AIAA Infotech@Aerospace 2007 Conference and Exhibit
7 - 10 May 2007, Rohnert Park, California

AIAA 2007-2747

Sensor Data Collection through Unmanned Aircraft Gateways

Andrew Jenkins,* Daniel Henkel,† and Timothy X Brown‡
University of Colorado at Boulder, Boulder, CO, 80309, USA

Current addressing and service discovery schemes in mobile networks are not well-suited to multihop disconnected networks. This paper describes an implementation of a highly mobile ad-hoc network (MANET) that may never experience end-to-end connectivity. Special gateway nodes are described which are responsible for intelligently routing messages to their intended destination(s). These gateway nodes qualify their links and announce their status to the MANET, a simple approach to service discovery that is effective in this implementation. This implementation has been tested in an outdoor environment.

I. Introduction

This paper considers the problem of collecting data from a widespread deployment of sensors via unamanned aircraft (UA). To reduce sensor costs the sensors have simple low-power radios and the role of the UA is to enable communication between the sensors and sensor monitoring stations (SMS). Different SMS may store, process, analyze, or display the sensor data but for our purposes an SMS is a destination for sensor data. It may be in the vicinity of the sensors or it may be remotely located somewhere on the Internet. Furthermore the number and location of SMS may change over the life of the sensors. Though the sensors are simple, they implement ordinary TCP/UDP/IP protocols and the wireless interface may enable networking among the sensor nodes in a local area. The sensor data consists of occasional sensor events initiated by the sensors that need to be replicated and sent to each SMS. An SMS may also wish to send commands or queries to specific sensors. Event and command communication must be reliable; they should be delivered eventually through UA mobility.

The challenges to building this system include:

1. The sensor can not be configured ahead of time with SMS addresses.
2. The SMS and UA may not know the number or exact location of the sensors.
3. SMS, sensors, and UA can dynamically join and leave the network
4. The UA can not guarantee continuous or even simultaneous end-to-end connections between widespread sensors and SMS.
5. The intra-sensor, sensor-to-UA, and UA-to-SMS networks may reside in different conflicting address spaces.

This paper describes our implementation and testing of a system that addresses these challenges and updates and summarizes and earlier description.[1] The system protocols consists of automatic discovery protocols; protocols for reliable forwarding through disconnected networks; multi-stage multicast protocols between sensors and SMS; and network address translation protocols between different networks. The hardware consists of mesh network radios running on low-cost single board computers that are mounted on ground and UA nodes.

*Student, Electrical and Computer Engineering, jenkinaj@colorado.edu
†Student, Interdisciplinary Telecommunications Program, henk@colorado.edu
‡Associate Professor, Electrical and Computer Engineering, timxb@colorado.edu

II. Related Work

Sensor networks have a rich body of research associated with them. However, the backbone data delivery problem is not typically considered; rather, it is often assumed that a stable, well-connected backbone of sufficient bandwidth is locally available as the "data sink."[2–4] This work departs from this assumption: there is no local data sink; and the remote data sinks are numerous. The network must deliver data to all of the remote sinks with efficiency, and particular efficiency requirements are placed on the sensors.

The problem of a remote data sink is most closely related to Delay Tolerant Networks (DTNs).[5–7] This emerging class of networks is devoted to delivering data over occasionally-connected or disconnected networks, in which end-to-end paths may not exist. The issue of routing in delay-tolerant networks has prompted novel solutions.[8,9] Our approach differs in that by enforcing hierarchy through staged delivery, we eliminate the routing burden on the end-points, and intentionally concentrate data at specific aggregators (terminus and gateway nodes).

We have observed TCP's poor performance over low-range wireless local area networks. This performance degrades further when one TCP endpoint is on an airborne UA. Because TCP assumes packet losses are due to congestion, it aggressively constricts the send rate in a wireless environment where packet losses are an inherent property of the channel.[10] This behavior is a conservative approach to the distributed congestion control problem, well-suited for a wired Internet, but less so for wireless ad-hoc networks.

Extensions to improve TCP over wireless have been proposed,[11–13] although none so far has emerged as a clear winner. As well, primary work on these extensions have been designed for infrastructured wireless LANs, rather than multi-hop ad-hoc wireless networks. Even in ad-hoc networks, mobility is often considered to be slow and lower-dimensional (such as a human walking) relative to UA maneuverability. As the UA maneuvers, the antenna pattern changes relative to a communicating ground node causing frequent losses. We implement an alternative to TCP that is less sensitive to losses.

Real-world deployments of sensor networks are emerging. In the ZebraNet,[14,15] the mobility patterns of zebra in a Kenyan research center are delivered over a disconnected sensor network with strict power and size constraints; the designers trade high latency (months) for low power and complexity. Our approach allows the network user to decrease latency and increase throughput through two means: increasing the connectedness of the MANET, or adding gateway UA to relay across any gaps in connectivity. As well, current ZebraNet sensors flood all data to any other sensors they comes in contact with, whereas our collection uses a hierarchy to prevent data flooding while still achieving multicasting.

The CarTel project[16] explores delay-tolerant sensor data collection from users' automobiles. Sensor nodes on cars record data during trips, and opportunistically deliver the data to a portal (analogous to our monitoring station). This occurs during periods of end-to-end connectivity or utilizing appointed one-hop data mules (such as a hand-carried USB memory dongle). An elegant API (called CafNet) is provided to sensor applications. The work focuses on the data filtering problem by implementing an intermittently-connected database query model to filter and prioritize data at the sensors. Multicast (i.e. many portals) is not a feature of the CarTel network.

III. Implementation Architecture

The structure of the network is presented in Fig. 1. Sensor nodes generate event messages that are passed to so-called terminus nodes (T). A terminus bridges between the sensor and a mobile ad hoc network (MANET) consisting of mesh network radio nodes (MNR) and gateway nodes (GW). The GW bridges between the MANET and external network connected to SMS. An SMS can in turn send a command message back through this network to sensors.

Gateway nodes forward event messages to as many SMS as possible (event messages are broadcast), and forward commands to the specific sensor to which they are addressed (command messages are unicast). The gateways may be highly mobile nodes, typically onboard unmanned aircraft or ground vehicles. They communicate with SMS through links external to the MANET that support IP. Ethernet, 802.11 wireless LAN, CDMA, and satellite connections have been tested within the architecture. Nodes that participate in the MANET do so through 802.11. Each gateway may have connectivity to any number of SMS, or none at all. As well, each SMS may be served by multiple gateways. This many-to-many association presents considerable challenges in promoting reliability.

In our network, *terminus* nodes are attached to sensors. Terminus participate in the MANET, and

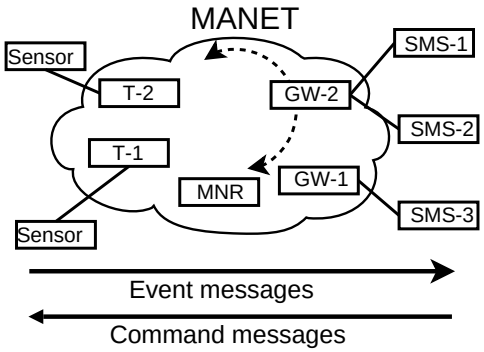

Figure 1. Sensor data collection over a MANET, with gateways to other networks

also present a second interface to sensors. Terminus nodes are intermediary between the MANET and the sensors, because it is assumed that sensors may be very simple devices incapable of implementing MANET routing protocols. Sensors can communicate to terminus over links as simple as serial line IP (SLIP). Wired ethernet is used in the implementation under consideration. More capable sensors can implement the terminus function directly.

As well, nodes called MNRs (mesh network radios) may be attached to the network. These participate in some MANET protocols, but do not share a network with sensors (as terminus do), or SMS (as gateways do). These nodes can be added to extended coverage of the MANET.

The network consists of different classes of nodes. In our implementation, these nodes are embedded single-board computers using CompactFlash for permanent storage, and roughly equivalent to 486 personal computers in terms of processing power. They are battery powered, and have two 802.3 wired ethernet interfaces as well as a 1 Watt 802.11a,b,g interfaces that operate in ad-hoc mode. Linux is used as the embedded operating system, and the *Click Modular Router*[17] (referred to as Click) forms the backbone of our custom networking software.

The MANET uses an implementation of the Dynamic Source Routing (DSR) protocol, also written for Click.[18] DSR permits MANET nodes to use the mesh network to reach destinations through hops that may not support our delay tolerant networking protocols, but do participate in DSR.

IV. Protocol Architecture

This network uses a staged delivery approach to relay data (called events) from sensors to a multicast group of sensor monitoring stations. As well, sensor monitoring stations can direct data (called commands) to a specific sensor. This section outlines the features and design of our architecture. These cover reliable end-to-end delivery over unreliable links, discovery of the next stage, and addressing schemes.

IV.A. Disconnected Networks

Ordinary TCP/IP requires a simultaneous end-to-end connection throughout a data exchange. Such end-to-end connectivity may never be available in the network under consideration. To solve this, events and commands are forwarded via a reliable packet forwarding protocol (RPF). The path between sensor and SMS is divided into three stages; sensor to terminus, terminus to gateway, and gateway to SMS. In each stage packets are persistently sent until the packet is acknowledged by the transfer point (terminus or gateway) which takes custody of the packet and ensures it is sent across the next stage.

The staged-delivery approach utilizes hop-by-hop reliability to ensure that data is not lost, while permitting efficient communication over disconnected networks. This procedure supports a variety of UA-based gateway sensor data collection strategies. In particular, it supports a data ferry strategy where UA gateways physically carry data between sensors and distant SMS (Section V).

This delivery from one stage to the next is on top of underlying, unreliable networks using an automatic repeat request (ARQ) policy. For instance, when the terminus attempts to deliver to a gateway (transporting data from the terminus stage to the gateway stage), the data is routed through a multi-hop MANET. The MANET is unreliable, so intermediate hops in the MANET may drop packets, or transmissions between

hops may be lost. If the terminus receives an acknowledgement from the gateway, then the terminus is no longer the custodian of the data; if an acknowledgement is not received, then the terminus will retry delivery to a gateway. Thus, the hop-by-hop reliability is provided between the terminus and the gateway (as stages in the staged delivery), rather than between each hop of the MANET (which is an underlying unreliable network between stages).

IV.B. Gateway and SMS discovery

Each stage discovers the next stage in the relay process through a discovery mechanism. The simplest discovery mechanism is *static*: the stage knows the next stage through some out-of-band means, such as pre-configuration. Two other discovery mechanisms are *probing* (the stage actively searches for the next stage), and *advertisement* (the stage waits to receive an announcement from the next stage). This section oulines the discovery mechanisms employed in our network.

IV.B.1. Probing

Gateways use probing to discover SMS. A gateway has a preconfigured list of network addresses (on the internetworks between a gateway and SMS) of the SMS that may be present. The gateway periodically attempts to contact the SMS in this list through what we call a heartbeat protocol. An unreliable UDP/IP datagram is periodically sent to each SMS, containing a *valid time*. If the SMS receives this datagram and is ready to receive sensor data, it echoes the original valid time back to the gateway. If it forsees terminating before the valid time has expired (perhaps a shutdown is planned within the next few minutes), it may reduce the valid time in the reply accordingly.

When the gateway receives this heartbeat response, it considers the link to this SMS to be valid for the valid time. Validity does not indicate that the link will be error-free, or is guaranteed to be connected. Rather, it indicates that it is probable that this gateway will be able to deliver data to this SMS for the duration of valid time. As explained in further detail in Section IV.C, this indication helps terminus decide which gateways are good choices for delivering data.

If a gateway has SMS that it considers valid according to this heartbeat protocol, it announces its presence periodically to the MANET (and thus all reachable terminus and other gateways) through the advertisement protocol, discussed next.

IV.B.2. Advertisement

The terminus discover nearby gateways through an advertisement protocol. In this protocol, the terminus passively listen for advertisements from gateways. When gateways have valid links to SMS, as determined by the probing protocol (Section IV.B.1), they emit periodic advertisements to the MANET. These are addressed to the broadcast address of the MANET. These advertisements reach nodes currently within the connected MANET.

If the MANET supports an efficient multi-hop broadcast mechanism (such as a spanning tree), this mechanism can be utilized. If no such mechanism exists, a simple flooding approach is utilized for advertisements. In this scheme, every MANET node that hears an advertisement repeats it exactly once (each advertisement has a unique identifying sequence number). Because the period of the advertisements is typically large (on the order of minutes), the inefficiency of the flooding approach has little impact in our testbed.

IV.C. Equivalence Classes

Gateways have equivalence classes assigned to them along with their initial SMS address list. These numbers uniquely identify the SMS address list, so if gateways have identical address lists, they have the same equivalence class. If a terminus has advertisements from multiple gateways of the same equivalence class, they can reduce data replication by only delivering to one of these gateways.

This feature also enables gateways to forward undeliverable messages to other equivalent gateways. Gateways listen to advertisements that they hear from other gateways just as terminus do. If a gateway is completely unable to deliver messages to the SMS (its valid time expires), then it will check to see if another gateway is in its equivalence class and has recently announced itself as valid. If so, the gateway forwards events to this new gateway to permit delivery. In scenarios where gateway UA leverage their altitude to deliver data over line-of-sight links, this would permit a gateway that has landed to forward messages to a

gateway still in the air. Thus, UA can be rotated from landed to in-the-air and will automatically deliver events to SMS with high reliability.

IV.D. Addressing

To solve the problem of addressing when multiple possibly conflicting address spaces are used, we develop a network address and port translation (NAPT) scheme. Addresses in the sensor network (from sensors to terminus) are not revealed to the MANET (from terminus to gateways) or to the internetwork from gateways to SMS. This means that a sensor network can be deployed without regard to what other sensor networks will also utilize our sensor data collection architecture, and addresses can be reused across different sensor networks without conflict.

The addressing scheme utilizes network address and port translation to operate without a global addressing scheme. Since it is assumed that sensors and sensor monitoring stations may be simple, the NAPT scheme also shields these nodes from knowing any addressing details of the MANET. The scheme is detailed in Figure 2.

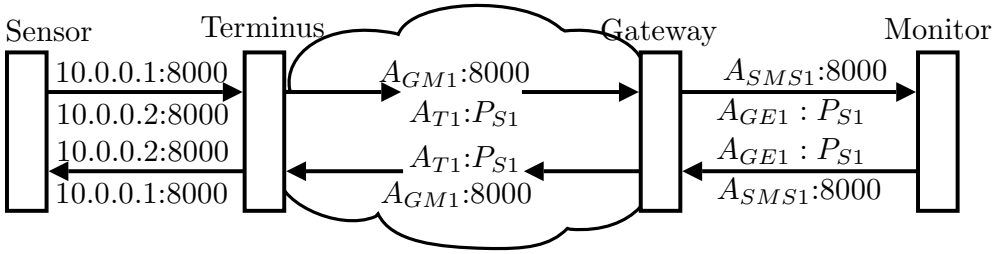

Figure 2. Network Address and Port Translation (NAPT) scheme provides conflict-free internetwork addressing and reduces addressing burden on sensors and sensor monitoring stations.

Each sensor sends all events (sensor data) to a single terminus address and port. This address and port pair may be common among all sensor networks. Thus, to multicast events to all connected SMS over a disconnected, multi-hop delay-tolerant network, the sensor must simply send data to one address and port.

Once it has arrived at the terminus, the terminus selects gateways to forward the data to according to the advertisement mechanism (Section IV.B.2). These data packets have the terminus MANET address as the source address, and a special port, called the *key port*, as the source port. This port is derived from the sensor network address of the sensor, and uniquely identifies that sensor within the sensor network attached to the terminus. This is the first stage of forward NAPT: the sensor address and port are translated to a terminus address and key port. The terminus obtains the mapping from sensor address to key port from a NAPT table, so that when commands arrive intended for a sensor, it performs the inverse mapping to reach the correct sensor. Once the data arrives at the gateway, it replaces the source address with its own, but retains the sensor key port. This mapping exists in the gateway's NAPT table, so that commands arriving at this gateway can have the inverse mapping applied to reach the correct terminus, with the correct key port.

When sensor monitoring stations receive sensor data events, they can respond with a command by simply sending a command packet to the source address and port of the event. This will be the gateway which delivered the event, so it can perform the inverse NAPT to reach the terminus. The terminus performs an inverse NAPT again, and the command reaches the intended sensor. Using this scheme, neither the sensor nor the sensor monitoring station need to maintain any NAPT or addressing tables to reliably send and receive events and commands. If the sensor monitoring station wishes to identify sensors (such as by GPS coordinates), the data should contain this information as an application-layer option.

This approach resolves conflicting network addressing. For instance, if Sensor A is in Sensor Network 1 (with Terminus 1), and Sensor B is in Sensor Network 2 (with Terminus 2), it is of no consequence if Sensor A and Sensor B have identical network addresses. When their data arrives at their corresponding terminus, it is assigned a key port and the terminus address. This combination of key port and terminus address is what affects the routing of command responses (rather than the sensor addresses themselves). Upon receipt at the SMS, the sensor data from Sensor A will have a different source port than that of Sensor B, so the SMS can distinguish the two.

This architecture permits very simple sensor network addressing schemes, which reduces the assignment

burden: Sensors can be added and removed from the network at will, without requiring any centralized addressing coordination; the only requirement is that the sensor address is unique within that sensor network. As previously discussed, SMS can come online or offline throughout the network duration; when present, they will receive sensor data, and after alerting gateways that they are offline (or not responding to many heartbeat probes), they will leave the multicast group and the network will not expend energy trying to route sensor data to them.

IV.E. Reliable Packet Forwarding

An alternative transport mechanism to TCP was developed. Named Reliable Packet Forwarding (RPF), this transport exists inside the UDP header format and provides performance that is less variable over time than TCP in the wireless, high frame-error-rate environment. Similar to TCP and other ARQ schemes, RPF uses a sliding window with cumulative acknowledgments. Selective acknowledgements are also included; this keeps the window large in the face of channel losses.

An additive increase/additive decrease algorithm manages the window size. The standard TCP additive increase and multiplicative decrease is a conservative approach to reducing losses in intermediate hops due to congestion, but this overcompensates for losses due to channel conditions.

Experiments with emulated links of high delay (up to 2 seconds) and high loss (33% frame error rate) showed that our simple approach still permits data delivery in scenarios for which standard TCP does not. For these tests of extreme links, it was necessary to tune the additive increase and decrease factors to promote more aggressive channel utilization. This adversely affects the behavior of the protocol with respect to congestion and a hybrid approach (be conservative if congestion is detected, but less so if the channel is inherently bad) may be warranted.

V. Gateway Relaying

A primary goal of this network is to support the use of a UA as a data relay. In this case, the UA physically relays data from a set of simple, low-cost and low-power sensors to sensor monitoring stations over a geographical distance.

In one case, the sensors are partitioned into individual sensor networks. Each sensor network has a terminus, which functions as the data sink for that sensor network. When the UA gateway is nearby, the terminus forwards accumulated data to it. The UA can visit several terminus (and thus collect data from all these sensor networks) before relaying to SMS. Each terminus could be a waypoint on an orbit of the UA. Since the terminus communicates with the gateway on behalf of the sensors, the sensors can use low-range and low-power technology (thus lowering cost), while only the terminus needs a more powerful radio to reach the UA gateway.

In another case, the sensors each implement the terminus functionality directly (i.e. sensor and terminus are one physical entity). This increases the complexity and cost of this sensor node, but eliminates the need for seperate terminus for each sensor network.

The gateway UA can use one or more different wireless data communication technologies to reach SMS. We have tested our approach using 802.11 wireless, data over CDMA, and an Iridium satellite connection to the public Internet.

This gateway-as-relay approach means that each sensor network does not need collocated high-cost, high-power backbone links to deliver data over large distances. The gateway UA can instead function as the backbone; either providing a single backbone link to many sensor networks or physically relaying the data.

VI. Testbed

To test our design, we have implemented the protocols described in Section IV as modules in the Click Modular Router, written in C++. Click, with our modules, is loaded on the CompactFlash cards of our Soekris single-board computers. These nodes are configured with one of the roles outlined in Figure 1.

In one version of our testbed, nodes are connected with wired 10Mbps ethernet. This provides channel performance approximate to wireless 802.11, but enables us to easily control the connectivity of nodes and eliminate the variability of the wireless channel as a factor in initial testing. We can enforce relaying by activating links from the gateway node to a terminus, and then to an SMS network, and back to terminus

again. Intermittent links are emulated via simple plugging and unplugging of cables. We use an application we developed, called the "DTNClock" to visualize performance in this scenario (Figure 3).

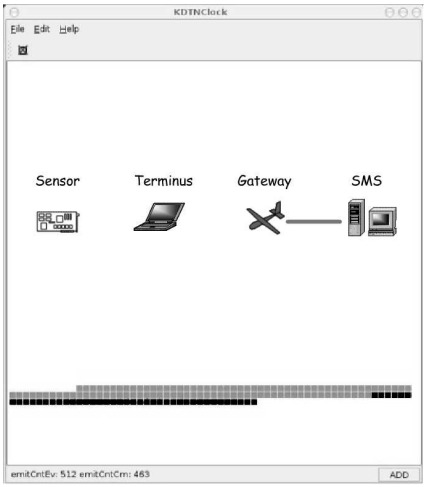

Figure 3. The DTNClock graphically depicts the delivery of sensor data to monitoring stations. When the sensor generates data (once per second in this example), a light grey square is painted. When this data arrives at the SMS and its receipt is acknowledged back to the sensor, that square is painted solid black. The icons above the square are added to indicate the state of the links.

We have also deployed our nodes to field scenarios. In this case, the battery-powered nodes use 802.11b and are tripod-mounted. A gateway is attached to a car rooftop, and a sensor network is seperated from a monitoring station (no direct connectivity between them is ensured). The gateway is driven back and forth, and we observe the reliable relaying of sensor data over this divide.

VII. Conclusion

Emerging technology in both UA and wireless data networks can facilitate new approaches to sensor networks. Our approach to collecting data from remote sensor networks and delivering it to sensor monitoring stations leverages ad-hoc networks and UA. This reduces the the cost and complexity of sensors. Sensors and sensor monitoring stations can join and leave the network over its lifetime, and our protocols handle these network dynamics without requiring intervention.

We leverage UA mobility to provide a gateway service to sensor networks. Our approach utilizes UA mobility to ensure that eventually sensor data is delivered to monitoring stations.

Our work reports on the design, implementation, and some testing of this architecture. It enables mobile gateways to perform sensor data collection and deliver this data to remote sensor monitoring stations. This is achieved by a new set of protocols that provide the staged relay structure, automatically discover the next stage (through probing or advertisement discovery mechanisms), and then use a new and simple ARQ scheme to reliably transfer data. By moving aspects of multicasting to more powerful aggregator stages (terminus and gateways), we reduce the burden on the sensors and monitoring stations.

Because gateways on UA may experience changing link conditions and may have to remove themselves from network participation due to environmental conditions or operational constraints, gateway-to-gateway forwarding is supported. Our approach has been applied to a diversity of links in outdoor and indoor testbeds.

Acknowledgments

This work was supported by NSF award CNS 0428887, AFOSR grant no. FA9550-06-1-0205, and a contract from L3-ComCept Corporation.

References

[1]Jenkins, A., Henkel, D., and Brown, T. X., "Sensor Data Collection Through Gateways in a Highly Mobile Mesh Network," *IEEE Wireless Communications and Networking Conference (WCNC)*, March 2007.

[2]Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E., "A survey on Sensor Networks," *IEEE Communications Magazine*, August 2002, pp. 102–114.

[3]Akan, O. B. and Akyildiz, I. F., "Event-to-sink reliable transport in wireless sensor networks," *IEEE/ACM Trans. Netw.*, Vol. 13, No. 5, 2005, pp. 1003–1016.

[4]Schurgers, C., Tsiatsis, V., Ganeriwal, S., and Srivastava, M., "Topology management for sensor networks: exploiting latency and density," *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, ACM Press, New York, NY, USA, 2002, pp. 135–145.

[5]Fall, K., "A delay-tolerant network architecture for challenged internets," *SIGCOMM '01*, 2003, pp. 27–34.

[6]Cerf, V. G., Burleigh, S. C., Durst, R. C., Fall, K., Hooke, A. J., Scott, K. L., Torgerson, L., and Weiss, H. S., "Delay-Tolerant Network Architecture," March 2006.

[7]Warthman, F., "Delay-tolerant networks (DTNs): A tutorial," DTN research group tutorial, URL: http://www.dtnrg.org/docs/ tutorials/warthman-1.1.pdf, 2003.

[8]Burgess, J., Gallagher, B., Jensen, D., and Levine, B., "MaxProp: Routing for vehicle-based delay-tolerant networks," *IEEE INFOCOM*, March 2006.

[9]Jain, S., Fall, K., and Patra, R., "Routing in a delay tolerant network," *SIGCOMM Comput. Commun. Rev.*, Vol. 34, No. 4, 2004, pp. 145–158.

[10]Xylomenos, G., Polyzos, G. C., Mahonen, P., and Saaranen, M., "TCP performance issues over wireless links," *IEEE Communications Magazine*, Vol. 39, April 2001, pp. 52–58.

[11]García, M., Choque, J., Sánchez, L., and Muñoz, L., "An experimental study of Snoop TCP performance over the IEEE 802.11b WLAN," *5th Int'l. Symp. Wireless Personal Multimedia Commun.*, Vol. 3, October 2002, pp. 1068–72.

[12]Ratnam, K. and Matta, I., "WTCP: An Efficient Mechanism for Improving TCP Performance over Wireless Links," *ISCC*, Vol. 00, 1998, pp. 74.

[13]ElRakabawy, S. M., Klemm, A., and Lindemann, C., "Gateway adaptive pacing for TCP across multihop wireless networks and the Internet," *Proceedings of the 9th ACM international symposium on modeling analysis and simulation of wireless and mobile systems*, 2006, pp. 173–182.

[14]Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L. S., and Rubenstein, D., "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet," *ACM SIGPLAN Not.*, Vol. 37, No. 10, 2002, pp. 96–107.

[15]Martonosi, M., "Embedded systems in the wild: ZebraNet software, hardware, and deployment experiences," *ACM SIGPLAN Not.*, Vol. 41, No. 7, 2006, pp. 1–1.

[16]Hull, B., Bychkovsky, V., Zhang, Y., Chen, K., Goraczko, M., Miu, A., Shih, E., Balakrishnan, H., and Madden, S., "CarTel: a distributed mobile sensor computing system," *SenSys '06: Proceedings of the Fourth International Conference on Embedded Networked Sensor Systems*, October 2006, pp. 125–138.

[17]Kohler, E., Morris, R., Chen, B., Jannotti, J., and Kaashoek, M. F., "The Click modular router," *ACM Transactions on Computer Systems*, Vol. 18, No. 3, August 2000, pp. 263–297.

[18]Doshi, S., Bhandare, S., and Brown, T. X., "An on-demand minimum energy routing protocol for a wireless ad-hoc network," *Mobile Computing and Communications Review*, Vol. 6, 2002, pp. 50–66.