International Journal of Computer Applications Technology and Research
Volume 3– Issue 6, 325 - 328, 2014, ISSN: 2319–8656

Distributed Addressing Protocol for Node Auto configuration in Ad Hoc Networks using Bloom Filters

Bini M Issac
Amal Jyothi College of Engineering
Kanjirappally, India

Deepu Benson
Amal Jyothi College of Engineering
Kanjirappally, India

Abstract: The importance of wireless ad hoc networks in community and commercial connectivity cannot be underestimated in view of the benefits associated with such networks. An ad hoc network must assemble itself from any devices that happen to be nearby, and adapt as devices move in and out of wireless range. High levels of self-organization will minimize the need for manual configuration. However, efficiently providing unique addresses in ad hoc networks is still an open research question. The goal of this paper was to develop algorithms for address auto-configuration. The paper addresses the following among other problems: Achieving high levels of address uniqueness without compromising on latency and communication overhead. The proposed protocol uses bloom filters to reduce the storage overhead and remove communication overhead.

Keywords: Ad Hoc Networks, Bloom filters, Simulation, IP address, Birthday paradox

1. INTRODUCTION

An ad hoc network is a type of peer to peer wireless network mode where wireless devices communicate with each other directly, without the aid of a wireless access point device. Wireless networks typically depend on a base station or WAP device to manage and direct the stream of data between wireless devices. In an ad hoc setup, the network is built spontaneously as and when devices communicate with each other. These devices should ideally be within close range of each other; however quality of connection and speed of the network will suffer as more devices are added to the network. The term “ad hoc” tends to imply “can take different forms” and “can be mobile, stand alone, or networked”. Ad hoc implies that the network is formed in a spontaneous manner to meet an immediate demand and specific goal. Ad hoc networks have the ability to form “on the fly” and dynamically handle the joining or leaving of nodes in the network. Mobile nodes are autonomous units that are capable of roaming independently. Typical mobile ad hoc wireless nodes are Laptops, PDAs, Pocket PCs, Cellular Phones, Internet Mobile Phones, Palmtops or any other mobile wireless devices. Mobile ad hoc wireless devices are typically lightweight and battery operated.

2. BACKGROUND

The nodes of a network need some mechanism to interchange messages with each other. The TCP/IP protocol allows the different nodes from the network to communicate by associating a distinct IP address to each node of the same network. In wired or wireless networks with an infrastructure, there is a server or node which correctly assigns these IP addresses. Mobile ad hoc networks, on the other hand, do not have such a centralized entity able to carry out this function. Therefore, some protocol that performs the network configuration in a dynamic and automatic way is necessary, which will utilize all the nodes of the network (or only part of them) as if they were servers which manage IP addresses.

2.1 Related Work

Address auto configuration proposals that do not store the list of allocated addresses are typically based on a distributed protocol called Duplicate Address Detection (DAD) [4]. In this protocol, every joining node randomly chooses an address and floods the network with an Address Request message (AREQ) for a number of times to guarantee that all nodes receive the new allocated address. If the randomly chosen address is already allocated to another node, this node advertises the duplication to the joining node sending an Address Reply message (AREP). When the joining node receives an AREP, it randomly chooses another address and repeats the flooding process. Otherwise, it allocates the chosen address. This proposal does not take into account network partitions and is not suitable for ad hoc networks [1].

Other proposals use routing information to solve the addressing problem. Weak DAD [3], for instance, routes packets correctly even if there is an address collision. In this protocol, every node is identified by its address and a key. Collisions with the other nodes are identified by information from the routing protocol. Weak DAD can continuously detect duplicate addresses with information added to routing protocol packets. The main idea is to add a key to each address that is distributed by the routing protocol. Thus, the routing protocol packet format has to be modified. Other more complex protocols were proposed to improve the performance of network merging detection and address reallocation [6]. In these protocols, nodes store additional data structures to run the addressing protocol.

MANETconf [4] is a stateful protocol based on the concepts of mutual exclusion of the Ricart Agrawala algorithm. Using MANETconf, each configured node is able to assign addresses to new nodes and, therefore, maintains an allocation table of already assigned addresses in the network. In this protocol, nodes store two address lists: the Allocated list and the Allocated Pending list. A joining node asks for an address to a neighbour, which becomes a leader in the address allocation procedure. The leader chooses an available address, stores it on the Allocated Pending list, and floods the network. If all MANETconf nodes accept the allocation request and

positively answer to the leader, then the leader informs the allocated address to the joining node, moves the allocated address to the Allocated list, and floods the network again to confirm the address allocation. After receiving this message, each node moves the address from the Allocated Pending list to Allocated list. MANETconf handles address reallocation, but partition detection depends on periodic flooding. Therefore, this protocol incurs in a high control overhead.

3. PROPOSED PROTOCOL

The proposed protocol uses IP addresses for communication. Every new node in the ad hoc network randomly selects an IP address. But due to birth day paradox problem, there is a chance for address collision. So the protocol performs duplicate address detection. The protocol uses a distributed bloom filter to represent the current set of allocated addresses. This filter is present at every node to reduce the control overhead required to solve address collisions inherent in random assignments [1]. If more than one node selects the same IP address then address collision will occur and it will detected with less overhead.

In probability theory, the birthday problem or birthday paradox[5] concerns the probability that, in a set of n randomly chosen people, some pair of them will have the same birthday. By the pigeonhole principle, the probability reaches 100% when the number of people reaches 367 (since there are 366 possible birthdays, including February 29). However, 99.9% probability is reached with just 70 people and 50% probability with 23 people.

3.1 Distributed Approach

The choice of a distributed approach [1] alleviates the need for instituting a process for the election of a central node that performs the address allocation process. The responsibility for address configuration has to be borne by all nodes that are already part of the network. There must not be a single Dynamic Host Configuration Protocol (DHCP) server since it is impossible to guarantee that the server will always be available. All nodes should collectively perform the functionality of a DHCP server.

3.2 Bloom Filter

The Bloom filter [6] is a compact data structure used on distributed applications. The Bloom filter is composed of an m-bit vector that represents a set $A=a_1,a_{2,\ldots},a_n$ composed of n elements. The elements are inserted into the filter through a set of independent hash functions, whose outputs are uniformly distributed over the bits. First, all the bits of the vector are set to zero. After that, each element is hashed by each of the hash functions, whose output represents a position to be set as 1 on the m-bit vector. To verify if an element belongs to A, we check whether the bits of the vector corresponding to the positions are all set to 1. If at least one bit is set to 0, then A is not on the filter. Otherwise, it is assumed that the element belongs to A. There is, however, a false-positive probability that an element be recognized as being in A. This may happen when the bits at the positions are all set by previously inserted elements.

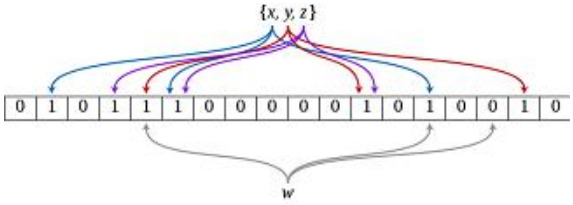

Figure 1 Bloom Filter

The steps involved in the protocol implementation are:

- Creation of traffic and topology
- Implementation of address configuration protocol and bloom filter
- Simulation using nam
- Analysis of trace file and Performance evaluation

4. SIMULATION ENVIRONMENT

We implemented the protocol in the Network Simulator- 2 (NS-2) and evaluated it considering the Two Ray Ground model for radio propagation and the NS-2 IEEE 802.11 model for the Medium Access Control. These models account for creating a scenario similar to a real community network, using parameters of commercial equipment. Simulation parameters are shown in table 1.

Table 1. Simulation Parameters

Parameters	Environment
Number of nodes	10
Maximum node speed	100.0
Radio propagation	Two Ray Ground
Network interface	Wireless physical
Area	1000 x 1000
MAC type	802_11
Link layer type	LL
Antenna model	Omni Antenna
Interface queue	DropTail/PriQueue
Simulation time	20 seconds
Routing protocol	AODV
Recorded parameters	Average end to end delay, total Transmission time ,Throughput

5. ANALYTICAL RESULTS

The probability of address collision is analyzed. A collision occurs when two different joining nodes generate AREQs with the same address. 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. A joining node always sends an AREQ and, when any node that has already advertised the address receives the AREQ, it must check for a collision, regardless of its current state. Assuming there is no malicious behavior in the network, this situation occurs only in the initialization or when nodes join the network at approximately the same time because both nodes could choose the same available address in the filter. Therefore, if two or more nodes choose the same address, then address collision is detected.

The probability that two nodes choose the same address for an AREQ, causing a collision is given by equation 1 , can be derived by considering the birthday paradox, with being the space size of the concatenation of the address with the identifier number, and the number of nodes that are trying to access the network at approximately the same time, which means a set of initiator nodes or a set of joining nodes that search.

The protocol were run a number of times to evaluate the probability of collision and the results shows that for getting one address collision, the protocol has to be simulated at least 5 times.

5.1 Parameters Evaluated

The following metrics were chosen to evaluate the performance of the protocol.

1) Average Throughput: The throughput is directly related to address collisions. Throughput obtained will be zero, if an address collision is detected. Otherwise, a positive value will be obtained.

2) Total transmission time: Total transmission time for the packets is observed to be 20 seconds.

3) Average end to end delay: End to end delay refers to the time taken for a packet to be transmitted across a network from source to destination.

Table 2. Sample Throughput and Delay Values

Simulation No	Average Throughput	Average End to end delay
1	461.059KB	159.180 ms
2	234.053 KB	243.123 ms
3	339.581KB	305.110 ms
4	402.124 KB	151.240 ms

Some of the values obtained for throughput and end to end delay are shown in table 2.

Figure 2 shows the throughput versus time graph of the proposed protocol.

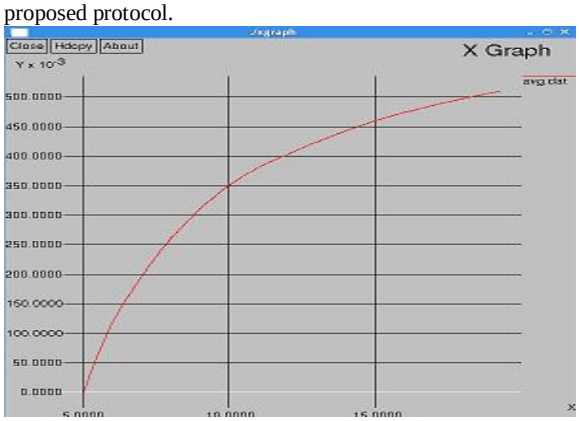

Figure 2 Throughput versus time graph

6. CONCLUSION

Lack of manual management in ad hoc networks means that automatic configuration is highly desirable. Automatic configuration of nodes in wireless ad hoc network will help in reducing administration efforts by users and network administrators. Initial investigation into this area identified the need for achieving high levels of address uniqueness without compromising on latency and communication overhead. Simulation experiments were done in NS2 to test the performance of the protocol and the various parameters were evaluated. Address filters avoids address collisions, reduces the control load, and decrease the address allocation delay.

7. REFERENCES

[1] Natalia Castro Fernandes, Marcelo Duffles Donato Moreira, and Otto Carlos Muniz Bandeira Duarte "An Efficient and Robust Addressing Protocol for Node Autoconfiguration in Ad Hoc Network" ,IEEE/ACM Transactions on Networking, VOL. 21, NO. 3, JUNE 2013

[2] C. E. Perkins, E. M. Royers, and S. R. Das, "IP address autoconfiguration for ad hoc networks," Internet draft, 2000Tavel, P. 2007 Modeling and Simulation Design.

[3] N. H. Vaidya, "Weak duplicate address detection in mobile ad hoc networks," in Proc. 3rd ACM MobiHoc, 2002, pp. 206216

[4] S. Nesargi and R. Prakash, "MANETconf: Configuration of hosts in a mobile ad hoc network,"in Proc. 21st Annu. IEEE INFOCOM, Jun. 2002, vol. 2, pp. 10591068.

[5] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in Proc. IEEE Symp. Security Privacy, May 2005, pp. 4963.

[6] Deke Guo , Jie Wu , Honghui Chen , Ye Yuan , Xueshan Luo "The Dynamic Bloom Filters" In Proc. IEEE infocom citations: 16 – 24 (2006)

[7] M. Fazio, M. Villari, and A. Puliafito, "IP address autoconfiguration in ad hoc networks: Design, implementation and measurements," Comput Netw., vol. 50, no. 7, pp. 898920, 2006