Active Prefixes for Mobile Ad-Hoc Networks

Renato C. Dutra, Héberte F. Moraes, Claudio L. Amorim
Laboratório de Computação Paralela - COPPE/UFRJ, Rio de Janeiro, Brasil
{rcdutra, heberte, amorim}@lcp.coppe.ufrj.br

ABSTRACT

We introduce and evaluate the active prefix (AP), a novel communication element consisting of node characteristics and application interests used to build efficient AP - based mobile ad-hoc networks (APNs) with four distinguishing properties. First, APs define node addressing and application interests in a distributed way. Second, nodes communicate by sending messages using their APs as message headers to collaboratively support probabilistic message forwarding and addressing. Third, nodes use APs as application matching filters at the network layer to implement cross-layer communication. Fourth, APN can be adapted to run IP - based applications. We used the NS - 3.8 to compare the performance of APN protocols against those of representative IP - based protocols for MANETs (AODV and AODV + Gossip3) and the results showed that APNs achieved 16 percent higher message delivery rates and one order of magnitude lower latency and message overhead in scenarios with 150 mobile nodes. To evaluate the performance of APN protocols in practice, we ran experiments in the laboratory, including a chat application on a 20 - node APN where each node consisted of a host coupled with a Tmote. Our preliminary experiments validated the simulation results and indicate the feasibility of using APs as effective building blocks for ad-hoc networks, in particular for mobile ad-hoc networks.

Categories and Subject Descriptors

C.2.1 [**Computer Communication Networks**]: Network Architecture and Design Network topology

General Terms

Interest-based, MANET, Wireless network, Active Prefix, AODV, Gossip, Content-network, Naming data

1. INTRODUCTION

The widespread use of mobile devices with wireless communication interfaces has made applications for mobile networks, in particular mobile ad-hoc networks (MANETs), increasingly attractive for physical environments with defective or infrastructure-less communication. However, despite many years of research efforts, routing protocols for MANETs [1, 3] still present limited message delivery, high latency, and large message overhead, which strongly inhibit the applications for MANETs in practice [5].

Recently, it has been argued that the main reason for the limited performance of routing protocols for MANETs is the fact that they assumed node addressing and protocol stack models that were originally designed for IP-centric wired networks, whereas MANETs have characteristics and dynamics radically distinct from their wired network counterparts [13]. In wired networks, the topological locations of nodes and their interconnections are practically stable with few disruptions, thus allowing routing protocols to efficiently use hierarchical addressing under centralized control, namely, IP-based communication, routing tables, and multilayer protocols. However, these assumptions fit MANETs poorly, as dynamic changes of node connectivity and network topology, along with transmission interference, neutralize most adaptations made to routing IP-based protocols for these networks.

Next, we propose and evaluate an approach that contrasts to the current one of adapting IP-based protocols to work in MANETs; specifically, we introduce a novel mechanism that can build efficient routing protocols for ad-hoc networking in such a way that they can also be adapted to run IP-based applications; moreover, our mechanism does not rely on IP identifiers, routing tables, or IP-centric communication.

Concretely, we introduce and evaluate the active prefix (AP), a novel communication element consisting of node characteristics and application interests, which allows more effective ad-hoc networks to be built through AP-based ad-hoc networks (APNs). In APNs, every node sets one AP for each local application and nodes work collaboratively to accomplish four main functions. First, each node defines its own characteristics independently of the other nodes, whereas node applications configure their own interest attributes; thus, APs allow nodes and applications to be identified in distributed way. Second, nodes communicate in AP networks by using their APs as message headers. APN protocols use these headers to implement both message forwarding and two addressing modes, namely, source - to - destination (S2D) and a new addressing mode based on interest matching that we call AP addressing. Third, nodes implement cross - layer communication to further reduce the end - to - end latency of applications [2]. Fourth, APNs can be adapted to run IP - based applications.

We used the NS-3.8 Network Simulator to compare the effectiveness of message delivery in APNs using node characteristics for probabilistic message forwarding with that of the Gossip3 algorithm [9]. In addition, we evaluated protocol overheads for the AP addressing mode in APNs and compared the performance of APN protocols against those of representative IP-based protocols (AODV [14] and the AODV+Gossip3 (G3AODV) [8]) for MANETs in a scenario with 150 mobile nodes uniformly distributed in a rectangular area of 750 m x 300 m. The simulation results showed that APN protocols achieved on average 16 percent higher delivery rates as well as latency and message overhead one order of magnitude lower than either AODV or G3AODV. In addition, to evaluate the performance of APN protocols in practice and validate our simulation results, we ran experiments and a chat application on a 20-node APN in which each node consisted of a host coupled with a Tmote. Our preliminary experiments validated the simulation results and indicate the feasibility of using APs as effective building blocks for ad-hoc networks, in particular for mobile ad-hoc networks.

In summary, the main contributions of this paper are the following:

- Introducing the active prefix, a novel mechanism for building effective active-prefix ad-hoc networks, in particular active-prefix mobile ad-hoc networks.
- Using active prefixes to build efficient cross-layer ad-hoc protocols with lower latency between the application and network layers.
- Using active prefixes for AP and S2D addressing in mobile ad-hoc networks, and showing that active prefixes can support IP-based applications.
- Introducing a new probabilistic message forwarding algorithm based on active prefixes, which offers greater flexibility to the communication subsystem in mobile ad-hoc networks.
- Presenting results from a large number of simulations and practical experiments including of a chat application to demonstrate the feasibility of APNs and the higher performance of APNs in comparison with representative IP - based protocols for MANETs.

This paper is organized as follows. In Section 2, we describe the design of AP networks in more detail. In Section 3, we assess the bimodal behavior of message delivery of AP networks and present simulation results of a comparative evaluation of the performance of APN protocols against those of AODV and G3AODV, along with a sensitivity analysis and preliminary experimental evaluation of APNs. Section 4 discusses related work and Section 5 concludes the paper.

2. DESIGN OF AP NETWORKS

In designing the Active Prefix we assume that the node characteristics would support probabilistic message forwarding and node addressing in a distributed way. To accomplish the former we built on previous work [7, 9] that showed that gossiping exhibits bimodal behavior and proposed gossip-based routing protocols for ad-hoc networking, in which a suitable gossiping probability is used for message forwarding to ensure message delivery to almost every node in almost every execution. To achieve both requirements, however, we extended their results in significant ways. Specifically, instead of using a single gossiping probability for which one random variable suffices, we resorted to multiple discrete random variables with probabilistic distributions. Moreover, we observed that a set of discrete random variables with few bits each can jointly establish both node addressing and matching filters for message forwarding in a probabilistic way, as we describe in detail next.

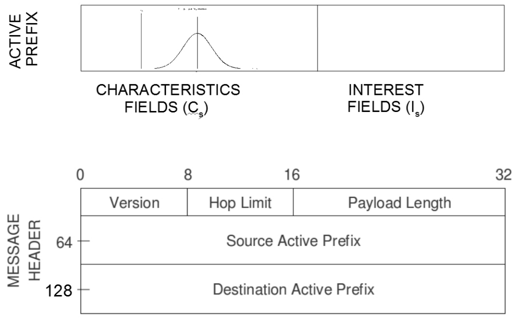

Figure 1: Active Prefix (AP) and AP message header (Components)

2.1 Message Forwarding

Figure 1 presents the basic structure of APs and AP message header in a typical APN. As can be seen in the figure, the C and I fields of an AP reserve up to 64 bits. The message header consists of protocol version, hop limit, payload length that depends on the communication technology being used, and two AP descriptors that identify the origin and destination nodes.

Node characteristics (Cs) are central to building the address space of APNs and to supporting message forwarding. The C component of an AP is structured for node addressing by using an appropriate number of fields, with each field value being set independently by the node using discrete random variables with a selected probability distribution. The node address is then constituted by the sequence of bits of the C fields. In particular, we chose discrete random variables that are normally distributed as many phenomena in nature follow this distribution, and observing the central limit theorem, leads C to follow a multivariate normal distribution. For example, in our simulations we defined C with eight fields formed with eight 3-bit random variables with normal probability distributions to build APNs with 24-bit addressing space.

In addition to node addressing, node characteristics allow mobile nodes to cooperatively forwarding messages in an efficient probabilistic way and independently of the application/user interests of individual nodes. Specifically, we use the C fields as matching filters for message forwarding in such a way that APNs achieve message delivery that exhibits bimodal behavior, as will be shown in Section 3.1. In general, there are configurations of C fields with different lengths that provide efficient message delivery, and thus one can choose a configuration that has the appropriate number of bits for a particular addressing space size. For instance, the above C configuration achieves message delivery rates of nearly 100 percent.

Nodes implement message forwarding by using their C fields as a matching filter with the corresponding C fields of received AP messages, wherein a particular node compares every pair of C fields and forwards the received message if any pair matches; otherwise, the node discards the message. Moreover, once a node configures its C fields, it can automatically start message forwarding independently of whether its I fields have been set.

A node may have no common characteristics with any of its neighbors and thus discard all of the messages it sends and vice-versa. Upon detecting this shortcoming, the node either regenerates or adjust its C fields, for example, by coping a C field that it received from one of its neighbors.

Note that there are two main reasons for using characteristics instead of application interests for message forwarding. First, popular interests can cause flooding in the network whereas unpopular ones can prevent messages from reaching their destinations. Second, the use of characteristics can reduce the node isolation caused by unmatched interests, as the C fields enable neighbor nodes to forward messages collaboratively even if they do not have any common interests.

2.2 Message addressing

In APNs, each running application on a particular node sets its I fields with the appropriate keywords and terms to identify its main interests. The I fields together with the node characteristics define the node's AP, which is inserted in the header of messages that the application sends to identify the origin node of the messages. In addition, APNs use APs to implement two different addressing modes, conventional S2D and the AP addressing as follows.

In S2D mode the source node requires a full AP descriptor of the destination node to send messages. In contrast, in AP addressing mode, the source node only indicates the interest(s) that the I fields of the destination node should match. Thus, AP addressing allows a given node to start communicating by sending an AP message either to search for other nodes with a common interest (to establish an interest-based group communication) or to identify interest-matching APs (to further communicate in S2D mode selectively).

To find other node(s) with common interest(s), the AP message header is used as a matching filter with the APs of other intermediate nodes, as the message is forwarded from node to node over the network until it finds its destination or is discarded. We assume that an interest matching occurs if there is at least one common interest. In this case, the message is copied and delivered to the associate AP application locally. Whenever an interest match occurs, the corresponding source and intermediate nodes can continue to communicate in AP mode or can change to S2D mode. Note that the intermediate nodes can store the APs of the messages that they receive to allow AP protocols to use message footprints for the optimization of APN addressing; however, this issue is outside the scope of this work.

To support IP - based applications, APN protocols need to access the source and destination IPs of the messages sent by those applications. An APN protocol will insert the source IP and the destination IP of each message in the corresponding I field of the source and destination APs, respectively, and will use the AP message header to envelop the IP packets. Therefore, it is necessary to integrate the APN protocol with the IP protocol to allow the messages sent by an IP-based application to be transparently transferred to the APN protocol, which will properly forward them over the network.

An AP descriptor can be formally expressed according to equation 1, where $AP(C(b_C); I(b_I))$ is the active prefix, C is a characteristic field, and b_C is the number of bits for each characteristic field, I is an application interest field. b_I is the number of bits for each interest field that is typically used as an index to a dictionary of terms of an application, except from IP-based application in which the IP is itself the interest. For instance, consider that in equation 1 the C fields have the same

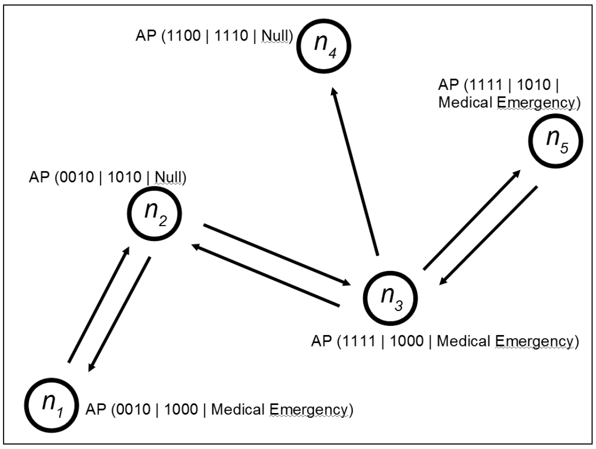
AP (1100 | 1110 | Null)
n_4
AP (1111 | 1010 | Medical Emergency)
n_5
AP (0010 | 1010 | Null)
n_2
n_3
AP (1111 | 1000 | Medical Emergency)
n_1
AP (0010 | 1000 | Medical Emergency)

Figure 2: Service-discovery in APNs

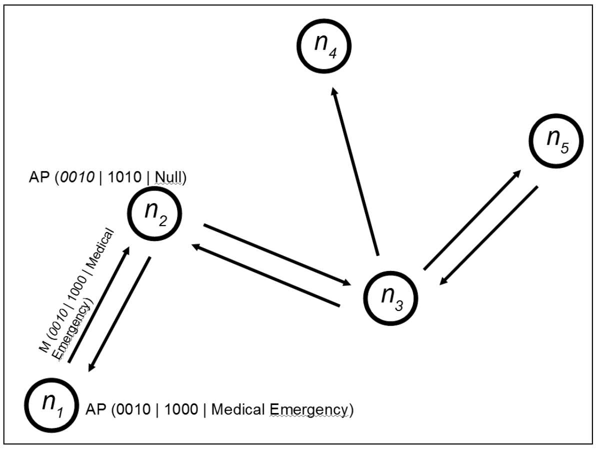
n_4
n_5
AP (0010 | 1010 | Null)
n_2
M (0010 | 1000 | Medical Emergency)
n_3
n_1
AP (0010 | 1000 | Medical Emergency)

Figure 3: n_1 sends an AP message

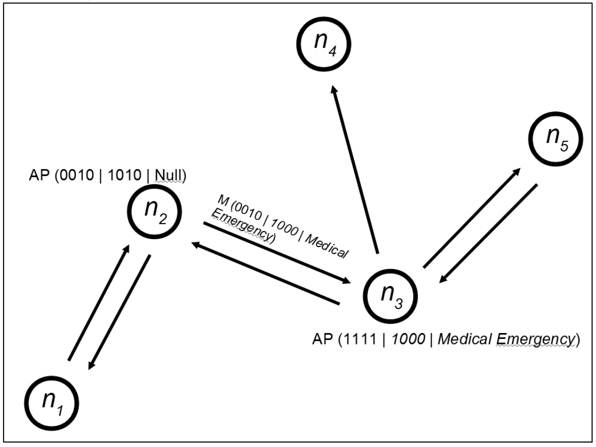
n_4
n_5
AP (0010 | 1010 | Null)
n_2
M (0010 | 1000 | Medical Emergency)
n_3
AP (1111 | 1000 | Medical Emergency)
n_1

Figure 4: $n_2(C_a) = n_1(C_a)$

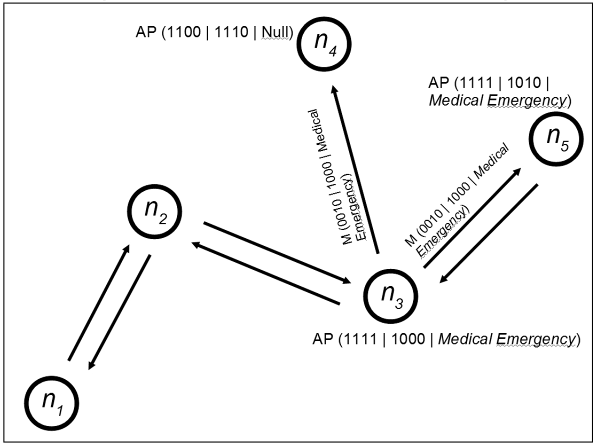
AP (1100 | 1110 | Null)
n_4
AP (1111 | 1010 | Medical Emergency)
n_5
M (0010 | 1000 | Medical Emergency)
M (0010 | 1000 | Medical Emergency)
n_2
n_3
AP (1111 | 1000 | Medical Emergency)
n_1

Figure 5: $n_3(C_b) = n_1(C_b)$

b_C length. Assume that C has five fields, each one with 3-bit $b_C(a,b,c,d,e)$, and I has one field with 32-bits $b_I(y)$. Equation 2, shows the resulting AP descriptor.

$$AP(C(b_C);I(b_I)) = P(1(b_1),2(b_2)...C(b_C); \\ 1(b_1),2(b_2)...I(b_I)) \ \forall\, C,I \ \in \ \mathbb{N}^* \quad (1)$$

$$AP(5(3);1(32)) = P(a(3),b(3),c(3),d(3),e(3);y(32)) \quad (2)$$

As an example of using Cs as node addresses, consider an AP descriptor $(8(3);1(32))$ with an I field having 10 options for application interests chosen from terms of a given dictionary. The probability of having two APs with the same bit patterns is as follows:

$$Pr_{C=8} = \prod_{k=8} 1/8^k \cong 10^{-8} \quad (3)$$

$$Pr_{P(8,3;1)} = Pr_{C=8} * Pr_{I=1} \cong 10^{-9} \quad (4)$$

Therefore, the probability is lower than 10^{-9}. Moreover, the use of multiple application interests in AP messages can further reduce the probability of two nodes having the same address, that is, the same C bit pattern and interests. An option of a particular APN implementation is to associate multiple interests to a given application to further avoid address conflicts.

The security of data in AP messages can be attained by either cryptographic signatures or passwords according to the level of security that is required by the application. For example, a certain APN application can use an asymmetric key system to encrypt its message payloads and includes the destination's public key as one of the I fields, and a destination node can decrypt the message payloads with its private key. Usually, public keys are large bit vectors, thus APNs should support I fields with greater lengths for this kind of applications.

To illustrate the use of AP networks, consider a simple example of a service-discovery application [16] running on a 5-node APN whose active prefixes are presented on Table 1 and the associate graph is shown in Figure 2.

Assume that an origin node n_1 needs assistance from a medical emergency team in a given disaster-relief area

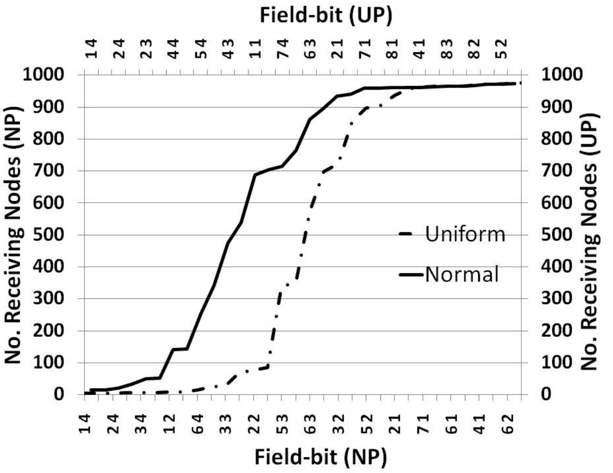

Figure 6: Bimodal behavior (APN 20x50 grid)

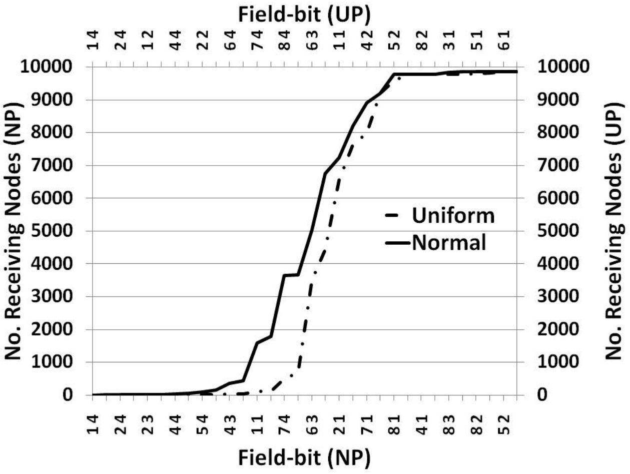

Figure 7: Bimodal behavior (APN 20x500 grid)

Table 1: Active Prefix configuration for service-discovery (refer to Fig. 2)

node	neighbor	C		I
		a	b	y
n_1	n_2	*0010*	*1000*	Medical Emergency
n_2	n_1, n_3	*0010*	**1010**	
n_3	n_2, n_4, n_5	**1111**	*1000*	Medical Emergency
n_4	n_3	1100	1110	
n_5	n_3	**1111**	**1010**	Medical Emergency

where the existing communication infrastructure is not working. Node n_1 sends a message with the interest application I = "medical emergency" (Fig. 3) chosen from a list of interests that the emergency team broadcasts periodically using the AP addressing mode. Node n_2 receives the message and forwards it as their APs match on $C_a = 0010$ (Fig. 4). Node n_3 receives the message from n_2 and forwards it because $C_b = 1000$ (Fig. 5), and also n_3 is a destination because its filter (I = "medical emergency") matches with the message's I field. Nodes n_4 and n_5 receive the message, then n_4 drops the message ($C(n_4) \neq C(n_1)$), whereas also n_5 is a destination and transfers the message to the local application. In this way, the proper destination nodes, n_3 and n_5, are reached, and they store the AP of n_1 as the source node address for further interactions with n_1 using the S2D addressing mode. Note that C fields of n_3 does not match with C fields of n_2; thus, n_1 will not receive messages from n_3.

3. EVALUATION AND PERFORMANCE OF APNS

The use of APs in routing protocols for ad-hoc networks raises some important questions, which we address in this section. First, we assess whether the use of APs for message delivery presents bimodal behavior similar to gossiping, and whether different probabilistic distributions can influence the performance of AP protocols, all while verifying the necessary C-field length to ensure node addressing. Next, we compare the performance of APN protocols with those of two representative IP-based protocols for MANETs, namely, AODV and G3AODV, after which we assess the overheads of AP addressing for group communication in APNs.

3.1 Bimodal behavior

To verify bimodal behavior in AP networks' message delivery, we simulated two grids of 20x50 and 20x500 with $1,000$ and $10,000$ static nodes, respectively, uniformly distributed over the grids in the NS-3.8 network simulator. We configured the simulation parameters with 4-neighbor degree, −30 dBm of transmitter and receiver gains, *AdhocWifiMac*, *Constant Rate WiFi manager*, *Wifib-2mbs*, 512-byte packet size, and a simulation time of 120 s. We also simulated Gossip3 with the same parameters for 20x50 grid with $1,000$ nodes. In each simulation, a source node sent one message and the number of receiving nodes (RNs) was measured. For comparative evaluation, we used two APN implementations based on the Normal probability (NP) and Uniform probability (UP) distributions to generate APN node characteristics. Figures 6 e 7 present the RN results using diverse combinations of C fields(bits); for example, 1(4) (or 14 in Figures) means one C field with 4 bits.

Figure 6 shows that APNs with $1,000$ nodes exhibited bimodal behavior with C fields generated by either NP or UP distributions. For the UP distribution, with the C configuration either equal to 6(1) or any other configuration on the right, more than 97 percent of the nodes received all of the messages; with the C configuration equal to either 5(4) or any other configuration on

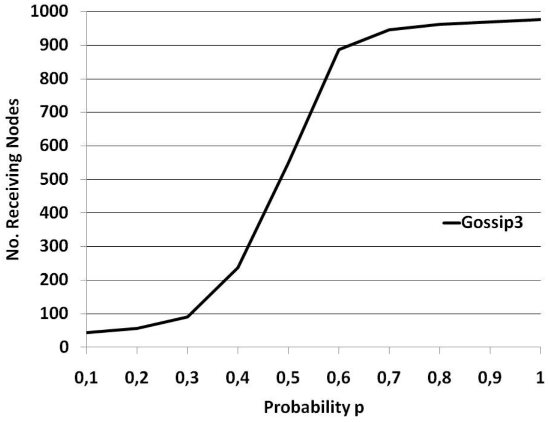

Figure 8: Bimodal behavior Gossip3 20x50 **grid**

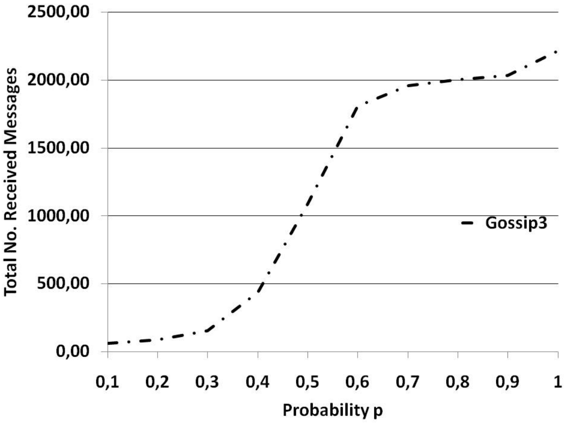

Figure 9: Received message Gossip3 20x50 **grid**

the left, most nodes received no message, whereas the equivalent configurations are $C = 7(3)$ and $C = 2(4)$, respectively, for the NP distribution. Figure 6 shows that APN (NP) and APN (UP) achieved equivalent message delivery rates for C configuration 8(3) and for the other ones shown in the figure.

Figure 7 presents simulations with 10,000 nodes and again a tendency towards bimodal behavior can be observed for APN (UP) and APN (NP). It is noteworthy that using AP(8(3); 1(32)), that is, a single 32-bit interest, we verified (not shown) that no duplicity of AP addresses occurred between any pair of $500,000$ that were generated for $10,000$ nodes in 50 simulation runs.

Figures 8 and 9 show the number of receiving nodes (RN) and message overhead (MO) for Gossip3. The bimodal behavior of Gossip3 is shown in Figure 8 for a $1,000$ - node grid, wherein the number of receiving nodes (RNs) was greater than 900 when using a probability $p > 0.6$.

3.2 Simulation Results

We evaluated the performance of APN protocols against those of G3AODV and AODV. To this end, we implemented Gossip3 (with p=0.65, m=1, and k=4), the G3AODV protocol, and the AP network protocol in the NS-3.8 network simulator, and we used the native version of the AODV protocol. We configured the simulator with 150 nodes uniformly distributed in a rectangular area of 750 m x 300 m; node mobility using *Randomwaypoint* standard with [2,8] m/s; *AdhocWifiMac*, *Constant Rate WiFi manager*, *Wifib-2mbs*, 512-byte packet size; transmission range of 100 m; with transmitter & receiver gains of -4 dBm; and 1 s, 100 s, and 200 s for pause times. We simulated an ad-hoc network (IEEE Standard 802.11b) with a real link layer for protocol evaluations. We present the average values for 50 simulation runs with a 95 percent confidence interval.

The APN simulations used $AP(8(3); 1(32))$ and nodes that stored the APs of received messages, which were forwarded just once, although destinations could receive duplicate messages because of multiple path effects. We simulated 5 and 30 pairs of source-to-destination nodes that were randomly selected, and each selected source node sent one AP message to its destination node. We measured, message delivery rate, message overhead, latency, hop count, and failure message overhead defined as follows:

- Message delivery rate - the average of unique messages received by the destination nodes divided by the number of messages sent by source nodes. The delivery rate indicates the degree of resilience of a given network;
- Message overhead (MO) - the total number of messages received by all 150 nodes for a given number of messages sent by source nodes. This measure indicates how efficient a protocol is at using the network's capacity to maximize message delivery rates while minimizing the amount of messages broadcast in the network;
- Latency - the average elapsed time between an application sending a message and a remote application receiving the message in the destination node;
- Hop count - the average number of hops of a message takes from sender to receiver;
- Failure message overhead - the total number of received message with error by all 150 nodes. It indicates the message delivery failures using the shared channel (failure messages caused by the MAC layer were nearly zero).

3.2.1 Performance Comparison

As can be observed in Figure 10, message delivery rates achieved nearly 100 percent in the APN protocol

(APN-a) and at most 80 percent in the G3AODV and AODV protocols for 5 pairs. Figure 11 shows that as the number of communication node pairs increased to 30, the delivery rate of APN-a decreased to 68 percent, whereas those of G3AODV and AODV decreased to 55 percent and 50 percent, respectively.

Figures 12 and 13 present the message overhead for each protocol according to the number of pairs and the pause time. As can be seen in Figure 12, MO achieved $9,000$ in APN-a regardless of the pause time, whereas in G3AODV and AODV, MO varied from $8,000$ to $80,000$ for 5 pairs depending on the pause time.

Figure 13 shows that MO was equal to $12,000$ in APN-a, whereas MO varied from $80,000$ to $140,000$ in AODV and was $120,000$ in G3AODV for 30 node pairs as the pause time varied from 0s to 200s. These results reveal that G3AODV and AODV generate a great lot of maintenance messages that increase transmission interference and explain their one order of magnitude higher message overhead.

In Figure 14, APN-a latency achieved 90ms, whereas in both G3AODV and AODV it was much higher at $1,200$ ms for 5 pairs. In Figure 15, as the number of node pairs increased to 30, latency in APN-a increased to 120 ms, whereas latency in G3AODV and AODV reached $1,500$ ms. Therefore the latency of APN-a was between 12 and 13 times lower than that of either G3AODV or AODV. The reason for the superior latency is the fact that APN-a does not attempt to find routes as AODV and G3AODV do, instead APN-a uses several alternative paths for message forwarding, thus making message delivery faster.

Figures 16 and 17 show that the hop count achieved 8 and 5.5 in APN-a whereas it achieved 6 and 4 in G3AODV and AODV for 5 and 30 pairs, respectively; Note that G3AODV and AODV achieved smaller number of hops as they use a short - path algorithm for route discovery.

The failure message overhead as shown in Figures 18, and 19, was one order of magnitude lower in APN-a than either G3AODV or AODV, for 5 pairs; and two orders of magnitude lower in APN-a for 30 pairs. These largely favorable performance results for APN-a can be explained again by the transmission interference caused by the maintenance messages required by either G3AODV or AODV.

3.2.2 *Performance of AP addressing*

To evaluate overheads using AP addressing for group communication (APN-g), we used two groups of 5 and 30 nodes, in which each group had a common interest, the AP destination was set with $C = null$, and I with the same interest attribute; thus, the APN-g protocol had to find the other group nodes with the same interest to delivery the message.

As can be observed in Figure 20, the message delivery rate in APN-g was practically 100 percent for the group size of 5 nodes, and it was reduced to 68 percent as the group size increased to 30.

Figure 21 shows that in APN-g the message overhead achieved 700 and $3,200$ messages for group sizes of 5 and 30 nodes, respectively, regardless of the pause time.

Figure 22 presents the latency results for APN-g, and shows that APN-g achieved latencies of 120 ms and 240 ms for communication group sizes of 5 and 30 nodes, respectively.

In Figure 23, The hop count in APN-g achieved low values at 4 and 2 for 5 and 30 nodes, respectively. These values are lower than those of corresponding APN-a, and suggest that as the number of nodes increases, the average number of hops decreases because of alternative shorter paths, which results in faster group communication.

Figure 24 presents the failure message overhead, and shows that it increased from $3,700$ to $14,000$ messages, for 5 and 30 nodes, respectively. This result reveals that failure message overhead in APN-g grows less than linearly with the group size.

3.2.3 *Sensitivity analysis and experimental evaluation*

We performed a sensitivity analysis to evaluate the impact of the transmission medium interference on the performance of APN protocols by varying the time interval in which the nodes transmitted the messages they received. We evaluated the transmission interference only for 30 pairs of nodes as the message delivery rate was nearly 100 percent for 5 pairs. Figures 25, 26, 27, and 28 present message delivery rate, latency, message overhead, and failure message overhead. We configured the simulator with the same parameter values defined in Section 3.2.2 to run APN-g for 30 pairs.

In Figure 25, the message delivery rate in APNs is practically constant at 45 percent for a transmission delay below 10 ms, and increases to 64 percent for a transmission delay of 200 ms, which demonstrate a direct relationship between transmission delay and message delivery. Similarly, it can be observed in Figure 26 that the latency in APNs increased from 45 ms to 150 ms for the same transmission delay interval. Again, both message overhead and failure message overhead presented the same behavior, specifically MO increased from $14,000$ to $20,000$ and failure message from $15,000$ to $22,000$. Thus, higher delivery rates imply on greater latencies and larger message overheads in APNs.

We also built a basic 20 - node APN to validate the simulation results, where each node consisted of a host coupled with a Tmote (Zigbee) to allow hosts to communicate either directly or by multihop. Recall that the node mobility had little impact on the APN per-

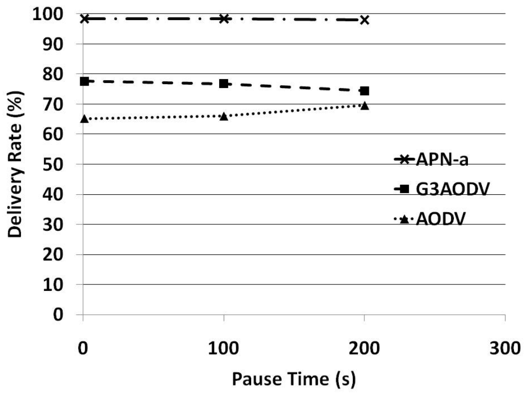

Figure 10: Delivery rate (S2D, 5 pairs)

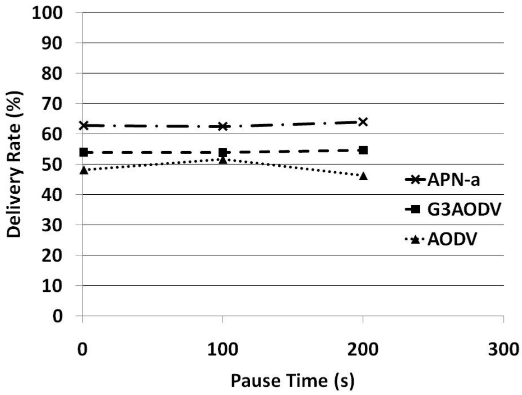

Figure 11: Delivery rate (S2D, 30 pairs)

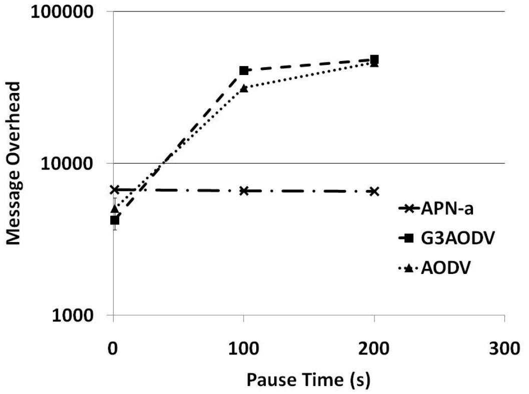

Figure 12: Message overhead (S2D, 5 pairs)

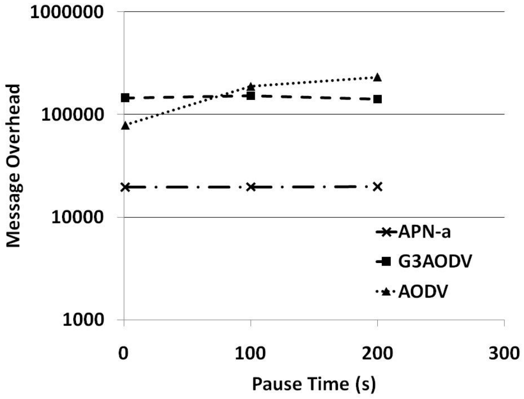

Figure 13: Message overhead (S2D, 30 pairs)

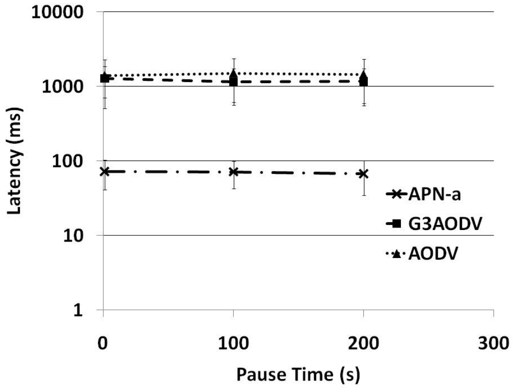

Figure 14: Latency (S2D, 5 pairs)

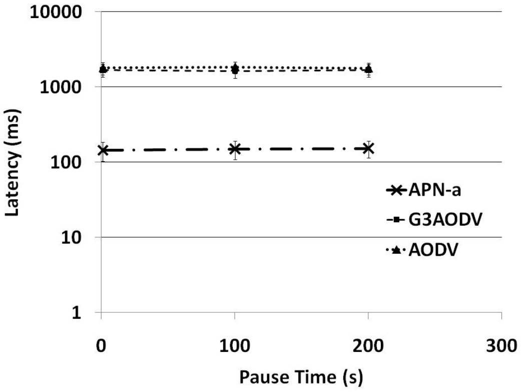

Figure 15: Latency (S2D, 30 pairs)

formance, thus our testbed suffices. In particular, we compared message delivery rate of the APN ($C(8(3))$ with that of the Gossip algorithm (p = 85 percent). Moreover, we developed a chat (Java) application that our lab team used to test the APN over one week and a network monitoring tool to track APN's overheads. In particular, we evaluated the APN for transmission of 100 messages between each of 2 pairs of S2D nodes with transmission delay varying from 5 s to 60 s. Overall, the results indicated on average that the message delivery rate achieved 84 percent for the APN and the Gossip algorithm; in addition, the APN protocol delivered messages with 2.4 hops, 19ms latency between two neighbor hosts, 360ms for application-to-application communication which includes 200 ms of Java's overhead, and 1,450 for message overhead. These experimental results

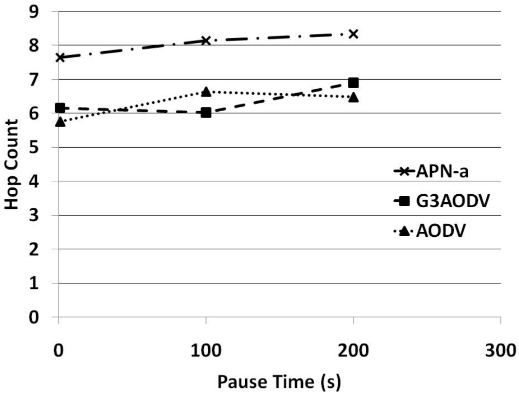

Figure 16: Hop Count (S2D, 5 pairs)

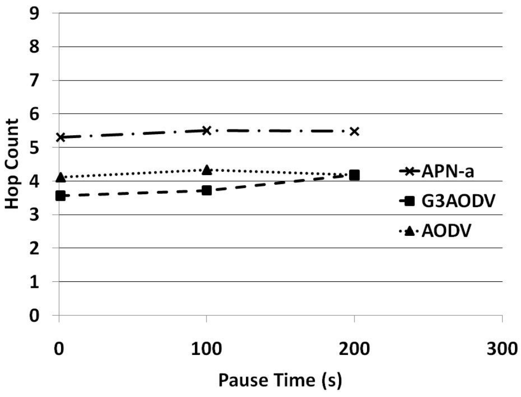

Figure 17: Hop Count (S2D, 30 pairs)

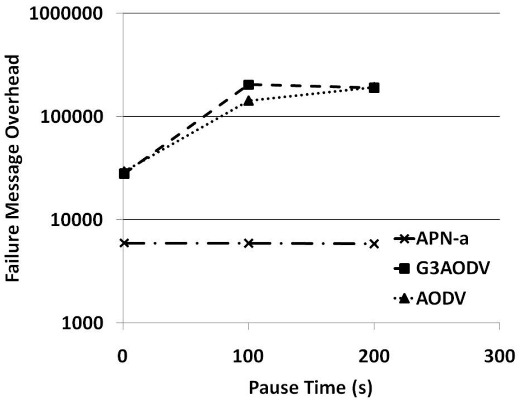

Figure 18: Failure messages (S2D, 5 pairs)

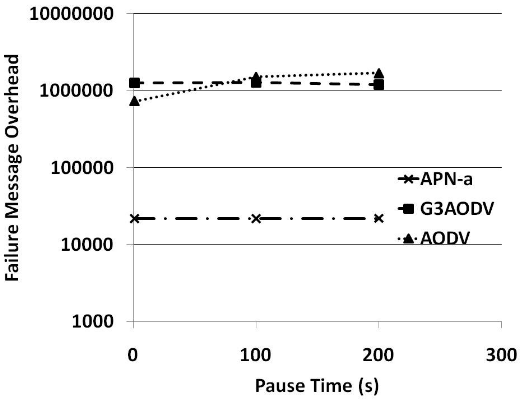

Figure 19: Failure messages (S2D, 30 pairs)

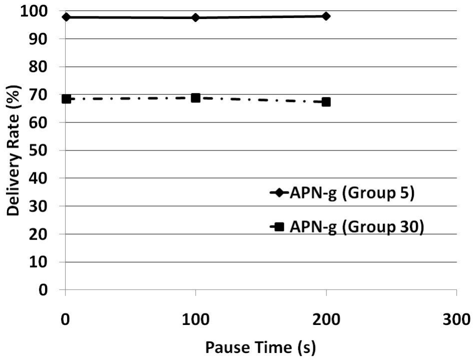

Figure 20: Delivery rate in AP addressing

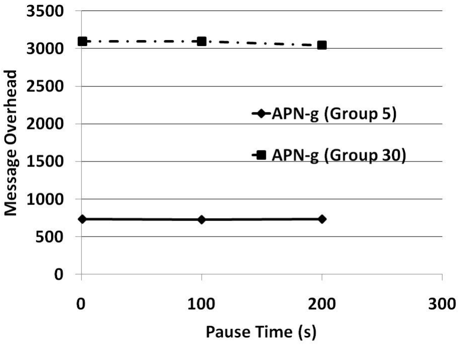

Figure 21: Message overhead in AP addressing

were similar to the simulation ones, though using a different transmission standard (Zigbee); specifically, they supported the simulation results and confirmed the significant impact of transmission delay on the protocol performance as well as they indicated the feasibility of using APNs in practice.

3.3 Discussion

The above analyses showed that APNs exhibit bimodal behavior for the two different probabilistic distributions with a small difference in message delivery ratios. The use of APs as node addresses and as identifiers of local applications was validated by our simulations and experiments, whose results indicated that AP-based MANETs are significantly more effective than IP-based MANETs for the scenarios we studied. In addition, APNs can be adapted to run IP-based applications.

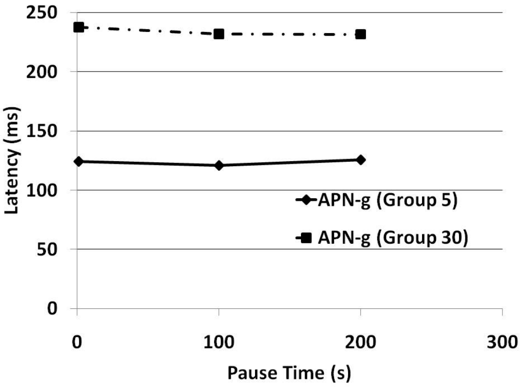

Figure 22: Latency in AP addressing

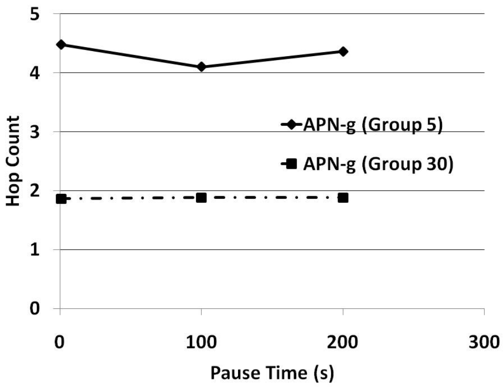

Figure 23: Hop count in AP addressing

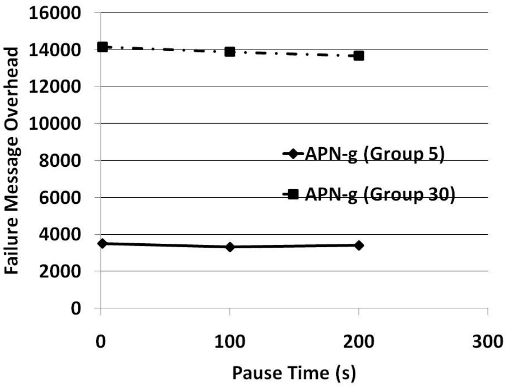

Figure 24: Failure messages in AP addressing

In contrast to G3AODV and AODV, APN protocols do not implement routing tables, thus they do not incur associate maintenance messages and transmission interference, which allow APNs to significantly reduce message overhead and to achieve one order of magnitude lower latency. Moreover, active prefixes enable cross-layer communication to further reduce end-to-end latency of applications. Surprisingly, APN protocols were insensitive to pause time; thus, the simulation results of APN performance should also be valid for other mobility models, in addition to the random waypoint one.

We also note that other mechanisms developed for IP-based protocols for MANETs can be integrated into APN protocols to produce even more effective APN protocols for specific ad-hoc networks. For example, the APs of messages can be stored in the intermediate nodes to be used as message footprints by the APN protocols to implement efficient routing in wireless sensor networks.

These promising results corroborate our argument for adopting an inverse approach to building routing protocols for MANETs, because APN protocols are excellent candidates for group applications (as is the case with many ad-hoc applications) and they provide superior S2D addressing performance to efficiently support IP-based applications in comparison to representative IP-based routing protocols for MANETs.

4. RELATED WORK

There is a vast literature on routing protocols for MANETs with specific methods for route discovery and maintenance messages for node identification and naming [5]. They mostly inherited IP-based addressing and topological node identification [5, 6, 13]. To improve performance of routing protocols, Gossip-based algorithms [7, 11] have been proposed such as the representative G3AODV protocol [8], which implements the Gossip3 algorithm for probabilistic forwarding in the AODV protocol. Although AP networks also use probabilistic forwarding they do not rely on the topological identification of nodes or routing tables; instead, they use active prefixes to support node addressing, application identification, and probabilistic forwarding in a distributed way.

Content-based networks [4] support publish / subscribe overlays on MANETs, using IP-based addressing, and also exhibit high maintenance overheads. Moreover, the use of content for node identification is challenging because of classical database problems [6, 15]. In contrast, APNs have no centralized identification control and use application-defined interests only as matching filters to find and establish preliminary communication between APN nodes.

Active messages (AMs) were initially proposed as a communication architecture for highly-dense multiprocessing systems, after which they were used as a TinyOs network component for sensor networks [12]. Basically, message headers in AMs contain information for message handlers at the hardware level in each node to forward them to their destination. In contrast, the active prefix uses information from the application layer

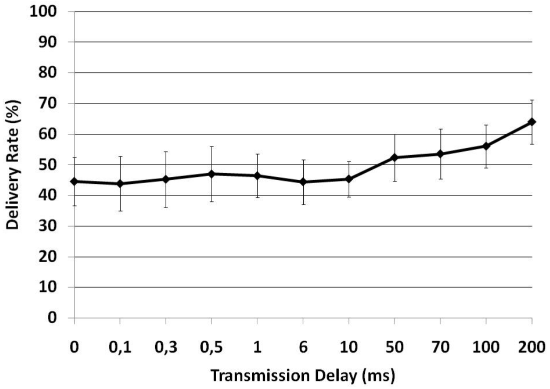

Figure 25: Message delivery rate

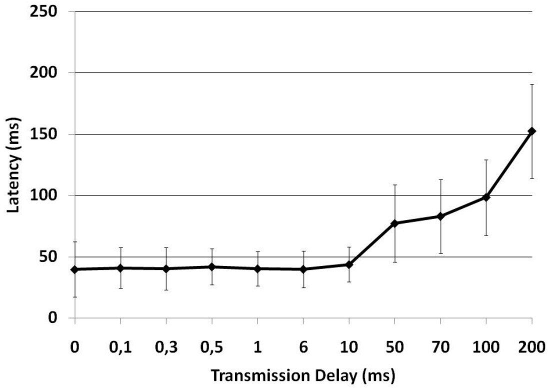

Figure 26: Latency

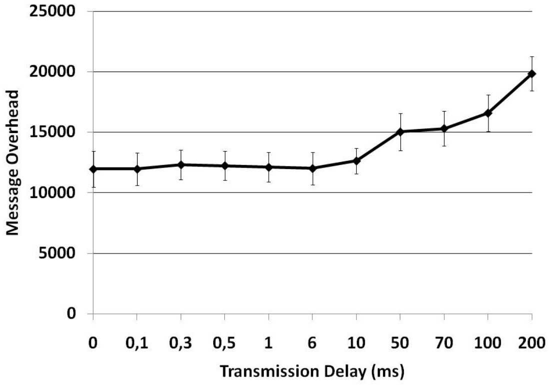

Figure 27: Message overhead

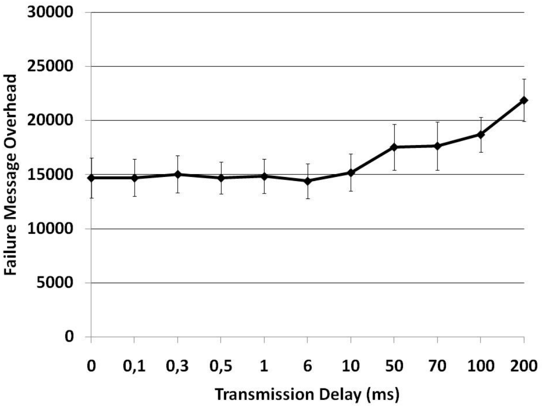

Figure 28: Failure messages

to both forward and address at a higher level, regardless of the hardware or operating system.

Recent works on naming [10, 13] are the closest to ours as they are not IP-based but data-centric. In one study [10], nodes communicate by sending messages based on user's information and interests and utilize the Direct Diffusion method by means of gradients to reach destinations. Direct Diffusion generates a new gradient for each new piece of information, which is stored in network nodes even if a network node has no interest in the message. They use in-networking processing, a list of attribute-value-operation tuples, and cache data. By contrast, APNs neither cache any data nor perform in-networking processing, but they use active prefixes for addressing and forwarding and use AP interest attributes only for primary matching of application interests. Also, their work does not support node addressing, and thus no end-to-end addressing is possible; in contrast, APNs use APs as node addresses and provide source-to-destination addressing.

Other authors [13] proposed an algorithm for message forwarding in mobile ad-hoc networks based on distance tables stored in the nodes and message headers that indicate source and destination distances, requested data name, and source/destination identification possibly using MAC addresses. By contrast, AP networks decouple application interest from message forwarding, use multivariate probability distribution for message forwarding, AP-based identification of nodes and local applications, do not cache data in intermediary nodes, and can support IP-based applications.

5. CONCLUSION

This paper introduces the active prefix (AP), a novel communication element consisting of node characteristics and application interests used to build efficient AP-based mobile ad-hoc networks (APNs), in which APs are used for probabilistic message forwarding and addressing, and also as matching filters for searching for nodes with the same application interest over APNs. A distinguishing property of APs is that they enable nodes and application interests to be uniquely identified in distributed way. In addition, APs support interest group communication and source-to-destination addressing; besides, they allow nodes to implement cross-layer communication. Moreover, APNs do not rely on IP addressing or identification, centralized IP access control, or routing tables, but support communication focused on users / applications and can be adapted to run IP-based applications.

Simulation results indicated that AP - based protocols for MANETs are significantly more effective than either AODV or G3AODV, two representative IP-based protocols for MANETs, in a scenario with 150 mobile nodes. Furthermore, the simulation results were supported by the experimental ones based on the performance of a 20 - host APN and a chat application. Overall, the combination of our simulation and experimental results indicates the feasibility of using APs as effective building blocks for ad-hoc networks, in particular for mobile ad-hoc networks.

These promising results strongly support our approach to the problem of building effective routing protocols for MANETs; besides, the performance of APN protocols indicates that they are excellent candidates for applications based on group communication, which is the case with many MANET applications such as service - discovery and collaborative virtual environments. On-going work includes development of APNs and applications for smartphones, run Internet application benchmarks on APNs, and development of adaptations of APNs to wired networks, in particular to the Internet.

6. REFERENCES

[1] I. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: A survey. *Computer Networks*, 47:445–487, 2005.

[2] A. Boukerche, A. Zarrad, and R. B. Araujo. A cross-layer approach-based gnutella for collaborative virtual environments over mobile ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 21:911–924, 2010.

[3] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 85–97, New York, NY, USA, 1998. ACM.

[4] A. Carzaniga and C. P. Hall. Content-based communication: a research agenda. In *SEM '06: Proceedings of the 6th international workshop on Software engineering and middleware*, pages 2–8, New York, NY, USA, 2006. ACM.

[5] M. Conti and S. Giordano. Multihop ad hoc networking: The theory. *Communications Magazine, IEEE*, 45(4):78 –86, april 2007.

[6] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, PODC '87, pages 1–12, New York, NY, USA, 1987. ACM.

[7] R. Friedman, D. Gavidia, L. Rodrigues, A. C. Viana, and S. Voulgaris. Gossiping on manets: the beauty and the beast. *SIGOPS Oper. Syst. Rev.*, 41:67–74, October 2007.

[8] Z. Haas, J. Halpern, and L. Li. Gossip-based ad hoc routing. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1707 – 1716 vol.3, October 2002.

[9] Z. J. Haas, J. Y. Halpern, and L. Li. Gossip-based ad hoc routing. *IEEE/ACM Trans. Netw.*, 14(3):479–491, 2006.

[10] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan. Building efficient wireless sensor networks with low-level naming. *SIGOPS Oper. Syst. Rev.*, 35:146–159, October 2001.

[11] A. V. Kini, V. Veeraraghavan, N. Singhal, and S. Weber. Smartgossip: an improved randomized broadcast protocol for sensor networks. In *IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks*, pages 210–217, New York, NY, USA, 2006. ACM.

[12] P. Levis, E. Brewer, D. Culler, D. Gay, S. Madden, N. Patel, J. Polastre, S. Shenker, R. Szewczyk, and A. Woo. The emergence of a networking primitive in wireless sensor networks. *Commun. ACM*, 51:99–106, July 2008.

[13] M. Meisel, V. Pappas, and L. Zhang. Ad hoc networking via named data. In *Proceedings of the fifth ACM international workshop on Mobility in the evolving internet architecture*, MobiArch '10, pages 3–8, New York, NY, USA, 2010. ACM.

[14] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing. RFC 3561, 2003.

[15] G. V. Popescu and Z. Liu. Network overlays for efficient control of large scale dynamic groups. In *DS-RT '06: Proceedings of the 10th IEEE international symposium on Distributed Simulation and Real-Time Applications*, pages 135–142, Washington, DC, USA, 2006. IEEE Computer Society.

[16] A. C. Viana, M. D. de Amorim, Y. Viniotis, S. Fdida, and J. F. de Rezende. Twins: A dual addressing space representation for self-organizing networks. *IEEE Transactions on Parallel and Distributed Systems*, 17:1468–1481, 2006.