Performance Analysis for Node Auto Addressing in Mobile Ad Hoc Networks

Mr.S. Mohanalingam[1]Mr.D.Bright Anand M.Tech (Ph.D)[2] Mrs.S,Priya[3] M.Tech[3]
[1]M.E (Networking and Internet Engineering) [2]Assistant Professor of Dept. of Computer Science & Engineering[3]Assistant Professor
[1,2]RatnaVel Subramaniam College of Engineering & Technology, Dindigul – 624 005 [3]G.Tech Engg College

Abstract---The key challenge in ad hoc networks is the Address assignment, due to the lack of infrastructure. A distributed and self-managed mechanism, to avoid address collisions in a dynamic network with fading channels, joining/leaving of nodes, and frequent partitions, require in an autonomous addressing protocols. A bloom filter is a simple, space-efficient, randomized data structure for concisely representing a static data set, in order to support approximate membership queries. It has great potential for distributed applications where systems need to share information about what resources they have. The space efficiency is achieved at the cost of a small probability of false positive in membership queries. However, for many applications the space savings and short locating time consistently outweigh this drawback. This paper introduces Dynamic Bloom Filters (DBF) to support concise representation and approximate membership queries of dynamic sets, and study the false positive probability and union algebra operations. It proves that DBF can control the false positive probability at a low level by adjusting the number of standard bloom filters used according to the actual size of current dynamic set. The space complexity is also acceptable if the actual size of dynamic set does not deviate too much from the predefined threshold. Thus this paper propose and analyze a dynamic bloom filter based addressing that configures mobile ad hoc nodes based on a distributed address database stored in filters that reduces the control load and makes the proposal robust to packet losses and network partitions. And also it evaluates the performance of the protocol, considering joining nodes, partition merging events, and network initialization. Simulation results show that the Dynamic Bloom Filter resolves all the address collisions and also reduces the control traffic when compared to previously proposed protocols.

I. INTRODUCTION

A mobile ad-hoc network (MANET) is the collection of mobile nodes that are equipped by several wireless mobile devices which are using for communication. The transmission of the data of a particular mobile node is received by all nodes within its transmission range. It is because of broadcast nature of wireless communication and help of directional antennas. Other mobile hosts located between the two wireless hosts can forward their messages, which are out of their transmission ranges in the ad hoc networks. It will effectively improve the performance of the MANET. Each host needs to be equipped with the capability of an autonomous system due to the mobility of wireless hosts, or a routing function without any statically established infrastructure or centralized administration. Without notifying other nodes the mobile nodes can move and turned on or off. Mobility and autonomy introduces a dynamic topology of the networks and it is because of is transient nature of the end host and intermediate hosts on a communication path.

Mobile Ad hoc Networks do not rely on extraneous fixed infrastructure and can be installed without base station and dedicated routers. This makes the nodes as ideal candidate nodes for rescue and emergency operations. The nodes in these networks have limitations in battery power and bandwidth, and each node needs the assistance from other nodes to forward their packets. The conventional protocols like WRP, DSDV, AODV and DSR are assuming that all the nodes in MANET are cooperative fully and it always does so truthfully.

II. OVERVIEW OF AD- HOC NETWORK

A crucial and usually unaddressed issue of ad hoc networks is the frequent network partitions. Network partitions, caused by node mobility, fading channels and nodes joining and leaving the network, can disrupt the distributed network control. Network initialization is another challenging issue because of the lack of servers in the network Information representation and query processing are two core problems of many computer applications, and are often associated with each other. Representation means organizing information according to some format and mechanism, and making information operable by the corresponding method. Query processing means making decisions about whether an element with a given attribute value belongs to a given set.

A standard Bloom filter (SBF) is a space-efficient data structure for representing a set and answering membership queries within a constant delay. The space efficiency is achieved at the cost of false positives in membership queries, and for many applications, the space savings outweigh this drawback when the probability of an error is sufficiently low. Although the SBF and its variations have found suitable applications in different fields, the following three obstacles still lack suitable and practical solutions.

For stand-alone applications that know the upper bound on set cardinality for a dynamic set in advance, a large number of bits are allocated for an SBF to represent all possible items of the dynamic set at the outset. This approach diminishes the space-efficiency of the SBF, and should be replaced by new Bloom filters which always use an appropriate number of bits as set cardinality changes.

For stand-alone applications that do not know the upper bound on set cardinality of a dynamic set in advance, it is difficult to accurately estimate a threshold of set size and assign optimal parameters to an SBF in advance. In the

event that the cardinality of the dynamic set exceeds the estimated threshold gradually, the SBF might become unusable due to a high false positive probability.

For distributed applications, all nodes adopt the same configuration in an effort to guarantee the inter-operability of SBFs between nodes. In this case, all nodes are required to reconstruct their local SBFs once the set size of any node exceeds a threshold value at the cost of large (sometimes huge) overhead. In addition, this approach requires that the nodes with small sets must sacrifice more space so as to be in accordance with nodes with large sets, hence reducing the space-efficiency of SBFs and causing large transmission overhead.

The SBF and variants do not take dynamic sets into account. To address the three obstacles, the proposed filter, the Dynamic Bloom filters (DBF) to represent a dynamic set instead of rehashing the dynamic set into a new filter as the set size changes.

The Dynamic Bloom filter can control the false positive probability at a low level by expanding its capacity as the set cardinality increases. Through comprehensive mathematical analysis, it's clearly shown that the dynamic Bloom filter uses less expected memory than the Bloom filter when representing dynamic sets with an upper bound on set cardinality, and also that the dynamic Bloom filter is more stable than the Bloom filter due to in frequent reconstruction when addressing dynamic sets without an upper bound on set cardinality. Moreover, the analysis results hold in standalone applications as well as distributed applications.

III. Problem Definition

Automatic address allocation is more difficult in a MANET environment than that in hardwired networks due to instability of mobile nodes, low bandwidth of wireless links, openness of MANET, and lack of central administration. Therefore, more overhead occurs to avoid address conflict compared to the protocols for hardwired networks, such as DHCP and SAA. However, since address allocation is the first step toward the practical application of the MANET, it is worth further research effort.

Before discussing address allocation issues, several scenarios are described to illustrate the difficulty of the problem. In the simplest scenario, a mobile node joins and then leaves a MANET once, such as nodes A and B illustration Figure 1. An unused IP address is allocated on its arrival and becomes free on its departure.

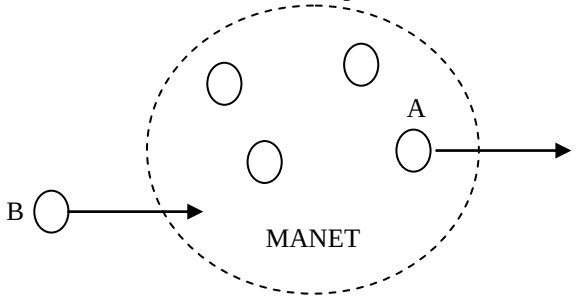

Fig. 1: A node joins and leaves the MANET once.

However, nodes are free to move arbitrarily during it session in the MANET. If one or more configured nodes go out of others transmission range for a while, the network becomes partitioned as illustrated in Figure 2 (a). When they approach each other, the partitions merge later. Because mobile nodes may not be aware of partitioning, they still use the previously allocated IP addresses. If a new node, say B, arrives at one partition and is assigned an IP address belonging to the other partition, say A's IP address.

Another scenario is when two separately configured MANETs merge, which is illustrated in Fig. 3. Because address allocation in one MANET is independent of the other, there may be some duplicate addresses in both of them. For example, node A in MANET 1 has the same IP address as node B in MANET 2. As a result, some (or all) nodes in one MANET may need to change their addresses.

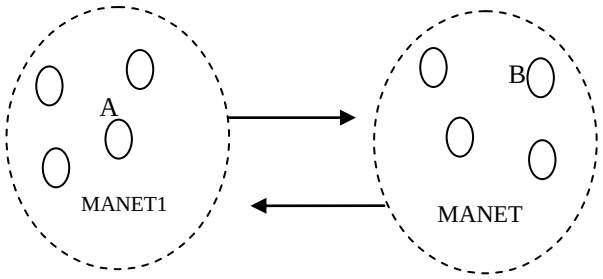

Fig. 2: Mergers of Two Independent MANET's

In another scenario, a mobile node leaves one MANET and then joins another MANET. This node could be regarded as the special case of the situation mentioned above because the single node could be viewed as a one-node partition. The last scenario is fairly rare. Suppose there are two independent MANETs that are close to each other. A node in between decides to join a MANET nearby and functions as a relay node, which leads to connection of the two MANETs. This is the same as merger of two independent MANETs.

In summary, a feasible auto configuration algorithm should handle the following three general scenarios:

A. Scenario A:

A mobile node simply joins a MANET and then leaves it forever;

B. Scenario B:

A MANET partitions and then the partitions merge later;
Scenario C: Two separately configured MANETs merge.

IV. Objective

The main objective of this project is to reduce the expected memory, to reduce low communication overhead and low latency, to resolve all address collisions in network partition merging events on a dynamic network environment.

For a static set, it is possible to know the whole set in advance and design a perfect hash function to avoid hash collisions. In reality, an SBF (Standard Bloom Filter) is usually used to represent dynamic sets as well as static sets. Therefore it is impossible to know the whole set and design *k* perfect hash functions in advance. Hence the DBF (Dynamic Bloom Filter) focuses mainly on these and fits well on the dynamic network environment to meet these objectives.

A. Dynamic Host Configuration Protocol (Dhcp):

DHCP was developed as a predecessor to BOOTP. This provides configuration parameters to internet host that consists of two components: first one is a protocol for

delivering host-specific configuration parameters from a DHCP server to a host and second one is a mechanism for allocation of network addresses to hosts. DHCP is built on a client-server model, in which designated DHCP server hosts allocate network addresses and deliver configuration parameters to dynamically configured hosts. DHCP supports three mechanisms for IP address allocation.

B. Reverse Address Resolution Protocol (Rarp):

RARP protocol that belongs to TCP/IP group allows a computer to obtain its IP address from a RARP server in the bootstrap procedure [20]. Before obtaining an IP address, a computer has to use its MAC address to communicate with others. It first broadcasts a RARP request that specifies itself as a target. The RARP server on the same network keeps the database of IP addresses. Upon receiving a RARP request message, the RARP server looks up the IP address based on the requester's physical address and replies to the requester.

C. Bootstrap Protocol (Bootp):

BOOTP was developed to overcome some of the drawbacks of RARP [21]. It uses UDP to carry messages and hence it can be implemented with an application program. Before obtaining an IP address, a computer can broadcast an IP datagram on the local network by using the limited broadcast IP address 255.255.255.255.The BOOTP server then broadcasts the reply message on the local network, which contains the requester's IP address, the router's IP address, etc. BOOTP is designed for a relatively static environment, and it provides only a static mapping from the physical address to the corresponding network parameters. It is not suitable for a dynamic environment.

D. Strong Duplicate Address Detection (Sdad):

The SDAD presented in [6] is the base for all stateless approaches. It consists of a simple mechanism that allows an ad hoc node to choose an IP address and test if it's already used or not. We can consider this proposal as an extension of the Zero conf. for multi hope networks. When a node initializes, it picks 2 addresses, a "temporary address" and a "tentative address" in the range 169.254/16 (02047 and 204865534 respectively). The temporary address is used only in the initialization phase as a source address for requests flooded to detect if the tentative address is already used or not. The new node floods the network with an address request (ICMP) packet destined to the tentative address and waits a certain period of time.

E. Weak Duplicate Address Detection (WDAD):

The WDAD proposed in aim at extending the duplicate address detection mechanism for the whole lifetime of the network. The idea behind WDAD is that duplicate addresses may be tolerated as long as packets reach the destination node intended by the sender, even if the destination node's address is also being used by another node. That's why each node selects an identification key to make routing capable of differentiating between potential duplicate IPs. Each node generates a key at initialization phase, and distributes it with its IP address in all routing messages.

F. Passive Duplicate Address Detection (PDAD)

PDAD is a duplicate detection mechanism designed for link state routing protocols. The idea behind PDAD protocol is that instead of explicitly trying to detect and solve address duplication by sending control information, each node can investigate routing information and deduce address duplication from events that never occur in case of unique addresses but do occur if there are address duplicates. With proactive routing, the nodes periodically flood the network to inform other nodes about their neighborhood.

G. Ad Hoc IP Address Auto configuration:

The Internet draft presented in combines the mechanisms of SDAD and WDAD to accomplish address consistency. Thus the duplication detection mechanism not only checks for duplication during initialization, but also checks and resolves potential address duplication detected by intermediate nodes using routing messages.

H. Zero Configuration Networking (ZEROCONF):

Address configuration without a dedicated server has been investigated by the Zero Configuration Networking (Zero conf) working group of the Internet Engineering Task Force (IETF). The goal of the Zero conf Working Group is to enable networking in the absence of configuration and administration. The Internet draft [5] describes a method for dynamic configuration of IPv4 link-local addresses used for local communications. When a node wishes to configure a link-local address, it selects an address pseudo-randomly, uniformly distributed in the range 169.254.1.0 to 169.254.254.255. Then it tests whether or not this address is already in use by broadcasting an ARP request for the desired address.

I. IPV6 Stateless Address Auto Configuration:

IPv6 stateless address auto configuration is performed only on multicast capable links [8]. A node starts the auto-configuration mechanism by generating a link-local address for its interface. This link-local address is generated by appending the interface's identifier to the well-known link-local prefix. Before assigning the link-local address to its interface, a node must attempt to verify that this link-local address is not used by another node on the same network. This is done by the Duplicate Address Detection (DAD) procedure.

J. Dynamic Address assignment Protocol (DAP):

The Dynamic Address assignment Protocol in mobile ad-hoc networks (DAP), which is based on the allocation of available address sets for each node, on Hellos, and on network identifiers. In DAP; a node subdivides its available address set with a joining node whenever it is argued for an address by the joining node. When a node has an empty address set, it asks for an address set reallocation. This reallocation and the detection that a given address is not being used anymore can cause a high control load in the network, depending on how the addresses are distributed among nodes.

V. EXISTING APPROACH

A. Rreqs:

Perkins et al. propose a solution for address auto-configuration in ad hoc networks. An address is randomly chosen within the range 2048 to 65534 from the 169.254/16 address block. A node floods Route Requests (RREQs) for the selected IP address. If no Route Reply (RREP) is received within a timeout period, the node retries for RREQ RETRIES times. At the end of all the retries, if no response is received, the chosen IP address is assumed to be free. The node assigns itself that IP address. Here the latency is the timeout value multiplied by RREQ RETRIES.

B. Ipv6:

IPv6 Stateless Auto-configuration specifies the steps a node takes to configure its interfaces in IPv6. The steps include construction of link-local address, Duplicate Address Detection, and construction of a site-local address. During Duplicate Address Detection for MANETs flooding is required, thus making the approach a scalable. To overcome this scalability issue, an extension is proposed in by building a hierarchical structure. But the cost incurred in maintaining such a hierarchical structure may be high.

C. Fap:

The Filter based Addressing Protocol (FAP), is based on the Bloom Filter technique where, the Bloom filter and its variants just focus on how to represent a static set and decrease the false positive probability to a sufficiently low level. By investigating mainstream applications based on the Bloom filter, it reveals that dynamic data sets are more common and important than static sets. However, existing variants of the Bloom filter cannot support dynamic data sets well.

VI. Proposed Approach

A. Dynamic Filter based Addressing Protocol (D-FAP):

The proposed protocol aims to dynamically auto configure network addresses, resolving collisions with a low control load, even in joining or merging events. To achieve all these objectives, D-FAP uses a distributed compact filter to represent the current set of allocated addresses. This filter is present at every node to simplify frequent node joining events and reduce the control overhead required to solve address collisions inherent in random assignments. Moreover, in the proposed system included the filter signature, which is the hash of the address filter, as a partition identifier. The filter signature is an important feature for easily detecting network merging events, in which address conflicts may occur.

There are two different filters used in proposed work depending on the scenario: the Dynamic Bloom filter, which is based on hash functions, and the Sequence filter, which compresses data based on the address sequence.

B. Advantages of Proposed Approach:

- The dynamic Bloom filter uses less expected memory than the Bloom filter.
- In stand-alone applications, a DBF can enhance its capacity on-demand via an item insertion operation. It can also control the false positive probability at an acceptable level as set cardinality increases. DBFs can shorten their capacities as the set cardinality decreases through item deletion and merge operations.
- In distributed applications, DBFs always satisfy the requirement of interoperability between nodes when handling dynamic sets and occupying a suitable amount of memory to avoid unnecessary waste and transmission overhead.
- In standalone, as well as distributed applications, DBFs use less expected memory than SBFs when dealing with dynamic sets that have an upper bound on set cardinality. DBFs are also more stable than SBFs due to infrequent reconstruction when dealing with dynamic sets that lack an upper bound on set cardinality.

VII. System Model

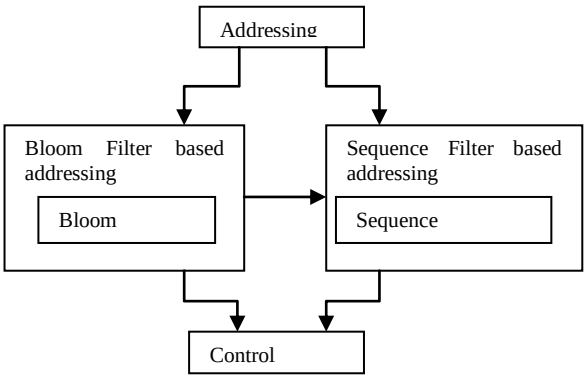

Fig. 3.1: System Architecture Diagram

A. Dynamic Bloom Filter:

DBF seems to be a logical addition to BF for a scalable environment - just before the false positive (FP) rate of a particular BF starts growing fast, so simply switch to a new filter and store the old one. In the following algorithm description, refer to a constant n_0. This constant is fixed in advance and indicates the number of elements that want to store in each filter. This paper present somewhat simplified versions of the algorithms given in the original paper and omit some details to underline simplicity. In the algorithms, DBF is an object maintaining: n (the number of elements), n_0 (the threshold parameter) and filters (array of legacy bloom filters).

B. Sequence Filter:

The other filter structure that proposed is called Sequence filter, and it stores and compacts addresses based on the sequence of addresses. This filter is created by the concatenation of the first address of the address sequence, which we call initial element, with an -bit vector, where is the address range size. In this filter, each address suffix is represented by one bit, indexed by, which gives the distance between the initial element suffix and the current element suffix. If a bit is in 1, then the address with the given suffix is considered as inserted into the filter; otherwise, the bit in 0 indicates that the address does not belong to the filter.

C. Filter Selection:

The best filter for FAP depends on network characteristics such as the estimated number of nodes in the network and the number of available addresses. It also depends on the false-positive and false-negative rates of the filter. Bloom filters do not present false negatives, which mean that a membership test of an element that was inserted into the filter is always positive. Hence, a membership test of an element that was not inserted into the Bloom filter may be positive. The size of the Sequence filter depends only on the size of the address range, and on the address size A_s .The address range size is defined by the number of bits in address suffix, b, so that r = 2^b .The number of bits in the Sequence filter is given by $S = A_s + r$.

D. Probability of Collision in FAP:

In this module, it analyzes FAP to evaluate the probability that our scheme causes an address collision. A collision occurs when two different joining nodes generate AREQs with the same address and the same identifier number or if two disjoint partitions own exactly the same filters. In the

first case, the joining nodes do not notice that their addresses are the same because the message from the other node seems to the first node like a retransmission of its own message.

E. Control Overhead Estimation

The main procedures in addressing protocols are network initialization, node joining/leaving, and merging. Usually, these procedures, as well as the ordinary protocol operation, generate control overhead, reducing the available bandwidth. And also estimate the number of control messages sent by all these procedures for FAP, the extension of DAD [4], hereafter called DAD with partition detection (DAD-PD), and MANET conf (M conf).

VIII. CONCLUSION

This paper proposed a distributed and self-managed addressing protocol, called Dynamic Filter-based Addressing protocol, which fits well for dynamic ad hoc networks with fading channels, frequent partitions, and joining/leaving nodes. The key idea is to use address filters to avoid address collisions, reduce the control load, and decrease the address allocation delay Moreover, the Dynamic Filter-based Protocol increases the protocol robustness to message losses, which is an important issue for ad hoc networks with fading channels and high bit error rates. The use of the hash of the filter instead of a random number as the partition identifier creates a better representation of the set of nodes. Hence, a change in the set of nodes is automatically reflected in the partition identifier. This identifier is periodically advertised, allowing neighbors to recognize if they belong to different sets of nodes. In the other proposals, a mechanism to change the arbitrated partition identifier is requested, which increases the complexity and the packet overhead of the protocol.

IX. FUTURE WORK

The proposed protocol efficiently resolves all address collisions even during merging events, as showed by simulations. This is achieved because D-FAP is able to detect all merging events and also because D-FAP is robust to message losses. D-FAP initialization procedure is simple and efficient, requiring a control load similar to the control load of DAD, which is a protocol with a small overhead but that does not handle network partitions. Moreover, D-FAP presents smaller delays in the joining node procedure and on network partition merging events than the other proposals, indicating that the proposed protocol is more suitable for very dynamic environments with repeated partition merging and node joining events. In future Dynamic Bloom Filter can be used and implemented for better features than Bloom filters when dealing with dynamic sets. Dynamic Bloom Filters use less expected memory than Bloom filters when dealing with dynamic sets with upper bounds on set cardinality, and that Dynamic Bloom filters are more stable than Bloom filters due to infrequent reconstruction when addressing dynamic sets without upper bounds on set cardinality.

REFERENCES

[1] N. C. Fernandes, M.D.Moreira, and O. C. M. B. Duarte, "Aself-organized mechanism for thwarting malicious access in ad hoc networks," in Proc. 29th IEEE INFOCOM Miniconf., San Diego, CA, Apr.

[2] N. C. Fernandes, M.D.Moreira, and O. C. M. B. Duarte, "An efficient filter-based addressing protocol for auto configuration of mobile ad hoc networks," in Proc. 28th IEEE INFOCOM, Rio de Janeiro, Brazil.

[3] N. H. Vaidya, "Weak duplicate address detection in mobile ad hoc networks," in Proc. 3rd ACM MobiHoc,.

[4] M. D. D. Moreira, R. P. Laufer, P. B. Velloso, and O. C.M. B. Duarte, "Capacity and robustness tradeoffs in Bloom filters for distributed applications," IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 12, pp. 2219–2230.

[5] H. Kim, S. C. Kim, M. Yu, J. K. Song, and P.Mah, "DAP: Dynamic address assignment protocol in mobile ad-hoc networks".

