

Study on IP Anycast in Ad-Hoc Networks

Husam Sarhan and Chansu Yu

Cleveland State University
Department of Electrical and Computer Engineering
Fenn College of Engineering
2121 Euclid Avenue, Stilwell Hall 332
Cleveland, Ohio 44115-2425

Abstract

Anycast is a communication protocol for obtaining services or sending data to one of a multitude of servers that share one address. This protocol holds great promise for both wired and wireless networks for various reasons. This paper is an attempt to look at what makes anycast the emerging technology of choice for accessing distributed services, especially in wireless ad-hoc networks.

1. Introduction

Mobile Ad-Hoc Networks (MANET) hold the promise of an anytime, anywhere connection to information in areas that have little or no infrastructure. Rescuers, soldiers in the battlefield, or even ordinary folk on a fishing trip could access the information they need by connecting to other wireless units forming a network. Networks could be set up quickly in industrial areas or historical buildings where it would be very difficult or impossible to put together a wired infrastructure.

If MANETs are to deliver on this promise, then users must be able to access services with regularity from anywhere in the network. MANETs are highly dynamic in nature, which means there could be lots of link failures and disconnections. Therefore, services provided to mobile ad-hoc users have to be robust. One way of ensuring the robustness of these services is to have them distributed over a number of different servers. If a user is disconnected from one server, he could obtain the same or similar service from another server. To make things simpler, the user would automatically connect to the closest server. Not only is such an arrangement robust in nature, it allows

users to save power by communicating only with the closest server. Management and configuration of such services is also made simpler by such a technique. This is the premise of what is called ‘*anycast*.’

This paper is an attempt to look at what ‘anycast’ is, how it could be applied to wireless ad-hoc networks, and some of the works that have already been done on the subject. We take a quick look at *IP version 6 (IPv6)* in Section 2, as it adopts anycast as a standard, and because most of the works done on the subject come from an IPv6 standpoint. In Section 3, the concept of anycast is explained as it pertains to wired networks and the Internet. In Section 4, we explore the potential applicability of anycast in wireless ad-hoc networks, and then take a look at some of the works that address the issues inherent in applying anycast to wireless ad-hoc networks.

2. A Quick Look at IPv6

The Internet owes its existence largely to a few protocols that were invented back in the late seventies and early eighties. Chief among those protocols is the TCP/IP protocol stack, which provides an addressing and transport mechanism for delivering packets over the Internet. In particular, the *Internet Protocol (IP)* is the one that allows each node to have a unique IP address by which it can be reached. The current standard today, *IPv4*, has not strayed much from the original specification for Internet addressing. However, fundamental as this protocol may be, it is not without limitations.

One of the biggest problems with the current IPv4 is the limited address space. IPv4 addresses were represented by only 32 bits, which meant that address space was running low as the number of nodes on the Internet grew. *IPv6* [1, 7] attempts to solve this problem by introducing 128 bit addresses. IPv6 retains the unicast and multicast address schemes of IPv4, but eliminates broadcast addresses in favor of a new form of addressing called “anycast” (see section 4). IPv6 addresses are represented by eight 16-bit hexadecimal integers. For example, ‘CDCD:910A:2222:5498:3900:20FG’ is a valid IPv6 address. Note that each of the hexadecimal integers is separated by a colon.

Much like IPv4, the IPv6 unicast addresses are aggregatable into subnets. IPv6 addresses are also similar to those of IPv4 in that the most significant bits represent the

network that the node is attached to, while the least significant bits represent the unique address of the node itself. The 64 most significant bits form the *network address*, while the lower 64 bits use the node's network interface MAC address to calculate a unique '*interface identifier*' that identifies the node on the network. This way, there can be 2^{64} (or approximately 18 billion) unique interfaces on one network. Another difference between IPv4 and IPv6 is that only 1/8 of the IPv6 address space is allocated to aggregatable addresses, while the rest of it is left unassigned or reserved for other purposes [2].

Another problem with IPv4 was its packet headers. If there were any options in the header, each router on the path to a packet's destination would have to strip the header of the packet and examine its contents before repackaging the packet with a new header and sending it back out on its way. IPv6 allows for options to be included in extension headers that are separate from the main packet headers. This way, the intermediate routers would not have to process any of these options, which provides for more efficient packet delivery. Another difference from IPv4 is that fragmentation is not handled by the intermediate nodes, but rather by the source node on an end-to-end basis. However, there is a hop-by-hop extension header which specifies something that must be done to the packet each time it is forwarded. IPv6 packets have 8 main headers, each of which is exactly 40 bits in length. Intermediate nodes are not allowed to perform any processing on these headers. Also, the IP header checksum has been removed from IPv6. These two measures provide for a more streamlined packet delivery process.

Finally, IPv6 has two more improvements over IPv4. The first is the possibility of having flows of packets. A flow is defined as "a sequence of packets sent from a particular source to particular (unicast or multicast) destination for which the source desires special handling by the intervening routers." This is done by having the routers retain flow information that persists from packet to packet within the flow. The second improvement over IPv4 is that IPv6 was designed for security and authentication purposes from the beginning. This is accomplished by using the IPsec standard (introduced in RFC 1825, updated recently in RFC 2401).

The reason we looked at IPv6 in this section is that 'anycast' is now a standard in the IPv6 protocol. While it is still possible to perform anycasting in IPv4 networks, the

addresses needed for that would have to be carved out of the existing IPv4 address space, which is already running low. As shown earlier, IPv6 has clear advantages over IPv4. This being the case, we will look at anycasting from an IPv6 point of view. Refer [1,2] for more information in IPv6.

3. What is Anycast?

Anycast means “*a point-to-point flow of packets between a single client and the ‘nearest’ destination server identified by an anycast address*” [6]. What this means is that a number of servers can be identified by the same ‘anycast address,’ and usually offer the same type of service. A node wishing to obtain a particular service could automatically connect to the nearest one (as determined by the routing system).

Anycast addresses are treated as a single instance of an anycast server. So even though there are multiple servers with the same address, a router will treat an anycast address as a specific host address. Anycast addresses are allocated from the unicast address space, and have a ‘*longest address prefix*’ that specifies the topological region in which the anycast servers reside [2]. If the servers reside in one topological region, then only one routing entry need be advertised on the Internet. However, within this region, each node must be advertised as a separate routing entry. If the servers do not reside in one topological region, then each must be advertised as a separate routing entry on the Internet. This is known as ‘*global anycast.*’ There are two restrictions on IP anycast addresses in IPv6. First, an anycast address cannot be assigned to a particular host, but rather must be assigned to a router interface. The second restriction is that the source address of a packet cannot be an anycast address.

One of the problems with anycast is the case where there is a change of topology. IP is a connectionless protocol, and anycast sends packets on a hop-by-hop basis to the nearest server based on the shortest path. A change in topology could lead to a new shortest path, which could result in packets intended for one anycast server to end up at another server with the same anycast address. This could also potentially result in degraded or lost TCP sessions. One way of handling the routing problem is to have the

routers treat each route to an anycast server as a host route, rather than a routing entry representing multiple hosts.

There are several ways to handle the TCP connection problem. The first is to have the unicast server send back its unique unicast address in its reply to the source node. This has the disadvantage of having to change TCP end points. Another approach is to include an option in IP packets that binds the anycast address with a particular unicast address. All the intermediate routers will have to look at each of the IP options fields and examine its contents, which could degrade performance. The ideal approach is to have some sort of “anycast-to-unicast” resolution mechanism that runs separate from TCP. HTTP could be used to redirect client connections to different hosts. The anycast server, upon establishing a connection with the client and realizing that it was dealing with an anycast address, could issue an HTTP redirect telling the client to connect to its unique unicast address.

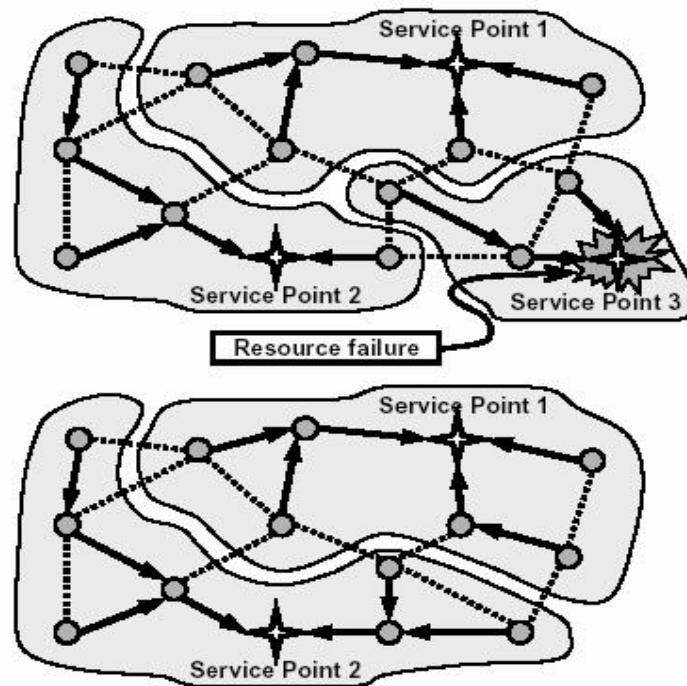


Figure 1: Illustration of dynamic anycast service areas before and after the failure of a service point [5]

Anycast’s advantages in wired networks are great. Anycast could be used to enhance network performance and provide recovery capabilities. For example, the

‘Anycast RP’ protocol introduced by Cisco Systems uses IP anycast to enhance performance in intra-domain multicast networks running *protocol-independent multicast (PIM)*. Several PIM rendezvous point (RP) routers are configured with the same IP anycast address. If an RP router goes away, all sources and receivers automatically forward their multicast traffic to the next closest RP that has the same address. Figure 1 illustrates the potential robustness of a network using the anycast protocol. Anycast also can be used to perform load balancing among several servers. This could be especially beneficial to applications and Internet server farms.

Another advantage of anycast could be in bridging IPv4 and IPv6 networks in order to migrate from IPv4 to IPv6. Initially, IPv6 networks will be configured as islands connected to each other by IPv4 tunnels. This “6to4” scheme uses border routers to encapsulate IPv6 packets in IPv4 packets. Currently, administrators of these border routers must manually configure a default route across the IPv4 network to a 6to4 relay router. The solution to this configuration problem is to have these relay routers advertise themselves on the IPv4 network using anycast addresses.

4. Potential for Anycast in MANETs

As great as the advantages of anycast are in wired networks, they become even more crucial in wireless networks, especially ad-hoc networks. Anycast has potential implications on improving performance and providing easier access to distributed services while simplifying the management and configuration of these services. It also has implications on improving the robustness of ad-hoc networks, especially given their highly dynamic nature.

Although the topic of anycasting in general is new, and especially in mobile ad-hoc networks, there have been several works done that either examine the potential of anycasting in ad-hoc networks or suggest solutions to some fundamental concerns about using anycast in ad-hoc networks. It is from the former point that we start this section.

4.1 An Anycast Routing Protocol for Ad-Hoc Networks

As mentioned earlier in Section 3, there exists a potential problem with TCP connections in an anycast environment. While using HTTP redirects may be feasible in a quasi-static network such as the Internet, it becomes impractical in ad-hoc networks given their highly dynamic nature. One suggestion is to perform anycasting service for routing in ad-hoc networks [5]. In simulations, it was found that, compared to the traditional routing schemes, network bandwidth utilization for control packets increased with the anycast scheme. However, bandwidth utilization for messaging decreased because packets were delivered to their destinations via shorter routes on average. Unicast schemes could simulate anycast by having each node maintain a prioritized list of anycast servers. This is more useful for constructing and maintaining anycast routes than it is for packet delivery. This unicast scheme would always deliver packets to the primary server, regardless of whether there are shorter routes available to other servers that offer the same service. Thus, depending on the network load, anycast's reduction in messaging traffic could outweigh its increase in control traffic.

Route availability in anycast is optimal, and is equivalent to the unicast scheme when it has a route available to the primary server. The problem with the unicast scheme is that it becomes impractical to maintain the priority list when the number of anycast servers becomes large or the set of servers is highly dynamic. It is evident then that anycast is inherently robust while providing simpler management and configuration for ad-hoc networks.

As for packet delivery delay, anycast wins because it delivers packets via shorter routes on average. Figures 2 and 3 depict the mean hop count and the mean packet delay. However, this may be affected by higher level protocols. Connection setup time and packet retransmission time for reliable packet delivery may increase somewhat the packet delivery delays.

There is great potential for applications using anycast technology in wireless ad-hoc networks. Anycast technology could play an important role in locating, gathering, and retrieving information in ad-hoc networks. Anycast makes simple the task of locating information that may be distributed among several higher-level services and/or applications. It makes possible the retrieval of information because it provides a server

address for subsequent retrieval requests. And it also makes possible the gathering and storing of information in a distributed, dynamic fashion. For example, several users could collect and merge data they have gathered independently via a higher-level application by simply forwarding their input to the proper anycast address.

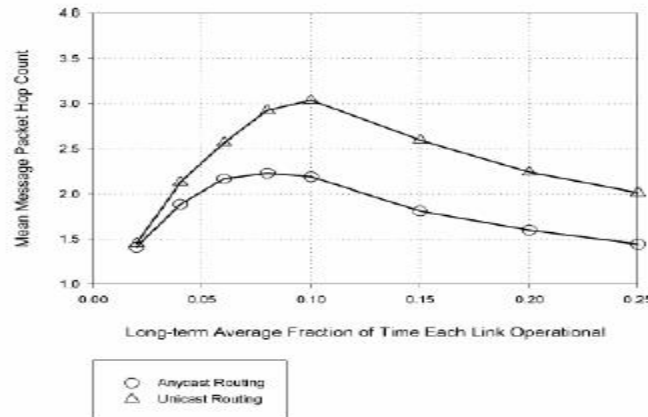


Figure 2: Mean packet hop count as a function of average network connectivity [5]

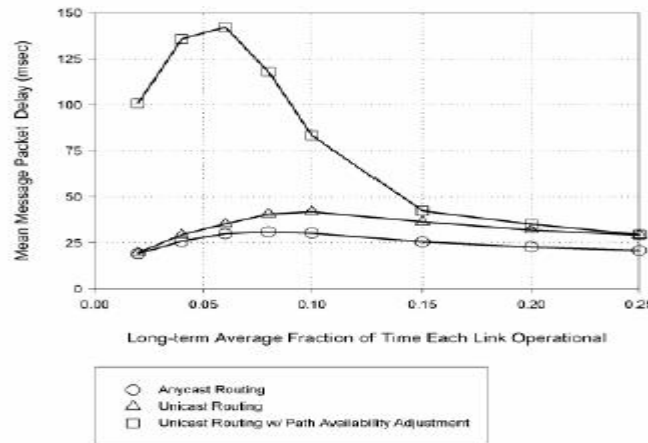


Figure 3: Mean packet delay as a function of average network connectivity [5]

4.2 Another Anycast Routing Protocol for Ad-Hoc Networks

Not much work has been done on the topic of routing protocols for anycast networks in general, much less for anycast networks in wireless ad-hoc networks. However, one protocol has already been suggested [4], and it makes use of the existing *Ad-Hoc On-*

demand Distance Vector (AODV) protocol. This protocol is called *AAODV*, which stands for *Anycast AODV*.

The premise of this protocol is simple. If a source wants a service (route to the destination) from an anycast server, it broadcasts an AODV *route request (RREQ)* packet including the destination anycast address. This packet can take multiple routes to multiple anycast servers. When an anycast server hears this request, it includes its unique unicast address in a *route reply packet (RREP)* that it sends along the reverse route. The source will retain information about all the anycast servers it hears from in its routing table, but will use the one with the smallest hop count. Each routing entry in this table has a timer associated with it. If the timer expires, the route is removed from the routing table. Each *anycast group* consists of a number of anycast servers and multiple anycast groups can coexist with different services, e.g. providing a different routing path to a destination.

Some of the advantages of this protocol are that there is always a route to an anycast server since the source keeps information about all the servers it hears a reply from. Furthermore, the timer it associates with each route in its routing table allows the routing information to remain fresh. The timer allows the node to continuously find the shortest route or even new routes to an anycast server.

Three metrics were used in the evaluation of this protocol: packet delivery ratio, routing overhead, and route optimality. It was found by simulation that packet delivery ratio is over 95% in most cases, while path optimality is above 80%. Routing overhead increases as mobility increases because maintaining the routing tables becomes more frequent. It was found however, that if the number of anycast groups or the rate of mobility increases, the performance of the protocol degrades. Also, the fewer the number of servers in an anycast group the worse the performance. The number of source nodes in a network didn't seem to affect performance much, according to the authors. Figures 4, 5 and 6 illustrate some of the simulation results for this protocol. In the figures, series 50-1-2-3 corresponding to network with 50 nodes, 1 source and two anycast groups with 2 and 3 members, 50-1-5 corresponding to network with 50 nodes, 1 source and one anycast group with 5 members.

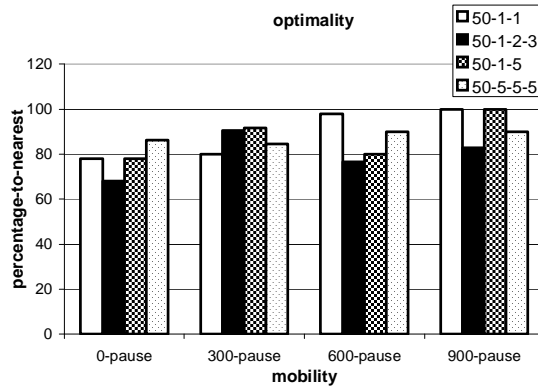


Figure 4: Path optimality for 50 nodes as a function of pause time [4]

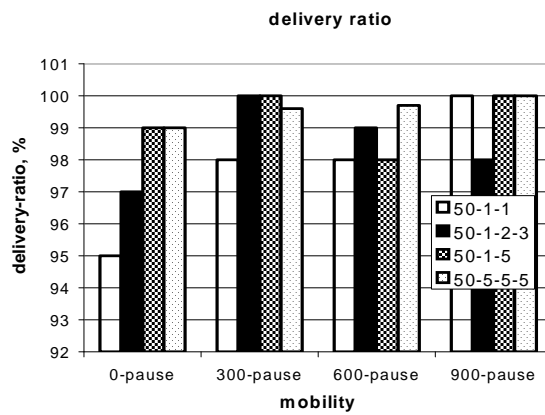


Figure 5: Packet delivery ratio for 50 nodes as a function of pause time [4]

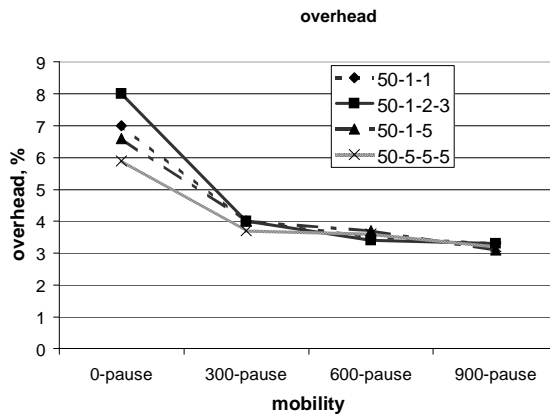


Figure 6: Routing overhead for 50 nodes as a function of pause time [4]

There are a few areas where the protocol could benefit from improvement. As mentioned above, a very high rate of mobility affects the protocol's performance. But a really low rate of mobility would have a much worse effect on its performance than a

very high rate of mobility. This is because of the timing out of routes in the routing table, which would require periodic broadcasts of RREQ packets that would consume unnecessary bandwidth. One way of solving this is to have some sort of exponential back off technique. If the newly discovered shortest route is the same as the one discovered in the previous round, the timer is doubled until it reaches a predetermined maximum. If the route changes after that, then the timer is halved until it reaches a predetermined minimum.

Some additional improvements can be made to the protocol, especially in handling link breakages. For example, bandwidth consumption could be reduced by not using local link breakage repair mechanisms, having the source initiate another anycast server discovery process if it detects link failures. Another technique the authors consider is to have an event-driven model, whereby the destination somehow notifies the source of link breakages or hop count changes.

4.3 Another Type of Anycast

Anycast also has potential in wireless sensor networks, where it could be used for distributed control or data gathering applications. For this purpose, a protocol called the *Sink-based Ad-Hoc Routing Protocol (SARP) for Wireless Sensor Networks* was proposed [3]. The goals of this protocol are to reduce power and bandwidth consumption while reducing packet delivery delays. The idea behind this protocol is that data is delivered to the closest sink, so there is no precise destination for a packet. This contrasts with the anycast schemes that have been discussed earlier, in which a source node requests services from an anycast server. However, this protocol is still considered anycast in the sense that it is a 'one-to-any' protocol. The advantage of such a protocol in such applications is a local adaptation behavior, quicker reaction to stimuli, and reduced bandwidth and power consumption. Thus, such a network could be set up quickly where there is no networking infrastructure or putting that infrastructure down is difficult.

There are several technical issues that the authors discuss. The first is how to route a packet along an optimal path in order to reduce network bandwidth and node power consumption. The second issue is how to achieve a high packet delivery ratio

given frequent node mobility. Finally, the third issue is how to maintain the routes with a minimum of routing overhead. In order to come up with a protocol that performs all three of these functions, the authors have made some assumptions. The first assumption is that links are symmetrical, which is a reasonable assumption. If node B can hear node A, then node A surely can hear node B. This assumption allows link reversals. The second assumption is that nodes must not move too fast in comparison with their transmission range. The third assumption is that nodes will operate in promiscuous mode, and thus cache new routes upon overhearing route reply packets. The last assumption is that reliable delivery issues will be handled at the application layer.

The central idea behind this protocol is that a node delivers packets to nodes with a lower hop count than itself until it reaches a sink. To that end, the protocol comprises three major mechanisms: *route establishment*, *route recovery*, and *hop count update*. In route establishment, a node interested in receiving packets announces its interest on the network by flooding an *interest packet*. By using such a packet, nodes in the network can also learn how many hops they are from the interested node. In case there are multiple sinks, only information about the sink with the lowest hop count is maintained. Route recovery can be done in two ways: either the link breakage detection mechanism of the IEEE 802.11 protocol or having the nodes operating in promiscuous mode and overhearing new routes. Forwarding nodes will set up a timer that is slightly longer than the packet round trip time. If they don't hear any action from a neighboring node before this timer expires, they assume route failure and start the route recovery process. Route recovery is only performed at the node that detects a link breakage, which conserves power and bandwidth and solves potential route failures for routes involving this node. In order to maintain the correctness of the hop count, a node checks its hop count by overhearing route reply packets. The node will then send a hop count update packet to its neighbors, which they will not rebroadcast. These neighbors will update their hop count and send out a hop count update packet of their own. This process will continue until there are no downstream nodes with hop counts greater than or equal to those of their upstream neighbors.

Simulations have shown that, in most cases, the packet delivery ratio was over 86%. For 10 to 20 sources, the packet delivery ratio was around 96%, although the

protocol struggled with 30 or more nodes. Increasing the number of sinks solved this problem, and increasing the number of nodes in the network in general seems to increase the packet delivery ratio. As for routing overhead, a higher number of sources mean a higher number of routing packets. Most of the routing overhead comes from one-hop acknowledgements, which are local in nature. However, increasing the number of sinks was found to reduce the routing overhead significantly. Finally, it was found that the average hop count of optimal paths differed from the average hop count of the actual paths taken by 0.1 hops, and increasing the number of sinks increases path optimality. In general, the greater the number of sinks, the better this protocol performs. This is because of greater load distribution among the sinks in the network. Figures 7, 8 and 9 illustrate the simulation results for this protocol.

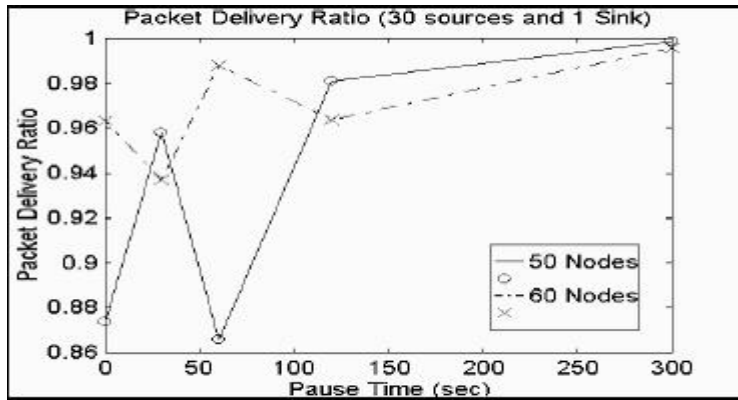


Figure 7: Packet delivery ratio as a function of pause time [3]

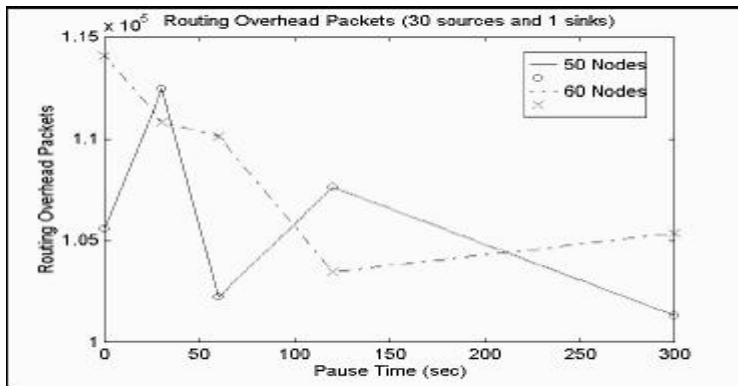


Figure 8: Routing overhead as a function of pause time [3]

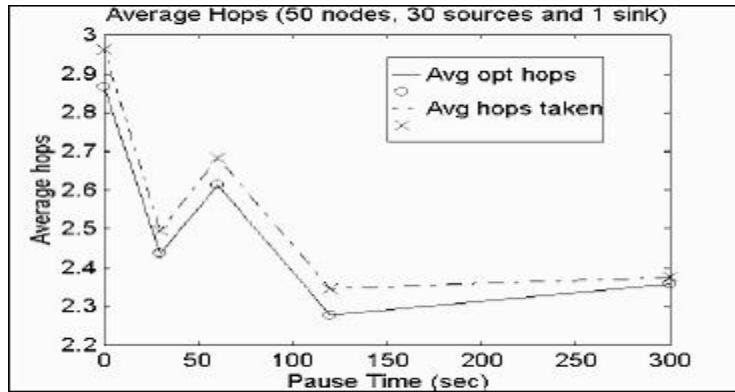


Figure 9: Difference between average optimal hops and average hops taken as a function of pause time [3]

5. Conclusion

Anycast is a point-to-any point communication method that holds a lot of promise for both wired and wireless networks. Its applications in wired networks are varied. Load balancing among servers is one of its key applications in wired networks. Ensuring service robustness is another key application. Also, bridging IPv4 and IPv6 networks is another application of no less importance than the previous two. In addition, it makes it simpler to configure and manage these distributed services. However, TCP performance may be degraded in an anycast scheme. There are, however, ways of dealing with this problem. Most wireless environments use UDP as their transport protocol of choice, which is a connectionless protocol. Thus the problem with TCP connections will be eliminated altogether.

In a wireless environment, anycast allows users access to distributed services much the same way as in a wired network, and is especially useful for ad-hoc networks. Being highly dynamic in nature, ad-hoc networks need services that could endure frequent disconnections and a high mobility rate, and anycast provides the capability for establishing those services. Connecting to the closest server also conserves node power, and furthermore reduces the routing overhead involved. It is also useful for distributed collection, location and retrieval of data. In addition, anycast can be applied to wireless sensor networks in distributed control applications.

Not many works have been done on anycast protocols for wireless ad-hoc networks, but there are a few such pioneering works. One of them extends the AODV

routing protocol for wireless ad-hoc networks to handle anycast traffic. This sort of protocol would make anycast services more accessible in ad-hoc networks because it is an extension of an already established and relatively efficient protocol. There are still improvements to be made to this and possibly other protocols, but anycast certainly has a bright future.

6. References

[1] Deering, S. and Hinden, R.; *Internet Protocol Version 6*, IETF RFC 2460; available at <http://www.ietf.org/rfc/rfc2460.txt>

[2] Deering, S. and Hinden, R.; *IP Version 6 Addressing Architecture*, IETF RFC 2373; available at <http://www.ietf.org/rfc/rfc2373.txt>

[3] De Lucia, Dante and Intanagonwiwat, Chalermek; *The Sink-based Anycast Routing Protocol for Wireless Sensor Networks*; available at <http://citeseer.nj.nec.com/cache/papers/cs/8765/ftp:zSzzSzftp.usc.eduzSzpubzSzcsinfozSztech-reportszSzpaperszSz99-698.pdf/the-sink-based-anycast.pdf>

[4] Huo, Jinye with Liu, Fang, and Swaminathan, Subramanian; *An Anycast Routing Protocol for Ad-Hoc Networks*; available at http://www.cs.ucsb.edu/~ebelding/courses/290I/f01/final_reports/subramanian.doc

[5] Macker, Joseph P. and Park, Vincent D.; *Anycast Routing for Mobile Networking*; available at http://www.argreenhouse.com/society/TacCom/papers99/01_1.pdf

[6] Metz, Chris; *IP Anycast: Point-to-(Any) Point Communication*; IEEE Computer Society Internet Computing magazine, available at <http://www.computer.org/internet/ic2002/w2094abs.htm>

[7] Loshin, Pete; *TCP/IP Clearly Explained*, pg. 247-268; Morgan Kauffman Press, 1999