Computer Communications 34 (2011) 1169–1181

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Realizing and benchmarking broadcast algorithms in wireless mesh networks

Ying-Dar Lin [a,b], Shun-Lee Chang [a,b,*], Shi-Hung Tao [a], Jui-Hung Yeh [b]

[a] Department of Computer Science, National Chiao Tung University, Taiwan
[b] Realtek-NCTU Joint Research Center, Taiwan

ARTICLE INFO

Article history:
Received 11 March 2010
Received in revised form 26 July 2010
Accepted 26 November 2010
Available online 3 December 2010

Keywords:
Broadcast algorithm
Multi-hop wireless networks
Mesh networks
802.11s
WMN

ABSTRACT

Broadcasting by flooding causes the broadcast storm problem in multi-hop wireless networks. This problem becomes more likely in a wireless mesh network (WMN) because WMNs can bridge wired LANs, increasing broadcast traffic and collision probability. Since the network control, routing, and topology maintenance of a WMN highly rely on layer-2 broadcasting, unreliable broadcast algorithms directly destabilize a WMN. Researchers have developed many algorithms for efficient and reliable broadcast in multi-hop wireless networks. However, real-world systems rarely verify or compare these approaches, especially in a WMN. This paper examines six representative broadcast algorithms: *simple flooding*, *dynamic probabilistic*, *efficient counter-based broadcast*, *scalable broadcast*, *domain pruning*, and *connected-dominating-set based* algorithms. This study addresses both common and algorithm-specific implementation in a real-world IEEE 802.11s WMN testbed. Experiments under various topologies and packet lengths reveal the reliability, forwarding ratio, and efficiency of these six algorithms. Quantitative survey results indicate that the scalable broadcast algorithm possesses the best reliability due to its lower collision probability. The domain-pruning algorithm is the most efficient algorithm when considering both reliability and the forwarding ratio.

1. Introduction

The *broadcast storm problem* refers to the serious redundancy, contention, and collision created by flooding broadcast packets in multi-hop wireless networks [1]. Fig. 1(a) illustrates this problem by depicting an entire network flooded with unnecessarily duplicate broadcast packets due to the overlapped coverage of relay nodes. Flooding also increases the likelihood of contention and collision, as multiple nearby stations simultaneously forward broadcast packets in a short period. A broadcast storm can also drastically hinder the transmission and routing of unicast data, decreasing the reliability of a multi-hop wireless network [2].

The broadcast storm problem is more likely to occur in wireless mesh networks (WMNs). As Fig. 1(b) shows, a WMN is a multi-hop wireless network consisting of both wireless ad hoc networks and infrastructure-based wireless networks.[1] The entities (called mesh nodes or nodes for short) in the ad hoc plane include mesh stations (MSTAs), which provide traffic-relaying functions for other mesh nodes, mesh access points (MAPs), which provide wireless services to non-mesh stations (STAs), and mesh portal (MPPs), which attach the WMN and wired LANs. The infrastructure plane is the area covered by a MAP to serve non-mesh STAs. The ad hoc plane forwards the traffic on the infrastructure plane, which increases the likelihood of collision when the two planes share the same channel and overlap their coverage area. Moreover, the attached wired LANs flood broadcast-based network-controlling protocols, such as the address resolution protocol (ARP) [6] and the spanning-tree protocol (STP) [7], into the WMN via MPPs. Finally, the network control, routing, and topology maintenance in a WMN also rely heavily on layer-2 broadcasting [8]. As a result, the broadcast storm problem is more likely to occur without a carefully designed broadcast algorithm, which in turn decreases the reliability of the WMN.

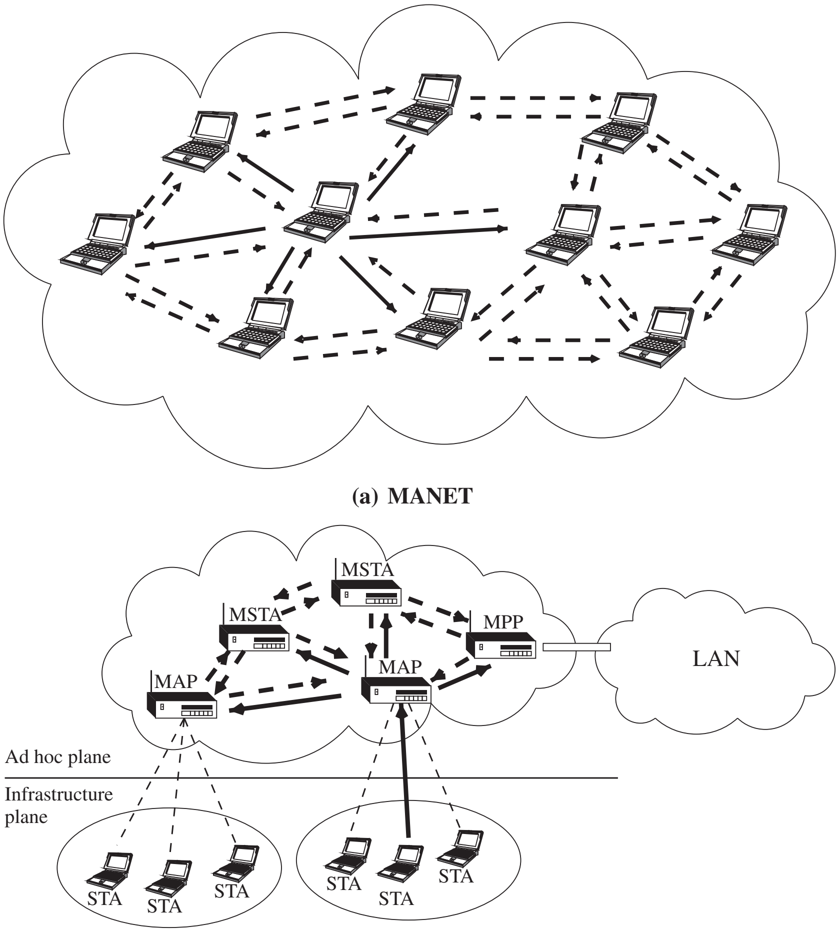
This study classifies previous work that addresses efficient and reliable broadcasting in wireless multi-hop networks into three categories based on their primary characteristics as Table 1 shows. First, nodes in the *straightforward* category immediately rebroadcast a received broadcast to all their neighbors either at all times [9] or following a predefined probability [1]. This approach has the advantages of simple implementation, but still leads to the broadcast storm problem. Nodes in the second category decide to *delay* rebroadcasting for a short period. Each node decides when to retransmit a broadcast according to various observations in this short time period. These observations include the density of neighbors [10], number of received duplicate packets [1,11], and the

* Corresponding author at: Department of Computer Science, National Chiao Tung University, Taiwan. Tel.: +886 3 571 2121x56667.

E-mail address: changsl@cs.nctu.edu.tw (S.-L. Chang).

[1] As described in [3–5], researchers typically classify a WMN into several network architectures according to node mobility. This paper only focuses on the most common architecture, Infrastructure/Backbone WMN [3].

0140-3664/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2010.11.011



(a) MANET

(b) WMN

Fig. 1. Broadcast problems in the (a) MANET and (b) WMN.

Table 1
Broadcast algorithm categories.

Idea	Design unit	Method	Publications
Straightforward	Single	Flooding Probabilistic	Ho et al. [9] Ni et al. [1]
Delay	Single	Dynamic probabilistic Counting-based Distance sensing	Zhang and Agrawal [10] Ni et al. [1], Mohammed et al. [11] Sun et al. [12], Li et al. [13]
Neighbor information	Single Cluster	Self pruning Self pruning/domain pruning Neighbor union/gateway selection	Peng and Lu [14] Lim and Kim [23], Lim and Kim [15], Lou and Wu [16], Shen et al. [17] Wu and Li [18], Qayyum et al. [19], Ingelrest and Simplot-Ryl [20], Keshavarz-Haddad et al. [21], Hasegawa et al. [22], Lou and Wu [2]

signal strength [12,13]. The advantage of this approach is that each node works independently without exchanging information with its neighbors. As a result, this approach does not introduce additional protocol overhead due to broadcasting. The last approach uses *neighbor information* to determine retransmission. With the help of neighbor information, each node can prune unnecessary rebroadcasting itself [14], or a group of nodes cooperatively decide how to retransmit a broadcast [2,15–22].

Previous researchers have compared various broadcast algorithms using simulation experiments. Williams and Camp [24] evaluated five algorithms through simulations, and pointed out that algorithms using neighbor information perform better than other algorithms. Similarly, Oliveira et al. simulated the operation of six algorithms [25]. Their results show that, in an error-prone scenario, i.e., low signal to interference and noise ratio (SINR) environment, inaccurate neighbor information updates can mislead neighbor information-based algorithms. They also concluded that redundant broadcasting is necessary to ensure the broadcast reliability.

However, without sufficient information on transmission mechanisms [26], link stability [27] and transmission reliability [27–29], simulation results may be quite different from real-world results [28,29]. Moreover, simulations cannot reveal resource consumption and computation complexity, which are important factors in realizing these algorithms. Furthermore, all the above broadcast algorithms were originally designed for wireless mobile ad hoc networks (MANETs) and must be revised for use in WMNs. Unlike MANET nodes, mesh nodes usually have *managed topologies* and *sufficient power* [3,8]. In other words, the issues of mobility and power consumption are not critical to a WMN.

This study examines six representative broadcast algorithms: *simple flooding* [9], *dynamic probabilistic broadcast* [10], *efficient counter-based broadcast scheme* (*ECS*) [11], *scalable broadcast algorithm* (*SBA*) [14], *domain pruning algorithm* [16], and a *connected-dominating-set* (*CDS*) based algorithm [18]. Simple flooding is the most common broadcast method for the off-the-shelf products. Dynamic probabilistic and ECS adopt the concept of delayed retransmission. The other three algorithms use neighbor information. We selected these algorithms for analysis because of their outstanding performance in previous simulations [25]. Unlike prior studies [24,25], which compare broadcast algorithms with simulation for MANETs, we conducted a *quantitative* survey based on the experiment results of a *real-world* WMN. We also developed a generic system architecture to realize these algorithms on a testbed, and identified the implementation issues for each algorithm.

The rest of this study is organized as follows. Section 2 introduces the background. Section 3 describes the implementation model and lists the solutions to various implementation issues. Section 4 presents experiment results and observations. Finally, Section 5 offers a conclusion and directions for future research.

2. Background

This section first describes the six algorithms implemented in this paper in detail, and then illustrates the latest literals correlating to WMN broadcasting.

2.1. Selected broadcast algorithms

This subsection provides an overview of the six algorithms implemented and evaluated in this study. Besides the most straightforward implementation, i.e., the simple flooding, the dynamic probabilistic and efficient counting broadcast algorithms represent the probabilistic and counting-based approaches, respectively. We also chose the scalable broadcast algorithm for its efficient use of neighbor information and self-pruning characteristics. Finally, we chose domain pruning algorithm and CDS-based algorithm to represent the gateway selection mechanism.

- *Simple flooding*: Flooding is the most basic and straightforward solution for broadcasting in a wireless multi-hop environment. After receiving a new broadcast, i.e., a packet never seen before, each node immediately rebroadcasts the packet to its neighbors. This process continues until all reachable nodes have received the broadcast. Most WMNs adopt this approach because it is so simple. However, previous research [1] indicates that the flooding approach leads to the broadcast storm problem including high redundancy, contention, and collision.
- *Dynamic probabilistic algorithm*: Zhang and Agrawal proposed the dynamic probabilistic algorithm [10], which combines the probabilistic scheme with the density information of neighboring nodes. The original probabilistic scheme is similar to the simple flooding method, except that a node only retransmits a broadcast with a predefined probability P. It is easy to show that the original scheme reduces broadcast overheads in a dense network, but performs worse in sparse networks. The dynamic probabilistic algorithm proposed in [10] *adjusts* the P value by considering both the *density* of neighbor nodes and the number of received duplicate broadcast packets. This algorithm decreases retransmission probability P for nodes in dense areas, and increases probability P for nodes in sparse areas.
- *Efficient counter-based broadcast scheme (ECS)*: Mohammed et al. proposed the ECS algorithm [11], which combines the probabilistic scheme with a counter for rebroadcasts received during a short interval. Unlike the dynamic probabilistic algorithm, ECS uses a *fixed* probability P that only takes effect when the counter is smaller than a predefined threshold. Otherwise, nodes will not forward a broadcast, as a higher counter value indicates that all its neighbors have likely received the broadcast.
- *Scalable broadcast algorithm (SBA)*: Peng and Lu proposed the SBA [14], which follows a simple idea that a node requires no rebroadcasting if all neighboring nodes have already received the broadcast. This algorithm operates in two phases: neighbor information discovery and packet forwarding. In the neighbor information discovery phase, each node broadcasts a HELLO announcement describing a list of one-hop neighbors. After receiving the HELLO messages from all the neighbors, a node can build a neighbor table that includes two-hop neighbor information. In the packet forwarding phase, a node stops rebroadcasting if the broadcasting range of a received broadcast (by checking its own neighbor table, i.e., a self-pruning process) has already covered all neighboring nodes. To reduce the collision and increase efficiency, SBA adjusts the maximum backoff interval to prioritize the node with the most neighbors (obtained from the two-hop neighbor information) a higher broadcast probability.
- *Domain pruning (DP) algorithm*: The DP algorithm also uses two-hop neighbor information to make routing decisions [15]. The DP concept is that only the smallest subset of one-hop neighbors, called the gateway nodes, can rebroadcast a broadcast packet. The DP algorithm assumes that one-hop broadcasting is reliable. Therefore, after receiving a broadcast from a neighbor t, a broadcasting node r first computes a gateway-node list from its one-hop neighbors $N(r)$. The nodes in the gateway-node list must be able to reach all two-hop nodes $N(N(r))$ not covered by the broadcasting of t (i.e., $N(N(r)) - N(r) - N(t)$). The DP algorithm then piggybacks the list in the broadcast packet, and schedules retransmission.
- *Connected-dominating-set (CDS) based algorithm*: A CDS is a connected subset of vertices in a graph in which any vertex not in the subset can connect to at least one vertex in the subset. Wu and Li proposed the first CDS-based algorithm for broadcasting [18]. The CDS-based algorithm combines self-pruning and domain-pruning mechanisms. Instead of choosing the gateway nodes on demand, as in DP, the CDS-based algorithm proactively maintains a local CDS, based on two-hop neighbor information. Gateway nodes are the nodes in the local CDS. This approach effectively reduces the number of forwarding nodes to the size of CDS. Notably, the CDS-based algorithm is so popular that it is standardized as Optimized Link State Routing Protocol (OLSR) [30]. The early draft of IEEE 802.11s also adopted OLSR as an optional routing protocol, but abandoned the protocol shortly due to its high computation complexity in implementation [31]. Many researchers have attempted to improve OLSR, and the most popular method is B.A.T.M.A.N [32]. This study does not choose OLSR or B.A.T.M.A.N, because these implementations are based on the User Datagram Protocol (UDP) layer, which is different from our layer-2 mesh assumption.

2.2. Current researches of broadcasting in WMN

Many researchers have examined WMN broadcasting methods in recent years. Chou et al. [33] showed that multi-rate broadcasting is unlike conventional single-rate broadcasting. They devised a centralized scheduling algorithm based on a multi-rate broadcast tree[2] to achieve low latency broadcast. Wang et al. [34] proposed a

[2] A broadcast tree is a rooted CDS where the root is the broadcast initiator.

fast multi-rate broadcast tree construction algorithm by considering the transmission range of each rate. Xin and Zhang [35] studied the construction of a broadcast tree in a wireless environment with failed links. Nguyen and Nguyen [36] proposed a channel assignment algorithm for multi-channel multi-radio WMNs that enables minimum interference broadcasting. Chiu et al. [37] considered the channel assignment problem when constructing the broadcast tree, and Song et al. [38] discussed the radio selection issues in a multi-radio node to forward a received broadcast. Finally, Koutsonikolas et al. [39] designed a TDMA-like MAC protocol that assigns nodes to different time slots to reduce the interference created by broadcasting.

This study does not compare the methods described in the studied above, because they adopt different wireless assumptions. The literals [33,34] assume the multi-rate broadcasting, but in the real world, the typical broadcast rate is 1 Mbps, as this is the most common data rate agreed upon all devices. Though the fault tolerant issue [35] is significant in a MANET, it is less important in a WMN due to the assumption of a robust topology and sufficient power of mesh nodes. Moreover, the multi-channel multi-radio design [36–38], except the two-radio design, is rare in current commercial products due to its high complexity. Finally, the TDMA-like MAC protocol [39] is incompatible with IEEE 802.11 CSMA/CA.

3. System design and algorithm implementation

As Section 1 mentions, simulation results do not necessarily match real-world experimental results. This is because simulations tend to ignore several important implementation issues, or hides them in assumptions. This section presents the implementation details of a real-world testbed. This section first introduces the system architecture of the testbed, then describes the generic and algorithm-specific implementation issues, and finally lists observations from this implementation.

3.1. System architecture

We adopted IEEE 802.11s [8] as the wireless mesh environment. The IEEE 802.11s amendment defines a wireless LAN mesh using IEEE 802.11 MAC/PHY layers, and is one of the most active mesh standards. Our testbed is based on a platform operating the embedded Linux (derived from the Linux kernel version 2.4.18). Fig. 2 depicts the system architecture of our design. In the driver layer, the IEEE 802.11/802.11s driver not only implements the functions for IEEE 802.11 MAC, but also supports the mesh functionalities in the IEEE 802.11s amendment. Within the driver, the neighbor table records the neighbor information to maintain the mesh topology. The proxy table and path selection table store routing information for nodes in the infrastructure plane and ad hoc plane, respectively. The interface layer multiplexes two virtual interfaces, *wlan0* and *mesh0*, to serve both IEEE 802.11 networks and IEEE 802.11s networks concurrently on a single physical wireless adaptor. To bridge the Ethernet traffic, an additional Ethernet adaptor collaborates with the wireless adaptor on the same platform via a virtual bridging interface, *br0*. In the user space, a Linux daemon program, called Path Selection Daemon, implements the mesh routing algorithm and updates the three tables in the driver layer.

Fig. 3 illustrates the packet-processing flow in the proposed design. The main goal of this design is to create a common packet-processing framework in which the implementation of each broadcast algorithm is independent of the mesh functions. After the hardware receiver (Wireless Rx) receives a packet and validates its sequence number and Time-To-Live field (TTL), it places the packet in the data queue (for data packets) or the management queue (for management packets). The system processes unicast and broadcast data separately. A unicast data packet is passed to the transmission queue if it must be forwarded. On the other hand, broadcast packets enter the black-box of the broadcast algorithms. The black-box determines whether or not to forward the broadcast. If forwarding is necessary, the system places the packet in the transmission queue. The hardware transmitter (Wireless Tx) then transmits the packets when it grabs the channel. The Path Selection Daemon shown in Fig. 2 provides the necessary black-box parameters, including two-hop neighbor information, after processing the management packets. Obviously, this design is flexible enough to support various broadcast algorithms.

The design can be extended to support multicast transmissions, because the address formats of both broadcast and multicast are

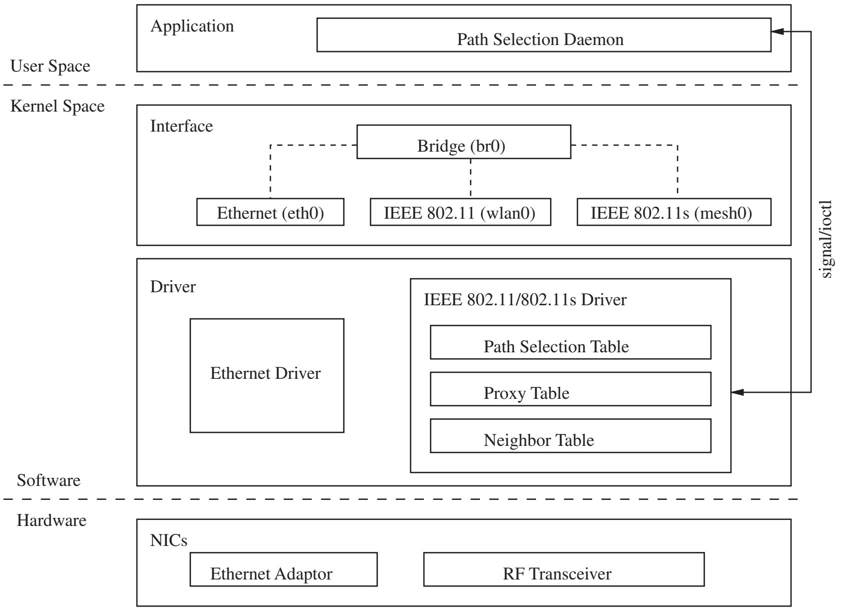


Fig. 2. System architecture.

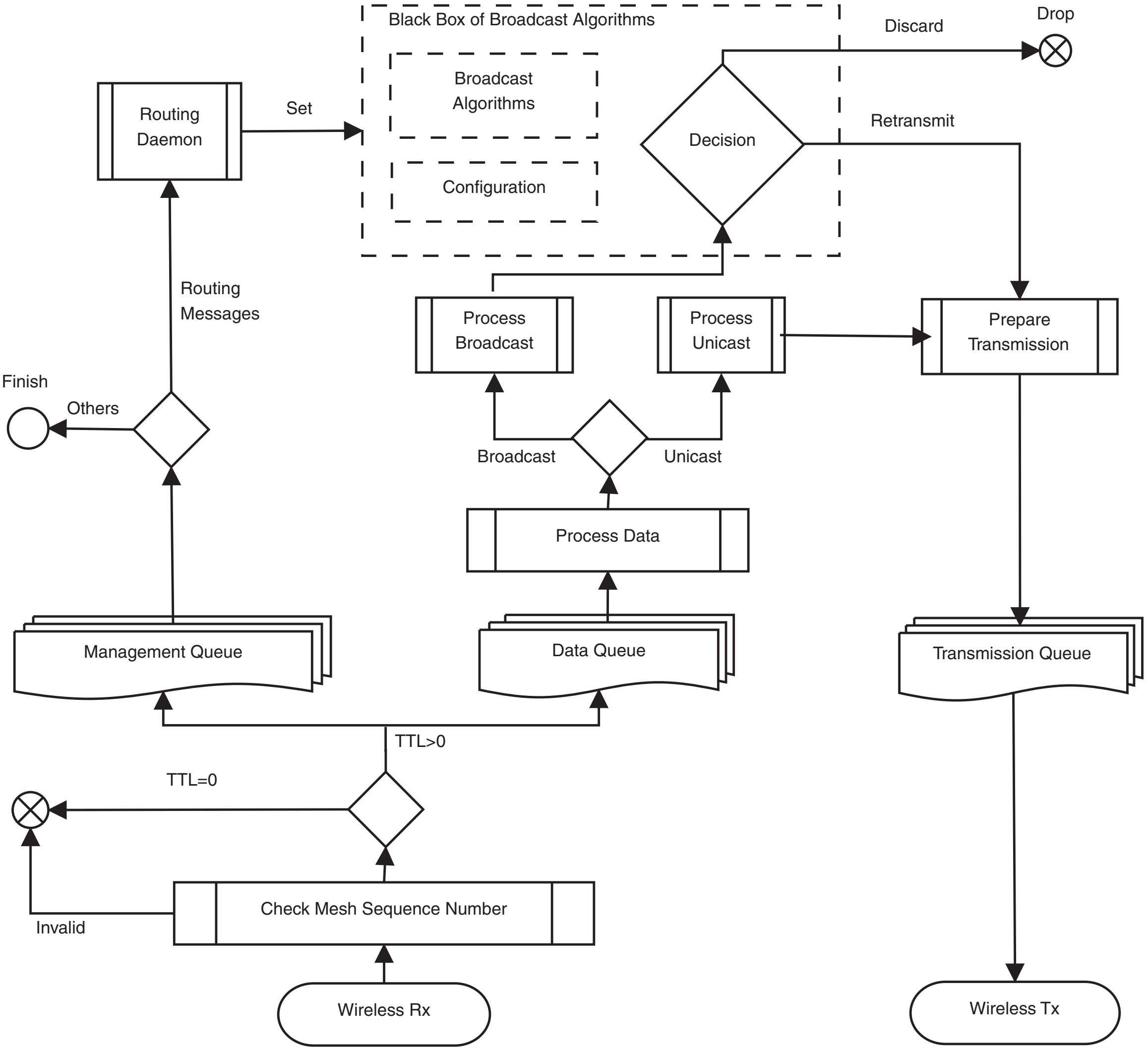


Fig. 3. Packet processing flowchart.

similar, i.e., the first bit is 1, in IEEE 802.11. The system must check a table, called multicast address table, before routing multicast data. The detail of processing multicast data is beyond the scope of this work.

3.2. Generic implementation issues

This subsection discusses three common implementation issues for the algorithms examined in this study.

1. *Duplicate packet validation*: A mesh node that blindly rebroadcasts packets to its neighbors may lead to infinite flooding in the loop topology. Therefore, a validation mechanism is necessary to detect duplicate broadcasts. To avoid an infinite broadcast loop, IEEE 802.11s embeds a unique Mesh-Sequence-Number (MSEQ) field in each packet. The combination of $\langle$source MAC, MSEQ$\rangle$ is as a unique signature that allows a relaying node to detect duplicate packets. Therefore, a system must implement a buffering method, e.g., the *Check Mesh Sequence Number* module in Fig. 3, to store and check the signature.
2. *Two-hop neighbor information collection*: SBA, DP, and CDS algorithms need the two-hop neighbor information. Our implementation uses the HELLO messages to help a node collect the information. The HELLO message in our testbed is a single-hop, i.e., $TTL = 1$, broadcast packet piggybacking a list of one-hop neighbors. Each node periodically broadcasts a HELLO message and stores the HELLO messages from neighbors. The two-hop neighbor information can be learned by collecting all the HELLO messages from one-hop neighbors.
3. *Gateway notification*: Several algorithms piggyback on-demand information on the data packets. For example, the DP algorithm piggybacks gateway information on each data packet. However, piggybacking makes the data packets incompatible with standard and therefore more likely to be dropped by a node not supporting the algorithm. As Section 1 mentions, mesh nodes have lower mobility than the MANET nodes, so the gateway information is stable and not necessary to be embedded in each data packet. In our implementations, therefore, the gateway information is only embedded in the periodical HELLO messages. The solution is simple and effective, and does not hinder the operation of a wireless system. This is because a node that does not support a specific algorithm can completely ignore the HELLO message.

3.3. Algorithm implementation

The six algorithms in this study have various requisite parameters and execution procedures. To easily implement these parameters and procedures in a node, the subsection presents a three-phase execution flow that unifies common requirements while allowing differences among algorithms.

1. *Periodical task*: The periodical task initializes algorithm-specific parameters, periodically announces the HELLO messages, and prepares one-hop and two-hop neighboring information. During the periodical task, the dynamic probabilistic algorithm updates the probability value whenever associating or disassociating a neighbor. The SBA, DP, and CDS-based algorithms exchange HELLO messages to retrieve neighbor information. The DP algorithm also selects its neighbors as gateways in this phase. A node using the CDS-based algorithm constructs the local CDS based on the received HELLO messages, and decides whether or not it is a gateway based on the local CDS.
2. *Observation phase*: In the observation phase, each node uses received broadcast packets to update the run-time parameters for a specific broadcast algorithm. In particular, the dynamic probabilistic algorithm counts the number of received broadcasts in a time period, and uses this value to adjust the probability value. The ECS and SBA algorithms store a broadcast and set a delay timer for retransmission decision. Besides setting the timer, the SBA algorithm updates the remaining neighbor list (RNL), which records the neighboring nodes that still have not received the broadcast. The DP and CDS-based algorithms skip this phase and enter the determination phase directly.
3. *Determination phase*: During the determination phase, each algorithm checks the algorithm-specific conditions, such as the received broadcast counter in ECS, to determine whether or not to forward a broadcast.

Table 2 presents the actions for each algorithm in each phase.

3.4. Observations

This section summarizes observations from implementing the broadcast algorithms.

1. *HELLO overhead:* HELLO messages may frequently occupy the channel and compete with normal data transmissions. This is because the HELLO messages are also implemented as broadcast packets, whose transmitting data rate is usually 1 Mbps. As a result, HELLO messages consume a lot of bandwidth. Fortunately, only the periodical task issues HELLO messages, so decreasing the task frequency can reduce the overhead caused by exchanging HELLOs.
2. *Timer resolution:* Delay-based algorithms such as ECS and SBA need a timer with a higher resolution than the default value. The Linux kernel limits the minimum timer interval, which is 10 ms by default. Compared with the interval of transmitting a lengthy data packet (1573 ms to transmit 1500 bytes Ethernet payload at 11 Mbps in IEEE 802.11b), the rebroadcasting delay with the default value is apparently too long. Reconfiguring this value by recompiling the Linux kernel can reduce the delay, but this increases side effects, such as the context switching overhead. Therefore, we suggest a hardware solution, such as using network co-processors, for implementing the observation phase in delay-based algorithms. Because we only address the software implementation issues, the detail of hardware solution is beyond the scope of this work.
3. *Buffer size:* Delay-based algorithms need a buffer to store broadcasts during the observation phase. An insufficient buffer causes data loss, while a larger buffer wastes memory resources. Apparently, the longer delay interval is, the larger buffer size is required. The buffer size is also proportional to the traffic arrival rate. As a result, buffer size design should consider both the delay interval and the traffic arrival rate.

4. Evaluation

This section presents the experimental results from our testbed. To conduct experiments with different parameters and scenarios, this section introduces a generic evaluation and logging mechanism, describes the experimental environment and the evaluation metrics, and finally presents the results and observation.

4.1. Evaluation and logging mechanism

Fig. 4 depicts the execution of the proposed evaluation and logging mechanism. This mechanism defines three roles for each node in the experiment: sender, receiver, and logger. A sender initializes experiments, generates various broadcast patterns, and collects experimental results. A logger is a program collocated with a mesh node. The logger program records the statistics from the running algorithm in each experiment and reports the results to the sender. A receiver acts as an end station, and identifies the reliability from the user's viewpoint.

At the beginning of an experiment, the *experiment round control* (*ERC*) module in the sender transmits a STARTUP command to all nodes to initialize the algorithmic parameters and reset the execution of the periodical task module. The ERC module then generates different broadcasting patterns. At the end of an experiment, the ERC module transmits a FINISH command to each node. After receiving the FINISH signal, the logger modules on all nodes transmit their statistics to the *log collection* module in the sender and reset the database for the next experiment.

4.2. Experiment environments and evaluation metrics

We implemented the experimental mesh node on a Realtek™ RTL8186 platform. The RTL8186 platform is a commercial

Table 2
Three phases of each algorithm.

Algorithms	Periodical task	Observation phase	Determination phase
Dynamic probabilistic algorithm	Initiate the probability value	Count the number of same broadcast packets	Probability
ECS	N/A	1. Queue the new broadcast packet 2. Set delay timer 3. Count the number of same broadcast packets	Counting number and probability
SBA	Retrieve neighbor info.	1. Queue the new broadcast packet 2. Initiate the RNL 3. Set delay timer 4. Update the RNL	Non-empty RNL
DP	1. Retrieve neighbor info. 2. Assign the gateways for each neighbor	N/A	Gateway of the sender
CDS-based algorithm	1. Retrieve neighbor info 2. Construct local-CDS 3. Decide gateway property	N/A	Gateway property

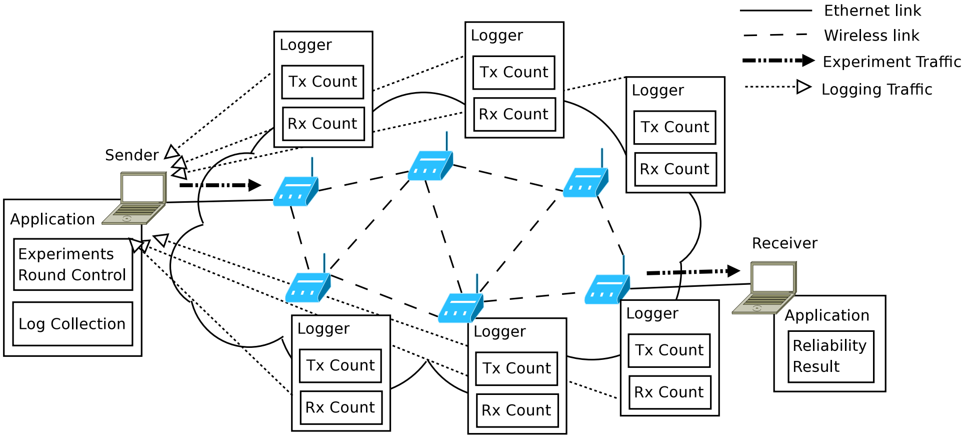
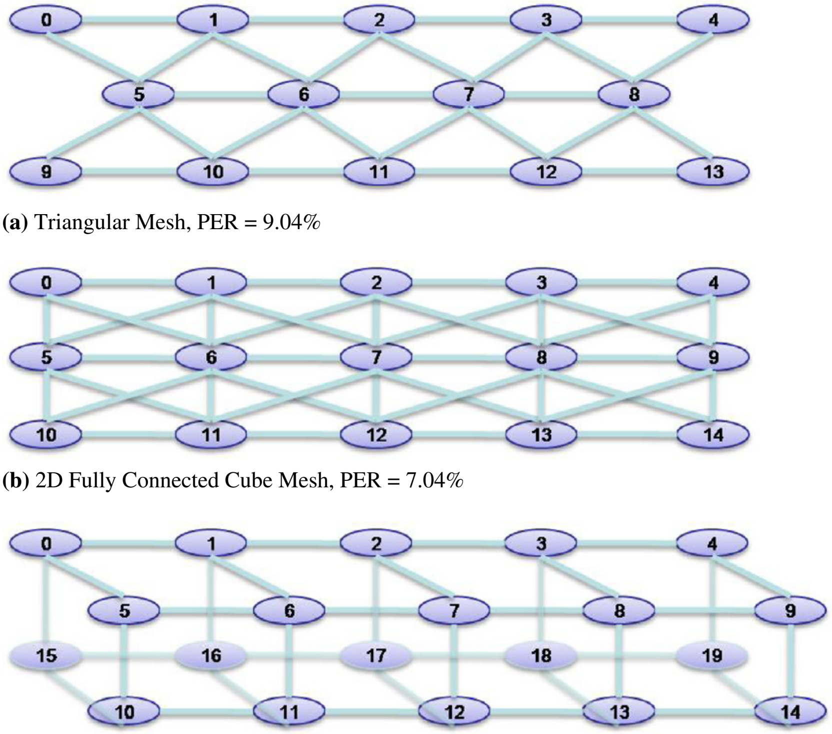


Fig. 4. Evaluation and logging mechanism.



(a) Triangular Mesh, PER = 9.04%

(b) 2D Fully Connected Cube Mesh, PER = 7.04%

(c) 3D Cube Mesh, PER = 9.01%

Fig. 5. Evaluation scenarios.

system-on-a-chip solution that integrates an Ethernet controller, single-radio 802.11b/g controller, and a 180 MHz 32-bit RISC processor compatible with MIPS R3000. We conducted all experiments in an IEEE 802.11s-based wireless mesh environment where all mesh nodes share one common channel for packet transmissions. A fixed attenuator attached to the antenna of each node regulates its transmission power. This greatly reduces the actual space required for the experimental testbed, and makes it possible to conduct small-scale mesh experiments in the laboratory.[3] Each evaluation generates 5000 broadcast packets. The inter-arrival time of the broadcast packets is 100 ms. In addition, the transmission rate for broadcast packets is 1 Mbps, following off-the-shelf WLAN solutions.

As Fig. 5 shows, we developed three general experimental grid scenarios: (a) the triangular mesh, (b) 2D fully connected cube mesh (2D mesh), and 3D cube mesh (3D mesh). The first two scenarios represent general grid structures, while the third scenario emulates the deployment in a building. For each scenario, the smallest and the largest numbered nodes are two MPPs attached to two LANs. The sender and receiver are located on these two LANs. The other nodes in the scenario are MSTAs. The distance between two neighboring nodes is 60 cm. The Packet Error Rates (PER) caused by the attenuators and collisions during our testing were observed as 9.04%, 7.04% and 9.01% for each scenario.[4]

Our experimental topologies, i.e., grid topologies, cannot represent all cases of the real-world deployment. However, those

[3] Notably, the properties of multi-path fading in our laboratory trials are unlike the results obtained from the field.

[4] In a small experimental space, the transmission range is hard to measure due to the sensitivity of the physical-layer components. Instead of measuring the transmission range, therefore, we observed the PER during experiments.

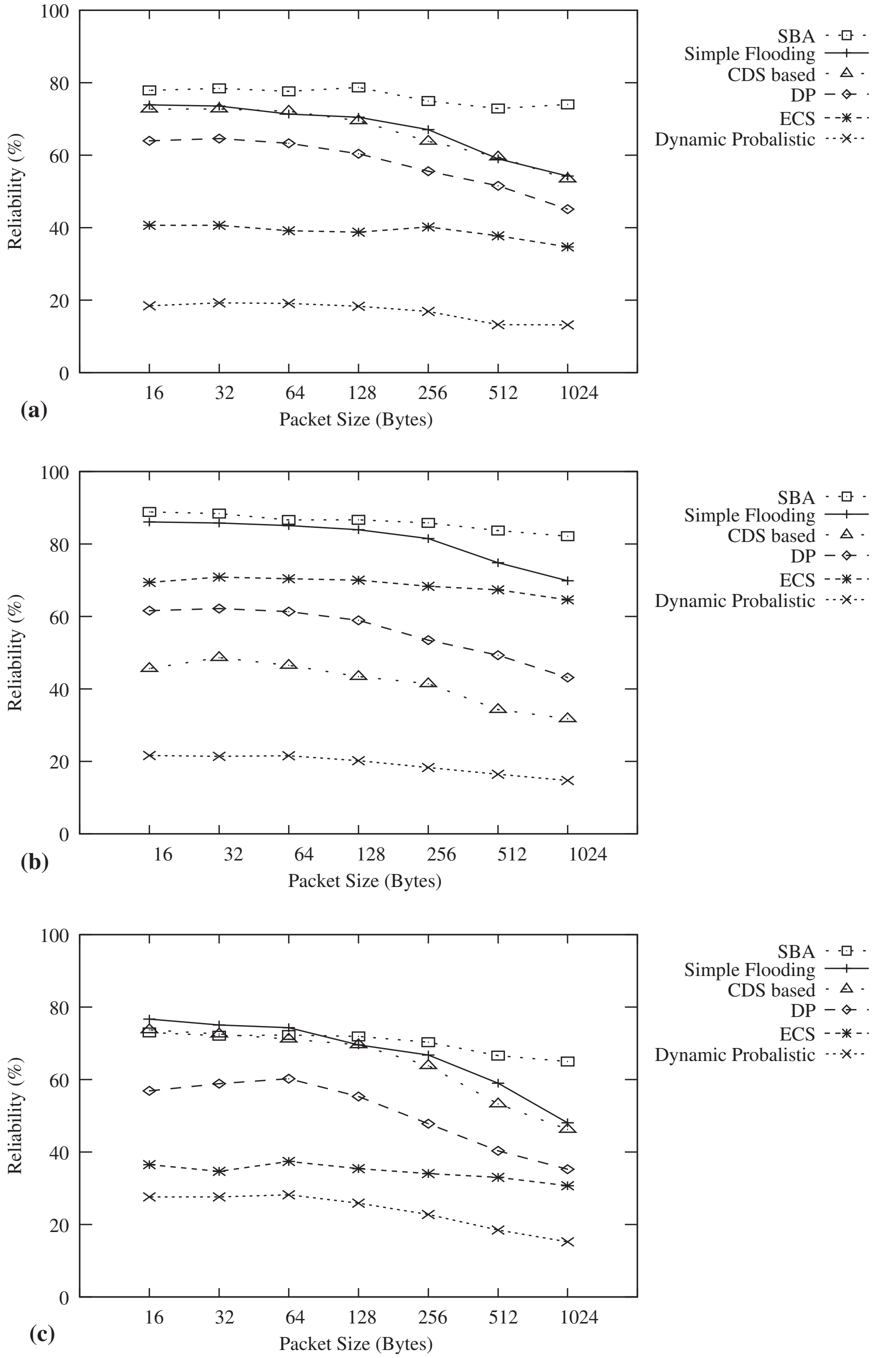


Fig. 6. Longest path reliability of (a) the triangular mesh; (b) the 2D mesh and (c) the 3D mesh.

topologies are chosen for three reasons. First, for large-scaled mesh deployment, Robinson and Knightly [40] showed that the grid topologies are more suitable than the random placement when considering the deployment cost. Second, a non-random experimental topology is easier for researchers who are interested in our work to reproduce in the real-world testbed or by simulation. Finally, for real-world testbed, the control and management of nodes can be simplified when using the grid deployment [41].

This study uses four metrics to evaluate the studied broadcast algorithms: *longest path reliability*, *average reliability*, *forwarding ratio*, and *broadcast efficiency*. We calculated longest path reliability by dividing the number of broadcast packets successfully received at the farthest mesh node by the number of broadcast packets generated by the source. The metric is an index of the availability for supporting broadcasting-based services, such as ARP, across a mesh, as the metric considers the receive status at the farthest node. The average reliability is the mean ratio of broadcast packets successfully received by all mesh nodes. The metric is an index of the case where a server advertises important service information to multiple clients. For instance, an MPP in the 802.11s mesh network periodically announces its existence too all nodes by broadcasting a portal announcement message. We computed the forwarding ratio by dividing the number of forwarding instances by the number of received broadcast packets. Obviously, larger ratio implies higher wireless media occupation.

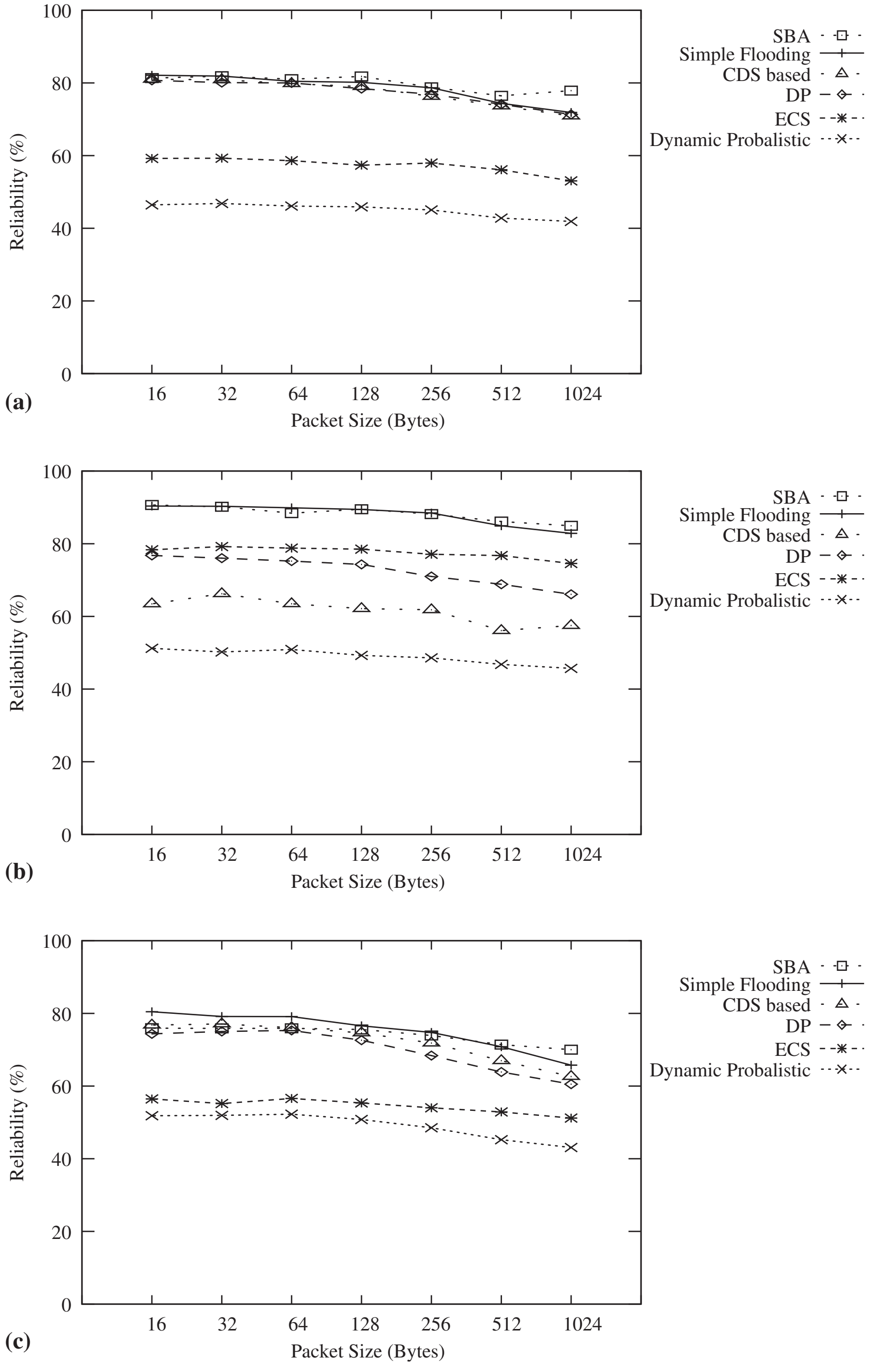


Fig. 7. Average reliability of (a) the triangular mesh; (b) the 2D mesh and (c) the 3D mesh.

Finally, we define *broadcast efficiency* to identify the most efficient broadcast algorithm. A broadcast algorithm is efficient when it delivers a broadcast packet to as many nodes as possible while utilizing minimal wireless media resources. We defined the following variables to evaluate broadcast efficiency:

- $N(i)$: number of neighbors of node i.
- $R(i)$: number of effective, i.e., correct and non-duplicated, broadcast packets received by node i.
- $F(i)$: number of broadcast packets generated or forwarded by node i.
- FE: forward efficiency.
- BE: broadcast efficiency.
- AR: average reliability.

Forward efficiency represents the *effectiveness* of the retransmitted broadcast packets. Thus, most of the retransmitted broadcast packets are effective in an algorithm with the high forward efficiency. In other words, a broadcast-receiving node receives fewer duplicated packets. Therefore, a high forwarding-efficiency algorithm consumes fewer wireless media resources. Taking the *PER* into consideration,[5] the forward efficiency is

$$FE = \frac{\sum_i R(i)}{F(src) \times N(src) \times PER + \sum_{\forall i \neq src} [F(i) \times (N(i) - 1) \times PER]}, \qquad (1)$$

[5] Though the PER of the whole network is same in the laboratory trial, the PER of each node is different in field deployment. Therefore, when applying Eq. (1) for field trials, the mean PER of neighbors shall be used instead.

where $F(src) \times N(src) \times PER$ denotes the number of broadcasts correctly received by the source's one-hop neighbors, and $F(i) \times (N(i) - 1) \times PER$ is the number of rebroadcasts correctly received by the one-hop neighbors of node *i*.

However, forward efficiency alone cannot reflect the reliability of a broadcast algorithm, i.e., the proportion of broadcast-receiving nodes in a mesh. For example, if nodes never forward broadcast packets, the forward efficiency is 100%, but only the source's one-hop neighbors receive broadcasts. Therefore, we defined the broadcast efficiency as the result of multiplying average reliability by forward efficiency,

$$BE = FE \times AR. \tag{2}$$

Based on this metric, a broadcast algorithm is *efficient* if both average reliability and forward efficiency are high.

4.3. Experimental results

4.3.1. Longest path reliability

Fig. 6(a) depicts the longest path reliability for different broadcast data sizes on the triangular mesh. The reliability of simple flooding, domain pruning, and CDS-based algorithms decrease by about 18–20% as the packet size increases. This is because a lengthy packet is more likely to collide with others during transmission. On the other hand, the decrease in reliability is relatively small for ECS and SBA because their delayed transmissions reduce the collision probability. The reliability of the dynamic probabilistic algorithm is significantly lower than that in other algorithms. The cumulative effect of hop count on retransmission probability *P* is the reason for lower reliability. Apparently, the cumulative retransmission probability, i.e., P^n, decreases exponentially as the

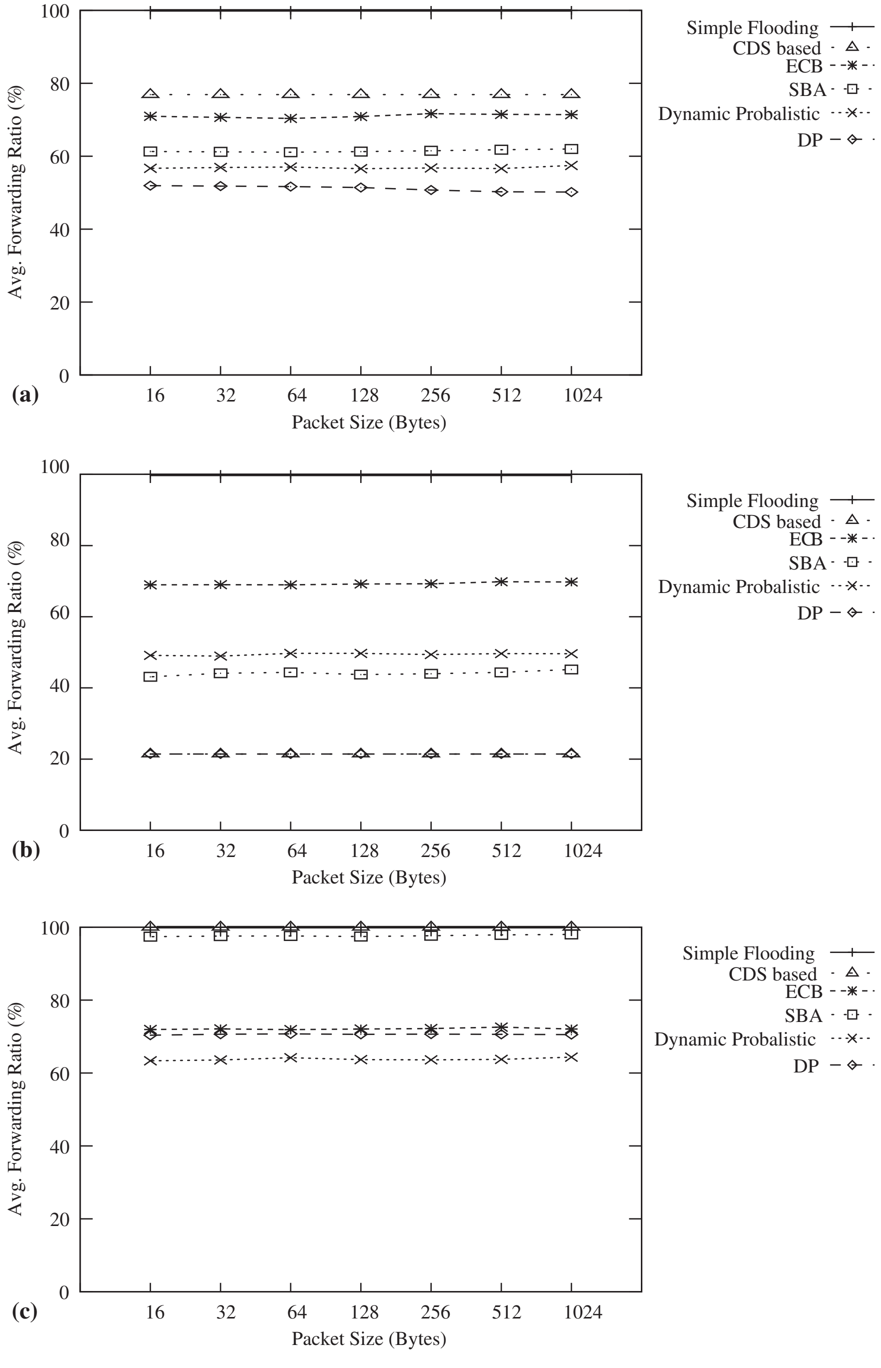


Fig. 8. Average forwarding ratio of (a) the triangular mesh; (b) the 2D mesh and (c) the 3D mesh.

hop count *n* increases. Fig. 6(b) and (c) presents the experimental results of the longest path reliability in 2D mesh and 3D mesh scenarios, respectively. These results are similar to the case of the triangular mesh. Therefore, we recommend adopting the delay-based algorithm for transmitting a service-discovery broadcast message. We also recommend avoiding the probabilistic-based algorithm which leads to poor reliability.

4.3.2. Average reliability

Fig. 7 shows the average reliability, which is the mean reliability of all mesh nodes. These results show that the simple flooding and SBA outperform the other algorithms. Due to the lower collision rate mentioned in the previous subsection, the average reliability of SBA is slightly better than that in simple flooding, and especially for larger size of data. Overall, the SBA shows the best reliability among all algorithms, which conflicts with the simulations results in [24,25]. These differences are likely because the simulated PERs were negligible, but about 7% in our real-world testbed. The SBA maintains the remaining neighbor list (RNL) to assure all the neighbors receive a broadcast packet, which also enhances reliability. Conversely, the dynamic probabilistic algorithm still has the worst performance in all three scenarios.

4.3.3. Forwarding ratio

The forwarding ratio is the retransmission ratio of the received broadcast packets. Fig. 8(a) illustrates the average forwarding ratio for different sizes of broadcast packet in a triangular mesh. Apparently, packet size is not a major determinant of the forwarding ratio. Fig. 8(b) and (c) shows that the results on a 2D and 3D mesh are also compliant with the same observation. In all three

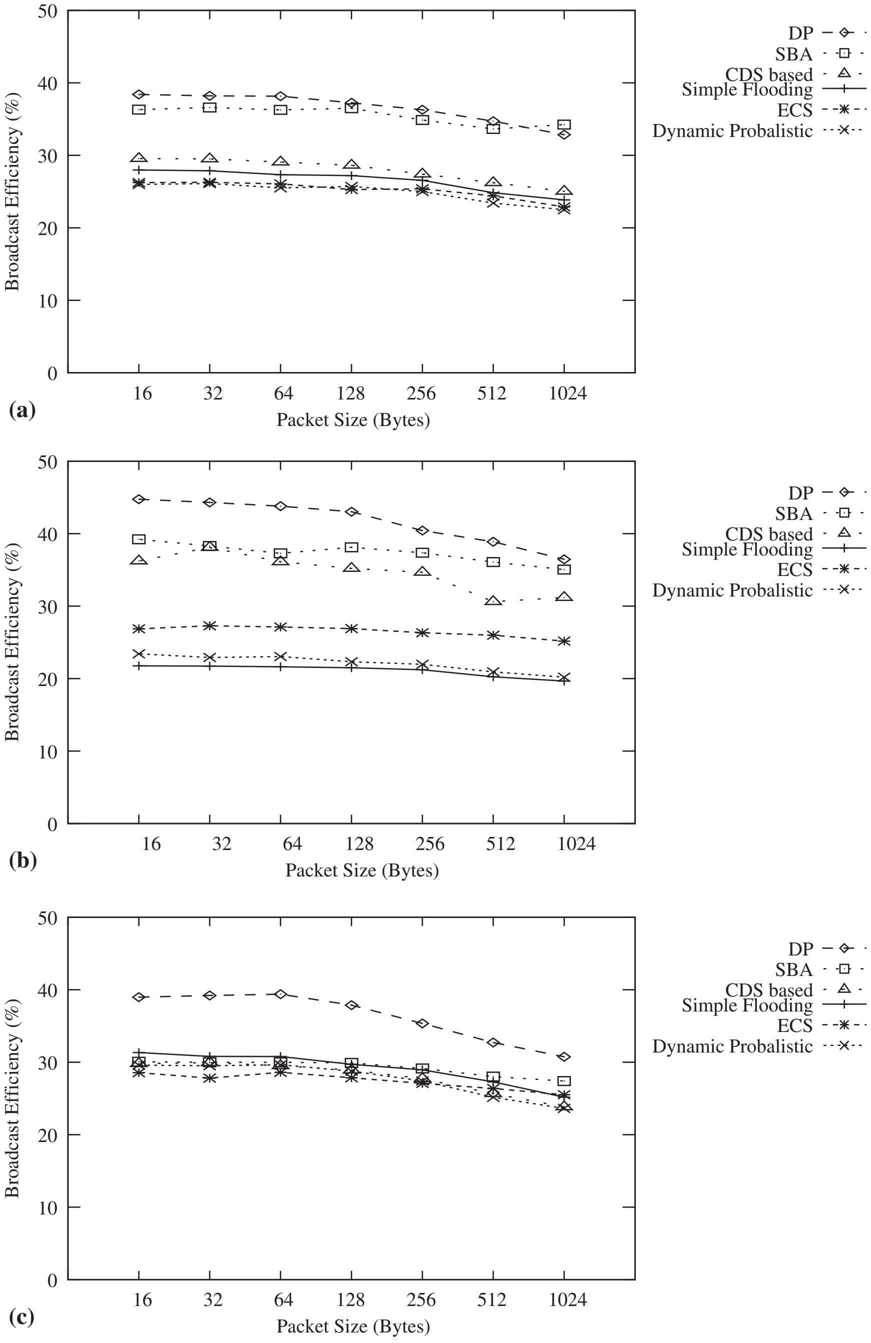


Fig. 9. Broadcast efficiency of (a) the triangular mesh; (b) the 2D mesh and (c) the 3D mesh.

scenarios, the topology does not influence the average forwarding ratios in simple flooding (100%), dynamic probabilistic algorithm (50%), and ECS (70%) are not influenced by the topology. However, topologies are critical to the forwarding ratio of algorithms using neighbor information. In particular, the CDS-based algorithm varies most (20–100%) of all algorithms. This is because a mesh node using the CDS-based algorithm can easily become a gateway node in a non-fully connected topology. In this case, a CDS-based algorithm may function as a simple flooding method.

4.3.4. Broadcast efficiency

Fig. 9 presents the results of broadcast efficiency for each algorithm. The DP algorithm outperforms all other algorithms not only for different sizes of broadcast data, but also for different topologies. A comparison of Figs. 7 and 8 suggests that this superior performance arise from the low forwarding ratio. Because the DP algorithm only allows gateway nodes to forward broadcasts, it effectively reduces the number of forwarders. On the other hand, the efficiency of another gateway-related algorithm, the CDS-based algorithm, degrades to that of the worst algorithm, simple flooding, in a 3D mesh topology. These differences may come from the gateway selection mechanism. The DP algorithm determines the gateway nodes in run-time by considering the broadcasting senders, whereas the CDS-based algorithm proactively computes the gateway nodes from the local-CDS regardless of the senders. As the previous subsection describes, all mesh nodes become forwarders in a local-CDS of a non-fully connected local topology, such as a 3D mesh. In this case, the performance of the CDS-based algorithm degrades to that of the simple flooding algorithm. Fig. 8(c) supports this conclusion by showing that the forwarding ratio of the CDS-based algorithm is 100%.

In addition, although the SBA performs best at both of the longest path reliability and average reliability, its broadcast efficiency is not the best of all algorithms. The forwarding method is a little inefficient than DP. In other words, it may generate more unnecessary broadcasts, resulting in worse broadcast efficiency. Besides, the results of the simple flooding, dynamic probabilistic algorithm, and ECS are similar and bad under the same topology.

4.3.5. Summary

Observations from the real-world testbed experiments are summarized as follows:

1. *The actual reliability of simple flooding is not as good as the results in simulation.* In the simulation results in [24,25], the reliability of simple flooding is approximately 100%. However, Figs. 7 and 8 show that the reliability of simple flooding varies from 50% to 90%, which differs from simulation. This is likely because the simulations underestimated interference and collision. These results also suggest that the design and simulation of broadcast algorithm cannot ignore the effect of PER, or that its execution in the real world may be different than the expectation.
2. *Mesh networks should use probability-based algorithms carefully.* Broadcast algorithms invoking a retransmission probability, such as dynamic probabilistic algorithm, show unacceptable reliability and efficiency in our evaluations. Though probability-based algorithms are a variant of simple flooding, our experiments show that their efficiency is equal to, or less than, the simple flooding method. This is because the decrease rate of cumulative retransmission probability is much higher than the effect of PER as the hop count increases. Therefore, we suggest that the design of probability-based algorithms must consider the cumulative effect of hop count on the retransmission probability.
3. *We recommend using the SBA algorithm in small-scaled mesh networks, and using the DP algorithm in large-scaled mesh networks.* This is because the SBA algorithm provides the best reliability despite its side-effect of longer latency due to delay. Therefore, using the SBA for a small-scaled mesh network is acceptable. For a large-scaled mesh network, we recommend the DP algorithm due to its good efficiency, lower forwarding ratio, lower collision probability, and acceptable reliability.

5. Conclusion

This study investigates the design issues of implementing broadcast algorithms in a WMN. We designed a flexible architecture capable of adopting different broadcast algorithms. To simplify the implementation of various algorithms, we developed and implemented a three-phase execution flow. This study evaluates and discusses six representative broadcast algorithms using a real-world testbed. Experiments on the real-world testbed show that simulation results differ from actual testbed results. These differences may arise from the simulation underestimating the effect of dynamic channel condition, contention and collision on the wireless environment. Therefore, our testbed results reveal new properties of broadcast algorithms that may affect real-world deployment.

Experiment results show that delay-based algorithms are reliable due to their reduced collision probability. Our results also show that probabilistic-based algorithms are not reliable because the cumulative retransmission probability decreases significantly as the hop count increases. The domain-pruning algorithm is the most efficient algorithm because of its run-time gateway selection. Different topologies favor different algorithms. Therefore, the results presented in this paper can serve as a basis for designing an adaptive broadcast algorithm that selects the most suitable algorithm for the observed topology in runtime.

References

[1] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, J.-P. Sheu, The broadcast storm problem in a mobile ad hoc network, in: Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), ACM, 1999, pp. 151–162.

[2] W. Lou, J. Wu, Toward broadcast reliability in mobile ad hoc networks with double coverage, IEEE Transactions on Mobile Computing 6 (2007) 148–163.

[3] I.F. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Computer Networks 47 (2005) 445–487.

[4] S.A. Mahmud, S. Khan, S. Khan, H. Al-Raweshidy, A comparison of MANETs and WMNs: commercial feasibility of community wireless networks and MANETs, in: Proceedings of the 1st International Conference on Access Networks (AccessNet), ACM, 2006, pp. 99–105.

[5] R. Bruno, M. Conti, E. Gregori, Mesh networks: commodity multihop ad hoc networks, IEEE Communications Magazine 43 (2005) 123–131.

[6] D.C. Plummer, An ethernet address resolution protocol, RFC 826, 1982.

[7] IEEE Computer Society, IEEE standard for local and metropolitan area networks media access control (MAC) bridges, IEEE Std 802.1D-2004, 2004.

[8] 802.11 Working Group, Draft Amendment to Standard IEEE 802.11: Mesh Networking, IEEE P802.11s/D4.0, 2009.

[9] C. Ho, K. Obraczka, G. Tsudik, K. Viswanath, Flooding for reliable multicast in multi-hop ad hoc networks, in: Proceedings of the International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM), ACM, 1999, pp. 64–71.

[10] Q. Zhang, D.P. Agrawal, Dynamic probabilistic broadcasting in MANETs, Journal of Parallel and Distributed Computing 65 (2005) 220–233.

[11] A. Mohammed, M. Ould-Khaoua, L. Mackenzie, An efficient counter-based broadcast scheme for mobile ad hoc networks, Lecture Notes in Computer Science 4748 (2007) 275–283.

[12] S. Qiang, L. Layuan, C. Niansheng, S. Aziz, A cross-layer design for broadcast algorithm in MANETs, in: Proceedings of the IEEE Wireless Communications, Networking and Mobile Computing (WiCOM), IEEE Press, 2006, pp. 1–4.

[13] L. Layuan, Z. Feng, C. Li, Q. Sun, A distributed broadcast algorithm for wireless mobile ad hoc networks, Lecture Notes in Computer Science 4352 (2007) 494–501.

[14] W. Peng, X.-C. Lu, On the reduction of broadcast redundancy in mobile ad hoc networks, in: Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), IEEE Press, 2000, pp. 129–130.

[15] H. Lim, C. Kim, Flooding in wireless ad hoc networks, Computer Communications 24 (2001) 353–363.

[16] W. Lou, J. Wu, On reducing broadcast redundancy in ad hoc wireless networks IEEE Transactions on Mobile Computing 1 (2002) 111–123.

[17] B. Shen, B. Li, B.-X. Shi, Improved approach to enhanced internet connectivity for mobile ad hoc networks, Journal of Southeast University (English Edition) 23 (2007) 16–21.

[18] J. Wu, H. Li, On calculating connected dominating set for efficient routing in ad hoc wireless networks, in: Proceedings of the International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM), ACM, 1999, pp. 7–14.

[19] A. Qayyum, L. Viennot, A. Laouiti, Multipoint relaying: an efficient technique for flooding in mobile wireless networks, Technical Report 3898, INRIA, 2000.

[20] F. Ingelrest, D. Simplot-Ryl, Maximizing the probability of delivery of multipoint relay broadcast protocol in wireless ad hoc networks with a realistic physical layer, Lecture Notes in Computer Science 4325 (2006) 143–154.

[21] A. Keshavarz-Haddad, V. Ribeiro, R. Riedi, DRB and DCCB: efficient and robust dynamic broadcast for ad hoc and sensor networks, in: Proceedings of the IEEE International Conference on Sensors, Mesh and Ad Hoc Communications and Networks (SECON), IEEE Press, 2007, pp. 253–262.

[22] K. Hasegawa, T. Fujii, K. Umebayashi, Y. Kamiya, Y. Suzuki, A dialogue-based mac protocol for efficient broadcasting in ad hoc networks, in: Proceedings of the Mobile and Wireless Communications Summit, 2007, pp. 1–5.

[23] H. Lim, C. Kim, Multicast tree construction and flooding in wireless ad hoc networks, in: Proceedings of the ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM), ACM, 2000, pp. 61–68.

[24] B. Williams, T. Camp, Comparison of broadcasting techniques for mobile ad hoc networks, in: Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc), ACM, 2002, pp. 194–205.

[25] T.B. ao de Oliveira, V.F. Costa, F. Greve, On the behavior of broadcasting protocols for MANETs under omission faults scenarios, Lecture Notes in Computer Science 4746 (2007) 142–159.

[26] J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K. chan Lan, Y. Xu, W. Ye, D. Estrin, R. Govindan, Effects of detail in wireless network simulation, in: Proceedings of the SCS Multiconference on Distributed Simulation, 2001, pp. 3–11.

[27] K.-W. Chin, J. Judge, A. Williams, R. Kermode, Implementation experience with MANET routing protocols, SIGCOMM Computer Communication Review 32 (2002) 49–59.

[28] F. Haq, T. Kunz, Simulation vs. emulation: evaluating mobile ad hoc network routing protocols, in: Proceedings of the International Workshop on Wireless Ad-hoc Networks (IWWAN '05), 2005.

[29] W.T.H. Woon, T.-C. Wan, Performance evaluation of IEEE 802.15.4 wireless multihop networks simulation and testbed approach, International Journal of Ad Hoc Ubiquitous Computing 3 (2008) 57–66.

[30] T.H. Clausen, P. Jacquet, Optimized Link State Routing Protocol (OLSR), RFC 3626, 2003.

[31] A. Neumann, C. Aichele, M. Lindner, B.A.T.M.A.N Status Report, 2007. Available from: <http://downloads.open-mesh.net/batman/misc/batman-status.pdf>.

[32] The B.A.T.M.A.N. team, B.A.T.M.A.N. (Better Approach To Mobile Ad-hoc Networking), 2010. Available from: <http://www.open-mesh.net/>.

[33] C.T. Chou, A. Misra, J. Qadir, Low-latency broadcast in multirate wireless mesh networks, IEEE Journal on Selected Areas in Communications 24 (2006) 2081–2091.

[34] T. Wang, X. Du, W. Cheng, Z. Yang, W. Liu, A fast broadcast tree construction in multi-rate wireless mesh networks, in: Proceedings of the IEEE International Conference on Communications (ICC '07), 2007, pp. 1722–1727.

[35] Q. Xin, Y. Zhang, Optimal fault-tolerant broadcasting in wireless mesh networks, in: Proceedings of the International Conference on High Performance Switching and Routing (HSPR), IEEE Press, 2008, pp. 151–157.

[36] H.L. Nguyen, U.T. Nguyen, Minimum interference channel assignment for multicast in multi-radio wireless mesh networks, in: Proceedings of International Wireless Communications and Mobile Computing Conference (IWCMC '08), 2008, pp. 626–631.

[37] H.S. Chiu, B. Wu, K. Yeung, K.-S. Lui, Widest spanning tree for multi-channel multi-interface wireless mesh networks, in: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '08), 2008, pp. 2194–2199.

[38] M. Song, J. Wang, Q. Hao, Broadcasting protocols for multi-radio multi-channel and multi-rate mesh networks, in: Proceedings of the IEEE International Conference on Communications (ICC '07), 2007, pp. 3604–3609.

[39] D. Koutsonikolas, S. Das, Y.C. Hu, An interference-aware fair scheduling for multicast in wireless mesh networks, Journal of Parallel and Distributed Computing 68 (2008) 372–386.

[40] J. Robinson, E. Knightly, A performance study of deployment factors in wireless mesh networks, in: Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '07), 2007, pp. 2054–2062.

[41] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, M. Singh, Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols, in: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '05), 2005, pp. 1664–1669.