Title
Improving Route Discovery in On-Demand Routing Protocols Using Two-Hop Connected Dominating Sets

Permalink
https://escholarship.org/uc/item/8dv299xr

Author
Garcia-Luna-Aceves, J.J.

Publication Date
2006

Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Ad Hoc Networks 4 (2006) 509–531
Ad Hoc Networks
www.elsevier.com/locate/adhoc

Improving route discovery in on-demand routing protocols using two-hop connected dominating sets

Marco Aurélio Spohn [a,*], J.J. Garcia-Luna-Aceves [b,c]

[a] Department of Computer Science, University of California at Santa Cruz, Santa Cruz, CA 95064, USA
[b] Department of Computer Engineering, University of California at Santa Cruz, Santa Cruz, CA 95064, USA
[c] Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304, USA

Received 28 July 2004; received in revised form 25 January 2005; accepted 22 April 2005
Available online 5 July 2005

Abstract

Many signaling or data forwarding operations involve the broadcasting of packets, which incurs considerable collisions in ad hoc networks based on a contention-based channel access protocol. We propose the *Three-hop Horizon Pruning* (THP) algorithm to compute *two-hop connected dominating set* (TCDS) using only local topology information (i.e., two-hop neighborhood). Because every node has the two-hop neighborhood information, it is possible to maintain fresh routes to all nodes within two hops. In this situation, a TCDS is ideal for the propagation of *route request* (RREQ) messages in the route discovery process of on-demand routing protocols. THP is shown to be more efficient than all prior distributed broadcasting mechanisms, when a TCDS is preferred over a *connected dominating sets* (CDS). Like all other algorithms that depend on local topology information, THP is not reliable when the topology changes frequently, and there is a clear trade-off between reliability and efficiency. We describe and analyze two enhancements to THP that address the lack of reliability of neighbor information. First we adopt a *virtual radio range* (VR), shorter than the physical *radio range* (RR), and consider as one-hop neighbors only those nodes within VR (we do not use two different radio ranges, as in prior work, because it can incur additional interference). The gap between VR and RR works as a buffer zone, in which nodes can move without loss of connectivity. Second, upon receiving a broadcast packet, the forwarder list in the packet header is analyzed together with the current information about the local neighborhood. Based on that, a node may decide to broadcast the packet even though it has not been selected as a forwarder. We conduct extensive simulations and show that AODV-THP with these two enhancements attains better performance than AODV in terms of delivery ratio, control overhead, packet collisions, and end-to-end delay.

Keywords: Route discovery; On-demand routing protocol; Dominating sets; Two-hop connected dominating sets; Broadcasting

* Corresponding author.
E-mail addresses: maspohn@cse.ucsc.edu (M.A. Spohn), jj@cse.ucsc.edu (J.J. Garcia-Luna-Aceves).

1570-8705/$ - see front matter
doi:10.1016/j.adhoc.2005.04.001

1. Introduction

Broadcasting operations involve the transmission of the same packet to many receivers or the entire network. However, achieving packet broadcasting using blind flooding[1] in a network using a contention-based medium access control (MAC) protocol can induce excessive contention. This effect is called the *broadcast storm* problem [1]. To reduce the impact of broadcasting signaling or data packets, the resulting mechanism must reduce the number of nodes that must attempt to forward broadcast packets, adapt to the dynamics of a mobile ad hoc network (MANET), and run in real time with only limited knowledge of the network topology.

We introduce the *three-hop horizon pruning* (THP) algorithm to make broadcast operations more efficient in ad hoc networks using contention-based MAC protocols. THP builds a *two-hop connected dominating set* (TCDS) of the network, which is a set of nodes such that every node in the network is within *two* hops from some node in the dominating set. Efficiency of broadcast operations is attained by implementing forwarding schemes that take advantage of a TCDS. More specifically, every node provides its one-hop neighbors with a list specifying one or more tuples, each with the identifier of a one-hop neighbor and a bit indicating if that neighbor dominates *any* two-hop neighbor. To forward a broadcast packet, a node tries to obtain the smallest subset of *forwarders*, which are one-hop neighbors that use some of the node's two-hop neighbors to reach *any* node that is three hops away. After such a selection of forwarders, the node broadcasts its packet with a header specifying its forwarder list, and each forwarder in turn repeats the process.

THP is the first heuristic to take into account three-hop information in the selection of relay nodes for the broadcasting of packets, while incurring signaling overhead that is much the same as that of heuristics based on two-hop information. THP is also the first neighbor-designated algorithm for computing a TCDS. The one-hop neighbor list and the *one-hop dominating list* communicated to a node by its one-hop neighbors provide the node with a three-hop horizon of how a broadcast message can be propagated to nodes that are three hops away, even though they are unknown.

When a broadcast protocol based on neighbor information is used it is possible to maintain fresh routes to all nodes within two hops, because every node has the two-hop neighborhood information. For example, in on-demand routing protocols (e.g., the *Ad hoc On-demand Distance Vector Protocol* (AODV) [2]) it is not necessary to broadcast the *route request* (RREQ) packet to every node in the network: disseminating it to a TCDS of the network is enough.

THP is shown to improve the performance of networks with low mobility when it is used for broadcasting of route request (RREQ) messages in AODV. However, because THP relies on an accurate view of the two-hop neighborhood, high mobility can degrade its performance considerably. To address this problem, we propose two enhancements to THP, such that it can perform well even in high-mobility scenarios.

First, a *virtual radio range* (VR), shorter than the physical radio range (RR), is used for gathering information about the local neighborhood (i.e., two-hop neighborhood). Instead of using two different transmission powers as proposed by Wu and Dai [3], a single transmission power is used while still managing to have a *buffer zone* in which neighbors can move without compromising network connectivity. Having two transmission powers, $t_{\min}$ and $t_{\max}$ (with $t_{\min} < t_{\max}$), can incur additional interference compared to having just one transmission power $t < t_{\max}$, because the transmit power of each node appears as interference noise degrading the *signal-to-noise ratio* (SNR) [4]. In general, the greater the transmit power the higher the interference to other nodes' transmissions and receptions.

Second, upon receiving a broadcast packet, the forwarder list in the packet header is analyzed together with the current information about the local neighborhood. This is done to find inconsistencies between the most up-to-date *one-hop dominating list* and the one used by the sender to compute

[1] With blind flooding, a node receiving a broadcast packet retransmits it if the node has not transmitted a copy of the packet before.

the sender's forwarder list. Changes in the local topology may have impacted the *one-hop dominating list*. If that is the case, a node may decide to relay a broadcast packet even though it was not selected as a forwarder by the sender.

The rest of this paper is organized as follows. Section 2 summarizes the related work on enhancements to broadcasting of packets in MANETs, and establishes the nomenclature used to describe our approach. Sections 3 and 4 present the three-hop horizon pruning (THP) algorithm and an example of how it works. Section 5 presents simulation results comparing THP against the best-performing broadcast techniques known to date in terms of the efficacy with which the heuristics build TCDS independently of the reliability with which data are disseminated. Section 6 applies THP to the route discovery process of on-demand routing protocols operating over a contention-based MAC protocol (the IEEE 802.11 DCF), and presents simulation results comparing AODV against AODV-THP, in which THP is used in the processing of route requests. Section 7 concludes this work.

2. Related work

Several broadcasting techniques have been proposed, differing among each other on the heuristics applied to reduce the redundancy on broadcast transmissions. Broadcasting protocols can be categorized into the following four classes [5]:

Blind flooding [6]: Each node broadcasts a packet to its neighbors whenever it receives the first copy of a broadcast packet; therefore, all nodes in the network broadcast the packet exactly once.

Probability-based methods [1]: A node re-broadcasts a packet with a given probability p (if $p = 1$, we have blind flooding).

Area-based methods [1]: A node broadcasts a packet based on the information about its location and the location of its neighbors (e.g., if a node receives the packet from a neighbor really close to it, probably it will not reach other nodes other than the nodes reached by the first broadcast).

Neighbor information methods [7]: In these methods, a node has partial topology information, which typically consists of the topology within two hops from the node (two-hop neighborhood). There are two main classes of methods in this category. In a *neighbor-designated method* a node that transmits a packet to be flooded specifies which one-hop neighbors should forward the packet. In a *self-pruning method* a node simply broadcasts its packet, and each neighbor that receives the packet decides whether or not to forward the packet.

Williams and Camp [5] have shown that *neighbor information* methods are preferred over other types of broadcast protocols. Between the two classes of neighbor information methods, Lim and Kim [7] show that the simplest form of neighbor-designated algorithm outperforms the simplest form of self-pruning, and Wu and Dai [8] show that an improved self-pruning technique outperforms the most efficient neighbor-designated algorithm (both algorithms based on the two-hop neighborhood information).

Dominating Sets (DS) play a major role in deciding the forwarder list in neighbor-designated algorithms. A DS is a set of nodes such that every node in the network is either in the set or is the neighbor of a node in the set. If the graph induced by the nodes in DS is connected, we have a *connected dominating set* (CDS) (Fig. 1(a)). The problem of determining the *minimum connected dominating set* (MCDS) is known to be NP-complete [9]. Extensive work has been done on finding good approximations for MCDS. A protocol with a constant approximation ratio of eight has been proposed by Wan et al. [10]. However, their approach requires that a spanning tree be constructed first in order to select the dominating nodes (forwarders), and only after the tree has

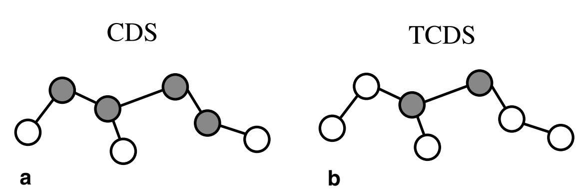

Fig. 1. (a) In a CDS, any *dominated* node is one hop from a *dominating* node (gray nodes). (b) In a TCDS, a *dominated* node is at most two hops from a *dominating* node.

been constructed a broadcast can be performed. To improve the route discovery process, an approach based on determining the CDS in real time is required. Accordingly, we focus only on techniques that satisfy this basic requirement.

A variety of conditions may be imposed on the dominating set D in a graph $G = (V, E)$. One of these conditions is the *distance domination*, which consists of requiring that each vertex in V–D be within distance k of at least one vertex in D for a fixed positive integer k [11]. In this category we have the *k-dominating set*, which can be defined as follows: for $k \geqslant 1$, a set D of vertices of a graph $G = (V, E)$ is a k-dominating set of G if every vertex of V–D is within distance k from some vertex of D. It follows that a *two-connected dominating set* (TCDS) is defined as a *two-dominating set* whose graph induced by D is connected (Fig. 1(b)). A localized algorithm for building d-dominating sets is proposed by Amis et al. [12], who also show that the problem of computing the minimum d-hop dominating set is NP-complete for unit-disk graphs.

Lim and Kim [7] show that the MCDS problem can be reduced to the problem of building a *minimum cost flooding tree* (MCFT). Given that an optimal solution for the MCFT problem is not feasible, they propose heuristics for flooding trees, resulting in two algorithms: *self-pruning and dominant pruning* (DP). They show that both algorithms perform better than *blind flooding*, and that DP outperforms the simplest form of *self-pruning*.

DP [7] is a neighbor-designated method (i.e., the sending node decides which adjacent nodes should relay the packet). The relaying nodes are selected using a distributed CDS algorithm, and the identifiers (IDs) of the selected nodes are piggy-backed in the packet as the forwarder list. A receiving node that is requested to forward the packet again determines the forwarder list.

Multi-Point Relay (MPR) [13] is another efficient broadcast technique that is similar to DP. MPR is used for reducing duplicate transmissions of control packets (i.e., link state information) in the *Optimized Link State Routing* (OLSR) protocol.

A few enhancements to dominant pruning have been reported recently [14,15]. Lou and Wu [14] propose two enhancements to DP: *total dominant pruning* (TDP), and *partial dominant pruning* (PDP). Simulation results assuming an ideal MAC layer with which no contention or collisions occur show that both TDP and PDP improve DP in a static environment. A dynamic scenario is also evaluated, and DP is shown to perform better than both TDP and PDP. We proposed *enhanced dominant pruning* (EDP) [15], which we applied to AODV to show its improvements compared to DP. We also showed that EDP improves the performance of AODV in the context of directional antennas [16].

A general framework for self-pruning has been reported by Wu and Dai [8], who proposed two approaches for broadcasting through self-pruning, one static and another dynamic. In the static approach, a CDS is constructed based on the network topology, but not relative to any broadcasting. In the dynamic approach, a CDS is constructed for a particular broadcast, and its result depends on the source and the progress of the broadcast process. For both approaches, two *coverage conditions* are presented: Coverage Condition I (CC-I), and Coverage Condition II (CC-II).

Wu and Dai showed that CC-I performs better than CC-II when node IDs are used as priority values, and when node degrees are used as priority values they present similar results. They also showed that there is a trade-off between efficiency and overhead, and that CC-I with two-hop neighborhood information, two-hop routing history, and node degrees as priority values (referenced as the *Base* configuration), outperforms the best neighbor-designated algorithm (i.e., TDP).

Several other existing algorithms (i.e., Rules 1 and 2 [17], Stojmenovic's algorithm [18], Rule k [19], Span [20], and LENWB [21]) have been shown to be special cases in the general framework. Simulation results show that the Base configuration outperforms all the others, but the difference amongst Base, Span, and LENWB is marginal. The neighborhood size is also analyzed, and it is shown that a neighborhood size larger than three hops does not add much power to the coverage conditions. In other words, the coverage conditions do not reduce much more the average number of forwarder nodes for an increasing size of the neighborhood information.

Wu and Dai [22] further analyzed the coverage conditions they reported previously [8] and showed that several other algorithms can be derived from the generic framework. The impact of four implementation issues, namely timing (static or dynamic), selection (self-pruning, neighbor-designated, and hybrid), space (network topology information), and priority (e.g., node ID, node degree), is analyzed. It is also shown that self-pruning and neighbor-designated algorithms can be combined together forming hybrid algorithms.

All distributed algorithms that rely on knowledge of the two-hop neighborhood are prone to error in the presence of mobility. And the main reason is that nodes may have inconsistent information about the neighborhood, compromising network connectivity. Wu and Dai [23] propose a solution to address the link availability problem using two transmission ranges. Information about the neighborhood and the set of forwarders is computed using a smaller radio range. And the broadcast process is performed using a larger radio range. The objective is to give nodes a *buffer zone* in which they can move without compromising local connectivity.

Our approach differs from Wu and Dai's in that we do not use two different radio ranges. Having two radio ranges increases interference, because a larger radio range means more neighbors, and more contention. We use a virtual range for computing the neighborhood and the set of forwarders. In addition to that, we modify THP to accommodate changes in the local topology. The new approach combines efficiency and reliability, performing well even in high-mobility scenarios.

3. Three-hop horizon pruning

The most efficient broadcasting algorithms that have been proposed to date prune unnecessary transmissions using two-hop topology information at each node. Each node selects a subset of one-hop neighbor nodes whose transmissions reach all its two-hop neighbor nodes. Because every nodes carries out the same type of pruning, a broadcast packet can potentially reach all network nodes using fewer transmissions, depending on the reliability of the MAC layer.

In DP, the forwarder list is a set of one-hop nodes such that all two-hop nodes are covered. The approach we use in the Three-Hop Horizon Pruning (THP) algorithm is to make the pruning process in DP more efficient by using topology information three hops away from a given node, while incurring very limited additional signaling overhead in conveying such information.

The information about the two-hop neighborhood of a node can be disseminated by means of a *neighbor protocol* that is independent of the routing protocol, or by periodically advertising the one-hop neighbor list using HELLO messages as part of the routing protocol. Without loss of generality, let us assume that nodes use HELLO messages to advertise the one-hop neighbor lists of nodes.

Based on the one-hop neighbor lists from its one-hop neighbors, each node can determine which one-hop neighbor it can use to reach any two-hop neighbor. Hence, node n_j could derive a *one-hop dominating list*, $D^j_{1-\text{hop}}$, by running standard DP over the two-hop neighborhood as if node n_j were the source (refer to Table 1 for notation).

In addition to informing its one-hop neighbors about its one-hop neighbor list, node n_j also communicates its *one-hop dominating list* $D^j_{1-\text{hop}}$ to its one-hop neighbors. To reduce the space required for this additional information, the *one-hop dominating list* is encoded in a bit-map format. Because a node lists all its one-hop neighbors in its HELLO message, and because the *one-hop dominating list* is a subset of the one-hop nodes (i.e., $D^j_{1-\text{hop}} \subset N^j_1$), it suffices to signal (i.e., one bit per node) which neighbors are *one-hop dominating nodes*.

The one-hop neighbor list and the *one-hop dominating list* communicated to a node by its one-hop neighbors provides the node with a *three-hop horizon* of how a broadcast message can be propagated to nodes that are three hops away, even though they are unknown. For node n_i, the set of all $D^j_{1-\text{hop}}$ for all $n_j \in N^i_1$, contain the set of two-hop nodes covering all three-hop nodes of node n_i. Fig. 2(a) shows an example network. Node *a*

Table 1
Notation

N_1^i	The set of one-hop neighbors of node n_i
N_2^i	The set of two-hop neighbors of node n_i
N_i	The two-hop neighborhood of node n_i (i.e., $N_i = N_1^i \cup N_2^i$)
$D_{1-\text{hop}}^j$	*one-hop dominating nodes* of node n_j (computed via DP or via MPR). That is, $D_{1-\text{hop}}^j \subset N_1^j$ such that $(\bigcup_{k\in D_{1-\text{hop}}^j} N_1^k) = N_2^j$
$\mathscr{F}_i$	The THP forwarder list as computed by node n_i, and included in the RREQ header
$\mathscr{C}$	List of candidates to be forwarders
$\mathscr{U}[j]$	List of one-hop dominating nodes of node n_j (i.e., $\mathscr{U}[j] \subset D_{1-\text{hop}}^j$) that need to be covered

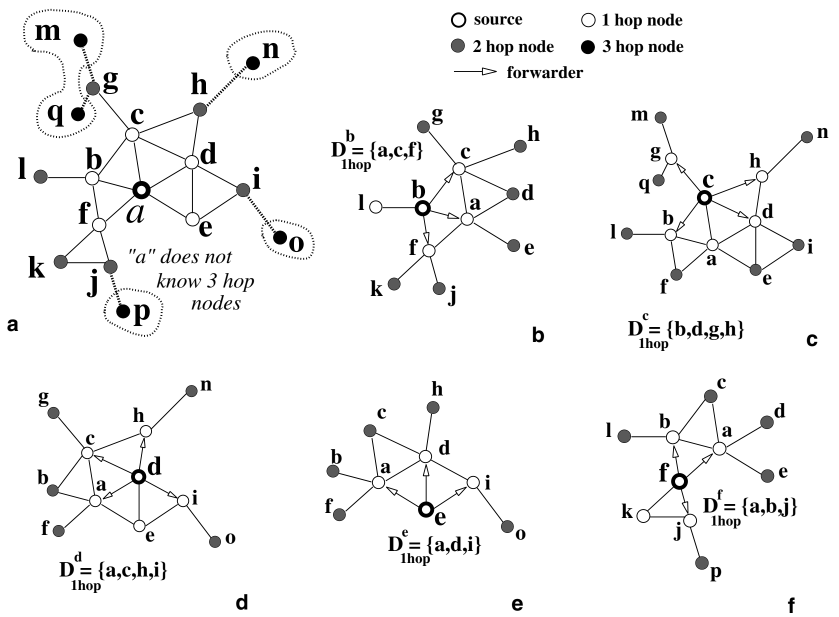

Fig. 2. Network example: (a) Node a knows its two-hop neighborhood, and the *one-hop dominating nodes* (i.e., D_{1-hop}) selected by each one-hop neighbor. A subset of nodes from $\bigcup_{n_j\in N_1^a} D_{1-\text{hop}}^j$ (i.e., $\{g,h,i,j\}$) cover all nodes in the three-hop neighborhood of node a. (b–f) show the network from the point of view of each neighbor of node a, and how each $D_{1-\text{hop}}$ list is obtained via DP.

knows its two-hop neighborhood, and also the *one-hop dominating list* advertised by each one-hop neighbor (along with the one-hop neighbor list). Fig. 2(b–f) show the network from the point of view of each one-hop neighbor of node a, and how they get to the *one-hop dominating list* (i.e., $D_{1-\text{hop}}$) by running DP. Excluding node a itself and its one-hop neighbors, the list of nodes from all $D_{1-\text{hop}}^j$ for all $n_j \in N_1^a$ is reduced to $\{g,h,i,j\}$, and we can see that all three-hop nodes of node a are covered by these set of nodes.

Instead of simply using the two-hop neighbor coverage as the main criteria for selecting forwarders as is done in standard DP, THP uses the advertised neighbor's *one-hop dominating list* (i.e., $D_{1-\text{hop}}$) to compute which one-hop neighbors have forwarders other than nodes in $N_1^i + n_i$ (i.e., nodes other than the node itself and its one-hop neighbors). Algorithm 1 presents the pseudo-code for THP (see notation on Table 1). Let $\mathscr{C}$ be the list of nodes to be considered as candidates for forwarders. One-hop neighbors of the sender

S do not need to be taken into account (line 1), because the sender already did it. For all candidates to forwarders $n_k \in \mathscr{C}$, the list of nodes to be covered (i.e., set $\mathscr{U}[k]$) is built. From the list $D^k_{1-\text{hop}}$, only nodes that are not one-hop neighbors of the current node, n_i, and are not node n_i itself, are included in the list $\mathscr{U}[k]$ (lines 2–6). The set to be covered, $\mathscr{U}$, is composed of all subsets $\mathscr{U}[k]$ for all nodes $n_k \in \mathscr{C}$. Nodes in $\mathscr{U}[k]$ that are covered (i.e., in another subset of $\mathscr{U}$ or a neighbor of some node in $\mathscr{C}$) by another node in $\mathscr{C}$ can be eliminated (lines 7–12, and Fig. 3). For all candidates $n_k \in \mathscr{C}$ and for every node $n_m \in \mathscr{U}[k]$, the algorithm checks if there is another candidate to forwarder $n_l \in \mathscr{C}$ such that node n_m is a neighbor of node n_l. If this is the case, then node n_m can be removed from the set covered by node n_k (i.e., $\mathscr{U}[k]$). In other words, if there is some candidate n_l that is neighboring a node n_m (which may or not be in $\mathscr{U}[l]$) that is in the set to be covered by candidate node n_k, then node n_m does not need to be covered by node n_k, given that node n_l being a neighbor of node n_m did choose it as *one-hop dominating node* or has another neighbor covering the nodes covered by node n_m. In case node n_l did not choose node n_m as a *one-hop dominating node*, it may be the case that node n_l has another neighbor(s) covering the nodes advertised by node n_m, or all neighbors of node n_m are also neighbors of node n_l. If the set $\mathscr{U}[k]$ becomes empty, then node n_k is no longer a candidate to forwarder, and can be removed from the set $\mathscr{C}$ (lines 11 and 12). One restriction when eliminating redundancy from the set $\mathscr{U}$, is that a node n_k must have all its nodes in the set $\mathscr{U}[k]$ checked before proceeding to the next node in the set $\mathscr{C}$. After all nodes in $\mathscr{C}$ are processed, the nodes remaining in the set $\mathscr{C}$ are selected as forwarders.

Algorithm 1. (*THP*)

```
Data: n_i (any given node), S (sender), D^k_{1-hop} for all k ∈ N^i_1
Result: F_i, the forwarder list
begin
1     C ⟵ N^i_1 − N^S_1
      /* Select neighbors with one-hop dominating nodes other than
         one-hop neighbors and the node itself                        */
2     for n_k ∈ C do
3         U[k] ⟵ ∅
4         for n_l ∈ D^k_{1-hop} do
5             if n_l ∉ (N^i_1 + n_i) then
6                 U[k] ⟵ U[k] + {n_l}
      /* Exclude candidates covered by another candidate in C         */
7     for n_k ∈ C do
8         for n_m ∈ U[k] do
9             if ∃ (n_l ≠ n_k) ∈ C | n_m ∈ N^l_1 then
10                U[k] ⟵ U[k] − n_m
11                if U[k] == ∅ then
12                    C ⟵ C − n_k
      /* For every node n_k ∈ C, and for every n_m ∈ U[k], there is no
         other n_l ∈ C such that n_m ∈ U[l]; therefore, all nodes in C are
         forwarders.                                                  */
13    F_i ⟵ C
14    return F_i
end
```

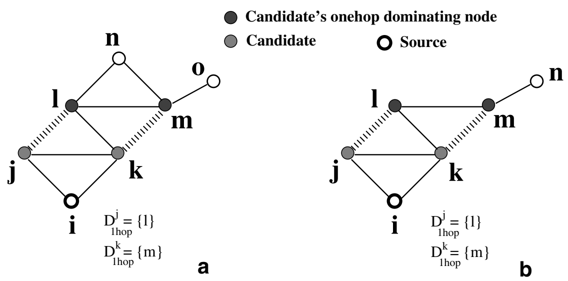

Fig. 3. Node k does not choose node l as a forwarder (standard DP, for the hello message). In Case (a), another node (i.e., neighbor m) covers all nodes covered by l (excluding those nodes covered by k), plus node o. In Case (b), all nodes covered by l are one-hop neighbors of node k. In any case, it is safe to remove l from $\mathscr{U}[j]$, because node k covers all nodes covered by l, or has other neighbor(s) covering the nodes covered by l.

The following theorem proves that THP forms a TCDS in a connected network.

Theorem 1. *Given a connected graph* $G(V,E)$, *the node subset* N', *computed using the THP algorithm, forms a TCDS of G.*

Proof. By the definition of a *one-hop dominating set*, for any node n_k in the network, the set $D^k_{1-\text{hop}}$ is a subset of nodes of N^k_1 such that all nodes in N^k_2 are covered.First, we consider the set of forwarders defined by the source, n_i, and then from the initial set of forwarders, $\mathscr{F}_i$, we show how the TCDS is constructed.For the source node n_i, the list of candidates to forwarders, $\mathscr{C}$, include all the one-hop neighbors of node n_i (i.e., N^i_1).Because n_i is the source, $S = \emptyset$.The set $\mathscr{U} = \sum_{j \in N^i_1} \mathscr{U}[j]$ cover all three-hop nodes of node n_i, because it includes all the nodes covering the two-hop neighborhood of all neighbors of node n_i (i.e., $\forall n_j \in N^i_1$, node n_i knows $D^j_{1-\text{hop}}$).A node $n_k \in \mathscr{U}[j]$, such that node $n_k \in N^l_1$ for node $n_l \in \mathscr{C}$ ($n_l \neq n_j$), can be excluded from $\mathscr{U}[j]$, because node n_k is covered by node n_l, which is another valid candidate to forwarder.This assertion holds given that all nodes in $\mathscr{U}[j]$ are processed before proceeding to the remaining nodes in $\mathscr{C}$ (i.e., for any node $n_j \in \mathscr{C}$, check this condition for all nodes in $\mathscr{U}[j]$, before proceeding to the next node $n_l \in \mathscr{C}$).Hence, the nodes in $\mathscr{U}$ cover all two-hop and three-hop nodes of node n_i.The set of forwarders, $\mathscr{F}_i$, is a subset of nodes in the set $\mathscr{C}$, such that all nodes in $\mathscr{U}$ are covered.On their turn, nodes $\{n_{j_1}, n_{j_2}, \ldots, n_{j_m}\} \in \mathscr{F}_i$ compute their sets $\mathscr{C}$, excluding the sender (i.e., $S = n_i$), and the one-hop neighbors shared with the sender ($N^j_1 \cap N^i_1$), because these nodes are already considered by node n_i when deriving the set $\mathscr{F}_i$.Nodes $\{n_{j_1}, n_{j_2}, \ldots, n_{j_m}\} \in \mathscr{F}_i$ derive their list of forwarders, i.e., $\{\mathscr{F}_{j_1}, \mathscr{F}_{j_2}, \ldots, \mathscr{F}_{j_m}\}$ (which can be an empty list in case no candidates lead to three-hop nodes).Each individual set in$\{\mathscr{F}_{j_1}, \mathscr{F}_{j_2}, \ldots, \mathscr{F}_{j_m}\}$ cover the three-hop neighborhood of nodes $\{n_{j_1}, n_{j_2}, \ldots, n_{j_m}\}$, respectively. Given that the set of nodes $\{n_{j_1}, n_{j_2}, \ldots, n_{j_m}\}$ cover the three-hop nodes of node n_i, the joint sets $\{\mathscr{F}_{j_1}, \mathscr{F}_{j_2}, \ldots, \mathscr{F}_{j_m}\}$ cover the four-hop nodes of node n_i. Therefore, the set of forwarders chosen subsequently cover all nodes $d+3$ hops away from the source, where d is the distance from the forwarder to the source. Because a forwarder is selected by a previous forwarder, or by the source, the set of forwarders is connected. Furthermore, because a forwarder checks for neighbors that reach three-hop nodes, it is guaranteed that, whenever there is at least one three-hop node, a forwarder is selected among the forwarder's one-hop neighbors. Because the selection process ends when no more three-hop nodes can be reached from a forwarder, it is guaranteed that any node in the network is at most two hops from a forwarder. **2** □

4. Example of THP operation

Fig. 4 depicts an example of applying THP to compute a TCDS, having node a as the source. First, lets consider the *one-hop dominating lists* announced by the neighbors of node a: $D^k_{1-\text{hop}} = \{g,h,o,p\}$, $D^p_{1-\text{hop}} = \{a,b,h,k\}$, $D^s_{1-\text{hop}} = \{a,p,b\}$, $D^o_{1-\text{hop}} = \{a,k,r\}$, and $D^r_{1-\text{hop}} = \{a,o,w\}$. Because node a is the source, all its one hop neighbors are candidates to be forwarders. We have that $\mathscr{U}[k] = \{h,g\}$, $\mathscr{U}[o] = \emptyset$, $\mathscr{U}[p] = \{h,b\}$, $\mathscr{U}[r] = \{w\}$, and $\mathscr{U}[s] = \{b\}$. Node o is not a candidate, because it does not provide *one-hop dominating nodes* other than one-hop neighbors of node a, or node a itself. In other words, node o has no two-hop neighbors other than those reachable through node a's neighbors or node a itself.

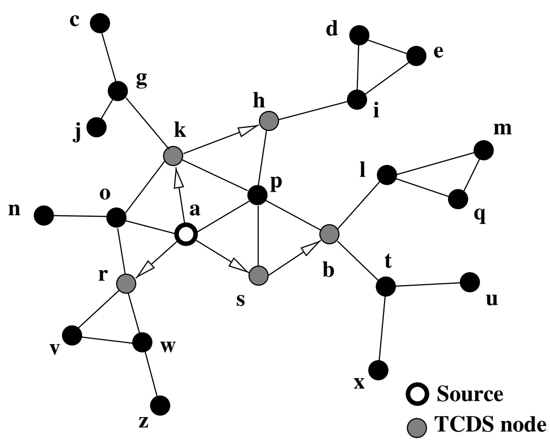

Fig. 4. Building a TCDS using THP. Every *dominated* node is at most two hops from a *dominating* node.

Therefore, there is no use to forward the packet toward node o. After excluding candidates covered by another candidate to be a forwarder (considering nodes in $\mathscr{C}$ are processed in alphabetical order), we obtain: $\mathscr{U}[k] = \{g\}$, $\mathscr{U}[p] = \emptyset$, $\mathscr{U}[r] = \{w\}$, and $\mathscr{U}[s] = \{b\}$. Notice that node h is not listed in any of the $\mathscr{U}$'s list. However, there is at least one node (node k) in $\mathscr{C}$ covering node h, because a node n_m can only be removed from list $\mathscr{U}[k]$ if there is another valid candidate n_l covering node n_m (i.e., $n_m \in N_1^l$). The source's forwarder list is then defined as $\mathscr{F}_a = \{k, r, s\}$. Now we look at each node in $\mathscr{F}_a$. Node k has $\mathscr{C} = \{g, h\}$, $\mathscr{U}[g] = \emptyset$, and $\mathscr{U}[h] = \{i\}$. Note that both p and o are excluded from the set $\mathscr{C}$, because they are also neighbors of the sender. Based on that, we have that $\mathscr{F}_k = \{h\}$. Node s has $\mathscr{C} = \{b\}$, and $\mathscr{U}[b] = \{l, t\}$, which gives us $\mathscr{F}_s = \{b\}$. Node r has $\mathscr{C} = \{v, w\}$, $\mathscr{U}[v] = \emptyset$, and $\mathscr{U}[w] = \emptyset$; hence, it has no forwarders. Node h has no forwarders, because the only candidate, node i, has no node in $D^{i=h}_{1-\text{hop}}$ other than node h itself (i.e., $\mathscr{U}[i] = \emptyset$). For node b, we have a similar situation, where both candidates, nodes l and t, lead to no other three-hop neighbor (i.e., $\mathscr{U}[l] = \mathscr{U}[t] = \emptyset$); hence, it has no forwarder.

5. Efficacy of THP

This section evaluates the efficacy with which THP operates relative to other heuristics, when a TCDS is preferred over a CDS. We compare THP against the best-performing heuristics reported to date, namely DP, TDP, EDP, MPR, CC-I (with 2-hop neighborhood information, 2-hop routing history, and node-degrees as priority values), and an approximation to the MCDS problem. With the exception of MCDS (which is used as a lower bound for comparison purpose), all the other algorithms require the information about the two-hop neighborhood of nodes.

There is a clear trade-off between efficiency and reliability; that is, fewer nodes broadcasting reduces contention and collision of packets, but it may also reduce the chances of all nodes in the network receiving the broadcast packet. Therefore, the reliability of the MAC protocol is expected to affect the performance of any such broadcasting algorithms. To focus on the efficiency of the heuristics themselves, we use a customized simulator and assume an *ideal* MAC protocol with which no collisions can occur. This is the same approach adopted in [8,22,19,14] to compare the efficacy of heuristics.

DP is a distributed algorithm that determines a set cover based on the knowledge of the two-hop neighborhood. DP uses the *greedy set cover* (GSC) algorithm to compute the forwarder list of a packet. GSC recursively chooses one-hop neighbors that cover the most two-hop neighbors, repeating the process until all two-hop neighbors are covered.

Like DP, MPR also applies GSC in the selection of dominating nodes. However, MPR first chooses as forwarders those candidates that have exclusive coverage of some two-hop neighbor, and only then apply GSC over the remaining nodes. Fig. 5 shows an example illustrating the benefits of this approach over standard DP. With standard DP, node a in Fig. 5 can start choosing *any* one of its neighbors, because they all have the same coverage area (i.e., two nodes, excluding node a and the one-hop neighbors of a). Therefore, with standard DP node a could choose node c as a forwarder first, and in this case it would be forced to select its other two neighbors, because node b is the only neighbor covering node e, and node d is the only neighbor covering node f. In the MPR approach, nodes b and d must be selected first,

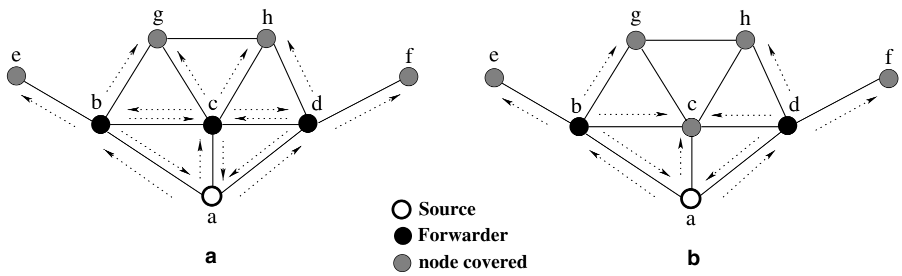

Fig. 5. DP (a) versus MPR (b).

and in this case node *c* is not chosen, because nodes *b* and *d* cover all the two-hop nodes.

TDP [14] requires that the two-hop neighborhood of the sender be piggy-backed in the header of the packet. This information reduces the size of the two-hop neighbor set that needs to be covered by the forwarders. The header size increases proportionally to the number of nodes in the two-hop neighborhood, which may become a problem in dense networks. PDP [14] enhances DP by eliminating the two-hop nodes advertised by a neighbor shared by both the sender and the receiver (forwarder). EDP [15] requires the *second-to-previous* (STP) forwarder list in addition to the forwarder list, reducing the number of forwarders compared to DP.

In CC-I (dynamic), a node n_i does not broadcast the packet if for any two neighbors n_j and n_k, there is a path connecting them via several intermediate nodes with either higher priority values (e.g., node degree, node IDs) than node n_i, or with visited node status (i.e., the *h* most recently visited nodes are included in the packet header).

Because the MCDS problem is an NP-complete problem, we use an approximation algorithm as a lower bound for the MCDS problem when comparing against the other algorithms. The algorithm used is based on the solution provided by Guha and Khuller [24]. This algorithm runs in polynomial time and achieves an approximation factor of $O(H(d))$, where *d* is the maximum degree, and $H(d)$ is the d^{th} harmonic number (i.e., $H(d) = \sum_{i=1}^{d} 1/i$). Nevertheless, this algorithm is not suitable for wireless ad-hoc networks, because it requires the knowledge of the whole network topology. The approximation algorithm used in [14] is not a good approximation because it uses a scanning rule that fails in some circumstances according to Guha and Khuller [24].

For the simulations, we vary the network size (i.e., number of nodes and terrain size) and measure the total number of forwarders for flooding the whole network. For each configuration (i.e., number of nodes and terrain size) we obtain the value for the metrics for 500 arbitrary networks (nodes are randomly placed over the terrain, and connectivity is tested to ensure that the network is connected). Results represent the average over the 500 different networks. The network size is varied from 20 nodes to 200 nodes. For the same number of nodes, we vary the terrain size according to two configurations so that we can test the algorithms for different node density (see Table 2). Configuration 1 has a node density of 80 nodes/km^2, and Configuration 2 has 125 nodes/km^2. For both configurations the radio range is set to 250 m; consequently we have that nodes in Configuration 2 have, in average, larger node degree than nodes in Configuration 1.

Because THP prunes over the *one-hop dominating lists* advertised by the one-hop neighbors, computing the *one-hop dominating lists* using MPR instead of DP does not add much power to THP. Table 3 shows the results for THP (Configuration 1) based on DP and based on MPR. As we can see, there is a marginal difference between *THP with DP* and *THP with MPR*.

5.1. Configuration 1

Fig. 6 presents the total number of forwarders for the six broadcasting algorithms. Because in

Table 2
Terrain size (in meters)

# of nodes	Configuration 1	Configuration 2
20	499 × 499	400 × 400
30	612 × 612	489 × 489
40	707 × 707	565 × 565
50	790 × 790	632 × 632
60	866 × 866	692 × 692
70	935 × 935	748 × 748
80	999 × 999	800 × 800
90	1060 × 1060	848 × 848
100	1118 × 1118	894 × 894
110	1172 × 1172	938 × 938
120	1224 × 1224	979 × 979
130	1274 × 1274	1019 × 1019
140	1322 × 1322	1058 × 1058
150	1369 × 1369	1095 × 1095
160	1414 × 1414	1131 × 1131
170	1457 × 1457	1166 × 1166
180	1500 × 1500	1200 × 1200
190	1541 × 1541	1232 × 1232
200	1581 × 1581	1264 × 1264

Table 3
THP using DP versus THP using MPR: number of forwarders (average ± standard deviation)

# of nodes	THP using DP (average ± std)	THP using MPR (average ± std)
20	2.102 ± 0.066	2.094 ± 0.066
30	4.356 ± 0.085	4.324 ± 0.084
40	6.978 ± 0.096	6.932 ± 0.096
50	9.928 ± 0.118	9.888 ± 0.116
60	12.884 ± 0.127	12.814 ± 0.128
70	16.226 ± 0.146	16.15 ± 0.145
80	19.474 ± 0.154	19.4 ± 0.153
90	22.706 ± 0.165	22.578 ± 0.166
100	26.3 ± 0.187	26.198 ± 0.184
110	30.104 ± 0.188	30.022 ± 0.186
120	33.576 ± 0.21	33.426 ± 0.209
130	37.468 ± 0.212	37.344 ± 0.209
140	40.882 ± 0.23	40.664 ± 0.228
150	44.654 ± 0.23	44.516 ± 0.23
160	48.978 ± 0.249	48.826 ± 0.248
170	52.438 ± 0.249	52.278 ± 0.25
180	56.156 ± 0.260	55.98 ± 0.259
190	60.584 ± 0.291	60.306 ± 0.293
200	64.256 ± 0.304	64.014 ± 0.304

THP nodes are at most two-hop away from a node in the TCDS, we have situations (e.g., small network sizes, 20–50 nodes) where THP produces a TCDS with smaller number of nodes than a CDS in MCDS. Anyway, MCDS is used as the reference to the best possible results for calculating the DS (but not feasible because we do not want to require the nodes in the wireless network to keep fresh information about the whole network topology). As expected, TDP improves DP for all network sizes, but it is more noticeable for larger networks (i.e., 100 nodes or more). EDP and TDP present similar results, with TDP performing slightly better for some network sizes. MPR performs better than DP, TDP, and EDP, for networks larger than 140 nodes. We notice that CC-I starts performing better than TDP only for larger networks (i.e., 140 nodes or more). In all circumstances, THP outperforms the other distributed approaches.

5.2. Configuration 2

The networks in Configuration 2 are denser (i.e., larger node degree) than the networks in Configuration 1. Fig. 7 shows the average number of forwarders for all broadcast algorithms. The difference between DP and TDP is more noticeable, because the networks are denser it pays off to have the two-hop neighborhood of the sender (i.e., in TDP) when calculating the set to be covered. EDP and TDP present similar results, but unlike in the previous configuration, EDP performs better for networks with more than 130 nodes. TDP performs better than CC-I for networks smaller than 120 nodes. MPR starts performing better than DP, TDP, and EDP, for networks larger than 130 nodes. But the difference between MPR and the other DP variants is more noticeable compared to the previous configuration. For all network sizes, THP performs better than the other distributed broadcast algorithms. We also notice that THP performs better than MCDS for networks with 70 or fewer nodes. Once again, this particular behavior takes place because THP builds a TCDS instead of a CDS, and fewer nodes exist in the TCDS than in the CDS, especially for dense an small networks. The difference between THP and CC-I is more accentuated than in Configuration 1 for all the network sizes tested. This shows that the performance improvements attained with THP increases as the network gets denser.

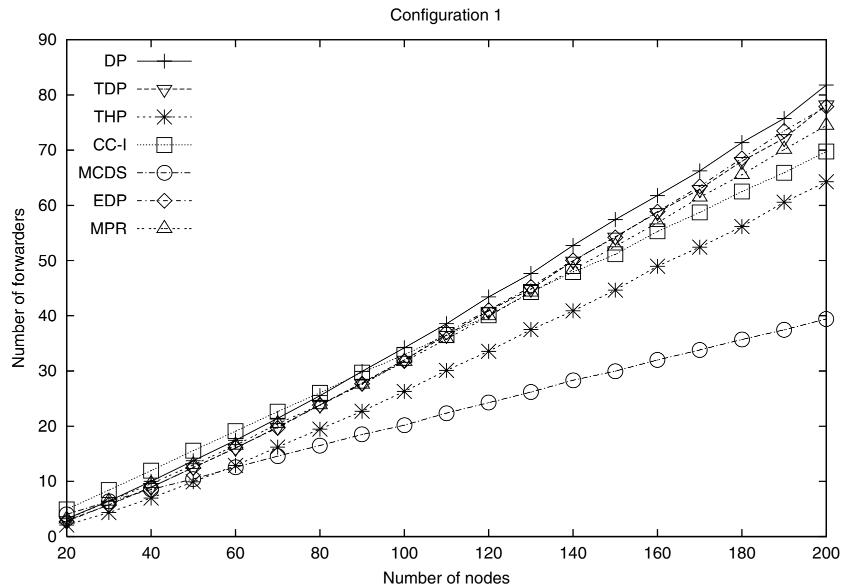

Fig. 6. Configuration 1: average number of forwarders varying the number of nodes.

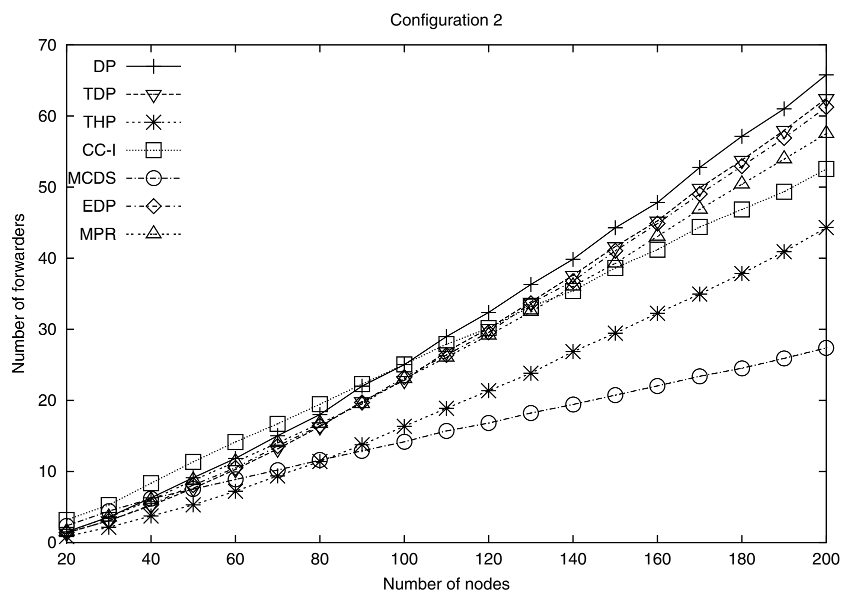

Fig. 7. Configuration 2: average number of forwarders varying the number of nodes.

6. Using THP for route discovery

THP can be applied to any type of broadcast operation that can take advantage of TCDS. One such operation is the dissemination of route requests (RREQ) in the route discovery process of on-demand routing protocols. For the purpose of discovering a route to a destination, it suffices that the RREQ reaches those nodes with a route to the desired destination. There are two cases to consider in terms of how THP can be used in this context.

If routes to two-hop neighbors are maintained pro-actively, then a node that is one or two hops away from the destination can reply to the RREQ directly.

On the other hand, if routes to two-hop neighbors are not available pro-actively, then a RREQ can be propagated in a number of ways once it reaches a node that is two hops away from the destination. The RREQ can be relayed using the expanding ring search with TTL set to 2. Alternatively, a node can compute forwarders within the two-hop neighborhood using a dominating set technique different than THP (e.g., DP).

To study the impact of THP on the route discovery process, we implemented THP as the basis for deciding which nodes should broadcast RREQ messages in the route discovery process of AODV. We named the resulting protocol AODV-THP, and implemented it in *Qualnet* [25]. To compare AODV-THP against AODV, we use traffic and mobility models similar to those previously reported for the performance of AODV [26].

To address reliability, we used two versions of AODV-THP. First, AODV-THP implements THP as described previously. Second, we increase the coverage requirement of DP when computing the *one-hop dominating list* advertised in the HELLO messages (i.e., $D_{1-\text{hop}}$). Instead of requiring *at least one* dominating node (forwarder) per two-hop neighbor, every two-hop neighbor is covered by *at least two* forwarders (except when just one one-hop neighbor covers a two-hop node). This increases the chances that a two-hop neighbor receives a RREQ. This second variant is referenced as AODV-THP *two-cover*. The two variants of AODV-THP and AODV are tested with HELLO messages sent at a rate of 1 s and 2 s. For AODV, we also present results without the use of HELLO messages.

AODV-THP would certainly incur much less overhead if it worked over a MAC protocol that exchanged the neighbor and forwarder information that we assume is exchanged as part of the routing protocol itself.

Experiments are repeated for 10 trials with different random-number seeds, traffic endpoints, and topologies. Topology and traffic patterns are fixed using off-line generated mobility and packet generation scripts. This means that all protocols are compared having identical node mobility and traffic demands. Each data point represents the average of the 10 trials.

Four performance metrics are evaluated:

- *Packet delivery ratio*, the ratio of the data packets delivered to the destination to those generated by the CBR sources.
- *Average end-to-end delay* for data packets, including all possible delays caused by route discovery latency, queuing at the interface, retransmission delays at the MAC layer, and propagation and transfer times.
- *Normalized routing load*, the number of routing packets transmitted per data packet delivered to the destination, where each hop traversed by the packet is counted as one transmission.
- *MAC collisions*, the number of collisions detected at the MAC layer.

Table 4 presents the set of parameters used in the simulations. The network is composed of 50 nodes spread over an area of $1500\,\text{m} \times 300\,\text{m}$. The radio model used is a 2 Mbps IEEE 802.11

Table 4
Set of parameters used in the simulations of AODV–THP

Number of nodes	50
Terrain size	1500 m × 300 m
Data rate	2 Mbps
Radio range	280 m for standard THP, and 250 m for enhanced THP
MAC protocol	IEEE 802.11
Data traffic, packet size	CBR, packets of 512-bytes
Number of flows, and duration	30 active flows, lasting in average for 30 s (exponential distribution); in average 580 flows are created during the simulation time
Mobility model	random way-point (velocities between 1 and 20 m/s)
Pause times	0 s (always moving), 50 s, 100 s, 300 s, 400 s, and 600 s (static)
Simulation time	600 s

device with a nominal transmission range of 280 m. Traffic sources are continuous bit rate (CBR). Only 512-bytes data packets are used. The source-destination pairs are chosen randomly among the nodes in the network. Flows last in average for 30 s (following an exponential distribution). Source nodes keep active flows during all simulation time (new destinations are randomly selected as needed). During the simulation time, an average of 580 flows are initiated, and at any given time there are at least 30 active flows. Nodes begin transmitting at 50 s plus an offset uniformly chosen over a 5 s period to avoid synchronization in their initial transmissions. The simulation time is set to 600 s, and identical mobility and traffic scenarios are used for all protocols. Nodes are placed uniformly over a grid initially. Nodes move according to the random way-point model with velocities between 1 and 20 m/s. Six pause times are tested: 0 s (always moving), 50 s, 100 s, 300 s, 400 s, and 600 s.

Fig. 8 presents the packet delivery ratio results. As expected, AODV-THP does not perform very well in scenarios with frequent topology changes. One of the main reasons is that it is more difficult to get an accurate view of the local topology when it changes more frequently. As we increase the rate of HELLO messages (i.e., AODV-THP: 1 s HELLO), THP improves its performance because, even though we are introducing more broadcast transmissions, nodes respond to topology changes faster. AODV-THP with HELLOs sent every 1*s* starts performing better than AODV at 300 s pause time. When we increase the DP coverage from one to two dominating nodes (i.e., AODV-THP: 2 cover), we observe that the extra redundancy benefits the protocol in all situations, but specially for the high mobility scenarios. Here we can see the trade-off that exists between efficiency and reliability, and its relation with redundancy in broadcast transmission. For low mobility scenarios, it pays off to take advantage of a more accurate view of the local topology when making decisions about which node should broadcast a packet. For high mobility scenarios, THP using a *2 cover* (i.e., AODV-THP: 2 cover) increases the delivery ratio by about 10% compared to the worst variant of THP.

Fig. 9 presents the average end-to-end delay results. Given that approximately 580 flows are initiated during the simulation time, we observe that the large number of redundant broadcast transmissions (i.e., due to the route discovery process) affect the end-to-end delay in AODV.

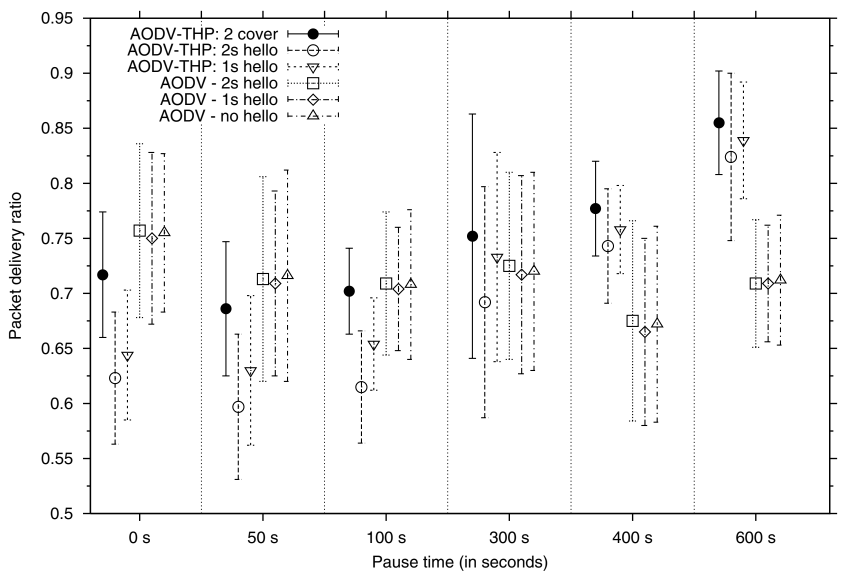

Fig. 8. 50 nodes, 30 active flows (average of 580 total flows): Packet delivery ratio.

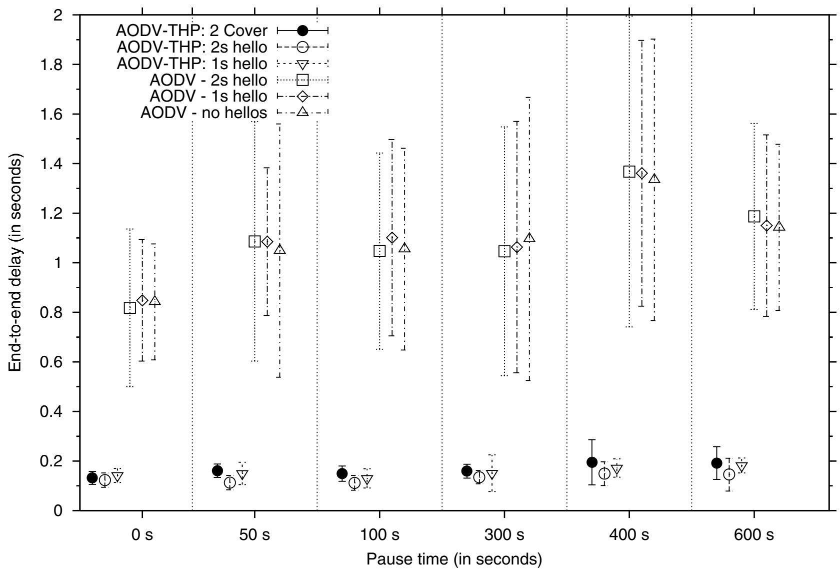

Fig. 9. 50 nodes, 30 flows (average of 580 total flows): average end-to-end delay.

As shown previously, THP prunes more redundant broadcast transmissions than any other localized broadcast algorithm, and we can see here how it reflects in the overall performance of an on-demand routing protocol. The periodicity of HELLO messages reflect on the end-to-end delay as well. For THP, nodes keep a more accurate view of the local topology when HELLO messages are transmitted every 1 s. In this case, more packets are delivered, but they are also delivered faster. THP with *2 cover* redundancy (with hellos transmitted every 1 s HELLO) presents a slightly larger average delay but it also delivers more packets for all pause-time values. For AODV, the frequency of HELLO transmissions do not affect much the end-to-end delay in such a scenario with a large number of flows. Together with the previous results for the delivery ratio, we can see that the reduction of redundant broadcast transmissions translate in a better and faster response to the route discovery process; consequently, more packets are delivered at a smaller cost.

Fig. 10 shows the normalized routing load results (with respect to data packets delivered at the destination). All the THP variants present a much smaller overhead than AODV, because of the reduction on the number of redundant broadcast transmissions. As for the impact of the periodicity of HELLO messages, we observe slightly more control overhead in AODV when HELLO messages are sent every 1*s*, compared to the two other variants. AODV without HELLOs, performs just slightly better than AODV with 2 s HELLOs in terms of control overhead, delivery ratio, and end-to-end delay.

Fig. 11 presents the results for the number of packet collisions. AODV with and without HELLOs attains similar results, showing that the increase in collisions is not due to the introduction of HELLO messages. The extra redundancy of RREQ transmissions is what results in more contention and collisions. As for AODV-THP, we observe that the periodicity of HELLO messages has a direct impact on the number of collisions, and that is because we reduce significantly the number of redundant broadcast, such that the introduction of any extra broadcast transmissions (i.e., HELLO messages) reflects in more contention and collisions in the network. Considering all the previous results, THP is shown to improve AODV performance in all aspects for scenarios with low mobility (i.e., pause time larger than 100 s).

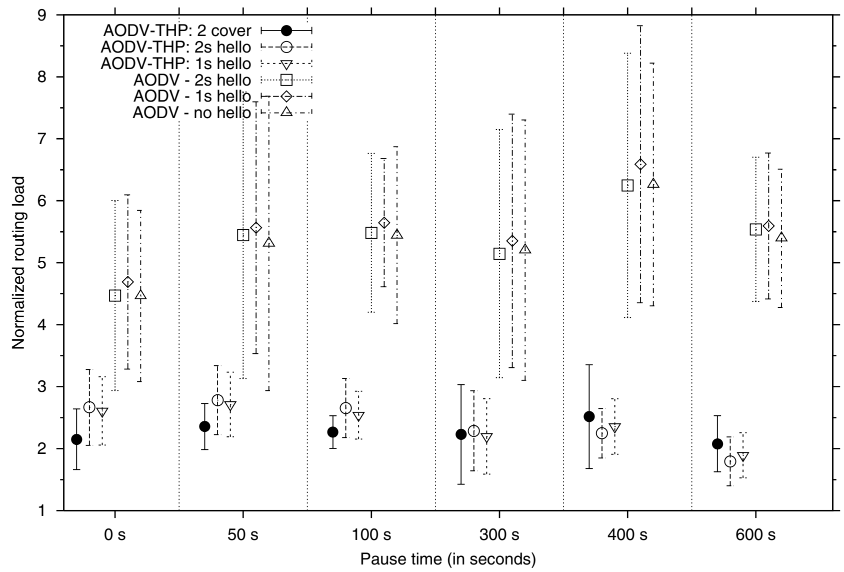

Fig. 10. 50 nodes, 30 flows (average of 580 total flows): normalized routing load.

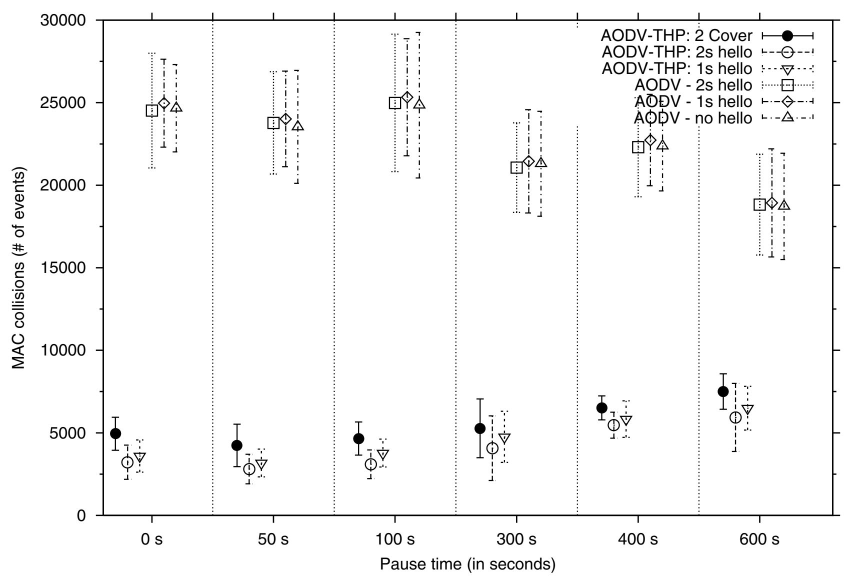

Fig. 11. 50 nodes, 30 flows (average of 580 total flows): number of MAC collisions.

6.1. Enhancing THP

Because techniques such as THP rely on an accurate view of the two-hop neighborhood, high mobility can degrade its performance considerably. To tackle this problem, we propose two enhancements to THP. The two improvements are:

- Maintain neighbor information using a virtual radio range (shorter than the physical radio range), rather than using two different radio ranges [23,3]. We use neighbor location, and regard as neighbors only those nodes within the virtual radio range.

- Use both information provided by the forwarder list, and the freshest information about the local neighborhood to decide if the node should broadcast the packet even though it is not selected as a forwarder.

As in the work by Wu and Dai [23,3], the gap between the virtual and physical ranges constitutes a buffer zone in which neighbors can move without incurring loss of connectivity. However, our approach applies just one transmission power, instead of two different transmission powers [23,3]. Having two transmission powers, $t_{\min}$ and $t_{\max}$ (with $t_{\min} < t_{\max}$), can incur additional interference compared to having just one transmission power $t < t_{\max}$, because the transmit power of each node appears as interference noise degrading the *signal-to-noise ratio* (SNR) [4]. In general, the greater the transmit power the higher the interference to other nodes' transmissions and receptions.

To know if another node is within virtual radio range, a node can either use node location information (provided by GPS, for instance) or estimate the distance to the node based on the signal strength of the receiving packet [27]. In the first case, the information about the node location should be piggy-backed in HELLO messages together with the neighbor list. The second option is effective and does not add as much complexity to the system as the first one. In either case, the exact location of a node is not needed because a node needs to estimate only if a node is within virtual radio range. For simulation purposes, we assume that nodes exchange their location informations using the periodic HELLO messages.

Routes to one-hop neighbors (i.e., nodes within physical radio range) are kept as in standard AODV. Upon receiving a HELLO message, nodes update the route to the node sending the packet. The neighbor list advertised in the HELLO message contains only the neighbors within virtual radio range, and the $D_{1-\text{hop}}$ list is also computed using the virtual neighbor list.

Fig. 12(a) and (b) presents an example with node *a* starting within the virtual range of node *s*, and moving away from *s* but still within radio range of *s*. In this case, even though node *a* is no longer a one-hop neighbor (for the purpose of forwarding computation), it is still reachable. Fig. 12(c) and (d) shows an example where a new node (i.e., node *b*) moves within virtual radio range. Supposing node *S* later on (i.e., after t_1) receives a broadcast packet for which it is not listed as a forwarder (i.e., $S \ni \mathscr{F}_{\text{sender}}$), node *S* would still broadcast the packet in case there is no forwarder in $\mathscr{F}_{\text{sender}}$ covering node *b*.

Algorithm 2 shows the pseudo-code for handling route requests (these are extensions to the regular route request procedure as specified in [28]). First, the node checks if there is a valid route to the destination. If that is the case, a RREP is sent back to the source of the RREQ. If no valid

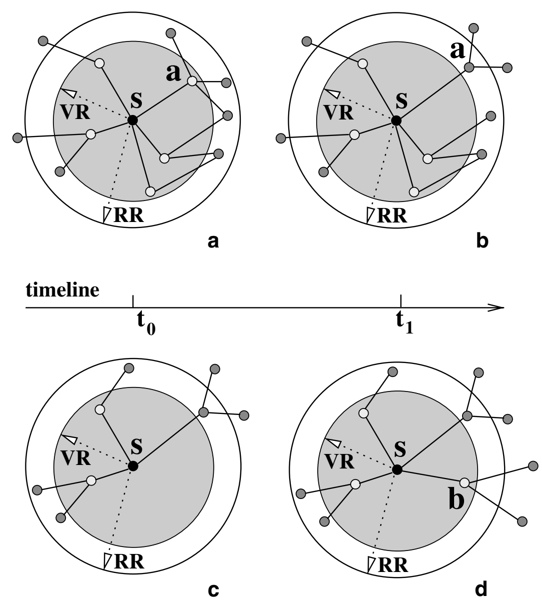

Fig. 12. (a) When computing the *one-hop dominating list*, only nodes within virtual radio range (VR) are considered one-hop neighbors. (b) Even though node *a* moved out virtual range, it is still within radio range (RR). (c and d) Node *b* moves into virtual radio range of node *S*. In any previous HELLO message sent by *S*, node *b* was not in the advertised $D^{S}_{1-\text{hop}}$ list. If after time t_1 node *S* receives a broadcast for which it is not selected as a forwarder, and if no selected forwarder is covering node *b* then *S* will broadcast the packet anyway.

route exists, there are two cases that make the node relay the RREQ packet. If the node is listed in the forwarder list (i.e., $\mathscr{F}_{\text{sender}}$, available from the RREQ header), it means the node must forward the packet. If that is not the case, recent modifications to the local topology may have changed the *one-hop dominating list*, and it may be different from the list used by the sender to compute the THP forwarder list. To check that, the node computes (see Algorithm 3) the *one-hop dominating list* (i.e., $D^i_{1-\text{hop}}$, advertised in periodic HELLO messages and used to compute the THP forwarder list). If there is any node in $D^i_{1-\text{hop}}$ that is not covered by at least one forwarder in $\mathscr{F}_{\text{sender}}$, then the node should relay the packet even though it has not been selected as a forwarder by the sender. In other words, if there is no forwarder in $\mathscr{F}_{\text{sender}}$ covering any *one-hop dominating node*, then the broadcast might not reach the segment of the network connected to these *one-hop dominating nodes*.

Nodes relaying a RREQ packet, first compute the THP forwarder list, update the RREQ header, and only then broadcast the packet. As in standard AODV, the RREQ eventually reaches a node with a valid route to the destination, or the destination itself (considering the network is connected). Because fewer nodes relay the same RREQ packet, we expect less contention and fewer packet collisions, as well as a smaller end-to-end delay, because the RREQ message propagates faster.

Algorithm 2. (*Handle Request*)

Data: n_i, S (sender), D (destination), $\mathcal{F}_S$ (sender's forwarder list)
begin
  if *Valid Route to D* **then**
    Send RREP back to Source
  else
    if $n_i \in \mathcal{F}_S$ *OR Check Status*$(n_i,\ \mathcal{F}_S) ==$ *Broadcast* **then**
      Compute $\mathcal{F}_i$ and update RREQ header
      Relay RREQ
end

Algorithm 3. (*Check Status*)

Data: n_i, $\mathcal{F}_S$ (sender's forwarder list)
Result: Broadcast or Silent
begin
  /* First compute *one-hop dominating list* (i.e., applying DP) using the freshest information about the neighborhood */
  $D^i_{\text{1-hop}} \longleftarrow DP(N_i)$
  for $n_k \in D^i_{\textit{1-hop}}$ **do**
    /* If there is any *one-hop dominating node* in $D^i_{\text{1-hop}}$ that is not covered by a node in $\mathcal{F}_S$ then broadcast */
    if $\nexists\ n_l \in \mathcal{F}_S \mid n_l \in N^k_1$ **then**
      return *Broadcast*
  return *Silent*
end

The two enhancements are implemented in AODV-THP *two-cover*, because it was shown previously to perform better than the other AODV-THP variant. In order to make the network more sparse, we have reduced the nominal transmission range to 250 m. And to increase spatial reuse, directional reception is used in place of omni-directional reception. The other parameters are identical to those in the previous scenario.

To evaluate the impact of the two enhancements, we run simulations for different virtual radio ranges. The following list summarizes all variants under consideration: AODV-THP 1.0R, with virtual range and radio range the same (this way we can see the impact of the second enhancement alone); AODV-THP 0.85R, with virtual range set to 85% of the radio range; AODV-THP 0.75R, with virtual range set to 75% of the radio range; AODV-THP, AODV with standard THP; and AODV with and without HELLO messages.

Fig. 13 presents the results for the packet delivery ratio. As expected, AODV-THP does not perform very well in scenarios with frequent topology changes. One of the main reasons is that it is more difficult to get an accurate view of the local topology when it changes more frequently. For static networks, AODV-THP delivers around 10% more packets compared to AODV. AODV-THP 1.0R shows the improvement due to the second enhancement by itself. It shows that it helps to compare any recent changes to the local topology to check if the sender is using any stale information (i.e., the last advertised *one-hop dominating list*, $D^i_{1-\mathrm{hop}}$, may not include some new *one-hop dominating node*) when computing the list of forwarders (i.e., $\mathscr{F}$). AODV-THP 1.0R starts performing better than AODV as mobility decreases (i.e., from 300 s pause time on), and it has the best results for static networks. Even though the topology actually does not change, because of transient link failures, and increased contention, the second enhancement helps to cope with transient changes to the local topology.

When mobility is present, we observe that the enhancements to THP improve the performance of AODV in all circumstances. It also shows that a VR of 0.85*RR* is better than 0.75*RR* for the scenarios under consideration. It means that a buffer zone of 0.15*RR* is enough to reach nodes moving out virtual range, and that it is better to keep more nodes within VR for purposes of computing THP.

Fig. 14 presents the average end-to-end delay results. Because around 580 flows are initiated during the simulation time, we observe that the

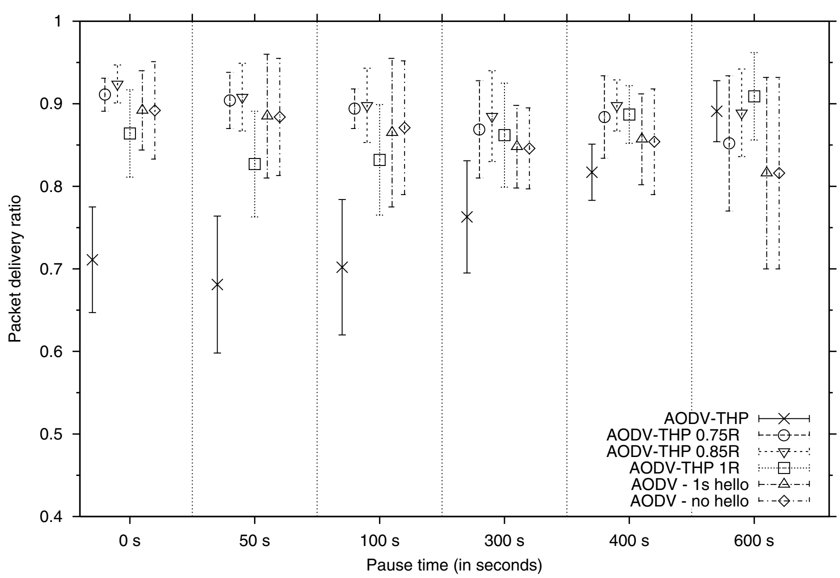

Fig. 13. 50 nodes (*directional reception*): packet delivery ratio.

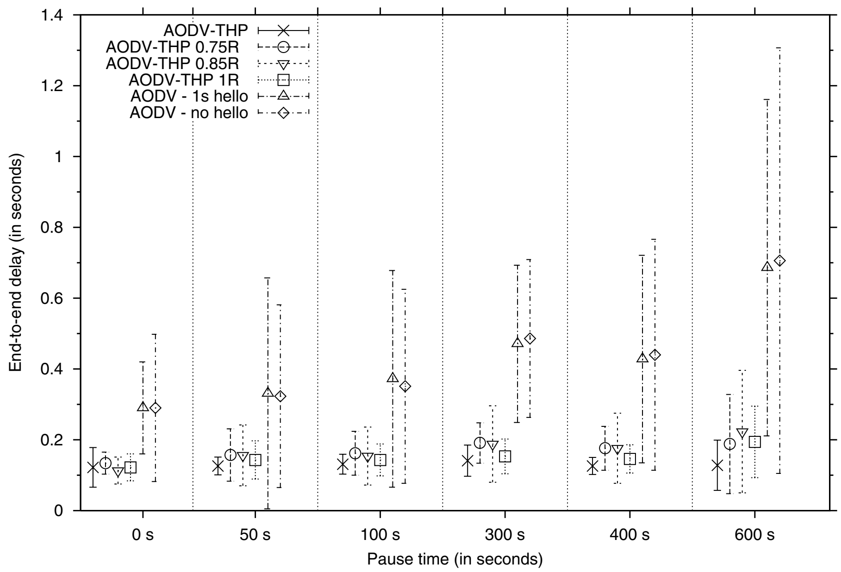

Fig. 14. 50 nodes (*directional reception*): average end-to-end delay.

large number of redundant RREQ transmissions affect the end-to-end delay in AODV. AODV incurs two to three times as much delay than any of the variants. Hence, pruning redundant broadcast transmissions pays off, because it reduces contention. The two enhancements are clearly effective for reducing delay, while at the same time keeping the delivery ratio high. The extra control overhead introduced by periodic HELLO messages does not impact the end-to-end delay much, because most of the routing load comes from RREQ transmissions. Together with the previous results for the

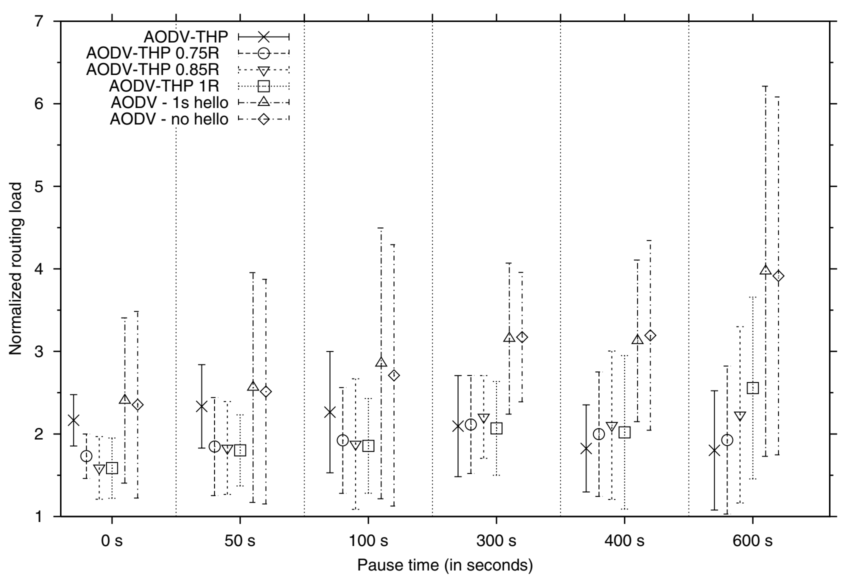

Fig. 15. 50 nodes (*directional reception*): normalized routing overhead.

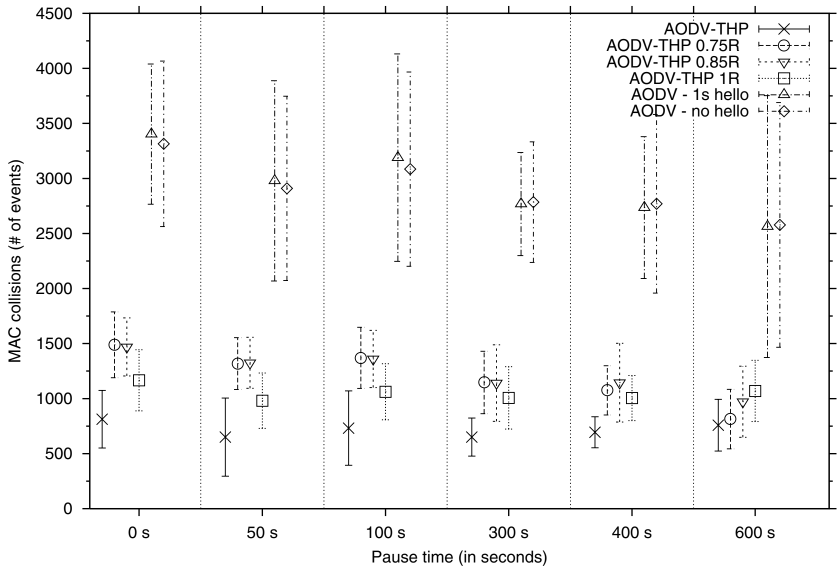

Fig. 16. 50 nodes (*directional reception*): number of MAC collisions.

delivery ratio, we can see that the reduction of redundant broadcast transmissions translate in a better and faster response to the route discovery process; consequently, more packets are delivered at a smaller cost.

Fig. 15 shows the normalized routing overhead results. All the THP variants present a much smaller overhead than AODV, because of the reduction on the number of redundant broadcast transmissions. As for the impact of the HELLO messages, in AODV we observe slightly more control overhead when HELLO messages are present. For static networks, AODV-THP presents the best cost effective performance; its delivery ratio is the second, and it has the smallest end-to-end delay and control overhead. But on the other hand, for high mobility scenarios, AODV-THP with the two enhancements show better performance than the other protocols.

Fig. 16 presents the number of collision of packets. AODV with and without HELLOs exhibit similar performance, which suggests that the introduction of HELLO messages is not responsible for increasing the number of collisions of packets. On the contrary, the extra RREQs are responsible for more contention and collision. For static networks, AODV-THP presents the best overall performance with the only exception of a slightly smaller delivery ratio than AODV-THP 1.0R. With mobility, even though the enhancements to THP incur slightly more collision of packets, they do improve the overall performance of the network by delivering more packets, with smaller delays, and less control overhead. Because there is a clear trade-off between efficiency and reliability, the two enhancements increase the reliability at the cost of increasing the number of redundant broadcast transmissions, but at the same time being efficient.

7. Conclusions

We presented THP, a localized algorithm for computing *two-hop connected dominating sets* (TCDS). In a TCDS, all nodes in the network are at most two-hops distant from some dominating node. We showed how THP can be applied to the route discovery process of on-demand routing protocols. The main contributions of THP are that (a) THP is the first heuristic to take into account three-hop information in the selection of relay nodes for the broadcasting of packets, while incurring signaling overhead that is much the same as that of heuristics based on two-hop information, and (b) THP reduces the number of redundant broadcast transmission. We show through extensive simulations that THP outperforms the best-performing self-pruning and neighbor-designated algorithms known when a TCDS is preferred over a CDS.

To improve the route discovery process of on demand routing protocols, THP is implemented in AODV (the new variant is named AODV-THP) as the mechanism for disseminating RREQ messages. The first simulation results show that THP improves, in all aspects, the performance of AODV in low mobility scenarios. We also show how to increase the reliability of THP (i.e., AODV-THP *2 cover*) by using *double coverage* instead of *single coverage* when computing the *one-hop dominating list*.

To address the lack of reliability in the presence of high mobility, we present two enhancements together with THP. First, a *virtual radio range* (VR), shorter than the physical *radio range* (RR), is used for gathering information about the two-hop neighborhood. Instead of using two different transmission powers, which can incur additional interference, we use a single transmission power while still managing to have a *buffer zone* in which neighbors can move without compromising network connectivity. Second, upon receiving a broadcast packet, the forwarder list in the packet header is analyzed together with the current information about the local neighborhood. This is done to find inconsistencies between the most up-to-date *one-hop dominating list* and the one used by the sender to compute the sender's forwarder list. Changes in the local topology may have impacted the *one-hop dominating list*. If that is the case, a node may decide to relay a broadcast packet even though it was not selected as a forwarder by the sender.

Extensive simulation results show that AODV-THP (*2 cover*) with the two enhancements attains better performance than AODV for all mobility scenarios in terms of delivery ratio, control overhead, packet collision, and end-to-end delay.

Acknowledgements

This work was supported in part by CNPq (Brazil), the Baskin Chair of Computer Engineering, and by the National Science Foundation under grant CNS-0435522.

References

[1] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, J.-P. Sheu, The broadcast storm problem in a mobile ad hoc network, Wirel. Netw. 8 (2002) 153–167.
[2] C. Perkins, Ad hoc on-demand distance vector routing, in: Second IEEE Workshop on Mobile Computing Systems and Applications, 1999.
[3] J. Wu, F. Dai, Mobility management and its applications in efficient broadcasting in mobile ad hoc networks, in: IEEE INFOCOM, 2004.
[4] M.M. Carvalho, J.J. Garcia-Luna-Aceves, A scalable model for channel access protocols in multihop ad hoc networks, in: MobiCom '04: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, ACM Press, 2004, pp. 330–344.
[5] B. Williams, T. Camp, Comparison of broadcasting techniques for mobile ad hoc networks, in: Proceedings of MOBIHOC, 2002, pp. 194–205.
[6] K. Obraczka, K. Viswanath, G. Tsudik, Flooding for reliable multicast in multi-hop ad hoc networks, Wirel. Netw. 7 (6) (2001) 627–634.
[7] H. Lim, C. Kim, Flooding in wireless ad hoc networks, Computer Communications 24.
[8] J. Wu, F. Dai, Broadcasting in ad hoc networks based on self-pruning, in: INFOCOM, 2003.
[9] M.R. Garey, D.S. Johnson, Computers and Intractability, Freeman, San Francisco, 1978.
[10] P.-J. Wan, K.M. Alzoubi, O. Frieder, Distributed construction of connected dominating set in wireless ad hoc networks, in: Proceedings of INFOCOM, 2002.
[11] T.W. Haynes, S.T. Hedetniemi, P.J. Slater (Eds.), Fundamentals of Domination in Graphs, Marcel Dekker, Inc., 1998.
[12] A.D. Amis, R. Prakash, T. Vuong, D. Huynh, Max–min d-cluster formation in wireless ad hoc networks, in: IEEE INFOCOM, 2000.
[13] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, L. Viennot, Optimized link state routing protocol for ad hoc networks, in: Proceedings of the IEEE International Multi-Topic Conference, 2001.
[14] W. Lou, J. Wu, On reducing broadcast redundancy in ad hoc wireless networks, IEEE Transactions on Mobile Computing 1(2).
[15] M.A. Spohn, J.J. Garcia-Luna-Aceves, Enhanced dominant pruning applied to the route discovery process of

on-demand routing protocols, in: Proceedings of the 12th IEEE ICCCN, 2003.
[16] M.A. Spohn, J.J. Garcia-Luna-Aceves, Enhancing the route discovery process of on-demand routing in networks with directional antennas, in: Proceedings of The Third IFIP-TC6 Networking Conference, 2004.
[17] J. Wu, H. Li, On calculating connected dominating set for efficient routing in ad hoc wireless networks, in: Proceedings of the third DiaLM, 1999, pp. 7–14.
[18] I. Stojmenovic, M. Seddigh, J. Zunic, Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks, IEEE Trans. Parall. Distrib. Syst. 13 (1) (2002) 14–25.
[19] F. Dai, J. Wu, Distributed dominant pruning in ad hoc networks, IEEE International Conference on Communications, vol. 1, 11–15 May 2003, pp. 353–357.
[20] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, in: Mobile Computing and Networking, 2001, pp. 85–96.
[21] J. Sucec, I. Marsic, An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks, CAIP Technical Report 248 – Rutgers University, September 2000.
[22] J. Wu, F. Dai, A generic distributed broadcast scheme in ad hoc wireless networks, in: 23rd ICDCS, 2003.
[23] J. Wu, F. Dai, Mobility-sensitive topology control in mobile ad hoc networks, in: IEEE IPDPS, 2004.
[24] S. Guha, S. Khuller, Approximation algorithms for connected dominating sets, in: European Symposium on Algorithms, 1996, pp. 179–193.
[25] Scalable network technologies, qualnet simulator. Available from: http://www.scalable-networks.com, 2003.
[26] C. Perkins, E. Royer, S.R. Das, M.K. Marina, Performance comparison of two on-demand routing protocols for ad hoc networks, IEEE Pers. Commun. 8 (1) (2001) 16–28.
[27] J. Hightower, G. Borriello, A survey and taxonomy of location systems for ubiquitous computing, Tech. Rep. UW-CSE 01-08-03, University of Washington, Computer Science and Engineering (2001).
[28] C. Perkins, E. Belding-Royer, S. Das, Ad hoc on-demand distance vector (aodv) routing, rfc 3561, July 2003.

Marco Aurelio Spohn received the B.S. degree in Computer Science from Universidade Federal do Rio Grande do Sul, Brazil, in 1996, an M.Sc. degree in Computer Science from University of California at Santa Cruz, USA, in 2002. He is currently a Ph.D. candidate at University of California at Santa Cruz. His research interests lie in broadcasting and routing in wireless ad hoc networks.

J.J. Garcia-Luna-Aceves received the B.S. degree in Electrical Engineering from Universidad Iberoamericana (“La Ibero”), Mexico City, Mexico in 1977. M.S. and Ph.D., Electrical Engineering, University of Hawaii at Manoa in 1980 and 1983, respectively. He holds the Baskin Chair of Computer Engineering. Prior to joining UCSC in 1993, he was a Center Director at SRI International in Menlo Park, California. He first joined SRI as an SRI International Fellow in 1982. He was a Visiting Professor at Sun Labs in Menlo Park, California in 1999, and was a Principal of Protocol Design for NOKIA from 1999 to 2003. He is a Principal Scientist at the Palo Alto Research Center (PARC). He has coauthored the book Multimedia Communications: Protocols and Applications (Prentice-Hall), and has published six patents and more than 270 papers on computer communication in journals and conferences. His current research interest is the analysis and design of algorit hms and protocols for computer communication. At UCSC, he leads the Computer Communication Research Group (CCRG), which is home to many research projects that focus on wireless networks and internetworking, and which are currently sponsored by the National Science Foundation (NSF), the UC Office of the President (UCOP), the Defense Advanced Research Projects Agency (DARPA), the US Army Research Office (ARO), the US Air Force Office of Scientific Research (AFOSR), and industry.