A System Architecture for Delay-Tolerant Content Distribution

(Invited Paper)

Martin May
Computer Engineering
and Networks Laboratory
ETH Zurich
8092 Zurich, Switzerland
Email: may@tik.ee.ethz.ch

Gunnar Karlsson
Ólafur Ragnar Helgasson
Laboratory for Comm. Networks
KTH, Royal Inst. of Tech.
100 44 Stockholm, Sweden
Email: {gk,olafur.helgason}@ee.kth.se

Vincent Lenders
Department of Electrical Engineering
Princeton University
Princeton, NJ 08544, USA
Email: vlenders@princeton.edu

***Abstract*—The goal of the system architecture presented herein is to have a wireless content distribution system which extends the coverage of classical local area wireless networks by enabling content transfers between the mobile nodes in an ad hoc manner. The ad hoc mode is important for extending the availability of contents beyond the reach of the cellular systems and for providing dissemination of contents generated in an ad hoc domain. The system may use any data-link layer that provides node-to-node connectivity; routing is carried out at the application layer by means of a solicitation protocol and addressing is relating to contents rather than to the identity and location of nodes. Consequently, the system does not require any routing protocol at the network layer. Our analysis provides results on the performance of a PodNet distribution system based on simulations of the system. Among other results, we show the existence of a virtual storage effect in PodNet, which results in resident content in areas where all nodes are transient.**

I. INTRODUCTION

Over the past few years, DTN has become an established and recognized research field. In the beginning, delay tolerant networks were networks that tolerated very long delays specifically in environments where communication interruptions are mostly predictable. With the emerging wireless mobile ad hoc networks also opportunistic networking schemes are discussed in the DTN community. In those networks however, the mobility patterns of the participating nodes often are not predictable and end-to-end connectivity between the nodes can not be maintained. As a result, DTNs have challenged many well established Internet design assumptions due to this lack of end-to-end connectivity. For example routing is strongly affected when end-to-end connectivity can not be assumed and a lot of effort has been pushed into the development of efficient routing schemes. The proposed schemes - an excellent overview is provided in [1] - cover flooding-based approaches (with inherently high overhead) as well as encounter-based approaches (with a minimal number of message exchanges, but without any guarantees for message delivery). Due to the unreliable and unpredictable nature of the node mobility all schemes aim at optimizing the tradeoff between overhead and delivery probability, while only the flooding schemes guarantee message delivery.

In this paper, we propose to draw the attention to a different communication method for opportunistic networks: broadcasting. Instead of aiming at point-to-point communication, we propose to rely on message broadcasting and develop content distribution applications. Such applications are very attractive and do not suffer from sub-optimal routing algorithms. While in [2] and [3] we introduced the concept of PodNet, a system architecture for delay tolerant content distribution, we discuss in this paper the general concept of receiver-driven broadcasting for opportunistic networks and the performance of such a system in an urban setting. In delay-tolerant or opportunistic networks, the availability and mobility of nodes in the system is highly unpredictable and it is thereby impossible to achieve any reliability guarantees. As a consequence, instead of relying on end-to-end semantics between a requesting node and a content provider, our dissemination mechanism relies on opportunistic content forwarding while abstaining from any routing substrate; content is routed implicitly through the combination of an application level receiver-driven solicitation protocol and the actual node mobility. Consequently, the system architecture does not include a network layer and implements the transport layer directly on top of the link layer.

There are two modes of public content distribution that are commonly used for mobile devices today. One is podcasting where the device downloads contents when docked to a computer with an internet connection; the other is the live streaming available in 3G cellular networks. The latter provides good coverage and thereby also continuous access to contents. The technological limit is the capacity in a cell, around 5 Mb/s, that quickly may become saturated if streaming becomes popular. In contrast, podcasting is enabled by the massive storage that is available in mobile devices. The distribution is limited to the contents that are available at the time of the download, and contents published thereafter remain inaccessible. Even further increase in storage will not alleviate this. Wireless internet access, via public IEEE 802.11 networks for instance, may provide more frequent downloading opportunities when the devices are on the move.

The system described herein, PodNet, is based on wireless communication and aims at increasing the opportunities for devices to obtain contents while being mobile. PodNet downloads content when docked or more generally when connected to the Internet, and it distributes contents opportunistically from node to node otherwise. This opportunistic node-to-node distribution is important for extending the availability of contents beyond the reach of the infrastructure, and it enables distribution of contents that are generated by the mobile nodes without a supporting infrastructure.

PodNet is particularly suited to distribute contents that are popular and delay-tolerant. Typical contents of that kind are:

- **Software updates:** As with most regular operating systems, mobile devices also has to be patched and updated regularly. The PodNet architecture could be used to distribute such updates to a large user group without relying on the costly 3G distribution systems. Note that security concerns have to addressed separately and are out of scope of this paper.
- **News distribution:** We envision that short news reports could be distributed. Imagine for example that small video clips of the goals in a football match are distributed along with the current score to interested people in an urban area.

One of the important properties of the proposed PodNet architecture is its openness: Anybody who wishes to publish contents (or to retrieve contents from the system) is free to do so. At the same time it is possible to restrict access to contents to closed user groups by means of digital rights managements (we do however not discuss this issue in this paper).

To efficiently distribute contents, PodNet classifies them into *channels*. A channel in PodNet is similar to a TV or radio channel and typically includes all contents related to one specific topic, e.g., weather news, rock music. The contents in a channel is provided by one or more producers. Channels allow the mobile nodes to more efficiently search and identify contents of their interest in a peer node. PodNet further supports interrupted transfer caused my mobility and allows nodes to resume downloads from the same or another peer in the system at the place where the transfer was interrupted.

The content distribution performance of PodNet is depending mainly on four characteristics: (i) the mobility pattern of the mobile nodes; (ii) the connection setup delay of two mobile devices; (iii) the content solicitation/caching strategy of the nodes, and (iv) the availability and locations of supporting network infrastructure, namely the WLAN access points. While (i) must be considered as given by the operational environment and (ii) is partly determined by the used link layer, PodNet aims at providing the best solutions for the parameters that can be changed in the design and the operations. Thus, we study the performance of the system in an urban-area setting where mobile nodes come and go in a city area. The speed distributions for mobility models are chosen to represent pedestrians; vehicular mobility is not captured though PodNet could be used to distribute contents between vehicles also. We investigate if content can become resident in areas without any fixed infrastructure. Our simulations show the existence of a virtual storage effect in PodNet: it is possible that individual content remains in an area in which there are no fixed nodes and the mobile nodes traverse the area without stopping.

II. Related Work

Of course, research on delay tolerant networking is very related to this work. We do not aim for unicast communication in our work but rather for content distribution. Multicast routing in DTN has been addressed in [4]. While the goal of our system is also to deliver data to a group of people, our approach is decoupled from any multicast semantics (such as group memberships, et cetera). The infostation concept is akin to our proposal and the paper studies means for avoiding exploitation of other nodes [5]. We differ in that we make the nodal exchanges governed by a protocol instead of a social contract between users.

The Delay Tolerant Network Research Group (DTNRG) [6] has proposed an architecture [7] to support messaging that may be used by delay tolerant applications. The architecture consists mainly of an overlay, called the bundle layer, above a network transport layer. This architecture differs from PodNet in many ways. First, PodNet does not implement a routing layer. PodNet does not even assigns identifiers to nodes but only to contents. Second, in our architecture, we do not aim at end-to-end reliability. Our assumption is that node mobility and availability is so unpredictable that any end-to-end semantic has little usefulness.

Leguay et al. [8] and Lindgren et al. [9] have studied different content distribution schemes using Bluetooth connectivity traces. Both works provide useful insights into the impact of human mobility on the opportunistic contacts in real environments. However, none of these works propose any specific system architecture for the content distribution.

TACO-DTN [10] is a content-based dissemination system for delay tolerant networks. It is implemented as a publish/subscribe system and was mainly designed to distribute temporal events whereas PodNet is implemented as a pure receiver-driven system and optimized for dissemination of high-throughput media. BlueTorrent [11] is an opportunistic file sharing application for Bluetooth enabled devices. The concept of distributing large files using small resumable atomic chunks is similar to our approach. However, BlueTorrent relies on Bluetooth whereas our proposal leverages any link-layer technologies. Furthermore, we propose to structure the data in the network into channels and rely on an entirely receiver-driven content dissemination protocol.

We show in [12] that delay-tolerant broadcasting between mobile nodes results in sufficiently high application level throughput even for streaming. This is the case in urban pedestrian areas with reasonably high densities of users, as well as in public transportation and in places where people gather occasionally (e.g., sport fields, shopping malls, recreational areas).

Content-based routing and the publish/subscribe paradigm have been used in the context of selective information dissemination on the Internet [13], [14], [15] and in sensor networks [16]. The conceptual idea is to have publishers acting as information providers, subscribers as information consumers, and a broker mechanism routes relevant publications to interested subscribers. Traditional pub/sub systems typically allow much more sophisticated information dissemination schemes than in our architecture. It is for example possible to steer the data flow towards particular nodes that are subscribed based on matching attributes values or even range queries. In contrast, the flow of contents in PodNet is determined by the interest of the users and their mobility, and as a benefit, our system does not require any network-layer routing.

III. PodNet System Architecture

This section describes the key design ideas of our delay tolerant content distribution architecture. PodNet has to deal with network properties that are different from the Internet and provides a new type of service, which requires a re-thinking of networking basics: While existing network architectures focus on addressing *nodes* and forward packets between such nodes, our system aims at addressing and disseminating *content*. Consequently, we do not rely on end-to-end semantics between a source and a destination, but we target a system where contents are requested and disseminated with the help of users and mobile nodes. As a result, sophisticated multi-hop communication protocols, where for example routes have to be build up and be maintained, are not necessary.

A. Content dissemination

Content distribution with PodNet is characterized by two basic design choices. First, we make only pairwise associations between mobile nodes even when the MAC layer supports multi-point communication. Second, we never push contents in the network and rely instead on a receiver-driven dissemination scheme (pull mode), i.e., every node has entire control of the content it downloads. The two main arguments for these choices are simplicity; and best exploitation of short contact durations. Besides being much simpler to implement (compared to parallel connections), the pairwise connections also maximizes the throughput for the nodes and thereby also the chances for the nodes to complete their downloads; we presume that it is better to allow two nodes to complete their downloads instead of ending up with several downloads interrupted by mobility. Furthermore, the transport layer is able to optimize the flow control between the two associated nodes and is not constrained by the slowest receiver in range, as would be the case in a broadcasting scheme.

Another key property of PodNet is the absence of any network layer routing algorithm. Conceptually, content simply propagates in the system on a per-contact basis according to the solicitation strategy of the individual nodes. This approach questions the need for a networking layer. In fact, we design PodNet without a networking layer and implement the transport layer directly on top of the data link layer, as illustrated in figure 1. As a result, since we do not perform multi-hop routing explicitly, the distribution performance is mainly determined by the selection of nodes we are synchronizing with and the order by which we download contents from the peers.

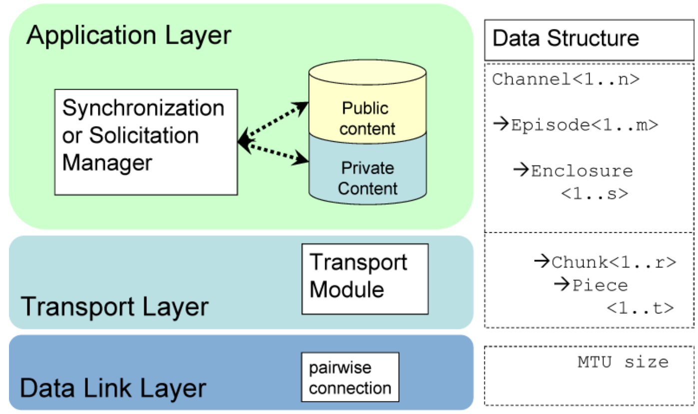

Fig. 1. System architecture and data structures of PodNet.

When two nodes associate, they start to solicit content from each other. First, a node solicits contents for channels that the user carrying the device is subscribed to. This type of content is stored in the device's *private* cache. If the other node does not have any contents for those channels, a node solicits public contents, i.e., any other channel. This type of content is stored in the *public* cache of the device and contributes to a better global dissemination of content.

B. Organization and addressing of content

As indicated before, we divide and group content in channels. Typical *channels* are for example sports channels, news channels, or music channels (of different types), but also more local or private channels like photo albums or audio recordings that users are willing to share with their surrounding. Each channel has a name and is uniquely identified with a globally unique id, a permanent identifier assigned by the creator of the channel. A channel acts like an unlimited container for *episodes*. An episode is the actual data object of interest and can be interpreted as programs within the channel. Each channel may contain more than one episode. An episode contains ,among other information, a release date and a description field which carries arbitrary text, meta information describing the content of the episode. An episode also contains an ID, but this id must be unique only in the scope of the channel. Episodes are further divided into *enclosures*. An enclosure is an attachment that consists of one file. An enclosure would typically be an audio, video, or text file.

To be able to efficiently transfer episodes and enclosures over opportunistic contacts, we divide the episodes into *chunks*. Chunks are small data units of a fixed size (256 KB in our implementation), which easily can be exchanged during a single contact of limited duration. The chunks are indexed starting from zero. The nodes can use these indexes to resume interrupted downloads. Chunks are usually too big to fit into one data link frame. They are hence further divided into *pieces*

Channel	Episode	Enclosure	Chunk 256 KBytes	Piece MTU size
Fox News	Today at noon	Video 1	Chunk 1	Piece 1
				Piece 2
				Piece 3-x
		Video 2	Chunk 1	Piece 1-x
Classic Music	Bach	1^{st} track	Chunk 1	Piece 1-x
			Chunk 2-z	Piece x-y
	Mozart	2^{nd} track	Chunk 1	Piece 1-x
			Chunk 2	Piece x-y
Rock Music	Queen	1^{st} song	Chunk 1	Piece 1-x
		2^{nd} song	Chunk 1	Piece 1-x
	Metallica	1^{st} song	Chunk 1	Piece 1-x
			Chunk 2	Piece x-y
		2^{nd} song	Chunk 1	Piece 1-x
Application Layer			Transport Layer	

TABLE I
ORGANIZATION AND STRUCTURE OF CONTENT IN PODNET.

that fit into the payload of a MAC frame. To keep track of the correct order of pieces, they are always accompanied by an offset that refers to a position inside a chunk; the first offset is zero. A summary of the content organization and data structure is given in Table I.

The reader might have noticed that our wording is to some extent analogous to the one used in podcasting and BitTorrent, two popular Internet-based content distribution systems. Indeed, our organization of the content is inspired by those two successful concepts.

IV. EVALUATION

To evaluate the performance of PodNet, we have implemented and tested the impact of parameter changes to the system performance as well as the influence of the mobility and node density. The transport mechanism for the contents is primarily the mobility of the users; it can be studied but not affected. We devote most of the space in this section to dependence of the performance on the mobility. There are not many system parameters that can be tuned in the design: The air interface and the available transmit power determines the communication range and bit rate for a mobile node. The range affects the rate and the duration of contacts, and the bit rate determines the volume of data that can be sent and received during a contact. We have decided to rely on existing, standardized link layers such as Bluetooth and IEEE 802.11 and consequently we do not address the data and physical link design.

Towards these ends, we have built a simulation model that takes a grid topology of an open network of interconnected streets as input. For each street i that enters into the network, we attach an arrival process for the mobile nodes, with an average arrival rate of λ_i; we only consider Poisson arrivals herein, as would be the case if people arrive independently of one another. Each arriving node independently selects a random speed from an arbitrary but known probability distribution. The node then traverses the street at constant speed until it arrives at an exit or the next intersection. When a node arrives at an intersection, it probabilistically selects one of the adjacent streets as its next destination. These routing probabilities at the intersection are configuration parameters in the simulator. After selecting a street, the node picks a new speed from the speed distribution and continues its journey to the next intersection, where the process is repeated. Whenever the distance between two nodes is less than, or equal to, the transmission range Δ they can establish a connection. Establishing a connection takes a non-negligible time, t_{setup}, and the nodes can only start exchanging contents of interest after that time.

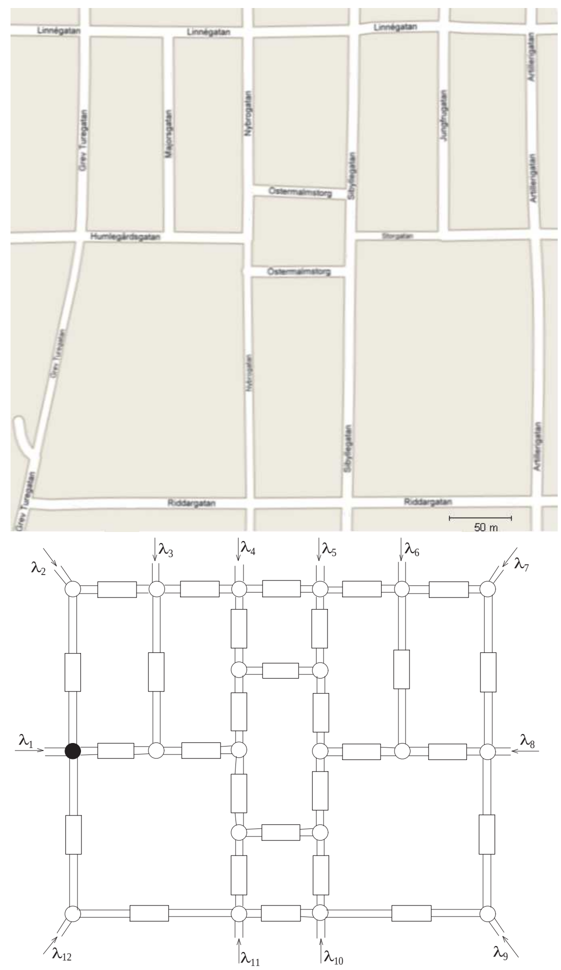

Fig. 2. A part of a downtown Stockholm (left) and the corresponding network of street segments (right) used in our simulations.

The topology we use for our simulations is that of an urban area consisting of one-dimensional streets connected by intersections. For this purpose, we have used part of the *Östermalm* area in central Stockholm, shown in Figure 2. The area is approximately $350 \times 380m^2$ and consists of 28 street segments with lengths that vary between 20 m and 200 m. There are 12 intersections that connect this area to the rest of the city and we assume herein that the arrival rates are the same $\lambda_i = \lambda, i = 1, ..., 12$. The routing probabilities are configured as follows: at an intersection, nodes continue on the same street (if possible) with probability 0.5, or they turn

$\lambda\,(s^{-1})$	$\lambda_{tot}\,(s^{-1})$	$1/\lambda_{tot}\,(s)$	$\rho\,(m^{-1})$
0.01	0.12	8.33	0.009
0.02	0.24	4.17	0.017
0.03	0.36	2.78	0.026
0.04	0.48	2.08	0.034
0.05	0.60	1.67	0.043

TABLE II
RELATIONSHIP BETWEEN PER-ENTRY ARRIVAL RATE λ, TOTAL ARRIVAL RATE λ_{tot}, INTER-ARRIVAL TIMES $1/\lambda_{tot}$ AND NODE DENSITY ρ.

to the adjoining streets with equal probabilities.

In the simulations, we set the node transmission range to $\Delta = 20m$ and the contact setup time is $t_{setup} = 20s$ (this time includes the MAC setup, the discovery, and the synchronization time). The simulations are intended to represent pedestrian mobility and thus nodes choose their speed from a uniform distribution with support [1.00,1.86]. The average speed is 1.43 m/s, which has been measured as the average walking speed of pedestrians in Stockholm [17]. Each node recalculates its position and position-dependent parameters periodically. The length of this update interval controls the precision at which the parameters can be measured. We have set the simulator update interval to 0.1s.

Many of the measures we are interested in are steady-state averages of the stochastic process under inspection. When the stationary distribution of the system is known, the simulation model can be initialized according to it and the system starts in steady state. In our case, the stationary distribution is however not known, and we employ Welch's graphical procedure to estimate the length of the initial transient [18].

We run the simulator for a warm-up period of length l until the steady state has been reached and the total arrival and departure rates for the area have converged. For convenience we say that the simulator is started at time $t = -l$ and steady state has been reached at time $t = 0$. At $t = 0$ a single content station at the intersection of entry 1 (black in Figure 2) releases an episode that is of interest to all nodes. Nodes will start obtaining the episode, either directly from the content station or from peers that already have the episode. We assume that the episode is small and that its transfer time is negligible compared to the set up time, t_{setup}. A node will thus obtain the episode whenever it establishes a contact with a peer that has the episode or with the content station as long as the contact is at least as long as the setup time.

We have simulated for five different per-entry arrival rates. There are 12 entries into the area so the total arrival rate into the area is 12λ. Table II shows the relationship between the per-entry arrival rate λ, total arrival rate λ_{tot}, inter-arrival times $1/\lambda_{tot}$, and node density ρ. The node density ρ in nodes/meter is calculated as $\rho = \lambda_{tot} \cdot \overline{D}/\sum_i l_i$ where $\overline{D}$ is the average time a node spends in the area and l_i is the length of street i, $i = 1, ...28$.

For each arrival rate, we have conducted 100 simulation runs and in each run we collect the time-series of the fraction of nodes carrying the episode in 1 s intervals. In Figure 3 a), we have plotted the average time-series for each of the arrival

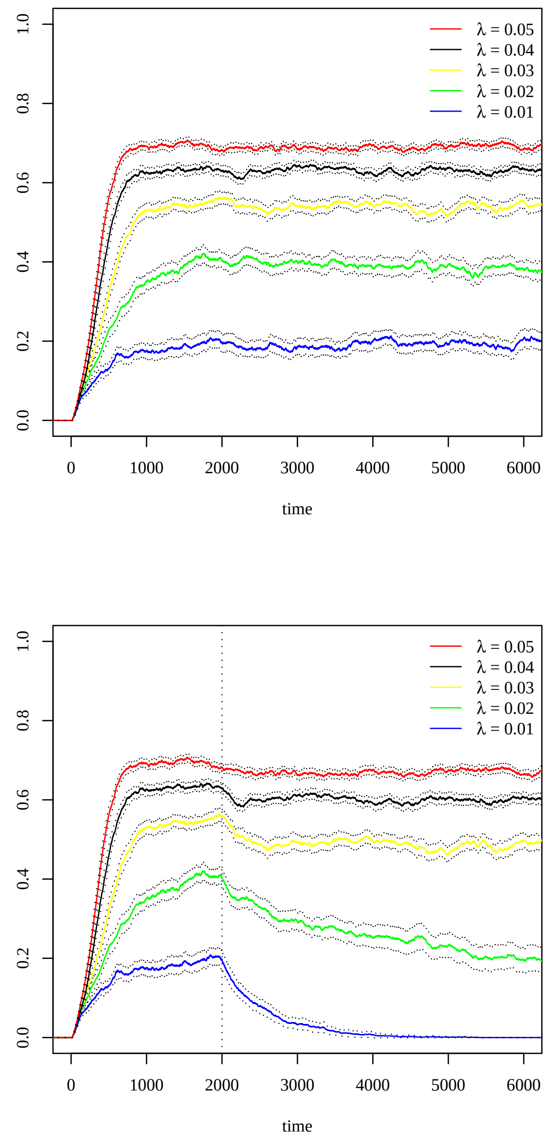

Fig. 3. Average fraction of nodes that have the episode as a function of time (left); the same when the content station is turned off after 2000s (right). Dotted lines show 95% confidence intervals.

rates. The results confirm that the spreading of the episode is strongly dependent on the density of the nodes in the area. For $\lambda = 0.05s^{-1}$, approximately 70% of the nodes in the area are carrying the episode in steady state and it takes approximately 800s to reach the steady state average. For $\lambda = 0.01s^{-1}$ the average fraction of nodes carrying the episode is much lower, or just below 20%, and it takes at least 2000s to reach this steady state average.

Virtual storage effect: It is interesting to study what happens with the distribution when we turn off the content station that initially provides the episode. In Figure 3b), we plot the same scenario as in Figure 3a) except that we now turn off the content station when the episode distribution has reached its steady state. For all arrival rates we consider that a steady state has been reached at at $t = 2000s$. It is interesting to note that when the arrival rate is low the episode disappears from the area because the density of nodes in the area is not high enough to facilitate the ad hoc spreading. This becomes evident at an arrival rate of $\lambda = 0.01$. At higher arrival rates, we see however that the spreading is not dependent on the availability of content at the content station and the episode becomes residential in the area as long as there are new nodes to which it can be passed before old nodes disappearing from the area. In other words, the spreading process results in a virtual storage effect as the episode resides in the area even though there is no infrastructure support and nodes are coming and going.

V. CONCLUSION

We have presented the design and implementation of PodNet, a system architecture for wireless and delay-tolerant public content distribution. PodNet relies on opportunistic data transfers between the mobile nodes to spread the content when they are outside the coverage of fixed access points. PodNet is simple in its design and its communication protocols. We do not setup a multi-hop network but rely solely on pair-wise associations between the mobile nodes. Routing is carried out at the application layer by means of a receiver-driven content solicitation protocol. Addressing is related to contents rather to the node identity or locations.

We have performed various type of simulations to understand the behavior of such a system when the nodes move and exchange contents. Not surprisingly, we have seen that the performance of PodNet highly depends on the node density. However, we have found an interesting effect we refer to as the virtual storage effect. PodNet makes it possible that a piece of content brought into an open area by a mobile node remains into that area forever if the arrival rate of the mobile nodes into this area is large enough. This result shows that PodNet is not only effective to extend the coverage of local area wireless content distribution systems. Under particular conditions, PodNet could also replace the role of a fixed wireless LAN infrastructure for content distribution in crowded urban areas.

ACKNOWLEDGMENTS

We would like to thank Vladimir Vukadinović at KTH for helpful discussions on the work presented herein. Also many thanks to Clemens Wacha from ETH Zurich for the support in the implementation of the system and the very valuable input in the system architecture discussions.

REFERENCES

[1] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges," *EEE Communications Surveys and Tutorials*, pp. 8(1):24–37, 2006.

[2] V. Lenders, M. May, and G. Karlsson, "Wireless Ad Hoc Podcasting," in *Proceedings of IEEE SECON*, San Diego, CA, June 2007.

[3] M. May, C. Wacha, V. Lenders, and G. Karlsson, "Wireless Opportunistic Podcasting: Implementation and Design Tradeoffs," in *Proceedings of the ACM Mobicom CHANTS Workshop*, Montreal, Canada, September 2007.

[4] W. Zhao, M. Ammar, and E. Zegura, "Multicasting in delay tolerant networks: Semantic models and routing algorithms," in *Proceedings of the Sigcomm Workshop on Delay Tolerant Networking*, August 2005.

[5] W. H. Yuen, R. D. Yates, and S. C. Sung, "Noncooperative content distribution in mobile infostation networks," in *Proceedings of the IEEE WCNC*, 2003.

[6] "Delay Tolerant Network Research Group (DTNRG)," http://www.dtnrg.org.

[7] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, and K. Scott, "Delay-tolerant Networking: An Approach to Interplanetary Internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, 2003.

[8] J. Leguay, A. Lindgren, T. Friedman, J. W. Scott, and J. Crowcroft, "Opportunistic Content Distribution in an Urban Setting," in *Proceedings of the ACM SIGCOMM workshop on Challenged Networks (CHANTS)*, Pisa, Italy, August 2006.

[9] A. Lindgren, C. Diot, and J. W. Scott, "Impact of Communication Infrastructure on Forwarding in Pocket Switched Networks," in *Proceedings of the ACM SIGCOMM workshop on Challenged Networks (CHANTS)*, Pisa, Italy, August 2006.

[10] G. Sollazzo, M. Musolesi, and C. Mascolo, "TACO-DTN: A Time-Aware COntent-based dissemination system for Delay Tolerant Networks," in *Proceedings of the First International Workshop on Mobile Opportunistic Networking (Mobiopp)*, Puerto Rico, June 2007.

[11] S. Jung, U. Lee, A. Chang, D.-K. Cho, and M. Gerla, "BlueTorrent: Cooperative Content Sharing for Bluetooth Users,," in *In Proceedings of PerCom*, White Plains, NY, USA, March 2007.

[12] G. Karlsson, V. Lenders, and M. May, "Delay-Tolerant Broadcasting," in *Proceedings of the ACM SIGCOMM CHANTS Workshop*, Pisa, Italy, September 2006.

[13] G. Banavar, T. Chandra, B.Mukherjee, J. Nagarajarao, R. E. Strom, and D. C. Sturman, "An Efficient Multicast Protocol for Content-based Publish-subscribe Systems," in *Proceedings of the 19th IEEE International Conference on Distributed Computing Systems (ICDCS '99)*, Austin, TX, USA, May 1999.

[14] A. Carzaniga, D. Rosenblum, and A. L. Wolf, "Interfaces and Algorithms for a Wide-area Event Notification Service," in *ACM Transactions of Computer Systems*, vol. 19, no. 3, 2001, pp. 332–383.

[15] G. Cugola, E. D. Nitto, and A. Fuggetta, "The JEDI Event-based Infrastrucutre and its Applications to the Development of the OPSS WFMS," *IEEE Transactions on Software Engineering*, vol. 27(9), 2001.

[16] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proceedings of ACM Mobicom*, Boston, USA, 2000.

[17] "The pace of life," www.britishcouncil.org/paceoflife.pdf.

[18] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*. Reading, MA: McGraw-Hill, 3rd edt., 2000.