CARPOOL: Extending Free Internet Access over DTN in Urban Environments

Ioannis Komnios and Vassilis Tsaoussidis
Space Internetworking Center
Dept. of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, 67100, Greece
{ikomnios, vtsaousi}@ee.duth.gr

ABSTRACT

In order to address the challenge of digital exclusion, we introduce an access method based on message ferrying that enables free delay-tolerant Internet access to all. Targeting an urban scenario, where means of public transport, such as buses and trams, follow a predefined route and schedule, we utilise *a priori* knowledge about their current location to extend Internet access provided by hotspots to users and areas that are not typically covered. In this paper, we explore how the deployment of DTN-capable nodes can extend free Internet coverage in metropolitan areas, we depict the inefficiency of existing DTN routing protocols in highly-dense environments and we propose Connectivity plAn Routing PrOtOcOL (CARPOOL), a reference routing protocol to support our claims. CARPOOL exploits the connectivity plan of public transportation, achieving high delivery ratio with minimum overhead.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design – *Store and forward networks*, C.2.2 [**Computer-Communication Networks**]: Network Protocols – *Routing protocols*

Keywords

Delay Tolerant Networking; DTN routing; Free Internet

1. INTRODUCTION

Internet has now become a critical infrastructure for the society with its availability levels increasing and its traffic volume constantly growing. In a constantly evolving and expanding digital world, however, geographical isolation and socio-economic restrictions pose barriers to the invasion of the Internet to all parts of the society. In an effort to provide Internet access to all members of the society, several economic models, such as providing restricted Internet access during night at a lower price, have been proposed in the recent past. Nonetheless, these models are not affordable to all, leaving certain members of the society with the only alternative of using random hotspots when available.

In this paper, we focus on metropolitan environments with an ultimate goal to extend free delay-tolerant Internet access to the under-privileged society that is currently excluded from today's digital world. To achieve that, we extend the existing free Internet access provided by public hotspots that are usually scattered around a city by deploying DTN nodes [3, 7] both on typical means of public transport (ferries), such as buses and trams, and their corresponding stops. Offline DTN gateways located near ferry stops collect Internet access requests from end-users in that area and DTN ferries act as relays between offline gateways or designated gateways that have access to the Internet and are capable of handling such requests. For this reason, we have designed and evaluated CARPOOL routing protocol that utilizes *a priori* knowledge of future contacts between DTN ferries and stationary DTN gateways. Through simulations we have identified that existing DTN routing solutions underperform in such dense environments due to their associated high overhead and their excessive energy needs and thus fail to guarantee some level of service.

Our novelty lies in the utilization of *a priori* knowledge of contacts between gateways and ferries, in an effort to achieve high delivery ratio with minimum overhead. The proposed solution: (i) constitutes an easy-to-deploy access method that exploits information regarding the schedule of the ferries, which is already available in all major cities worldwide, (ii) provides free delay-tolerant access to the Internet for everyone, and (iii) has energy-efficient design that delegates all expensive computing operations to gateways with increased computing and power capacity.

The remainder of this paper is structured as follows: in Section 2, we describe the proposed access method along with CARPOOL routing protocol, while in Section 3 we present our initial simulation results. We conclude in Section 4, where we discuss also our future plans.

2. ARCHITECTURE OVERVIEW

Our access model consists of two major components: *DTN gateways* that are responsible for handling requests from end-users within their radius and *DTN ferries* that are responsible for transferring messages (bundles) across the gateways. We intentionally delegated all computational tasks to the gateways, since we assume that DTN ferries have restricted energy and computational capabilities. Typically, the travel plan of buses, trams and trains is predefined and only minor delays can occur. Therefore, in our model we assume that all gateways have global knowledge of the connectivity plan. Of course, in case of a major delay, the updated traffic schedule is flooded into the network through a central administrative node. Figure 1 contains a sample topology corresponding to our model.

DTN gateways are resource-capable fixed nodes located near ferry stops. We assume that certain gateways have access to the Internet through a hotspot that exists in the area (*online DTN gateways*), while the majority is offline. All gateways have effectively enough storage to store bundles from several end-users and are equipped with network interfaces for data exchange with the mobile devices of the end-users and the DTN ferries. Once an

LCDNet'13, September 30, 2013, Miami, Florida, USA.
ACM 978-1-4503-2365-9/13/09.
http://dx.doi.org/10.1145/2502880.2502891

end-user device discovers a DTN gateway in its radius, a request to/from the Internet is transferred from/to the relevant application.

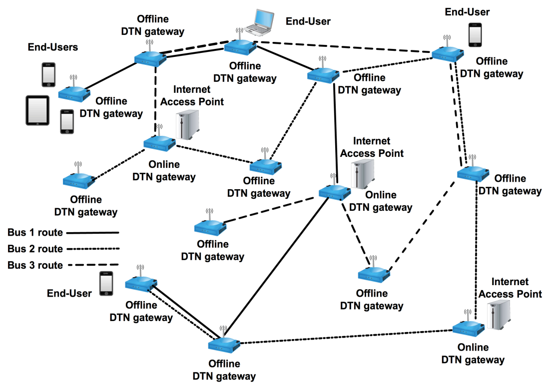

Figure 1. Sample topology

When a bundle is received by an offline gateway, CARPOOL protocol calculates valid paths between this gateway and online gateways, based on the connectivity plan, and selects a path that achieves earliest bundle delivery to an online gateway. The search of valid paths starts from the destination, moving towards the source in a hop-by-hop manner. In particular, all gateways hold the list of online gateways and the overall connectivity plan, which includes all contacts between gateways and ferries along with the scheduled start of each contact. The entries of the connectivity table for each ferry have the following 3-tuple structure (*GatewayID*, *FerryID*, *ContactTime*). The values required as input to the algorithm are *PreviousGateway* and *NewArrivalTime*. Initially, *PreviousGateway* is set to the ID of an online gateway and *NewArrivalTime* equals to bundle creation time plus TTL. The current gateway first identifies all contacts in the overall connectivity table that satisfy the following requirements:

- *GatewayID* equals *PreviousGateway* and
- *ContactTime* is greater than current time and less than the latest arrival time (*NewArrivalTime*).

For each of the aforementioned contacts, we store a set of 3-tuples: the contact itself and the exact previous contact (in terms of time) between the same ferry and another gateway. When the previous gateway that this ferry has traversed becomes the current gateway, we have identified a direct contact, where the current gateway is only one hop away from an online gateway. Otherwise, the algorithm re-executes using as input the *GatewayID* and the *ContactTime* of the previous contact. Thus, we now search for valid contacts that are two hops away from an online gateway. This process is continued until a path is found. Once a path has been discovered, the *GatewayID* of the next gateway on the path, the *FerryID* of the ferry that will transfer the bundle and the *TimeToForward* that corresponds to the time that the bundle will be forwarded are added to the header of the bundle; the bundle is then stored in the buffer of the gateway, until a connection between the gateway and *this* ferry exists.

In essence, instead of storing the end-to-end path through the network, we only store the next gateway on the path. If the full path to an online gateway was stored, the time-shift of an intermediate contact even for a few seconds would lead to a significant delay, let alone bundle expiration. The proposed method reacts to changes or bias to the initial contact plan by re-evaluating the best route for a bundle at every gateway. Moreover, CARPOOL re-calculates the path for all bundles in the network with *TimeToForward* greater than current time aligned within a fixed threshold, set according to the arrival time of the next ferry to this gateway. This cancels the impact of the schedule deviation and typically suffices to accommodate most schedule drifts.

When a connection is up, the gateway uploads bundles waiting to be forwarded through that ferry and downloads bundles from the ferry that are destined to that gateway. When an online gateway receives a new bundle, the bundle is forwarded to the receiving application through the Internet. When an offline gateway receives a bundle, the algorithm is re-executed and the corresponding fields in the header of the bundle are updated. In order to reduce the complexity and the associated computational overhead of our algorithm, instead of identifying all possible paths and selecting one that achieves earliest delivery, we first sort valid contacts to an online gateway starting from the earliest, prior to applying our selection algorithm. This way, we need not calculate all paths from the current gateway to all online gateways; instead we simply select the first plausible path to an online gateway, which is also a path that guarantees earliest delivery. Our access model faces two limitations: the finite buffer size of gateways and ferries, as well as the small window of communication opportunities among them. In the event that this window does not suffice for all bundles to be delivered at the gateway or the ferry respectively, the path for the unserviced bundles is re-calculated.

Unlike upload operations, downloading data from the Internet requires an additional publish/subscribe session layer (e.g. similar to the one presented in [4]), in order to allow for applications such as RSS content distribution and web access over DTN. The proposed routing protocol can function efficiently in both cases.

Our approach shares the philosophy of Contact Graph Routing [2], which is the most prominent routing solution in space internetworking. Similar to prescheduled contacts between ferries and gateways, Contact Graph Routing extracts a path for space data transmission utilizing *a priori* knowledge of contacts between space assets. It should be noted that CARPOOL is not a replication scheme; only a single copy of each bundle exists in the network at any given time, keeping overhead to minimum.

3. EVALUATION

The CARPOOL protocol has been implemented and evaluated using the Opportunistic Network Environment (ONE) simulator [5]. We study the impact of increased traffic load on its performance comparatively with four widely-used routing protocols, namely, Epidemic [9], PRoPHET [6], binary Spray-and-Wait with 10 message copies [8] and MaxProp [1]. Our aim is not to compare CARPOOL with DTN routing solutions that have no or partial knowledge of the network, but to highlight the inefficiency of these protocols in dense urban environments. Prior to the actual simulation, adequate training time was given to protocols that need initialization (i.e. PRoPHET and MaxProp).

Initially, we created the connectivity plan for the entire simulation using as input: (1) the ID and the coordinates of each gateway, (2) the ID and the speed of each ferry, along with the gateways on the path of the ferry in the order it traverses them, (3) the waiting time at each stop and (4) the start times of each ferry. We assume that all ferries follow the reverse path once they reach their destination. All gateways become aware of the connectivity plan. We selected a topology for our simulations that corresponds approximately to an abstraction of the transport service of Thessaloniki, Greece, that includes both the city center and the suburbs. In total, our simulation environment covers an area of approximately 100 km^2 that includes 106 offline gateways (i.e. bus stops) and 15 online gateways. Our scenarios follow 60 ferries (i.e. buses) travelling on 20 routes. The speed of the ferries ranges from 18km/h to 50km/h. All gateways and ferries are equipped with 2GB storage size and wireless network cards supporting a 10Mbps data rate and 50m communication radius. The overall duration of every simulation is 48 hours, including a sufficient warm-up period for protocols to initialize themselves. The traffic load varies from 2500 to 50000 bundles per 12 hours. Bundle size ranges from 500kB to 2MB. Given the delay-tolerant nature of the applications, bundle TTL is set to 20h, sufficiently large to accommodate all communication attempts by all protocols.

Results in Figure 2 illustrate the delivery ratio of the five routing schemes for increasing traffic load. We notice that in low traffic load conditions (less than 20000 bundles in 12h) only CARPOOL and MaxProp manage to deliver all bundles; the three other protocols fail to achieve maximum delivery ratio. CARPOOL achieves increased delivery ratio, since contacts between gateways and ferries are known *a priori* and in the event of unexpected delays, new paths to online gateways are being re-discovered. It should be noted, however, that in contrast to other protocols, our current version of CARPOOL does not exploit short contacts between ferries. Through its scheduling tactics at the gateways and the ferries, MaxProp achieves high delivery ratio. Epidemic routing suffers from its excessive overhead and experiences worst performance. In high traffic load conditions (more than 25000 bundles in 12h) the delivery ratio of all protocols decreases when traffic load increases. However, CARPOOL performs significantly better than all other protocols, managing to successfully deliver 82% of the created bundles even in the worst scenario, despite heavy congestion. This allows for an optimistic future for CARPOOL; unlike other protocols, CARPOOL's delivery ratio, even in worst-case scenarios, suffices in its own right to guarantee some level of service. A user may feel confident that even if one attempt fails, most likely this will not be repeated. As far as overhead is concerned, CARPOOL presents minimum overhead, since there exists only one copy of each bundle in the network at any given time, minimizing the energy consumption of battery-powered devices, but also allowing for better bandwidth utilisation, which practically means our network can accommodate more users.

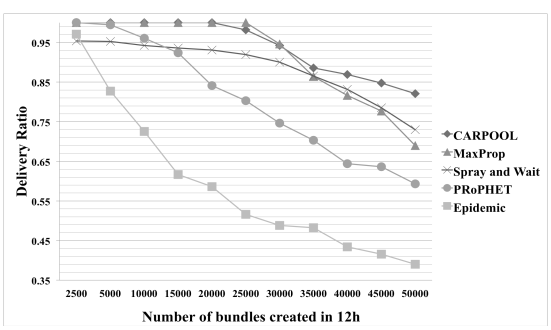

Figure 2. Delivery ratio for increasing traffic load

4. CONCLUSIONS AND FUTURE WORK

We have proposed an access model for urban environments suitable to extend Internet access opportunities to free users. We have employed delay-tolerant networking properties into CARPOOL routing protocol and have shown that an acceptable level of service can be provided. This was justified by our results of delivery ratio (i.e. service probability) and overhead (i.e. potential to accommodate more users). Our work constitutes a first attempt to promote free Internet to all, relied on the potential to exploit the transportation schedules of large cities and was further enhanced with dynamic decisions to avoid schedule deviations. However, our work is not confined only within the limitation of fixed schedules. Ferries that extend geographically Internet coverage to isolated regions, and mechanisms to cope with optimal routes when congestion evolves, will be grafted in the future. Moreover, the exploitation of contact opportunities among ferries, as well as the recalculation of the path at the ferries, if needed, could result in enhanced performance. Complexity and reliability issues for increased network size and overloaded network conditions will also be investigated.

5. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme ([FP7/2007-2013, FP7-REGPOT-2010-1, SP4 Capacities, Coordination and Support Actions) under Grant Agreement n° 264226 (Project title: Space Internetworking Center - SPICE). This paper reflects only the authors' views and the Community is not liable for any use that may be made of the information contained therein.

6. REFERENCES

[1] Burgess, J., Gallagher, B., and Levine, B.N. 2006. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *Proceedings of IEEE Infocom.* DOI= http://dx.doi.org/10.1109/INFOCOM.2006.228.

[2] Burleigh, S. 2010. Contact Graph Routing. *Internet Draft.*

[3] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, L., Scott, K., Fall, K., and Weiss, H. 2007. Delay–Tolerant Networking Architecture*. RFC4838.*

[4] Demmer, M., and Fall, K. 2009. The design and implementation of a session layer for delay-tolerant networks. *Computer Communications*, Vol. 32, Issue 16, 1724-1730. DOI= http://dx.doi.org/10.1016/j.comcom.2009.02.010.

[5] Keranen, A., Ott, J., and Karkkainen, T. 2009. The ONE simulator for DTN protocol evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques.* New York. DOI= http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5674.

[6] Lindgren, A., Doria, A., Davies, E., and Grasic, S. 2011. Probabilistic routing protocol for intermittently connected networks*. Internet Draft.*

[7] Scott, K., Burleigh, S. 2007. Bundle protocol specification. *RFC 5050.*

[8] Spyropoulos, T., Psounis, K., and Raghavendra, C. S. 2005. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proc. ACM SIGCOMM Workshop on Delay Tolerant Networking* (WDTN). DOI=http://dx.doi.org/10.1145/1080139.1080143.

[9] Vahdat, A., and Becker, D. 2000. Epidemic routing for partially connected ad hoc networks. *Technical Report CS-200006*. Duke University.