Peer-to-Peer Netw. Appl. (2018) 11:181–197
DOI 10.1007/s12083-016-0536-2

Delay-tolerant sensing data delivery for IoT network by using signal strength information

Sun-Hyun Kim[1] · **Seung-Jae Han**[1]

Received: 12 May 2016 / Accepted: 30 November 2016 / Published online: 16 December 2016

Abstract Conventional cellular technology is being extended for Internet of Thing(IoT) network, which needs to support relatively small amount of traffic generated by a large number of devices. This so-called Machine-Type Communication(MTC) cellular however is not a universal solution for IoT connectivity. It may be infeasible to equip some IoT devices with MTC cellular due to manufacturing cost, operation fee, limited battery lifetime, etc. For low-power IoT devices, using shorter range radios (e.g., IEEE 802.15.4) is more viable. Sensing data generated by the IoT devices are delivered to the IoT gateway by the forwarding of other nodes without relying on cellular links. Such peer-to-peer data delivery may utilize 'muling' (i.e., data is stored and carried by mobile nodes) if the forwarding path may not be immediately available. Since muling will extend the delivery delay, it is suitable for the IoT applications that can tolerate relatively large message delivery delay. In this paper, we propose a peer-to-peer DTN (Delay Tolerant Network) routing scheme for IoT network. To enhance routing efficiency, we utilize the location information of the nodes. Our scheme does not rely on expensive location tracking methods like GPS or triangulation. Instead, we use RSSI (Received Signal Strength Indicator) which is readily available in virtually any wireless network at low cost. It is shown that the proposed scheme clearly outperforms the existing schemes via extensive simulations.

Keywords DTN routing · Signal strength · Machine-type communication (MTC) · Peer-to-peer data delivery · Internet of things

1 Introduction

Cellular network is designed to provide instant and continuous connectivity over single hop wireless links between base stations and mobile terminals that human carry. Recently, cellular network technology is being extended to provide connectivity for IoT(Internet of Things) devices. This so-called Machine-Type Communication(MTC) [1] is quite different from the conventional Human-to-Human(H2H) communication which is optimized to provide high data rate to a relatively small number of devices. MTC needs to support much larger number of devices which usually require the transmission of small amount of data, while extending the lifetime of the devices is oftentimes critical. Current cellular network is known as being particularly inefficient when a large number of IoT devices perform uplink random access operation when they first turn on or they need timing synchronization [2, 3]. Such techniques as enhanced Power Saving Mode (PSM) and extended discontinuous Reception (DRX) are adopted for MTC cellular to achieve the scalability and terminal energy efficiency at the expense of extended communication delay.

MTC cellular network, however, is not a universal solution for IoT connectivity. It may be infeasible to equip some IoT devices with MTC cellular interfaces due to manufacturing cost, operation fee, limited battery lifetime, etc. Some IoT devices are powered only by small batteries

An earlier version of this paper is published in [28].

✉ Sun-Hyun Kim
molkey@yonsei.ac.kr

✉ Seung-Jae Han
seungjaehan@yonsei.ac.kr

[1] Yonsei University, Seoul, Korea

and are equipped with very limited computing power. For such devices, shorter range radios (e.g., Bluetooth, IEEE 802.15.4, IEEE 802.11) are more suitable than MTC cellular. By using short range radios, IoT devices are connected to a so-called 'IoT gateway' [21, 22] that provides the connectivity with the external Internet. Sensing data generated by the IoT devices are delivered to the IoT gateway by the forwarding of other nodes (e.g, mobile phones or other IoT devices) without relying on cellular links. When neighbor nodes for forwarding do not exist, sensing data may be carried by the nodes, which is called 'data muling'. Because of data muling, the data delivery from the IoT devices to the IoT gateway may take relatively long delay as compared to MTC cellular. Therefore, this approach is suitable for the IoT applications that can tolerate large delay, while the applications that require immediate message delivery should be served via MTC cellular. Using the peer-to-peer forwarding for sensing data delivery is called 'DTN-IoT' or 'Opportunistic IoT' in existing studies [4, 5].

In this paper, we focus on the aspect of peer-to-peer routing for DTN-IoT. Peer-to-peer DTN routing may take either the forwarding method or the replication method. In the forwarding method, a node deletes the message after forwarding it to another node, so that only one copy of a message exists in the network at a moment. Example schemes include [9, 14, 15]. In the replication method, the message is not deleted after forwarding and as a result multiple replicas of a message exist in the network. While the replication method uses more resources than the forwarding method, the chance of successful message delivery is higher and the message delivery delay is lower. For this reason, most DTN routing adopts the replication method.

Epidemic routing [7] is the most basic replication scheme that adopts flooding. When a node that carries a message encounters other nodes which do not have the message, the message is copied to those nodes. Spray and Wait routing [10] is designed to reduce the high traffic overhead of flooding. In this scheme, the maximum number of the copies of a message is predetermined. If a node meets another node that does not have a copy of the message, a half of the replicas that the node owns are sent to the new node. When there is no more message replica to spray, the node carries the message until it meets the destination. Spray and Focus routing [11] is a variation of Spray and Wait routing. The message spraying phase is the same as Spray and Wait routing. After spraying, message forwarding is performed on the basis of the node encounter history (i.e., Higher frequency of encountering of the destination node is preferred). Encounter Based routing [12] is another variation of Spray and Wait routing. In the spraying phase the number of copies to split is determined by the node encounter history, instead of splitting the replicas by half. In DA-SW(Density-Aware Spray and Wait), the number of copies to spray is adaptively determined by considering connectivity and expected delay.

Mobile devices are typically carried by people and therefore the contact pattern of devices is decided by social ties between people. In Prophet routing [13], each node keeps track of the contact history with other nodes. The scheme uses not only direct contacts but also indirect contacts to utilize the transitivity. The strength of social tie is decided by contact frequency. There exist other DTN routing schemes that utilize the social ties [16–18].

In general, it is desirable to reduce the traffic overhead while ensuring successful message delivery (and short delivery delay). To meet these conflicting goals, existing schemes utilize the past contact histories or given social relations between nodes. In this paper, we take an approach that utilizes the location information for routing decision. The biggest challenge of location-based routing is the high cost of obtaining location information. Our scheme is different from the existing location-based routing in that we utilize the easily available information. While GPS provides accurate location information, GPS consumes large amount of battery power and is unusable in the indoor environment. Cellular triangulation requires complicated coordination among base stations. Instead of relying on these expensive location tracking methods, we use a very simple location information, RSSI (Received Signal Strength Indicator), which is available in virtually all types of wireless networks and can be used by all types of devices including low-profile IoT devices. However, RSSI provides only partial location knowledge (i.e., the distance from a device to the base station) instead of precise location coordinate. We propose a scheme, called 'contour routing'[1] to use this partial location information.

Huo et al [26] uses RSSI between a node and the destination node to estimate the distance between them. To this end, the destination periodically broadcasts beacon, which requires high power radio transmission. Mei et al [27] propose a DTN routing scheme by utilizing the assistance of the cellular network. This scheme uses the location coordinate of the destination instead of the RSSI information. To obtain the full location coordinate, multiple base stations must be involved, which is a much more expensive task than using the RSSI. When node makes to message relay decision to a neighbor node, perfect coordinate information of each other is also needed.

The proposed scheme decides the peer-to-peer DTN-IoT routing to deliver the sensing data generated by IoT devices to the IoT gateway. The IoT gateway is assumed static while IoT devices can be mobile, and IoT devices may locate

[1] In mountain climbing, a person who knows the height of the destination climbs up the mountain. The RSSI of the destination corresponds to the height of the destination in mountain climbing.

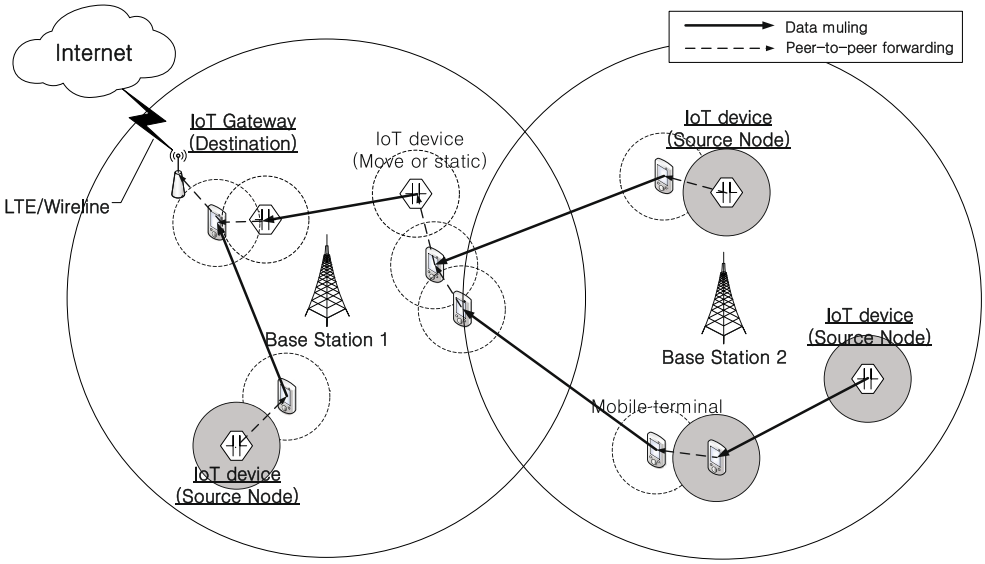


Fig. 1 Illustration of the proposed peer-to-peer DTN-IoT delivery

in difference cells from the IoT gateway. Sensing data is forwarded via peer-to-peer relaying or data muling as illustrated in Fig. 1. The RSSI information is used to select the next forwarding node.

The remainder of the paper is organized as follows. In Section 2, the proposed scheme for the routing within a single cell is described. In Section 3, the performance analysis model of the proposed scheme is presented. In Section 4, the simulation results using random mobility are presented. In Section 5, the simulation results using real mobility trace are presented. In Section 6, we extend our scheme for the multi-cell case. In Section 7, we discuss how to handle the instability of RSSI measurement. Section 8 concludes the paper.

2 *Contour routing* in a single cell

In this section, we present the basic version of the proposed scheme for the case that IoT Gateway (i.e, destination node) and IoT devices (i.e., source nodes) are within the same cell. Later in Section 6, we extend our scheme to deal with the case that the destination and the source locate at different cells (i.e., delivery across the cell boundary).

In cellular networks, channel quality between a user node and the base station is periodically measured for various purposes such as adaptive modulation, power control, handover decision, etc. We term RSSI(Received Signal Strength Indicator) to represent the channel quality measured at a user node. We assume that each node knows the RSSI of itself and the destination node (i.e., IoT gateway). The RSSI measurement allows the estimation of the distance between base station (BS) and user node. Therefore, when the RSSI of a node is known, we can identify a circle around the BS, at which the node locates. We choose a donut-shape region in which the destination node locates and call it 'target contour', the width of which is denoted by w.

Data forwarding is done differently inside and outside of the target contour. Routing is performed in two phases:(i) the approach phase and (ii) the spin phase. In Fig. 2, these two phases are depicted. The approach phase deals with the routing outside the target contour (i.e., until the message reaches to some node in the target contour). The RSSI information is used for routing decision, and single-copy forwarding is used in this phase. The spin phase deals with the routing within the target contour. In this phase, the RSSI information is not useful any more, and we rely on the

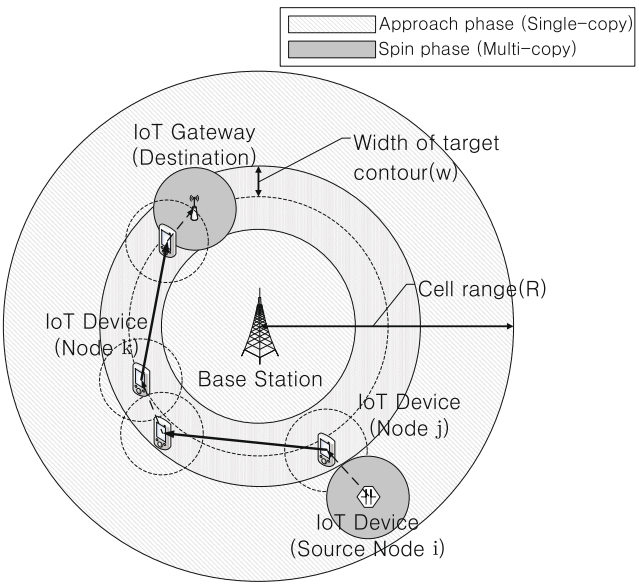


Fig. 2 The proposed scheme for a single cell

replication-based approach in which messages are flooded within the target contour.

The message forwarding in the approach phase is performed as follows. When two nodes approach within each other's communication range, they first exchange the RSSI values. Then, the RSSI value of each node is converted to the distance between the node and the base station. The distance between a node i and the destination is estimated as follows, where $h()$ is a function that converts RSSI to distance.

$$H_i = |h(RSSI_d) - h(RSSI_i)| \quad (1)$$

When a node i encounters a node j, i receives the RSSI of j (i.e., $RSSI_j$) from j. i calculates H_j by using $RSSI_j$. When H_i is bigger than H_j, i forwards the message to j and deletes the message from its buffer. If H_j of a node that carries a message is smaller than the width of target contour w, j is inside the target contour and the spin phase starts.

In the spin phase, limited flooding is performed only among the nodes within the target contour. When a node j encounters a node k, it receives $RSSI_k$. When H_k is smaller than w, it means that k is also within the target contour. If k has not received the message yet, j sends the message to k. In this way, the message will be eventually delivered to the destination node. When a node which is in the spin phase moves out from the contour range, the node stops message replication and reenter the approach phase. By the approach phase algorithm, the node forwards message to the contour range again. This can cause the undesirable excessive ping-pong effect, and we discuss how to deal with this issue later.

In the spin phase, the traffic overhead increases as the number of nodes within the target contour increases. There is, however, a certain point at which the delivery success ratio does not increase any further even if more copies are created. The traffic overhead increase after that point is wasteful. To avoid this, we may limit the number of replicas in the spin phase by adopting a similar method to Spray and Wait routing [10] and its variants.

3 Performance analysis model

In this section, we mathematically analyze the delivery delay, the delivery success ratio, and delivery overhead of the proposed scheme. For analytic modeling, we assume that the users move within the cell coverage according to the Random Way Point(RWP) mobility model. We consider a circle-shape cell with the radius of R, resulting the cell size (N) of $R^2\pi$. The number of users in the network is denoted by n. The communication range of each node's peer-to-peer link is r. The key notations used in the model are summarized in Table 1.

Table 1 Notations

R	The radius of cell
N	The size of network area
r	Peer link's communication range
v	Average velocity of nodes
w	The width of target contour
n	The number of nodes
T	Average node moving time
T_{stop}	Average node pause time
EM_{rwp}	Expected meeting time between nodes
n_c	The number of nodes in target contour
S	Probability that a node is in target contour

The modeling starts with deriving the expected meeting time of two nodes, which is denoted by EM_{rwp}. The expected meeting time represents how long it takes for a node to meet another node (i.e., two nodes approach within the transmission range of each other). By borrowing the following equation from [6], we compute the expected meeting time when there exist only two nodes in the network, which is denoted by E_m.

$$E_m = \frac{1}{p_m \hat{v}_{\text{rwp}} + 2(1 - p_m)} \frac{N}{2rL} (T + T_{\text{stop}}) \quad (2)$$

where $\hat{v}_{\text{rwp}}$ (≈ 1.75) is the normalized relative velocity. p_m is the probability that a node is moving at any time and $p_m = T/(T + T_{\text{stop}})$ where T is L/v. r is the transmission range and L is the expected epoch length and $L = 0.5214\sqrt{N}$. When there are n nodes in the network, the expected meeting time (EM_{rwp}) of any two nodes becomes E_m/n.

3.1 Delivery delay

The delivery delay (i.e., the delay for delivering a message from the source to the destination) is computed by combining EM_{rwp} with the expected number of hops that a message goes through during routing. Here, a hop means a meeting between two nodes for message forwarding.

We first derive the expected delay of the approach phase. In this phase, single-copy forwarding is used and therefore there always is only one node that carries the data. Under the assumption that the nodes are uniformly distributed in the network, the source node locates in $R/2$ distance from base station on average. We can calculate the probability that a node is in the target contour by using the ratio of area sizes. The size of the cell is $R^2\pi$. The size of the target contour is

$$(R/2 + w)^2\pi - (R/2 - w)^2\pi = 2Rw\pi$$

As the result, the probability S that a node locates in the target contour is

$$S = \frac{2Rw\pi}{R^2\pi} = \frac{2w}{R}$$

Now, the probability that a node encounters another node which is not in the target contour becomes $1 - S$. We model the forwarding behavior in the approach phase by using the probability graph shown in Fig. 3. Using this model, we obtain the following equation for the expected number of encounters during the approach phase.

$$\sum_{k=1}^{\infty} kS(1-S)^{(k-1)}$$

Finally, we calculate the expected delay of the approach phase by multiplying the expected encounter time to the above equation.

$$\sum_{k=1}^{\infty} kS(1-S)^{(k-1)} \times EM_{rwp}$$

Now we turn to the delay modeling of the spin phase. As a preliminary step, let us first consider the delivery delay of flooding. We begin with calculating the number of hops that a message goes through when it is routed from the source to the destination via flooding. When a node receives a copy of the message during routing, we term that the node is 'infected'. At the beginning, there is only one infected node (i.e., the source node), named A. After one EM_{rwp}, the node A will encounter another node B. The probability that B is the destination is $\frac{1}{n-1}$. The probability which B is not the destination is $1 - \frac{1}{n-1}$. If B is not the destination, A will send a copy of the message to B and flooding will be continued. After the another passage of EM_{rwp}, B will encounter node C and A will also encounter node D. The probability that C or D is the destination is $\frac{2}{n-2}$ because there exist two infected nodes (i.e., A and B). By generalizing this reasoning, we obtain the following. After $k \times EM_{rwp}$ (i.e., after k hops of forwarding), the probability that a message is delivered to the destination is $\frac{2^{k-1}}{n-2^{k-1}}$. The probability that a message is not yet delivered to the destination is $1 - \frac{2^{k-1}}{n-2^{k-1}}$ There is one condition for the above equation. The equation does not hold when 2^{k-1} is bigger than $n - 2^{k-1}$. In such a case, the probability becomes 1.

We consider all possible ways in which the message is delivered to the destination by using the probability graph shown in Fig. 4. Now, The expected number of hops ($EH_{flooding}$) before meeting the destination is computed as follows:

$$EH_{flooding} = \sum_{k=1}^{k_f} kP_k,$$

$$P_k = \begin{cases} \frac{2^0}{n-2^0} & \text{if } k = 1 \\ \left(\prod_{i=1}^{k-1} \frac{2^{i-1}}{n-2^{i-1}}\right) \times \frac{2^{k-1}}{n-2^{k-1}} & \text{if } 2^{k-2} < n - 2^{k-2} \\ \prod_{i=1}^{k-1} \frac{2^{i-1}}{n-2^{i-1}} & \text{if } 2^{k-2} > n - 2^{k-2} \end{cases}$$

where $k_f = min_k\{2^{k-2} > n - 2^{k-2}\}$ which means that after $k_f \times EM_{rwp}$, all nodes become infected.

The expected delivery delay becomes:

$$EH_{flooding} \times EM_{rwp}.$$

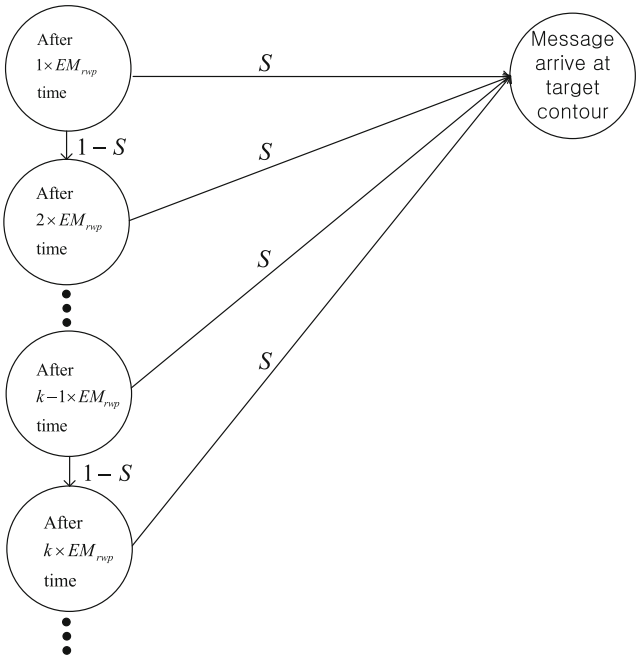


Fig. 3 The delay estimation model for the approach phase

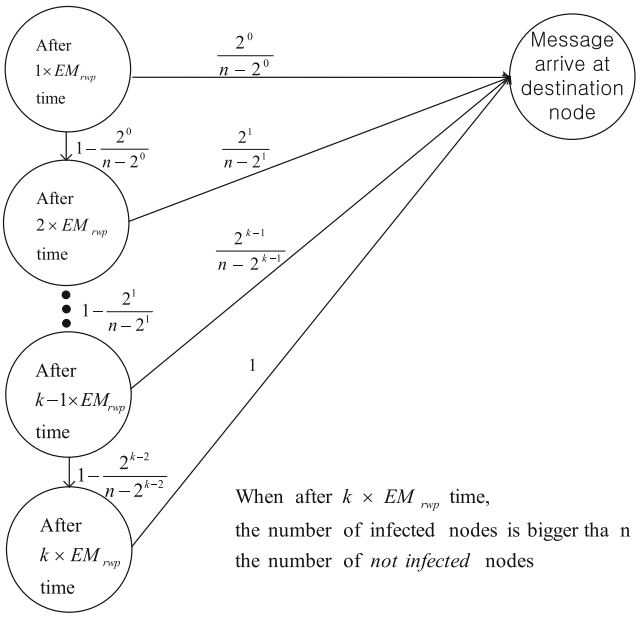


Fig. 4 The delay estimation model for flooding

The delay modeling of the spin phase is very similar to that of flooding. In the spin phase, flooding is limited within the target contour. The number of nodes in the target contour, which is denoted by n_c, is $n \times S$. After EM_{rwp} from the beginning of the spin phase, the first infected node will encounter another node within the target contour. The probability that the encountered node is the destination is $1/(n_c - 1)$. If the encountered node is not the destination, the message will be copied to the node and the number of infected node will become two. After another EM_{rwp}, the two infected node will encounter another two nodes. The probability that one of the two encountered nodes is the destination is $1 - 2/(n_c - 2)$. By repeating this reasoning, the expected number of hops in the spin phase until the message reaches to the destination can be calculated as follows:

$$EH_{spin} = \sum_{k=1}^{k_{fc}} kP_k,$$

$$P_k = \begin{cases} \frac{2^0}{n_c - 2^0} & \text{if } k = 1 \\ \left(\prod_{i=1}^{k-1} \frac{2^{i-1}}{n_c - 2^{i-1}}\right) \times \frac{2^{k-1}}{n_c - 2^{k-1}} & \text{if } 2^{k-2} < n_c - 2^{k-2} \\ \prod_{i=1}^{k-1} \frac{2^{i-1}}{n_c - 2^{i-1}} & \text{if } 2^{k-2} > n_c - 2^{k-2} \end{cases}$$

where k_{fc} is $min_k\{2^{k-2} > n_c - 2^{k-2}\}$.

The expected delay time is $EH_{spin} \times EM_{rwp}$. Our model assumes that the nodes stay within the target contour during the spin phase, which is approximation. That is, our equation does not consider the case that a node moves out of target contour before meeting the destination in the spin phase. The simulation results given in Section 4, however, indicates that this approximation does not have a substantial impact on the accuracy of our model.

3.2 Delivery success ratio and delivery overhead

So far, we have assumed that the message lifetime is infinite. As a result, messages are always delivered to the destination. In realistic setting, messages have finite TTL(Time To Live), and the delivery ratio may not always be 100 %. In such a case, the delivery ratio becomes the key performance metric for given TTL. We use a similar method as in the delivery delay model for the delivery ratio modeling. The main difference is that the number of hops over which a message can be forwarded before the expiration of TTL is considered.

We denote this hop number by TTL_h, which is calculated by $\lfloor TTL/EM_{rwp} \rfloor$. If a state is reachable after larger number of hops than TTL_h from the initial state, we delete that state in the model. If k is the number of hops that a message goes through during the approach phase, $TTL_h - k$ is the maximum number of hops that a messages can afford during the spin phase. The probability of message delivery in k-hops, which is denoted by P_k, can be calculated as follows.

$$P_k = \begin{cases} \frac{2^0}{n_c - 2^0} & \text{if k = 1} \\ \left(\prod_{i=1}^{k-1} \frac{2^{i-1}}{n_c - 2^{i-1}}\right) \times \frac{2^{k-1}}{n_c - 2^{k-1}} & \text{if } 2^{k-2} < n_c - 2^{k-2} \\ \prod_{i=1}^{k-1} \frac{2^{i-1}}{n_c - 2^{i-1}} & \text{if } 2^{k-2} > n_c - 2^{k-2} \end{cases}$$

By considering all possible cases that a message is delivered to the destination within TTL_h, the estimated delivery ratio can be calculated as follows.

$$\sum_{k=1}^{TTL_h} \left(S(1-S)^{k-1} \times \sum_{i=1}^{TTL_h - k} P_i \right)$$

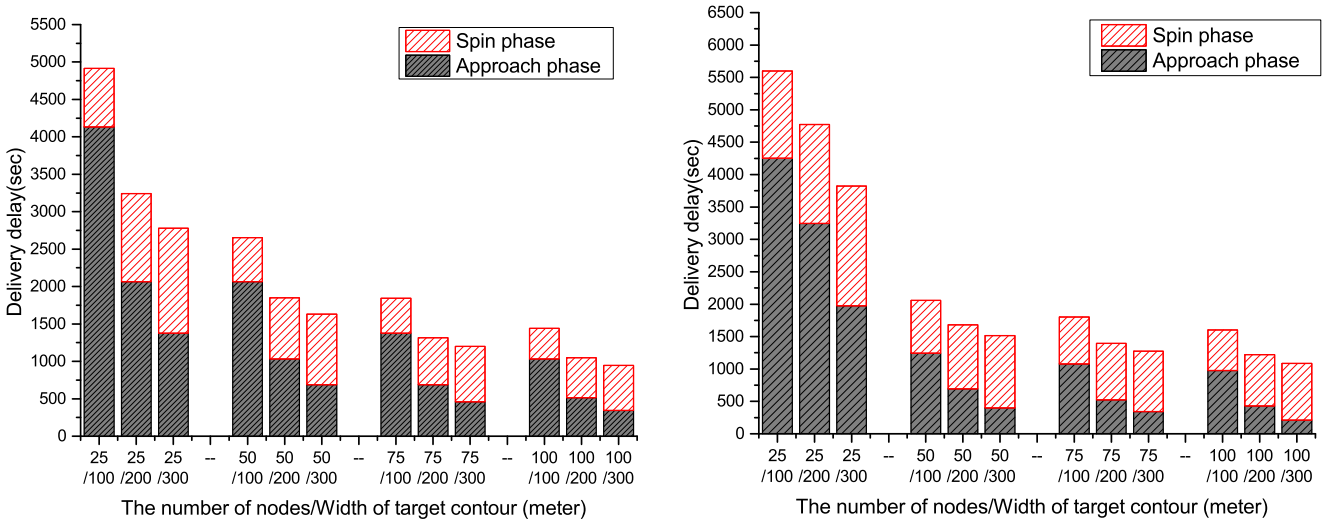
Delivery overhead is measured by counting the expected number of message copies. Modeling the expected number of copies is done by reusing the model that is used to derive the expected delivery ratio. Replicas are only generated in the spin phase, and therefore we consider only the spin phase. Notice that a state transition means that the total number of copies doubles in our model. Based on this property, we find all possible state transitions and count the number of passed states. The expected number of copies can be calculated as follows.

$$\sum_{k=1}^{TTL_h} \left(S(1-S)^{k-1} \times \sum_{i=1}^{TTL_h - k} P_i \cdot (2^i - 1) \right)$$

4 Simulation results using random mobility

We use the ONE Simulator [8]. The simulation map is set to a single cell with $2km$ radius. The number of nodes in the network is maintained equal during simulation. To this end, if a node hits the boundary of the map, the node is reflected back to the map at a random angle. The nodes move according to the random way point model. The node speed is randomly selected from the range of [1.5 ∼ 2.0m/s] and the node pausing time is randomly selected from the range of [1 ∼ 3 seconds]. The transmission range (r) of a node is set to 50m, if not specified otherwise. There is a single IoT gateway, whose location is randomly chosen within the simulation map. A subset of nodes (i.e., IoT devices) generate messages toward the destination node (i.e., IoT gateway). 25 source nodes among all nodes are randomly selected for traffic generation. The size of a message is randomly chosen from the range of [5KB ∼ 10KB], which is consistent with the existing IoT traffic analysis [29].

Two performance metrics are used which are delivery delay and delivery ratio. If the TTL of messages is set to infinity and the size of buffer at each node is large enough



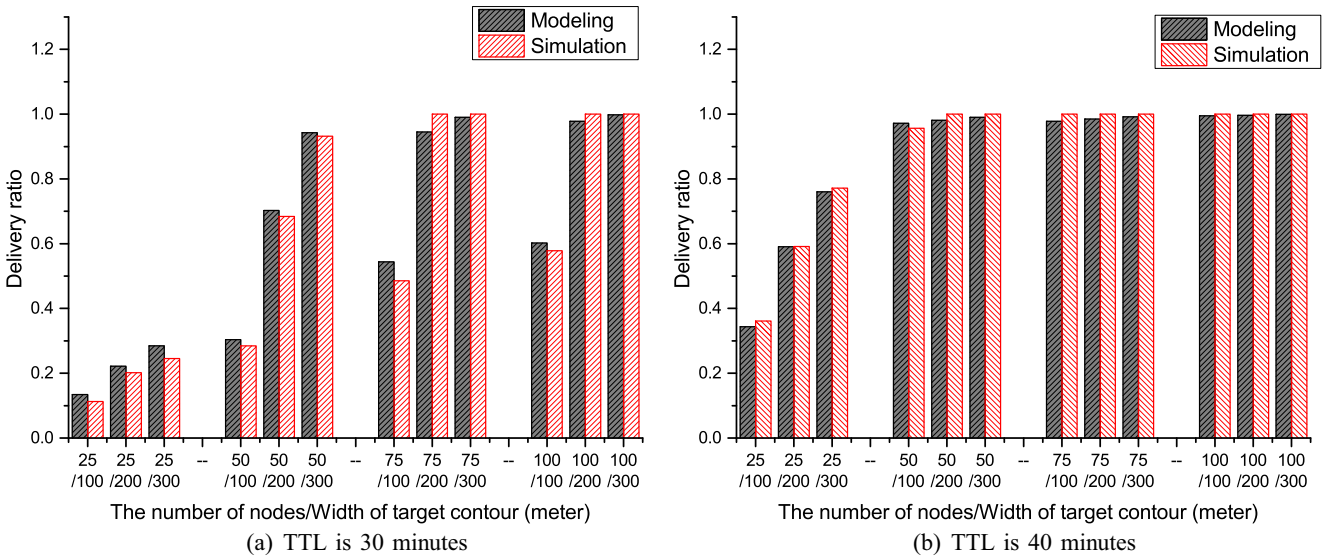
(a) Analytic modeling results (b) Simulation results

Fig. 5 Accuracy of delivery delay modeling

to prevent message loss due to buffer overflow, all messages are eventually delivered to the destination (i.e., the delivery ratio is 100 %). In such a case, delivery delay is the performance metric. If the TTL of messages is a finite value, messages are discarded when their TTL expire. As a result, some messages may not be delivered to the destination. In this case, we use the delivery ratio as the performance metric. The total number of message replicas in the network is used for the overhead metric, which called 'overhead ratio'. For example, the overhead ratio of 20 means that there exist at most 20 message copies in the network.

4.1 Comparison of analytic modeling results and simulation results

We first assess the accuracy of the delivery delay modeling. In Fig. 5a, the analytic modeling results are plotted for various combinations of the number of user nodes and the size of target contour. The corresponding simulation results are plotted in Fig. 5b. TTL of messages is set to infinity. As the number of nodes increases, the delivery delay decreases regardless of the target contour size. Delay decreases slowly as the number of nodes increases. As the target contour



(a) TTL is 30 minutes (b) TTL is 40 minutes

Fig. 6 Delivery ratio modeling and simulation results

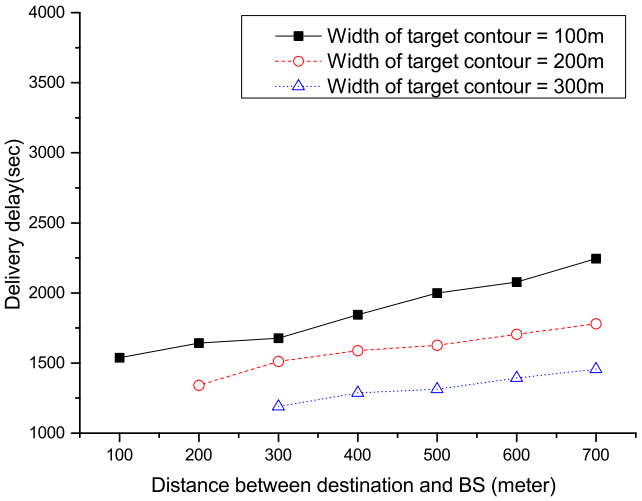


Fig. 7 Delivery delay depending on the location of IoT gateway

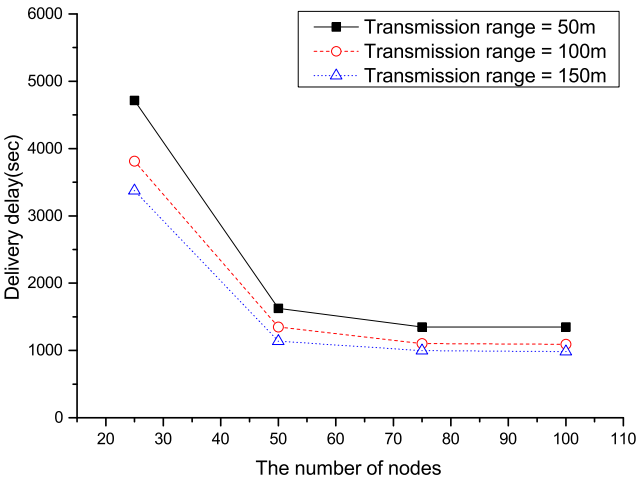


Fig. 9 Delivery delay for various transmission ranges

size increases, the delivery delay decreases regardless of the user density (i.e., the number of user nodes). To get a deeper insight, we divide the delivery delay into two parts: approach phase delay and spin phase delay.

The delivery delay in the approach phase decreases as the number of nodes increases. It is because as the number of nodes increases, the node encounter time decreases and the message is forwarded faster. When the size of target contour increases, the delay in the approach phase also decreases. It is because the distance that a message is forwarded by the approach phase decreases, as the size of target contour increases. The same as the approach phase, as the number of nodes increases, the delay in spin phase decreases. It is because flooding within the target contour speeds up as the node encounter time decreases. As the size of target contour increases, however, the delay in the spin phase increases. It is because the distance between the destination node and the first infected node of the spin phase increases. Overall, increasing the size of target contour reduces the delivery delay, as the delay in the approach phase decreases more sharply than the increase of the spin phase delay.

Now we assess the accuracy of the delivery ratio modeling. We compare the modeling results and the simulation results in Fig. 6a and b. TTL is set to 30 minutes and 40 minutes, in Fig. 6a and b, respectively. Overall, the modeling results match well with the simulation results. Key observations are as follows. As the number of nodes increases, the delivery ratio increases regardless of the target contour size. The delay ratio increases slowly as the number of nodes increases. As the target contour size increases, the delivery ratio increases regardless of the user density. Higher TTL results higher delivery ratio. Under the current simulation setting, TTL of 40 minutes or higher results in near 100 % delivery ratio when more than 50 users exist.

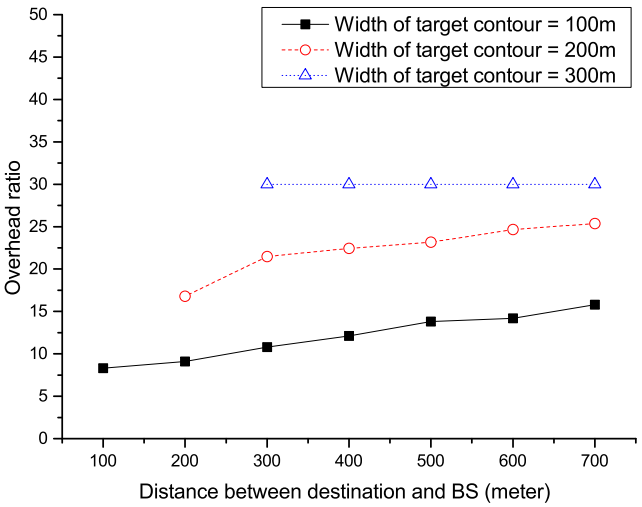


Fig. 8 Overhead ratio depending on the location of IoT gateway

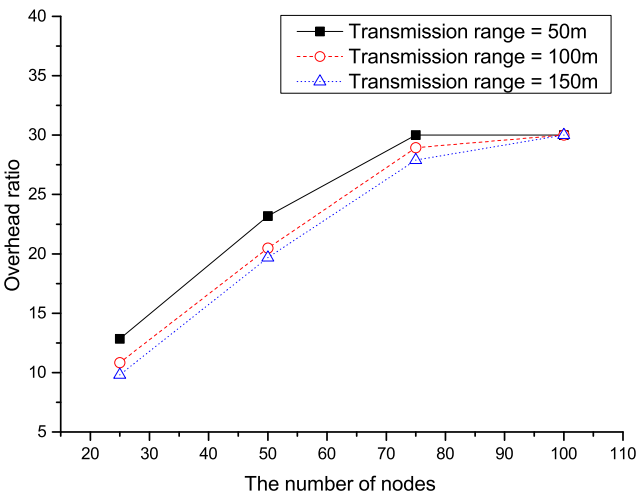


Fig. 10 Overhead ratio for various transmission ranges

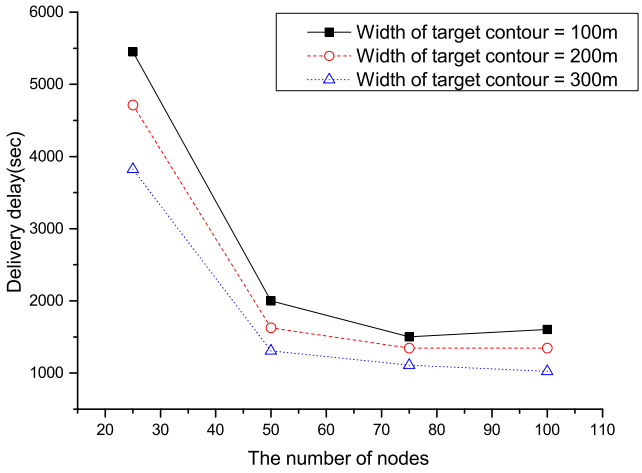


Fig. 11 Delivery delay for different contour sizes

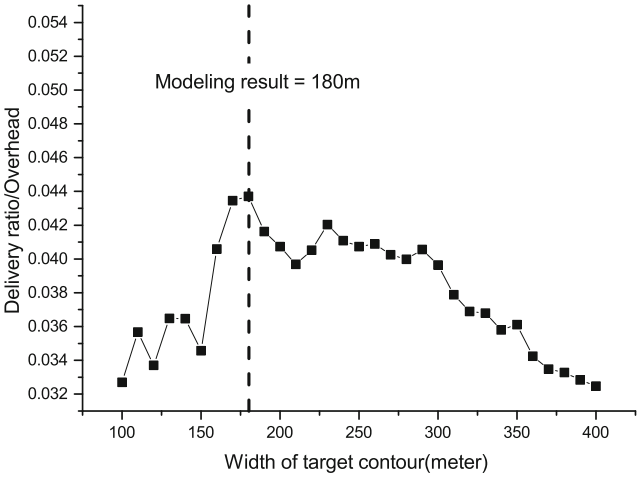


Fig. 13 Optimal target contour width(transmission range=50m)

The comparison of modeling results and simulation results with respect to the delivery overhead also shows good match. We omit the graphs.

4.2 Impact of IoT gateway location

We examine the impact of the location of the IoT gateway (i.e., destination node). The message TTL is set to infinite and the delivery delay is the performance metric. In Fig. 7, the delivery delay is plotted as the location of destination node changes. The X-axis represents the distance of the destination from the BS. The corresponding overhead ratios are plotted in Fig. 8.

The key observation is as follows. Longer delivery delay is obtained, as the destination is farther from the BS. For farther destination from the BS, the traffic overhead also increases. This indicates that the proposed scheme is more effective (i.e., smaller delivery delay with less traffic overhead) if the destination locates near the BS.

4.3 Impact of link transmission range

We examine the impact of the transmission range of peer-to-peer link. So far, we set the transmission range to 50m. Figures 9 and 10 show the results for various transmission ranges. As expected, the delivery delay decreases as the transmission range increases. The decrease of delay, however, is not proportional to the increase of the transmission range. The delay decrease between the transmission range of 100m and 150m is far smaller than that between 50m and 100m. It is because the delay for the spin phase does not decrease proportionally. Once the transmission range is bigger than the contour range, the efficiency of spin phase does not increase significantly even if bigger transmission range

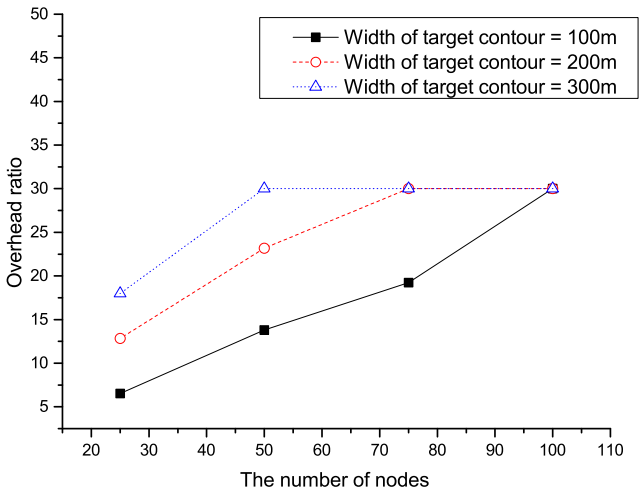


Fig. 12 Overhead ratio for different contour sizes

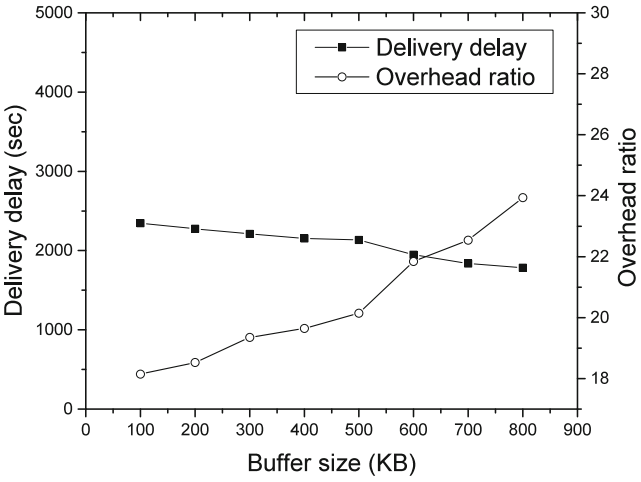


Fig. 14 Delivery delay and overhead ratio for different buffer sizes

Table 2 Simulation result summary(copy limit=30, TTL=30min)

The number of nodes		25	50	75	100
Epidemic	D;O	0.2917;14.8	0.8427;36.5	1.0000;55	1.0000;84.2
	D/O (%)	1.97	2.31	1.82	1.19
Prophet	D;O	0.0800;4.5	0.2000;16.8	0.3400;21.6	0.3800;25
	D/O (%)	1.77	1.19	1.57	1.52
Spray and Wait	D;O	0.2917;14.8	0.7500;30	0.8750;30	0.9512;30
	D/O (%)	1.97	2.50	2.92	3.17
Contour(range:100)	D;O	0.1124;3.2	0.2845;8.7	0.4853;9.8	0.5782;11.2
	D/O (%)	3.51	3.27	5.22	5.11
Contour(range:200)	D;O	0.1823;5.7	0.5475;12.8	0.8234;30	1.0000;30
	D/O (%)	3.20	4.28	2.74	3.33
Contour(range:300)	D;O	0.2147;6.3	0.6842;19	1.0000;30	1.0000;30
	D/O (%)	3.40	3.60	3.33	3.33

Note: Delivery Ratio (D), Overhead (O) and Percentage of Delivery ratio/Overhead ratio (D/O)

is used. A very similar trend is observed from the overhead ratio comparison.

4.4 Impact of target contour width

Now, we examine the impact of target contour width. The simulation result is presented in Fig. 11 for various target contour width (100m, 200m and 300m). The corresponding overhead ratio is plotted in Fig. 12. The key observation is as follows. The approach phase which uses single copy forwarding becomes more effective when the node density is higher (i.e., more user nodes). As a result, when the node density is sufficiently high (e.g., over 50 nodes in our simulations), the difference of delivery delay between small contour range and large contour range becomes much smaller than the difference of corresponding overhead ratio. This observation implies that it is desirable to set the target contour small when the node density is high. In contrast, when the node density is low, it is desirable to set the target contour large.

We can make trade off between the delivery ratio and the overhead ratio by setting the target contour width. Moreover, it is beneficial to identify the optimal target contour width that maximizes the efficiency of the proposed scheme. To estimate the efficiency, we measure the delivery success ratio (denoted by D) and the overhead ratio (denoted by O), and then we compute (delivery ratio/overhead ratio) $\times$ 100, which is denoted by D/O. Higher value of D/O indicates higher efficiency. We examine the relationship between the target contour width and D/O. In Fig 13, the simulation results are given. The transmission range is set to 50m, the number of nodes is set to 50, and TTL is set to 30min. When the width is too small, D/O is low. As the width increases, D/O increases but after a certain optimal point D/O starts to decrease. The simulation results shows that our modeling can accurately estimate the optimal target contour width.

Table 3 Simulation result summary(copy limit=30, TTL=40min)

The number of nodes		25	50	75	100
Epidemic	D;O	0.6125;21.8	1.0000;48.2	1.0000;65	1.0000;100
	D/O (%)	2.80	2.07	1.54	1.00
Prophet	D;O	0.1200;8	0.4230;16.2	0.8100;38.1	0.8390;53.2
	D/O (%)	1.50	2.61	2.12	1.57
Spray and Wait	D;O	0.6125;21.8	1.0000;30	1.0000;30	1.0000;30
	D/O (%)	2.80	3.33	3.33	3.33
Contour(range:100)	D;O	0.3609;9.5	0.9565;24.7	1.0000;30	1.0000;30
	D/O (%)	3.78	3.87	3.33	3.33
Contour(range:200)	D;O	0.5913;14.2	1.0000;30	1.0000;30	1.0000;30
	D/O (%)	4.16	3.33	3.33	3.33
Contour(range:300)	D;O	0.7712;18.3	1.0000;30	1.0000;30	1.0000;30
	D/O (%)	4.21	3.33	3.33	3.33

Table 4 Simulation result summary(copy limit=25, TTL=30min)

The number of nodes		25	50	75	100
Epidemic	D;O	0.2917;14.8	0.8427;36.5	1.0000;55	1.0000;84.2
	D/O (%)	1.97	2.31	1.82	1.19
Prophet	D;O	0.0800;4.5	0.2000;16.8	0.3400;21.6	0.3800;25.1
	D/O (%)	1.77	1.19	1.57	1.52
Spray and wait	D;O	0.2917;14.8	0.7132;25	0.875;25	0.9512;25
	D/O (%)	1.97	3.00	3.50	3.80
Contour(range:100)	D;O	0.1124;3.2	0.2845;8.7	0.4017;9.8	0.4987;11.2
	D/O (%)	3.51	3.27	4.19	4.45
Contour(range:200)	D;O	0.1823;5.7	0.4789;12.8	0.8134;25	0.9458;25
	D/O (%)	3.20	3.74	3.25	3.78
Contour(range:300)	D;O	0.2147;6.3	0.6842;19	0.8700;25	1.0000;25
	D/O (%)	3.41	3.60	3.48	4.00

4.5 Impact of node buffer size

IoT device has relatively low computing power and small buffer size. Here we examine the impact of buffer size of the IoT devices on the performance of our scheme. In today's commercial IoT devices, various size of flash memory is equipped. Various size flash memory is e to the commercialized IoT device. Some IoT device has up to 5MB flash memory while other cheaper IoT device has only 512KB of memory. We simulate delivery delay and overhead ratio by varying the buffer size in the range of [100KB∼1MB]. Figure 14 shows the results. As expected, the delivery delay increases as buffer size increases. It is because buffer overflow interferes message spreading in the spin phase. For the same reason, the overhead ratio increases as buffer size decreases. But the impact of the buffer size on performance is not substantial with reasonable buffer size.

4.6 Comparison with existing DTN routing schemes

Now, we compare the efficiency of the proposed scheme with existing DTN routing schemes. Chosen are three well-known schemes; Epidemic routing [7], Prophet routing [13] and Spray and Wait routing [10]. We here set the message TTL to finite values, imposing the message delivery deadline.

In Table 2, the simulation results are summarized when TTL is set to 30min. For both the proposed scheme (marked as 'Contour') and the Spray and Wait scheme, the copy limit is set to 30 while Epidemic routing does not have the copy limit. Not surprisingly, Epidemic routing results the highest delivery ratio, but its overhead ratio is also the highest. Regardless of the selection of target contour size, the proposed scheme clearly outperforms the existing schemes under comparison. That is, the proposed scheme results much higher D/O values.

Table 5 Simulation result summary(copy limit=25, TTL=40min)

The number of nodes		25	50	75	100
Epidemic	D;O	0.6125;21.8	1.0000;48.2	1.0000;65	1.0000;100
	D/O (%)	2.80	2.07	1.54	1.00
Prophet	D;O	0.1200;8	0.4230;16.2	0.8100;38.1	0.8390;53.2
	D/O (%)	1.50	2.61	2.12	1.57
Spray and Wait	D;O	0.6125;21.8	0.8963;25	0.9063;25	0.9234;25
	D/O (%)	2.80	3.58	3.63	3.70
Contour(range:100)	D;O	0.3609;9.5	0.8135;24.7	0.8026;25	0.8146;25
	D/O (%)	3.80	3.29	3.30	3.26
Contour(range:200)	D;O	0.5913;14.2	0.8247;25	0.8912;25	0.8812;25
	D/O (%)	4.16	3.30	3.56	3.52
Contour(range:300)	D;O	0.6712;18.3	0.9247;25	0.9475;25	0.9575;25
	D/O (%)	3.67	3.70	3.80	3.98

Table 6 Simulation result summary(copy limit=30, TTL=30min, IoT gateway at 200m from BS)

The number of nodes		25	50	75	100
Epidemic	D;O	0.3001;14.8	0.8581;36.5	1.0000;55	1.0000;84.2
	D/O (%)	2.03	2.35	1.82	1.19
Prophet	D;O	0.1600;8.2	0.2000;12.3	0.3390;26.1	0.4110;28.8
	D/O (%)	1.95	1.53	1.30	1.43
Spray and wait	D;O	0.2980;14.8	0.7712;30	0.8917;30	0.9631;30
	D/O (%)	2.01	2.57	2.97	3.21
Contour(range:100)	D;O	0.2047;3.0	0.3481;8.0	0.5423;9.2	0.7523;11.0
	D/O (%)	6.82	4.35	5.89	6.84
Contour(range:200)	D;O	0.5314;5.1	0.7900;12.1	1.0000;30	1.0000;30
	D/O (%)	10.42	6.53	3.33	3.33
Contour(range:300)	D;O	0.7120;5.9	1.0000;18	1.0000;30	1.0000;30
	D/O (%)	12.07	5.55	3.33	3.33

The simulation results when TTL is set to 40min are summarized in Table 3. In this case, if the number of nodes is more than 50, the delivery ratio approaches to 100 % for most schemes, and D/O of the proposed scheme becomes the same as that of the Spray and Wait scheme. This indicates that if TTL is sufficient large and there exist enough nodes, messages will eventually be delivered. But when the number of nodes is small, our scheme outperforms other schemes. When the number of nodes is large, the proposed scheme can afford to use a smaller copy limit without sacrificing the delivery ratio, which will increase the D/O.

In Tables 4 and 5, we reduce the copy limit from 30 to 25. Because the maximum number of copies decreases, the delivery ratio and overhead ratio also decrease. Nevertheless, the proposed scheme clearly outperforms the other schemes. The proposed scheme has two configurable parameters: the size of target contour and maximum number of copies. By properly selecting these parameters, we can maximize the routing efficiency. Even when we use the same copy limit as the Spray and Wait scheme, we can still achieve higher efficiency.

The efficiency of the proposed scheme substantially increases when the IoT gateway is near the BS. In Tables 6 and 7, simulation results are summarized. When the destination is near the BS, the proposed scheme is particularly effective in a sparsely populated environment (i.e., the number of nodes is relatively small). Target contour of small size yields higher D/O. For example, in Table 6, the proposed scheme produces about 3.6 times higher D/O than the Spray and Wait scheme when the number of nodes is 25.

5 Simulation results using real mobility traces

We evaluate the performance of our scheme with a real mobility trace (NCSU trace). In this trace, the movement of a total of 20 students are recorded for about three months. For our simulation, we apply the SWIM tool [23] to this

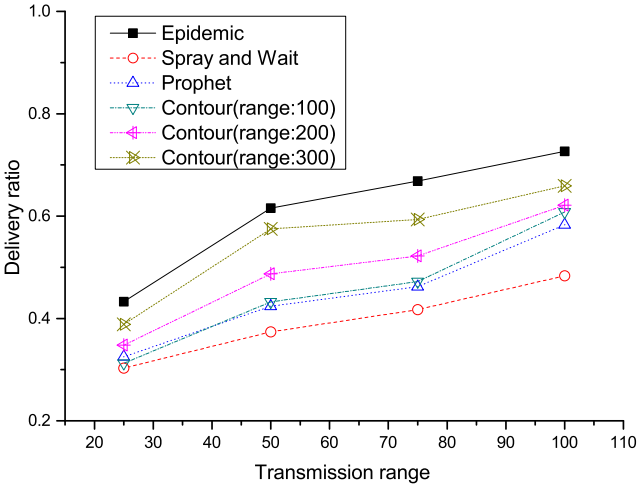
Table 7 Simulation result summary(copy limit=30, TTL=40min, IoT gateway at 200m from BS)

The number of nodes		25	50	75	100
Epidemic	D;O	0.6390;21.8	1.0000;48.2	1.0000;65	1.0000;100
	D/O (%)	2.93	2.07	1.54	1.00
Prophet	D;O	0.2800;13.1	0.4000;22.6	0.6820;36.3	0.7900;58.9
	D/O (%)	2.13	1.77	1.88	1.34
Spray and wait	D;O	0.6260;21.8	1.0000;30	1.0000;30	1.0000;30
	D/O (%)	2.87	3.33	3.33	3.33
Contour(range:100)	D;O	0.4260;9.0	1.0000;21.6	1.0000;30	1.0000;30
	D/O (%)	4.73	4.63	3.33	3.33
Contour(range:200)	D;O	0.8623;13.9	1.0000;29.3	1.0000;30	1.0000;30
	D/O %)	6.20	3.41	3.33	3.33
Contour(range:300)	D;O	0.9513;17.8	1.0000;30	1.0000;30	1.0000;30
	D/O (%)	5.34	3.33	3.33	3.33

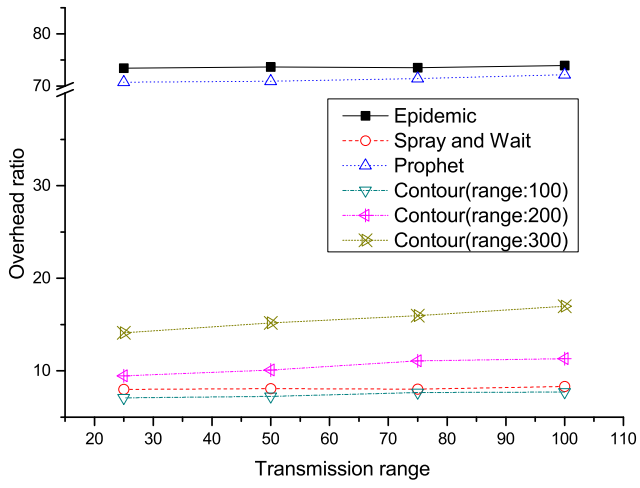
trace. SWIM generates synthetic traces from the original trace. For realistic trace generation, SWIM fits the model of inter-contact rate and contact duration using the statistic properties of the original trace based on the community-based mobility model [24, 25]. As a result, we transform the original trace to many traces with various number of users up to 99 users. The traces generated by SWIM is used in the ONE simulator. Other simulation setting is the same as in Section 4.

The simulation results are given in Figs. 15 and 16. The number of users and the transmission range of the peer-to-peer link are varied. The results show that the proposed scheme clearly outperforms the existing schemes and the same trend as in the simulations using the RWP mobility can be observed.

Finally, we examine how accurately our model can identify the optimal target contour width. Figure 17 shows the optimal target contour widths of the simulation results match well with values obtained from the mathematical model even if the trace does not strictly follow the random way point mobility model.

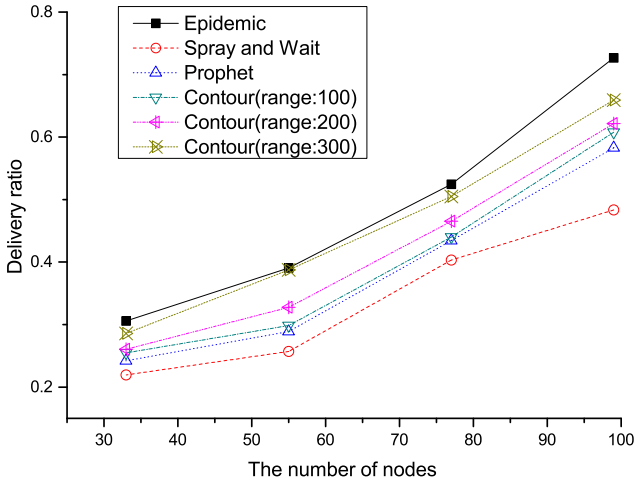


(a) Delivery ratio

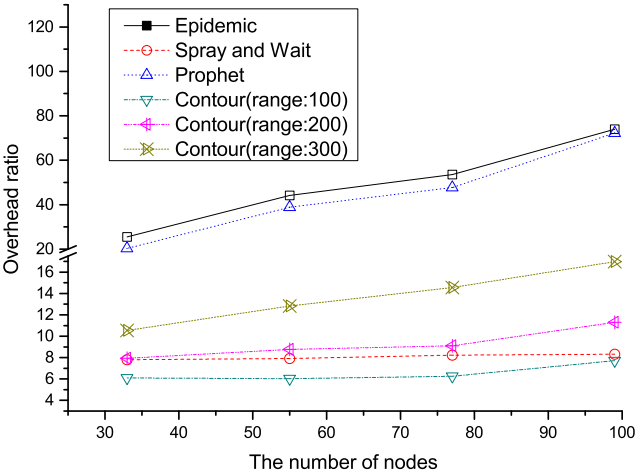


(b) Overhead ratio

Fig. 15 Simulation results for various transmission ranges using the NCSU trace (number of users=99)



(a) Delivery ratio



(b) Overhead ratio

Fig. 16 Simulation results for various number of nodes using the NCSU trace (transmission range=100m)

6 Extension for multi-cell environments

So far, we have considered the case that IoT devices and the IoT gateway are in the same cell. In this section, we extend our scheme for the scenario that IoT devices and IoT gateway are in different cells. In such a case, we utilize the concept of 'home cell' (or 'home community), which has been exploited by many existing social-based DTN routing schemes. For example, Label routing [16] is built on the premise that people in the same community meet each other more frequently. If you pass a message to a node that

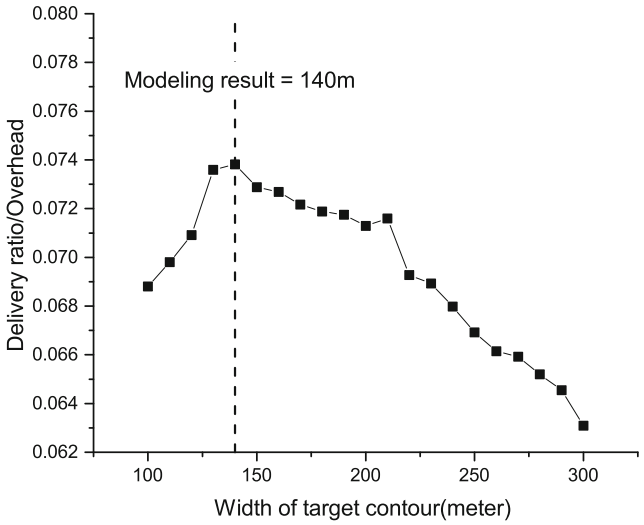


Fig. 17 Optimal target contour width for the NCSU trace (transmission range=50m)

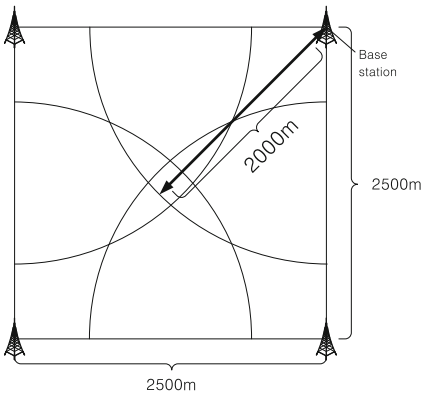


Fig. 18 Four-cell simulation map

belongs to the same community as the destination, the message is more likely to be delivered to the destination. SimBet Routing [18] uses the so-called 'betweenness centrality and similarity' which express popularity and friendship among nodes Using these properties, each node recognizes its own social community and identifies the bridge nodes which can relay the message between communities. Bubble Rap Forwarding [17], an extension of SimBet routing, is another example that utilizes the home community concept.

In multi-cell environments, the proposed scheme works as follows. When an IoT device is in a different cell from the IoT gateway, the home cell information of the mobile nodes is used for the forwarding decision. That is, when a node that carries the data meets another node, they exchange their home cell information. Then, the proximity of the home cells of the two nodes to that of the cell where IoT gateway locates is compared. The data is forwarded to the encountered node, if the new node's home cell is closer to the IoT gateway cell than the current node with data. Once a node with data arrives at the cell where IoT gateway locates, the basic version of the proposed scheme is executed.

To assess the efficiency of our scheme in the multi-cell scenario, we performed simulations on a four-cell simulation map which is shown in Fig. 18. The network area is $2.5km \times 2.5km$, where four cells are located at the corners of the map and the radius of each cell is $2km$. The NCSU trace is used for the simulation in the same way as in Section 5. The cell in which a user has the longest staying time is chosen as the home cell of the user. We randomly choose the location of the stationary IoT gateway in the simulation. In the simulation, the number of nodes and the transmission range of the peer link are varied. Message size is randomly chosen from [5KB $\sim$ 10KB]. Buffer size is set to a high value. The simulation results are given Figs. 19 and 20. It can be observed that the proposed scheme outperforms the existing schemes.

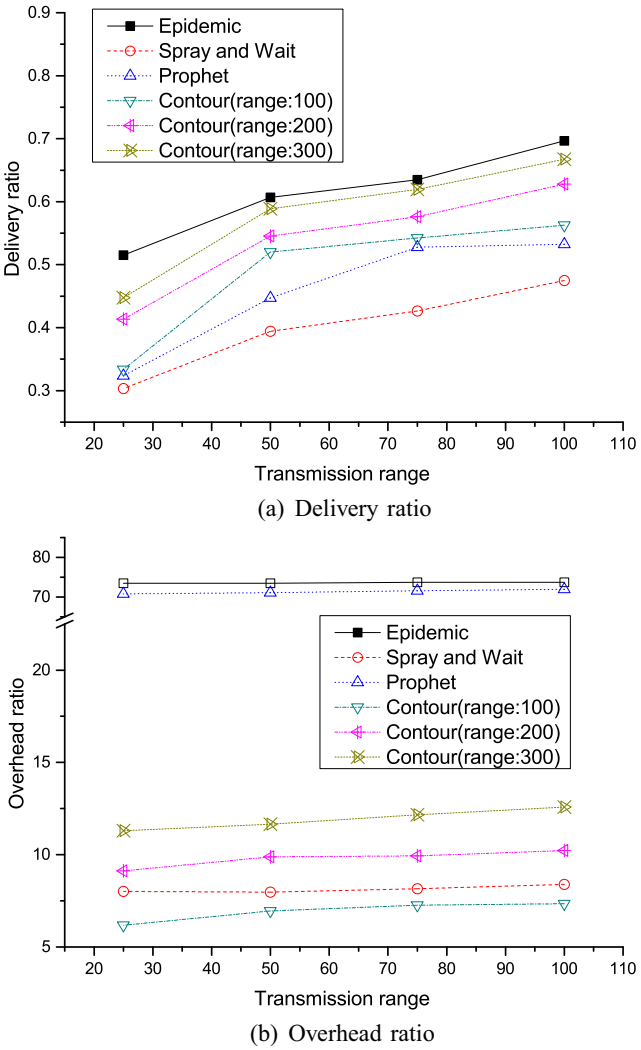
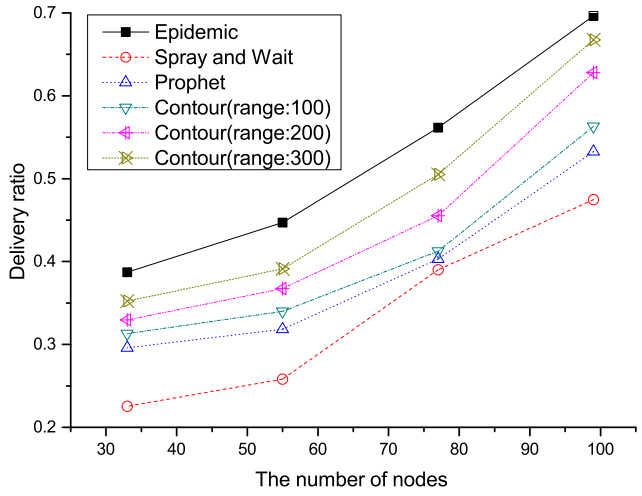
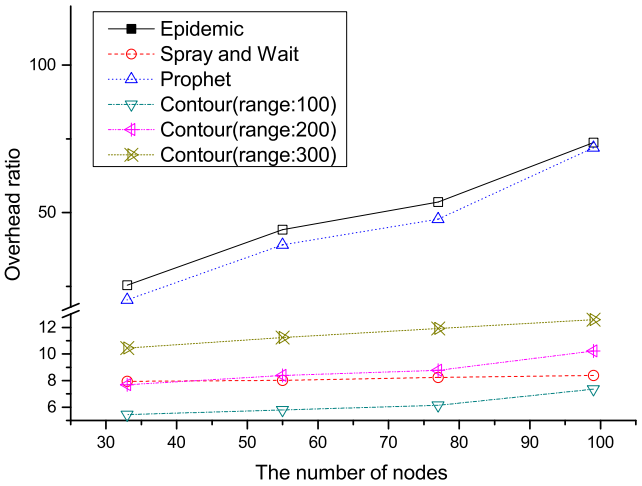


Fig. 19 Simulation results for various transmission ranges in multi-cell environment using NCSU trace (number of users=99)



(a) Delivery ratio



(b) Overhead ratio

Fig. 20 Simulation results for various number of nodes in multi-cell environment using NCSU trace (transmission range=100m)

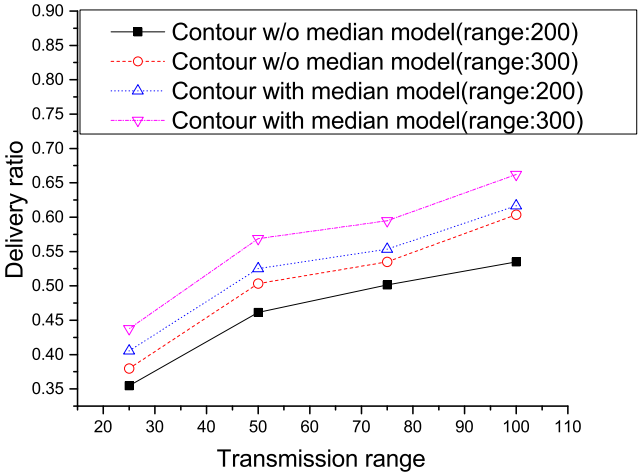
7 RSSI measurement instability

In the real world, RSSI measurement is oftentimes unstable due to such issues as multipath propagation, fading, etc. The inaccurate RSSI measurement can result two undesirable effects. First, the ping-pong effect occurs near the boundary of the target contour. Second, node makes a wrong decision of the next hop selection (i.e., message relay to a node which is farther from destination than the current node). As this phenomenon impedes efficient routing decision, the overhead ratio may significantly increase.

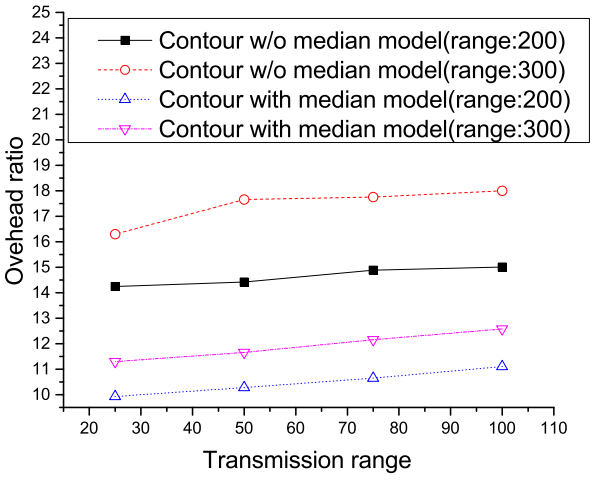
To overcome the RSSI instability, RSSI calibration by manipulating the radio channel model has been proposed. However the existing RSSI calibration algorithms are too complex to apply in IoT device that usually has low computing power and limited battery. Instead, we adopt a simple median method. Median model works as follows. Each node i measure RSSI values n times continuously. After that, obtained RSSI values are sorted in the increasing order. Sorted RSSI value set is $RSSI_i^{sorted} = \{RSSI_i^1, RSSI_i^2, \cdots, RSSI_i^n\}$. $RSSI_i^1$ is the smallest obtained RSSI value and $RSSI_i^n$ is the largest obtained RSSI value. Then, the median value shown by the Eq. 3 is chosen.

$$RSSI_i = \begin{cases} RSSI_i^t & \text{if n is an even number, t=n/2,} \\ (RSSI_i^t + RSSI_i^{t+1})/2 & \text{if n is an odd number, t=(n+1)/2.} \end{cases} \quad (3)$$

In the simulation, we use the RSSI fluctuation model based on a Gaussian distribution which is described in [30]. When a node needs a RSSI value, a RSSI value is picked from the Gaussian distribution whose mean value is the



(a) Delivery ratio



(b) Overhead ratio

Fig. 21 Simulation results for the propose scheme with and without(w/o) median model in multi-cell environment using NCSU trace (number of users=99)

accurate RSSI value and the standard deviation is set to 1. To assess the efficiency of the median model, we use the multi-cell environment the same as in Section 6. We compare the two cases when the RSSI calibration model is applied or not. The simulation results are given in Fig. 21. When the RSSI calibration is not activated, delivery ratio decreases and the overhead ratio increases significantly. After the median model is applied, delivery ratio and the overhead ratio show similar values when RSSI fluctuation does not exist (i.e., the results in Section 6).

8 Conclusion

In this paper, we propose a sensing data delivery scheme from IoT devices to the IoT gateway. Peer-to-peer DTN routing is combined with the location information that is available from the cellular network. The key idea is the use of RSSI information, incomplete but inexpensive location information, of the nodes. Our scheme is designed to complement a more reliable but more expensive IoT data delivery solution (i.e., MTC cellular).

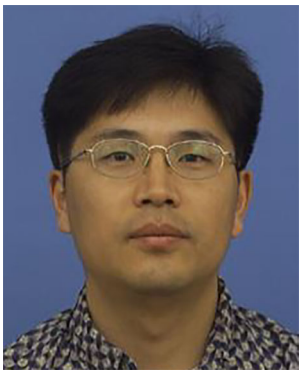
Our scheme utilizes the location information to cleverly combine the single copy relaying and multi-copy limited flooding. The timing of the conversion from the single copy relaying to limited flooding is critical to the effectiveness of the proposed scheme. The simulation results indicate that our analytic model can quite accurately estimate this conversion timing even in case of real trace-based simulations. Our scheme is particularly effective when the node density is high and the destination locates near the base station. We show that our scheme clearly outperforms the existing DTN routing schemes via extensive simulations.

Acknowledgments This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MEST) (NRF-2013R1A2A2A01068325).

References

1. 3GPP TS 23.682 V.13.3.0, Architecture enhancements to facilitate communications with packet data networks and applications, 3GPP, 2015
2. R2-104662: MTC simulation results with specific solutions, presented at the 3GPP TSG RAN WG2 Meeting 71, Madrid, Spain, 2010
3. Cheng RG, Wei CH, Tsao SL, Ren FC (2012) RACH collision probability for machine-type communications. In: Proceedings IEEE VTC spring
4. Guo B, Zhang D, Wang Z, Yu Z, Zhou X (2013) Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things. Elsevier J Netw Comput Appl 36(6):1531–1539
5. Al-Turjman FM et al (998) A delay-tolerant framework for integrated RSNs in IoT. Elsevier Comput Commun 36(9)
6. Spyropoulos T, Psounis K, Raghavendra CS (2006) Performance analysis of mobility-assisted routing. In: Proceedings ACM MobiHoc
7. Vahdat A, Becker D (2000) Epidemic routing for partially connected ad hoc networks Technical Report CS-200006. Duke University
8. Keranen A, Ott J, Karkkainen T (2009) The ONE simulator for DTN protocol evaluation. In: Proceedings ACM SIMUTools
9. Jain S, Fall K, Patra R (2004) Routing in a delay tolerant network. In: Proceedings ACM SIGCOMM
10. Spyropoulos T, Psounis K, Raghavendra CS (2005) Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In: Proceedings ACM WDTN
11. Spyropoulos T, Psounis K, Raghavendra CS (2007) Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In: Proceedings IEEE PerCom Workshop
12. Nelson S, Bakht M, Kravets R, Harris AF (2009) Encounter-based routing in DTNs. In: Proceedings IEEE INFOCOM
13. Lindgren A, Doria A, Schelén O (2003) Probabilistic routing in intermittently connected networks. In: Proceedings ACM SIGMOBILE
14. Henrksson D, Abdelzaher T, Ganti RK (2007) A caching based approach to routing in delay-tolerant networks. In: Proceedings IEEE ICCCN
15. Spyropoulos T, Psounis K, Raghavendra CS (2004) Single-copy routing in intermittently connected mobile networks. In: Proceedings IEEE SECON
16. Hui P, Crowcroft J (2007) How small labels create big improvements. In: Proceedings IEEE PERCOM
17. Hui P, Crowcroft J, Yoneki E (2008) BUBBLE Rap: Social-based forwarding in delay tolerant networks. In: Proceedings ACM MobiHoc
18. Daly EM, Haahr M (2007) Social network analysis for routing in disconnected delay-tolerant manets. In: Proceedings ACM MobiHoc
19. Thompson N, Nelson S, Bakht M, Abdelzaher T, Kravets R (2010) Retiring replicants: congestion control for intermittently-connected networks. In: Proceedings IEEE INFOCOM
20. Rhee I, Shin M, Hong S, Lee K, Kim S, Chong S (2009) {CRAWDAD} trace ncsu/mobilitymodels/GPS/NCSU(v.2009-07-23)
21. Zhu Q, Wang R, Chen Q, Liu Y, Qin W (2010) Iot gateway: Bridgingwireless sensor networks into internet of things. In: Proceedings IEEE EUC
22. Guoqiang S, Yanming C, Chao Z, Yanxu Z (2013) Design and implementation of a smart IoT gateway. In: Proceedings IEEE GreanCom and iThings/CPSCom
23. Mei A, Stefa J (2009) SWIM: A simple model to generate small mobile worlds. In: Proceedings IEEE INFOCOM
24. Musolesi M, Mascolo C (2006) A community based mobility model for ad hoc network research. In: Proceedings ACM REALMAN
25. Boldrini C, Passarella A (2010) HCMM: Modelling spatial and temporal properties of human mobility driven by users' social relationships, vol 33, pp 1056–1074
26. Huo G, Wang X (2008) An opportunistic routing for mobile wireless sensor networks based on RSSI. In: Proceedings IEEE WICOM
27. Mei H, Jiang P, Bigham J (2011) Augmenting coverage in a cellular network with DTN routing. In: Proceedings IEEE WCNC
28. Kim S, Han S (2012) Contour routing for peer-to-peer DTN delivery in cellular networks. In: Proceedings IEEE COMSNETS
29. Orrevad A (2009) M2M traffic characteristics. KTH Information and Communication Technology, Stockholm, Sweden
30. Sichitiu ML, Ramadurai V (2004) Localization of wireless sensor networks with a mobile beacon. In: Proceedings IEEE MASS

Sun-Hyun Kim received the B.S. degree in Computer Science from the Yonsei University Seoul, Korea in 2010. Currently, he is a Ph.D. student of the Department of Computer Science in the Yonsei University, Seoul, Korea. His research interest includes Delay-Tolerant Network, Opportunistic Network and Internet of Things (IoT).



Seung-Jae Han received the B.S. and M.S. degrees in computer engineering from Seoul National University, Seoul, Korea, and the Ph.D. degree in Computer Science and Engineering from the University of Michigan, Ann Arbor. He was a Member of Technical Staff of the Wireless Research Laboratory, Bell Laboratories, Alcatel-Lucent, Murray Hill, NJ. Currently, he is an professor in the computer science department of Yonsei University, Seoul, Korea. His research interests include mobile networking, wireless Internet, and network management.