Light-Weight Forwarding Protocols in Energy Harvesting Wireless Sensor Networks

Cong Pu[†], Tejaswi Gade[†], Sunho Lim[†], Manki Min[§], and Wei Wang[‡]
[†]T[2]WISTOR: TTU Wireless Mobile Networking Laboratory
Dept. of Computer Science, Texas Tech University, Lubbock, TX 79409
{cong.pu, sunho.lim}@ttu.edu, tejagade@gmail.com
[§]Dept. of Electrical Engineering and Computer Science, South Dakota State University, Brookings, SD 57007
manki.min@sdstate.edu
[‡]Dept. of Computer Science, San Diego State University, San Diego, CA 92128
wei.wang@ieee.org

***Abstract*—Due to the unavoidable battery replacement or replenishment, diverse energy harvesting techniques have been integrated with Wireless Sensor Networks (WSNs) to overcome limited battery power and extend the network lifetime. However, variable transmission power levels based on non-uniform energy harvesting rates can incur asymmetric links. In this paper, we propose light-weight forwarding protocols to reliably deliver sensory data to a sink over time-varying asymmetric links in energy harvesting WSNs. A *Weighted Confirmation* (WCFM) scheme is proposed to differentiate multiple paths between a data source and a sink by assigning multiplicative weights on the paths. A *Lazy Confirmation* (LCFM) scheme is also proposed to assure a reverse path by waiting for extended communication range. In addition, an *Asymmetric Link Aware Backoff* mechanism is proposed to avoid possible packet contentions and collisions. We evaluate the proposed techniques through extensive simulation experiments and their results indicate that the proposed forwarding protocols can be a viable approach in energy harvesting WSNs.**

***Index Terms*—Asymmetric link, energy harvesting, forwarding protocol, wireless sensor networks.**

I. Introduction

Wireless Sensor Networks (WSNs) often require long-term sensing/communicating operations on the order of days or even weeks in a hostile and unattended area, e.g., deployed as a mission-oriented network. As pointed out in [1], a TMote™ Sky node consumes 64.68 mW in a receive mode. Under the two standard 3,000 mAh AA batteries, if the node is highly utilized, network lifetime is only 5.8 days. In battery-powered WSNs, replacing or replenishing the batteries is ultimately unavoidable and it may be infeasible or even impossible in such a harsh environment. Due to the limited battery power, therefore, WSNs powered by diverse environmental sources (i.e., solar, vibration, wind, thermal, etc.) have been widely investigated. This paper is also motivated by the fact that the U.S. Army will eventually eliminate all the military batteries or at least reduce the frequency of replacing batteries for communication devices [2]. Soldiers will be equipped with batteryless or self-powered communication devices in near future.

With energy harvesting, sensor devices (later nodes) may contain a different amount of residual energy because of non-uniform energy harvesting rates in WSNs. Depending on the energy availability, nodes can deploy variable transmission power levels and thus, multiple communication ranges commonly exist in the network. For example, variable transmission power levels are easily witnessed in the CISCO Aironet 340 and 350 series and Wi-Fi networks [3] to provide customized services, where computation power, storage limit, and energy consumption are selectively considered. Note that multiple communication ranges can lead to *asymmetric links*. For example, if a node, n_a, can reach nodes, n_b and n_c, but both n_b and n_c or either n_b or n_c may not be able to reach n_a.

Since each node can change its transmission power levels based on energy harvesting, a link between two nodes may not be stable. Thus, a route from a data source to a sink also may become unreliable in the presence of time-varying asymmetric links. A bidirectional routing [4] and a tier-based routing framework [5] deployed in asymmetric mobile ad hoc networks (MANETs) cannot directly be applied in resource constrained WSNs. A probabilistic routing [6] and a multiple range convergecast routing [7] have been proposed for heterogeneous WSNs. In these approaches, a small number of nodes is dedicated to communicate with the extended communication range, or each node is able to change its multiple transmission power levels anytime. However, time-varying communication ranges motivated by energy harvesting have not been well considered. To the best of our knowledge, little work has been devoted in a forwarding methodology in the realm of energy harvesting WSNs.

In this paper, we propose light-weight forwarding protocols to reliably deliver sensory data to a sink in the presence of time-varying asymmetric links in energy harvesting WSNs. Our contributions are three-fold:

- First, we propose *Weighted Confirmation* (WCFM) and *Lazy Confirmation* (LCFM) schemes to reliably deliver sensory data to the sink. The WCFM scheme differentiates multiple paths between a data source and a sink by assigning multiplicative weights on the paths. In the LCFM scheme, nodes assure a reverse path by waiting for the extended communication range.

978-1-4799-6770-4/14 $31.00 © 2014 IEEE
DOI 10.1109/MILCOM.2014.179

- Second, an *Asymmetric Link Aware Backoff* mechanism is also proposed to avoid packet contentions and collisions by considering the historical statistics of routing and number of neighbor nodes.
- Third, we evaluate the proposed WCFM and LCFM schemes and their hybrid approach, called *Hybrid Confirmation* (HCFM), using OMNeT++. We modify a conventional explicit acknowledgment, called *Conventional Ack* (CAck) scheme, to work in energy harvesting WSNs for performance comparison.

The WCFM, LCFM, and HCFM schemes show higher packet delivery ratio but keep lower latency compared with the CAck scheme. Overall simulation results indicate that the proposed forwarding protocols is a viable approach for reliable asymmetric routing in energy harvesting WSNs.

The rest of paper is organized as follows. A system model and the proposed forwarding protocols are presented in Section II. Section III presents simulation results and their analyses. Finally, we conclude the paper with future research direction in Section IV.

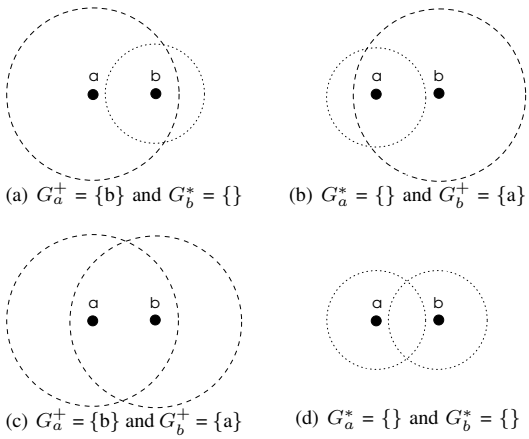
II. The Proposed Light-Weight Forwarding Protocols

In this section, we first present a system model of energy harvesting and neighbor node relationship in the network. Then we propose forwarding protocols in energy harvesting WSNs.

A. System Model

In this paper, energy harvesting is modeled by a two-state Markov process with harvest (S_{hv}) and normal (S_{nr}) states. In S_{hv} and S_{nr} states, nodes operate in the extended and normal communication ranges, respectively. A node stays in S_{nr} state for a random amount of time, which is exponentially distributed with a mean λ_{nr}, and changes its state into S_{hv} state. After energy harvesting for some amount of time in S_{hv} state, which is also assumed to be exponentially distributed with a mean λ_{hv}, the node changes its state back to S_{nr} state. Both S_{nr} and S_{hv} states are repeated. Upon energy harvesting, each node is able to operate in higher transmission power level to extend its current normal communication range.

Due to multiple communication ranges, each node can have a different set of neighbor nodes. In this paper, we consider dual communication ranges for the sake of simplicity and categorize node adjacency into four cases as shown in Fig. 1. Each node exchanges an one-hop *Hello* packet, overhears bypassing packets, and maintains a neighbor list, G. The list consists of a set of neighbor nodes reachable with either normal communication range (G^*) or extended communication range (G^+), respectively. For example, in Subfig. 1(a), n_a and n_b operate in the extended and normal communication ranges, respectively. n_a can communicate with n_b but n_b cannot. Similarly, in Subfig. 1(b), n_b can communicate with n_a but n_a cannot. Both n_a and n_b can communicate each other with the extended communication range in Subfig. 1(c).



(a) G_a^+ = {b} and G_b^* = {} (b) G_a^* = {} and G_b^+ = {a}

(c) G_a^+ = {b} and G_b^+ = {a} (d) G_a^* = {} and G_b^* = {}

Fig. 1. Neighbor lists with dual communication ranges. Here, normal and extended communication ranges are marked as dotted and dashed lines, respectively.

In Subfig. 1(d), both n_a and n_b are not located within their normal communication range.

B. Detail Operations

The proposed forwarding protocols consist of three major operations: broadcast-based forwarding, routing history update, and asymmetric link aware backoff. A basic idea is that each node forwards sensory data to the selected node based on its historical statistics of routing. Since a route from a data source to a sink is unreliable in the presence of time-varying asymmetric links, we do not maintain and update a routing table.

First, a simple broadcast-based forwarding is deployed to avoid the exchange of control packets and reliably deliver a data packet through multiple paths. Each node re-broadcasts the received data packet only if it has been forwarded from the node located further from the sink in terms of number of hops. Here, a sink floods an one-time *Hop* packet piggybacked with the number of hops (h, initially set by 0) to the rest of nodes at the initial network setup. When a node receives *Hop* packet, it increments h by one, stores the updated h in a local storage, and rebroadcasts the packet piggybacked with the updated h. When a node receives the *Hop* packet containing higher number of hops, h', it replaces h' with the stored h and rebroadcasts the packet. Thus, each node is aware of how many hops away from the sink.

Second, each node maintains a historical statistics of routing by updating a ratio of the number of delivered packets to the sink (d) to the number of forwarded packets (f), $DF = \frac{d}{f}$. When a node forwards a data packet, it increments the number of forwarded packets by different values (i.e., 1, 0.6, or 0.4). When the sink receives a data packet, it replies a confirmation (*Cfm*) packet, which is relayed back to a source node. When a node receives a *Cfm* packet, it increments the number of delivered packets. If a node has higher DF, it has frequently and successfully delivered data packets to the sink. Unlike prior approach [4], [6], each node does not actively find a

reverse path using additional control packets in energy harvesting WSNs. Note that a *Cfm* packet is relayed back to a source node through the intermediate nodes located along the path in best efforts. This is because of time-varying asymmetric links that incur frequent link disconnections. This approach is different from a conventional explicit acknowledgment scheme for the purpose of reliable routing, where a sink replies an acknowledgment (*Ack*) packet back to a data source if a data packet is successfully received. We observe that replying an *Ack* packet back to a data source is not very efficient in terms of *Ack* packet delivery ratio in energy harvesting WSNs, which supports our approach. As shown in Fig. 2, *Ack* packet delivery ratios against different energy harvesting periods are quite low because of time-varying asymmetric links. Since the data source frequently experiences timeouts and executes retransmissions, a large number of data packets are lost. In this paper, we do not consider an implicit acknowledgment scheme based overhearing [8], because the radio should be kept active, resulting in a non-negligible energy consumption.

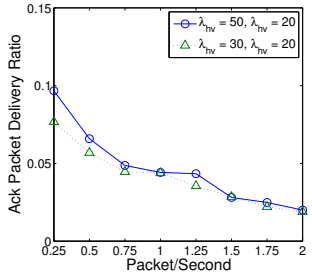


Fig. 2. Acknowledgment packet delivery ratio as a function of mean periods of harvest (S_{hv}) and normal (S_{nr}) states.

Due to the multiple paths, the sink may receive the same data packet from a data source multiple times. Upon receiving a data packet, the sink determines whether it already has received the packet routed with the same path. If not, the sink replies a *Cfm* packet. The sink also replies to the later arriving data packets routed with different paths. The sink accepts upto three duplicated data packets routed with different paths. Whenever the sink replies a *Cfm* packet, it piggybacks an increment factor (Δ, initially set by a value 1 (Δ_1)) into the *Cfm* packet. In this paper, we propose a *Weighted Confirmation* (WCFM) scheme. Whenever the sink repeatedly replies a *Cfm* packet for the same data packet, it reduces the increment factor, i.e., Δ_1, $\Delta_{0.6}$, and $\Delta_{0.4}$. When a node receives a *Cfm* packet, it adds the piggybacked increment factor to the current number of delivered packets (d). Thus, the DF increases with different increment factors. Δ_1 is assigned to the first arriving data packet, because it is expected that the packet has been routed through the shortest path or the path with higher DF. Intermediate nodes can forward the *Cfm* packet at most three times and adjust their number of forwarded packets accordingly. Here, we multiplicatively adjust the increment factor to clearly see the effect of the WCFM scheme on the performance. The rationale behind this approach is to have nodes with higher DF involve in the routing operation frequently and deliver data packets reliably. Due to the communication overhead, this approach limits the number of multiple paths by deploying three increment factors.

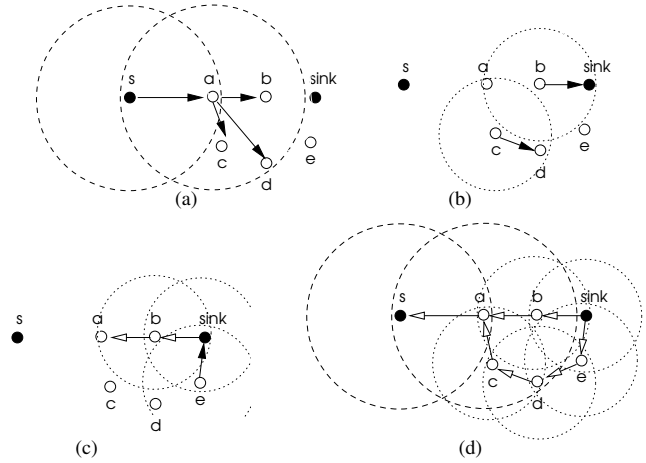


Fig. 3. The proposed WCFM scheme. Here, both data and *Cfm* packets are forwarded to the directions where black and white arrows indicate, respectively.

For example, n_s initially generates a data packet and sends it toward a sink in Subfig. 3(a). Both n_s and n_a operate in an extended communication range and the rest of nodes operate in a normal communication range. n_b, n_c, and n_d receive the forwarded data packet from n_a. Although n_c and n_d receive the packet simultaneously, let say, n_c forwards it. In Subfig. 3(b), n_b and n_c forward the packet and the sink receives the packet from n_b. The sink replies a *Cfm* packet piggybacked with the increment factor (i.e., Δ_1) to n_b and receives the same data packet from n_e, as shown in Subfig. 3(c). The sink also replies another *Cfm* packet piggybacked with a reduced increment factor (i.e., $\Delta_{0.6}$) for the later arriving packet. Multiple *Cfm* packets are sent back to the data source, n_s, through multi-hop relay as shown in Subfig. 3(d). All the intermediate nodes located along to the path between n_s and the sink update their DF based on the increment factors piggybacked in the *Cfm* packets. The pseudo code of major operations in the WCFM scheme is summarized in Fig. 4.

If n_a shrinks back to a normal communication range, its reverse link to n_s will be disconnected as shown in Subfig. 5(a). To support this, we propose a *Lazy Confirmation* (LCFM) scheme, where n_a does not search a reverse path to n_s but buffers any incoming *Cfm* packets. Then when n_a operates in an extended communication range, it forwards the buffered *Cfm* packets to n_s, as shown in Subfig. 5(b). A basic idea of the LCFM scheme is that nodes conservatively forward both data and *Cfm* packets only when their reverse path is available. In contrast to the WCFM scheme, the LCFM scheme does not adjust increment factor but always piggybacks Δ_1 to the *Cfm* packet. The pseudo code of major operations in the LCFM scheme is summarized in Fig. 6.

Third, we deploy a simple CSMA/CA MAC protocol for

Notations:
• DF_i, d_i, f_i: defined before.
• Δ_w, $|\Delta|$: An increment factor (w is 1, 0.6, or 0.4) and its number of increment factors, which is three.
• $pkt[type, src, seq]$: A packet is originally sent from a source node, n_{src}, with a sequence number, seq. Here, $type$ is either *Data* or *Cfm*.
• $Q_i[pkt[seq]]$: A queue of received packets in n_i.
• $C_i[pkt[seq]]$: A counter of received the same packets in n_i.
◇ When a sink, n_{sink}, receives a $pkt[Data, s, seq]$,
  if $pkt[seq] \notin Q_{sink}$
    Enqueue the $pkt[seq]$ into Q_{sink}, and $C_{sink}[pkt[seq]]$ ++;
    Reply the $pkt[Cmf, sink, seq]$ with $\Delta_{1.0}$;
  else
    $C_{sink}[pkt[seq]]$ ++;
    if $C_{sink}[pkt[seq]] == |\Delta| - 1$
      Reply the $pkt[Cmf, sink, seq]$ with $\Delta_{0.6}$;
    else if $C_{sink}[pkt[seq]] == |\Delta|$
      Reply the $pkt[Cmf, sink, seq]$ with $\Delta_{0.4}$;
    else
      Discard the pkt;
◇ When a node, n_i, receives a $pkt[type, s, seq]$,
  if $pkt[type] ==$ *Data*
    if $pkt[seq] \notin Q_i$
      $C_i[pkt[seq]]$ ++;
      Enqueue the $pkt[seq]$ into Q_i;
      f_i ++ and update DF_i;
    else
      $C_i[pkt[seq]]$ ++;
      if $C_i[pkt[seq]] == |\Delta| - 1$
        f_i += $\Delta_{0.6}$ and update DF_i;
      else if $C_i[pkt[seq]] == |\Delta|$
        f_i += $\Delta_{0.4}$ and update DF_i;
      else
        Discard the pkt and return;
    t_i^{boff} = Minimum($\frac{f_i}{|G_i| \cdot d_i} \cdot cw + \delta_i, cw) \cdot t_s$; /* Eq. 1 */
    if overhear a $pkt'[Data, k, seq]$ during t_i^{boff}, $k \in G_i^*$
      Discard the pkt;
    else
      Re-broadcast the pkt;
  else /* $pkt[type] ==$ *Cfm* */
    d_i += Δ_w, and update DF_i;
    Unicast the pkt after t_i^{boff};

Fig. 4. The pseudo code of the WCFM scheme.

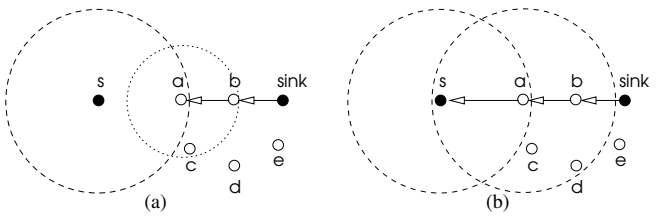


Fig. 5. The proposed LCFM scheme.

the link layer and propose an *Asymmetric Link Aware Backoff* mechanism. Whenever a node receives a data packet, it executes a backoff procedure before forwarding the packet to avoid possible packet contentions and collisions. A basic idea is that a node containing higher DF has lower backoff period because of its successful history of data deliveries. Also a node operating in an extended communication range has lower backoff period because of its potential to reduce the

Notations:
• $B_i[pkt]$: A buffer of received packets in n_i.
◇ When a sink, n_{sink}, receives a $pkt[Data, s, seq]$,
  if $pkt[seq] \notin Q_{sink}$
    Enqueue the $pkt[seq]$ into Q_{sink};
    Reply the $pkt[Cmf, sink, seq]$;
  else
    Discard the pkt;
◇ When a node, n_i, receives $pkt[Data, sink, seq]$,
  if $pkt[seq] \notin Q_i$
    Enqueue the $pkt[seq]$ into Q_i;
    f_i ++ and update DF_i;
    t_i^{boff} = Minimum($\frac{f_i}{|G_i| \cdot d_i} \cdot cw + \delta_i, cw) \cdot t_s$;
    if overhear a $pkt'[Data, k, seq]$ during t_i^{boff}, $k \in G_i^*$
      Discard the pkt;
    else
      Re-broadcast the pkt;
  else
    Discard the pkt;
◇ When a node, n_i, receives $pkt[Cfm, sink, seq]$,
  if $pkt[seq] \notin Q_i$
    Enqueue the $pkt[seq]$ into Q_i;
    d_i += Δ_1 and update DF_i;
    if a reverse path to n_j is available, $n_j \in G_i^*$
      Unicast the pkt to n_j after t_i^{boff};
    else
      Enqueue the pkt into B_i;
      Unicast the pkt to n_j after t_i^{boff}, when the reverse path is available;
  else
    Discard the pkt;

Fig. 6. The pseudo code of the LCFM scheme.

transmission latency by shortening the number of hops to the sink. To calculate a backoff period, we consider both the DF and the number of neighbors (i.e., $|G^*|$ or $|G^+|$). For example, when a node n_i receives a data packet, its backoff period is expressed as,

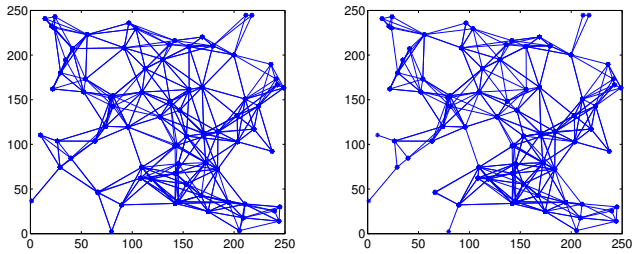
$$t_i^{boff} = \text{Minimum}(\frac{f_i}{|G_i| \cdot d_i} \cdot CW + \delta_i, CW) \cdot t_s, \quad (1)$$

where $|G_i|$ becomes either $|G_i^*|$ or $|G_i^+|$ depending on the current transmission power level. Here, $|G_i^+| \geq |G_i^*|$. Also δ_i becomes Uniform$(0, |G_i|)$. In case of $G_i^+ = \{\}$, which means $|G_i^*| = 0$ and $|G_i| = 0$, we replace $|G_i|$ with 1. A small contention window (CW) value (i.e., 32 slots) is used and each slot is 400 μsecs, similar to [9], t_s. If a node overhears a packet being routed during the backoff period, it aborts the backoff procedure and discards the received packet. Upon the backoff expire, if the node does not overhear a packet, it forwards the received packet.

III. Performance Evaluation

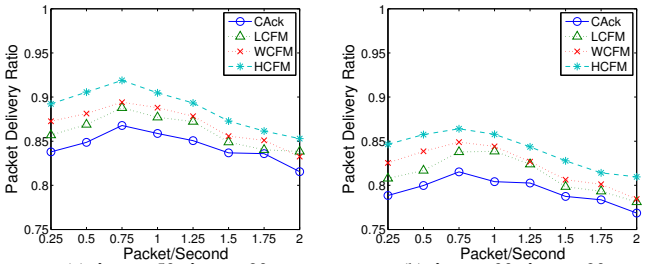
A. Simulation Testbed

We develop a customized discrete-event driven simulator using OMNeT++ [10] to conduct our experiments. A 250×250 m^2 rectangular network area is considered, where 140 nodes are randomly distributed in the network. An initial network topology is set in Subfig. 7(a), and it changes over simulation



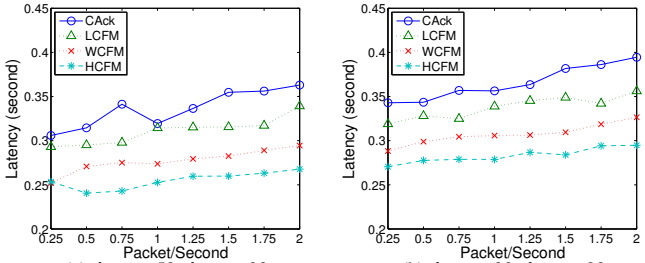
(a) Initial network topology (b) After 500 seconds

Fig. 7. Snapshots of network topology.



(a) λ_{hv} = 50, λ_{nr} = 20 (b) λ_{hv} = 30, λ_{nr} = 20

Fig. 8. Packet delivery ratio as a function of mean periods of energy harvesting and normal states.



(a) λ_{hv} = 50, λ_{nr} = 20 (b) λ_{hv} = 30, λ_{nr} = 20

Fig. 9. Latency as a function of mean periods of energy harvesting and normal states.

time due to the time-varying asymmetric links in Subfig. 7(b). The radio model simulates CC2420 with a nominal data rate of 250 Kbps [11]. The radio propagation model is based on the free-space model. A single node generates data traffic with 0.25 to 2 packet injection rates and the data packet size is 1 KByte. The periods of energy harvesting and normal states are assumed to be exponentially distributed with mean λ_{hv} (50 and 30 seconds) and λ_{nr} (20 second), respectively. Depending on the state, normal and extended communication ranges are 40.8 m and 52 m. The total simulation time is 1,000 seconds.

B. Simulation Results

We vary the key simulation parameters: packet injection rate and period of energy harvesting and normal states. Combinations of the simulation parameters are used to conduct extensive performance evaluation studies. Five performance matrics are measured: packet delivery ratio (PDR), latency, and changes of increment factors, DF, and backoff period. For performance comparison, we modify a conventional explicit acknowledgment mechanism to work in energy harvesting WSNs, called *Conventional Ack* (CAck) scheme as a base case. In the CAck scheme, a sink replies an *Ack* packet only to the first arriving data packet with the increment factor, Δ_1. The intermediate nodes located along the path relay the *Ack* packet back to a data source after a random backoff. For the sake of simplicity, we do not consider a timeout mechanism for retransmission in the data source in this paper. Based on the proposed WCFM and LCFM schemes, we also propose a hybrid approach by combining the weighted factor and reverse path, called *Hybrid Confirmation* (HCFM) scheme. In the HCFM scheme, the sink replies multiple *Cfm* packets piggybacked with multiplicative weights to the later arriving data packets. The intermediate nodes located along the path to the data source relay the *Cfm* packet after the asymmetric link aware backoff. They buffer any incoming *Cfm* packet, if the reverse path is not available.

1) Packet Delivery Ratio: Fig. 8 shows the PDR of four different schemes with varying packet injection rates and periods of energy harvesting and normal states in time-varying network topologies (see Fig. 7). Under longer energy harvesting period, as shown in Subfig. 8(a), higher PDR is achieved because more nodes operate in an extended communication range. Thus, each node is less likely disconnected with its neighbor nodes. The proposed WCFM, LCFM, and HCFM schemes show higher PDF than that of the CAck scheme. This is because of multiple *Cfm* packets with multiplicative increment factors and buffering any incoming *Cfm* packet, if the reverse path is not available, positively affects the PDR. The HCFM scheme shows the highest PDR for entire packet injection rates because it can identify and deploy multiple reliable paths based on the DF. However, the CAck scheme shows the lowest PDR for entire packet injection rates because data packets are routed through a single path, which is a time-varying asymmetric link and becomes unreliable. In Subfig. 8(b), overall PDRs decrease and performance saturation is delayed to 1.0 packet injection rate under shorter energy harvesting period.

2) Latency: Fig. 9 shows the latency of four different schemes. In Subfig. 9(a), the HCFM scheme shows the lowest latency for entire packet injection rates because data packets can be delivered reliably through multiple paths based on the DF. Multiple *Cfm* packets with extended communication range can increase the DF, reduce the backoff period, and identify the best path to the sink. Thus, the lowest latency can be achieved. Compared with the LCFM scheme, the WCFM scheme shows shorter latency because multiple *Cfm* packets can provide higher DF value that can lead to shorter backoff period. The CAck scheme shows the highest latency because of a blind random backoff period without considering asymmetric links. In Subfig. 9(b), four schemes show a similar pattern of the latency but higher latency is observed compared with the longer harvest period. Due to the short period of extended communication range, more link disconnections and longer waiting time of reverse paths are expected.

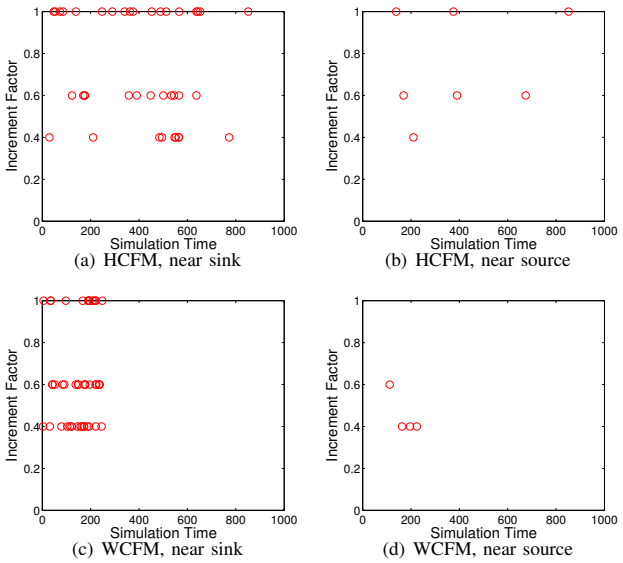


Fig. 10. Changes of received increment factor. Here, λ_{hv} = 50, λ_{nr} = 20, and packet injection rate = 1 packet/second.

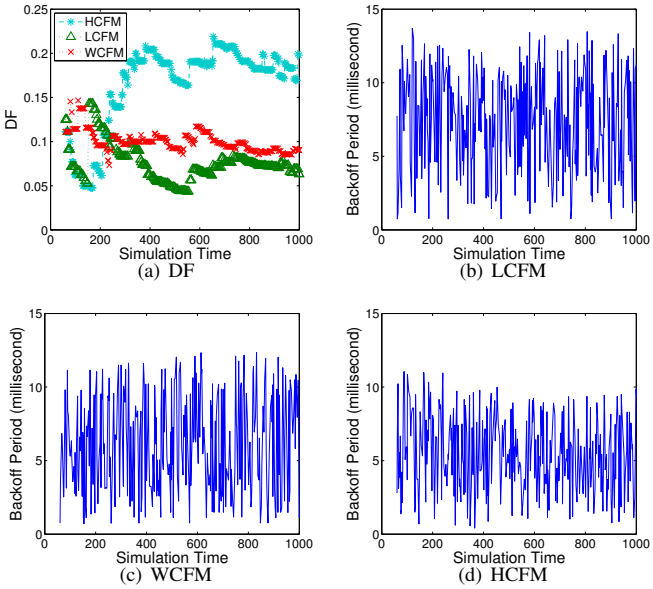


Fig. 11. Changes of the DF ratio and backoff period. Here, λ_{hv} = 50, λ_{nr} = 20, and packet injection rate = 1 packet/second.

3) Increment Factor: We randomly select the nodes located near to the sink and data source to trace the values of increment factors, piggybacked in *Cfm* packets, in the HCFM and WCFM schemes. In Subfig. 10(a), the node receives many increment factors in the HCFM scheme. The node can receive multiple *Cfm* packets with different increment factors because the sink replies a *Cfm* packet to the later arriving data packet. More number of Δ_1 than $\Delta_{0.6}$ and $\Delta_{0.4}$ are observed because *Cfm* packets for later arriving data packets are routed through a longer path or less reliable path. Since the node's DF increases, it is more frequently involved in the forwarding and thus, more number of *Cfm* packets are received. In Subfig. 10(b), however, the node has not been involved in the forwarding and receives very few *Cfm* packets. Subfig. 10(c) shows the effect of time-varying asymmetric links in the WCFM scheme. The node has been actively involved in the forwarding but in later it is removed from the path due to the asymmetric links. Unlike to the LCFM and HCFM schemes, where any incoming *Cfm* packets are buffered, the node can lose *Cfm* packets in the WCFM scheme. Subfig. 10(d) shows that the node almost does not receive *Cfm* packets and thus, it is rarely considered as a forwarding node.

4) DF and Backoff Period: In Subfig. 11(a), we observe the changes of the DF in the WCFM, LCFM, and HCFM schemes. The HCFM scheme shows higher DF than that of other schemes because the sink replies multiple *Cfm* packets, which can also be buffered. Thus, more intermediate nodes can update their DF, i.e., increment the number of forwarded packets. The LCFM scheme shows the lowest DF because the sink replies only to the first arriving data packet with the increment factor of Δ_1. The WCFM scheme shows higher DF than that of the LCFM scheme because of multiple *Cfm* packets with different factors. In Subfigs. 11(b), (c), and (d), we compare the backoff periods of three schemes. Since the backoff period is based on the DF and the number of neighbor nodes, the HCFM scheme shows the lowest backoff period compared with other two schemes. The LCFM scheme shows the highest and highly fluctuated backoff period. Note that the average backoff periods of the LCFM, WCFM, and HCFM schemes are 7.3619 msec, 6.8976 msec, and 5.2905 msec, respectively.

IV. Conclusion and Future Work

In this paper, we investigated light-weight forwarding protocols in the presence of time-varying asymmetric links in energy harvesting WSNs. We proposed weighted and lazy route confirmation schemes and an asymmetric link aware backoff mechanism to reliably deliver sensory data. We evaluated their performance through extensive simulation experiments, compared them with an modified conventional explicit acknowledgment scheme, and showed that the proposed forwarding protocols is a viable approach in energy harvesting WSNs.

To see the full potential of the proposed techniques, we relax our assumption on energy harvesting from environmental sources in WSNs. We implicitly assumed that each node uniformly harvests energy and extends its communication range in the network. We are currently investigating a piezoelectric (later piezo) based energy harvesting from ambient vibrations [12] in a mobile tactical network, where only actively moving nodes harvest energy and communicate with an extended communication range. For example, each soldier equipped with a piezo-based energy harvesting kit in his/her shoes moves according to a tactical maneuver within the network and disseminates captured information with other soldiers. We envision that the proposed forwarding protocols can be integrated for reliable data dissemination in a mobile tactical

network.

REFERENCES

[1] D. Raymond, R. Marchany, M. Brownfield, and S. Midkiff, "Effects of Denial of Sleep Attacks on Wireless Sensor Newtork MAC Protocols," in *Proc. Workshop on Information Assurance*, 2006, pp. 297–304.

[2] *A07-034 (Army), Harvesting Energy for Wireless Sensor Networks*, http://www.dodsbir. net/sitis/.

[3] A. Akella, G. Judd, S. Seshan, and P. Steenkiste, "Self-management in Chaotic Wireless Deployments," in *Proc. ACM MOBICOM*, 2005, pp. 185–199.

[4] V. Ramasubramanian and D. Mosse, "BRA: A Bidirectional Routing Abstraction for Asymmetric Mobile Ad Hoc Networks," *IEEE/ACM Trans. on Networking*, vol. 16, no. 1, pp. 116–129, 2008.

[5] T. Le, P. Sinha, and D. Xuan, "Turning heterogeneity into an advantage in wireless ad-hoc network routing," *Ad Hoc Networks*, no. 8, pp. 108–118, 2010.

[6] Y. Hu, W. Li, X. Chen, S. L. Xin Chen, and J. Wu, "A Probabilistic Routing Protocol for Heterogeneous Sensor Networks," in *Proc. IEEE NAS*, 2010, pp. 19–26.

[7] B. Romdhani, D. barthel, and F. Valois, "Routing for Data-Collection in Heterogeneous Wireless Sensor Networks," in *Proc. IEEE VTC*, 2011, pp. 1–5.

[8] S. Lim, C. Yu, and C. R. Das, "RandomCast: An Energy Efficient Communication Scheme for Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 8, pp. 1039–1051, 2009.

[9] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," in *Proc. Sensys*, 2004, pp. 95–107.

[10] *OMNeT++*, http://www.omnetpp.org/.

[11] *Castalia – A Simulator for WSNs*, http://castalia.npc.nicta.com.au/.

[12] S. Lim, J. Kimn, and H. Kim, "Analysis of Energy Harvesting for Vibration-Motivated Wireless Sensor Networks," in *Proc. International Conference on Wireless Networks*, 2010, pp. 391–397.