Achieving Anycast in DTNs by Enhancing Existing Unicast Protocols [*]

Samuel C. Nelson and Robin Kravets
Department of Computer Science
University of Illinois at Urbana–Champaign
Urbana, IL, USA
snelso20@cs.uiuc.edu, rhk@cs.uiuc.edu

ABSTRACT

Many DTN environments, such as emergency response networks and pocket-switched networks, are based on human mobility and communication patterns, which naturally lead to groups. In these scenarios, group-based communication is central, and hence a natural and useful routing paradigm is anycast, where a node attempts to communicate with at least one member of a particular group. Unfortunately, most existing anycast solutions assume connectivity, and the few specifically for DTNs are single-copy in nature and have only been evaluated in highly limited mobility models. In this paper, we propose a protocol-independent method of enhancing a large number of existing DTN unicast protocols, giving them the ability to perform anycast communication. This method requires no change to the unicast protocols themselves and instead changes their world view by adding a thin layer beneath the routing layer. Through a thorough set of simulations, we also evaluate how different parameters and network conditions affect the performance of these newly transformed anycast protocols.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Routing Protocols

General Terms

Algorithms, Design, Performance

Keywords

Anycast, DTN, Routing

1. INTRODUCTION

The ubiquitous availability of wireless communication has pushed researchers from looking at relatively connected ad hoc networks to frequently disconnected delay- and disruption-tolerant networks (DTNs). DTNs are key to supporting emergency response, social networking and community networks [5]. The key interesting feature is that the communication supported in these networks is based on group-based human interaction [5, 15]. Such groups can be geographic in nature, such as all the people on the same bus; social in nature, such as friends using mobile devices; or role-based, such as firefighters or police officers [12]. While user-based communication is naturally supported by network-level unicast communication, this group-based communication is more naturally supported by *anycast*, where communication with at least one member of a particular group is considered a success.

Two examples help illustrate the benefits of anycast communication in DTNs. First, in emergency response networks composed of groups such as police officers, ambulances, and civilians, group communication clearly trumps individual communication. A civilian is more likely to request the help of *any* police officer rather than a particular one. Similarly, police officers are more likely to request any ambulance, as opposed to a specific one. Second, in community DTN networks, which may be a composition of pedestrian social networks, vehicular networks, and local bus networks, group communication can also be very useful. As buses become equipped with internet-able gateways, individual cars on the road would have incentive to contact any bus, instead of a specific one, for its gateway capability. Furthermore, pedestrians may be more interested in using the network to call for any cab, as opposed to a specific one. Interestingly, anycast can also be useful for enhancing unicast in DTNs. Essentially, smarter unicast routing protocols can be designed to contact nodes geographically affiliated with a target destination node as a first step towards contacting the target node itself [8].

The goal of anycast routing is to reach at least one node (the specific one does not matter) in a particular group. In connected environments, basic anycast routing techniques are relatively straightforward, since messages can be unicast to a particular node in the group that has the lowest cost (i.e., quickest response) [3, 14]. This technique, however, does not work in disconnected environments, since it is extremely difficult to predict which of the nodes of the group would even get the message, let alone be able to respond the fastest. In such a disconnected and unpredictable environment, anycast protocols must instead be smarter and attempt to truly reach *any* node in the group. While existing routing techniques for DTNs seem to lend themselves well to

---

[*] This research was sponsored in part by Boeing and National Science Foundation Grant 0081308.

CHANTS'10, September 24, 2010, Chicago, Illinois, USA.

supporting anycast routing, current approaches to anycast in DTNs are very limited in scope, focusing on single-copy routing and/or targeted for highly constrained mobility patterns [7, 6, 8].

Current DTN routing protocols [11, 10, 4, 2, 17] are built on top of two key mechanisms: direct delivery, to support one-hop communication, and utility-based forwarding, to help guide messages to their destinations. Although both of these mechanisms can be used to support anycast routing, current unicast protocols are designed to route to a specific node as a destination and not to a group. The goal of our research presented in this paper is to investigate the use of these mechanisms in an anycast setting and how they need to be adapted to support groups as destinations. Additionally, we take this one step further and present a protocol-independent anycast layer that exposes group information in a meaningful way to unicast routing protocols, enabling these protocols to run unmodified to support anycast routing. If existing unicast protocols can be adapted to work in an anycast scenario, users could take advantage of a wide array of existing protocols that have been fine-tuned and thoroughly evaluated in many environments. These newly enhanced protocols could also be evaluated under a wide range of parameters and network conditions.

The main contributions of our research are three-fold. First, we explore the mechanisms used in current DTN routing protocols and show how these mechanisms can be adapted to allow for anycast communication. Second, using these modified mechanisms, we present a protocol-independent anycast layer that allows current unicast protocols to run unmodified in an anycast mode. Third, we explore how the resulting anycast protocols perform in various environments, and specifically what features of an environment should guide a user in which anycast protocol to choose. We show that the choice is actually different in the anycast case, as opposed to the unicast case, since there are more factors to take into account. In particular, we show that important factors when choosing an anycast protocol include group size, resource constraints and mobility levels, and the presence of an acknowledgement scheme. Interestingly, the presence of a back channel (free and quick intra-group communication) does not have a major impact on the selected metrics.

The rest of this paper is as follows. Section 2 describes anycast in a DTN environment, and shows limitations of related work. Section 3 shows how to enable anycast by first identifying key routing mechanisms that are found in unicast protocols, and then presenting an overarching way of adapting these mechanisms for anycast use. Using this, we present a protocol-independent way of transforming many current unicast protocols into anycast protocols. We show that a thin layer can be added beneath the routing layer to allow *unmodified* unicast routing protocols to run in an anycast mode. In addition, we present a new DTN routing architecture that allows for both unicast and anycast routing. Section 5 presents a thorough evaluation of many transformed anycast protocols, indicating which factors are important in selecting an anycast protocol. Finally, Section 6 concludes and discusses future directions.

2. ANYCAST IN DTNS

In group-based DTN scenarios, anycast can be used as the core routing paradigm. While there exists anycast solutions for connected environments [3, 14], these solutions all rely on stable end-to-end connectivity. In other words, they work under the assumption that the network allows for select group members to be reliably contacted and cannot be easily adapted to the disconnected and heavily partitioned environments found in DTNs. The challenge in disconnected and unpredictable environments is that anycast solutions cannot simply pick the "best" group member according to some metric and then use unicast techniques to reach it. Instead, they must take a group-based view where groups are the destination, not individual nodes.

The key to achieving anycast in DTNs lies in exposing knowledge of the groups in the network to the routing protocol, and having the routing protocol directly act on that knowledge. Similar to unicast protocols, anycast protocols can use single- or multi-copy techniques. In single-copy approaches, messages are either held until the destination is met (e.g., direct delivery) or forwarded through intermediate nodes via a utility metric. While single-copy techniques work in some environments, they are unreliable in unpredictable DTN environments, since even the best guesses at which node to forward to are often wrong. Unfortunately, there has been little work on anycast in DTNs, all of which focuses on single-copy routing. One attempt at anycast routing in DTNs explores the problem by evaluating different routing metrics for selecting forwarding nodes [7]. However, this approach only analyzes single-copy routing. Furthermore, nodes are all stationary, with the exception of a few mobile nodes that act as message carriers, presenting a very constrained environment for evaluation. A second anycast technique, also using single-copy routing, attempts to utilize genetic algorithms to explore route decisions [6]. This work, however, assumes all mobility, including future mobility, is deterministic and known ahead of time, which is not a good assumption in most DTNs.

One DTN unicast protocol that incorporates elements of grouping is BubbleRAP [8]. This is a social-based forwarding method that first attempts to send messages to the destination's local community, which then can more effectively send the message to the actual destination. Nodes carry around a global ranking, determining how "central" they are to the network, and a local ranking, determining how central they are to their local community, which are used as hints to reach the destination node. However, BubbleRAP is designed for improving unicast, not anycast, and is inherently a single-copy technique and so is limited by the same problems as other single-copy approaches for anycast in DTNs.

It is worth noting that the related concept of multicast in DTNs, where the goal is to deliver a copy of the message to *every* member of the destination group, has briefly been considered. One simulation study, where no new protocols were proposed, considered existing multi-copy unicast routing protocols in a multicast context [1]. The simulation results of existing protocols were interesting. However, these results do not apply to anycast scenarios since the end goals are very different. Most specifically, a primary result was to include a considerable amount of redundancy to reach all group members. This would not hold true in anycast scenarios, where only *one* member need be reached.

Similar to unicast protocols, anycast in DTNs can benefit from managed replication. Unfortunately, it is not possible to directly use the multitude of currently available multi-copy DTN unicast protocols for anycast, since they do not have a group-based view of the world and operate on individual nodes. However, given our goal of deploying unmodified unicast protocols on top of an anycast layer, it is useful to discuss a few of the most relevant protocols. Current multi-copy unicast protocols can be characterized into two main groups, flooding-based protocols and quota-based protocols [11]. Flooding-based protocols, such as epidemic [18], Prophet [10], MaxProp [4], and RAPID [2], do not attempt to put a hard limit on the number of times a message can replicate, and instead focus on smart buffer management and transmission ordering techniques to handle the potentially large number of replicas in the network. These protocols are more appropriate for environments not heavily constrained by limited resources. Quota-based protocols (e.g., Spray and Wait [16], Spray and Focus [17], and EBR [11]), on the other hand, set a hard limit on the number of times a message is allowed to replicate. Limited replication is guaranteed by attaching a quota to every message that indicates the number of replicas the message can split into in the future. The quota is split during each replication and a new quota is carried around with each replica, ensuring the total number of replicas of a particular message never exceeds the original quota for the message. These protocols are more suitable for resource-constrained environments.

While there unfortunately is a lack of effective DTN anycast protocols, it is important to note that there is a wide range of unicast protocols, each suited for different types of environments. Given the diversity of both DTN environments and DTN unicast routing protocols, it would be advantageous to be able to utilize these protocols for anycast routing. We next discuss which of the routing mechanisms commonly found in unicast protocols can be used for anycast and how they need to be modified to support groups as destinations.

3. ENABLING ANYCAST

Anycast, like other DTN routing protocols, needs mechanisms to guide replication, forwarding, and buffer management decisions. In this section, we show how common mechanisms found in unicast routing protocols can be adapted for anycast use. In particular, we show that a thin protocol-independent anycast layer sitting directly below the routing layer can allow a wide-range of unicast protocols to run unmodified in anycast mode. In addition, we present a new DTN routing architecture that accounts for both unicast and anycast routing.

3.1 Group Management

One of the main requirements for anycast communication is access to information about group membership. Essentially, there must be a means of informing the routing protocol which groups the current contact belongs to. Such a *group management component* may store a nodeID-to-group table in memory, and update that table as it receives new group information. A simple approach to group management lets each node carry its own group information throughout the network [8]. While sufficient for the discussions in this paper, such approaches are inherently susceptible to malicious attacks on the group membership lists. Although out of the scope of this paper, we are actively studying the issue of reliable and secure group management [13].

3.2 Routing Mechanisms

Most current DTN routing protocols perform two steps during a given contact opportunity: direct delivery, which supports one-hop delivery of messages, and utility-based forwarding, which guides messages and replicas towards their destination. These two steps can be enhanced to support a group-based view, instead of a node-based view, enabling anycast routing. Current protocols that follow this two-step process include Direct Delivery (with a non-existing utility step), Prophet, MaxProp, RAPID, and Spray and Focus (a follow-up protocol to Spray and Wait).

Direct delivery (referred to as *DD*) supports one-hop delivery of messages, where if the node has a message destined for the contact, that message is immediately transmitted to the contact. DD works by checking every message's destination ID against the contact node ID, as provided by the network layer. If the destination ID matches the contact node's ID, that message is immediately forwarded to the contact. To support anycast, DD must instead check every message's destination ID (which, in the case of anycast is a group ID) against groups that the contact is a member of. To support this, anycast routing protocols must obtain both a node ID from the network layer and a corresponding group ID from the group management component. If the destination ID matches any of the group IDs, that message is immediately forwarded to the contact.

After all messages destined for the contact are delivered, the protocol switches to the utility step (referred to as *Utility*). Based on the contact, a utility function is computed or looked up, and used to decide which, if any, of the stored messages to replicate, which order to send messages, and which order to drop messages. For unicast protocols, these utilities are *node-based*; every node the protocol knows about has a utility attached to it. An example utility is the probability of meeting a particular node. Utility values are updated either periodically or when contacts occur. Similar to DD, Utility can be adapted to support anycast by using group-based utility values where all routing policies work on groups instead of nodes. In other words, each node stores utilities for all *groups*, not individual nodes, that they are aware of. This, in essence, transforms groups into virtual nodes from the node's perspective. When a contact occurs, the node updates the utility for the contact's group(s) instead of the contact's actual node ID. This enables the routing protocols to capture mobility characteristics (such as meeting frequency) of groups instead of individual nodes.

3.3 Protocol Independent Anycast Layer

In the previous subsection, we have shown that the identification and modification of two commonly used routing techniques in DTNs, namely DD and Utility, can allow most current unicast protocols to operate in an anycast mode. Essentially, any unicast protocol that follows the two-step process can easily be adapted to support anycast. When the DD and Utility steps for a unicast protocol are transformed into their anycast counterparts (DD-A and Utility-A), the protocol's view of the world turns from node-based to group-based. While it is useful to know how to perform these transformations, it would also be beneficial to support anycast without modifying the unicast protocols at all. Since both DD-A and Utility-A work on group IDs instead of nodes IDs, it is possible to simply present a *group view* of the network to a unicast routing protocol, and have that

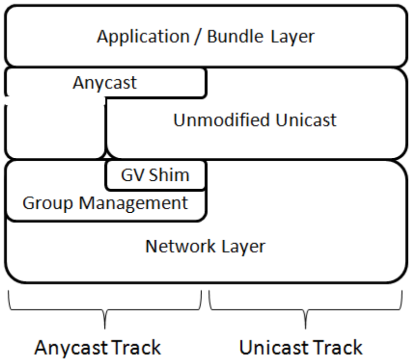

Figure 1: Architecture

protocol work as if it were an anycast protocol. The *group view layer* (GV layer) takes both the node ID from the network layer and the corresponding group ID from the group management component, and passes the group ID to the protocol *in the node ID field.* This presents a view to the routing protocol where every group is actually a virtual node, and every time a member of that group is encountered, it is like that single virtual node was encountered. This view combined with DD and Utility is equivalent to DD-A and Utility-A, enabling the unicast protocols to run in anycast mode unchanged.

3.4 Building a New DTN Routing Architecture

To properly integrate our anycast layer, we now describe a network architecture for DTNs that incorporates both unicast and anycast routing. Routing in DTNs has traditionally been unicast in nature, and therefore the network layer simply passed raw contact information to the unicast protocol, which then appropriately routes application data. However, anycast in DTNs require a slightly more complex architecture, since group IDs must be relayed to the routing protocol in addition to simple contact information.

Figure 1 illustrates a DTN network architecture, focusing on routing, that incorporates both anycast and unicast capabilities. Note that there are two primary tracks that lead up and down the stack: the anycast track and the unicast track. The unicast track is identical to before, the network and application layer can directly interact with the unicast routing protocol. The anycast track includes the group management component directly above the network layer. This allows both the nodes IDs of the current contacts, as well as their respective groups, to be passed to the anycast routing protocol. More specifically, the network layer passes node ID information about each of the nodes currently in communication range to the group management component. This group management component then looks up the corresponding group ID for each node ID, and attaches that information. This pairing is then sent either directly to the anycast routing protocol (the left side of the anycast track), or to the GV layer (the right side of the anycast track). The GV Layer, presenting a group-centric view of the world, indicates to the routing protocol which groups it is in contact with by substituting group IDs for the node ID fields. This gives the illusion that an entire group is actually a single (virtual) node, and allows many unicast protocols to run unmodified.

4. EVALUATION

The goal of our evaluation is to explore the behavior of the anycast adapted versions of a few representative unicast DTN routing protocols in different environments. These evaluations can then help determine which protocol should be use in which DTN environment. In general, unicast DTN protocols are affected by characteristics such as resource constraints, mobility levels and acknowledgements. However, anycast scenarios are also affected by other characteristics such as group size and group back channel availability (i.e., whether group nodes can communicate via an out-of-band back channel). We evaluate how both the traditional unicast characteristics as well as new anycast characteristics effect the transformed anycast protocols.

Our results show that the most important network characteristics for anycast protocols are group size, resource constraints, mobility levels, and the presence of an acknowledgement scheme. Interestingly, the presence of a back channel does not have a major impact on the selected metrics. Where acknowledgements can flush out already delivered messages, a back channel simply allows instantaneous intra-group communication, and can be used as a means to increase the spread of acknowledgement messages. Essentially, once one member of the group obtains a message, it can then use the back channel to instruct all of its other group members to flood out an acknowledgement for the message just received.

4.1 Anycast Protocols

In our evaluation, we include representatives of different classes of single- and multi-copy DTN routing, including flooding- and quota-based protocols, each of which was implemented in the ONE simulator [9]. Using the protocol-independent anycast layer, the flooding-based Prophet [10] becomes Prophet-A and the quota-based Spray and Focus [17] becomes Spray and Focus-A, where the focus function is time since last meeting the destination (i.e., the group). Note that, for Spray and Focus, the initial quota was set to 11, which allows a good tradeoff of resource usage and performance. In addition, we include a direct delivery anycast protocol (DD-A), as well as a pure epidemic protocol (Epidemic-A). Finally, we included an optimized epidemic approach where bandwidth is unlimited and message sizes are negligible (EpidemicOracle-A). This gives an upper bound on what anycast protocols can hope to achieve. Note that Epidemic-A by itself does not give optimal performance since there is no guiding factor in which messages to transmit first. In many cases, Epidemic-A may choose the “wrong message”, which in turn does not allow the “right message” to be sent before the contact is broken. In addition to the anycast layer, we implemented two additional features that can be turned on and off: flooded acknowledgements and group back channel connectivity. These features allow us to explore in-depth the network characteristics that affect anycast performance.

Note that there is no additional overhead, outside of what is incurred by the group management component, in these enhanced protocols. In fact, many of the protocols store *less* data, since the number of utility values is decreased in the enhanced version.

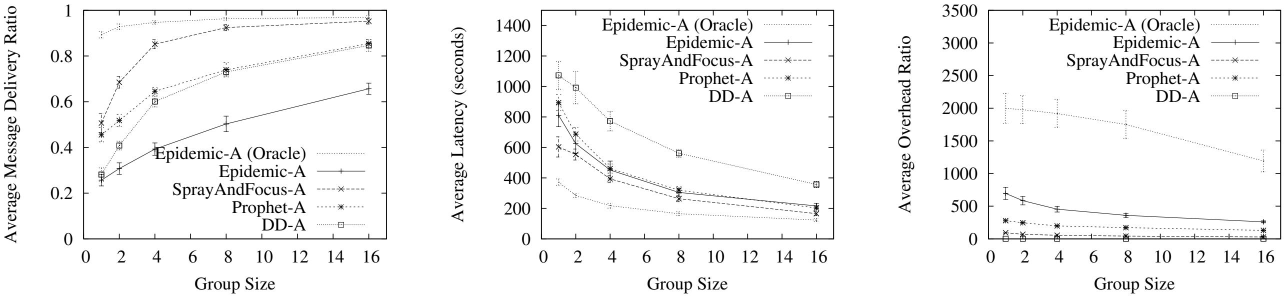

Figure 2: RC, no acks, no back channel (a) MDR, (b) Delay, (c) Overhead

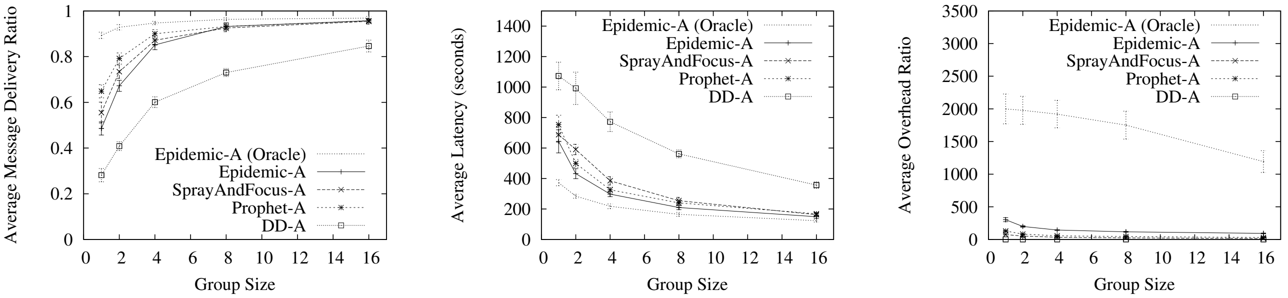

Figure 3: RC, acks, no back channel (a) MDR, (b) Delay, (c) Overhead

4.2 Metrics and Simulation Environment

The first, and primary, metric used for the evaluation is *message delivery ratio* (MDR), which is the ratio of total messages sent divided by total messages received. Due to the intermittently connected environment, not all messages sent will be delivered, and hence this metric gives important information about a routing protocol's effectiveness. The second metric is *average delay of delivered messages*, which is the average end-to-end delay over all delivered messages. This metric indicates how quickly routing protocols can deliver messages, which is important since many messages lose relevance after being delayed for a long period of time. This metric is more meaningful if the MDR's of all protocols being compared are similar, since only delivered messages are included. The third and final metric is *average overhead ratio*, which is the total number of transmissions divided by the total number of received messages. Intuitively, this ratio indicates the average number of transmissions required for each message delivered, giving an indication of resource use. The lower the overhead ratio, the less strain there is on network resources (battery life, network bandwidth, etc.). This is an important metric to indicate the "resource-friendliness" of the protocols.

All simulations were run in the ONE [9] simulator using the built-in community mobility model. This model simulates the movement of pedestrians, cars, and buses in the city of Helsinki, Finland. In all simulations, there are a total of 126 nodes: 80 pedestrians traveling between 0.5 and 1.5 meters per second, 40 cars traveling either between 2.7 and 13.9 meters per second (around 10 to 50 km per hour) in the normal case or between 2.7 and 3 meters per second in the slow case, and 6 buses traveling between 7 and 10 meters per second. This allows for an appropriate density to evaluate DTN protocols. All nodes can transmit at a distance of 100m and speed of 2Mbps, except two of the buses, equipped with high-speed interfaces, which can transmit at a distance of 1000m and a speed of 10Mbps. The nodes choose a location to move to with weighted probabilities (with real Helsinki hotspots being given higher probability), wait for a random amount of time between 0 and 120 seconds, and then repeat. All simulations are run for 4000 seconds, the world size is 4.5km x 3.4km, and node buffer sizes are 5MB. All data points are an average of 10 runs with a surrounding 95% confidence interval.

Nodes 0 to 39 are pedestrians, 40 to 79 are cars, 80 to 119 are pedestrians, and 120 to 125 are buses. In these scenarios, groups are simply chosen sequentially, and are disjoint. For instance, if the group size is 8, then nodes 0 to 7 are in group 0, 8 to 15 are in group 1, etc. This helps keep similar types of nodes in the same group (for example, pedestrians that have a high probability of visiting a particular hotspot), while at the same time allowing for a straightforward way of experimenting with different group sizes.

4.3 Performance Evaluation

The goal of this evaluation is to determine how different factors affect the transformed anycast protocols. The simulations are divided into two groups, a resource-constrained (RC) environment and a non-resource-constrained (Non-RC) environment. Group size are evaluated at 1 (which is equivalent to unicast), 2, 4, 8, and 16. Evaluation across this range indicates how anycast protocols perform when moving away from unicast and towards larger group sizes. For each set of simulations, the availability of acknowledgements and the availability of a back channel to help spread acknowledgements faster are both considered. There are three configurations: (1) no ACKs and no back channel, (2) ACKs and no back channel, and (3) ACKs and a back channel.

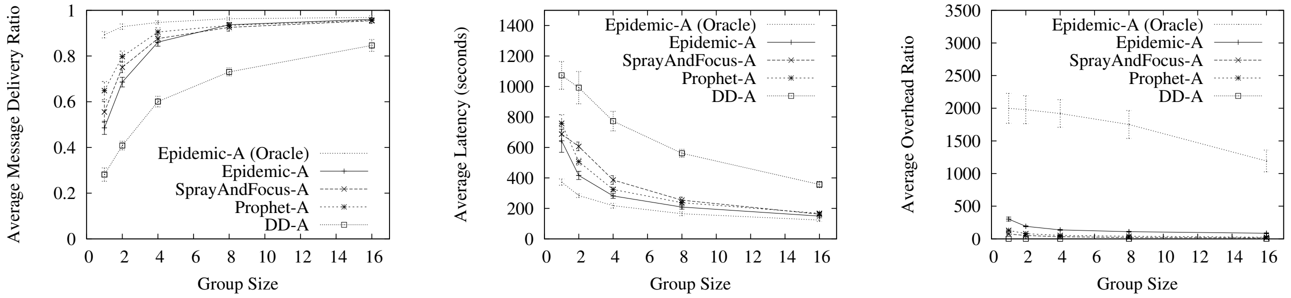

Figure 4: RC, ack, back channel (a) MDR, (b) Delay, (c) Overhead

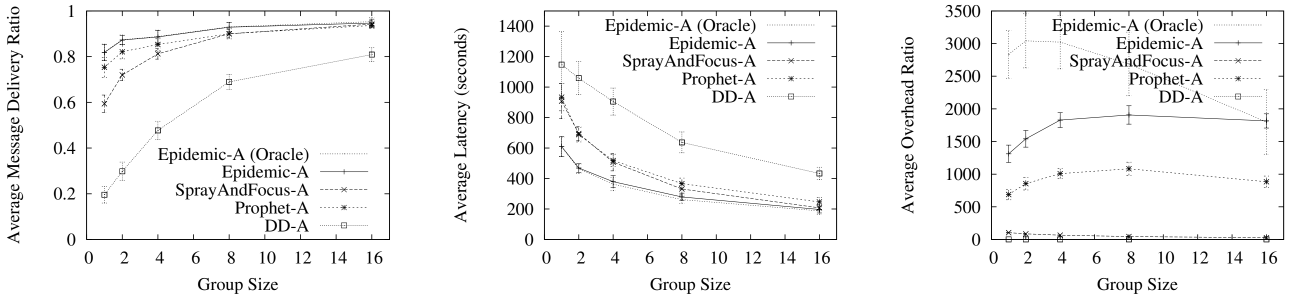

Figure 5: Non-RC, no acks, no back channel (a) MDR, (b) Delay, (c) Overhead

4.3.1 Resource-Constrained

In this set of simulations, the transformed anycast protocols attempt to deliver messages in a resource-constrained environment. Message sizes vary between 500k and 1M, and a message is generated by a random node every 25 to 35 seconds, destined to a random group. In this scenario, cars are traveling at normal speeds (2.7 to 13.9 meters per second, as previously mentioned).

With no acknowledgements and no access to a back channel (see Figure 2), anycast behavior is actually quite different than unicast behavior. In a unicast environment (when the group size is 1), quota-based protocols (e.g., Spray and Focus-A) perform only slightly better than smart flooding-based protocols (e.g., Prophet). However, the gap between quota-based protocols and flooding-based protocols becomes much larger as the group size increases (as shown in Figure 2 (a)). With larger groups, it gets increasingly easy to meet a group member, and hence limiting replication is not hurting the network. In this resource-constrained environment, flooding-based protocols (particularly Epidemic-A) quickly overwhelm resources (as shown in Figure 2 (c)) to the extent that only a small fraction of buffered messages are transmitted during every brief contact opportunity. This indicates that group size is a very important for protocol performance. Another surprising result is that as group size increases, DD-A actually performs as well as Prophet-A, although Prophet-A still unsurprisingly delivers messages quicker (as shown in Figure 2 (b)). This is because nodes running DD-A are able to eventually meet most groups when group sizes are large, without causing any strain on resources (no message drops, not as much worrying about contact duration, very little congestion, etc.). Overall, quota-based protocols are recommended in resource-constrained environments, particularly if group sizes are large and there is reasonable mobility.

Next, consider the presence of acknowledgements, but the lack of back channel access, as shown in Figure 3. It is immediately clear that acknowledgements drastically improve the performance of flooding-based protocols since these protocols, including Epidemic-A and Prophet-A, consume the most resources, to the point that they are near optimal, along with Spray and Focus-A, when the group size is above 8 (as shown in Figure 3(a)). Quota-based protocols are slightly improved, and DD-A is not improved at all since there is already a limit of at most 1 copy of every message in the network. Average delay also decreases slightly for all protocols. In terms of overhead, a significant improvement is seen, since wasteful transmissions (e.g., messages that have already been delivered) are minimized. As expected, the overhead ratio is much lower, especially for flooding-based protocols (as shown in Figure 3(c)).

Finally, consider the presence of an acknowledgement scheme and back channel access. In other words, if a group member receives a message destined for their group, that member immediately informs all other group members to transmit ACKs for that particular message. Interestingly, access to a back channel does not bring a significant improvement to any metric, as shown in Figure 4. This is because back channel access is expected to have the most effect when group sizes are large (since there are more nodes to initially spread ACKs), but when group sizes are large and ACKs are used, all protocols perform at a high level and hence the back channel ACKs do not help much.

4.3.2 Non-Resource-Constrained

In this set of simulations, the transformed anycast protocols attempt to deliver messages in a non-resource-constrained

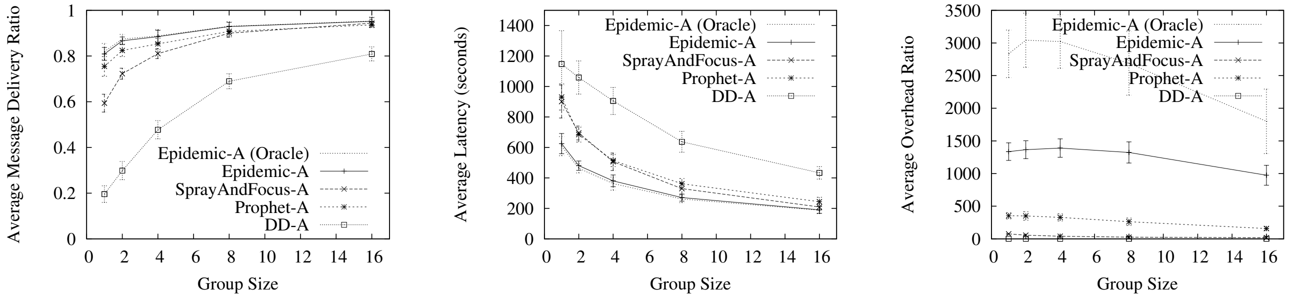

Figure 6: Non-RC, acks, no back channel (a) MDR, (b) Delay, (c) Overhead

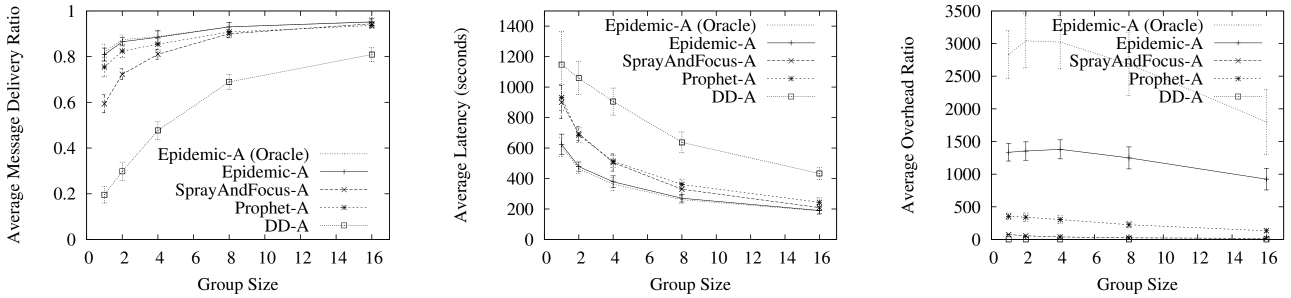

Figure 7: Non-RC, ack, back channel (a) MDR, (b) Delay, (c) Overhead

environment. Message sizes vary between 50k and 100k, and a message is generated by a random node every 50 to 70 seconds, destined to a random group. Furthermore, cars travel much slower (hence, contact time is less of a resource constraint) at speeds of between 2.7 and 3 meters per second, which also has the effect of allowing less node mixing. In addition to the first set, this helps us see how protocols react to different ends of the resource spectrum.

As before, we first consider the scenario where there are no acknowledgements and no back channel access (see Figure 5). What is immediately clear is that flooding-based protocols perform better than quota-based protocols in both MDR and average delay, particularly when group sizes are small, which mirrors unicast observations. This is because their flooding-based approach is appropriate in less resource-constrained environments, since adding extra messages to the network is more helpful than harmful when the messages are smaller and message production is less frequent. Furthermore, the reduced mobility allows less node mixing, which hinders quota-based protocols since many of the replicas may stay clustered together. One very interesting observation that differs from unicast behavior, is that after group sizes become reasonably large, the MDR difference between flooding and quota based protocols is almost negligible. This is because it is relatively easy to meet groups, even in lower mobility environments, when they are large and hence both types of protocols can perform well. In terms of delay, Epidemic-A performs best, as expected, with the flooding and quota-based protocols performing similarly. However, all are similar with large group sizes (16 or greater). Overhead results are as expected, with Epidemic-A and Prophet-A being the least resource-friendly and Spray and Focus-A being extremely resource-friendly. Therefore, if resources such as battery life or buffer size are severely limited, quota-based protocols are the best choice.

It is important to note that overhead cannot be compared to the first set of simulations, and must be compared only in a relative fashion between protocols in the second set of simulations. Due to the message sizes being much smaller in this second set, many more transmissions per contact occur, meaning the overhead ratios for the protocols are much greater in this set than the first. However, this does not mean that the resource utilization is greater, since the message sizes are much smaller.

The second case shows the results when an acknowledgement scheme is added to the protocols, but there are no back channels. Interestingly, this does not significantly affect either the MDR or the delay metrics, as shown in Figure 6. This is due to the low resource utilization to start with, and hence, freeing up the resources does not significantly impact the performance. It does, however, have a very large effect in terms of overhead. Prophet-A and Epidemic-A have their overhead significantly lowered, since many extra and useless transmissions are eliminated. For the same reasons as in the first set of simulations, adding a back channel for each group does not significantly change any of the metrics, as shown in Figure 7.

5. CONCLUSIONS AND FUTURE WORK

Groups are fundamental entities in many DTN environments, making effective and efficient anycast communication important. In this paper, we have proposed an overarching approach that allows many current DTN unicast protocols to be enhanced to support anycast communication. It is shown that this approach can be protocol-independent, and implemented in a thin shim beneath the routing layer, essentially changing the world view of the protocol. Using this approach, we were able to enhance many popular unicast routing protocols, including flooding-based protocols such as Prophet and epidemic, and quota-based protocols such as Spray and Focus. A thorough simulation-based evaluation of these newly enhanced protocols indicates that many factors, such as group size, the presence of an acknowledgement scheme, and the resource-constraints of the environment have a significant impact on anycast performance.

There are many avenues to be explored in future work. First, using the knowledge learned from the evaluation presented, we plan to specifically create anycast protocols from scratch and evaluate them against the enhanced unicast protocols. Second, we plan to explore the idea of what we refer to as *preferred anycast*, where the goal is to reach any member of a particular group, but it would be preferred to reach a specific subset of that group. This brings the idea of hierarchy into play. Third, we plan to explore the usefulness and practicality of multicast techniques, where the goal is to reach all members of a particular group.

6. REFERENCES

[1] M. Abdulla and R. Simon. A simulation analysis of multicasting in delay tolerant networks. In *Proceedings of Winter Simulation Conference*, 2006.

[2] A. Balasubramanian, B. Levine, and A. Venkataramani. DTN routing as a resource allocation problem. In *Proc. ACM SIGCOMM*, 2007.

[3] S. Bhattachargee, M. Ammar, E. Zegura, N. Shah, and Z. Fei. Application layer anycasting. In *Proceedings of IEEE INFOCOM*, 1997.

[4] J. Burgess, Br. Gallagher, D. Jensen, and B. Levine. MaxProp: Routing for vehicle-based disruption-tolerant networks. In *Proc. IEEE INFOCOM*, 2006.

[5] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6(6):606–620, 2007.

[6] Ederson Rosa da Silva and Paulo Guardieiro. Anycast routing in delay tolerant networks using genetic algorithms for route decision. In *Proceedings of IDCS*, 2008.

[7] Y. Gong, Y. Xiong, Q. Zhang, Z. Zhang, W. Wang, and Z. Xu. Anycast routing in delay tolerant networks. In *GLOBECOM*, 2006.

[8] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *Proceedings of ACM MobiHoc*, 2008.

[9] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *Proceedings of SIMUTools '09*, 2009.

[10] A. Lindgren, A. Doria, and O. Scheln. Probabilistic routing in intermittently connected networks. In *MobiHoc 03*, 2003.

[11] S. Nelson, M. Bahkt, and R. Kravets. Encounter-based routing in dtns. *Proceedings of IEEE InfoCom*, 2009.

[12] S. Nelson, A. Harris, and R. Kravets. Event-driven, role-based mobility in disaster recovery networks. In *Proceedings of ACM CHANTS*, 2007.

[13] Samuel C. Nelson and Robin Kravets. For members only: Local and robust group management in dtns. In *In Proc. ACM Mobicom CHANTS workshop*, 2010.

[14] V.D. Park and J.P. Macker. Anycast routing for mobile networking. In *Proceedings of IEEE Military Communications Conference*, 1999.

[15] J. Scott, P. Hui, J. Crowcroft, and C. Diot. Haggle: A networking architecture designed around mobile users. In *WONS*, 2006.

[16] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proceedings of ACM WDTN '05*, 2005.

[17] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In *Proceedings of IEEE PerCom*, 2007.

[18] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-2000-06, Department of Computer Science, Duke University, April 2000.