Available online at www.sciencedirect.com

ScienceDirect

Ad Hoc Networks 6 (2008) 424–440

Ad Hoc Networks

www.elsevier.com/locate/adhoc

A MAC layer protocol for wireless networks with asymmetric links

Guoqiang Wang, Damla Turgut *, Ladislau Bölöni, Yongchang Ji, Dan C. Marinescu

School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816-2450, USA

Received 5 June 2006; received in revised form 25 February 2007; accepted 17 March 2007
Available online 24 March 2007

Abstract

We introduce AsyMAC, a MAC layer protocol for wireless networks with asymmetric links and study a protocol stack consisting of AsyMAC and the A^4LP routing protocol. The two protocols are able to maintain connectivity where the standard IEEE 802.11 MAC protocol coupled with either AODV or OLSR routing protocols may loose connectivity. A comparative study shows that AsyMAC improves on two previously proposed protocols' accuracy in determining the nodes to be silenced to prevent collisions.
Published by Elsevier B.V.

Keywords: Asymmetric link; AsyMAC; Heterogeneous MANET

1. Introduction

Asymmetric links are present in wireless networks for a variety of physical, logical, operational, and legal reasons:

(a) *The transmission range is limited by the node hardware.* The hardware properties of the node (for instance, the antenna or the RF circuits) determine the maximum transmission range. The different transmission ranges of the nodes lead to asymmetric links, which cannot be avoided except by physically changing the nodes' hardware components, for instance by installing a different antenna.

(b) *Power limitation.* Different nodes may have different power constraints. For instance, node A may have sufficient power reserves and a transmission range enabling it to reach node B; however, node B has limited power, and either (i) cannot reach node A, or (ii) may choose not to reach node A to save power. The two scenarios influence the design of the protocols in different ways. In the second scenario, node B is capable to reach node A and we could exploit this capability for short transmissions when necessary, e.g., during a network setup phase.

[*] Corresponding author. Tel.: +1 407 823 6171; fax: +1 407 823 5835.
E-mail address: turgut@cs.ucf.edu (D. Turgut).

1570-8705/$ - see front matter Published by Elsevier B.V.
doi:10.1016/j.adhoc.2007.03.004

(c) *Interference.* Node A can reach node B and node B can reach node A, but if node B would transmit at a power level sufficient to reach node A, it would interfere with node C who might be a licensed user of the spectrum. This scenario is critical for transmitters which attempt to opportunistically exploit unused parts of the licensed spectrum (such as unused television channels). Even if operating in the unlicensed bands, dynamic spectrum management arrangements might have given the priority to node C, thus node B needs to refrain from sending at a power level above a given threshold.

(d) *Stealth considerations.* Node A and node B attempt to communicate and they wish to hide the existence or the exact location of node B from node O. One way to achieve this is to restrict the transmission power of node B to the minimum and/or transmit on frequencies which make location detection more difficult. This is especially important in military/battlefield applications where low probability of detection (LPD) is an important consideration [20,21].

(e) *Dynamic spectrum management.* In the emerging field of software defined radios, the nodes can transmit virtually in any band across the spectrum, but they need to share the spectrum with devices belonging to licensed operators as well as devices with limited flexibility. Once any of the reasons discussed previously force a link to be unidirectional additional constraints, e.g., the need for a reverse path between some pairs of nodes may cause other links to change their status and operate in a unidirectional mode, even when there is no other reason for unidirectionality.

Inability of some MAC protocols to exploit the asymmetry of some of the communication channels could lead to an inefficient bandwidth utilization, or, in the worst case, to inability to connect some of the nodes. To exploit the asymmetric links, the protocols must be able to deliver the acknowledgements back to the sender in a direction opposite to the direction of the asymmetric link. Furthermore, the problem of hidden nodes appears more often and in more complex forms than in the case of symmetric links. Depending whether the routing protocol of the wireless ad hoc network is able to handle asymmetric links, the MAC protocol might need to hide the existence of asymmetric links with a symmetric overlay. The challenge for a MAC layer protocol able to exploit asymmetric links is to solve the hard problems mentioned above, while keeping the cost incurred lower than the benefits obtained from the utilization of the asymmetric links.

MAC protocols for asymmetric links were previously proposed by Poojary et al. [15], Fujii et al. [6] and others. In this paper, we introduce a new protocol, AsyMAC (asymmetric MAC) that uses a geometric analysis of the hidden node problem in the presence of asymmetric links for a more precise determination of the nodes which need to be silenced during a transmission. Informally, a hidden node is one that can interfere with the reception of a data packet without the knowledge of the sender. As a note, there is a difference between the concept of a protocol, as the collection of features necessary to implement networking at a certain layer, and algorithm, which refers to the implementation of a specific functionality. In this paper, when we refer to a protocol, we concentrate on the subset of functionality necessary to implement the asymmetric links, thus the terms algorithm and protocol will be used interchangeably.

The paper is organized as follows. Related work is presented in Section 2. Section 3 presents AsyMAC protocol in every aspects. Section 4 describes the simulation environment and presents the results of the simulation study of the effect of network load, network mobility and number of nodes. We conclude in Section 5.

2. Related work

MAC layer protocols allow a group of users to share a communication medium in a fair, stable, and efficient way. A MAC layer protocol for wireless ad hoc networks must address several specific problems:

1. Mobility – the connection between nodes can become unstable because of the independent movement of the nodes.
2. Higher error rates – a wireless channel has a higher *bit error rate* (BER) than a wired network.
3. Inability to detect collisions during some periods of time – wireless transceivers work in a half-duplex mode; nodes do not “listen” when “talk” and do not “talk” when “listen”. The sender is unable to detect the collision and the receiver is

unable to notify the sender of the collision during the transmission of a packet. *Collision avoidance* is almost mandatory.

Carrier sensing multiple access (CSMA) [10] requires every node to sense the channel before transmitting, and if the channel is busy, refrain from transmitting a packet. CSMA reduces the possibility of collisions in the vicinity of the sender. Multiple access collision avoidance (MACA) [9] and its variant MACAW [2] are alternative medium access control schemes for wireless ad hoc networks that aim to solve the hidden node problem by reducing the possibility of collisions in the vicinity of the receiver.

The floor acquisition multiple access (FAMA) [7] protocol consists of both carrier sensing and a collision avoidance handshake between sender and receiver of a packet. Once the control of the channel is assigned to one node, all other nodes in the network should become silent. Carrier sensing multiple access based on collision avoidance (CSMA/CA), the combination of CSMA and MACA, is considered a variant of FAMA protocols. The IEEE 802.11 standard [8] is the best-known instance of CSMA/CA.

In a wireless network with symmetric links, *a hidden node is one out of range of the sender, but in the range of the receiver*. The solution provided by the 802.11 MAC to the hidden node problem is the RTS/CTS handshake mechanism. Xu et al. [11] analyzes the effectiveness of RTS/CTS handshake mechanism, and indicates that some of the hidden nodes may not be covered by the receiver due to the fact that it requires much lower power to interrupt a packet reception than to successfully deliver a packet.

We can define a hidden node in wireless ad hoc networks with asymmetric links as *a node out of the range of the sender and whose range covers the receiver* (see Fig. 1b). Thus, a hidden node is hidden from the sender and possibly from the receiver as well. The RTS/CTS handshake mechanism is not a solution for such networks since a CTS packet may not be able to reach hidden nodes.

Several solutions to the hidden node problem in wireless ad hoc networks with asymmetric links are discussed in the literature. Poojary et al. [15] propose that a node rebroadcasts a CTS packet if it is received from a low-power node. To decrease the probability of collisions, each node waits a random number (1,...,6) of SIFS (short inter-frame spacing) periods before transmitting a CTS packet. Fujii et al. [6] made several improvements relative to [15]: (i) not only CTS but also RTS packets are rebroadcasted; (ii) nodes with a CTS packet to rebroadcast, first sense the channel and transmit only if the channel is not busy; and (iii) only high-power nodes rebroadcast RTS or CTS packets. The solutions proposed by [15,6] can lead to inefficient use of the channel if nodes are *misclassified* as hidden nodes. In such situations, nodes that could have been active are silenced due to misclassification, severely degrading the channel utilization. Refs. [15,6] routinely assume routing over symmetric links so that the sender is able to receive both CTS and ACK packets. In the presence of asymmetric links, however, the sender might not receive the CTS or ACK packets, thus the sender cannot trigger the transmission of DATA packets, and does not know whether a transmission was successful or not.

Bao et al. [1] propose a collision-free dynamic channel access scheduling algorithm PANAMA. Two scheduling algorithms are proposed for networks with unidirectional links, NAMA-UN that is node activation oriented and supports broadcast traffic efficiently, and PAMA-UN that is link activa-

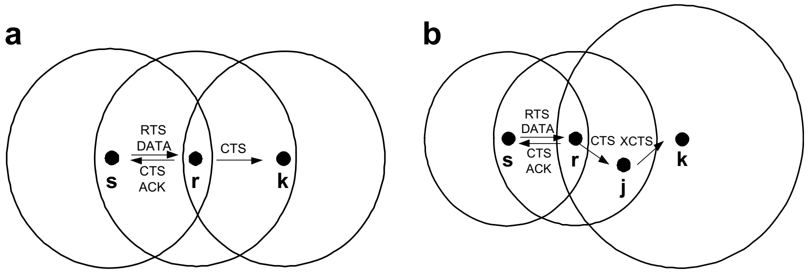


Fig. 1. (a) Hidden node problem in a "classical" wireless network with mobile nodes. All links are assumed to be bidirectional. A hidden node is a node out of the range of the source and in the range of the receiver node. *k* is a hidden node for a transmission from node *s* to node *r*. (b) Hidden node problem in a heterogeneous wireless network with mobile nodes. A hidden node is a node out of the range of the sender and whose range covers the receiver. *k* is a hidden node for transmission from node *s* to node *r*.

tion oriented and is more suitable for relaying unicast traffic. The channel access is allocated for NAMA-UN and PAMA-UN alternatively, with each scheduling algorithm lasting for a fixed amount of time. In PANAMA, the sender node is able to detect the hidden node that also attempts to relay traffic to the receiver. The winner of a contention is the node with higher priority. When the link from the hidden node to the receiver is unidirectional, the hidden node may not be aware of the sender. In these cases, the hidden node is automatically considered as having a higher priority.

The protocols considered previously are based on the modification of the MAC protocol. In contrast, the sub routing layer (SRL) project [17,16] handles asymmetric networks by adding an intermediary layer between the MAC and network layers. This layer partially isolates the routing protocol from the MAC layer, although it still allows the routing protocol to directly contact the MAC layer. For unidirectional links, reverse paths are computed using the reverse distributed Bellman–Ford algorithm. The SRL implementation also signals the detection of new neighbors and the loss of (unidirectional) links.

A MAC layer protocol able to utilize asymmetric links should be stacked together with routing protocols that can utilize asymmetric links as well. A^4LP [19] is a location-aware and power-aware routing protocol designed for ad hoc networks with asymmetric links. In A^4LP, neighbors are re-classified as In-bound, Out-bound, and In/Out-bound neighbors due to the asymmetry of links. A^4LP is composed by a neighbor discovery protocol, a path discovery protocol, and a path maintenance mechanism. A^4LP proposes an advanced flooding technique – *m-limited forwarding*. Receivers can rebroadcast a packet only if its *fitness value* exceeds a predefined threshold, specified by the sender. The fitness function used by m-limited forwarding can be tuned to minimize the power consumption, maximize the stability of the routes, minimize the error rates or the number of retransmissions. By avoiding a full broadcast, m-limited forwarding reduces the cost of path discovery. A^4LP is a hybrid ad hoc routing protocol, combining features of both pro-active and on-demand protocols. The routes to In-, Out-, and In/Out-bound neighbors are maintained by periodic neighbor update and immediately available upon request, while the routes to other nodes in the network are obtained by a path discovery protocol.

In the following sections, we introduce a new MAC layer protocol for ad hoc networks with asymmetric links (AsyMAC). AsyMAC currently works with A^4LP, since they share the process of neighbor discovery and neighbor maintenance.

3. The asymmetric MAC (AsyMAC) protocol

3.1. Topological considerations

The handling of the hidden nodes is an essential problem for wireless MAC protocols operating in the presence of asymmetric links. We introduce topological concepts necessary to define a hidden node of a network with asymmetric links.

The connection between two nodes is described by the Boolean *reachability function* $\mathcal{R}(i,j,t)$ which can be interpreted as follows: a node i can send a packet to node j at time t if and only if $\mathcal{R}(i,j,t) = \text{true}$. A link between two nodes is symmetric if $\mathcal{R}(i,j,t) = \mathcal{R}(j,i,t) = \text{true}$. Note that the reachability is a time varying function; the connection can be affected by various channel conditions, fading, the mobility of the node or the mobility of the obstacles in the field.

We assume that every node is aware of the current values of the reachability function between itself and the neighboring nodes (in both direction). In the A^4LP/AsyMAC protocol stack, it is the role of the *neighbor discovery protocol* of A^4LP to find these values and keep them up-to-date. Although neighbor discovery is a common feature of ad hoc routing protocols, most protocols will not detect outbound neighbors, because the confirmation message will not reach back to the originating node. Asymmetric routing protocols, such as A^4LP have a provision to route back the confirmation messages even in the absence of a direct link, thus allowing the discovery of the full asymmetric reachability matrix. In the following definitions we omit the time parameter even though all the sets are variable in time, a fact which needs to be considered by the protocols relying on them.

We define a series of topological concepts related to communication in the presence of asymmetric links and illustrate for the simple scenario in Fig. 2; a sender node s sends a packet to the receiver node r in the vicinity of nodes $1,\ldots,9$. The circles centered at s and r show the transmission ranges of the sender and the receiver, respectively. The reachability information of other nodes is shown by directed lines; to avoid cluttering the figure we do not

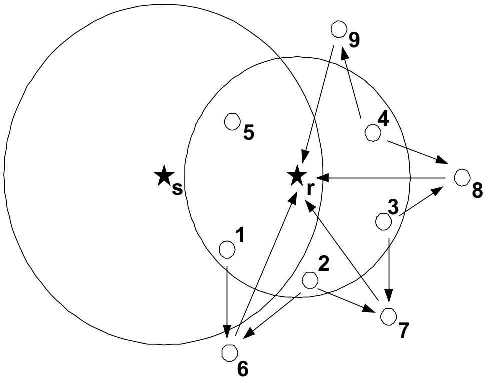


Fig. 2. An illustration for topology concepts. The transmission ranges of the sender *s* and the receiver *r* are reflected by the circles centered at them. The partial reachability information of other nodes is shown by directed lines.

include the links not relevant to the scenario. In this simple scenario we assume that the asymmetric links are caused by the nodes having different transmission ranges and the transmission range is a disk; this is not necessarily true in real life scenarios, and our definitions do not assume a unit disk model.

Definition 1. A set of *m* nodes $i_1, i_2, \ldots i_m \in \mathcal{N}$ are in an *m*-party proxy set if each node can reach the other $m-1$ nodes either directly or through a subset of the other $m-2$ members.

For instance, in the scenario in Fig. 2 the three-party proxy sets are $\{r, 1, 6\}$, $\{r, 2, 6\}$, $\{r, 2, 7\}$, $\{r, 3, 7\}$, $\{r, 3, 8\}$, $\{r, 4, 8\}$, and $\{r, 4, 9\}$.

Definition 2. Call the *vicinity* of node *i*, V_i the set of all nodes that could be reached from node *i*.

$$V_i = \{j|\mathcal{R}(i,j)\}. \tag{1}$$

In our scenario, the vicinity of the receiver node *r* is $V_r = \{1, 2, 3, 4, 5\}$.

Definition 3. Call H_{sr} the set of *hidden nodes of a transmission* T_{sr}. H_{sr} includes nodes that are not reachable from the sender, but from which the receiver is reachable:

$$H_{sr} = \{k|\neg\mathcal{R}(s,k) \wedge \mathcal{R}(k,r)\}. \tag{2}$$

Note that H_{sr} are the hidden nodes for the transmission of the DATA packets, while H_{rs} are the hidden nodes for the transmission of ACK packets.

In our scenario, the hidden nodes of the transmission from source node *s* to receiver node *r* are $H_{sr} = \{2, 3, 4, 6, 7, 8, 9\}$.

Definition 4. Call $P3_i$ the *three-party proxy set coverage* of node *i*. $P3_i$ is the set of nodes which are either reachable by node *i* directly or participate in a three-party proxy set with node *i* and a third node.

$$P3_i = \{k|\mathcal{R}(i,k) \vee \exists_j(\mathcal{R}(i,j) \wedge \mathcal{R}(j,k) \wedge \mathcal{R}(k,i))\}. \tag{3}$$

In the scenario of Fig. 2 the three-party proxy set coverage of node *r* is $P3_r = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

Definition 5. Call $H3_{sr}$ the *hidden nodes* of a transmission T_{sr} in the three-party proxy set coverage of node *r*. The set $H3_{sr}$ includes hidden nodes covered by $P3_r$.

$$H3_{sr} = H_{sr} \cap P3_r. \tag{4}$$

In our scenario, the hidden nodes in the three-party proxy set coverage of *r* are $H3_{sr} = \{2, 3, 4, 6, 7, 8, 9\}$.

Definition 6. Call $XH3_{sr}$ the *extended hidden nodes* of a transmission T_{sr} in three-party proxy set coverage of node *r*. The set $XH3_{sr}$ includes nodes in $H3_{sr}$ not covered by V_r.

$$XH3_{sr} = H3_{sr} - V_r. \tag{5}$$

In the scenario of Fig. 2, the extended hidden nodes of the transmission from source node *s* to receiver node *r* are $XH3_{sr} = \{6, 7, 8, 9\}$.

Definition 7. Call $XHR3_{sr}$ the *extended hidden nodes relay set* of a transmission T_{sr} in three-party proxy set coverage of node *r*. $XHR3_{sr}$ includes *all* nodes in $P3_r$ that could relay traffic from node *r* to nodes belonging to $XH3_{sr}$.

$$XHR3_{sr} = \{j|j \in V_r \wedge \exists_{k \in XH3_{sr}}(\mathcal{R}(j,k))\}. \tag{6}$$

The extended hidden nodes relay set of the transmission from *s* to *r* on the example scenario is $XHR3_{sr} = \{1, 2, 3, 4\}$.

Definition 8. Call $mXHR3_{sr}$ a *minimal extended hidden nodes relay set* of a transmission T_{sr} in three-party proxy set coverage of node *r*. $mXHR3_{sr}$ includes a subset of nodes from $XHR3_r$ ($mXHR3_r \subseteq XHR3_r$) such that (i) the node *r* can relay traffic to any node in $XH3_{sr}$ through some nodes from $mXHR3_{sr}$ and (ii) the removal of any node from $mXHR3_{sr}$ makes some nodes in $XH3_{sr}$ unreacheable from *r*.

$$\forall_{k\in XH3_{sr}}\exists_{j\in mXHR3_{sr}}(\mathcal{R}(j,k)) \tag{7}$$

and

$$\forall_{j'\in mXHR3_{sr}}\exists_{k\in XH3_{sr}}\nexists_{j\in mXHR3_{sr}-\{j'\}}(\mathcal{R}(j,k)). \tag{8}$$

Note that $mXHR3_{sr}$ may not be unique, and different minimal extended hidden nodes relay sets could contain a different number of nodes. Call $\{mXHR3_{sr}\}$ the set that contains all possible sets of $mXHR3_{sr}$.

For instance, in our scenario there are two possible minimal extended hidden nodes relay sets: $mXHR3_{sr}=\{2,4\}$ and $mXHR3_{sr} =\{1,3,4\}$. Also note that the two sets have a different number of nodes.

Definition 9. Call $MXHR3_{sr}$ the *minimum extended hidden nodes relay set* of a transmission T_{sr} in three-party proxy set coverage of node r. $MXHR3_{sr}$ is the instance of $mXHR3_{sr}$ with the smallest number of nodes. Call $\{MXHR3_{sr}\}$ the set that contains all possible sets of $MXHR3_{sr}$.

In our scenario, we need to simply pick the smallest of the possible $mXHR3_{sr}$ sets, which in our case will be $MXHR3_{sr}=\{2, 4\}$.

We note that all the definitions provided above are *constructive*, providing their own implementation methodology. Every set is defined based on the cascade of definitions preceding it, and all of them can be reduced to the reachability matrix $\mathcal{R}(i,j)$.

3.2. Determination of the sets in AsyMAC

The sets V_r and $P3_r$ of node r are the direct results of the neighbor discovery protocol of A^{4}LP. Based on which, we can determine the sets in AsyMAC.

1. H_{sr} includes all the hidden nodes of a transmission T_{sr}, which might be outside of the three-party proxy coverage of node r ($P3_r$), thus the complete set of nodes of H_{sr} may not be found and is not maintained.
2. The members of $H3_{sr}$ can be found by removing from the set $P3_r$ the nodes that can be reached by the other peer of the transmission. Note that in A^{4}LP/AsyMAC, the reachability information of two neighbors of a node can be calculated based on their locations and transmission ranges.
3. $XH3_{sr}$ is obtained by $XH3_{sr}=H3_{sr}-V_r$.
4. $XHR3_{sr}$ includes all nodes in V_r that can reach a node in set $XH3_{sr}$.
5. The calculation of $\{mXHR3_{sr}\}$ is described in Algorithm 1.

Algorithm 1 (*Calculation of* $\{mXHR3_{sr}\}$).

```
{$mXHR3_{sr}$} = $\Phi$;
List the complete permutation of $XHR3_{sr}$, call it $P$.

/* Find $mXHR3_{sr}$ for each permutation $P_i \in P$. */
for all permutations $P_i \in P$ do
    $mXHR3_{sr} = \Phi$;
    $T = XH3_{sr}$;
    found = false;
    while $P_i \neq \Phi \bigwedge found =$ false do
        remove the next node $p$ from $P_i$, $P_i = P_i - \{p\}$;
        $mXHR3_{sr} = mXHR3_{sr} \cup \{p\}$;
        for all nodes $t \in T$ do
            if $\mathcal{R}(p,t)$ then
                $T = T - \{t\}$;
            end if
            if $T = \Phi$ then
                found = true;
                break;
            end if
        end for
    end while
    add $mXHR3_{sr}$ to $\{mXHR3_{sr}\}$;
end for

/* Remove all sets $M$ from $\{mXHR3_{sr}\}$ if there exists $M' \in \{mXHR3_{sr}\}$ such that $M' \subset M$. */
for all $M \in \{mXHR3_{sr}\}$ do
    for all $M' \in \{mXHR3_{sr}\}$ do
        if $M' \subset M$ then
            remove $M$ from $\{mXHR3_{sr}\}$;
            break;
        end if
    end for
end for

return $\{mXHR3_{sr}\}$;
```

6. $\{MXHR3_{sr}\}$ includes the set(s) in $\{mXHR3_{sr}\}$ with the smallest cardinality. During the process of constructing $\{MXHR3_{sr}\}$, we can ignore the minimal extended hidden nodes set whose cardinality already exceeds the achieved minimum

value, which becomes our incentive to improve the algorithm. The calculation of $\{MXHR3_{sr}\}$ is described in Algorithm 2.

Algorithm 2 (*Calculation of* $\{MXHR3_{sr}\}$).

```
$\{MXHR3_{sr}\} = \Phi$;
List the complete permutation of $XHR3_{sr}$, call it $P$.
$min\_cardinality = \mathbf{MAX}$;

/* Find $mXHR3_{sr}$ for each permutation $P_i \in P$. */
for all permutations $P_i \in P$ do
    $mXHR3_{sr} = \Phi$;
    $T = XH3_{sr}$;
    $found =$ false;
    while $P_i \neq \Phi \bigwedge found =$ false $\bigwedge |mXHR3_{sr}| < min\_cardinality$ do
        remove the next node $p$ from $P_i$, $P_i = P_i - \{p\}$;
        $mXHR3_{sr} = mXHR3_{sr} \cup \{p\}$;
        for all nodes $t \in T$ do
            if $\mathcal{R}(p,t)$ then
                $T = T - \{t\}$;
            end if
            if $T = \Phi$ then
                $found =$ true;
                break;
            end if
        end for
    end while

/* if $|mXHR3_{sr}|$ is less than the current achieved minimum cardinality, update $min\_cardinality$ and remove all elements from $\{MXHR3_{sr}\}$. */
    if $found =$ true then
        if $|mXHR3_{sr}| < min\_cardinality$ then
            $min\_cardinality = |mXHR3_{sr}|$;
            $\{MXHR3_{sr}\} = \Phi$;
        end if
        add $mXHR3_{sr}$ to $\{MXHR3_{sr}\}$;
    end if
end for

return $\{MXHR3_{sr}\}$;
```

3.3. Accuracy metrics for node classification

We introduce a set of metrics characterizing the ability of a MAC protocol to silence nodes which could cause collisions. Ideally, an algorithm should silence all nodes that have the potential to be hidden nodes, as well as nodes that could potentially be affected by the transmission T_{sr}. Assume there exists an algorithm $\mathcal{I}$ which constructs the set of all the nodes that should be silenced during a transmission T_{sr}:

$$\mathcal{S}_{sr}(\mathcal{I}) = H_{sr} \cup H_{rs} \cup V_s \cup V_r. \tag{9}$$

In practice, the set of nodes silenced by an algorithm $\mathcal{F}$, $\mathcal{S}_{sr}(\mathcal{F})$, might contain nodes that are silenced unnecessarily (misclassified) and might lack nodes which should have been silenced (missed nodes).

Call $Misc_{sr}(\mathcal{F})$ the *misclassification ratio* of an algorithm $\mathcal{F}$ for a transmission T_{sr}. $Misc_{sr}(\mathcal{F})$ measures the ratio of nodes that are incorrectly silenced by $\mathcal{F}$

$$Misc_{sr}(\mathcal{F}) = \frac{|\mathcal{S}_{sr}(\mathcal{F}) - \mathcal{S}_{sr}(\mathcal{I})|}{|\mathcal{S}_{sr}(\mathcal{I})|}. \tag{10}$$

Call $Miss_{sr}(\mathcal{F})$ the *miss ratio* of an algorithm $\mathcal{F}$ for a transmission T_{sr}. $Miss_{sr}(\mathcal{F})$ measures the ratio of nodes which are not silenced by the algorithm $\mathcal{F}$, although they should have been

$$Miss_{sr}(\mathcal{F}) = \frac{|\mathcal{S}_{sr}(\mathcal{I}) - \mathcal{S}_{sr}(\mathcal{F})|}{|\mathcal{S}_{sr}(\mathcal{I})|}. \tag{11}$$

Let $\overline{Misc(\mathcal{F})}$ and $\overline{Miss(\mathcal{F})}$ be the *average misclassification* ratio and *average miss ratio* of an algorithm $\mathcal{F}$, respectively. The averages are computed over a network $\mathcal{N}$

$$\overline{Misc(\mathcal{F})} = \frac{\sum_{\forall_{s,r\in\mathcal{N}}\mathcal{R}(s,r)} |\mathcal{S}_{sr}(\mathcal{F}) - \mathcal{S}_{sr}(\mathcal{I})|}{\sum_{\forall_{s,r\in\mathcal{N}}\mathcal{R}(s,r)} |\mathcal{S}_{sr}(\mathcal{I})|}, \tag{12}$$

and

$$\overline{Miss(\mathcal{F})} = \frac{\sum_{\forall_{s,r\in\mathcal{N}}\mathcal{R}(s,r)} |\mathcal{S}_{sr}(\mathcal{I}) - \mathcal{S}_{sr}(\mathcal{F})|}{\sum_{\forall_{s,r\in\mathcal{N}}\mathcal{R}(s,r)} |\mathcal{S}_{sr}(\mathcal{I})|}. \tag{13}$$

3.4. A solution to the hidden node problem

In a wireless ad hoc network with asymmetric links, the sender may not be able to receive the CTS or ACK packets from the receiver. In such a case a DATA packet, or the next frame cannot be sent. The IEEE 802.11 protocol assumes that all the connections are symmetric. Our protocol relaxes this assumption, asymmetric links can be used provided that they are part of a *three-party proxy set* [19].

Our protocol retains the use of RTS, CTS, DATA and ACK frames defined in IEEE 802.11 standard. In addition, we introduce four new frames: XRTS (Extended RTS), XCTS (Extended CTS), TCTS (Tunneled CTS), and TACK (Tunneled ACK).

An ideal MAC layer protocol should be based upon a scheme that delivers the RTS and CTS packets to all hidden nodes in H_{rs} and H_{sr}, respectively. However, such a scheme can be impractical because (i) a node may not have knowledge of all its In-bound neighbors; (ii) the number of hops needed to reach an In-bound neighbor might be large, thus the time penalty and the power consumption required for the RTS/CTS diffusion might outweigh the benefits of a reduced probability of collision.

Our solution is to send RTS and CTS packets to the nodes in $H3_{rs}$ and $H3_{sr}$ respectively. In this way, a considerable number of nodes that are misclassified as "hidden" nodes by [15], referred to as protocol A, and [6], referred to as protocol B, are allowed to transmit. Note that our approach does not identify all hidden nodes, but neither methods A or B are able to identify all hidden nodes.

3.5. Node status

In IEEE 802.11, when a node overhears a RTS or a CTS packet, it becomes *silent* and cannot send any packet until its NAV expires. This way, nodes in the relay set cannot send XRTS/XCTS as they should be in a *silent* state after overhearing the RTS/CTS packet. To resolve this dilemma, we replace the *silent* state with a *quasi silent* state, in which a node is allowed to send control packets, except RTS and CTS.

The medium access control model proposed in this paper classifies a node as either *idle*, *active*, *quasi silent*, or *silent*. When a node is idle, it is able to send or receive any type of packets. When a node is active, it is either sending or receiving a packet. When a node is in the quasi silent state, it can either receive packets or send any packet type except RTS, CTS, or DATA. When a node is in the silent state, the node can receive packets but cannot send any packet.

3.6. Medium access control model

The medium access control (MAC) model of our protocol is based upon an extended four-way handshake (Fig. 3). For short data frames, there is no need to initiate a RTS/CTS handshake (see Fig. 4a and b). For long data frames, we recognize several phases (see Fig. 4c and d):

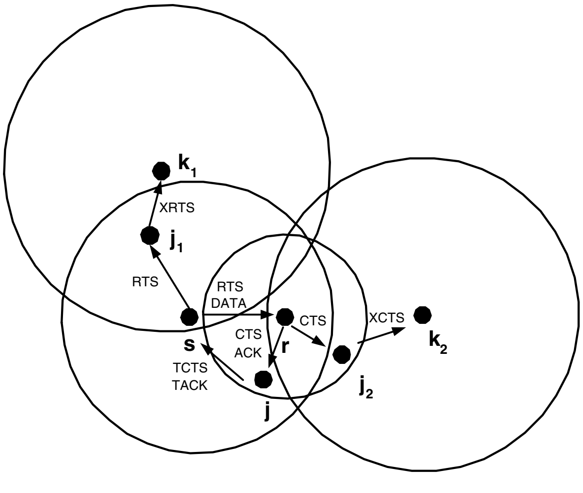


Fig. 3. Routing over asymmetric links in a heterogeneous wireless ad hoc network. Node s is the sender, r is the receiver, the link from node s to r is asymmetric, and node j is the proxy node that can relay traffic to s for r. Nodes k_1 and k_2 are hidden nodes for transmissions T_{rs} and T_{sr}, respectively. Nodes j_1 and j_2 are the proxy nodes that can relay traffic from s to k_1 and from r to k_2, respectively.

1. Sensing. The sender s senses the medium. If it does not detect any traffic for a DIFS period, the sender starts the contention phase; otherwise, it backs off for a random time before it senses again.
2. Contention. The sender s generates a random $\gamma \in$ [0, contention window] slot time. The sender s starts a transmission if it does not detect any traffic for γ time.
3. RTS transmission. The sender s sends a RTS packet to the receiver r. The RTS packet specifies the NAV(RTS), *link type* of L_{sr} and $MXHR3_{rs}$. The *link type* field is used to determine whether symmetric or asymmetric medium access control model is used.
4. CTS transmission. The receiver r checks whether the link is symmetric or not. If link L_{sr} is symmetric, node r sends a CTS packet back to node s; otherwise, node r sends a TCTS packet to node s. A TCTS packet specifies both the proxy node and the receiver r. The proxy node forwards the TCTS packet to the original sender s after receiving it. A CTS/TCTS packet can be sent only after sensing a free SIFS period. Instead of $MXHR3_{sr}$, $MXHR3_{rs} - MXHR3_{sr}$ is specified in the CTS/TCTS packet so that every extended hidden node relay is included only once. Thus, the duration of XCTS/XRTS diffusion phase can be reduced.

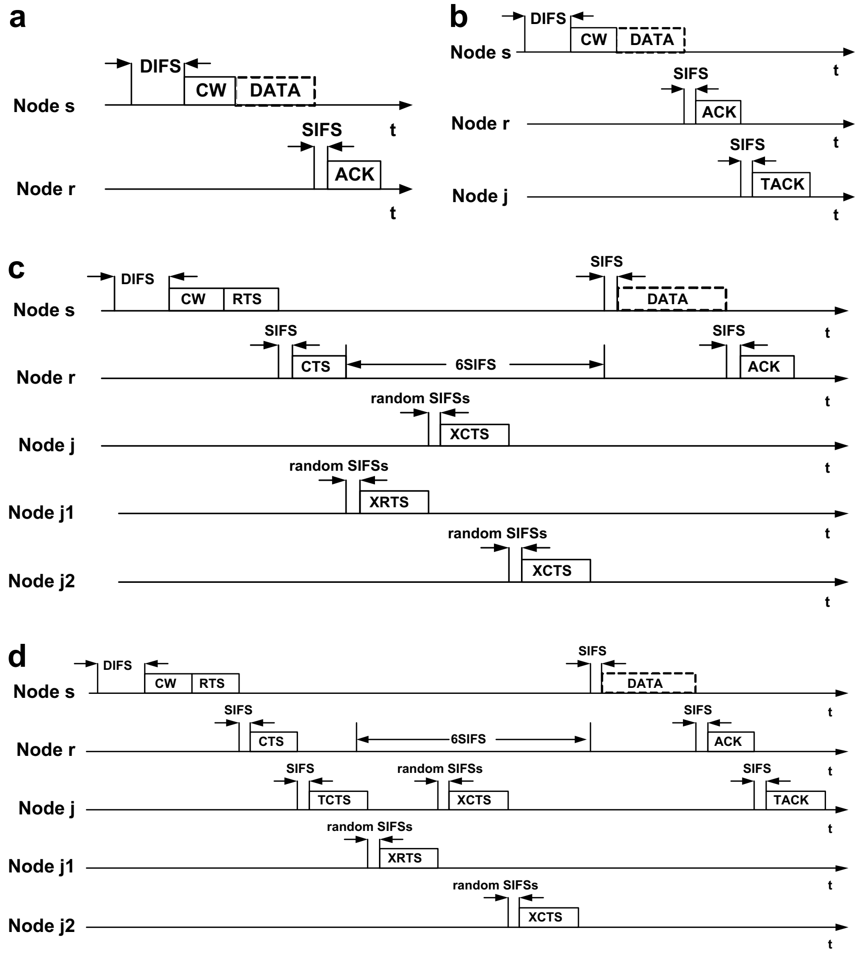


Fig. 4. The medium access control model for the MAC layer protocol introduced in this paper for the scenario in Fig. 3. (a) The medium access control model of proposed MAC protocol for short data frames over a symmetric link. (b) The medium access control model of proposed MAC protocol for short data frames over an asymmetric link. (c) The medium access control model of proposed MAC protocol for long data frames over a symmetric link. (d) The medium access control model of proposed MAC protocol for long data frames over an asymmetric link.

5. XRTS/XCTS diffusion. All nodes that overhear a RTS/CTS/TCTS packet enters a *quasi silent* state. After the CTS transmission phase, all extended hidden node relays that are either specified in RTS or CTS/TCTS starts contention for broadcasting XRTS/XCTS to its neighbors. When a node captures the medium, all other nodes back-off for a random number of (1,...,4) SIFS periods, and continue the contention until the XRTS/XCTS diffusion phase finishes. An XRTS/XCTS diffusion phase lasts for 6 SIFS periods, after which all nodes except the proxy node become *silent*.
6. Data transmission. When the XRTS/XCTS diffusion phase finishes, the sender *s* starts sending DATA packets to the receiver *r* after sensing a free SIFS period.
7. Acknowledgement. Once the receiver *r* successfully received the DATA packet from the sender *s*, it replies with an ACK if link L_{sr} is symmetric, or a TACK packet if link L_{sr} is asymmetric. An ACK/TACK packet can be sent only after sensing a free SIFS period. When the sender *s* receives an ACK/TACK packet, it starts contending the medium for the next frame. Meanwhile, the NAVs that are reserved for this transmission should expire.

At any moment, if a node overhears a packet containing new NAV information, it compares it with the currently stored NAV, and retains the NAV which expires later.

4. Simulation and case study

We have implemented the AsyMAC protocol in NS-2 [3,18], an object-oriented event-driven simulator developed at the Lawrence Berkeley National Laboratory, with the CMU wireless extensions [13]. As AsyMAC requires a routing protocol able to handle asymmetric links, we paired it with A^{4}LP routing protocol to form a complete ad hoc networking stack. In our experiments, we compare the A^{4}LP/AsyMAC pair against the standard IEEE 802.11 protocol coupled with AODV [14], a widely used on-demand ad hoc routing protocol and the more recent OLSR [5] protocol.

The simulation results reflect the performance of the pair of the corresponding MAC and routing protocols rather than the performance of the MAC or routing protocols alone. We had chosen this experimental setup because it provides the most informative comparison of real scenarios. We cannot run a routing protocol which does not support asymmetric links on top of AsyMAC. On the other hand, A^{4}LP can be run on top of MAC protocols which do not support asymmetric links. However, A^{4}LP has a higher overhead than routing protocols which assume symmetric connections, thus an A^{4}LP/802.11 combination would always perform somewhat worse than combination such as OLSR/802.11, because we can take advantage of the existence of asymmetric links only if they are supported throughout the stack. Thus, the only reasonable choices are to use either all symmetric protocols or all asymmetric-link aware ones in the full stack.

A possible study would involve the comparison of between asymmetric stacks, by substituting for AsyMAC the protocols described by [6,15]. In Section 4.3, we have implemented the core decision algorithms of these protocols for a comparison of the classification accuracy. However, there is no publicly available NS-2 implementation of these protocols, and a fully functional implementation of these protocols is beyond the scope of this paper.

First, we analyze the benefits of algorithms able to take advantage of asymmetric links in the maintaining the connectivity of a network. Through the study of a specific scenario, we show that a protocol stack composed by AsyMAC and the A^{4}LP routing protocol is able to maintain connectivity where the standard IEEE 802.11 MAC protocol coupled with AODV or OLSR loose connectivity.

Second, we perform a simulation study in which we measure the performance of the A^{4}LP/AsyMAC stack against the AODV/802.11 and OLSR/802.11 stacks in a series of randomized mobile ad hoc network scenarios with realistic traffic source patterns.

Finally, we compare AsyMAC against two previously proposed asymmetric MAC protocols in terms of the accuracy of the hidden node classification.

4.1. A connectivity scenario

In this section, we briefly discuss an example when A^{4}LP/AsyMAC uses asymmetric links to route packets from each pair of nodes while both AODV/802.11 and OLSR/802.11 fail to route packets. The connectivity scenario is given in Fig. 5. The initial position of nodes is depicted in the graph (a), which shows also the transmission range and the distance between the nodes. The graph (b) is a logical view of the above scenario. The nodes do not move during the simulation. The forward and reverse routes are found and established by A^{4}LP,

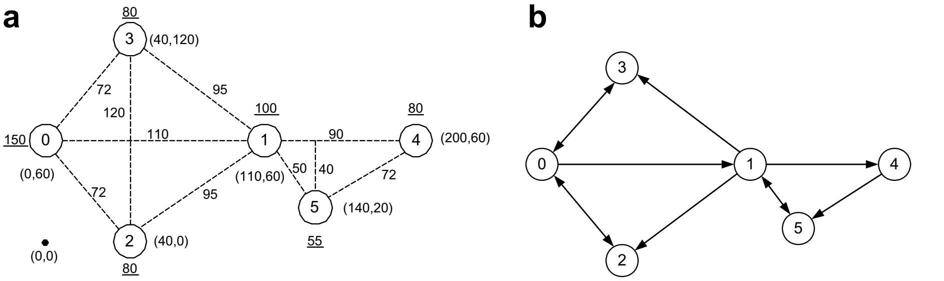


Fig. 5. (a) The physical topology of the network, where node 0 and 4 are exchanging packets. The numbers next to the nodes indicate the position in the (x,y) format and the transmission range (underlined). The numbers on the links represent the distance between the nodes. (b) The logical topology of a network.

and MAC layer acknowledgements are assured by AsyMAC. For instance, node 5 is a proxy node that forwards CTS and ACK packets for a unidirectional transmission from node 1 to node 4 at MAC layer. In this scenario, the two far-most nodes 0 and 4 are exchanging packets. The packets are successfully delivered and acknowledged by A^{4}LP/AsyMAC, while all packets are lost by AODV/802.11 or OLSR/802.11 during the transmission.

4.2. A study of alternative protocol stacks in a mobile ad hoc network

The previous scenario illustrates the case when the A^{4}LP/AsyMAC protocol maintained connectivity, while the AODV/802.11 and OLSR/802.11 stacks did not. However, these extreme cases might be relatively rare. In the following, we compare these protocol stacks in a series of simulations involving an ad hoc network with mobile nodes in a more realistic setup. To describe the movement of nodes in the system, we use the "random waypoint" model [4]. Each node randomly picks a destination on the map, moves to the destination at a *constant speed*, and then pauses for certain time, the *pause time*. After the pause time, it continues the movement following the same pattern. The nodes are classified into four classes C1, C2, C3 and C4 with different transmission ranges.

The traffic patterns are generated by *constant bit rate* (CBR) sources sending UDP packets. Each CBR source resides at one node and generates packets for another node. Each CBR source is active for a time interval called *CBR duration*. Our simulation allows a *setup time* to allow nodes gather certain routing information before generating any traffic. After the *setup time*, the simulation time is divided into equal time slices, called *switching intervals*. During each switching interval, we generate CBR sources for different pairs of senders and receivers. Table 1 illustrates the default settings and the range of the parameters for our simulation experiments.

To construct 95% confidence intervals, each experiment was repeated 20 times for a pair of scenario and traffic pattern, the two elements affecting the results of a performance study. This involves 200 individual runs for the each of the three studies. The average simulation time for a single experiment was about 3 h, for a total of 1800 h of computer time. By observing the evolution of the average values and the calculated confidence intervals after 5, 10 and 20 repetitions, we notice that at 20 repetitions the values reach quiescence, and future repetitions would provide only insignificant changes on the overall shape of the graphs.

Table 1
The default values and the range of the parameters for our simulation studies

Field	Value	Range
Simulation area	500×500 (m^2)	
Number of nodes	8(C1), 16(C2), 24(C3), 32(C4)	30–110
Ratio of nodes	C1:C2:C3:C4 = 1:2:3:4	
Transmission ranges	200(C1),150(C2), 100(C3),50(C4) (m)	
Speed	1 (m/s)	1–10 (m/s)
Pause time	15 (s)	
Simulation time	300 (s)	
Setup time	20 (s)	
Switching interval	10 (s)	
Number of CBR sources	10	4–40
CBR packet size	64 (bytes)	
CBR sending rate	512 (bps)	
CBR duration	5 (s)	

We are concerned with the impact of node mobility, network load, and network density upon power consumption, packet loss ratio, and latency. For each randomly generated scenario and traffic patterns, we run simulation experiments covering AODV with IEEE 802.11, OLSR with IEEE 802.11, A^{4}LP using 3-limited forwarding with distance metric (A^{4}LP-M3-F1/AsyMAC) with AsyMAC, and A^{4}LP using 3-limited forwarding with the metric proposed in [19] (A^{4}LP-M3-F2) with AsyMAC.

4.2.1. The influence of network load

The effect of the network load upon the packet loss ratio for two standard protocol stacks AODV/IEEE 802.11, OLSR/IEEE 802.11 and for A^{4}LP-M3-F1/AsyMAC and A^{4}LP-M3-F2/AsyMAC is summarized by the graphs in Fig. 6. The ratio of the packets lost by AODV/802.11 is roughly twice the rate of the packets lost by the other protocols. The major reason is that flooding, an inefficient broadcast solution, is used in AODV/802.11 for finding a route. Among the other protocols, A^{4}LP-M3-F2/AsyMAC performs the best, followed by OLSR/802.11, which delivers more packets than A^{4}LP-M3-F1/AsyMAC for similar scenarios and traffic patterns. OLSR/802.11 is able to deliver packets only via symmetric links, thus packets are dropped if at least one asymmetric link is on the *critical* path; however, A^{4}LP/AsyMAC is able to

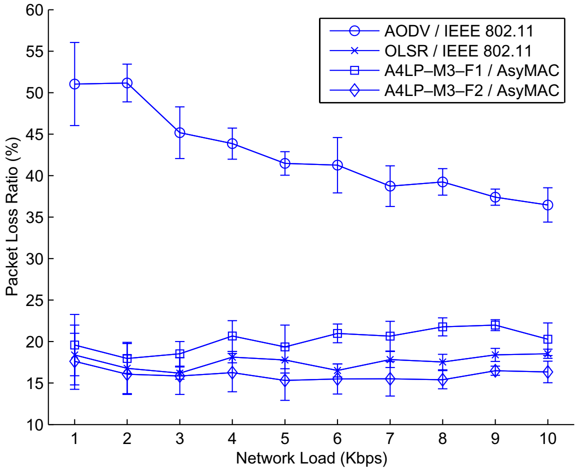


Fig. 6. Packet loss ratio versus network load. The ratio of packets lost by AODV/802.11 is roughly twice the ratio of packets lost by the other protocols. Among the other protocols, A^{4}LP-M3-F2/AsyMAC performs the best, followed by OLSR/802.11, which delivers more packets than A^{4}LP-M3-F1/AsyMAC for similar scenarios and traffic patterns.

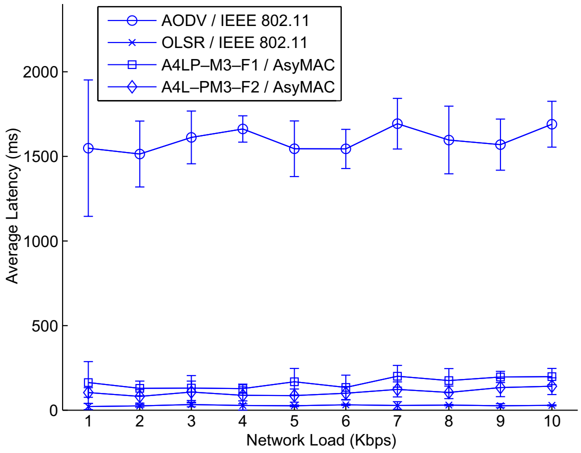


Fig. 7. Average latency versus network load. The average latency of AODV/802.11 is much higher than the other protocols. Among the other protocols, OLSR/802.11 has the shortest latency.

deliver those packets. Our experiment also shows the metric we proposed in [19] (A^{4}LP-M3-F2), a combined metric with distance, power level and class information, provides better performance than the distance only metric (A^{4}LP-M3-F1) in heterogeneous mobile ad hoc networks.

In our study, the measured values have relatively large confidence intervals, and most of these confidence intervals overlap. This means that we do not have 95% confidence that for any particular experimental instance the given protocol will perform better than the other protocol. Indeed, if there are no (or very few) asymmetric links, the symmetric protocols will likely outperform the asymmetric ones, due to the higher overhead of the asymmetric protocol. Unfortunately, the range of the measurable values for metrics such as packet loss is very wide – in some scenarios there might be no packet loss, in other ones, many of packets are lost. This variability is reflected in relatively large confidence intervals. We believe that often when the average value of packet loss is lower for one of the protocols, the protocol will perform *in average* better than the other ones.

The effect of the network load upon the average latency for two standard protocol stacks AODV/IEEE 802.11, OLSR/IEEE 802.11 and for A^{4}LP-M3-F1/AsyMAC and A^{4}LP-M3-F2/AsyMAC is summarized by the graphs in Fig. 7. The average latency of AODV/802.11 is much higher than that of the other protocols. AODV is a reactive protocol which finds routes only when needed. A^{4}LP is a hybrid protocol, routes to non-neighbors are still discovered when needed, however, routes to certain In-, Out-, and In/Out-bound neighbors are maintained proactively in a routing table; this fact contributes to the reduction of the average packet delivery latency.

OLSR/802.11 has the lowest average packet delivery latency, followed by A^{4}LP-M3-F2/AsyMAC, and A^{4}LP-M3-F1/AsyMAC. Note, however, that the average packet delivery latency is based only on the delivered packets. OLSR/802.11 drops more packets than A^{4}LP-M3-F2/AsyMAC; these are the packets which require a protocol able to deal with asymmetric links. The packets that could be delivered by A^{4}LP-M3-F2/AsyMAC but not by OLSR/802.11 generally have higher latency, and this could explain why the average packet delivery latency of A^{4}LP-M3-F2/AsyMAC is higher than that of OLSR/802.11.

4.2.2. The influence of network mobility

The average packet loss ratio versus node mobility is summarized in Fig. 8. With the network mobility increasing, the performances of A^{4}LP-M3-F2/AsyMAC and OLSR/802.11 are degraded, while the performance of AODV/802.11 fluctuates between 35% and 45%. AODV/802.11 performs the worst in case of ad hoc networks with low mobility, but it outperforms the other protocols for highly mobile ad hoc networks. The reason for

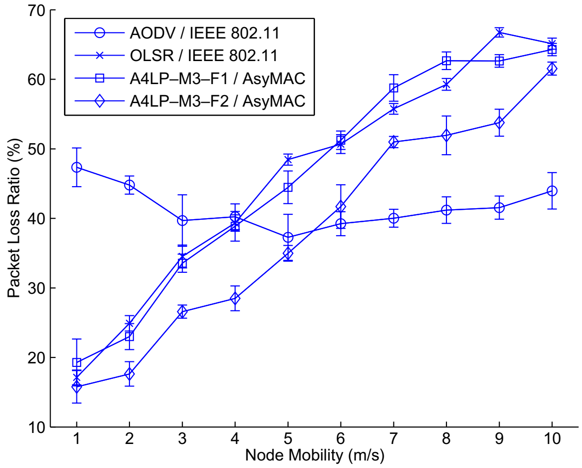


Fig. 8. Packet loss ratio versus node mobility. With the network mobility increasing, the performances of A^{4}LP-M3-F2 and OLSR/802.11 are degraded while the performance of AODV/802.11 fluctuates between 35% and 45%. AODV/802.11 performs the worst in case of ad hoc networks with low mobility, but it outperforms the other protocols for highly mobile ad hoc networks.

this is that for ad hoc networks with relatively high mobility, cached routes and neighbor information becomes stale rapidly, which degrades the performance of proactive (OLSR) or hybrid (A^{4}LP) protocols but not reactive (AODV) protocols. However, A^{4}LP-M3-F2/AsyMAC always outperforms OLSR/802.11 and A^{4}LP-M3-F1/AsyMAC at any network mobility in terms of packet loss ratio.

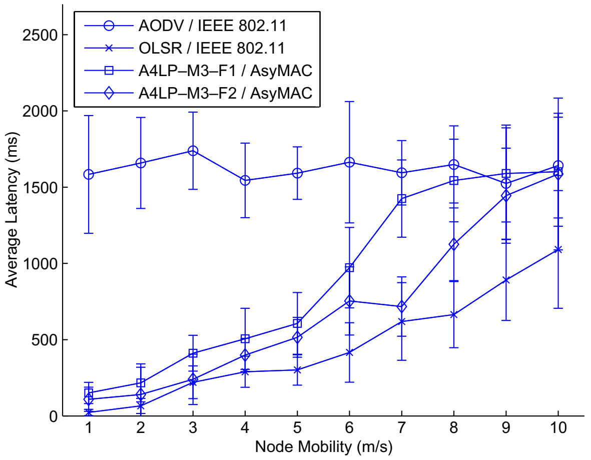


Fig. 9. Average latency versus node mobility. The average latency of AODV/802.11 is much higher than the other protocols that perform similarly.

Fig. 9 presents average packet delivery latency versus network mobility. AODV, which is an on-demand protocol, shows about the same, relatively long, latency irrespective of the mobility of the nodes. For A^{4}LP/AsyMAC and OLSR/802.11 the latency is increasing with the mobility, as the protocols need additional overhead to keep their topology information up-to-date. At the mobility of about 10 m/s, AODV/802.11, A^{4}LP-M3-F2/AsyMAC and A^{4}LP-M3-F1/AsyMAC show about the

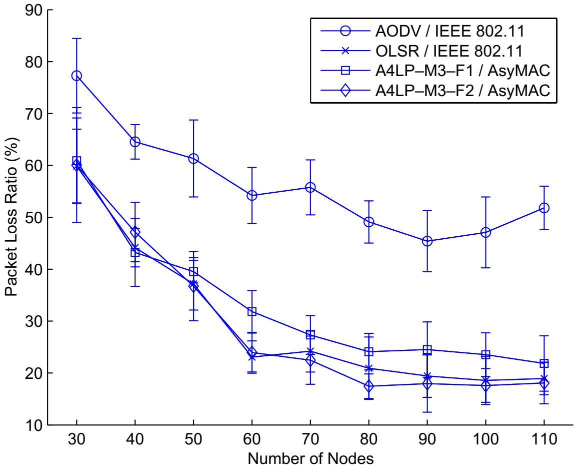


Fig. 10. Packet loss ratio versus number of nodes. A^{4}LP-M3-F2/AsyMAC delivers most packets, followed by OLSR/802.11, A^{4}LP-M3-F1/AsyMAC and AODV/802.11 for similar scenarios and traffic patterns. The packet loss ratio decreases when the number of nodes increases.

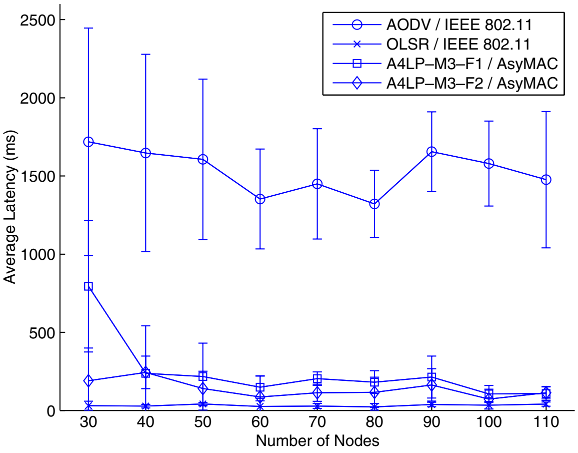


Fig. 11. Average latency versus number of nodes. The average latency of AODV/802.11 is much higher than the other protocols. The packet latency tends to decrease as the number of nodes increases for A^{4}LP/AsyMAC and OLSR/802.11.

same latency. In these tests, OLSR/802.11 outperforms A^{4}LP/AsyMAC because the amount of topology data it needs to maintain is lower, being restricted to the symmetric links only. This latency advantage comes at the cost of ignoring asymmetric links and therefore, potentially disconnecting nodes which would maintain connectivity with the A^{4}LP/AsyMAC solution.

4.2.3. The influence of the number of nodes

In the following set of experiments, we vary the number of nodes moving in the measurement area. As the nodes have a limited range, when the number of nodes is too low, some nodes might loose connectivity.

Fig. 10 illustrates the packet loss ratio versus the number of nodes. For similar scenarios and traffic patterns, A^{4}LP-M3-F2/AsyMAC delivers most packets, followed by OLSR/802.11, A^{4}LP-M3-F1/AsyMAC, and AODV/802.11. As the number of nodes in the network increases, the network connectivity increases as well, thus the packet loss ratio decreases. Fig. 10 shows that the packet loss ratio decreases from roughly 40% to about 10% as the number of nodes increases from 30 to 110.

Fig. 11 shows the average packet delivery latency versus the number of nodes. The average latency of AODV/802.11 is much higher than the other protocols. For A^{4}LP/AsyMAC and OLSR/802.11 the packet latency tends to decrease as the number of nodes increases. As the number of nodes in the network increases, more neighbors and routes are found during the neighbor information exchange process, thus the packet delivery latency decreases.

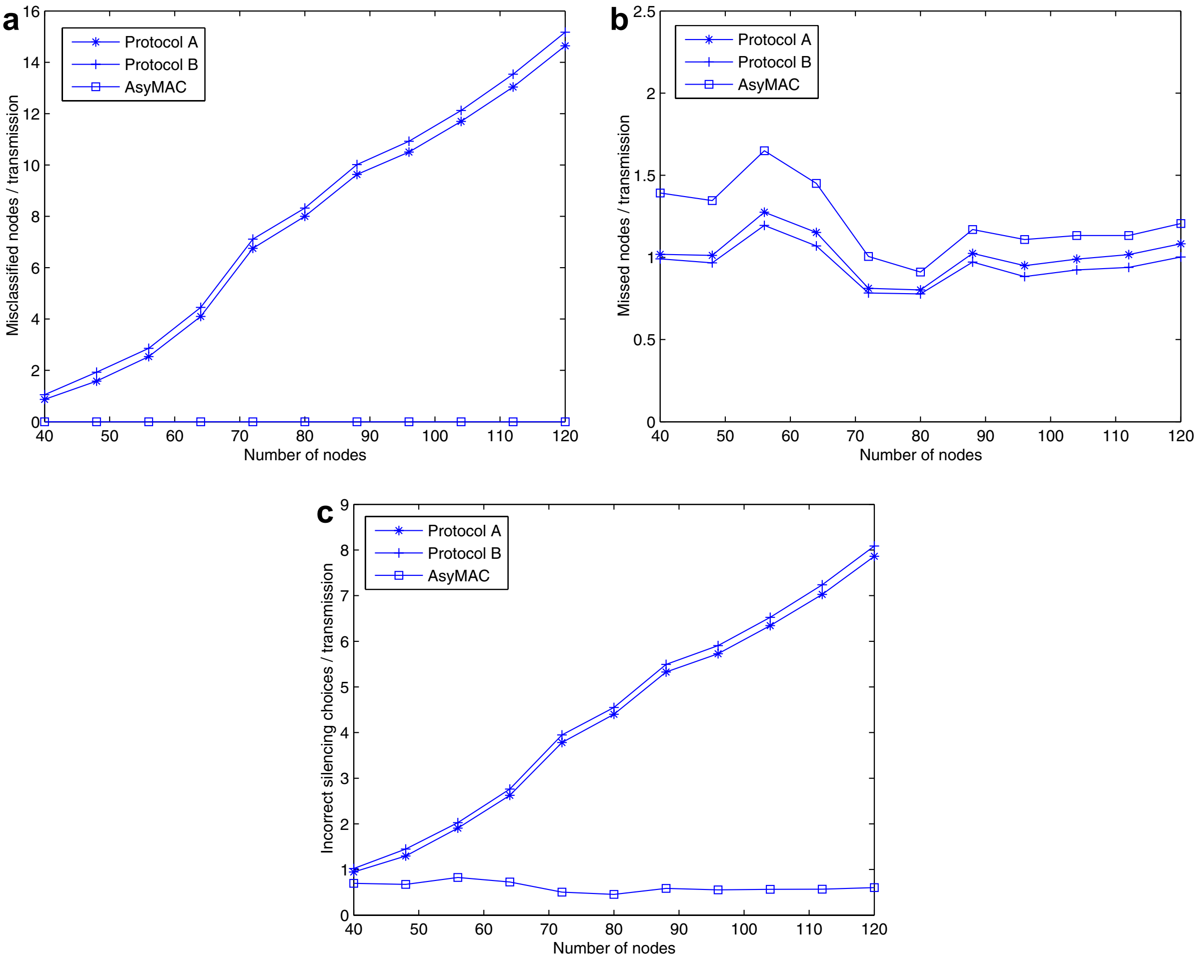


Fig. 12. (a) The average misclassified nodes/transmission as a function of the number of nodes. The AsyMAC protocol does not misclassify nodes in a static network. (b) The average missed nodes/transmission for protocols A, B, and our approach, as a function of the number of nodes. (c) The average number of incorrect silencing decisions per transmission for protocols A, B, and for our approach.

4.3. The accuracy of hidden node classification

A node is *misclassified* as hidden if it is silenced by the algorithm while it should not be silenced. Misclassification reducing bandwidth utilization because it leads to unnecessary silencing of nodes which could have been transmitting. A node is *missed* by the algorithm if it was not silenced although it should have been. Missed nodes lead to collisions. The more accurate is a protocol in classifying the nodes, the better the bandwidth utilization. A useful measure of the global performance of an algorithm is the number of incorrect silencing decisions per transmission – defined as the sum of misclassified and missed nodes.

We compare the accuracy of the classification of our proposed AsyMAC protocol with the accuracy of two well known protocols which are performing the same classification [6,15]. As a note, the basic IEEE 802.11 protocol does not perform any classification of nodes. The simulation environment is an area of 500 × 500 m. We populate our environment with a heterogeneous collection of nodes belonging to the four main classes of wireless nodes C1, C2, C3, and C4 (see [12,19]). The transmission ranges are normally distributed random variables with the mean 100, 75, 50, and 25 m, respectively and the standard deviations for each class is 5 m. The simulation scenarios are created using a set of 40–120 nodes including an equal number of nodes for each class, uniformly distributed in the area. For each generated scenario, we repeat the experiment 1000 times. The displacement of nodes are distributed around an initial position and the standard deviation is 20% of its transmission range.

The results of the simulation are shown in Fig. 12. The graph (a) shows the number of misclassified nodes per transmission. The AsyMAC algorithm does not misclassify nodes in a static network, because in the process of three-party proxy set formation, the nodes whose transmission range does not reach the current node are filtered out. However, misclassified nodes can appear with the AsyMAC protocol if the nodes are highly mobile and the current configuration does not reflect the one detected when the three-party proxy set was established. The graph (b) shows the missed nodes per transmission. Here the AsyMAC protocol performs worse than the other two protocols considered, as it is considering only the three-party proxy sets, and ignores possible higher order proxy sets. However, the number of missed nodes is very small for all the three protocols. Graph (c) shows the number of incorrect silencing decisions per transmission. Here, the AsyMAC protocol emerges with the lowest number of incorrect decisions, as its better performance at misclassification compensates for the lower performance in regards to missed nodes.

5. Conclusions

In this paper, we argue that asymmetry of the transmission ranges in wireless networks is a reality and should be treated as such. This asymmetry makes reliable communication more difficult and complicates medium access control, as well as network layer protocols.

The models of traditional multiple access networks assume that all nodes share a single communication channel and have access to the feedback (success, idle slot, collision) from any transmission. In this case, splitting algorithms allow sharing of the communication channel in a cooperative environment with reasonable efficiency and fairness. This is no longer the case for wireless networks with asymmetric or unidirectional links, where the sender and the receiver do not share the feedback channel and hidden nodes may interfere with a transmission.

In case of networks with asymmetric links, hidden nodes may be out of the reach of both the sender and the receiver, but their transmissions may interfere with the reception of a packet by the intended destination. The problem of hidden nodes is further complicated because the feedback from the receiver in an RTS/CTS exchange may have to pass through several relay stations before reaching all the nodes expected to be silent.

Some of the solutions proposed in the literature reduce the probability of a collision by requiring a larger than necessary set of nodes to be silent. In turn, this has negative effects upon the communication latency and the overall network throughput. We propose a MAC layer protocol, AsyMAC, which reduces the number of nodes that have to be silent but, as all the other schemes proposed, may miss some of the nodes which should have been classified as "hidden".

IEEE 802.11 assumes symmetric links between each pair of nodes while AsyMAC does not. For traffic over asymmetric links, AsyMAC relies on a proxy node in the three-party proxy set to relay acknowledgements back to the sender so that the reliability is assured. Our MAC protocol reduces average packet loss ratio and average packet delivery latency as asymmetric links are comprehensively utilized which dominate routing in heterogeneous ad hoc networks.

We conducted a simulation experiment using the NS-2 simulator and compared the performance of AODV/IEEE 802.11, OLSR/IEEE 802.11, A^{4}LP using 3-limited forwarding with distance metric (A^{4}LP-M3-F1) with AsyMAC, and A^{4}LP using 3-limited forwarding with the metric proposed in [19] (A^{4}LP-M3-F2) with AsyMAC. It is reported that A^{4}LP/AsyMAC performs much better than AODV/IEEE 802.11 in terms of average packet loss ratio and average packet delivery latency, in relatively stable ad hoc networks. A^{4}LP-M3-F2 with AsyMAC incurs a lower average packet loss ratio compared to OLSR/IEEE 802.11. Our simulation results also indicate that the fitness function proposed in [19] is better than the traditional distance function used in heterogeneous ad hoc networks.

Our future work is dedicated to remove the dependency of AsyMAC from A^{4}LP, and provide transparent interface to routing protocols so that it could be the underlying MAC protocol for any routing protocol in heterogeneous wireless ad hoc networks with asymmetric links.

Acknowledgments

The authors thank anonymous reviewers for their insightful comments which have helped improve the quality of this paper. The research reported in this paper was partially supported by National Science Foundation grants ACI0296035 and EIA0296179.

References

[1] L. Bao, J. Garcia-Luna-Aceves, Channel access scheduling in ad hoc networks with unidirectional links, in: Proceedings of Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM), 2001, pp. 9–18.

[2] V. Bharghavan, A. Demers, S. Shenker, L. Zhang, MACAW: A media access protocol for wireless LAN's, in: Proceedings of ACM SIGCOMM'94, 1994, pp. 221–225.

[3] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, H. Yu, Advances in network simulation, IEEE Computer 33 (5) (2000) 59–67.

[4] J. Broch, D.A. Maltz, D.B. Johnson, Y. Hu, J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: Proceedings of Mobile Computing and Networking, 1998, pp. 85–97.

[5] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, L. Viennot, Optimized link state routing protocol for ad hoc networks, in: Proceedings of IEEE INMIC, December, 2001, pp. 62–68.

[6] T. Fujii, M. Takahashi, M. Bandai, T. Udagawa, I. Sasase, An efficient MAC protocol in wireless ad-hoc networks with heterogeneous power nodes, in: The 5th International Symposium on Wireless Personal Multimedia Communications (WPMC'2002), Hawaii, vol. 2, 2002, pp. 776–780.

[7] C. Fullmer, J.J. Garcia-Luna-Aceves, Floor acquisition multiple access (FAMA) for packet-radio networks, in: Proceedings of ACM SIGCOMM'95, 1995, pp. 262–273.

[8] IEEE std 802.11b-1999. Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, August 1999.

[9] P. Karn, MACA – a new channel access method for packet radio, in: Proceedings of the 9th ARRL Computer Networking Conference, 1990, pp. 134–140.

[10] L. Kleinrock, F. Tobagi, Packet switching in radio channels: Part I – carrier sense multiple-access modes and their throughput-delay characteristics, IEEE Transactions on Communications COM-23 (12) (1975) 1400–1416.

[11] K. Xu, M. Gerla, S. Bae, Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks, Ad Hoc Networks Journal 1 (1) (2003) 107–123.

[12] D.C. Marinescu, G.M. Marinescu, Y. Ji, L. Bölöni, H. Siegel, Ad hoc grids: communication and computing in a power constrained environment, in: Proceedings of the Workshop on Energy-Efficient Wireless Communications and Networks (EWCN), 2003, pp. 113–122.

[13] CMU Monarch extensions to ns. URL: http://www.monarch.cs.cmu.edu.

[14] C. Perkins, E. Royer, Ad hoc on-demand distance vector routing, in: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 99–100.

[15] N. Poojary, S.V. Krishnamurthy, S. Dao, Medium access control in a network of ad hoc mobile nodes with heterogeneous power capabilities, in: Proceedings of IEEE ICC 2001, vol. 3, 2001, pp. 872–877.

[16] V. Ramasubramanian, R. Chandra, D. Mosse, Providing a bidirectional abstraction for unidirectional ad-hoc networks, in: Proceedings of INFOCOM 2002, vol. 3, 2002, pp. 1258–1267.

[17] V. Ramasubramanian, D. Mossé, Statistical analysis of connectivity in unidirectional ad hoc networks, in: Proceedings of the International Workshop on Ad Hoc Networking 2002, Vancouver, 2002, pp. 109–115.

[18] VINT project, The ucb/lbnl/vint network simulator-ns (version 2), URL: http://www.isi.edu/nsnam/ns.

[19] G. Wang, Y. Ji, D.C. Marinescu, D. Turgut, A routing protocol for power constrained networks with asymmetric links, in: Proceedings of the ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN), 2004, pp. 69–76.

[20] DARPA Advanced Technology Office – FCS Communications program, URL: http://www.darpa.mil/ato/programs/fcs_comm.htm.

[21] Joint Tactical Radio System (JTRS), URL: http://jtrs.army.mil.

Guoqiang Wang received a B.S. degree from Southeast University, Nanjing, China, in 2001. He is currently a Ph.D. Candidate in the School of Electrical Engineering and Computer Science at the University of Central Florida, Orlando.

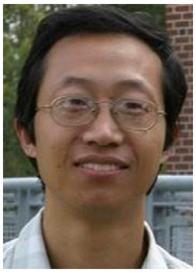
His research areas include ad hoc routing, grid computing, and parallel simulation.

Damla Turgut is an Assistant Professor with the School of Electrical Engineering and Computer Science at University of Central Florida.

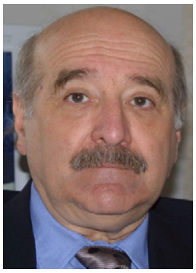
She received her B.S., M.S., and Ph.D. degrees from the Computer Science and Engineering Department of University of Texas at Arlington in 1994, 1996, and 2002, respectively. She has been included in the WHO's WHO among students in American Universities and Colleges in 2002. She has been awarded outstanding research award and has been recipient of the Texas Telecommunication Engineering Consortium (TxTEC) fellowship. She is a member of IEEE, member of the ACM, and the Upsilon Pi Epsilon honorary society. Her research interests include wireless networking, mobile computing, distributed systems, agents, and databases.

Ladislau Bölöni is an Assistant Professor with the School of Electrical Engineering and Computer Science at University of Central Florida.

He received a Ph.D. degree from the Computer Sciences Department of Purdue University in May 2000. He received a Master of Science degree from the Computer Sciences department of Purdue University in 1999 and Diploma Engineer degree in Computer Engineering with Honors from the Technical University of Cluj-Napoca, Romania in 1993. He received a fellowship from the Computer and Automation Research Institute of the Hungarian Academy of Sciences for the 1994–95 academic year. He is a senior member of IEEE, member of the ACM, AAAI and the Upsilon Pi Epsilon honorary society. His research interests include autonomous agents, grid computing and wireless networking.



Yongchang Ji received the M.S. and Ph.D. degrees in computer science from University of Science and Technology of China, Heifei, in 1996 and 1998, respectively. He was a Postdoctoral Researcher of computer sciences at Purdue University, West Lafayette, IN. He is currently a Postdoctoral Research Associate of computer science at University of Central Florida, Orlando. He has published more than 30 papers in professional journals and referred conference proceedings. His main research interests include high-performance computing, Grid computing, computational biology, parallel and distributed architecture, model, algorithm, and scalability.



Dan C. Marinescu joined the Computer Science Department at University of Central Florida in August 2001 as Professor of Computer Science. He has been an Associate and then Full Professor of Computer Science at Purdue University, in West Lafayette, Indiana, since 1984. He is the Scientific Director of the I2.Lab, an organization supporting interdisciplinary research in computer and information sciences (http://I2lab.ucf.edu) at UCF.

He is conducting research in parallel and distributed systems, computational biology, quantum computing, parallel and distributed computing, and has published more than 180 papers in journals and referred conference proceedings in these areas. He is the author of "Internet-Based Workflow Management: Towards a Semantic Web", published by Wiley in 2002 and has co-edited "Process Coordination and Ubiquitous Computing". The book "Approaching Quantum Computing", co-authored with Gabriela M. Marinescu was published in September 2004 by Prentice-Hall.