PARIX: A Protocol Adaptive Mechanism for Routing Information Exchange Across Heterogeneous Ad Hoc Routing Domains

Poornananda G R, Tulasi A S
Cisco Systems, Inc.
Bangalore, INDIA
{ poorna, tulasi } @cisco.com

Abstract

With a multitude of routing protocols being developed for Ad Hoc networks, it is a very realistic scenario for a set of ad hoc nodes to congregate and still be unable to communicate with one another simply because two or more subsets of these adhoc elements belong to heterogeneous routing domains each running a different routing protocol. This paper outlines a mechanism for ad hoc network elements forming such heterogeneous routing domains to be able to exchange routing information. Further, the mechanism offers a platform for both reactive and proactive protocols to inter-operate with one another while not losing their identity of being proactive or reactive

1. Introduction

An Ad Hoc network is characterized by a constantly changing topology. An obvious consequence of such a network is that the formation of the network is mostly random and unpredictable with limited mobility predictions. Another important aspect of ad hoc networks is the participant device itself. Ad hoc network elements can range from powerful laptops to low-end mobile devices which have constraints such as limited transmission capacity, limited power etc. Therefore there is a serious need to develop protocols for network level convergence which take into account both the changing topology and the nature of the devices that form the ad hoc network. To address this issue, a multitude of routing protocols have been developed the world over for ad hoc networks which meet some or all of the constraints that an ad hoc network typically imposes. Several legacy wired network routing concepts such as distance vector and link state have found their equivalents in the ad hoc world too.

Given the plethora of such protocols, a very realistic scenario can occur when a number of ad hoc devices congregate to form an ad hoc network but are unable to communicate with one another since each element may have its own version of routing protocol running. It is not difficult to imagine some groups of scientists assembling at a conference where each group arrives with its own routing protocol implemented on their laptops.

Each ad hoc routing protocol may broadly be classified as either proactive or reactive. Each protocol is designed keeping certain advantages in mind and each protocol has inevitably its disadvantages too. For example, reactive protocols are designed to be on-demand which means that the route discovery is initiated when an application desires to start some communication. While this reduces the control traffic needed to learn and maintain routes, it incurs a delay in response for the application. However, such protocols are well suited for devices that have limited processing power and memory. On the other hand, proactive protocols are designed with the goal of maintaining all possible routes at all times thus introducing much more control traffic than the reactive counterpart.

Therefore, in developing a method for such heterogeneous routing domains to interact with one another, it becomes important not to make nodes running one class of protocol change their behaviour and operate resembling another class of protocol. For example, making a node running a reactive protocol operate like a proactive protocol might not be feasible for the simple reason that the node might not have enough power or memory to maintain the kind of state information inherent to a proactive protocol.

2. Design goals

The following are the main goals that were kept in mind while coming up with the operational details of the PARIX mechanism:

1. Maintain routing protocol operational identity

0-7803-8396-6/04/$20.00 © 2004 IEEE

2. Adapt to the route discovery and maintenance facilities provided by the protocol that exists natively on the node.
3. Minimal overhead in terms of control traffic to perform the route exchange

3. Design overview

The behaviour of PARIX depends on the type of the routing protocol that runs on the node. This is important to meet the first design goal. The mechanism is primarily based on a subset of the ad hoc nodes communicating with one another and assumes the presence of PARIX nodes on one or more nodes in each ad hoc domain. A distinct feature of PARIX is that it does not depend on the establishment of adjacency based on a hello protocol. A given PARIX node optimistically sends out its state advertisements and route requests hoping there are other PARIX nodes nearby which may respond and/or learn the information sent out.

The communication depends on the type of the native routing protocol that runs on the device. In case of nodes running a proactive routing protocol, it becomes necessary to update the routing information constantly. This entails periodic use of either a broadcast or a multicast scheme to achieve the above purpose. On a node that runs a reactive protocol, the mechanism follows the on-demand request reply mode of operation.

In case of a proactive node, the mechanism periodically sends and receives updates from other nodes. The IP routing table is modified to reflect the new information that is learnt as a result of these updates. On a reactive node, the mechanism sends a request when a routing table lookup failure occurs. This request is received and processed by a subset of its peers. This subset must be chosen to minimize multiple retransmissions of the request. One solution that can be used in this scenario is the MPR (Multipoint Relay) method proposed by Amir et al. [3]. More such mechanisms which circumvent unnecessary retransmissions have been discussed by Ogier et al in their draft [6].

4. Operational details

As mentioned in the previous section, the mechanism assumes two separate modes of operation based on the type of protocol that runs on the node.

4.1 Originating node is reactive

When the node on which an application sends a packet down the IP layer is a Reactive Node, which means that the routing protocol is a reactive routing protocol, a failure in routing table lookup will trigger the native reactive routing protocol to start its own route discovery process. In order to avoid waiting for the reactive routing protocol to converge, PARIX sends out its own request to other PARIX peers. This request will be subsequently received by another PARIX peer, which is in the MPR subset of the originating node. There are two possibilities here. The receiving PARIX peer is either a Reactive Node or a Proactive Node.

4.1.1 Reactive receiver

In case the receiver is a reactive node, a check is made to see whether the routing protocol running on this node is the same as that running on the originating node. If the two protocols are the same, PARIX retransmits this request. The reason behind this design can be best understood with the help of an example. Consider two or more ad hoc domains as shown below, each running a different protocol. Consider node 1 in domain A, which wants to communicate with node 3 in domain B. A PARIX request generated on node 1 in domain A might not have in its vicinity, a PARIX peer that belongs to domain B. In this case, the request has to traverse multiple hops to reach the other domain. So, in our example the request can reach node 3 in domain B only by going through node 2.

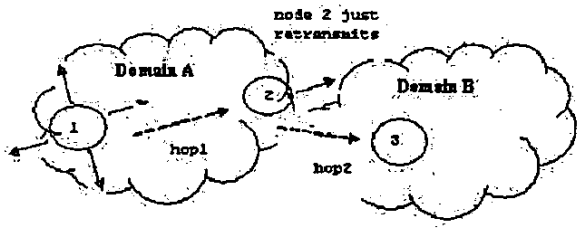

Figure 1: PARIX relay operation

If the two are different reactive routing protocols, then PARIX proceeds by issuing a routing table lookup for the destination IP address specified in the PARIX request. If the destination IP address is found, the route entry is sent back by PARIX in its reply. If the route lookup fails, this will in turn initiate a route discovery process by the current routing protocol [4]. PARIX will now sleep for enough time to allow the route to be learnt. This sleep time should be optimal and be equal to the average time it takes for a reactive routing protocol to converge. After the sleep timer expires, the routing table is checked again to see if the route

has been learnt. If the route has been discovered, PARIX sends this as the reply back to the originator.

4.1.2 Proactive receiver

If the receiving node is a proactive node, then a route lookup is performed for the given destination IP address. If the route is found, PARIX sends this information in the reply.

When PARIX receives a reply, it first examines the routing table to determine whether the native routing protocol has discovered the route in which case PARIX does not overwrite this route in the IP routing table.

4.2 PARIX operation on proactive nodes

On nodes that are proactive, the PARIX operation is a little more complex. While maintaining the proactive mode of route discovery, PARIX has been designed to keep the control traffic to a minimum. An important factor in case of proactive protocols is the size of the state table advertisement (similar to the LSAs of OSPF). Here, PARIX nodes maintain a subset of the routing table and advertise this list to one another. The idea behind this is seen to be a compromise between exchanging the entire topology information present on a given node among its peers and not doing so. This subset is constantly modified from two sources. The first source of modification comes when the native routing protocol adds, deletes or modifies a given route. The other case is when a peer advertises its subset to this node. In this case, the node will compare the list of route entries present in its subset to the ones received in the advertisement. Any difference results in the subset being updated. This change will also be synchronized down to the IP routing table. In the next advertisement that is scheduled out of this node, this new subset is made known to other PARIX peers.

Routing entries in mobile nodes are typically expected to age out quickly. Keeping this in mind, the amount of information that actually needs to be advertised can be minimized by defining the subset as a union of three parts. A third of the subset is a list of the most recently learnt routes. This is quite easy to implement and can be done by looking at the value of the age out timer of each route entry. The other third is a list of the most used routes. This can be done by assigning a counter that gets incremented each time the route lookup for a given destination hits that route entry. The final and third is a list of routes whose IP prefixes is the most dominant set in the IP routing table. The reason for choosing the dominant set is as follows. We can intuitively think of similar IP addresses as forming a cluster. Since this IP address cluster is the most dominant set in the routing table, it is reasoned that information about such clusters of IP addresses will be of interest to other PARIX nodes. The next section describes how this dominating set can be identified from the routing table. Therefore, it can be seen that even though the entire routing table is not advertised, PARIX arrives at a compromise by constantly disseminating that subset of the routing table which is likely to be of maximum interest to its neighbours.

4.2.1 Constructing the dominating set.

The size of the dominating has an upper bound defined to be one third of the subset size. IPv4 addresses have a total of 4 octets and identifying the dominating set involves the following steps.

Step 1 From the routing table pick out a list of the route entries in which the most significant octet of the IP addresses occurs with maximum frequency.

Step 2 Using the list resulting from Step 1; pick out the list of the route entries with the most frequently occurring second most significant octet.

For example, let us assume that there are 5 IP addresses:

25.10.2.0, 74.199.3.0, 9.8.0.0, 74.199.4.0, 74.20.6.0

In the first step, it is found that the octet 74 occurs most frequently. So, the list consists of 74.199.3.0, 74.199.4.0, 74.20.6.0.

In the second step, the most frequently occurring second significant octet is chosen. Thus the final list will consist of 74.199.3.0 and 74.199.4.0.

4.2.2 Reactive receiver with proactive originator

There are three possible methods to address the case when the subset advertisement from a proactive node is received by a node that runs a reactive routing protocol. One approach for a receiving reactive node is to also maintain a subset list and process the incoming list and to periodically transmit the list just like a proactive node. This approach however forces the reactive node to behave like a proactive node which might not be feasible. The other approach is to just ignore the subset advertisement. However, the latter approach has the risk of proactive nodes not being able to communicate at all with reactive nodes. A third intermediate approach is therefore chosen which does not need an immediate response to every request but which also does not ignore the advertisement altogether. Here, the reactive node delays sending the response to the subset advertisement by half the time it takes for the packet to traverse one diameter length of the reactive ad hoc domain. Thus, a reactive node waits for any other nearby node of the same reactive domain to respond to the subset

advertisement. Also, in order to avoid collisions a randomised delta can be added to this delay before the response is sent. Thus, by listening for another native protocol peer node to respond to the advertisement, the reactive node can avoid unnecessary transmissions.

5. Route maintenance

Routes learnt via PARIX are maintained by a simple method as follows. PARIX stores each route learnt via a PARIX peer as a node of a list. An age-out timer is associated with each node of this list. An entry's age-out timer is reset every time this route is used. When the age out timer expires, the route is deleted from the IP routing table as well as from the list.

6. Active and Passive modes

Running PARIX on every node of a network is not desirable due to the unnecessary overhead needed to process the requests and replies. Other than this with multiple retransmissions of PARIX replies, the network performance will also some down. The ideal case would be to activate PARIX only when necessary i.e., PARIX is needed when there is an ad hoc node nearby that belongs to another routing domain. PARIX achieves the above motto by transitioning between two modes - a passive mode and an active mode.

In the passive mode, a PARIX node does not process route requests or replies. Further, the node does not participate in the subset advertisement process. The node however relays route requests or the subset advertisements. All PARIX nodes periodically snoop to see if it receives any control packets of a different routing protocol. The presence of such foreign control packets is an indication of the presence of an alien node – a node that probably runs a different routing protocol. In this case, the PARIX node goes into the Active mode. Another case where PARIX shifts gear from Passive to Active mode is when the node itself is the originator of a communication session.

In the Active mode, PARIX nodes do the complete processing of route requests/replies or subset list advertising as described in the previous sections. However, like in the Passive mode PARIX nodes operating in the Active mode periodically go into the promiscuous mode described above and snoop to confirm the presence of alien nodes.

From the Active mode, PARIX nodes revert back to the Passive mode based on the following condition: While operating in the Active mode, if PARIX does not detect an alien node for three consecutive snoop-tries and if all the routes learnt via PARIX have been aged out, then PARIX reverts back to the Passive mode. The above method ensures that PARIX nodes have the most up-to-date information while reducing the routing overhead

7. Operation example

Figure 2 below illustrates a simple example of the various states that PARIX nodes go through when a communication is initiated from routing protocol domain N to domain F.

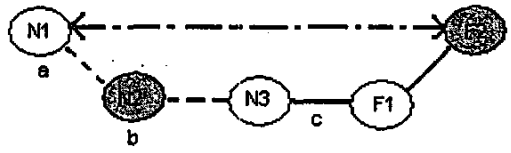

Figure 2: PARIX operation- A simple example

In this case, nodes Ni belong to routing protocol domain N while nodes Fj belong to routing domain F. Nodes in Passive mode are shown to be shaded. In this example, N1 is the originator node. In step (a), N1 undergoes an IP route lookup for destination F2. Because of this route lookup, N1 shifts from Passive mode to Active mode. In step (b) N2 receives the request but just retransmits it without processing since it does not detect Fj nodes. Further, N2 remains in Passive state (shown as shaded). Step (c) shows the scenario where two nodes belonging to different routing domains have detected each others presence and have hence started operating in the Active mode. Here, N3 relays the route request packet to F1 which finds that it has a route to F2. This example describes very simple scenario where a seamless route discovery between different protocol domains happens.

8. PARIX message contents

There are two classes of messages that are needed for the operation of PARIX. First, a format is needed to facilitate the request-reply sequence when the originating or source node is a reactive node. Next, a placeholder in the message is needed for PARIX running on proactive nodes to exchange periodic states of each other.

9. Native protocol discovery

All along, a key assumption has been made that PARIX has the knowledge of which is the native protocol that is running on the ad hoc element. There are two ways to solve this problem. One is to allow the system administrator or the super user of the ad hoc device to explicitly inform PARIX about the native protocol. The alternative is to auto-

learn the native protocol. This method involves storing the various protocol message formats in a database maintained by PARIX. Here, at boot up time or at the time PARIX is brought up, all IP based traffic that originates from the node is snooped on and the IP payload is matched to see to which protocol the packet belongs to, thus determining which the native protocol is. This protocol discovery phase needs to be run only at boot up time or when the PARIX module is first brought up.

10. Implementation and simulation

PARIX is yet to be implemented and will be simulated for generating data once that is over. Protocol design is not an exact science [4] and the only way of arriving at the best possible design is by simulating the implementation. The authors identify that simulation is important to evaluate the protocol and open up any issues with its design while also providing justification for its operational validity.

11. Conclusion and future work

This work outlined a method of route exchange across different ad hoc routing protocol domains. The mechanism described makes use of the native routing protocol that runs on a given node and provides a way to redistribute routing information across heterogeneous routing domains. Further, the mechanism is also designed to operate in modes tailored to the type of the native routing protocol. This is a work in progress and there still are a number of open issues such as choosing the optimal size of the subset list, age out mechanisms for PARIX learnt routes and so on. A simulation of the PARIX mechanism is planned in the future to measure, analyze and improve the protocol characteristics.

12. References

[1] Perkins, C., Belding-Royer, E. and S. Das, *"Ad hoc On-Demand Distance Vector (AODV) Routing"*, RFC 3561, July 2003.

[2] David B. Johnson, Davis A. Maltz, *"The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks"*, IETF Draft, 49 pages, October 1999.

[3] A. Qayyum, L. Viennot, A. Laouiti, *"Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks"*, 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 9, p298 January 07 - 10, 2002.

[4] Elizabeth M. Royer and Charles E. Perkins. *"An Implementation Study of the AODV Routing Protocol"*, Proceedings of the IEEE Wireless Communications and Networking Conference, Chicago, IL, September 2000.

[5] J. Wu and H Li. *"A Dominating-Set-Based Routing Scheme in Ad Hoc Wireless Networks"*, Telecommunications Systems, Vol18, pp. 31-36, 2001.

[6] Richard Ogier., *"Alternative Designs for OSPF Extensions for Mobile Ad Hoc Networks"*, draft-ogier-manet-ospf-extension-00.txt.