

Adaptive Reservation MAC Protocol for Voice Traffic in Wireless Ad-hoc Networks

Ghalem Boudour, Cédric Teyssié, Zoubir Mammeri,
IRIT, Paul Sabatier University,
Toulouse, France
{boudour, cedric tyssie, mammeri}@irit.fr

Abstract. Due to the On/Off nature of voice traffic, resource reservation for voice traffic in ad-hoc network is a very challenging task. On one hand, allocating resource to voice sources for all the duration of the connection results in bandwidth wasting because of the Off periods. On the other hand, releasing resources during the Off periods and reallocating them during active periods causes large and variable access delay and increases jitter. In this paper we propose an adaptive reservation protocol for voice/data support. The proposed scheme allocates slots to voice sources each time they wake up, and gives them high priority to send their reservation requests over data sources. The bandwidth unused by idle voice sources is rendered temporarily available for reservation. Simulation results show that the proposed scheme improves the performances of voice traffic in terms of dropping rate.

1 Introduction

Voice over Mobile Ad-Hoc Networks (MANETs) has attracted a great attention in recent years. However, the delay constraints of voice applications makes the MAC protocols design for MANETs a very challenging task. It has been shown that the IEEE 802.11 standard is not able to meet delay guarantees of such multimedia applications. Recently, some protocols based on bandwidth reservation start attracting the interest of research community. In reservation MAC protocols each source reserves a number of slots which fulfill its bandwidth requirements, and uses the reserved slots in subsequent super-frames with no contention. The key idea in time-slot reservation is the exchange of control packets between nodes. Each node is required to maintain a coherent view of reservations made by its neighbors through monitoring reservation control packets transmitted by its neighbors. Despite these protocols alleviate efficiently the effects of packet collision during the reservation phase thanks to the use of control mini-slots and collision resolution schemes, a solution that takes into account the dynamic characteristics of VBR traffic -like voice-remains problematic.

As commonly known, voice sources are not active during all the connection. They are equipped with a voice activity detector (VAD), and follow an alternating pattern of talkspurts and silence periods (On/Off). On one hand, if we allocate resource to voice sources for all the duration of the connection we can ensure minimal access delay and satisfy their delay requirements in high traffic load conditions. However,

the scarce bandwidth is wasted during the Off periods. On the other hand, making voice sources release their reservations during the idle periods and reallocating them during active periods results in large access delay and increases delay and jitter. Another issue with this scheme is that a voice source may not find available slots when it switches again to the active period, especially at high traffic load.

The main idea of our reservation scheme, which we call Adaptive Reservation Protocol for Voice (ARPV), is to release temporarily resources reserved by voice sources when they go to the sleep mode, and giving them the opportunity to restore their reservations when they wake up. Neighbor nodes which have data traffic are allowed to reserve slots released temporarily. ARPV is designed to be embedded on mobile nodes with time synchronization capabilities such as GPS.

The rest of this paper is organized as follows. In section 2, we give an overview of reservation MAC protocols proposed in the literature. In section 3, we present the ARPV protocol. Section 4 presents some simulation results. We end the paper by conclusions and future work.

2 Background and Related Work

The IEEE 802.11 [1] standard is considered as the de-facto MAC protocol for wireless networks. Its simplicity and reduced cost have contributed in its wide deployment. However, it suffers serious throughput degradation and unfairness due to the hidden and the exposed terminals and the binary exponential backoff, the thinks which make it unable to fit the requirements of multimedia applications over multi-hop networks. As alternative to this scheme, reservation-based schemes were proposed to provide delay guarantees for multimedia applications.

In FPRP [4], the super-frame is composed of a reservation frame (RF) followed by several information frames (IF). A node, which wants to reserve a slot in the IF, is required to follow a five-phase reservation process during the RF. These five phases are used by each node to compete to reserve a slot, and to inform neighbors about the result of the competition (reservation success or failure).

In CATA [12], the super-frame is composed of slots and each slot is composed of four control mini-slots and one Data mini-slot. Control mini-slots are used to reserve the slot, and prevent neighbors from reserving this slot when it is reserved.

In R-CSMA [5], each super-frame is composed of a contention period (CP) and a set of TDMA slots. A node, which wants to establish a reservation, follows a three way handshake during the CP in order to negotiate reservation of slots with the receiver. Neighbor nodes which hear the reservation packets record the reservation thus preventing any collision during reserved slots. In [8], we have proposed the ER-CSMA protocol, which is an extension of R-CSMA to resolve the reservations clash due to mobility. The reservation clash happens when two nodes which are far away from each other and which have reserved the same slot move. If one of them enters in the transmission range of the other, collisions happen in reserved slots and one or both of them loses its reservation.

RTMAC [3] is a reservation scheme that doesn't need global synchronization between nodes. Each node has its own super-frame which consists of reservation-slots

(resv-slots). To carry its real-time packets, a node reserves a block of consecutive resv-slots, which is called connection-slot and uses the same connection-slot to transmit in successive super-frames. Reservation of a connection-slot is achieved following a three-way handshake like in R-CSMA.

DARE [2] extends the concepts used in RTMAC and R-CSMA for point to point reservations to establish reservations on each hop along a path.

3 Adaptive Reservation Protocol for Voice Traffic (ARPV)

The super-frame of ARPV is composed of a SYNC slot, followed by a Reservation Sub-Frame (RSF), followed by a Data sub-frame composed of S Data slots. The SYNC slot is used for synchronization. All nodes are synchronized on the super-frame basis. Data slots are used to carry voice packets and data packets, and their length is set to the transmission time of one voice packet. The ACK mini-slot is used to acknowledge successful reception of voice and data packets through the transmission of ACK frame. The RSF is composed of R Collision Resolution Slots (CRS) used for the reservation requests and reservation release requests. Each CRS is composed of five control mini-slots used to reserve slots, and inform neighbor nodes about reservations.

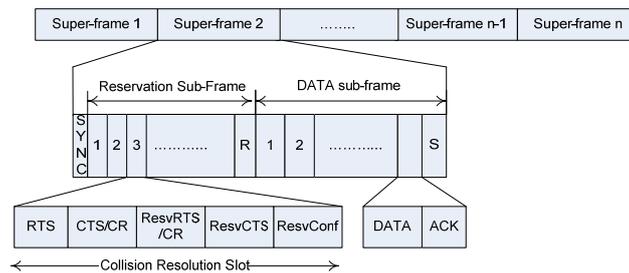


Fig. 1. Super-frame structure in ARPV.

Each node maintains a Slot State Table (SST) which is updated each time a slot is reserved locally or by neighbor nodes. Unlike the other reservation schemes where data sources use contention to send each data packet on available slots, data sources in ARPV can reserve Data-slots at low voice traffic load, and when voice sources go to silence mode. For a node i , a slot may be in one of the following states:

- - *Available for transmission*: No neighbor had reserved the slot for reception and node i can reserve it for transmission.
- - *Available for reception*: No neighbor had reserved the slot for transmission and node i can reserve it for reception.
- - *Transmission reserved for data*: a neighbor of node i has reserved the slot to transmit data packets. Node i can grab the slot to reserve it for voice reception.
- - *Reception reserved for data*: a neighbor of node i has reserved the slot to receive data packets. Node i can grab the slot to reserve it for transmission for a voice source.

- - *Transmission/Reception reserved for voice*: a neighbor has reserved the slot for voice packet transmission (reception respectively).
- - *Temporarily transmission released*: the slot is reserved by a voice source, but is temporarily released because the voice source is in the idle phase. The voice source can restore its reservation when it switches again to the activity period.
- - *Temporarily reception released*: the slot is reserved by a voice receiver, but is temporarily released because the voice source is in idle phase. A data source can reserve this slot for transmission, but it may lose it when the voice connection wakes up.

3.1 Basic reservation scheme

A node (voice or data source), which needs to establish a reservation, chooses a CRS following the scheme described in section 3.2, sends an RTS packet in the 1st mini-slot of this CRS and waits for a Collision Report packet from one of its neighbors or CTS from the intended receiver. The aim behind these two first control mini-slots is to ensure that there is no two-hop neighbor contending for reservation in the current CRS, and that the reservation request will be received by all one-hop neighbors. When the receiver receives correctly the RTS packet it replies with CTS in the 2nd mini-slot. If a collision of the RTS occurs at any neighbor of the sender, the neighbor sends a Collision Report packet in the 2nd mini-slot to indicate that the reservation request could not be recorded. In this case, the sender cancels its reservation request and restarts the collision resolution process. Otherwise, if no collision occurred during the 1st mini-slot, the sender receives correctly the CTS. In this case, the sender sends a ResvRTS packet in the 3rd control mini-slot. Beside source and destination addresses, the ResvRTS contains the list of available slots at the sender, and the number of requested slots, and the class of service (voice or data). The list of available slots specifies the list of slots of the data sub-frame available for transmission from the sender viewpoint. The requested bandwidth is derived from the number of slots.

Any neighbor of the receiver which senses collision during the 2nd mini-slot sends a Collision Report (CR) packet during the 3rd mini-slot. The aim behind this second CR is to indicate to the receiver that more than one node is establishing reservation at the same time and that the reservation could not be recorded. When the receiver receives the CR or senses collision during the 3rd mini-slot, it remains silent during the 4th mini-slot and the reservation process is stopped. In the event of not receiving ResvCTS in the 4th mini-slot, the sender concludes that the reservation has failed and retries the reservation process in another CRS. If no collision occurred during the 2nd mini-slot, the receiver will correctly receive the ResvRTS. It checks its SST and replies with a ResvCTS if there are common free slots between available slots list specified in the ResvRTS and its local reception available slots. ResvCTS specifies the set of slots which will be reserved with the sender. Each node that receives the ResvCTS updates its SST and prevents reserving the specified slots for transmission. When the sender receives ResvCTS, it replies with ResvConfirm packet during the 5th mini-slot indicating the set of slots which have been reserved for transmission. Sender

neighbors that hear ResvConfirm update their SST and will not accept reservation requests for the reserved slots.

3.2 Reservation requests transmission

However, the access scheme during the RSF has an important effect on the performance of the reservation protocol. In this section we propose two access schemes for contention resolution.

Static priority access scheme. In this scheme, voice sources have higher priority to send their requests. They send their reservation requests on CRSs with permission probability p^v , while data sources transmit their requests with probability p^d , such that $p^d < p^v$. The advantage of this scheme is its simplicity. However, it doesn't adapt to the dynamic traffic loads changes.

Dynamic priority access scheme. In this scheme, the permission probability of both voice sources and data sources is adapted to traffic load and collision rate in the network. We adapt the Binary Feedback collision resolution algorithm of Mikhailov [9] which was proposed to stabilize the Slotted-ALOHA scheme in wireless cellular networks. The algorithm defines a recursive function $S(t)$ and updates it slot by slot according to the channel state. $S(t)$ is given by:

$$S(t+1) = \begin{cases} \max\{1, S(t) - e + 1\}, & \text{if slot } t \text{ has } E \text{ feedback} \\ S(t) + 1, & \text{if slot } t \text{ has } NE \text{ feedback} \end{cases} \quad (1)$$

Where e is the base of the natural logarithm, $t \geq 1$, $S(1) = 1$, and E and NE are feedbacks sent by the base station at the end of each slot to indicate that the slot was empty (E) or nonempty (NE). A slot is in the E state if nodes did not transmit packets during this slot, and is NE otherwise. A node with a packet for transmission transmits its packet on slot t with a permission probability $p(t) = 1/S(t)$.

We propose to adapt this algorithm to be used for contention resolution during the *RSF* of our protocol. Every node monitors the channel during the control mini-slots of each CRS in order to detect transmissions of its neighbors. The permission probability is increased if idle channel is detected in the current CRS, i.e. idle channel detected in the 1st and the 2nd control mini-slots. A node reduces its permission probability if reservation failure is detected in the current CRS. The reservation failure is detected when a collision is sensed in the 1st or 2nd control mini-slots, or when a Collision Report is received in the 2nd mini-slot. The node maintains the same permission probability it used in the previous CRS if a successful reservation is established by a one or two-hops neighbor. Successful reservation in a CRS is detected when a ResvCTS or ResvConfirm is received indicating that one or two-hop neighbor is successfully establishing a reservation. The new S function is expressed as follows:

$$S(t+1) = \begin{cases} \max\{1, S(t) - \text{bonus}\}, & \text{if idle channel detected in the 1}^{\text{st}} \\ & \text{and 2}^{\text{nd}} \text{ mini - slot} \\ S(t) + \text{penalty}, & \text{if reservation failure detected} \\ S(t), & \text{if successful reservation established} \end{cases} \quad (2)$$

Where: $Penalty = 1$, $bonus = e$ for voice sources and $Penalty = e$, $bonus = e-1$ for data sources.

A node, which has a reservation request to transmit in the CRS t , calculates $S(t)$ based on the estimate of $S(t-1)$ and transmits its reservation request following the scheme described in section 3.1 with a probability $p(t)=1/S(t)$.

3.3 Adaptive bandwidth reservation for voice and data sources

Voice packets waiting for reservation are queued and dropped if their access delay exceeds the voice delay bound D_{max} . Such a bound is derived from delay constraint of speech communication. Since data sources have less stringent delay requirements, packets generated by data sources without reservation are never dropped. New voice calls are allowed to reserve slots available for transmission, and grab reception reserved slots for data sources. The ResvRTS sent by a new voice source includes slots in “*available for transmission*” and “*reception reserved for data*” states. The receiver can accept a reservation request on slots in the “*available for reception*” and “*transmission reserved for data*” states.

When a voice source switches to the silence period, it releases temporarily its reserved slot so that neighbor data sources can reserve it for reception. A slot reserved by a voice source is considered to be temporarily transmission-released when left empty. When neighbor nodes detect clear channel during this slot, they mark it as temporarily transmission-released and are allowed to reserve it for data reception. However, new voice calls are not allowed to reserve slots in the “*temporarily reception released*” state. This ensures that new voice calls are admitted into the system only if there is an available slot, and that already admitted voice sources can restore their resources when they wake-up from the silence state. A slot reserved for voice reception is considered temporarily reception-released when clear channel is detected during the ACK mini-slot of the slot. Unlike voice sources where slots are temporarily released during inactivity periods, a slot reserved by a data source is considered to be definitively released (available) if not used for transmission for one time. Data sources are required to reserve slots each time they have packets.

A voice source in the *Off* state checks permanently its packet queue. When packets are present in the queue, the node concludes that it switches to the activity period, and temporarily released slots must be restored. The node chooses a CRS, and follows the five reservation phases in order to restore its reservation with the intended receiver. Two reservations restoration strategies can be adopted.

Strategy 1. The sender and the receiver restore their previously reserved slots and don't take care of slot availability at their neighbors. All neighbor data sources, which have eventually reserved these slots, lose their reservations, and are required to reserve other slots. We call this scheme the aggressive reservation recovery. This scheme may lead to an increase of the delay of data packets.

Strategy 2. The waking-up voice source attempts to reserve other available slots than the previously reserved slots if these ones have been reserved by some neighbor

data source. If slot availability at the sender and the receiver doesn't permit to cancel the old reservation, the sender and the receiver restore their old reserved slots. This can be achieved through piggybacking the state of each slot on the ResvRTS. The receiver in its ResvCTS grants reservation on either the old reserved slots, or on other available slots. This strategy has the advantage that it avoids data sources to establish new reservation each time neighboring voice sources switch to the activity period. Consequently, the delay of such sources is optimized.

However, if a voice source which has temporarily released its slot moves far away, nodes which were in its neighborhood will not be able to update the state of the slot. The slot will remain indefinitely in the temporarily released state, and cannot be reserved by other voice sources. A solution to this issue consists in associating a timeout to each temporarily released slot as explained in section 3.5.

3.5 Reservation release scheme

We consider four cases to release reserved slots.

i) *A voice traffic source has finished its transmission.*

At the end of a voice session, the sender executes the collision resolution algorithm during the reservation sub-frame and sends a ResvRelease packet in the first control mini-slot of a CRS. The ResvRelease specifies the set of slots to be released. Any neighbor of the sender that hears a collision during the first mini-slot of the CRS sends a Collision Report (CR) in the 2nd mini-slot. If the sender hears a CR or collision in the 2nd mini-slot it chooses another CRS and continues sending the ResvRelease until no CR is received in order to ensure that all neighbors have received the ResvRelease. When the receiver receives the ResvRelease, it sends a ResvRelease in the same way as the sender in order to inform its neighbors about the reservation release. The ResvRelease sent by the receiver specifies the same set of slots that the sender has requested to release.

ii) *A data source has lost its reservation because a voice source has restored its reservation or a new voice source grabs the reservation.*

When a data source receives a ResvCTS which indicates a slot that it has reserved, the data source sends a ResvRelease packet to inform the receiver and release the reservation at neighbor nodes. Similarly, if the receiver of a data source receives a ResvConfirm for a slot which has been locally reserved, it releases the slot with the sender and with its neighbors.

iii) *Connectivity is lost because sender or receiver movement.*

As previously noticed, a slot reserved for voice is considered to be temporarily released when the voice source switches to the silence period. However, the slot may be also detected idle when the sender moves far away. The slot may stay indefinitely in the temporarily released state as it will never be restored. Consequently, the slot cannot be recycled. In order to avoid this situation, a connection timeout is associated with each temporarily released slot. If the node, which has reserved the slot, does not restore it before the timeout expires, the reserving node is considered moved far away, and the slot is made available for reception. Both neighboring voice and data sources will be able to reserve this slot for reception.

ARPV provides also a mechanism to allow reserving nodes to detect connectivity loss due to mobility of the receiver. When reserving node transmits a packet in its reserved slot, it is expected to receive either a positive or a negative acknowledgement during the corresponding ACK mini-slot. Loss of connectivity with the receiver is detected through monitoring the channel during this ACK mini-slot. If the sender does not receive any response to its transmission from the intended receiver, it concludes that the receiver is no more in the neighborhood. It releases its reserved slot, and stops sending voice packets during its reserved slot. The slot is made available for reception at the sender neighbors either when the connection timeout associated with the slot expires, or by making the sender sending an explicit ResvRelease packet to release the slot at its neighbors.

iv) Reservation is lost at the receiver side because collisions during reserved slots.

If a node detects collision during its reception-reserved slot, it concludes that the reservation is lost. It applies the reservation recovery process which consists in two steps. The first step consists in releasing the lost slot with the sender through sending a ResvRelease packet. The second step consists in the reservation repair. When the sender receives the ResvRelease it cancels its reservation and stops sending packets during reserved slot. Afterward, the sender restarts the reservation negotiation process with the receiver in order to reserve a new slot.

4 Simulation

We evaluate the performance of ARPV protocol and compare its behavior with the two contention resolution schemes using the NS-2 simulator. Performances of ARPV are also compared with the IEEE 802.11e EDCF scheme. For this purpose, we use the EDCF simulation model of Wiethölder [7]. Our protocol can be easily coupled with a QoS routing protocol to establish multi-hop end-to-end connections. However, in order to isolate the effectiveness of ARPV from that of the QoS routing protocol, we consider only single-hop connections in our current simulations.

Table 1. Simulation and protocol parameters

Parameter	Value
Channel bit rate (Mbps)	2
Transmission range (m)	150
MAC header (bytes)	38
PHY layer overhead (PLCP header+preamble) (bits)	4+48
Data slot payload size (bytes)	160
Data Slot length (ms)	1.18
Guard time between slots (μ s)	2
Super-frame length (ms)	20
Maximum voice packets tolerable delay D_{max} (ms)	200
Connection timeout (s)	10
(p^v, p^d) (for static priority scheme)	(0.35, 0.25)
Number of data slots per super-frame	11
Number of CRS per super-frame	15
Simulation time (s)	1000s

Table 2. Contention parameters for EDCF and DCF

Traffic category	CWmin	CWmax	AIFSN
Data traffic	31	1023	7
Voice traffic	7	15	2
DCF (both voice and data)	31	1023	2

4.1 Simulation Model

Each node can be the source of a G.711 voice flow that generates packets of 160 bytes payload at 64 kbps. The talkspurt and silence period durations are exponentially distributed with a mean of 1 s and 1.35 s respectively. Voice sessions have duration of 200 s, and are randomly started. The *connection timeout* value is 10 s.

The super-frame length is set to the inter-arrival time of voice packets (i.e. 20 ms). Each Data slot consists of the transmission time of one voice packet. Each voice source is required to reserve a slot per super-frame. Following the ITU G.114 recommendation [6], a one-way delay of 300-800 ms is acceptable for some users, while others would not accept a delay above 200 ms. Voice packet loss in the range of 5-10 % are acceptable. In this paper, we assume a 200 ms voice delay bound. For data traffic, we consider FTP sessions that transfer 10 MB files started at random instants. Simulation and protocol parameters are summarized in table I. As the performances of the static priority scheme are sensitive to p^v and p^d values, we did several experiments in order to choose the best values. Following these experiments, a choice of $p^v=0.35$ and $p^d=0.25$ gave the best performances for both data and voice traffics. Two topologies are considered: a grid topology without mobility and random topology with mobility. The first one is used to evaluate the global performance of the protocol. The second one is used to evaluate the effects of node mobility.

4.2 Performance evaluation in grid topology scenarios

We consider a topology where 64 nodes are placed on a grid of 700m×700m size with distance between pair of adjacent nodes equals to 100 m. Nodes are considered stationary. We uniformly increase the load by increasing data and voice sessions in equal numbers. Figure 2 shows the voice traffic delay achieved by ARPV, DCF, and EDCF with the increase of traffic load. The figure shows that ARPV achieves lower delay than EDCF and DCF at high traffic load, and it is less affected by the increase of traffic load. The low impact of traffic load on voice traffic delay with ARPV is provided owing to the protection of voice traffic streams from the fluctuation of traffic load. A slot is reserved to each voice source and voice packets are transmitted at regular interval regardless of the traffic load. The high delay of EDCF and DCF is due to the increase of contention and collision rates at high traffic load. However, at low traffic load, EDCF and DCF outperform ARPV because the low level of contention implies a small number of collisions and small backoff. The same conclusions are drawn for data traffic delay (figure 3) with higher delay in comparison to voice traffic delay.

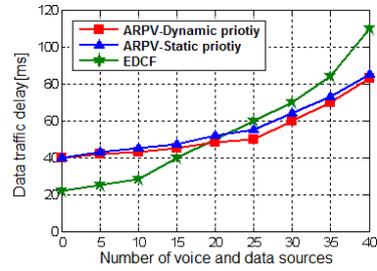
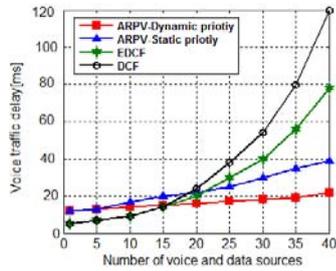


Fig. 2 (left). Voice traffic delay. **Fig. 3 (right).** Data traffic delay.

The dynamic priority scheme achieves lower voice traffic delay than the static priority one. This is due to the adaptive access scheme of the dynamic priority scheme during the *RSF*. As nodes adapt their permission probability according to the collision rate, the chance of successful access during the *RSF* is increased.

Figure 4 shows that all schemes achieve low dropping rate at low traffic load, while ARPV achieves better dropping rate than the EDCF at high load. At high traffic load, the DCF achieves the highest dropping rate (8%). The dropping rate with EDCF increases up to 5%. With the static priority it grows up to 2%, while the dynamic priority achieves the lowest dropping rate. Packet dropping with ARPV occurs mainly when voice sources switch from silence period to activity period at the beginning of talkspurts. Packets generated between the time a talkspurt is generated and a reservation is established are dropped if this time exceeds the maximum tolerable delay. Figure 5 shows that while the delay jitter with ARPV remains around 6 ms, it increases linearly with the EDCF and DCF. Unlike with ARPV where voice packets are transmitted at regular intervals, the voice packets delay with EDCF and DCF is unpredictable and depends on the values taken by the backoff timer.

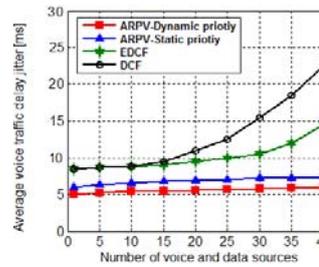
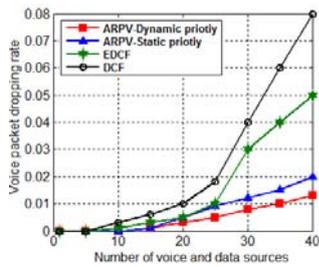


Fig. 4. Voice packet dropping with load increase **Fig. 5.** Voice traffic jitter with load increase

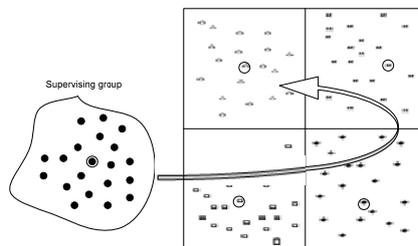


Fig. 6. Simulated area with 4 regions and 5 moving groups.

4.3 Performance evaluation in mobile environment

We consider a battlefield scenario where the army is assumed composed of groups of soldiers. Each group may have the task of scanning a particular region. To simulate soldier movements, we use the Reference Point Group Mobility (RPGM) model which is more appropriate to simulate such scenarios [10]. We consider 100 mobile nodes placed on a 1000m×1000m square area. Some nodes are pedestrians and others are vehicles. Simulated area is decomposed into four equal size regions as shown in figure 6. Nodes are partitioned into 5 groups of 20 nodes each, and each group has a leader. Four groups are assigned four regions. The 5th group is the supervising group that roams around the entire area.

In RPGM, the motion of a group leader characterizes the movement of group members. Hence, the group trajectory is determined by providing a path for the group leader in the form of a sequence of velocity vectors $V_{leader}(t)$. $|V_{leader}(t)|$ denotes the speed of group leader at time t , and $\theta_{leader}(t)$ denotes the angle made by $V_{leader}(t)$ with the X -axis. Initially, group leaders are randomly placed on the region to which they are affected. Then, nodes of a group are uniformly distributed in the neighborhood of their leader. Subsequently, each group member has a speed and direction that are derived by random deviation from those of the group leader. The velocity (speed and direction) of a group member at time instant t is calculated as follows:

$$|V_{member}(t)| = |V_{leader}(t)| + random() \times SDR \times max_speed \quad (3)$$

$$\theta_{member}(t) = \theta_{leader}(t) + random() \times ADR \times max_angle \quad (4)$$

Where SDR is the Speed Deviation Ratio, and ADR is the Angle Deviation Ratio, with $0 \leq SDR$ and $ADR < 1$. $random()$ returns a value uniformly distributed in $[-1, 1]$. The values of SDR and ADR control the deviation of the velocity of group members from that of their leader. In our simulation, max_angle is set to 2π and the max_speed is 10 m/s. In order not to have a strong deviation between the group leader velocity and the velocity of its group members, both ADR and SDR are set to 0.1. We use the IMPORTANT mobility generation framework [11] to generate mobility traces. The first four groups move randomly in there regions. For the 5th (supervising) group, the checkpoints list is chosen to give a circular motion to the group on the entire area. The number of checkpoints for each group leader is 20. The output trace file generated by IMPORTANT is used by the NS-2 in simulation.

We fixed the traffic load to 40 voice connections, 8 connections in each group, and increased the mobility from 0 m/s to 10 m/s with an increment of 1 m/s. The impact of mobility on the performance of ARPV is illustrated in figures 7 and 8.

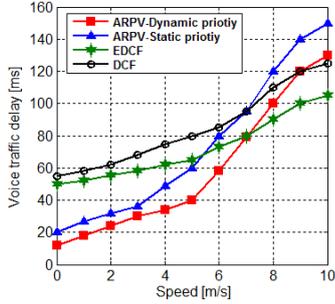


Fig. 7. Voice traffic delay under mobility

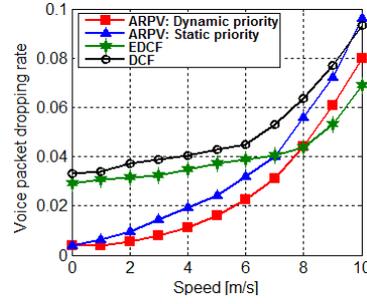


Fig. 8. Voice packet dropping under mobility

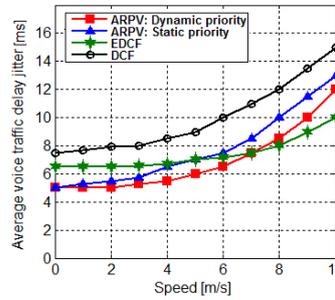


Fig. 9. Voice traffic delay jitter under mobility.

ARPV behaves well until a mobility of 7 m/s where the EDCF outperforms ARPV. We observe also that the delay and dropping rate with ARPV increase faster than with EDCF. At 10 m/s, the delay with the dynamic priority scheme increases up to 130 ms, and up to 150 ms with the static priority scheme. It increases up to 100 ms with EDCF and to 120 ms with DCF. The fast delay increase with ARPV is due to the increase of queuing delay of voice packets when nodes lose their reservations due to mobility. At high mobility, nodes lose their reservations more frequently. When a node loses its reservation, its voice packets are queued until a new reservation is established. The increase of dropping rate of ARPV at high mobility is mainly due to two reasons: at high mobility, packets collide during reserved slots more frequently, and many voice packets waiting for reservation are dropped when their queuing delay exceeds the maximal tolerable delay D_{max} . However, for a speed lower than 7 m/s, ARPV achieves lower dropping rate and delay than EDCF because at low speed the network topology is more stable and reservation disruption occurs less frequently. As shown in Figure 9, lower jitter is achieved by ARPV until a speed of 6m/s where higher delay jitter is observed due to the variation in queuing delay and frequent interruptions of the communications at high mobility.

5 Conclusion

In this paper, we propose a reservation MAC protocol for voice and data support in MANETs. In the proposed protocol, reservation priority is given to voice sources. The latter release temporarily their slots when they switch to the silence period so that other traffic sources can reserve them. In order to ensure that voice sources get enough available resources when they wake up from the silence period once they are admitted into the system, voice sources restore their temporarily released slots each time they wake up. It is feasible to use ARPV for transmission of voice traffic with minimal delay and dropping rate along with data traffic over mobile networks where the time synchronization is available. However, the results show that ARPV performs better in low-mobility and static scenarios. Thus, it seems to be suitable approach for voice traffic support in wireless mesh networks.

References

1. IEEE 802.11WG. Draft Supplement to Standard for Telecommunications and Information Exchange between Systems-LAN/MAN Specific Requirements-part11: MAC Enhancements for Quality of Service (QoS). IEEE 802.11e standard Draft/D13.0 (2005)
2. Carson, E., et al.: A Distributed End-to-End Reservation Protocol for IEEE 802.11-Based Wireless Mesh Networks. IEEE JSAC, Vol. 24, No. 11 (2006)
3. Manoj, B.S., Siva Ram Murthy, C.: Multimedia Traffic Support for Asynchronous Ad-hoc Wireless Networks. BROADNETS'04, pp. 569—578. California (2004)
4. Zhu, C., Corson, M.S.: A Five-phase Reservation Protocol (FPRP) for Mobile Ad hoc Networks. IEEE INFOCOM'98, pp. 322—331. San Francisco (1998)
5. Joe, I.: Qos-Aware MAC with Reservation for Mobile Ad-Hoc Networks. In: IEEE Vehicular Technology Conference, pp. 1108—1112. Los Angeles (2004)
6. ITU Rec. G.114, "One-way Transmission Time," ITU, February 1996.
7. Wiethölter, S., Hoene, C.: Design and Verification of an IEEE 802.11e EDCF Simulation Model in ns-2.26. TKN-03-19 Technical Report, Technische Universität Berlin (2003)
8. Boudour, G., Teyssie, C., Mammeri, Z.: Reservation Clash Handling to Optimize Bandwidth Utilization in MANETs. International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ'08), pp. 77--82. Bucharest (2008)
9. Mikhailov, V.A.: Geometrical Analysis of the Stability of Markov Chains in R^n_+ and its Application to Throughput Evaluation of the Adaptive Random Multiple Access Algorithm. Problemy Peredachi Informatsii, Vol. 24, No. 1, pp. 61--73, (1988)
10. Camp, T., Boleng, J., Davies, V.: A Survey of Mobility Models for Ad hoc Network Research. Wireless Communications & Mobile Computing, Vol. 2, No. 5 (2002)
11. Bai, F., Sadagopan, N., Helmy, A.: The IMPORTANT Framework for Analyzing the Impact of Mobility on Performance of Routing for Ad Hoc Networks. Ad-Hoc Networks Journal, Vol. 1, No. 4 (2003)
12. Tang, Z., Garcia-Luna-Aceves, J.J.: A Protocol for Topology-Dependent Transmission Scheduling in Wireless Networks. IEEE WCNC'99, pp. 133--1337. New Orleans (1999)