



On the implications of routing metric staleness in delay tolerant networks

Mike P. Wittie^{a,*}, Khaled A. Harras^b, Kevin C. Almeroth^a, Elizabeth M. Belding^a

^a Department of Computer Science, University of California, 616 Mulberry Ave., Santa Barbara, USA

^b Computer Science Department, Carnegie Mellon University, Qatar

ARTICLE INFO

Article history:

Available online 12 February 2009

Keywords:

DTN
Routing
Congestion

ABSTRACT

Delay Tolerant Network (DTN) routing addresses challenges of providing end-to-end service where end-to-end data forwarding paths may not exist. The performance of current DTN routing protocols is often limited by routing metric “staleness”, i.e., routing information that becomes out-of-date or inaccurate because of long propagation delays. Our previous work, ParaNets, proposed a new opportunistic network architecture in which the data channel is augmented by a thin end-to-end control channel. The control channel is adequate for the exchange of control traffic, but not data. In this paper we present Cloud Routing, a routing solution for the ParaNets architecture. We motivate the need for such a solution, not only because of stale routing metrics, but also because of congestion that can occur in DTNs. Unable to use up-to-date routing metrics to limit congestion, existing DTN routing solutions suffer from low goodput and long data delivery delays. We show how Cloud Routing avoids congestion by smart use of forwarding opportunities based on up-to-date routing metrics. We evaluate our solution using extensive OPNET simulations. Cloud Routing extends network performance past what is currently possible and motivates a new class of globally cognizant DTN routing solutions.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Delay Tolerant Network (DTN) routing protocols aim to provide the perception of full network connectivity even when an instantaneous source-to-destination path does not exist. Existing routing solutions leverage opportunistic node contacts to flood data across the network. Because flooding creates many packet copies in the network, a number of mechanisms have been developed to limit replication. These mechanisms scope flooding using hop count limits [1] or *curing* techniques [2], or exploit mobility patterns to forward data only to nodes promising more timely and more efficient delivery [3–7].

High delays inherent in opportunistic forwarding render DTN routing metrics less effective. Opportunistically disseminated information, on which routing metrics rely, can quickly become out-of-date, or stale. Worse, information from different network regions can experience varying amounts of delay and, when combined, represent a state of the network that never existed.

Existing DTN routing solutions deal with the uncertainty in routing information by replicating data. The most efficient routing solution, in terms of network resource usage, would forward only one data copy along some set of hops. Due to the uncertainty of

link availability in DTNs, current solutions create many copies of the same packet so that multiple paths can be explored. However, opportunistically propagated metrics can be stale and existing solutions are forced to replicate data without considering data copies that are already close to the destination. Replication performed independently of the global traffic state creates potential for congestion as more and more superfluous copies of the same data are created in the network.

The high degree of data replication in DTNs is at odds with limited node contact time. Nodes are often in contact only for short periods of time, during which only a small amount of data can be exchanged. Under these conditions, congestion can occur when a large amount of flooded data is queued and is expected to be forwarded when nodes come into contact with each other. Because of high degree of replication, even small increases in network load result in large increases in the amount of data queued at each node. The DTN research community has often assumed that buffer space is not a limiting factor. Even so, the growth in buffered data means that more data needs to be transmitted during a node contact, increasing the probability of packets being stalled during an exchange. Stalled packets occupy network resources for a longer period of time, increasing delivery time and reducing network goodput.

The evolution of DTNs themselves may provide a solution to the fundamental problem of metric staleness and ensuing congestion. While early DTN work assumed a flat topology of mobile nodes, recent research trends have proposed more diverse architectures,

* Corresponding author. Tel.: +1 8052801690.

E-mail addresses: mwittie@gmail.com, mwittie@cs.ucsb.edu (M.P. Wittie), kharras@cs.cmu.edu (K.A. Harras), almeroth@cs.ucsb.edu (K.C. Almeroth), ebelding@cs.ucsb.edu (E.M. Belding).

where the opportunistic network is augmented by an end-to-end communication channel [8–10]. The possibility of end-to-end connectivity supplementing an opportunistic network creates a new space for DTN routing solutions. These new solutions could even be designed to make globally optimal routing decisions using up-to-date network state information. As a result, such solutions would successfully avoid the problems of buffer growth and poor use of node contact time.

Recent work describes DTNs that have been augmented by another network. Jain et al. suggested using multiple delay tolerant networks in unison to deliver data of different throughput and delay characteristics [11]. Surana et al. described a rural data forwarding network where monitoring information is exchanged using a Short Message Service (SMS) back channel [8]. Finally our previous work, ParaNets, considered the possibility of linking multiple networks at different layers in the OSI stack [9], or more specifically, by Chandra and Bahl at the MAC layer [12].

The core idea of ParaNets is to use multiple networks simultaneously such that each network performs only the tasks of a protocol suitable to its characteristics. We focus on the scenario where bulk data is forwarded on an opportunistic primary network and protocol control information is exchanged using an end-to-end control channel. In previous work, we have shown how a ParaNets control channel can be used to exchange data delivery acknowledgments to purge the network of superfluous copies of delivered data [9], a solution independently arrived at by Yuen and Schulzrinne [10].

In this work we explore how an end-to-end control channel can be used to eliminate metric staleness in existing DTN routing solutions. We also introduce new mechanisms that take advantage of end-to-end up-to-date network state information exchanged over the control channel to address the problems of DTN congestion and inefficient use of node contacts. We propose to reorder data queued for exchange during a contact such that packets more likely to effect deliveries are forwarded first. Our solution, *Cloud Routing*, reorders data queued for transmission using two mechanisms: *Connection Utility Reordering* and *Cloud Rank Reordering*.

Connection Utility Reordering enables efficient use of node contact time by reordering packets in order of their routing metric. While effective at lowering delay, such greedy reordering can lead to starvation of packets far from their destinations. To assure forward progress of data in the network, we add Cloud Rank Reordering.

Cloud Rank Reordering forwards data in order of its routing metric with respect to other copies of the same data residing elsewhere in the network. This mechanism enables globally aware routing decisions. Cloud Rank Reordering also assures fairness by forwarding at least one copy of each packet with the highest priority. Further, we show that using Cloud Rank Reordering and Connection Utility Reordering in combination leads to even greater network performance improvements.

While we assume the availability of a ParaNets control channel, we show that Cloud Routing performs well even when end-to-end connectivity is intermittent. This result shows we can realize the benefits of additional infrastructure and retain the generality of *disconnectivity* that characterizes DTNs.

We evaluate our work using extensive OPNET simulations of TCP-based node contact connections. The use of TCP allows us to accurately model the amount of data that can be realistically exchanged during a node contact and show the effects of DTN congestion caused by routing metric staleness. We build on these findings to demonstrate the marked benefits of employing Cloud Routing.

We compare our results against the PROPHET and Epidemic Routing solution [3,1]. While more straightforward to implement, PROPHET's performance is comparable to other contact history

based routing solutions [13]. Cloud Routing achieves a factor of eight improvement in network goodput over PROPHET and a factor of two decrease in packet delay over Epidemic Routing. We also show 4720 kbps of network goodput on the ParaNets data channel for every one kbps of control channel traffic.

We conclude that the availability of an end-to-end control channel enables a fundamental shift in DTN routing and achievable performance improvements. We believe these gains make opportunistic forwarding an attractive augmentation to end-to-end networks where content dissemination over the opportunistic network is a possibility.

The remainder of this paper is organized as follows. In Section 2 we discuss current DTN routing solutions, as well as the ParaNets architecture. In Section 3, we demonstrate the challenges to DTN routing stemming from staleness of network state information. In Section 4, we detail the Cloud Routing protocol mechanisms. Section 5 describes the simulation setup used to obtain the results referenced throughout this work. In Section 6, we evaluate Cloud Routing using OPNET simulations. Finally, we conclude in Section 7.

2. Related work

A number of DTN routing solutions have been proposed by the research community. Epidemic Routing by Vahdat and Becker was the first work to achieve reliable delivery in DTNs [1]. Epidemic Routing floods data packets aggregated into bundles throughout the network. Eventually one of the bundle copies is expected to reach the destination. The work represents a proof of concept, opening the field to further study. Cerf et al. [14] standardized a more general bundle relay architecture under the auspices of the Internet Research Task Force (IRTF) Delay Tolerant Networking Research Group (DTNRG).

A different approach to DTN routing was proposed by Lindgren et al. [3]. The Probabilistic Routing Protocol using History of Encounters and Transitivity (PROPHET) maintains at each node a delivery probability vector to all other nodes. Delivery probabilities are calculated from node contact history information. If node A encounters node B in some time period, A updates its historical probability of meeting B using Eq. (1), where $P_{init} \in [0, 1]$ is an initialization constant.

$$P_{(a,b)} = P_{(a,b)old} + (1 - P_{(a,b)old}) \times P_{init} \quad (1)$$

If node A does not encounter node B in some time, node A is less likely to be a good forwarder to B and A's delivery probability should be reduced. A uses Eq. (2) to *age* its delivery probability to B with respect to the amount of time passed since last contact. Parameter $\gamma \in [0, 1]$ is the *aging constant* and k is the number of time units since last update of the delivery probability.

$$P_{(a,b)} = P_{(a,b)old} \times \gamma^k \quad (2)$$

Finally, PROPHET adjusts delivery probability using *transitivity*. If A is a good forwarder to B and B is a good forwarder to C, then A is likely to also be a good forwarder to C. Eq. (3) allows A to calculate its delivery probability to C from delivery probabilities of A to B and B to C. Parameter $\beta \in [0, 1]$ is a scaling constant that governs the impact of transitivity on a delivery probability.

$$P_{(a,c)} = P_{(a,c)old} + \beta(1 - P_{(a,c)old})P_{(a,b)}P_{(b,c)} \quad (3)$$

During contact, nodes exchange their vectors and recompute new delivery probabilities. PROPHET forwards data bundles only to nodes promising a higher delivery probability to a particular destination.

Context-Aware Routing (CAR) by Musolesi et al. uses a similar approach [4]. In addition to node contacts, CAR bases its delivery

probability calculations on additional node attributes such as rate of change of mobility and connectivity, as well as remaining battery charge.

MobySpace developed by Leguay et al. also uses multi-attribute algorithms to determine a node's utility as a forwarder [5]. A node's mobility pattern and other attributes are mapped into a multidimensional Euclidean space dubbed *MobySpace*. The closer a node's mapping is to that of the destination, the higher is the node's utility as a forwarder.

Routing solutions that rely on altering node mobility to aid network connectivity have also been developed. Burns et al. proposed MV Routing, which keeps track of node meetings as well as contacts to calculate a delivery probability vector at each node [6]. Additionally, MV Routing relies on *autonomous agents* to aid network connectivity. Scheduling the mobility of autonomous agents is a major contribution of the work. Li and Rus also took advantage of node mobility by having the communicating nodes themselves modify their trajectories to achieve contacts [15]. They evaluated their scheme under globally known and unknown mobility patterns.

A more recent work by Song and Kotz proposed estimating delivery probability within some period of time [13]. They evaluated their solution using a mobility model derived from user associations with campus wireless access points collected as part of the CRAWDAD project.¹ Similarly to Song and Kotz's work, Hsu et al. analyzed large CRAWDAD traces of user mobility profiles [7]. Their solution, Profile-Cast, delivers bundles to groups of nodes described by a multi-variate profile.

The performance of DTN routing solutions evolved from Epidemic Routing suffers from congestion created by high degree of data replication. Additionally, because routing information is rendered stale by opportunistic dissemination, the routing decisions based on this information may not be appropriate for the current state of the network. We discuss the problems of congestion and routing metric staleness plaguing DTN routing solutions rooted in Epidemic routing more fully in Section 3.

Our previous work introduced a multi-channel DTN architecture called ParaNets [9]. Due to the tradeoff between data rate, radio range, and power, radios achieve either high data rate at short range, like Wireless LANs (WLANs), or low data rate at long range, like the General Packet Radio Service (GPRS). ParaNets takes advantage of this duality, by forwarding data traffic on the short, high rate connection, dubbed the *primary network*, and the control traffic on the long, low rate connection, dubbed the *control network*. Examples of such ParaNets are: an IEEE 802.11 primary network with a cellular control channel, or a Vehicular Ad-hoc Network (VANET) supported by a satellite control channel. In these ParaNets, the control channel may not be suitable for high volume data traffic, due to bandwidth, price, or policy considerations, but it is able to efficiently exchange control traffic, resulting in a marked performance increase of protocols running in the primary network. Fig. 1 shows a DTN instantiation of the ParaNets architecture using a WLAN primary network and GPRS control channel. While the concept of multiple networks in DTNs had already been explored by Jain et al. [11], we believe ParaNets was the first work proposing the split of control and data traffic in the network stack for DTNs.

In this work, we explore the performance gains that stem from employing the ParaNets control channel to communicate routing information. We use this information to make informed traffic shaping and forwarding decisions via a new DTN routing protocol called Cloud Routing. When contact time is limited, data that is most likely to contribute to eventual delivery should be forwarded first. Cloud Routing uses up-to-date routing information to reorder

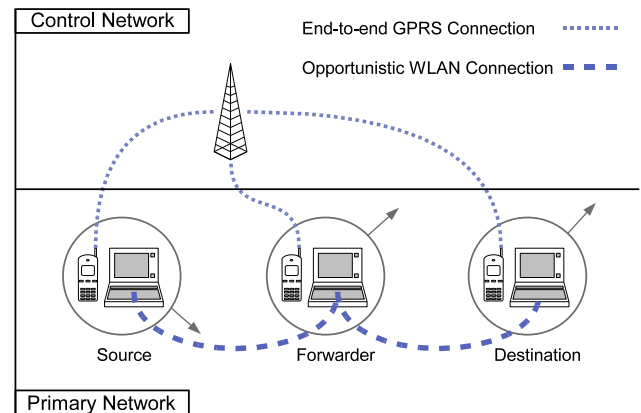


Fig. 1. A ParaNets-based architecture.

transmission queues such that bundles with best routing metrics are transmitted first. While traffic prioritization approaches for DTNs have been proposed previously, these solutions do not consider starvation that stems from greedy prioritization [3,16]. These solutions also do not take into account global traffic state and are able to make only locally optimal routing decisions, which often do not result in better network performance. We address the problems of greedy prioritization and globally agnostic routing decisions, as well as Cloud Routing solutions to these problems, more fully in Section 4.

3. Challenges of routing in DTNs

While mobility pattern discovery-based routing is a promising direction, we believe this approach is currently limited by the long propagation time of routing information. Out-of-date routing information degrades the accuracy of routing decisions and results in significant traffic costs and inefficient as well as of node contact time.

3.1. Routing metric staleness

Network state information is exchanged during opportunistic contacts and is likely to experience significant delays in traversing a DTN. When this information is then used to calculate routing metrics, these metrics are stale with respect to current network state. Due to varying delays of network state information collected from different network regions, routing metrics may not only represent an old state of the network, but worse, may represent a network state that never existed at any one point in time.

To demonstrate the difference up-to-date network state information can make in DTN routing, we experimented with a modified version of PROPHET to exchange contact information over the ParaNets control channel. The augmented solution, ParaNets PROPHET, operates as follows. Whenever two nodes meet, the delivery probability is updated according to Eq. (1). In addition to the standard PROPHET functionality, ParaNets PROPHET timestamps the obtained delivery probability and marks it as "updated." Similarly, when a delivery probability is updated during a node contact through transitivity using Eq. (3), the delivery probability entry is timestamped and marked. ParaNets PROPHET then multicasts the marked entries to all nodes with access to the ParaNets control channel and clears the marked flag. In our simulations we configured the update interval to 30 seconds, a value we found to offer an attractive tradeoff between ParaNets control channel overhead and primary network performance.

A node A receiving a delivery probability, $P_{(b,c)}$, from the control channel checks if the advertised delivery probability's timestamp is

¹ <http://crawdad.cs.dartmouth.edu>.

older than the timestamp for its probability, $P_{(a,c)}$. Node A also checks that it has a probability entry $P_{(a,b)}$. If both of these conditions hold, node A updates $P_{(a,c)}$, using Eq. (3) and timestamps the result with the timestamp of $P_{(a,b)}$ or $P_{(b,c)}$, whichever one is older. The node also marks the $P_{(a,c)}$ as “updated,” to be sent out during the next ParaNets PROPHET update. Finally, delivery probabilities are aged in ParaNets PROPHET using Eq. (2) just like in standard PROPHET.

The mechanisms of ParaNets PROPHET allow nodes to update their delivery probabilities more rapidly than using opportunistic contacts alone, effectively eliminating routing metric staleness. In Section 6 we show ParaNets PROPHET to achieve a factor of 2.4 goodput improvement and a factor of 1.3 delay improvement in primary network performance over standard PROPHET. This experiment shows the significant performance improvements that can be achieved by simply disseminating routing metrics on the ParaNets control channel and using this up-to-date information to make routing decisions. While successful at improving network performance, ParaNets PROPHET does not explicitly address the problem of congestion and its performance remains limited.

3.2. Potential for congestion

Current DTN routing solutions rely on scoped flooding. A bundle is forwarded during contact between the current node and a potential forwarding node if the potential forwarder is more likely to be able to deliver the bundle. The goal is to reduce delivery cost, counted as the number of bundle copies created in the network.

We have verified experimentally that indeed standard PROPHET and ParaNets PROPHET can deliver data at a significantly lower cost than Epidemic routing. The lower cost of delivery means that fewer copies of each bundle reside in the network, allowing the PROPHET solutions to also achieve lower buffer length than Epidemic Routing.

Using a simple simulation based on the setup described in Section 5, Fig. 2 shows mean bundle buffer length under PROPHET, ParaNets PROPHET, and Epidemic Routing as affected by network load. We observe that not only is the mean buffer length significantly lower for the PROPHET solutions than for Epidemic Routing, but the same is true of the rate of buffer growth with respect to traffic load: 0.68, 0.35, and 0.28 for Epidemic Routing, standard PROPHET, and ParaNets PROPHET, respectively. In practice, however, the smaller buffer size does not necessarily translate to efficient use of node contact time.

While buffer length under the PROPHET solutions is much smaller than under Epidemic Routing, the amount of data queued for transfer during a node contact still grossly exceeds the amount that can be forwarded. We demonstrate the imbalance between data queued for forwarding and what is actually exchanged in Fig. 3.

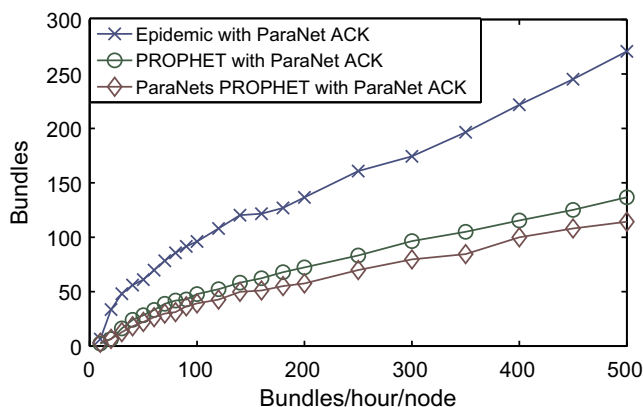


Fig. 2. Mean node buffer size versus network traffic load.

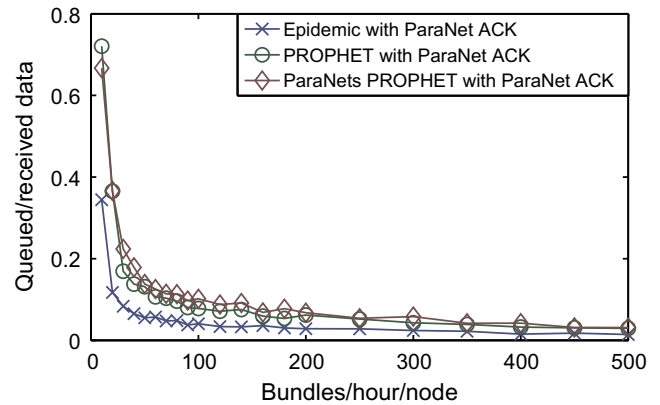


Fig. 3. Ratio of queued to received node connection data versus network traffic load.

Fig. 3 shows the mean ratio of data inserted into the sender’s TCP connection buffer to data received before the connection is closed or aborted. As bundle load increases, so does the mean amount of bundle data enqueued for each connection. However, the amount of received data during each connection depends on node contact time, which remains constant with respect to network load. Increased bundle load leads to a growing difference between connection load and connection throughput. This imbalance creates potential for network congestion at higher loads. While the up-to-date routing information used in ParaNets PROPHET reduces the ratio of received to forwarded data, the ratio remains small in absolute terms. The small percentage of data that can be exchanged during a node contact creates a need for efficient use of node contact time.

4. Cloud Routing

We believe that bundle traffic growth, the resulting congestion, and the inefficient use of node contact time, cannot be effectively solved by local use of routing metrics alone. DTN routing solutions need to be smarter and use up-to-date network state information to make globally cognizant routing decisions. We propose to leverage the ParaNets architecture to exchange global state information and allow up-to-date routing metric calculation at each node. To take advantage of up-to-date network state and address the problem of inefficient use of node contact time, we design two mechanisms, *Connection Utility Reordering* and *Cloud Rank Reordering*.

The routing metric referenced in the description of these mechanisms is the Cartesian distance-to-destination, which has the advantage of being simple to compute from node position information communicated over the ParaNets control channel, while offering meaningful information under the mobility model used in our simulations. However, the Cloud Rank Reordering and Connection Utility Reordering mechanisms could be adapted to use other routing metrics, such as contact history [3], mobility patterns [5], or multi-attribute node profiles [7].

We combine Cloud Rank Reordering and Connection Utility Reordering together into *Combined Cloud Reordering*. Combined Cloud Reordering capitalizes on the good network performance of Connection Utility Reordering and globally informed routing decisions of Cloud Rank Reordering to achieve better network performance than the constituent solutions do separately.

4.1. Connection Utility Reordering

Inefficient use of node contact time occurs when bundles forwarded during an exchange do not result in eventual delivery.

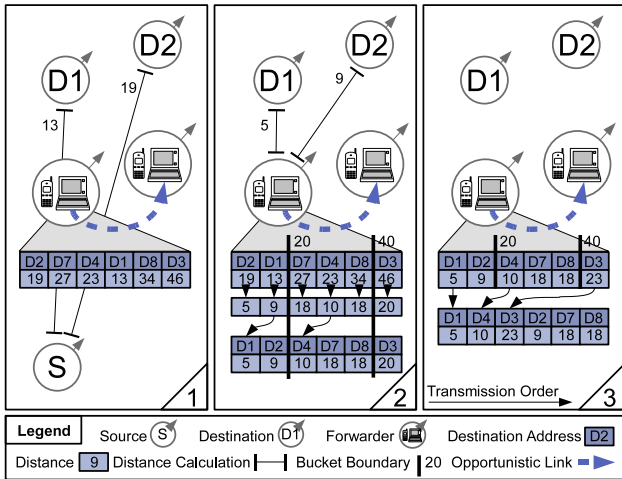


Fig. 4. Connection Utility Reordering operation.

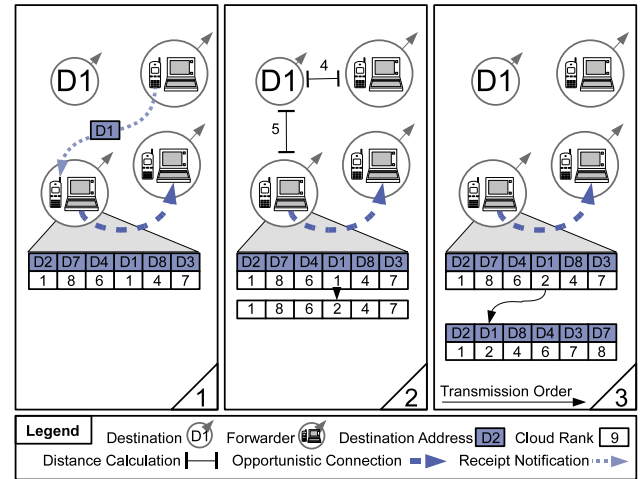


Fig. 5. Cloud Rank Reordering operation.

We propose Connection Utility Reordering, a greedy scheme to reorder data queued for transmission such that bundles that are closest to their destination are forwarded first.

However, forwarding bundles closest to their destinations first leads to starvation of long-range bundles. In such cases, we have observed that short range flows receive a disproportionately large share of the node contact time, resulting in local traffic causing the starvation of long-range network traffic. To assure long-range flow progress, we propose *fair shortest distance* ordering. Under fair shortest distance ordering, short-range flows compete for node contact time with other short-range flows, while long-range flows compete with other long-range flows.

Connection Utility Reordering operation is shown in Fig. 4. In Frame 1, the forwarding node calculates the distance to the destination, or *bundle range*, for each bundle in its send queue, using node position information disseminated over the ParaNets control channel. In Frame 2, the forwarding node sorts bundles by their range into a small number of buckets, three in our example. Each bucket represents a certain range of flows such as short, medium, and long. The forwarding node then replaces the range of each bundle with distance-to-destination measured from that node. Each bucket is then reordered by shortest distance first. Finally, in Frame 3, connection transmission order is created by removing the first bundle from each bucket in round robin order. Connection Utility Reordering allows bundles within each range that are closest to their destination to be transmitted first, thus making efficient use of node contact time. At the same time, long-range bundles close to their destinations are moved up in the forwarding queue; are forwarded even during short contacts; and avoid starvation behind short-range traffic.

Fair shortest distance allows similarly ranged bundles to compete with each other within each bucket. While the goodput of two or three bucket fair distance reordering is lower than that of one bucket reordering, the avoidance of starvation render a multi-bucket scheme preferable in networks where fair treatment is desired.

4.2. Cloud rank reordering

While Connection Utility Reordering allows longer range bundles their fair share of each node connection, that fairness is enforced in a somewhat arbitrary, albeit systematic, manner. Bundles close to their destination, but transmitted after a less urgent bundle, often miss critical forwarding opportunities. In practice, network resources are wasted if another copy of a bundle is

the one that results in eventual delivery. Cloud Rank Reordering aims to provide a more globally aware mechanism to assure fairness.

The Cloud Rank Reordering mechanism relies on the concept of a *bundle cloud*. A bundle cloud is an abstraction that logically includes all the nodes that carry copies of a particular bundle. As a bundle is replicated in the network, the logical cloud grows in size, spreading to network regions far from the source, but hopefully towards the destination. When described in terms of bundle clouds, the original problem of Connection Utility Reordering unfairness manifests itself when bundles in the clouds closer to their destinations are likely to be forwarded instead bundles in clouds farther from their destinations. In practice, large clouds close to their destinations prevent new clouds from making forward progress. We rectify this situation using Cloud Rank Reordering.

Cloud Rank Reordering reorders node send buffers such that forwarding priority is given to bundle copies, or *cloud members*, closest to the destination. Cloud Rank Reordering operation is shown in Fig. 5. In Frame 1 the forwarding node receives notification that another node received a copy of a particular bundle addressed to Destination 1. These notifications are multicast over the ParaNets control channel to all ParaNets enabled nodes.

We define *cloud rank*, as the position in the cloud according to increasing distance-to-destination. Cloud rank is calculated for each cloud member in a send queue with respect to other members of the same bundle cloud in the network. Using a list of known cloud members and node position information, in Frame 2 the forwarding node calculates the cloud rank of each bundle in its forwarding queue. Notice the cloud rank for the bundle destined to Destination 1 is updated. Finally, in Frame 3 the forwarding node reorders its send buffer in increasing cloud rank order.

Cloud Rank Reordering assures forward progress in a more meaningful way than fair shortest distance order. Under Cloud Rank Reordering, equally ranked cloud members of each cloud are forwarded at the same priority regardless of the distance to their destination. Additionally, Cloud Rank Reordering makes globally cognizant routing decisions by considering global traffic state. Cloud Rank Reordering forwards the leading edge of each cloud before more distant cloud members of other clouds. A more subtle point is that Cloud Rank Reordering also minimizes the impact of traffic growth.

During limited node contact time, the Cloud Rank Reordering mechanism assures that cloud members far from their destinations do not preempt traffic closer to their destinations. Since it is likely that high ranking cloud member bundles are not transmitted

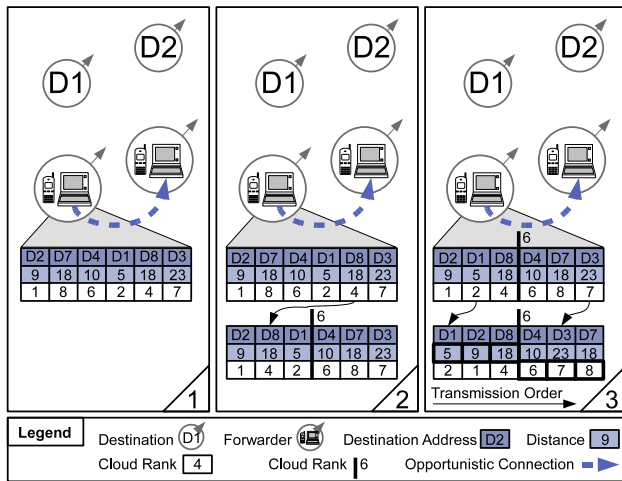


Fig. 6. Combined Cloud Routing operation.

during node contact, it is as if they were not present at the forwarding node at all. The net result is that cloud members far from a final destination are effectively “pruned” from the network if there is not enough node contact time to forward them. Thus, in networks limited by node contact time, Cloud Rank Reordering limits the effects of bundle traffic growth by rendering the impact of distant cloud members negligible. Because high ranking cloud members do not contribute to congestion, network performance improves.

However, as we see in Section 6, while outperforming contact history based routing, Cloud Rank Reordering does not perform as well as Connection Utility Reordering. Cloud Rank Reordering is not effective at arbitrating between equally ranked cloud members of different clouds, and so a high ranking cloud member far from the destination can be forwarded instead of a high ranking cloud member close to its destination. We combine Connection Utility Reordering and Cloud Rank Reordering to achieve good network performance while assuring fair forward progress for all bundles clouds.

4.3. Combined mechanisms

Combined Cloud Reordering, the integration of Cloud Rank Reordering and Connection Utility Reordering, has the potential to achieve better network performance than either mechanism does separately. Since both mechanisms reorder node send buffers, a meaningful method of using these solutions together is needed.

Fair distance ordering can suffer from *range inversions*, occurring when longer range bundles are transmitted before shorter range bundles. While some range inversions benefit the system by preserving fairness, inversions caused by high ranking cloud members can stall bundles close to their destinations without themselves resulting in bundle deliveries.

To eliminate range inversions caused by bundle copies far from their destination, we again make use of the concept of bundle cloud. Bundles with cloud rank below a specified cloud size are reordered using single bucket Connection Utility Reordering. Bundles whose cloud rank at a node is above the specified cloud size, are reordered by increasing cloud rank of Cloud Rank Reordering. The send buffer is the concatenation of the two orderings, such that bundles ranking below the specified cloud size are transmitted first.

Fig. 6 shows Combined Cloud Reordering operation in detail. In Frame 1 a forwarder node calculates the distance-to-destination and cloud rank of each bundle. Then in Frame 2, the node sorts

its send queue into two sets each representing bundles with cloud ranks above and below a specified cloud size. Finally in Frame 3 bundles below the specified cloud size are ordered using one bucket Connection Utility Reordering and bundles above cloud size are ordered using Cloud Rank Reordering. Bundles below the specified cloud size are transmitted first.

By moving high cloud rank bundles out of the Connection Utility Reordering ordering, we eliminate the need for fair shortest distance order and the range inversions it causes. This mechanism allows Combined Cloud Reordering to realize the performance benefits of Connection Utility Reordering, while still safeguarding traffic from starvation. The bundles with cloud rank above the specified cloud size may not be transmitted during limited contact opportunity, but as there are copies that are closer to their destinations, the potential stalling of high ranking cloud members has little effect on network performance. As we demonstrate in Section 6, Combined Cloud Reordering improves on the goodput and delay performance of Connection Utility Reordering, while maintaining the fairness of Cloud Rank Reordering.

A word on the choice of cloud size. Specifying a very large cloud size in Combined Cloud Reordering will result in greedy forwarding of one bucket Connection Utility Reordering. We have observed a natural limit to cloud size for a particular network and traffic load. If cloud size is set close to that value, Combined Cloud Reordering becomes greedy-shortest-distance-first, thus losing its fairness properties. On the other hand, setting a cloud size too low in a heavily loaded network, can result in restricting the number of bundles of each cloud being forwarded to that size, as the bulk of forwarded nodes rank below the specified cloud size. While specifying small cloud size assures fairness of access to network resources for each cloud, it does so at the cost of performance, as the network suffers from the long delays of single copy routing. In our simulations, we have set cloud size to six, a value we have found to achieve good performance, while still being well below that natural cloud size limit for our network. We have also found network performance in our experiments to be comparable under a fair number of cloud sizes.

5. Evaluation methodology

We performed extensive OPNET simulation studies to analyze the implications of routing metric staleness on DTN performance. We use results from these studies to illustrate the poor performance of routing protocols using stale routing information, as well as the potential for congestion that occurs when routing decisions are not cognizant of global network state. We also demonstrate the improvements of Cloud Routing over existing solutions in key network performance metrics.

The comparisons of Cloud Routing and existing work are based on a network configured to allow a fair comparison of these solutions. The mobility model used in our simulations supports both Cartesian distance-to-destination, used by Cloud Routing, and contact history information, used by PROPHET, as relevant metrics for bundle delivery. We model connection throughput in detail to show the effect of each solution’s cost of delivery on the level of network congestion and goodput. Finally, we collect our data after the network has reached a steady state. Results presented throughout this paper are based on the following simulation framework.

To allow performance comparisons between routing protocols based on position and contact history routing metrics, we based our simulations on a mobility model that supports both types of metrics. Developed by Hsu et al., the Time-Variant Community (TVC) mobility model used in our simulations restricts node mobility to predefined communities, or areas of the network [7]. Nodes also switch communities periodically. Unlike random mobility, the

Table 1
Control channel functionality of evaluated routing solutions.

Routing solution	Bundle ACK	Contact Update	Position Info	Bundle Receipt
Epidemic	X			
PROPHET	X			
ParaNets PROPHET	X	X		
Cloud Rank Reordering	X		X	X
Connection Utility Reordering	X		X	
Combined Cloud Reordering	X		X	X

TVC model allows nodes to build accurate contact history information to be used by protocols like PROPHET. Because TVC was developed to mimic user mobility observed in CRAWDDAD traces, node mobility generated by TVC is expected to represent real user behavior. We configure the rate of node mobility to driving speeds, which assures a high frequency of node contacts.

Traffic in our simulations is driven by each node injecting bundles into the network at a specified rate. To prevent the network from being gridlocked with an ever increasing amount of data, we set a bundle time-to-live (TTL) interval after which bundles that have not been delivered are dropped. We base the TTL assumption on the fact that there is such a thing as receiving data too late, even in a DTN, and that the utility of the network as perceived by users depends on receiving data within a “reasonable” amount of time.

There also is an important interaction between our traffic and mobility settings. To assure bundles are not created to destinations within the well-connected community, the minimum bundle range needs to be larger than the maximum community dimension. Creating bundles for destinations only outside the community allows us to measure the efficiency of opportunistic forwarding in a sparse mobile network, rather than in the denser communities.

Network congestion occurs in DTNs when the amount of data queued during a node contact exceeds that contact’s capacity. To model congestion created by different routing protocols, we simulate node connection throughput using TCP Reno. TCP allows us to accurately model connection throughput as affected by a node’s contact time and transmission rate, in turn determined by the number of contending nodes at the MAC layer. Modeling actual connection throughput allows us to accurately gauge the effects of DTN congestion on the key metrics of network goodput and bundle delay.

Table 1 summarizes the control channel traffic employed by each routing solution. All solutions send bundle acknowledgments on the control channel. Except for the experiments that vary the availability of the ParaNets control channel, where bundle acknowledgments may be disseminated opportunistically, we assume that the source node learns of a bundle delivery as soon as a bundle reaches its destination. This assumption allows us to compare performance gains of Cloud Routing to protocols like Epidemic Routing and PROPHET, without these protocols being unfairly penalized with delays of opportunistic propagation of acknowledgments.

ParaNets PROPHET uses the control channel to send node encounter information. We configure PROPHET and ParaNets PROPHET with the parameters recommended by Lindgren [3]. Additionally, ParaNets PROPHET, described in Section 3.1, has an additional ParaNets update interval parameter. The Cloud Routing solution sends periodically position information used to calculate the distance to destination routing metric. Bundle progress receipts, used in cloud membership bookkeeping, are sent whenever a new bundle copy is created. Cloud Routing configuration parameters are discussed in Section 4. Table 2 lists the key parameters used in our simulations.

Table 2
Simulation parameters.

	Parameter	Value
Network	Simulation time	2 h
	Data collection start time	1 h
	Network area	8 by 8 km
	Network size	100 nodes
	Transmission range	250 m
	PHY data rate	54 Mbps
	Node connection protocol	TCP Reno
	ParaNets control channel	0–100% available
Traffic Model	Bundle size	1 MB
	Bundle creation rate	10–500/h/node
	Bundle TTL	30 min
	Minimum bundle range	1 km
Mobility	Mobility model	TVC
	Number of communities	4
	Community size	400 by 400 m
	Speed	30–60 mph
PROPHET	P_{init}	0.75
	β	0.25
	γ	0.99
	ParaNets update interval	30 s
Cloud Routing	Neighbor beacon interval	1 s
	Position update interval	1 s
	Bundle buffer size	Unlimited
	Reordering buckets	1&3
	Cloud size	6

Finally, to assure meaningful network performance measurement, statistics for our simulations are collected after network steady state has been reached. Nodes start the simulation carrying only the bundles they themselves generate. As the simulation progresses, node buffers contain an increasing number of bundles being forwarded on behalf of other nodes. Eventually some bundles’ age exceeds the specified TTL and they are dropped. During this process, key metrics of network goodput and bundle delay are unstable due to buffer length fluctuations. These metrics, however, do stabilize in our simulations sometime between one and two TTL intervals after simulation start. All the values presented in our figures are averages obtained from relevant measurements collected after the network steady state has been reached.

6. Evaluation

We use simulation of DTNs to show the effectiveness of Cloud Routing at leveraging up-to-date global network state information to improve network goodput and lower bundle delay. We also quantify the cost of using the ParaNets control channel. We compare network goodput and bundle delay performance of our scheme to Epidemic Routing, standard PROPHET, and ParaNets PROPHET described earlier. We also show the impact on network performance of limited ParaNets control channel availability.

6.1. Network goodput

The traditional DTN metric used to evaluate network performance is cost of bundle delivery, measured as the number of bundle copies needed to effect delivery. However, we believe network goodput is more representative of network performance from a user’s point of view. Because limited network resources are shared by competing flows, we have observed that delivery cost decreases under high traffic load. This phenomenon is misleading of network performance since congestion at higher loads hampers network goodput and increases bundle delay in spite of the lower cost of delivery. In the extreme, congestion can be so high that data can only be delivered by a direct contact of source and destination nodes, resulting in a delivery cost of only one. We measure good-

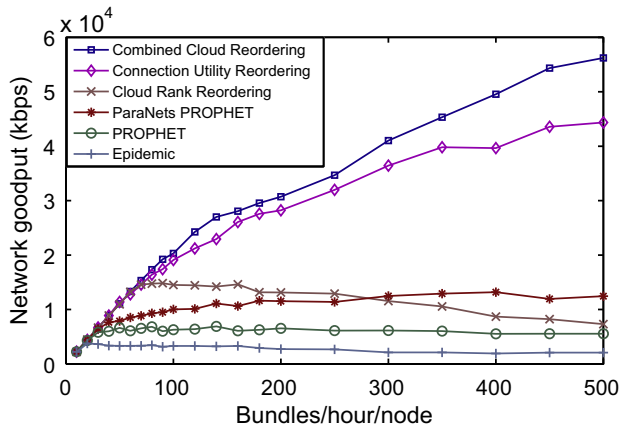


Fig. 7. Mean network bundle goodput versus network load.

put as the amount of bundle data in kbps acknowledged at the source.

Fig. 7 shows the mean network goodput versus traffic load. An increase in network load results in increased network goodput as more data is being created and delivered. This trend holds true until the network becomes congested. Different solutions reach this point of congestion at different network loads, corresponding to different maximum goodput.

Epidemic Routing becomes congested very early, as compared to other solutions, at twenty bundles/hour/node. The PROPHET solutions prove more resilient to congestion. We observe that ParaNets PROPHET outperforms standard PROPHET across the network load range. This difference is a testimony to the benefit of using up-to-date routing information.

Cloud Rank Reordering is able to achieve higher total goodput than PROPHET, but does not scale as well. We attribute the decrease of Cloud Rank Reordering goodput at higher loads to the globally aware routing decisions, which means that forward progress is made only by a small number of highest ranked cloud members. In a congested network, Cloud Rank Reordering suffers from the problems of single copy routing, which has been shown to perform poorly in DTNs [17].

Finally, we observe the significant goodput gains for Connection Utility Reordering and Combined Cloud Reordering. Based on their differences, especially at high loads, we believe that distance-to-destination is a good DTN routing metric where node motion is not restricted to predefined paths or locations. We also observe that Combined Cloud Reordering achieves higher goodput than Connection Utility Reordering alone. We believe this result is due to Cloud Rank Reordering being a more effective method of assuring forwarding fairness than fair shortest distance order, which allows Combined Cloud Reordering to take full advantage of greedy distance based routing for bundles below the specified cloud size, while avoiding range inversions caused by traffic far from its destination. Overall, Cloud Routing achieves a factor of 8 improvement in goodput over PROPHET and a factor of 15 over Epidemic Routing.

The goodput trends in Fig. 7 are representative of networks with lower node densities as well. Lower network density can affect the shape of the traffic “cloud,” but not the effectiveness of Cloud Routing. Under lower network density, the nodes carrying bundles comprising the traffic cloud may traverse the network without meeting other nodes and forwarding the bundle. As a result, the traffic cloud can become disjoint in the sense that nodes not carrying cloud member bundles are located amid the traffic cloud. However, Cloud Routing is still able to prioritize bundle transmissions

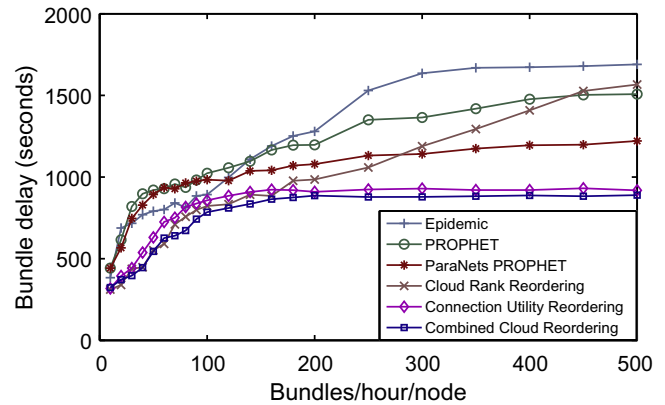


Fig. 8. Mean bundle delay versus network load.

during node contacts, as well as prevent the propagation of strayed bundles, regardless of traffic cloud cohesion. In general lower density networks offer fewer contact opportunities and reach lower network goodput and higher delay.

6.2. Bundle delay

Fig. 8 presents mean bundle delay as affected by network traffic load. Bundle delay is measured as the time between bundle creation and reception of a delivery acknowledgment by the source node. As traffic load increases, so does bundle delay. The increase in bundle delay is due to bundles being stalled in longer queues and not being transmitted during limited node contact time. This increase in delay, however, is capped by the thirty minute bundle TTL. The Cloud Routing solutions effectively reduce bundle delay by either reducing the impact of high ranking cloud members in Cloud Rank Reordering, or forwarding higher utility traffic first in Connection Utility Reordering.

The three Cloud Routing mechanisms, which all use distance-to-destination as their routing metric, are more effective at reducing bundle delay than contact history based solutions. An exception to this is Cloud Rank Reordering, which performs well under low loads, but experiences delays higher than the PROPHET solutions at higher loads. Again we believe this is due to the single copy routing problem experienced by the lowest-cloud-rank-first transmission ordering under high network load.

The ParaNets PROPHET and standard PROPHET solutions outperform Epidemic Routing delay at high loads. While Epidemic Routing is widely assumed to be unbeatable in terms of delay, that assumption does not hold in a highly loaded DTN, where the number of bundle copies created can prevent forward progress.

Finally, Combined Cloud Reordering achieves the lowest delay. The difference between Combined Cloud Reordering and Connection Utility Reordering is not as pronounced as in the case of goodput. We believe the smaller difference between Combined Cloud Reordering and Connection Utility Reordering in terms of delay is due to the thirty minute maximum delay determined by TTL. The delivery times of both solutions are interspersed in the interval defined by TTL, which, in practice, translates to the mean delay being close to half of the TTL for both Combined Cloud Reordering and Connection Utility Reordering.

6.3. Delivery rate

A metric often considered in the DTN routing arena is bundle delivery rate. Table 3 presents the delivery rate achieved by each routing solution under that solution’s highest goodput. While high-

Table 3
Goodput measurement points for percent enabled graphs.

Routing solution	Max goodput load (bundles/hour/node)	Delivery rate (percent)
Epidemic	20	83.4%
PROPHET	140	22.1%
ParaNets PROPHET	400	14.8%
Cloud Rank Reordering	90	71.7%
Connection Utility Reordering	500	40.0%
Combined Cloud Reordering	500	50.2%

er delivery rates can be achieved for each solution at lower loads, such as 100% for all solutions at the bundle creation rate of ten bundles/hour/node, we believe that measuring delivery rate at the highest goodput is most indicative of a routing solution's performance.

We observe that Epidemic Routing has the highest delivery rate, though the result is somewhat misleading because of the low corresponding network load. The PROPHET solutions have low delivery rates, but achieve a maximum goodput value that is much higher than that of Epidemic Routing.

Cloud Rank Reordering is able to achieve a high delivery rate but low corresponding goodput. We believe this result is due to Cloud Rank Reordering achieving fairness by equalization of cloud size in the network. Cloud Rank Reordering achieves high delivery rates, but at the cost of keeping all traffic in cloud size lockstep and limiting goodput. Connection Utility Reordering achieves a low delivery rate, but high goodput. We believe this low delivery rate is caused by range inversions, which limit the performance of the fair distance reordering mechanism. Finally, Combined Cloud Reordering achieves both high goodput and high delivery rate. We limit the bundle creation rate in our simulations at 500 bundles/hour/node as this is the point, at which Combined Cloud Reordering can deliver just over half the data injected into the network.

6.4. Primary network/control channel overhead

Our final result for a fully ParaNets enabled network examines the control channel overhead as it relates to goodput. Fig. 9 shows the mean ratio of goodput to ParaNets control traffic graphed on a log scale. We measure control channel usage as the mean kbps of data transmitted on that channel.

At higher loads, the usage ratio separates routing solutions into three groups. The highest usage ratio is achieved by Epidemic Routing and standard PROPHET, both of which use the ParaNets

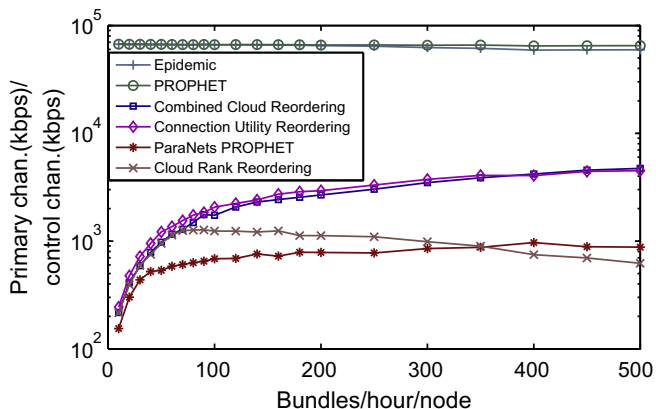


Fig. 9. Mean primary ParaNets kbps delivered for kbps of control channel used versus network load.

control channel only for bundle acknowledgments. While achieving a high ratio, these solutions may not be attractive candidates for deployment due to their low goodput.

The lowest usage ratio is achieved by Cloud Rank Reordering and ParaNets PROPHET. These solutions rely heavily on network state dissemination over the control channel, without the high goodput to offset that expense. This result shows the importance of good protocol design and adhering to the goal of limited use of the control channel.

Finally, Connection Utility Reordering and Combined Cloud Reordering strike a balance between high primary network goodput and control channel usage. The greater gain of Connection Utility Reordering and Combined Cloud Reordering is achieved by Combined Cloud Reordering with 4720 kbps of primary network goodput for every one kbps of control channel used. We believe this usage ratio makes Cloud Routing a very attractive solution for off-loading data transfers from end-to-end networks where an opportunistic channel is available. Instead of using the end-to-end infrastructure to forward bulk data directly between clients, a significant reduction in end-to-end infrastructure load can be achieved by a Cloud Routing integration of the end-to-end channel and opportunistic node contacts. Indeed, we believe Cloud Routing offers a more attractive end-to-end service than forwarding data separately on the end-to-end and opportunistic channels.

In addition to control channel load, it is also important to consider the number of control channel messages required by each routing solution. As presented in Table 1, Cloud Routing sends periodic position information, bundle progress, and acknowledgment packets. Nodes in our scenario send one position packet every second, which requires each node to process a hundred of these packets per second. Position update processing requires an insert into a node position hash table and is not computationally intensive. Additionally, an average total of 27.51 bundles are exchanged in the network every second. The forwarding of these bundles creates 27.51 control messages, 6.86 of which are bundle acknowledgments. Similarly, the processing of these messages is not computationally intensive.

Network scenarios where a larger number of bundles can be exchanged, exhibit a corresponding increase in the number of control messages. Cloud Routing control messages are small and can be easily aggregated reducing their number. Message aggregation may increase control traffic delay, which could potentially impact bundle delay and increase routing metric staleness. However, in the ParaNets network scenario the difference in delay between control channel and opportunistic propagation is so large that even the introduction of control traffic aggregation delay is not likely to be significant in the timescales of the opportunistic propagation and, as such, is not likely to noticeably impact end-to-end network performance. Another effective technique for reducing the number of control messages is filtering, where only messages meeting certain criteria are transmitted. We plan on looking at both aggregation and filtering as part of future work.

6.5. Limited paramets deployment

We also study the performance of routing solutions described in this paper under a limited ParaNets control channel availability. While we have shown that significant network performance improvement can be achieved by leveraging an end-to-end connection, such connections may not be available to all nodes, or to all nodes at all times. To retain the generality of DTNs, Cloud Routing is able to take advantage of additional infrastructure to a degree that such infrastructure is available. To assess the impact of limited ParaNets deployment, we repeat the experiments described in the previous sections while varying the percentage of

ParaNets control channel transmissions each node receives or transmits. Thus, if the availability of the ParaNets control channel is 10%, each node will be able to receive and transmit only one out of ten uniformly distributed transmissions.

In our simulations, nodes temporarily without access to the ParaNets control channel can neither send nor receive ParaNets updates. They can however take advantage of their neighbors who have access to the ParaNets control channel. Node position, bundle cloud membership, and contact history information is exchanged along with bundle metadata during a node contact. We exchange this metadata before any bundle is sent. In practice however, the size of the exchanged metadata is negligible with respect to contact throughput.

We make network performance measurements for limited ParaNets enabled networks at the highest goodput value for each routing solution. Since solutions presented in this paper draw their performance benefits from control information exchanged on the ParaNets control channel, choosing the best performance point allows us to assess the maximum impact of limiting the end-to-end functionality. Table 3 shows the network load corresponding to the best goodput for each solution. Measurements in the following sections are based on these loads.

6.5.1. Network goodput

Fig. 10 shows the degradation of network goodput performance as affected by a decrease in ParaNets control channel availability. As the intermittency of the ParaNets control channel increases, all of the routing solutions experience a similar gradual decline in network goodput. There are, however, a couple interesting features in Fig. 10.

The Cloud Routing solutions experience a more pronounced reduction in goodput as the ParaNets channel availability decreases from 20% to 0%. We believe this change is due to the volatility of node position as a routing metric. Node positions change rapidly and cached data at individual nodes can quickly become out of date if they are not refreshed regularly. As the availability of the control channel drops below 20%, position information can be no longer disseminated quickly enough to allow accurate calculation of the distance-to-destination routing metric. Since contact history based delivery probabilities are less likely to change over time, the PROPHET solutions' performance is more resilient to changes in control channel availability.

The second interesting feature of Fig. 10 is the steep reduction in goodput of the ParaNets PROPHET solution at around 50% control channel availability. In ParaNets PROPHET, there are two mechanism for the dissemination of contact history information, one using the control channel, and one relying on opportunistic

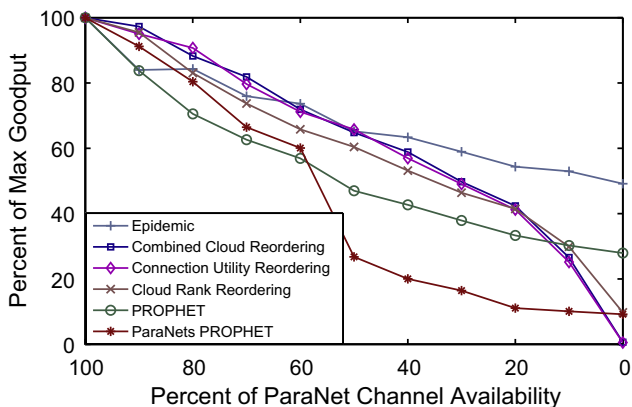


Fig. 10. Mean percent change of maximum network bundle goodput versus ParaNets control channel availability.

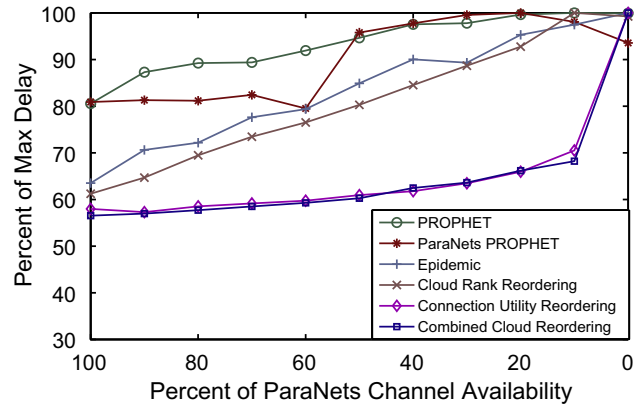


Fig. 11. Mean percent change of maximum delay versus ParaNets control channel availability.

contacts. These mechanism disseminate contact history information representing slightly different states of the network, the first one less stale than the other. As more than 50% of contact history updates are not transmitted or received, the stale state is used predominantly in routing metric calculation. Thus, at 50% control channel availability, ParaNets PROPHET loses the benefits of up-to-date network state and reverts to essentially the standard PROPHET performance. In Fig. 10 the absolute performance of ParaNets PROPHET below 50% control channel availability is comparable to standard PROPHET performance.

We also observe that the degradation of performance of the Epidemic and standard PROPHET solutions due to limited availability of the control channel is less pronounced than for other routing solutions. The Epidemic and PROPHET solutions do not draw benefits from the control channel other than acknowledgment propagation, so when that functionality is not available the difference in the performance of these protocols is not profound.

The gradual decline in goodput of Combined Cloud Reordering is a result of Cloud Routing's robustness to ParaNets control channel intermittency. Cloud Routing is able to take advantage of a parallel channel to the degree such infrastructure is available. Thus, Cloud Routing is able to realize the benefits of an opportunistic routing protocol using an end-to-end connection without losing the generality of loose connectivity DTNs.

6.5.2. Bundle delay

Fig. 11 shows the change in bundle delay as affected by ParaNets control channel availability. We observe a general increase

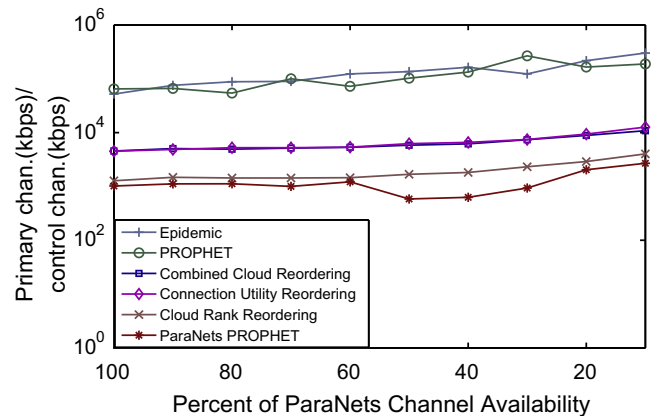


Fig. 12. Mean percent change of primary ParaNets kbps delivered for kbps of control channel used at maximum network goodput for each solution versus ParaNets control channel availability.

in bundle delay corresponding to a decrease in the availability of the ParaNets control channel. The ParaNets PROPHET solution experiences a less gradual increase in delay than standard PROPHET until the control channel availability drops below 50%. Again, we believe this is due to ParaNets PROPHET degenerating to standard PROPHET performance. We also observe that the Combined Cloud Reordering solution is the most resilient to ParaNets control channel unavailability. Cloud Routing is able to maintain a delay close to that of 100% control channel availability until that availability drops below 10%. Again, this result shows the general applicability of Cloud Routing even under intermittent connectivity to the end-to-end infrastructure.

6.5.3. Primary network/control channel ratio

Finally, Fig. 12 shows the ratio of primary network goodput to ParaNets control channel usage as affected by increasing intermittency of the ParaNets control channel. We do not record the ratio for 0% ParaNets availability, since control channel usage in these scenarios is zero.

In Fig. 12 we observe that the primary/control channel usage ratio remains relatively stable for all solutions regardless of the additional ParaNets infrastructure. This result implies that the usage of the control channel is primarily created in response to opportunistic traffic. As opportunistic throughput decreases in response to a lower efficiency of bundle forwarding, so does the amount of control traffic overhead required from the control channel. Data in Fig. 12 implies that primary network goodput to control channel usage ratios demonstrated in Section 6.4 remain true even under intermittent availability of the control channel.

7. Conclusions and future work

In this paper, we have presented Cloud Routing, a novel data forwarding solution for opportunistic networks. We have identified a number of challenges to achieving good opportunistic network performance stemming from routing metric staleness. We proposed Cloud Rank Reorder and Contact Utility Maximization to address these challenges. We have shown through simulation the performance benefits of these mechanisms, as well as their potential to achieve further performance increases by using them in conjunction with existing opportunistic routing solutions.

Based on the performance improvements achieved by Cloud Routing, at the negligible cost to the ParaNets control channel, we believe that whenever two or more networks with different characteristics are available, they should be treated as a ParaNets architecture and employ Cloud Routing to improve the joint performance of all channels.

In the future we would like to evaluate Cloud Routing under different ParaNets instantiations and routing metrics. One interesting area for future research is the design of protocols to disseminate control traffic where one-to-all reachability cannot be assumed, or exists only by traversing a hierarchical end-to-end infrastructure. In such scenarios, solutions for efficient management of bundle clouds using localized end-to-end communication are needed.

Finally we would like to explore other routing mechanism able to take advantage of up-to-date network state information to make globally cognizant routing decisions. Such mechanism could exist not only in the DTN arena, but also within the context of wireless

mesh networks or VANETs. Indeed, Ott et al. has already demonstrated the benefits of applying DTN forwarding protocols in well-connected networks [18]. We believe the use of up-to-date network state disseminated over a control channel opens new areas of inquiry for a number of networking problems, including routing, traffic engineering, or content dissemination, to name just a few.

Acknowledgement

We thank Wei-Jen Hsu for making available the code for TVC, as well as an early version of his work on Profile-Cast [7].

References

- [1] A. Vahdat, D. Becker, Epidemic routing for partially connected ad hoc networks, Tech. Rep., Duke University, 2000.
- [2] K. Harras, K.C. Almeroth, E.M. Belding-Royer, Delay tolerant mobile networks (DTMNs): controlled flooding schemes in sparse mobile networks, in: Proceedings of the Seventh IFIP Networking Conference, 2005.
- [3] A. Lindgren, A. Doria, O. Schelén, Probabilistic routing in intermittently connected networks, SIGMOBILE Mobile Computing Communications Review 7 (3) (2003) 19–20.
- [4] M. Musolesi, S. Hailes, C. Mascolo, Adaptive routing for intermittently connected mobile ad hoc networks, in: WOWMOM '05: Proceedings of the Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks, 2005.
- [5] J. Leguay, T. Friedman, V. Conan, Evaluating mobility pattern space routing for DTNs, in: INFOCOM '06: Proceedings of the 25th IEEE International Conference on Computer Communications, 2006.
- [6] B. Burns, O. Brock, B.N. Levine, MV routing and capacity building in disruption tolerant networks, in: INFOCOM '05: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, 2005.
- [7] W. Hsu, D. Dutta, A. Helmy, Profile-Cast: behavior-aware mobile networking, in: WCNC '08: IEEE Wireless Communications and Networking Conference, 2008.
- [8] S. Surana, R. Patra, E. Brewer, Simplifying fault diagnosis in locally managed rural WiFi networks, in: NSDR '07: Proceedings of the 2007 workshop on Networked systems for developing regions, 2007.
- [9] K.A. Harras, M.P. Wittie, K.C. Almeroth, E.M. Belding, ParaNets: a parallel network architecture for challenged networks, in: HotMobile '07: Eighth IEEE Workshop on Mobile Computing Systems and Applications, 2007.
- [10] W.H. Yuen, H. Schulzrinne, Performance evaluation of time-based and hop-based TTL schemes in partially connected ad hoc networks, in: ICC '06: IEEE International Conference on Communications, 2006.
- [11] S. Jain, K. Fall, R. Patra, Routing in a delay tolerant network, in: SIGCOMM '04: Proceedings of the 2004 Conference on Applications, technologies, architectures, and protocols for computer communications, 2004.
- [12] R. Chandra, P. Bahl, MultiNet: connecting to multiple IEEE 802.11 networks using a single wireless card, in: INFOCOM '04: Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, 2004.
- [13] L. Song, D.F. Kotz, Evaluating opportunistic routing protocols with large realistic contact traces, in: CHANTS '07: Proceedings of the 2007 SIGCOMM workshop on challenged networks, 2007.
- [14] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, H. Weiss, Delay-Tolerant Networking Architecture, IRTF RFC 4838 (Informational) (April 2007).
- [15] Q. Li, D. Rus, Communication in disconnected ad hoc networks using message relay, Journal of Parallel and Distributed Computing 63 (1) (2003) 75–86.
- [16] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, R. Krishnan, Prioritized epidemic routing for opportunistic networks, in: MobiOpp '07: Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking, 2007.
- [17] T. Spyropoulos, K. Psounis, C.S. Raghavendra, Single-copy routing in intermittently connected mobile networks, in: SECON '04: First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004.
- [18] J. Ott, D. Kutscher, C. Dwertmann, Integrating DTN and MANET routing, in: CHANTS '06: Proceedings of the 2006 SIGCOMM workshop on Challenged networks, 2006.