Energy-Aware On-Demand Scatternet Formation and Routing for Bluetooth-Based Wireless Sensor Networks

Xin Zhang and George F. Riley, Georgia Institute of Technology

ABSTRACT

Bluetooth is a promising short-range wireless communication technology with the characteristics of interference resilience and power efficiency, both desirable for wireless sensor networks. The new Intel Mote sensor devices have Bluetooth technology incorporated as the standard wireless communications interface. When using Bluetooth in applications where multihop routing is required, groups of Bluetooth piconets combine together to form a scatternet. However, most of the existing scatternet formation protocols are designed to facilitate communications between any two pairs of devices, regardless of the actual traffic demand pattern. For wireless sensor network applications with low-duty-cycle traffic patterns, an on-demand scatternet formation protocol can achieve significant power saving by avoiding unnecessary network connectivity. To that end, we introduce an on-demand scatternet and route formation protocol designed specifically for Bluetooth-based wireless sensor networks. Our protocol builds a scatternet on demand, and is able to cope with multiple sources initiating traffic simultaneously. In addition, our energy-aware forwarding nodes selection scheme is based on local information only, and results in more uniform network resource utilization and improved network lifetime. Simulation results show that our protocol can provide scatternet formation with reasonable delay and good load balance, which results in prolonged network lifetime for Bluetooth-based wireless sensor networks.

INTRODUCTION

Advances in micro-sensing technology as well as numerous novel applications have led to a substantial volume of research on wireless sensor networks (WSNs) [1]. Bluetooth [2] is a short-range wireless technology based on time-division multiple access (TDMA) and frequency hopping spread spectrum (FHSS). The use of the spread spectrum technique leads to substantial interference resilience, which makes Bluetooth a good choice for wireless communications in WSNs. Other protocols, such as IEEE 802.11, use idle listening and collision avoidance actions, making such protocols less appropriate for WSNs. Newer protocols, such as IEEE 802.15.4, may be appropriate for WSN applications, but as yet devices with such interfaces are not widely available. Interconnecting these sensor devices with external networks is difficult.

In [3] various advantages and limitations are discussed concerning Bluetooth-based WSNs. The main advantage of employing Bluetooth is that all sensor nodes within radio range of each other can use separate channels to avoid interference, instead of competing for a shared channel and reacting to interference. In addition, Bluetooth's low-power modes allow the radio to enter power saving states when there is no active communication. Given these desirable properties for WSN applications, several sensor devices using Bluetooth as the wireless interface have been introduced, including BT nodes [4] from ETH Zurich and the Intel Mote [5].

The applications for WSNs are widespread and with diverse requirements. The protocols for WSNs are more application-driven than universal. An important class of WSN applications have a multilevel network architecture with a large number of sensor nodes dispersed within a geographic area, and often communicating to the external network through a sink node. Since the transmission range of Bluetooth devices is only tens of meters, the communication between the sensor nodes and the sink node will often require multihop routing. WSN applications with this structure include habitat monitoring [6], civil infrastructure health monitoring [7], weather monitoring and reporting, and others. In these applications, data transfers occur infrequently, with a few unpredictable bursts. This type of traffic pattern is ideal for Bluetooth-based WSNs [3]. Bluetooth connections for such applications are established as needed, depending on the traffic requirements, and torn down when traffic ceases in order to save power.

 0163-6804/05/$20.00

One of the primary requirements for Bluetooth-based WSNs to work effectively is an efficient scatternet formation and routing protocol for multihop networks. Such protocols become even more complicated when multiple sources start route discovery concurrently. In addition, in WSNs where sensor motes are powered by batteries, uniform battery drain on all nodes is desirable and will lead to longer network lifetimes.

We introduce an on-demand scatternet formation and routing protocol designed specifically to address these requirements. A modified *Inquiry* method with extended ID (EID) packets is used for route discovery. During scatternet formation and route reply phase, modified *POLL* packets are used in *Page* mode. Furthermore, the resulting scatternet maintains cross routes for multiple sources initiating traffic at the same time. The cross route formation for sources with concurrent traffic is essential for densely deployed sensor networks. The data aggregation within the network also benefits from the concurrent process when data from multiple sources are correlated. In order to balance network load, a local decision of whether to forward a received route request is made by each node based on its own residual energy. The energy-aware intermediate nodes selection prolongs the network lifetime without the overhead of neighbor information exchange.

The remainder of this article is organized as follows. We first give the related work of Bluetooth scatternet formation schemes. We then describe our detailed energy-aware on-demand scatternet and route formation protocol. We present simulation results showing the efficiency of our protocol afterward. Finally, our conclusions are discussed.

Related Work

In the literature on Bluetooth scatternet formation protocols, the major solutions can be categorized as proactive and on-demand mechanisms.

Bluetooth was initially designed as a cable interconnect replacement technology. Thus, continuous connectivity is the primary concern for most of the existing work on multihop construction (scatternet formation) schemes. A good overview of the major proactive scatternet formation protocols and performance comparisons are presented in [8].

For WSNs with low-duty-cycle traffic, maintaining connection of the entire network is a significant power drain. Hence, the on-demand scatternet formation approach is more feasible. To the best of our knowledge, the only existing work addressing on-demand scatternet formation is presented in [9–11].

In [9], an EID packet broadcast mechanism is introduced to reduce the route discovery delay. However, the ID packet in Bluetooth is designed to be small in order to save power, since the number of ID packets transmitted in the Bluetooth *Inquiry* phase can be very large. Substituting all ID packets with the much longer EID packets used to transfer source

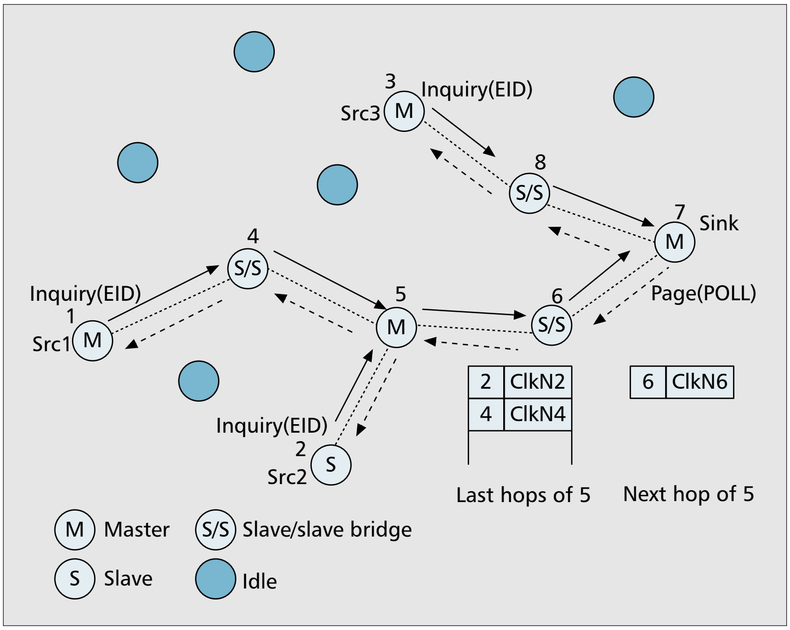

Figure 1. *Scatternet and route formation.*

information in scatternet formation is unnecessary and uses additional power. This is because most of the ID packets are simply for neighbor probing and synchronization rather than route discovery. In addition, the simulations of route discovery in [9] only consider the single-source scenario. When multiple sources in a network initiate a scatternet formation simultaneously and involve common intermediate nodes, they will interfere with each other and degrade the performance significantly.

Another on-demand Bluetooth scatternet formation algorithm, called on-demand Bluetooth (ODBT), is presented in [10]. ODBT constructs a scatternet with a tree topology. It includes the ability to cope with Bluetooth devices dynamically joining and leaving the scatternet. However, it still tries to connect all the nodes within the network and cannot operate in the presence of multiple sources simultaneously starting the formation of a scatternet involving common intermediate devices.

In [11] a two-phase scatternet formation (TPSF) protocol is introduced that supports dynamic topology changes. A control scatternet is constructed in the first phase to support topology changes and route determination, while an on- demand scatternet is created in the second phase whenever data communication is needed. However, maintaining the control scatternet constructed in the first phase can be power consuming and makes TPSF similar to proactive approaches.

The on-demand scatternet formation and routing protocol we introduce here forms traffic-dependent scatternets and routes. It allows multiple sources to initiate the formation simultaneously. Furthermore, our approach increases network lifetime by avoiding energy depletion at critical nodes in the routing path. A detailed description of the protocol is presented next.

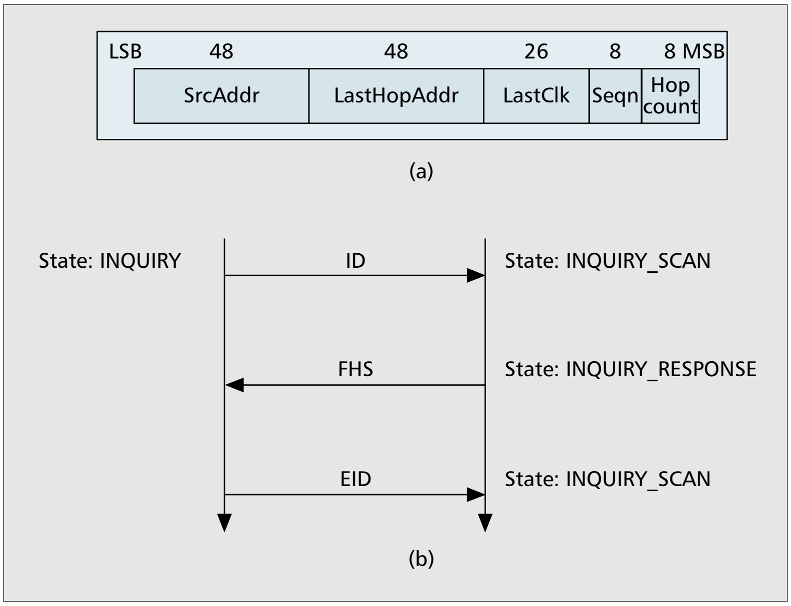

Figure 2. *Modified* Inquiry *with EID packet: a) EID packet format; b) modified* Inquiry *process.*

ENERGY AWARE ON-DEMAND SCATTERNET AND ROUTE FORMATION PROTOCOL

As mentioned earlier, no protocol for WSNs is universal, but rather the choice of an appropriate protocol is application-dependent. We consider a typical WSN architecture as shown in Fig. 1 of [6], which is popular in habitat and environment monitoring, data collection. Two classes of Bluetooth nodes exist in the network: a high-power sink node and low-power nodes such as Intel motes [5]. Sensor motes communicate with the sink in order to send data to an external network such as the Internet. Since the sink may not be in the radio range of all the motes, a Bluetooth scatternet must be formed. The scatternet formation protocol used in the Intel motes is a proactive approach and maintains complete connectivity between all devices. However, the traffic demand from sensors is typically not continuous. Therefore, an on-demand formation protocol would be more energy-efficient and result in improved network lifetime.

OVERVIEW

The Bluetooth specification [2] defines a network with a MASTER/SLAVE structure. One MASTER and up to seven active SLAVEs form a piconet. The MASTER is in charge of the connection and communication of the piconet, whereas SLAVEs are required to synchronize to the MASTER. Multiple piconets can be linked together to form a scatternet. To interconnect Bluetooth devices into a scatternet, some devices act as bridges between adjacent piconets. In our scatternet formation protocol, SLAVE/SLAVE bridges[1] are chosen to reduce the number of piconets within a scatternet. Hence, a structure of strict alternating of MASTER/SLAVE roles is maintained along a route from a source to the sink.

Instead of running a routing protocol after a scatternet is formed, our approach combines these two processes. An overview of our protocol is shown in Fig. 1. The route requests propagate through an *Inquiry* broadcast and are relayed from sources to the sink, while the scatternet formation and route replies are delivered in the opposite direction by *Page* messages. Since the common destination is the sink node, all route requests arriving at the same intermediate node (e.g., route requests from Src1 and Src2 arrive at node 5 in Fig. 1) are merged, which avoids redundant request transmissions. It also makes nodes on cross routes join the same piconets as often as possible (e.g., nodes 2 and 4 join the same piconet). In addition, support for concurrent cross routes formation makes data aggregation easier, which is important for WSNs. All intermediate nodes buffer their last hop nodes' device addresses and clock values in order to page them when route replies come back. After a Bluetooth device discovers a route to the sink, the next hop information is cached for a period of time. The timeout value for the cached route should be a function of network mobility. If new route requests arrive before the cached route to the sink expires, the next hop node will be paged first. The neighbor information cached at node 5 is also shown in Fig. 1. The selection of forwarding nodes for route requests is based on the residual energy of each node receiving the requests. The details of scatternet formation and route discovery are discussed in the following sections.

ROUTE REQUEST

Extended ID Packet and Modified Inquiry — In the Bluetooth *Inquiry* phase, a MASTER is able to get the device addresses and clock values from the SLAVEs, whereas SLAVEs have no information about the MASTER. In order to propagate source information in the downstream direction from the source to the sink during route discovery, we use an EID packet, which includes the *Inquiry* MASTER's address and clock as well as other route request information. EID packets in our protocol are used in the modified *Inquiry* rather than replacing the original ID packets as in [9, 11]. In the modified *Inquiry*, after the MASTER discovers new devices, it sends an EID packet to transfer route requests to the newly discovered devices.

Each field and its corresponding length in an EID packet is shown in Fig. 2a. The *Inquiry* process in our protocol is modified to accommodate the new EID packet. The modified *Inquiry* process is illustrated in Fig. 2b. An EID packet is sent by the upstream node after a new device discovery. In this case, the downstream node can get the source and last hop information in an *Inquiry* process, while normal small ID packets are still used for the neighbor and synchronization probes. Upon receiving the EID packet, intermediate nodes function as relays.

Route Request Forwarding — When a source has traffic to send to the sink, it starts a scatternet and route formation using *Inquiry* to

[1] *The node acts as a SLAVE in both piconets in which it participates. For example, node 4 in Fig. 1 acts as a SLAVE in piconets of nodes 1 and 5.*

search bridge nodes in order to reach the sink. The modified *Inquiry* process shown in Fig. 2b is in effect. A scatternet formation timer *ScatFormTO* is started. This timer is stopped as soon as the scatternet formation and route reply arrives at the source. The intermediate nodes receiving route requests save the information about the source and last hop in a structure, *PrecursorList*, which is used to relay route reply to the upstream nodes as well as avoid flooding loops.

The intermediate nodes receiving route requests initiate their own *Inquiry* process to probe the next hop nodes toward the sink. At *Inquiry* timeout, the nodes that have detected their next hops switch between *Page_Scan* and *Inquiry_Scan* states. The state switch enables the nodes to wait for the scatternet and route formation replies in *Page* messages from the downstream nodes. At the same time, the nodes in route discovery can still accept new route requests from other sources. This enables multiple sources to start route requests simultaneously. The period of the state switch affects the performance in terms of scatternet formation delay, discussed later.

When route requests arrive at a node that is already in route discovery, the information from the new source requests will be saved in the *PrecursorList*. However, no *Inquiry* messages are generated. If a route request arrives at a node from a source already in the node's *PrecursorList* but traverse through a different route, the entry in the *PrecursorList* is updated to the route with the shorter path. For a route request with the same source address but larger *Seqn*, a new *Inquiry* message is generated since the previous request will get no response due to packet loss or *Inquiry*/*Page* failure.

After the scatternet and route formation requests arrive at a sink, the route reply will propagate in the reverse direction of the route request and the scatternet will be formed hop by hop. There is a delay between the first route request arrival at the sink and the initiation of route reply and scatternet formation. This short delay enables multiple requests to arrive at the sink and share the same scatternet formation using the sink's immediate hop. In addition, during this delay period, requests from the same source but via different paths can arrive at the sink. In this case, the route with smaller *HopCount* will be chosen, which decreases the number of piconets on a route.

SCATTERNET FORMATION AND ROUTE REPLY

Upon receiving the route request and after a short delay, the sink responds with a route reply using a *Page* message and initiates the scatternet formation process.

Modified Page — In our on-demand scatternet formation approach, a strict MASTER/SLAVE role alternation is maintained along any route from a source to a sink. To transfer the scatternet role assignment information along a route without extra overhead, we use a *Page* procedure with modified POLL packets.

In the Bluetooth Page process, a MASTER assigns a non-zero active member address (*AMAddr*) to each SLAVE within its piconet through the POLL packets. In a scatternet with alternating MASTER/SLAVE roles (*ScatRole*), the *AMAddr* assigned by the nodes with *ScatRole* set to SLAVE has no meaning since the *Page* from these nodes is only for scatternet formation and route reply information propagation. In addition, broadcast packets with *AMAddr* of zero are not used in scatternet connection establishment. When a node with a *ScatRole* of SLAVE tries to page its last hop nodes to relay the scatternet formation information, the *AMAddr* in the *POLL* packet is set to zero, while the *AMAddr* assigned by nodes with *ScatRole* of MASTER is between 1 and 7. In this case, the upstream nodes in the scatternet route can decide their *ScatRole* based on the *AMAddr*. With the modified *Page* scheme, *ScatRole* information is transferred without overhead and there is no effect on the proper operation of scatternet formation.

Route Reply Propagation and Scatternet Formation — Scatternet formation is initiated by the sink. The sink sets its *ScatRole* to MASTER and forms its piconet by paging all the last hop nodes in its *PrecursorList* to establish connection channels. The intermediate nodes being paged get their next hop's address and clock value through the *Page* process, and then page their own upstream nodes. For nodes with *ScatRole* of SLAVE, they only page their upstream nodes to transfer scatternet formation and route reply information, and then switch to *Page_Scan* state to wait for participating the piconets of their upstream nodes. On the other hand, the nodes with *ScatRole* of MASTER page both their last hop nodes and next hop node to form their own piconets.

To avoid multiple nodes with *ScatRole* of MASTER paging the same next hop node simultaneously, a random backoff is used. The scatternet formation for these MASTERs starts when the backoff timer expires.

ENERGY-AWARE SCATTERNET FORMATION

In wireless sensor networks, the energy consumption of each node is important due to the limited power provided by batteries. A large number of sensor nodes deployed in neighboring terrains form a dense multihop ad hoc network, in which every node can work as a router to forward data to the sink. Therefore, how to balance the traffic load over the entire network and prolong the lifetime through efficient scatternet formation and routing is one of the primary concerns for WSNs. The network lifetime is defined as the time at which the first node in the network is depleted of its energy.

Our new protocol for scatternet formation is energy-aware in that the neighboring nodes with more power are selected preferentially over neighbors with less power. It has been observed that the sensor mote's battery life linearly declines with current consumption [12]. Hence, in order to extend the network lifetime, the selection of relaying nodes from the sensor

How to balance the traffic load over the entire network and prolong the lifetime through efficient scatternet formation and routing is one of the primary concerns for WSNs. The network lifetime is defined as the time at which the first node in the network is depleted of its energy.

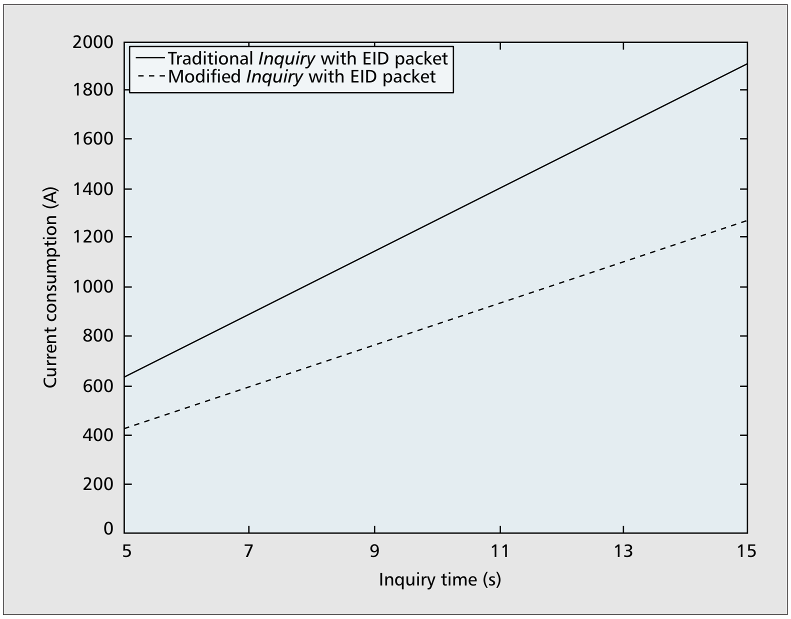

Figure 3. *Current consumption of* Inquiry.

sources to the sink should make use of the information of residual current in the intermediate nodes.

In our on-demand scatternet formation and routing protocol, the scatternet is formed depending on traffic patterns, and no neighbor information between sensor nodes is exchanged. Therefore, when the scatternet and route request arrives at an intermediate node, the node makes the decision whether to forward it or not based on its own residual current level. Since prolonging the network lifetime results from delaying the first node to be energy depleted, our objective is to keep the energy consumption of all the nodes within the network at approximately the same rate, and avoid extra load at any given nodes. A node with the residual current (C_R) satisfying Eq. 1 stops forward route requests for other sensor nodes.

$$C_R < \left[C_B - \frac{T_{Active}}{2 * T_{Slot}} * (C_{MaxTx} + C_{MaxRx}) \right] \alpha, \qquad (1)$$

where C_B is the battery current, T_{Active} is device active time, T_{Slot} is Bluetooth slot time, C_{MaxTx} and C_{MaxRx} are maximum transmission and reception current, respectively, and α is the current factor.

If a node continually transmits and receives data packets during its active time, its residual current will equal the right side of Eq. 1 ($\alpha = 1$). However, during connection establishment, most of the packets being transmitted and received are small size control packets. Thus, for the same active time, the residual current should be larger than the extreme condition. In this case we use a current factor α. The tuning of α balances the traffic load throughout the network. When $\alpha = 1$, every node's residual current is greater than the extreme condition. Therefore, no traffic balance is achieved. With $\alpha > 1$, some intermediate sensor nodes with heavy forwarding traffic will satisfy Eq. 1 at some time, and thus other nodes with large residual current will take on the load. Since our scatternet and route formation is initiated in an on-demand manner, the nodes with faster current consumption previously may become available after some recovery time and assume forwarding responsibility again when new requests arrive. The tuning of the current factor is discussed later.

Performance Evaluation

In this section we provide a quantitative evaluation of our on-demand scatternet formation and routing protocol by means of analysis and simulation.

EID Power Saving

The power consumption due to the transmission of an EID packet is much more than that of smaller ID packets. Compared to [9, 11], which also introduce types of EID packets and substitute EID packets for ID packets in all cases, we retain the smaller ID packets for neighbor probing in *Inquiry* and only transmit EID packets when the source and last hop information is necessary for scatternet formation by our modified *Inquiry* process.

The power saving of our proposed *Inquiry* scheme is related to the period of *Inquiry* and the number of nodes participating in scatternet formation in the network.

According to the Bluetooth specification [2], the MASTER must stay in the *Inquiry* state for 10.24 s to collect sufficient responses from its neighbors. However, the time to get enough responses varies significantly depending on the alignments of device clocks. Simulations reveal that 5 s is sufficient most of the time.

The current consumption[2] comparison of our modified *Inquiry* to the traditional *Inquiry* with EID packets for one node is shown in Fig. 3. The current consumption goes up linearly as the time for *Inquiry* increases, due to more EID packets being transmitted. Howev-

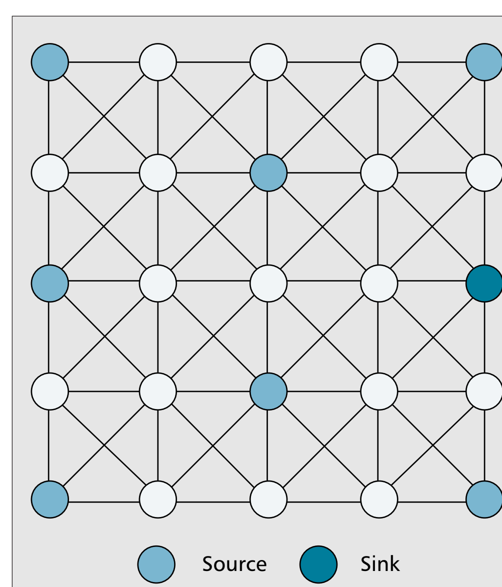

Figure 4. *Multihop network topology.*

[2] *The current drain for a typical Bluetooth device to transmit an ID packet is 26.5 mA; it is 39.8 mA for an EID packet.*

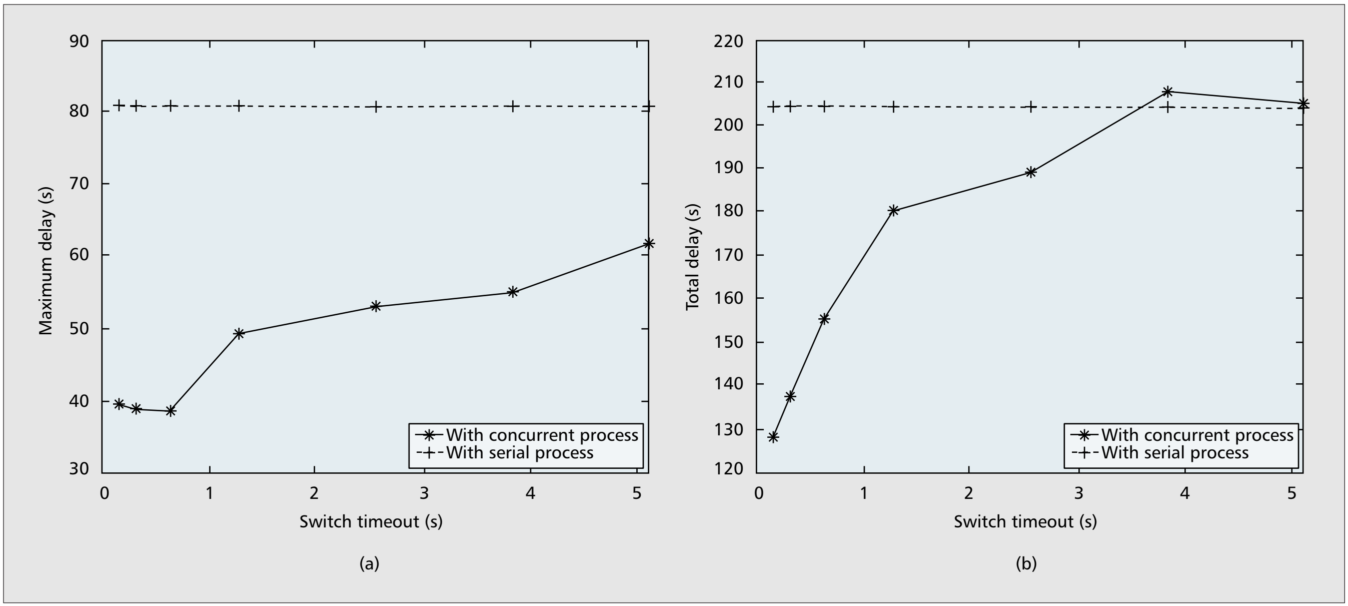

Figure 5. *Scatternet formation delay vs. switch timeout: a) maximum scatternet formation delay; b) total scatternet formation delay.*

er, the current consumption of our scheme achieves substantial savings. With *Inquiry* time of 10.24 s, 33.41 percent current saving is achieved by our modified *Inquiry* scheme. In WSNs with a large number of nodes, this saving is significant.

SCATTERNET FORMATION DELAY

Our intention in choosing on-demand rather than proactive scatternet formation is to conserve power used by connection maintenance for the entire network. However, the trade-off of on-demand approaches vs. proactive approaches is the scatternet formation delay incurred by on-demand methods. Therefore, the scatternet formation delay is an important metric. To measure the scatternet formation delay, we implemented our scatternet formation protocol in the Georgia Tech Network Simulator (GTNetS) [13], a packet-level simulator designed for large-scale network simulation. In our previous work we designed and implemented a detailed Bluetooth model for GTNetS [14]. The network topology we chose for our simulations is shown in Fig. 4. This grid topology has one sink and multiple sources. Every node not residing on edges has eight neighbors within its radio range. This topology is similar to a typical monitoring or data collection sensor network topology with all possible source distributions relative to the sink.

As stated earlier, the value of the switch timeout (*SwitchTO*), which controls the alternation of *Inquiry_Scan* and *Page_Scan* states after routing request forwarding, has significant effect on scatternet formation delay. In Fig. 5 we vary *SwitchTO* from 0.16 s to 5.12 s to measure the maximum as well as total delay for all the sources to finish the scatternet formation process. Figure 5a shows that the maximum scatternet formation delay with our simultaneous process for cross routes is greatly reduced compared to the serial formation process. The *SwitchTO* value of 0.64 s achieves the best performance for the maximum scatternet formation delay, which is only 47.94 percent of the delay with the serial process. The optimal *SwitchTO* value occurs at 0.64 s. This is because when the timeout value is too small, the node switches frequently between the two states (*Inquiry_Scan* and *Page_Scan*) and stays a short time in each state. Since Bluetooth uses FHSS, the switch interval is not enough for the *Inquiry* or *Page* nodes to hop to the frequencies being listened to by the scanning nodes. If the timeout value is too large, time is wasted waiting for state switch. This optimal *SwitchTO* is also coincident with the average *Page* delay, which is half of the *Page_Scan* window (1.28 s).

The total scatternet formation delay of all the sources is illustrated in Fig. 5b. The delay with the concurrent process is lower than that with the serial process (75.99 percent with *SwitchTO* of 0.64 s) when the *SwitchTO* value is less than 2.56 s, which is the value for *Page* timeout. The optimal *SwitchTO* is no longer 0.64 s. This is because some sources with small numbers of hops to the sink sacrifice their own formation delays, which increases the total formation delay, but benefits nodes with large numbers of hops from the sink to complete the scatternet formation process promptly.

Another parameter that must be tuned carefully is the timeout value for the scatternet formation (*ScatFormTO*). The source nodes initiate new scatternet formation requests when no responses arrive and *ScatFormTOs* expire. If the *ScatFormTO* is set to be too small, new requests are sent out before normal replies come back. Thus, the scatternet formation is initiated repeatedly without any success. On the other hand, setting *ScatFormTO* too large may incur unnecessary delay to wait for *ScatFormTO* and recover from failures. We vary the *ScatFormTO* from 10 s to 30 s to look for the optimal selection. We observe in Fig. 6 that the delays keep stable when *ScatFormTO* is less than 20 s and

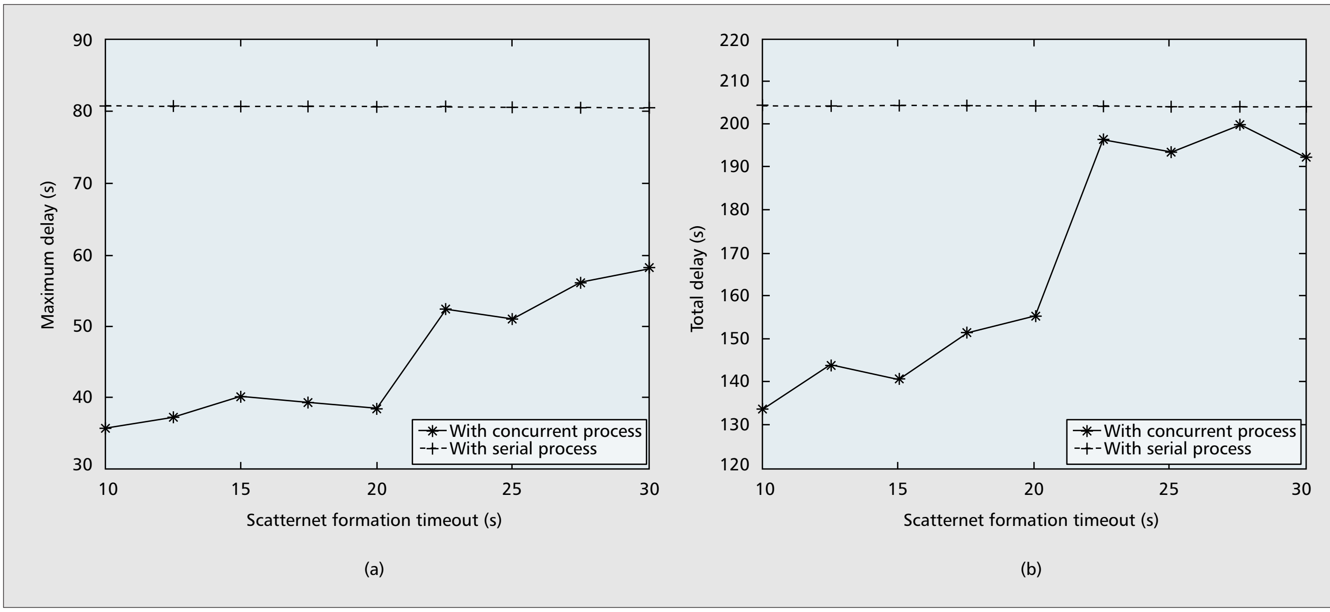

Figure 6. *Scatternet formation delay vs. scatternet formation timeout: a) maximum scatternet formation delay; b) total scatternet formation delay.*

increase significantly with *ScatFormTO* larger than 20 s. This is due to the time spent waiting in vain for a reply.

PROLONGING NETWORK LIFETIME

One of the benefits of on-demand scatternet formation and routing schemes compared to proactive methods is that traffic can be routed through different paths every time a new request is initiated. Therefore, the traffic load can easily be distributed in order to balance the energy consumption of each node and decelerate energy depletion of nodes lying along overloaded paths. We compare the lifetime of the network with and without an energy-aware scheme under the same traffic patterns. Simulation results for various numbers of simultaneous sources are shown in Fig. 7.

As discussed earlier, the tuning of the current factor α can balance the traffic in the network. We vary α from 1 to 1.8 to search for the best performance in resource utilization. When $\alpha > 1$, after some nodes participate in the formed scatternet and data relay, their residual current decreases. As new requests arrive at these nodes, they keep silent if their residual current satisfies Eq. 1. In this case other nodes with high residual current will accept the requests and forward the traffic.

From Fig. 7a it can be observed that the network lifetime first increases with the current factor, then decreases as the current factor goes up. The increase is due to the load balance discussed before. When the current factor becomes too large, no node considers itself capable of supporting the present scatternet formation and route request. Thus, a lot of energy is drained by the transmission of scatternet request control packets until some nodes can accept the requests as time goes on. The standard deviation is in the opposite trend of the lifetime since lower deviation means a higher degree of load balance, and hence longer lifetime.

CONCLUSIONS

We have described an on-demand scatternet formation and routing protocol used for Bluetooth-based wireless sensor network applications. We deal with the complicated problem of supporting multiple sources that initiate scatternet and route formation involving common Bluetooth devices at the same time. In addition, we modified the Bluetooth *Inquiry* process with extended ID packets for scatternet formation and route request propagation. We show the power efficiency of this scheme compared to traditional *Inquiry* with EID packets. Furthermore, we employed a mechanism using POLL packets in *Page* mode to transfer scatternet formation information without extra expense. The energy-aware scatternet formation and routing property makes the scheme attractive by prolonging the network lifetime, which is a primary concern for wireless sensor networks. Simulation results demonstrate that our protocol achieves significant improvement in scatternet formation delay over serial scatternet formation for multiple sources with concurrent traffic. The load balance scheme extends the network lifetime based on local information. It meets the requirements of Bluetooth-based wireless sensor networks in terms of power efficiency due to our on-demand rather than proactive approach to scatternet formation. At the same time, the protocol does not incur large scatternet formation delay and achieves uniform resource utilization.

REFERENCES

[1] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach,* Morgan Kaufmann, 2004.
[2] Bluetooth SIG, "Bluetooth Specification Version 1.1," http://www.bluetooth.com
[3] M. Leopold, M. B. Dydensborg, and P. Bonnet, "Bluetooth and Sensor Networks: a Reality Check," *Proc. 1st Int'l. Conf. Embedded Networked Sensor Sys.*, 2003, pp. 103–13.

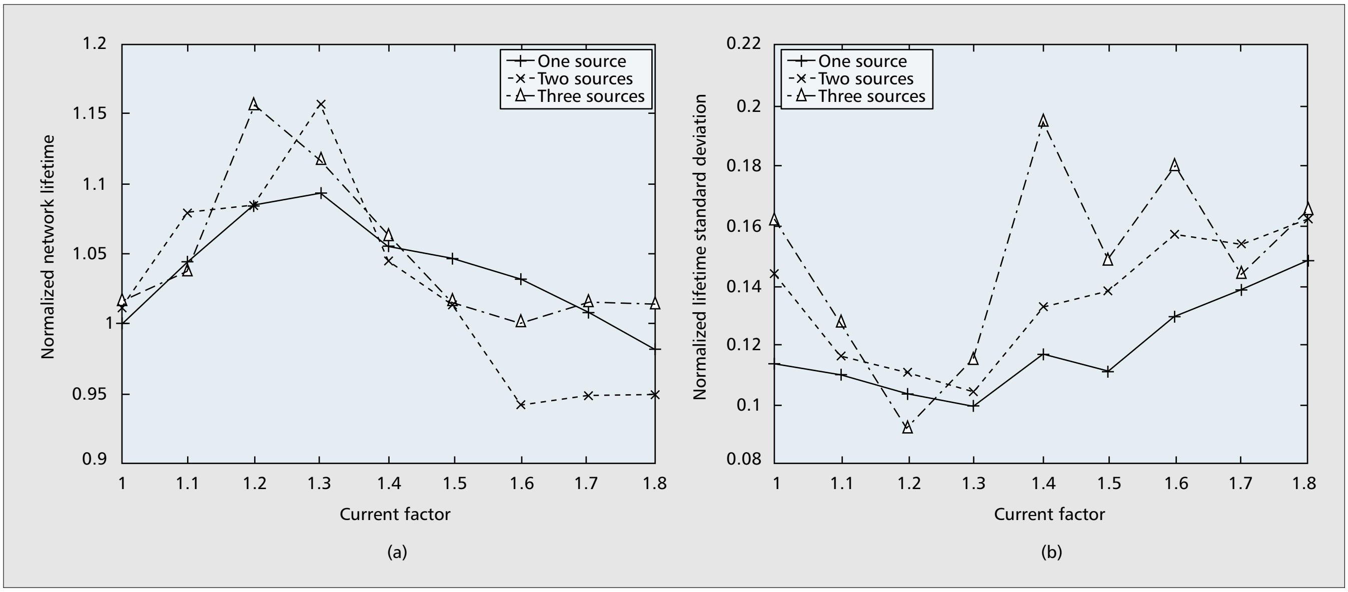

Figure 7. *Network lifetime comparison with/without load balance: a) network lifetime; b) network lifetime deviation.*

[4] J. Beutel *et al.*, "Prototyping Wireless Sensor Network Applications with BTnodes," *Proc. 1st IEEE Euro. Wksp. Wireless Sensor Net.*, 2004, pp. 323–38.

[5] R. M. Kling, "Intel Mote: An Enhanced Sensor Network Node," *Proc. Int'l. Wksp. Advanced Sensors, Structural Health Monitoring, and Smart Structures*, 2003.

[6] A. Mainwaring *et al.*, "Wireless Sensor Networks for Habitat Monitoring," *Proc. 1st ACM Int'l. Wksp. Wireless Sensor Networks and Apps.*, 2002, pp. 88–97.

[7] V. Mehta and M. E. Zarki, "A Bluetooth Based Sensor Network for Civil Infrastructure Health Monitoring," *ACM/Kluwer J. Wireless Net.*, vol. 10, no. 4, 2004, pp. 401–12.

[8] S. Basagni *et al.*, "Comparative Performance Evaluation of Scatternet Formation Protocols for Networks of Bluetooth Devices," *ACM/Kluwer J. Wireless Net.*, vol. 10, no. 2, 2004, pp. 197–213.

[9] Y. Liu, M. J. Lee, and T. N. Saadawi, "A Bluetooth Scatternet Route Structure for Multihop Ad Hoc Networks," *IEEE JSAC*, vol. 21, no. 2, 2003, pp. 229–39.

[10] E. Pagani, G. P. Rossi, and S. Tebaldi, "An On-demand Bluetooth Scatternet Formation Algorithm," *Proc. 1st IFIP Wkg. Conf. Wireless On-demand Net. Sys.*, 2004, pp. 130–43.

[11] Y. Kawamoto, V. Wong, and V. Leung, "A Two-Phase Scatternet Formation Protocol for Bluetooth Wireless Personal Area Networks," *Proc. IEEE Wireless Commun. and Net. Conf.*, 2003, pp. 1453–58.

[12] Crossbow Technology, "MICA2 AA Battery Pack Service Life Test," http://www.crossbow.com/Support

[13] G. F. Riley, "The Georgia Tech Network Simulator," *Proc. ACM SIGCOMM Wksp. Models, Methods, and Tools for Reproducible Net. Res.*, 2003, pp. 5–12.

[14] X. Zhang and G. F. Riley, "Bluetooth Simulations for Wireless Sensor Networks using GTNetS," *Proc. 12th IEEE Int'l. Symp. Modeling, Analysis, and Simulation of Comp. and Telecommun. Sys.*, 2004, pp. 375–82.

BIOGRAPHIES

XIN ZHANG (xinzhang@ece.gatech.edu) is a Ph.D candidate in the School of Electrical and Computer Engineering at Georgia Institute of Technology. She received her M.S. degree from Georgia Institute of Technology and B.E. degree from Beijing University of Posts and Telecommunications, China. Her research interests are in wireless and sensor network protocols, and large-scale simulation of wireless networks.

GEORGE F. RILEY (riley@ece.gatech.edu) is an assistant professor of electrical and computer engineering at Georgia Institute of Technology. He received his Ph.D. degree in computer science from Georgia Institute of Technology, College of Computing, in August 2001. His research interests are in large-scale simulation of computer networks, and networking research using these simulation techniques.