This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.

A Prioritization-Based Application-Oriented Broadcast Protocol for Delay-Tolerant Networks

Huazhi Gong and JongWon Kim
NetMedia Lab, Gwangju Institute of Sci&Tech (GIST)
1 Oryongdong, Buk-gu, Gwangju, Korea
{hankgong, jongwon}@nm.gist.ac.kr

Abstract—**In this paper, we will study an broadcast protocol based on prioritization for delay-tolerant networks (DTNs). Considering the feature of the DTNs where there may be rare stable links, we propose an application-oriented broadcast (AOB) protocol built atop "bundle layer" in DTNs, which adjusts the forwarding probability for different types of traffic categories. As a result, AOB can provide differentiated services for prioritized data packets and simultaneously reduce the broadcast overhead. The performance of the proposed protocol is evaluated by extensive simulations for scenarios with not only Random Way Point mobility model but also realistic mobility traces collected from cars moving in the city.**

I. Introduction

With the emergence of new wireless networking challenge, a new type of mobile multihop wireless networks known as delay/disruption tolerant networks (DTNs) has become a new active research paradigm. The Delay-Tolerant Networking Research Group (DTNRG) [1] is working on designing new architectures and protocols to satisfy the need of providing interoperable communications with and among extreme and performance-challenged environments where continuous end-to-end connectivity cannot be assumed. Examples of DTNs include inter-planetary networks [2], vehicle ad hoc networks (VANETs) [3], [4] and networks in under-developed areas [5], [6], or some specific type of sensor networks [7], [8].

The main characteristic of the DTN from mobile ad hoc network (MANET) is that there may be no stable end-to-end paths existing in the network mainly because of node mobility. Thus, previously well addressed MANET routing protocols are not valid and tend to perform poorly in such situations. As a result, the architecture of the DTN node has been proposed to be added with a new layer "bundle layer" between the transport and the application layer [9]. The bundle layer stores and forwards messages between mobile nodes. In the seminal work [10], epidemic routing is proposed based on the store-and-forward idea for the first time.

In this paper, we tackle the problem of broadcasting application-oriented data packets across DTNs. The broadcast service is a basic operation for all types of networks. Although it can be simply described as disseminating messages from the source to all other nodes in the network, the broadcast protocol should be reliable and low-overhead in general. The most straightforward broadcasting approach is *flooding*, in which each mobile node immediately retransmits the packets received for the first time. Flooding generates many redundant transmissions, which may cause a serious *broadcast storm problem* [11]. Note that, most of the previously proposed broadcast protocols do not utilize the bundle layer for DTNs. Without store-and-forward operation, performance of unicast protocol could be very poor in DTNs [10]. The same case is with DTN broadcast protocol. Moreover, although the broadcast packets carry different priorities that represent their applications in nature, most existing approaches do not differentiate them. As a consequence, new broadcast protocols properly serving different applications should be developed for the challenging environment of DTNs.

Deterministic ways have been proposed to reduce the overhead for broadcast [12]–[14]. Generally, they should maintain a global overlay structure based on neighborhood information for the whole network and then centralizedly generate optimal solutions to reduce redundant messages. However, it's very difficult to collect and distribute topology information in DTNs due to the node mobility. The alternative is probabilistic broadcasting approaches (or *gossiping*) [15]–[17]. The main idea behind them is that, when a node receives a message, it determines whether to broadcast the packet by computing a probability with local information. Straightforwardly, the probabilistic method is adaptive and robust to failures and mobility due to its simplicity. In a most related work [16], Drabkin *et al.* propose a reliable broadcasting protocol for MANETs by cumulatively announcing the received packets information. However, it still does not explicitly store the packets in the packet buffer for store-and-forward operation and does not consider the packet priority. Considering the priority of different application categories could differentiate the importance of the traffic and help the overall performance. Prioritization-based unicast and broadcast has been extensively addressed in wire networks [18], [19]. In [20], Saleh *et al.* have proposed several methods to protect the high-priority safety message broadcasting for VANETs. However, properly processing packets from other application categories cannot be overlooked.

In this work, we propose an Application-Oriented Broadcast (AOB) protocol based on prioritization for DTNs. AOB is derived from epidemic routing in which each mobile node stores the received packets in the buffer and forwards the

The work reported in this paper was supported by the Basic Research Program of the Korea Science and Engineering Foundation (KOSEF) under Grant No. (R01-2006-000-10556-0).

packets during the meeting of the nodes pairs. To reduce the overhead of broadcast, we apply different forwarding probability for each traffic category. The proposed scheme is of low overhead and robust in nature. With extensive simulations based on model-based and trace-based scenarios, we prove the performance of the proposed protocol.

This paper is organized as following. We introduce the system model and problem statement in Section II. Then the proposed broadcast protocol is explained in Section III. After that, we show the simulation results in Section IV. Finally, we draw the conclusions in Section V.

II. System Model and Problem Statement

In this section, we first introduce the basic assumptions and the application domain of this work. After that, we will define the challenging broadcast problem for DTNs.

A. Assumptions and Application Domain

In the following paper, we assume that each node follows the standard of DTN research group. There is a bundle layer between the transport and application layer. Nodes are of same communication capabilities (communication range and bandwidth) and can only communicate if their sight distance is below the communication range. The MAC protocol deployed is belonging to IEEE 802.11 series of wireless standards. Totally, there are $|\mathcal{N}|$ mobile nodes (denoted by set $\mathcal{N}$) moving around with specific mobility pattern. Each node $n \in \mathcal{N}$ has N_n neighbors. The broadcasting data packet carries different priorities indicating its importance and urgency. We use the priority index I_c to denote this value. The important packet should be assigned with higher priority value and vice verses. For each packet, a field named "valid-time" is added to limit its life time.

We use VANET as an example to explain the application domain of the proposed protocol because it is perhaps the first instance of mobile networks with a direct influence by communication on the behavior and, in particular, mobility of nodes. One example scenario of VANET is shown in Fig. 1. Recently, IEEE 802.11p has been proposed to operate in the 5.9GHz band and deal with roaming from cell to cell in a fast-moving vehicle. For VANET, different priorities of applications can be interpreted as: safety warning/road management, normal car-to-car communication, and advertisement/entertainment. Especially, the safety warning or road management information should be reliably disseminated within short period. On the other hand, the advertisement or entertainment packets are of low priority and those exceeding the valid-time can be removed from the packet buffer in bundle layer with no harm.

B. Problem and Solution Statement

In this work, we try to propose a broadcast protocol to provide different levels of services for data packets with various priorities for DTNs. The proposed protocol should be able to handle the challenges proposed by intermittent links of DTNs. Beside this, we should also reduce the broadcast overhead as much as possible.

Fig. 1. An example scenario of VANET.

Prioritization of network traffic is straightforward in concept: give important network traffic precedence over unimportant network traffic. The network traffic can be classes into categories such as high, medium, and low. When the network has reached its capacity, packets with the lowest priority should be dropped first. Prioritization has been supported on different layers. For MAC layer, the IEEE 802.1q frame-tagging scheme defines a method to insert a tag into an IEEE MAC layer frame. In network layer, the IP packet header has a field called ToS (Type of Service). This field has previously been redefined for the IETF Differentiated Services (DiffServ) strategy.

Several ways to provide differentiated services for prioritized traffic have been addressed for wired/wireless networks before. The queuing methods are the most common ways to handle the prioritization for wired networks. Major networking device manufacturers have already incorporated practical schemes into their commercial products. For example, Cisco implements prioritization using four methods: first-in-first-out queuing, weighted fair queuing, priority queuing, and custom queuing. However, it's hard to provide accurate small timescale performance guarantees for DTNs. As another point of view, IEEE 802.11e defines Quality of Service (QoS) mechanisms to support bandwidth-sensitive applications such as data, voice, and video. A higher-priority traffic category has a shorter Arbitration Interframe Space (AIFS) than a lower-priority traffic category so that it will get more chance to transmit.

Although our protocol is above MAC layer, we have actually borrow the idea of assigning various transmission probabilities for different application categories. Instead of providing different channel accessing probabilities, we let the packet forwarding probabilities be adaptive to the traffic categories. Moreover, a bundle layer with buffered packets for store-and-forward operation will be utilized for the intermittent links in DTNs. Also note that, the priority queuing methods can still be applied for the management of the packet buffer.

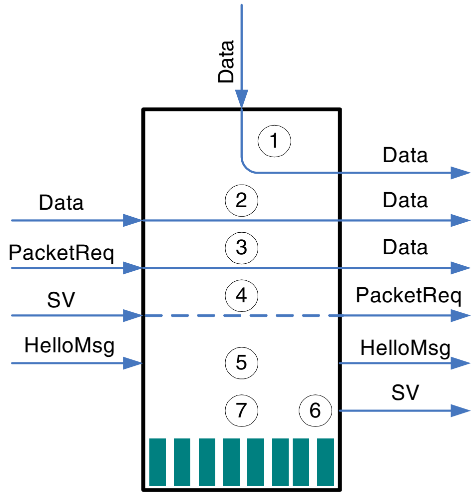

Fig. 2. The diagram to illustrate the proposed broadcast protocol.

III. Proposed Broadcas Protocol

In this section, we first briefly introduce the basic operations of AOB and then describe it in details.

A. Basic Operations

Neighbor discovery: Each node maintains the neighborhood information by using HelloMsg or detecting any received packets from other nodes.

Summary Vector (SV): This operation is defined as exchanging with neighbors the summary vector of buffered packets [10]. Upon receiving a SV message from others, each node can decide the useful packets to fetch by comparing with its own buffered packets. This is the basic approach to overcome the problem of the intermittent connections in DTNs.

Probabilistic forwarding (or gossiping): Mobile node decides whether to broadcast the data packet with a probability computed locally according to neighborhood information and priority index of the received packet.

Random delay: Before broadcasting the data packet, we add a random jitter. This random delay will be decreased with the number of neighbors since we want to give higher forwarding priority to the nodes with more neighbors. Moreover, if the packet waiting for transmission has been received again, the transmission will be cancelled.

B. Protocol in Details

The proposed protocol is illustrated in Fig. 2. The big rectangle represents the broadcast protocol stack. The lines with arrow head represent the event of receiving or transmitting packets. A packet buffer is at the bottom of the big rectangle. These numbered operations are defined as following:

1) If the data packet is from the application layer, AOB simply broadcasts the packet.
2) If AOB receives a data packet from other nodes, AOB will broadcast the packet with a forwarding probability p. The value of p should be increased with the priority index I_c and decreased with the number of neighbors. Assuming a mobile node n with N_n neighbors receives a data packet with priority index I_c, we simply compute the forwarding probability as:
$$p = \min(\frac{I_c}{2N_n}, 1.0).$$
Note that, for extremely important messages like safety warning messages in VANETs, we can simply let p be 1. Moreover, we add a random delay before the transmission. This delay should decrease for node with more neighbors . We let the delay jitter τ be expressed as a linear function:
$$\tau = \begin{cases} rand[(\Delta - N_n)T, (\Delta + 1 - N_n)T] & 1 \leq N_n \leq \Delta \\ rand[0, T] & N_n > \Delta \end{cases}.$$
where T is a unit of time interval and Δ is a constant number representing the average number of neighbors. The function $rand[a, b]$ is to generate a random value between a and b. In the following simulation, we set T and Δ as 0.01 second and 10, respectively.
3) If AOB receives the packet request (PacketReq) message from other nodes, AOB also uses probability p to rebroadcast the data packets. Here, the delay jitter before the broadcast can be simply set as $rand[0, T]$ because we do not need to compete with other nodes in this situation. The packet destination is set as broadcast address to let other nodes be able to overhear it.
4) If a node receives the SV packet, it will compare the carried SV information with its own buffer. If there are useful packets not locally stored, this node will send packet request message to the destination which is the source of SV packet. Moreover, we set a maximum number of requesting packets as ten to avoid occupying the wireless channel too long by one mobile node.
5) The processing of HelloMsg includes two parts. Upon receiving the HelloMsg packet, the node will update the neighbor list so that AOB can periodically compute the number of neighbors with moving average. On the other hand, each node will send a HelloMsg if it has kept quiet for some time. Normally, the interval of broadcasting HelloMsg can be set as one second or more.
6) The overhead of broadcasting SV packet is much higher than that of HelloMsg because it should carry the header information of some stored packets. Broadcasting SV can be based on three conditions: I) the number of received data packets is more than a constant value II) a fixed time interval III) discovery of new neighbors. In our implementation, we use II and III where the time interval of II is configured as five seconds.
7) The packet buffer in bundle layer should be managed. When one node receives a packet, it will only store the packet into its buffer with correct valid-time. Furthermore, the buffered packets that exceed their valid-time should be removed. When the buffer is full, the new packet will replace the earliest low-priority packet.

From the description above, we can conclude that the proposed broadcast protocol is distributed and scalable because each node are only required to maintain the one-hop neighborhood information. As for overhead, not only the computation of forwarding probability is simple but also we have carefully designed the protocol to reduce the number of rebroadcasting messages.

IV. Simulation Results

In this section, we will first describe the simulation environment and some simulation parameters. After that, the simluation results from different scenarios will be shown and discussed.

A. Simulation Description

To evaluate the proposed scheme, we have built the simulation platform based on JiST/SWANS package [21]. JiST is a high-performance discrete event simulation engine that runs over a standard Java virtual machine. SWANS is a scalable wireless network simulator built atop the JiST platform, which is able to simulate much larger networks than NS2 [22]. Comparing with other DTN simulators [23], [24], JiST/SWANS possesses the accurate modelling of physical and MAC layer for wireless networks that is necessary for our simulation purpose. Thus, we have extended JiST/SWANS to support application prioritization and broadcast routing. Moreover, to use the realistic mobility data provided with NS2 mobility format [25], we have also imported the code from [26] to parse the mobility file. Finally, we have developed a NS2 Mobility Visualizer with Python, which is able to show the animation of the moving pattern.

We have simulated AOB with both Random Way Point (RWP) mobility model and the realistic VANET mobility data. RWP model is a simple mobility model widely used for wireless networking research, in which the mobile nodes randomly move and pause in the topology. The realistic mobility data is generated by TraNS [25] which links two open-source simulator, a traffic simulator, SUMO [27] and a network simulator, NS2 [22]. The resulting mobility data is saved as a TCL file with NS2 mobility format.

For comparison, we have also implemented epidemic-style flooding and plain flooding. In epidemic-style flooding, the node stores the received packets in bundle layer and disseminates the data messages with forwarding probability as 1. Straightforwardly, epidemic-style flooding can provide the best reachability of data packets at the cost of highest overhead. We use packet delivery ratio as the metric to measure the reachability of data packets across the network:

$$\frac{|\text{Recv. packets in DTN}|}{|\text{Send packets in DTN}| \times |\mathcal{N}|}.$$

Another important metric to measure the protocol overhead is the ratio of the number of broadcasting transmissions using the proposed protocol to that number in epidemic-style flooding. For each scenario, we repeat the simulations for three times with different random seeds and use the average value. The

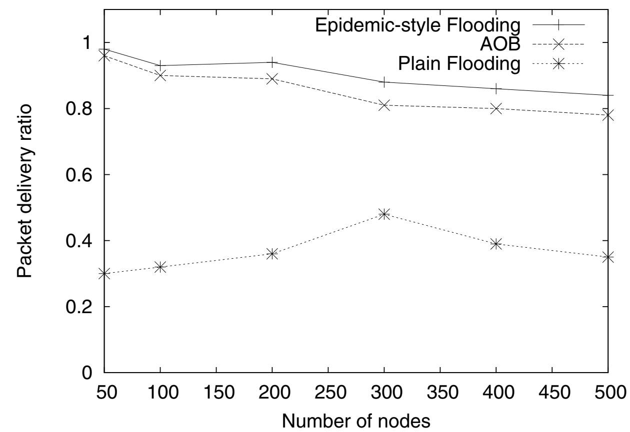

Fig. 3. Packet delivery ratio w.r.t. number of mobile nodes in RWP scenarios.

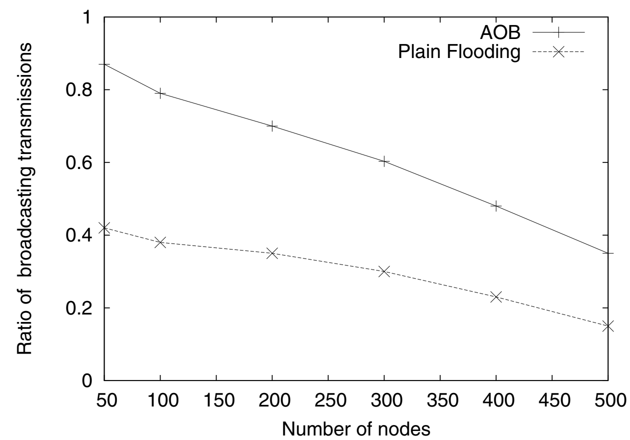

Fig. 4. Ratio of broadcasting transmissions w.r.t. the number of mobile nodes in RWP scenarios.

parameter of MAC protocol is configured according to the specifications of IEEE 802.11p standards. The bandwidth of the wireless channel is 3Mbps with Rayleigh fading. For all simulations, we set the maximum size of packet buffer as storing 100 data packets. We configure different traffic categories with priority index I_c and packet valid-time according to the purpose of our simulations.

B. Results

First, we compare the performance of three schemes in the scenarios with RWP mobility pattern. The nodes are configured with an average velocity of $10\ m/s$ moving in a $2000m \times 2000m$ field. The simulation time is configured as $100s$. Traffic priority is uniform and set as $I_c = 3$. The packet valid-time is set as long enough till the end of the simulation. Among all, 10% mobile nodes broadcast one packet each second. This amount of traffic load is not enough to saturate the wireless channel. In Fig. 3, we have plotted the packet delivery ratio of the three schemes with respect to (w.r.t.) the number of mobile nodes. According to the figure, with the proposed protocol, we can achieve almost same packet delivery ratio as

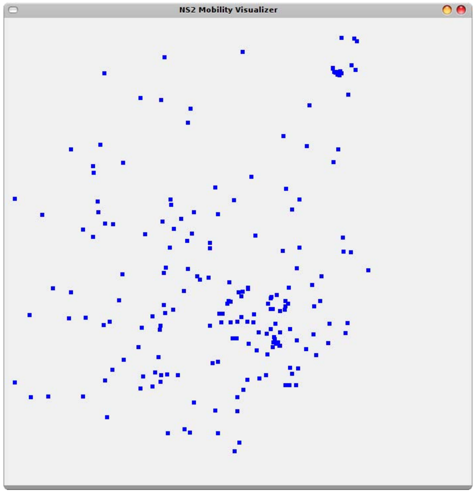

Fig. 5. Visualization of the simulated VANET scenario based on NS2 mobility file.

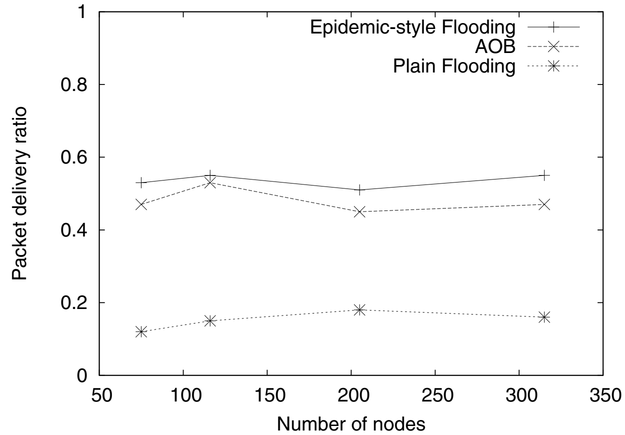

Fig. 6. Packet delivery ratio w.r.t. number of mobile nodes in realistic VANET.

epidemic-style flooding. This packet delivery ratio is almost double of that of plain flooding. For plain flooding, when there are only 50 nodes in the topology, the packet delivery ratio is even lower that of scenario with 300 nodes because higher node density provides better coverage of the field. However, more collisions and overhead will decease the performance when the density keeps increasing. On the other hand, we compare the overhead of broadcasting transmissions of AOB with epidemic-style flooding in Fig. 4. Straightforwardly, plain flooding should have the lowest overhead because it does not buffer the data packets. Here, we can also observe that the low-overhead benefit of AOB will be more apparent when the number of nodes is increased. Especially, when there are 500 nodes in the topology, the proposed protocol can save more than 50% broadcasting transmissions.

Next, we have simulated the scenarios based on realistic mobility data. A snapshot of the simulated topology is captured by our NS2 Mobility Visualizer shown in the Fig. 5. The data is collected from the city of San Francisco of size $4000m \times 4000m$ area. Simulation time is configured as $600s$. The traffic pattern is same as the simulations conducted for RWP. We plot the results in Fig. 6 and Fig. 7. Due to the limitation of the mobility of cars in the city, the epidemic-style flooding can only achieve around 60% packet delivery ratio which is much lower than 90% in RWP scenarios. With plain flooding, the packet delivery ratio is even less than 20%. The difference of delivery ratio between epidemic-style flooding and AOB is still small. Furthermore, the ratio of broadcasting transmissions for the proposed protocol shows similar trend as we have mentioned above. Note that, the transmission ratio of AOB to epidemic-style flooding will keep decreasing when we increase the simulation time. As a result, more free airtime of wireless channel will be available for others.

In the following two set of experiments, we use a VANET scenario of 75 cars in a $2000m \times 2000m$ field with three types of traffic priorities (I_c =1, 3, and 5). The first set

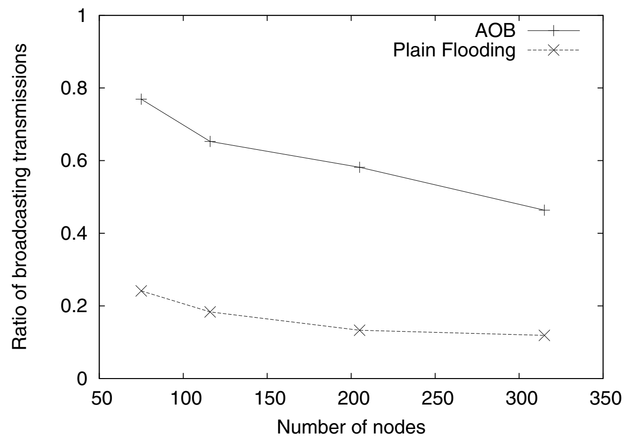

Fig. 7. Ratio of broadcasting transmissions w.r.t. number of nodes in realistic VANET.

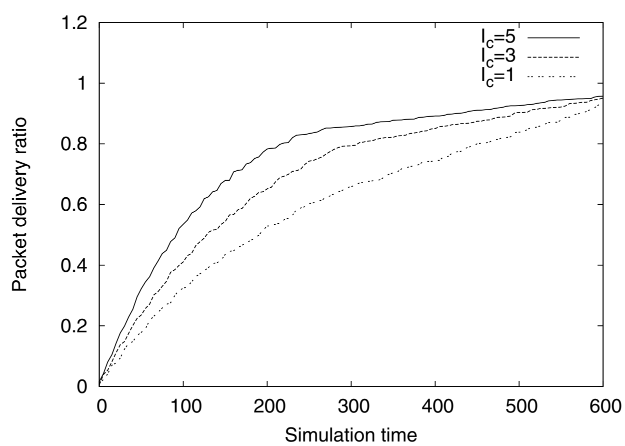

Fig. 8. Packet delivery ratio w.r.t. simulation time for traffic categories with different priority index I_c .

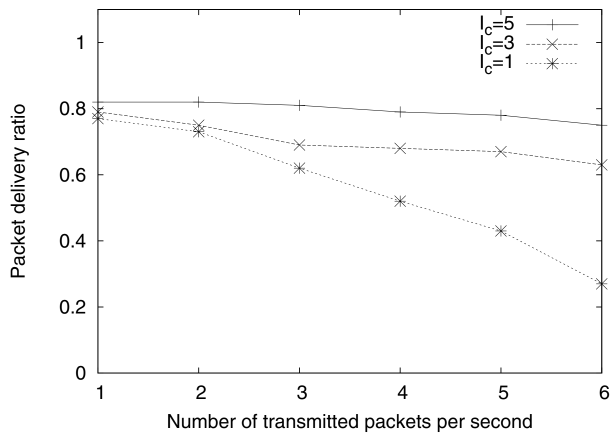

Fig. 9. Packet delivery ratio w.r.t. traffic load for traffic categories with different priority index I_c .

of experiments is to show the ability of serving different traffic categories with AOB. Although the total simulation time is $600s$, the traffic broadcasted from application layer is only enabled between 0~$280s$. In Fig. 8, we show the packet delivery ratio w.r.t. simulation time for each traffic category. We can observe that the high-priority traffic always has better delivery ratio than low-priority traffic. Especially, this ratio of high-priority traffic increases with a much faster speed between 0 and $300s$ and slows down after $300s$ because no new traffic generated from the application layer. However, the packet delivery ratio of these three categories will finally converge to a similar value which is more than 90%. The reason for this high packet delivery ratio is due to that the data packets generated before $300s$ are still carried and forwarded by the cars among 300~$600s$.

Finally, we want to see the impact of the load from low-priority traffic on other traffic categories. The valid-time of low-priority traffic ($I_c = 1$) is set as $10s$ while this field of other two traffic categories is set as long enough to finish the simulation. We totally choose 20 cars to broadcast a fixed number of packets each second. In Fig. 9, we plot the trend of packet delivery ratio when increasing the number of packets broadcasted each second. According to it, we can observe the delivery ratio of the packets with high-priority ($I_c = 5$) has not been seriously harmed (less than 10%) when the traffic load ($I_c = 1$) is greatly increased. On the other hand, the delivery ratio of packets with priority index ($I_c = 1$) will sharply drop as we have expected. The observations above also confirm the original motivation of our proposed protocol.

V. CONCLUDING REMARKS

In this paper, we have proposed a broadcast protocol (AOB) based on prioritization to serve different application categories in DTNs. By assigning priorities and limiting the valid-time for the packets from the application layer, we apply an forwarding probability to the received packets. The probability is able to be locally computed with one-hop neighborhood information. Moreover, the received packets will be buffered for store-and-forward operation. The performance from extensive simulations shows that the proposed protocol not only can differentiate the service for packets with different priorities but also can greatly control the broadcasting overhead. For the future work, we hope to do more simulations for other types of DTNs due to that the current simulation results are mostly based on VANET traces.

REFERENCES

[1] "Delay tolerant networking research group (DTNRG)." http://www.dtnrg.org/wiki.
[2] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: An approach to interplanetary internet," *IEEE Communication Magazine*, vol. 41, no. 6, pp. 128–136, 2003.
[3] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *Proc. IEEE INFOCOM*, 2006.
[4] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav, "Low-cost communication for rural internet kiosks using mechanical backhaul," in *Proc. ACM MobiCom*, 2006.
[5] "Wizzy project." http://www.wizzy.org.za.
[6] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: Rethinking connectivity in developing nations," *IEEE Computer*, vol. 37, no. 1, pp. 78–83, Jan. 2004.
[7] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experience with ZebraNet," in *ASPLOS-X*, 2002.
[8] M. Dunbabin, P. Corke, I. Vailescu, and D. Rsu, "Data muling over underwater wireless sensor networks using an autonomous underwater vehile," in *Proc. Intl Conf on Robotics and Automation (ICRA)*, IEEE, May 2006.
[9] K. Fall, "A delay-tolerant network architecture for challenged Internets," in *Proc. SIGCOMM*, ACM, Aug. 2003.
[10] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," technical report cs-2000-06, Duke, July 2000.
[11] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *Proc. MobiHoc*, ACM, 2002.
[12] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," in *Proc. MSWIM*, pp. 61–68, ACM, 2000.
[13] J. Wu and F. Dai, "Broadcasting in ad hoc networks based on self-pruning," in *Proc. INFOCOM2003*, IEEE, 2003.
[14] W. Peng and X. Lu, "On the reduction of broadcast redundancy in mobile ad hod networks," in *Proc. MOBIHOC*, ACM, 2000.
[15] J. Cartigny and D. Simplot, "Border node retransmission based probabilistic broadcast protocols in ad-hoc networks," *Telecommnication Systems*, vol. 22, no. 1-4, pp. 189–204, 2003.
[16] Z. Haas, J. Halpern, and L. Li, "Gossip-based ad hoc routing," in *Proc. INFOCOM*, pp. 1707–1716, IEEE, 2002.
[17] V. Drabkin, "RAPID: reliable probabilistic dissemination in wireless ad-hoc networks," in *26th International Symposium on Reliable Distributed Systems*, IEEE, 2007.
[18] K. Kilkki, *Differential services for the Internet*. Indianapolis, IN, USA: Macmillan Technical Publishing, June 1999.
[19] C.-H. Yeh, E. M. A. Varvarigos, and A. Eshoul, "A priority-based balanced routing scheme for random broadcasting and routing in tori," in *Proc. ICPP*, IEEE, 2003.
[20] Y. Saleh, F. Mahmood, and B. Abderrahim, "Performance of beacon safety message dissemination in vehicular ad hoc networks (vanets)," *Journal of Zhejiang University SCIENCE A*, vol. 8, no. 12, pp. 1990–2004, 2007.
[21] "JiST/SWANS: Java in Simulation Time/Scalable Wireless Ad hoc Network Simulator." http://jist.ece.cornell.edu/.
[22] "The network simulator 2." http://www.isi.edu/nsnam/ns/.
[23] "DTNSim2." http://watwire.uwaterloo.ca/DTN/sim/.
[24] A. Keränen and J. Ott, "Increasing reality for dtn protocol simulations," technical report, Helsinki University of Technology, July 2007.
[25] "TraNS: Real mobility traces." http://trans.epfl.ch/.
[26] "Vanet.info." http://vanet.info/.
[27] "SUMO: Simulation of Urban MObility." http://sumo.sourceforge.net/docs/bibliography.shtml.