# Mobility Helps Data Delivery in Disruption Tolerant Networks

Kaoru Sezaki[1], Niwat Thepvilojanapong[2], and Yoshito Tobe[3]

[1] Center for Spatial Information Science, University of Tokyo
4–6–1 Komaba Meguro Tokyo 153–8505 Japan
`sezaki@iis.u-tokyo.ac.jp`
[2] Department of Information and Communication Engineering, University of Tokyo
7–3–1 Hongo Bunkyo Tokyo 113–8654 Japan
`wat@acm.org`
[3] Department of Info. Systems and Multimedia Design, Tokyo Denki University
2–2 Kanda-Nishikicho Chiyoda Tokyo 101–8457 Japan
`yoshito_tobe@osoite.jp`

**Abstract.** Sensor networks using mobile robots have recently been proposed to deal with data communication in disruption tolerant networks (DTNs) where an instantaneous end-to-end path between a source and destination may not exist. In such network scenarios, a node should move to deliver data to the destination. In this paper, we study adaptive formations of mobile robots based on the knowledge of network topology held by each node. Different node formations are applied when nodes have neighboring, clustering, or perfect information of network. Node formations also depend on traffic patterns, e.g., single and multiples packets per event. We introduce a straight line formation called pipeline for delivering multiple packets continuously. The benefit of controlled mobility in DTNs is validated through the ns–2 simulation tool by comparing with the ideal cases.

**Keywords:** routing, mobile sensor networks, disruption tolerant networks, mobile robots, evaluation, simulation.

## 1 Introduction

As technology rapidly advances, diverse sensing and mobility capabilities will become more readily available to devices. Once mobility becomes feasible, we fully expect that large systems of mobile autonomous agents performing various important tasks will soon follow. Of course, communication will be an essential function of these *Mobile Sensor Networks (MSNs)*. The objective of this paper is to explore the capability of such networks using controlled mobility to help data delivery.

One can envision many settings where mobility may potentially be used to improve network communications. One such scenario is a long term deployment of self-organizing mobile sensors designed to intercept or record, and then report

H. Ichikawa et al. (Eds.): UCS 2007, LNCS 4836, pp. 298–305, 2007.
© Springer-Verlag Berlin Heidelberg 2007

as much data as possible. One promising scenario is the use of sensing robots to conduct difficult or dangerous tasks that cannot be done by human beings. The movement of mobile robots is a key consideration in constructing an efficient network and performing sensing adaptive to important events in a wider area. Sensor networks that employ mobile robots have recently been proposed to conduct more efficient sensing tasks in disruption tolerant networks (DTNs) [1,2] where an instantaneous end-to-end path between a source and destination may not exist. Path disruption could exist because tiny-sized devices cannot offer very long radio range. In such network scenarios, a node should move to deliver data to the destination.

This paper extensively studies *WISER (Wireless Interactive SEnsing Robot)* protocol [3], and then validates the benefit of mobility through simulations. One of the major features of WISER is the criteria for selecting between physical movement and wireless transmission for data transfer at each hop. WISER determines the formation of mobile sensing nodes depending on available information about network topology. Unlike the movement in mobile ad hoc networks (MANETs), a node moves intentionally to transfer data. Although an end-to-end path does not exist, a WISER node can control not only its movement but also the mobility of other mobile sensors to make communication both feasible and efficient. Such explicit collaboration of nodes makes data transfer in partitioned networks possible.

The remainder of the paper is organized as follows. Section 2 introduces some related work. Section 3 describes problem statement. Section 4 articulates the WISER protocols. Section 5 evaluates the benefit of mobility through simulations. We conclude our work in Section 6.

## 2 Related Work

GPSR [4] is a geographic routing that uses location information to decrease overhead of route discovery. GPSR uses greedy forwarding at each hop by routing each packet to the neighbor closest to the destination. However, any location-based routing (LBR) protocols do not work well in DTNs. To solve the problem of LBR protocols, previous works exploited mobility when nodes meet each other by chance, i.e., natural and uncontrolled movement. Epidemic routing [1] that relies on the theory of epidemic algorithms is a routing protocol for intermittently connected networks. Similarly, moving objects in [5] move randomly and disseminate data across disconnected static nodes by trying to distribute data to all nodes. In [6,7], randomly moving humans and animals act as data mules and collect data opportunistically from sensor nodes when in range. Such approaches are orthogonal to our work which exploits controlled movement.

Message ferrying [2] uses non-randomness movement which is known by other nodes to help deliver data. Goldenberg et al. [8] also exploits controlled mobility to improve communication performance. The Infostations model [9] of wireless ad-hoc networks aims to provide trade-offs between delay and capacity of these networks by providing geographically intermittent connectivity. Fall [10] proposed the Delay Tolerant Network architecture to solve the internetworking issues in scenarios where partitions are frequent and a connected path between message senders and receivers may be not present. This approach relies on routing mechanism presented in detail in [11].

## 3 Problem Statement

We consider a sparse sensor network where end-to-end path is not available for all pairs of nodes in the network. A node or robot composes of sensing and moving components, and it is called mobile robot, mobile sensor, or sensor node throughout the paper. The mobile robot is also equipped with a short-range wireless communication interface (RF). Each node identifies its location by using GPS or other means of positioning systems. The location of sink node or data collector is known in advance, while the location of other nodes can be obtained by beacon messages. Our study divides knowledge of location information into three levels. First, *neighboring information* is location information of all neighboring nodes. Second, *clustering information* means location information of all nodes in a cluster, where the cluster means a group of nodes that can reach each other using multi-hop, wireless communication. Third, *perfect information* is location information of all nodes in a network. We propose different protocols to cope with each level of location information.

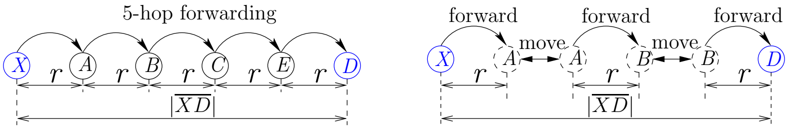
We consider two traffic models in our study. First, each node generates one packet per event in a *single-packet model*. The event may be warning, notice, or any sensed data detected in tracking systems, intrusion detection, etc. Second, a node reports sensed data periodically for a fixed period of time in a *multiple-packet model*. Actually, sensing data may not fit into a single packet and need fragmentation in this model. The protocol for each level of location information is further divided into two cases according to our traffic models.

## 4 Data Delivery Protocols for DTNs

### 4.1 The WISER Protocol

**Single Data Packet.** Basically, a node uses the *greedy forwarding (GF) scheme* [4] whenever possible. In particular, next hop is a neighboring node geographically closest to the destination $D$. If the GF scheme fails, the node moves towards $D$ for a distance of its communication range ($r$). After its journey, the node discovers new neighbors and decides the next hop based on the GF scheme. The process of moving $r$ meters is repeated until the GF scheme is valid. After transmitting data to the next hop, the node moves back to its original position.

**Multiple Data Packets.** Data delivery within a cluster uses the GF scheme. When the GF fails, nodes form a straight line formation called *pipeline* between partitioned clusters. In particular, current node $X$, which is the node geographically closest to destination $D$ in the cluster, moves towards $D$ to check the



(a) Standstill pipeline (b) Moving pipeline

**Fig. 1.** Robots construct a pipeline collaboratively

number of potential *helper-nodes* ($m$). $X$ then calculates whether $m$ is enough to construct a *standstill pipeline* (Fig. 1a) by using the following inequality: $m \geq |\overrightarrow{P_D} - \overrightarrow{P_X}|/r - 1$. If $m$ is enough (the inequality is true), we use minimum number of helper-nodes, i.e., $m = |\overrightarrow{P_D} - \overrightarrow{P_X}|/r - 1$, where each helper-node is separated equally as shown in Fig. 1a. After deciding the helper-nodes and their positions, $X$ requests the helper-nodes to move to the defined position during its journey back to the original position. Therefore, the total delay ($\Delta t$) of the first packet using the standstill pipeline composes of communication ($|\overrightarrow{P_D} - \overrightarrow{P_X}|(d + s/b)/r$) and moving ($2|\overrightarrow{P_D} - \overrightarrow{P_X}|/v$) delays, where $d$ is transmission delay, $s$ is packet size, $b$ is bandwidth, and $v$ is node velocity. However, it costs only communication delay for the second and following packets.

If the number of helper-nodes ($m$) is not enough, we introduce a *moving pipeline* (Fig. 1b). Helper-nodes need to move forth-and-back to get, carry, and forward data packets. Each node should move forth-and-back the same distance. The total delay excluding round-trip movement (helper-node discovery) of $X$ also composes of communication ($(m+1)(d+s/b)$) and moving ($|\overrightarrow{P_D} - \overrightarrow{P_X}|/v - r(m+1)/v$) delays.

### 4.2 The WISER/c Protocol: Beyond Neighboring Information

**Single Data Packet.** Each node calculates a minimum-hop path in its own cluster by using Dijkstra's algorithm. A temporary destination in the cluster is the node geographically closest to the destination. The time ($\Delta t$) needed for a packet traversing $k$ hops from a current node to the temporary destination is $\sum_{i=0}^{k-1}(d + s/b)$. Assume transmission ($d$) and propagation ($s/b$) delays are equal for the same-sized packet at any wireless link, the total delay becomes $k(d+s/b)$.

Because wireless communication is prone to loss, nodes should include loss rate when determining a path. Let $(1-p)$ be the probability that a packet loses when being delivered to the next hop. If the packet loses, the sender retransmits the packet until it is successfully delivered to the next hop. The expectation value of one-hop delay from node $i$ to node $j$ is $E[\Delta t_{ij}] = \sum_{k=1}^{\infty} p_i(1-p_i)^{k-1}(k\Delta t_{ij}) = \Delta t_{ij}/p_i$. Therefore, the delay taking link quality into consideration is $\Delta t = (d + s/b)\sum_{i=0}^{k-1} p_i^{-1}$.

When the next hop does not exist (i.e., the packet arrived the temporary destination), the temporary destination uses the same procedures as the WISER protocol (Sect. 4.1) by moving $r$ meters towards final destination. Instead of using the GF, it selects the next hop whose cluster is geographically closest to the destination.

**Multiple Data Packets.** A node uses the same algorithm as the single-packet scenario to discover the next hop within a cluster. If the next hop does not exist, the node also applies standstill or moving pipeline as the WISER protocol. However, the pipeline is created between the current node and the node in the next cluster that is geographically closest to the destination.

### 4.3 The WISER/p Protocol: An Ideal Case of Data Delivery

**Single Data Packet.** A node (says $X$) uses the same algorithm as WISER/c to forward data within a cluster. If the next hop does not exist, the node uses a *greedy moving* scheme. In particular, the node moves in the direction towards the next-hop node ($N$) which satisfies two conditions: (i) $N$ must be geographically closest to itself and, (ii) $N$ must be closer to the destination than itself. It stops at the edge of $N$'s communication range. Actually, the node stops after passing the communication border for a short distance ($\delta$). If the next hop does not exist, i.e., no any node satisfies the above two conditions, the destination will be the next-hop node.

However, if there are at least two movements when a packet is sent from a source to a destination, the node uses divide-and-conquer method to achieve minimal movement. For example, if $A$ must move towards $B$ ($|\overrightarrow{P_B} - \overrightarrow{P_A}| - r + \delta$), and then $E$ must move towards $F$ ($|\overrightarrow{P_F} - \overrightarrow{P_E}| - r + \delta$), the node compares a direct distance from $A$ to $F$ ($|\overrightarrow{P_F} - \overrightarrow{P_A}| - r + \delta$) with the two-step movement ($|\overrightarrow{P_B} - \overrightarrow{P_A}| + |\overrightarrow{P_F} - \overrightarrow{P_E}| - 2r + 2\delta$), and chooses the shorter one. As usual, the total delay from a source $S$ to a destination $D$ composes of communication ($(d + s/b)\sum_{i=0}^{k-1} p_i^{-1}$) and moving ($\sum_{\forall j}(l_j - r + \delta)/v$) delays, where ($l_j - r + \delta$) is a distance of the $j^{th}$ movement.

**Multiple Data Packets.** The protocol is based on the single-packet algorithm described above. However, when wireless transmission is not possible, WISER/p agent creates a pipeline between the current node and temporary destination in the next cluster. The current node moves and sends a request message to the node that follows greedy moving scheme, i.e., the node that is closest to itself and, closer to the destination than itself. After sending the request message, it moves back to its original position and starts forwarding data packets.

## 5 Performance Evaluation

We used the *ns*–2 simulation tool to run a number of simulations so as to validate the benefit of intentional mobility. We compared WISER and WISER/c with the ideal solution, WISER/p.

**Table 1.** Packet delivery ratio and delay

| | Packet delivery ratio | | | Delay (seconds) | | | Messages/energy (KB/J) | | |
|---|---|---|---|---|---|---|---|---|---|
| Scenario | WISER | WISER/c | WISER/p | WISER | WISER/c | WISER/p | WISER | WISER/c | WISER/p |
| I | 100% | 100% | 100% | 6.92 | 5.21 | 2.24 | 3.21 | 6.12 | 13.93 |
| II | 100% | 100% | 100% | 4.42 | 3.81 | 1.45 | 4.25 | 4.80 | 13.45 |
| III | 96% | 99.73% | 100% | 4.94 | 1.82 | 1.35 | 7.11 | 15.00 | 25.95 |

### 5.1 Simulation Environment

50 mobile sensor nodes were randomly placed in a 180 m by 180 m square region. Each node had fixed radio coverage of 25 m, the bandwidth of 19.2 kbps, and moved with a speed 10 m/s. 30 random sources sent 36-byte data packet(s) to randomly chosen destination(s). Receive power dissipation (395 mW) was nearly 60% of transmit power dissipation (660 mW) [12]. Another key ingredient of our simulation setup is the cost of mobility. We chose to use a distance proportional cost model, $P_m(d) = kd$, where $d$ is distance and $k$ is 1 J/m [8].

The simulation scenarios varied the number of packets per event (single and multiple) and the number of destinations (single and multiple) as follows.

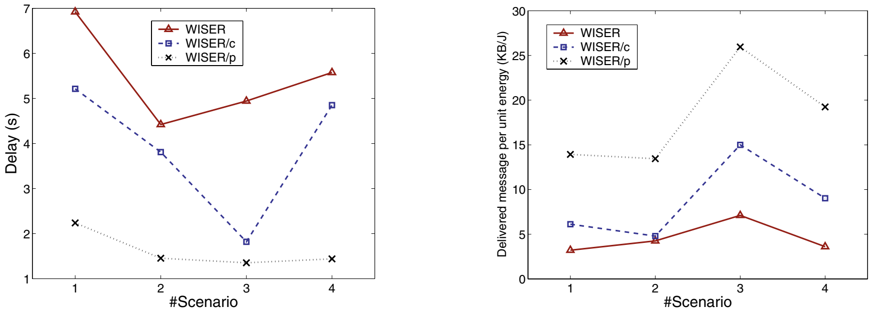
- **Scenario I** – Each source generated single packet per event. Every packet destined to the only destination in the network.
- **Scenario II** – Each source generated single packet per event as the first scenario but the destinations of generated packets were randomly chosen.
- **Scenario III** – 25 sources generated single packet per event, while other five sources generated ten packets per event. All packets destined to the only destination in the network.
- **Scenario IV** – All are the same as Scenario 3, except the pipelining scheme is turned off.

Performance metrics are packet delivery ratio, end-to-end delay, and delivered messages per unit energy.

### 5.2 Simulation Results

Simulation results of Scenario I, II, and III are shown in Table 1. First we consider packet delivery ratio to study the correctness of the protocols. Three WISER protocols deliver 100% in Scenario I and II where single packet is generated per event. As some sources generate multiple packets per event (Scenario III), 4% and 0.3% of the originated packets are dropped by WISER and WISER/c, respectively, while WISER/p can deliver all generated packets. Because helper-nodes in WISER and WISER/c move longer than those of WISER/p, some packets were dropped if queue is full. After the simulations finished, some packets were still on the way to the destination. Those packets are another cause of loss.

Next we consider end-to-end delay used by the protocols. For all of three scenarios, delays of WISER/p are shorter than those of WISER/c, which in turn are



(a) (b)

**Fig. 2.** Impact of pipeline: (a) delay; (b) delivered messages per unit energy

shorter than those of WISER. This is because WISER protocol exploits physical movement of node as a mean of data transfer the most, followed by WISER/c and WISER/p, respectively. The reason behind this fact is that WISER/p has knowledge of the entire network while WISER/c knows only the nodes within the cluster and WISER has only information of neighbors. Hence WISER/p can discover shorter-delay paths compared to the paths decided by WISER/c and WISER. Since mobile nodes in WISER and WISER/c move in a similar way, i.e., they move towards destination for $r$ m, the delays do not differ so much. However, the trend of results in Scenario III is different where the delays of WISER/p are a little bit shorter than those of WISER/c when delivering continuous data packets, while WISER takes more than two times to deliver the packets in the same scenario. In this scenario, WISER/p and WISER/c have more helper-nodes than WISER to construct pipeline, thereby, moving pipeline happens frequently in WISER comparing to WISER/p and WISER/c. Because moving pipeline takes longer time than standstill pipeline to deliver the data, the delays of WISER are much longer than those of WISER/p and WISER/c.

Finally we study energy-efficiency of each protocol by considering how much data can be delivered per unit energy. Because physical movement consumes much more energy than wireless communication, WISER/p which employs less movement than WISER/c and WISER is the best energy-efficient protocol, followed by WISER/c and WISER for all scenarios. The reason can be explained in the same way as the results of delay discussed above.

**Impact of Pipeline.** We study the effectiveness of pipeline by comparing Scenario III and IV. When the pipelining scheme is turned off, all protocols use more time to deliver packets (Fig. 2a) because a node must move multiple round-trips between its current position and destination or next hop. Energy consumption is also higher due to more movements (Fig. 2b). We conclude that the pipelining scheme increases the performance of the protocols in terms of end-to-end delay and energy efficiency because node movement decreases.

## 6 Conclusions

WISER protocols allow network partitioning and utilize controlled mobility of nodes as a means of data transfer in addition to wireless transmission. Mobile sensors in WISER protocols move adaptively according to traffic models and the knowledge of network topology. A node establishes a session for stream data and instructs other mobile robots to construct standstill or moving pipeline as a bridge connecting partitioned networks. We used the *ns*–2 network simulator to study the performance of WISER protocols in many aspects. Simulation results demonstrated the benefit of physical movement, i.e., 100% or nearly 100% of the originated packets were correctly delivered. Perfect data delivery could not be feasible, if prior schemes (e.g., [4]) were applied. Explicit collaboration of robots to deliver stream data also reduces latency and energy dissipation.

## References

1. Vahdat, A., Becker, D.: Epidemic routing for partially-connected ad hoc networks. Technical Report CS-2000-06, Duke University (July 2000)
2. Zhao, W., Ammar, M., Zegura, E.: A message ferrying approach for data delivery in sparse mobile ad hoc networks. In: Proceedings of MobiHoc, pp. 187–198 (May 2004)
3. Thepvilojanapong, N., Tobe, Y., Sezaki, K.: Impact of intentional mobility in sparse sensor networks. In: Proceedings of SenSys, pp. 286–287 (November 2005)
4. Karp, B., Kung, H.T.: GPSR: greedy perimeter stateless routing for wireless networks. In: Proceedings of MobiCom, Boston, MA, USA, pp. 243–254 (August 2000)
5. Beaufour, A., Leopold, M., Bonnet, P.: Smart-tag based data dissemination. In: Proceedings of WSNA, pp. 68–77 (September 2002)
6. Shah, R.C., et al.: Data mules: Modeling a three-tier architecture for sparse sensor networks. In: Proceedings of SNPA, pp. 30–41 (May 2003)
7. Jain, S., et al.: Exploiting mobility for energy efficient data collection in sensor networks. In: Proceedings of WiOpt (March 2004)
8. Goldenberg, D.K., Lin, J., Morse, A.S.: Towards mobility as a network control primitive. In: Proceedings of MobiHoc, pp. 163–174 (May 2004)
9. Small, T., Haas, Z.J.: The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way). In: Proceedings of MobiHoc, pp. 233–244 (June 2003)
10. Fall, K.: A delay-tolerant network architecture for challenged internets. In: Proceedings of SIGCOMM, Karlsruhe, Germany, pp. 27–34 (August 2003)
11. Jain, S., Fall, K., Patra, R.: Routing in a delay tolerant network. In: Proceedings of SIGCOMM, Portland, Oregon, USA, pp. 145–158 (August 2004)
12. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: Proceedings of MobiCom, Boston, Massachusetts, USA, pp. 56–67 (August 2000)