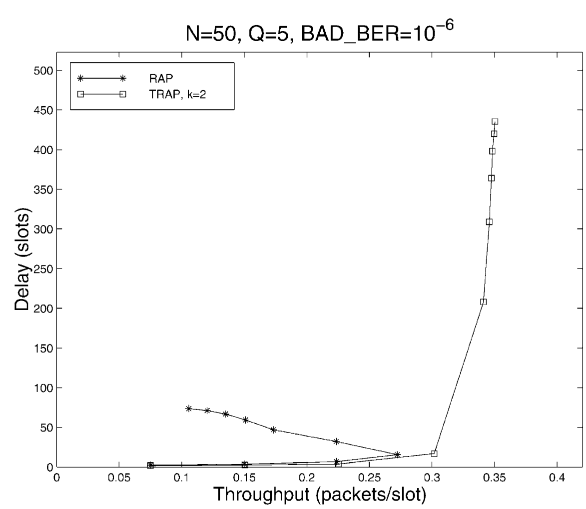

Fig. 9. The delay versus throughput characteristics of RAP and TRAP when applied to network N_4.

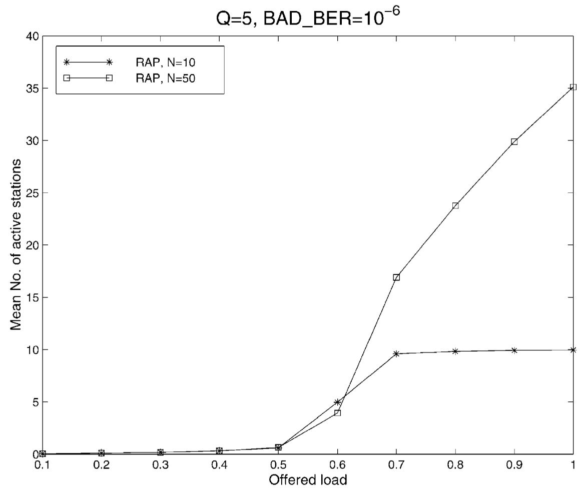

Fig. 10. Number of active stations at the beginning of each CRC for RAP for $N = 10$, $BAD\_BER = 10^{-6}$ (network N_1) and $N = 50$, $BAD\_BER = 10^{-6}$ (network N_3).

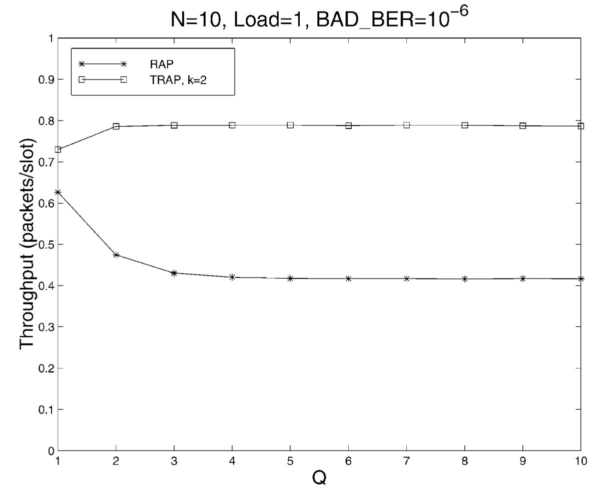

Fig. 12. Throughput of RAP and TRAP when the offered load is 1 packet/slot, for a buffer size $Q \in [1...10]$. $N = 10$, $BAD\_BER = 10^{-6}$.

To better understand this phenomenon Figs. 10 and 11 plot, for networks N_1 and N_3, the mean number of active stations at the beginning of a CRC versus the offered load for RAP and TRAP, respectively. From Fig. 10, we observe that the number of active stations at the beginning of a CRC for RAP is larger for $N = 50$ than for $N = 10$ for the same high load values. Thus, more stations contend when $N = 50$ than when $N = 10$ and since $P = 5$ more collisions occur for $N = 50$, resulting to a smaller throughput of RAP at high loads for $N = 50$ than for $N = 10$. TRAP, on the other hand, is able to adjust the number of random addresses dynamically, according to the number of active stations. Thus, TRAP exhibits a high throughput, even in situations with many active stations (more than $P = 5$), which occur in cases of high loads (Fig. 11, for offered loads more than 0.8 packets/slot, especially in the case of $N = 50$). The same kind of reasoning can explain the fact that for high loads, the throughput of RAP for network N_4 is less than that for network N_2, while TRAP still achieves a better performance than RAP in both cases.

Figs. 2–9, reveal that the achieved superiority of TRAP at medium and high loads comes at no expense over its performance at low loads, since both the throughput versus offered load and delay versus throughput characteristics of both the protocols are practically the same for network loads ranging from 0 to 0.5 packets/slot.

Figs. 12 and 13 present simulation results that show the superiority of TRAP against RAP in terms of throughput for an offered load of 1 packet/slot, for various values of $Q \in [1...10]$. It can be seen that the performance improvement of TRAP over RAP is bigger for a buffer size $Q > 1$. This is to be expected, since $Q > 1$, the total load offered to the network rises due to backlogged packets, leading to

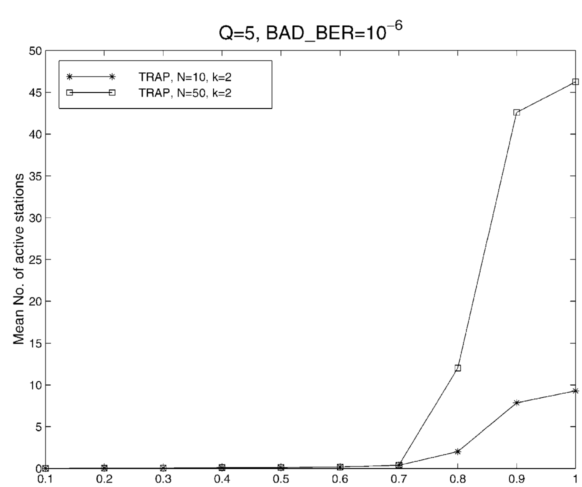

Fig. 11. Number of active stations at the beginning of each CRC for TRAP for $N = 10$, $BAD\_BER = 10^{-6}$ (network N_1) and $N = 50$, $BAD\_BER = 10^{-6}$ (network N_3).

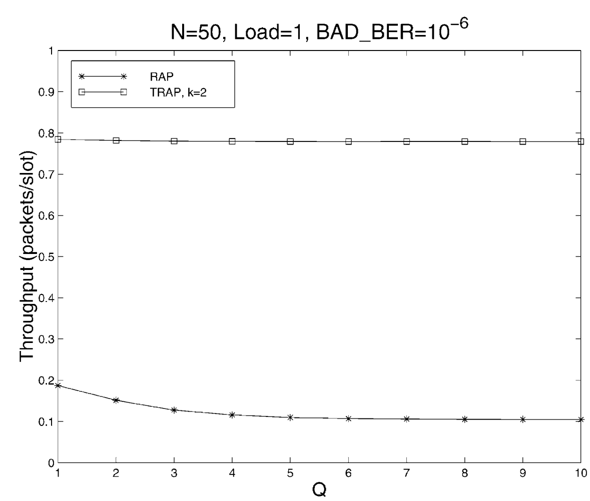

Fig. 13. Throughput of RAP and TRAP when the offered load is 1 packet/slot, for a buffer size $Q \in [1...10]$. $N = 50$, $BAD\_BER = 10^{-6}$.

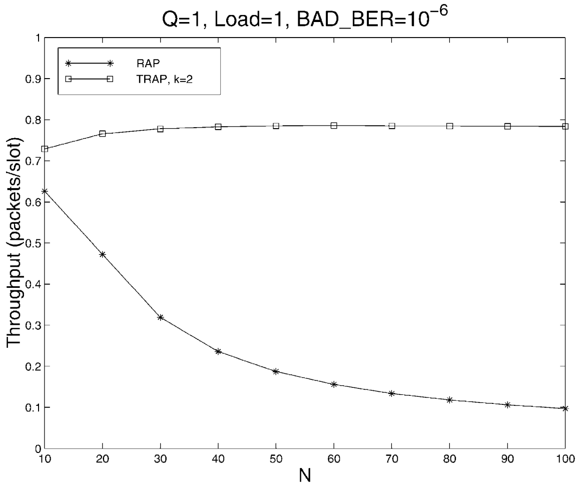

Fig. 14. Throughput of RAP and TRAP when the offered load is 1 packet/slot, for number of stations $N \in [10\ldots100]$. $Q = 1$, $BAD\_BER = 10^{-6}$.

increased contention and thus collisions for RAP. TRAP, on the other hand, manages to efficiently transmit the total offered load by dynamically adjusting the number of random addresses, thus leading to high throughput values.

Finally, Fig. 14 presents simulation results that reveal the superiority of TRAP against RAP in terms of throughput for an offered load of 1 packet/slot, for various values of $N \in [10\ldots100]$. Based on the results of Figs. 12 and 13, we set $Q = 1$ in this experiment, in order to measure the lower limit in the performance improvement of TRAP over RAP. It can be seen that the higher the number of stations N, the higher the performance gain for TRAP. This performance superiority of TRAP over RAP will obviously increase for $Q > 1$, as the total load offered to the network will be higher due to the existence of backlogged packets, a fact that leads to performance degradation for RAP. For $Q = 5$, this can be seen for $N = 10$ and 50 in Figs. 2 and 6, respectively. For an offered load of 1 packet/slot, the achieved throughput for RAP for $Q = 5$ (Figs. 2 and 6) is lower than for $Q = 1$ (Fig. 14), while that of TRAP rises from about 70% for $Q = 1, N = 10$ to reach 80% for $Q = 5$ for the various values of N.

5. Conclusion

The fixed number of available random addresses for RAP poses a significant problem in terms of scalability in cases of many contending stations. In such cases, the protocol's performance is significantly degraded. This paper introduced a TRAP. The protocol employs a variable-length TDMA-based contention stage with the length based on the number of active stations. At the beginning of each polling cycle, each active mobile station registers itself to the base station via the transmission of a short pulse. The base station uses the aggregate received pulse in order to obtain an estimate of the number of active stations, schedules the contention stage to comprise an adequate number of random addresses P in the form of time slots for the registration of these stations to complete with few collisions and transmits a READY message carrying this number to the active mobiles. Each mobile station calculates a random address in the interval $[0\ldots P-1]$ and transmits its registration request in the respective time slot. Then, the base station polls according to the received random addresses. Simulation results that were obtained reveal the superiority of TRAP against the RAP protocol under medium and high loads. Furthermore, the implementation of TRAP is much more simple than that of CDMA-based versions of RAP, since no extra hardware is needed for the orthogonal reception of the random addresses.

References

[1] W. Stallings, Data and Computer Communications, 6th ed, Prentice Hall, Englewood Cliffs, NJ, 2000.
[2] J. Geier, Wireless LANs, Implementing Interoperable Networks, Macmillan Network Architecture and Development Series 1999.
[3] P. Bhagwat, P. Bhattacharya, A. Krishna, S. Tripathi, Enhancing throughput over wireless LANs using channel state dependent packet scheduling, Proceedings of IEEE INFO-COM'96, 1996, pp. 1133–1140.
[4] P. Bhagwat, P. Bhattacharya, A. Krishna, K. Tripathi, Using channel state dependent packet scheduling to improve TCP throughput over wireless LANs, ACM/Baltzer Wireless Networks (1997) 91–102.
[5] E. Gilbert, Capacity of a burst noise channel, Bell System Technology Journal 39 (1960) 1253–1265.
[6] K.-C. Chen, Medium access control of wireless LANs for mobile computing, IEEE Network (1994) 50–63.
[7] K.-C. Chen, C.-H. Lee, RAP—a novel medium access control protocol for wireless data networks, Proceedings of IEEE GLOBECOM, TX, USA, 1993, pp. 1713–1717.
[8] K.-C. Chen, C.-H. Lee, Group wireless addressed polling for multicell wireless data networks, Proceedings of IEEE ICC'94, New Orleans, LA, USA, 1994, pp. 913–917.
[9] J.-W. Dai, The stability of randomly addressed polling protocol, IEICE Transactions on Communications (1997) 1502–1508.
[10] V. Bharghavan, MACAW, a MAC protocol for wireless LANs, ACM SIG-COMM'94, London, UK, 1994, pp. 212–225.