Journal of Computer Science

Original Research Paper

On the Need of Source Address for Route-Error Delivery in 6LoWPAN

Shafique Ahmad Chaudhry

Department of Computer Science, Dhofar University, Salalah, Oman

Article history
Received: 03-04-2015
Revised: 11-11-2015
Accepted: 13-11-2015

E-mail: shafique@du.edu.om

Abstract: Low-power Wireless Personal Area Networks (LoWPANs) comprise devices that conform to the IEEE standard 802.15.4. These networks need to be connected with other wireless and wired networks in order to maximize the utilization of information and other resources which are mainly associated with the Internet Protocol (IP)-based networks. The transmission of IPv6 over Low Power Wireless Personal Area Networks (6LoWPANs), requires a fragmentation and reassembly layer that also carries out header compression for transmission efficiency. The existing proposed header format includes the originator's address in adaptation layer, to ensure along with other issues, that in case of a link failure, route error messages are delivered back to the originator. We propose the Unicast Back-Propagation Mechanism (UBP) that delivers the Route Error Message (RERR) to the source even without having the originator's address in 6LoWPAN adaptation layer packet format. We make use of MAC layer address for sending the RERR to the previous hop node, back tracking the route hop by hop, eventually to the source. The simulation results show that our solution provides considerable header compression yet shows better diagnostic performance as the RERR delivery mechanism in standard ad-hoc routing.

Keywords: LoWPAN, Route Error Delivery, Unicast Back-Propagation, 6LoWPAN

Introduction

Low-power wireless personal area networks (LoWPANs) comprise devices that conform to the IEEE 802.15.4-2003 standard (IEEE LoWPAN, 2003). A LoWPAN typically includes devices that work together to connect the physical environment to real-world applications, e.g., wireless sensors. Generally, these networks consist of large number of sensor nodes, densely deployed in a specific region of interest. They are designed to sense or detect an event, collect the data and transmit it to the sink i.e., a designated node or user. IEEE 802.15.4 devices are extremely resource constrained in terms of power, computation and communication capabilities.

The LoWPANs need to be connected with other wireless and wired networks in order to maximize the utilization of information and other resources which are mainly associated with the Internet Protocol (IP)-based networks. Likewise, the information available on the LoWPAN domain may be equally important to the IP community. The motivation for IP connectivity, in fact, is manifold: (a) The pervasive nature of IP networks allows the use of existing infrastructure, (b) IP based technologies, along with their diagnostics, management and commissioning tools, already exist and are proven to be working and (c) IP based devices can more easily be connected to other IP networks, without the need for translation gateways etc.

The 6LoWPAN (6LoWPAN, 2005) working group, Standardizes the use of IPv6 over IEEE 802.15.4. There are many characteristics of IPv6 which make it a strong choice for its integration with IEEE 802.15.4 networks. First, the large number of devices in a LoWPAN make manual network configuration highly infeasible. Therefore, network auto configuration and statelessness is strongly desirable in LoWPANs, for which, IPv6 has ready solutions. Second, the large number of devices requires a huge address space and is duly provided by IPv6 addressing. Third, given the limited packet size of LoWPANs, the IPv6 address format allows subsuming of IEEE 802.15.4 addresses if so desired.

The RFC 4919 (Kushalnagar *et al*., 2007) describes the problems and challenges associated with transmission of IPv6 over LoWPAN. The hardest

problem of incompatibility arises due to the difference in packet sizes in both the technologies. The Maximum Transmission Unit (MTU) for IPv6 is at least 1280 octets, which cannot be mapped directly onto IEEE 802.15.4 physical layer frame, which is 127 octets-that further includes the 25 octets Media Access Layer (MAC) header. Remaining 102 octets at the MAC sublayer may include a maximum of 21 bytes for link layer level security, leaving 81 octets at higher layers. The 40 octets of IPv6 header leave 41 octets for upper layer protocols, e.g., User Datagram Protocol (UDP). UDP uses 8 bytes header that would leave only 33 octets for application data. This situation demands for an adaptation layer, for fragmentation and reassembly, below IP layer. The proposed adaptation layer consumes even more octets. All these facts support the disposition that header compression is inevitable. Although a header compression mechanism has also been presented in RFC 4944 (Montenegro *et al*., 2007) along with a proposed adaptation layer, we observe that in the current work, still there is room for further compression. The existing packet-format includes the originator's address field, i.e., EUI-64 bit address or 16-bit short address, to ensure that, in case of a link failure, Route Error messages (RERR) are delivered to the originator.

We contend that the RERR messages can be sent to the originator even without having the originator's address in the adaptation layer. In this study, we propose Unicast Back-Propagation (UBP), a scheme to deliver the RERR message to the originator, without using originator's address, i.e., IEEE defined 64-bit Extended Unique Identifier (EUI-64) address or 16-bit short address. Our mechanism helps header compression up to a maximum of 64 bits and yet shows better performance against the approach when the originator's address is available.

Related Work

For routing in 6LoWPAN, the Ad-hoc On-Demand Distance Vector (AODV) (Perkins and Royer, 2000) has been considered as a strong candidate because of its efficacy for finding routes. AODV is a reactive routing protocol, which provides route discovery and route maintenance. It uses well known Route Request (RREQ) and Route Reply (RREP) messages for route discovery. A pre-cursor list is associated with each entry in the routing table. This list contains the upstream nodes that use this very node to forward traffic towards the same destination. The Route Error (RERR) messages allow AODV to update the routes when there are link breaks. These link breaks can result due to the occurrence of many events including node mobility, node battery power drainage and environmental interference etc. In case there is no mechanism for RERR generation, the originator will keep sending the data packets which results into data loss, reducing the network throughput and performance. In AODV, a node, say 'N' initiates the RERR in three scenarios:

- When *N* receives a data packet for forwarding but it does not have a routing table entry for the destination-this situation implies that some nodes contain stale routing entry that a certain destination is accessible through node *N*. In this case the node *N* initiates a *RERR* and delivers it to its precursors, so that the invalid information in the routing tables can be corrected
- When *N* receives a *RERR* message which invalidates at least one of its routing table entries. The node *N* then sends a *RERR* to the precursor list corresponding to the entries which have become invalid because of the received *RERR*
- When a node detects that it cannot communicate with its neighbor node. This can be known by the absence of *hello* messages or when link layer reports a link failure. In such case, the node initiates a *RERR* based on this link failure

In response to any of the above mentioned situations, the RERR may be sent to all the pre-cursors using broadcast, unicast or iterative multi-cast. The RERR traffic overhead largely depends upon the number of pre-cursors and active paths' lengths.

Due to the heavy computing and memory requirements imposed by AODV, some slimmer versions of AODV, e.g., AODVjr (Chakeres and Luke, 2002) and AODV for WSN (Salom *et al*., 2012) have also been proposed. In AODVjr the RERR is not supported and the originator nodes initiates fresh route discovery if it does not receive connect message from the destination node for a specific time. The absence of RERR can reduce the system throughput considerably as it takes a while for the originator node to know that a link is broken. TinyAODV provides the RERR delivery but it also needs originator's address to deliver the RERR to the originator.

The 6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD) (Kim *et al*., 2007) is another simplification of AODV for 6LoWPAN. Contrary to AODV, LOAD does not use the destination sequence number, Gratuitous RREP, precursor list and hello packets. LOAD generates a Route Error (RERR) message toward the originator of the data delivery when it detects that the destination is no longer reachable by way of the broken link. In LOAD, RERR is forwarded only to the originator of the failed data delivery. The format of RERR is also simplified to include only one unreachable destination while the RERR of AODV may include multiple ones. LOAD assumes to have originator's address and does not discuss the situation where RERR reporting could be done without having originator address.

Unicast Back-Propagation Algorithm (UBP)

The key idea stems out from the way Internet Protocol (IP) works and interacts with the link layer technologies. In an end-to-end transmission, the originator and destination IP addresses remain the same-whereas the Media Access Control (MAC) addresses for the source and the destination change at every hop. When a node receives a packet from previous node, the receiving node knows the MAC address of the previous hop node. Against this received message, a control message can be sent to the previous hop node, recursively to the originator, without requiring the originator's address.

The main idea in this study is to propagate the *RERR* message one hop backwards for every frame transmission from the source, forwards. We make use of MAC layer address for sending the *RERR* only to the previous hop node. At each step, i.e., on arrival of next link layer frame, the *RERR* is propagated to the node which is one-hop closer to the source. In this fashion the *RERR* is delivered, back tracking the route of the frame, hop by hop in discrete steps, eventually to the source.

Figure 1 describes the process, when node *A* receives, from the previous node *B*, a link layer frame for node *D,* after detecting the link break between *A* and *D.* Node *A,* then sends a *RERR* message, notifying node *B* that the route to node *D* is no more available. Node *B* updates it routing table by deleting the respective entry for node *D*. The phenomenon at node *A* is repeated through node *B, C, D*, all the way till node *X*. The *RERR* is eventually delivered to the originator node *S*, when the node *X* sends a *RERR* to node *S*. The UBP algorithm is formally described in Fig. 2. Unlike RERR reporting mechanism in AODV, UBP does not notify all the sources for each link failure, because the main purpose of the algorithm is to notify the source nodes individually. The *RERR* provision mechanism in AODV aims to notify all the potential sources in order to prevent any data loss which could occur because of the broken link. Therefore, each *RERR* message generation initiates a limited flooding within in the network. In a 6LoWPAN a route is generally setup for event sensing and notification and is used sporadically. This feature is in fact the basis for selecting such a *RERR* mechanism for 6LoWAN. This could be viewed as a tradeoff between data loss and *RERR* traffic overhead.

The unicast from the node which initiates a *RERR* to the originator requires the source address which could be a 16-bit short address or EUI-64 bit address. UBP obviates the requirement of having source address in the adaptation layer and helps compress the header. This compression means additional payload capacity with no extra communication cost per data packet-resulting into increased Good put of 6LoWPAN.

We have also proposed two variants of UBP by considering delay and throughput as optimization problems. These are Broadcast Back-Propagation and Route-aware Back- Propagation algorithms. These two variants help execute the delivery of *RERR* to the source without using source address in the adaptation layer header.

Broadcast Back-Propagation Algorithm (BBP)

BBP works similar to the unicast propagation mechanism except that the failure detecting node broadcasts the route error message to its single hop neighbors and keeps a TTL value of '1'. This kind of transmission gives rise to one hop broadcast. The neighbors update their routing tables by deleting this destination's route entry. In case any of these neighbors receives a packet for the very destination, it will broadcast the route error message and thus the route error will be propagated back. The algorithm is presented in Fig. 3.

It may take the same time as the unicast back propagation mechanism to notify the originator but it generates more traffic. The advantage against an added traffic is the notification of the RERR to the one-hop neighbors of all the nodes along the path. This scheme works well in the situations where multiple nodes are forwarding data through the same link. This scheme helps analyzing the effect of notifying more source nodes by generating more traffic as compared to the data loss which would occur if we don't notify all the potential source nodes.

Routing table Aware Back-Propagation Algorithm

This technique tries to emulate the RERR delivery mechanism of AODV, without using the source address. It is developed to exploit the notion that if route error is propagated to all the potentially affected nodes then the future packet loss, which could occur by sending packets on failed link, can be saved. All the nodes that have the route entry for the destination, upon receiving this route error message, will delete the route entry for the destination and broadcast the message again towards the nodes closer to the originator. The algorithm is described in Fig. 4.

Performance Analysis

The main purpose of *RERR* mechanism is to increase overall system performance. The performance objectives include minimizing the *RERR* traffic, reducing link-failure discovery time to maintain a consistent topological view and avoiding the looping of stray packets. However, the provision of *RERR* delivery mechanism is not without a cost. The *RERR* traffic can be considered as the routing overhead which consumes the network bandwidth. In a highly dense network, where the link failure rate is very high, the *RERR* traffic can utilize a major portion of the network bandwidth-which is a highly constrained resource in LoWPAN.

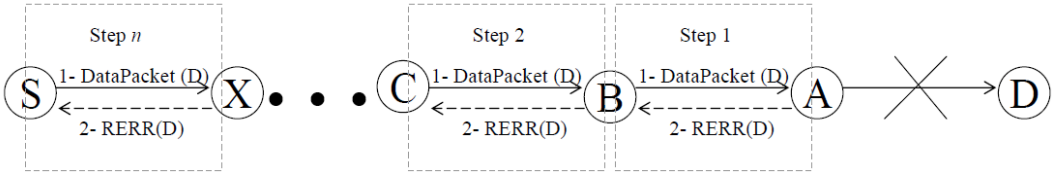

Fig. 1. The UBP mechanism: RERR delivery from node A to node S

Legend: *P(p,d) : Data packet P, received from previous hop node 'p' for the destination 'd'*
E(d,n) : Route entry for destination node 'd' , node 'n' is the next hop node
RERR(d): Route error message notifying a link failure for node 'd'

```
Begin Proc
    If a node receives P(p,d)
        If there is an E(d,n) in Routing Table
            Send  P(d) to n
            If a link failure was notified by MAC Layer
                //Initiate the RERR reporting procedure
                Unicast RERR(d) to p
                Discard P(d)
            End If
        Else //there is no E(d,n) in Routing Table
            Unicast RERR (d) to p
        End if
    End if
    If a node receives RERR(d)
        Discard RERR(d)
    End if
End Proc
```

Fig. 2. UBP algorithm

Legend: P (p,d) : Data packet P, received from previous hop node 'p' for the destination 'd'
E(d,n) : Route entry for destination node 'd' , node 'n' is the next hop node
RERR(d): Route error message notifying link failure for node 'd'

```
Begin Proc
    If a node receives P(p,d)
        If there is an E(d,n) in Routing Table
            Send  P(d) to n
            If a link failure was notified by MAC Layer
                //Initiate the RERR reporting procedure
                Broadcast RERR(d) in one hop
                Discard P(d)
            End If
        Else //there is no E(d,n)
            Broadcast RERR(d) in one hop
        End If
    End if
    If a node receives RERR(d)
        Delete E(d,k) from Routing Table
    End If
End Proc
```

Fig. 3. BBP algorithm

```
Legend:  P (p,d)      : Data packet P, received from the previous hop node 'p' for the destination 'd'
         E(d,n)       : Route entry for the destination node 'd' , node 'n' is the next hop node
         RERR(d,h) :Route error message notifying a link failure for the destination 'd' where 'h' is
                     the hop count from the node initiating the RERR to 'd'
         HC(n,d)   :Hop count from  node 'n' to the destination 'd'
Begin Proc
    If a node receives P(p,d)
        If there is an E(d,n) in Routing Table
            Send P(d) to n
            If a link failure was notified by the MAC Layer
                //Initiate the RERR reporting procedure
                Broadcast RERR(d,h) in one hop
                Discard P(d)
            Else //there is no entry E(d,n) in Routing Table
                Broadcast RERR(d,h) in one hop
            End if
        End if
    End if
    If a node receives RERR(d,h)
        If (there is an E(d,n) in Routing Table) AND ( HC(n,d) > h )
            Delete E(d,k) from RT
            Broadcast RERR(d,h) in one hop
        Else
            Discard RERR(d,h)
        End If
    End if
End Proc
```

Fig. 4. RTABP algorithm

The *RERR* delivery time is an essential parameter that plays a very significant role in the process. It could affect the data loss and system throughput, depending on the transmission rates and traffic patterns. If the *RERR* delivery time is high, the data loss will be higher, deteriorating the performance. We must also consider analyzing the throughput as an overall performance measure as a function of *RERR* delivery mechanism. This metric gives an overall view of the performance gain against the *RERR* provisioning algorithm. Taking all these points into consideration, we have chosen *RERR traffic*, *RERR delivery time* and *throughput* as performance metrics to compare the performance of our mechanism with AODV. The selection of AODV was not an automatic choice but it is made because it is essentially considered as a core routing protocol for ad-hoc networks. Moreover, most of the simplified versions also follow the core algorithms for AODV. Therefore, the AODV provides are general and common framework for performance of on-demand ad-hoc routing.

We analyze the performance of AODV and UBP for the best and the worst case scenarios. The best case scenario refers to the minimum hop counts between the originator and the link failure detecting node. Likewise, the worst case scenario could refer to the maximum hope counts between the originator node and link failure detecting node. Figure 5a shows the best case scenario, i.e., the node which detects the link failure is the upstream one-hop neighbor of the originator node. In this case the number of *RERR* packets, number of route-failures and the time to propagate the *RERR* is the minimum. For the worst case situation, as presented in Fig. 5b, the node which detects the node failure is the downstream one-hop neighbor of the destination node. Since it is the farthest possible node, from the originator on this route, the number of *RERR* packets, the route-error failures and the time to propagate the *RERR* reach to the possible maximum limit.

Following is the list of variables which are use for this analysis:

- x: Number of nodes in the network
- n: Average number of active links through a node in the network
- h: Average number of hops in an active path, i.e. average active path length in terms of hop count
- b: Rate of link failure for a node
- P_{Ti}: Probability that a node i will transmit data on a specific active link again
- D_{qi}: Average queuing delay experienced by a packet at node i
- D_t: Average transmission delay experienced at a node in the network
- A: The subscript used to denote AODV

U: The subscript used to denote UBP
m: The subscript used to represent the best case scenario calculations
M: The subscript used to represent the worst case scenario calculations

RERR Traffic

The number of *RERR*s initiated in the network, depends on the number of link failures in the network. Higher the link failure rate, higher is the number of *RERR*s initiated within the network. In case of AODV, whenever a *RERR* is initiated, the objective is to deliver it to every node which is currently using the broken link. Each node which receives a *RERR* message, forwards it to its pre-cursors. The overall traffic generated against each initiated *RERR* depends on the average pre-cursors for a node and average active path length. Figure 6 shows the effect of average path length and average active paths per node over the *RERR* traffic. Theoretically, the *RERR* traffic increases linearly, against average path length as well as against average active links per node. Higher the generated amount of traffic, more adversely it affects the network capacity.

In this section we analyze the *RERR* traffic generated by AODV and UBP for the best and worst case scenarios.

Best Case Scenario

For AODV, in the best case scenario there can be one or more sources sending the data through the node which detects the link failure. In case there is only one source, the *RERR* overhead per link failure is a unicast message to the source. In case there are multiple sources, all one hop neighbors, a single broadcast is the *RERR* overhead against each link failure. For the whole network of *x* nodes the *RERR* traffic for AODV can be calculated as:

$$O_{A,m} = b \times x \tag{1}$$

For UBP, if there is only one source, there is a single unicast *RERR* message. In case there are *n* sources, all one hop neighbors, UBP notifies the specific source only when the detector node receives a data packet from the source. In case there are *n* sources, the maximum traffic *RERR* traffic overhead shall be *n* unicast messages.

In 6LoWPANs, the links are generally established to notify a specific event to another node. The route once established is used once and is used sporadically. Therefore, in practical situations the generated *RERR* is much lower than the theoretical maximum in this case. To calculate the *RERR* overhead for UBP, we include the factor 'probability of transmission', P_{Ti}, in the equation. The *RERR* traffic for UBP, $O_{U,m}$ in case of *n* sources can be calculated as:

$$O_{U,m} = \sum_{i=1}^{n} P_{Ti}(1) \times b \times x \; where \; 1 \le P_{Ti} \le 0 \tag{2}$$

The probability of transmission depends on the type and rate of traffic generated by the originator node, e.g., in case of streaming traffic the P_T is higher and lower in case of event-based traffic. An established route may not be used again before the expiration of the route. The plausibility, that 6LoWPANs are generally used for event-driven data transmission, reduces the probability that a path shall be used continuously for a longer span of time. Therefore, P_{Ti}, is central to the *RERR* overhead traffic. The *RERR* traffic comparisons in Fig. 7 and 8 show the importance of P_T over the amount of traffic generated by UBP in case of link failures.

It is very clear that *RERR* traffic generated by UBP is always less than AODV except all the source nodes use the broken link again, i.e. P_{Ti} is 1 for all the sources. The average path length and average number of links per nodes have the same effect on AODV as well as on UBP.

The delivery of *RERR* to the source is dependent on the availability and quality of the link. The retransmission may also be needed in case of collision etc. We assume that the communication channel will remain available for the *RERR* delivery. Since the collisions occur in both the cases; i.e., in either AODV or UBP case, therefore, in Equation (2) we did not include the retransmission effect due to collision.

Worst Case Scenario

The worst case scenario can also be dealt with considering two situations; (a) when there is only one source using the broken link and (b) there are multiple source nodes. If there is only one source the maximum *RERR* traffic generated by AODV as well as UBP, O_M, is *h*-1messages. In case there are *n* sources, the maximum *RERR* traffic generated by AODV, *OA,M*, can be calculated as:

$$O_{A,M} = n \times (h-1) \times b \times x \tag{3}$$

It is important to know that in case of AODV, the *RERR* will be delivered to each active source, even if the source node may never use the established route again.

In case, there are *n* sources, the maximum *RERR* traffic generated by UBP, $O_{U,M}$ can be calculated as:

$$O_{U,M} = \sum_{i=1}^{n} P_{Ti} \times (h-1) \times b \times x \tag{4}$$

If the value for P_{Ti} for each of the sources is 1 in the above equation, the RERR traffic generated by UBP will approach to that of AODV. Generally, this is not the case, therefore, UBP generates less RERR traffic as compared to AODV.

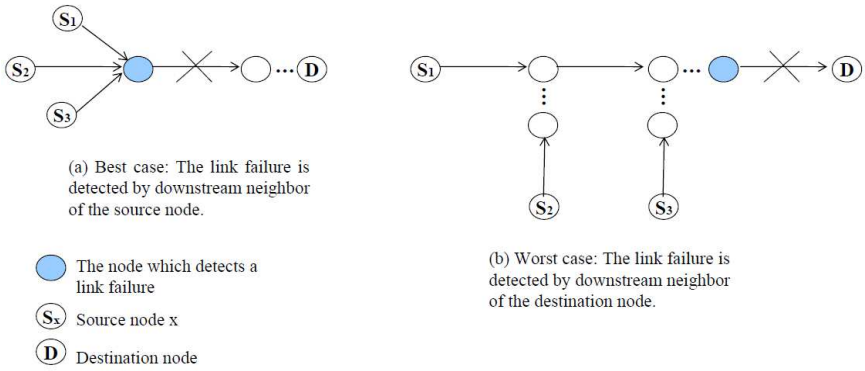

Fig. 5. Link failure scenarios

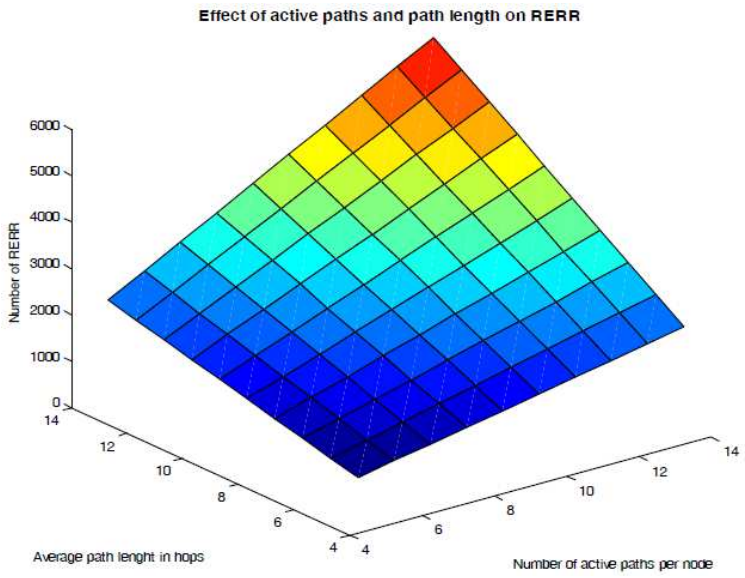

Fig. 6. Effect of path length and active paths per node on RERR traffic

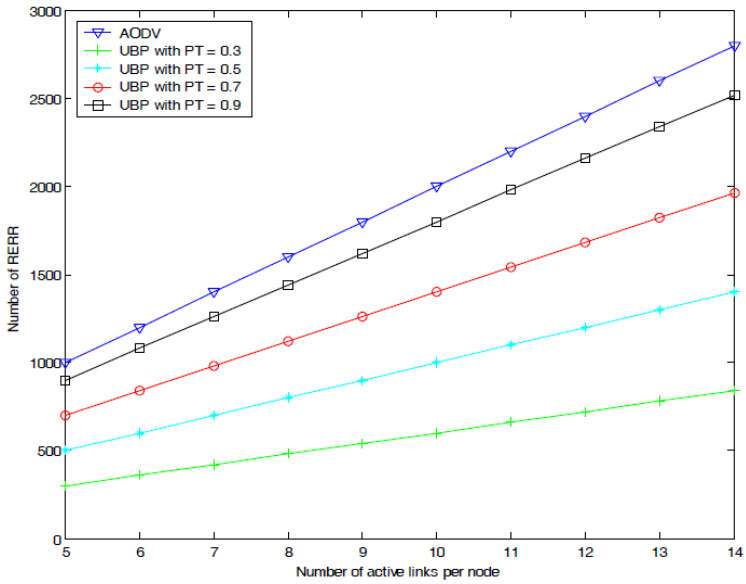

Fig. 7. RERR traffic comparison of AODV and UBP

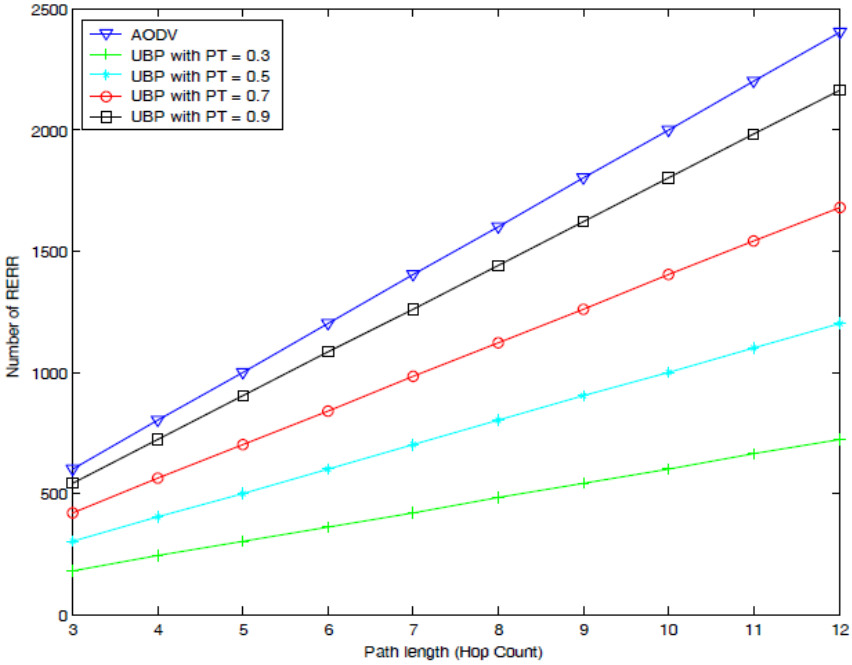

Fig. 8. Effect of average path length over RERR traffic

RERR Delivery Time

The *RERR* delivery time depends on the propagation delay, queuing delay and transmission delay for a specific path. The propagation delay can be ignored in the case of LoWPANs because these networks are assumed to be covering small areas, e.g., tens of meters. The total delay for each hop, therefore, can be given by $D_{ti} + D_{qi}$, where D_{ti} and D_{qi} denote transmission delay and queuing delay respectively for the node *i,* which receives and forwards the *RERR* message.

The messages in the queue are transmitted sequentially. We assume that the message generation process is Poisson and that the message length is exponentially distributed with average value *L.* The messages are generated at a rate λ (messages per second). In such case each multi-hop route is a tandem of queues and the whole network can also be viewed in similar fashion. As a result the M/M/1 queuing model can be applied to each individual node. The average delay that a *RERR* packet experiences from source to destination is obtained as the sum of the average delays experienced at each intermediate node.

Best Case Scenario

If *R* is the transmission rate for the node, the average queuing delay (including transmission delay) that a packet experiences at each node it traverses is given by:

$$D = \frac{1}{R/L - \lambda} + D_t \tag{5}$$

If there is only one source, in case of AODV, the time to propagate the *RERR* can be represented as:

$$T_{o,m} = D = \frac{1}{R/L - \lambda} + D_t \tag{6}$$

This equation also holds for any number of sources provided that all are single-hop neighbors of the node which initiates *RERR* message. The transmission type may vary between, unicast or broadcast based on the pre-cursor list.

The delay is same for UBP if there is only one source. However, in case of *n* single hop paths, the time to propagate the *RERR*, to all *n* sources is given by:

$$T_{U,m} = \sum_{i=1}^{n} P_{Ti}\left(\frac{1}{R/L - \lambda} + D_t\right) \tag{7}$$

Since all the sources are sharing the same medium, the *RERR* will be delivered sequentially. A unicast *RERR* message is sent to each originator individually. It seems like this is a slow process and takes more time to notify all the sources. Figure 9 shows how RERR delivery time increases with the increase in active number of links per node. It is very clear that for higher P_{Ti} in the network, UBP shows higher *RERR* delivery time as compared to AODV. On the contrary, for the networks where the P_{Ti} is low, the RERR delivery time of UBP is better or closer to that of AODV.

In real situations, an established route may not be used again before the expiration of the route. This fact

reduces the P_T for the source nodes, which implies that not all source nodes are needed to be notified. We contend that only active nodes should be notified, with a main purpose of reducing the *RERR* traffic overhead.

Worst Case Scenario

The delay involved in delivering the RERR message over an h hop path can thus be the sum of the average queuing delay experienced over each intermediate node. The route error delivery delay time for AODV thus can be given as:

$$T_{O,M} = \frac{1}{R/L - \lambda} \times (h-1) + D_t \times (h-1) \quad (8)$$

For the worst case scenario, the maximum time needed to propagate a *RERR* message, $T_{U,M}$ to a single source, over a path of length *h* hops, is given by:

$$T_{U,M} = \frac{1}{R/L - \lambda} \times (h-1) \quad (9)$$

If there are *n* sources using fully disjoint paths, $T_{U,M}$ can be calculated as:

$$T_{U,M} = \sum_{i=1}^{n} P_{Ti} \left[\left(\frac{1}{R/L - \lambda} \times (h-1) \right) + D_t (h-1) \right] \quad (10)$$

In practical topologies, this situation (where all the routes are disjoint) rarely occurs. According to the hop count optimality rule, various paths share optimal links-once a *RERR* is propagated along one link, it serves to notify all sources which are along this path-resultantly, speeding up the *RERR* delivery process and reducing the traffic overhead considerably.

Figure 10 depicts the effect of average path length over RERR delivery time. The RERR delivery time increases linearly against the increase in path length. The average RERR delivery time for UBP is lower than that of AODV if there are few active links per node. On the other hand, for higher number of active links per node, UBP exhibits higher RERR delivery time.

Packet Delivery Ratio and Goodput

The packet delivery ratio and network throughput depend upon the network topology, network traffic, the rate of link failures and the average recovery time. In case a link failure occurs the source node keeps sending the data until it receives a *RERR* message about the link break. After receiving the *RERR* message, the originator node finds the new route to the destination and starts sending the data again. The time from detection of link-failure till the time the originator is ready to send data again to the source is termed as recovery time. The recovery time largely depends upon the *RERR* delivery time. More time a protocol takes to deliver a *RERR*, the higher is the recovery time, which results into higher data loss and lower packet delivery ratio. In case of UDP where end-to-end acknowledgement mechanism is not available, the longer recovery time drastically deteriorates the packet delivery ratio and, therefore, the throughput.

Simulation Results

In this section, first we describe our simulation results, showing the comparison of UBP, AODV and under the case when no provision is provided for RERR delivery. Second, we compare the performance of UBP with its variant, the BBP scheme. As we shall see, the results confirm that the UBP reduces the RERR traffic overhead and increases the throughput of the system.

We have modified AODV to evaluate our schemes using Network Simulator-2 (ns-2). The simulation setup consists of 100 nodes, with a radio range of 15 m each, spread over an area of 117×117 m. Every simulation run is for 100 sec, with Constant Bit Rate (CBR) being the traffic type. Inter-packet transmission delay varies between 0.05 and 0.5 sec.

RERR Traffic Overhead

The simulation results, as in Fig. 11, show that UBP generates almost 40% less traffic as compared to AODV for higher data rates. It can be explained by the operation of UBP-it delivers the RERR only through the paths which are sending the data, contrary to AODV which notifies all the active sources, irrespective of their transmission probabilities. Since UBP notifies only single source at a time for one route failure, there is a possibility that some other nodes may have a stale entry about the broken link. This situation could increase the data loss. But, generally the routes are made for certain event notification and are not used continuously for a longer span of time which means that a route used once is rarely used again. This fact reduces the potential data loss.

Figure 12 shows the comparison between RERR traffic generated by UBP, BBP and RTABP. The other variants of UBP i.e. BBP and RTABP generate big amount of traffic as compared to UBP. The main reason is that these mechanisms use the broadcast. These mechanisms may inform the potential sources as a proactive measure but it means there shall be more RERR packets on the network. RTABP tries to inform all the sources using the routing table information to emulate AODV without using the source address which results into the highest RERR traffic overhead.

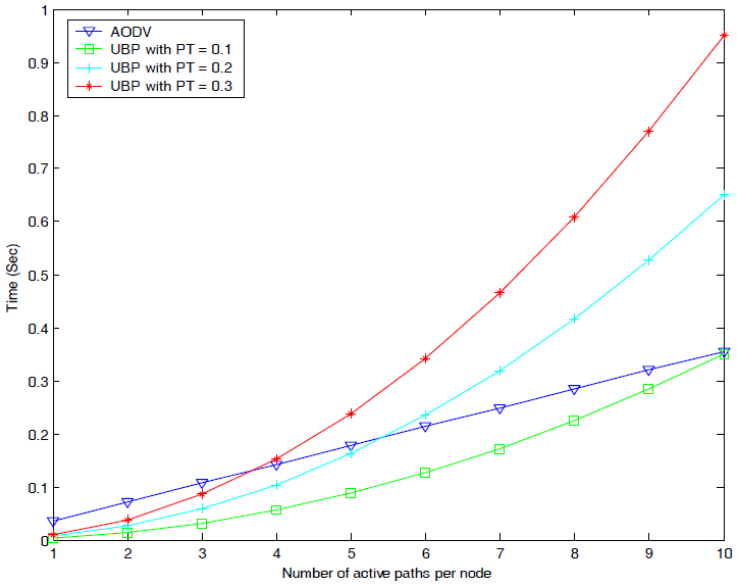

Fig. 9. Effect of active links per node on RERR delivery time in AODV and UBP

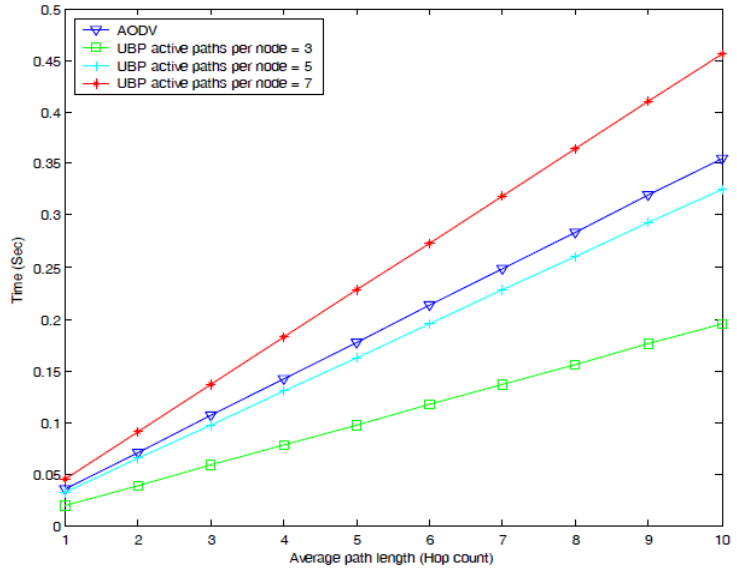

Fig. 10. RERR delivery time comparison of AODV and UBP

Packet Delivery Ratio

It is defined as the percentage of packets delivered to the destination over total number of packets transmitted. As Fig. 13 shows, the packet delivery ratio is very similar in UBP and AODV. Packet delivery ratio only seems to improve with the use of back propagation when compared to the case where no mechanism for propagating the route error message is present. This improvement can be explained by the fact that RERR message delivery to the originator stops further packet loss and thus improves the delivery ratio. The overall delivery ratio is also attributable to inherent characteristics of wireless media where data errors and link errors substantially affect the transmission.

When we compare UBP, BBP and RTABP, it is clear that packet delivery ratio is high for UBP. For higher data rates, a link failure adversely affects the delivery ratio. In case of a link failure, the source node(s) continue sending data until they receive a RERR message. It means that if the RERR delivery time remains the same, the data loss would be higher for high data rates. Moreover, there are more collisions and higher retransmissions for higher data rates which explain the low delivery ratio. Figure 14 shows that delivery ratio is low for higher data rates where as it improves for relatively lower data rate.

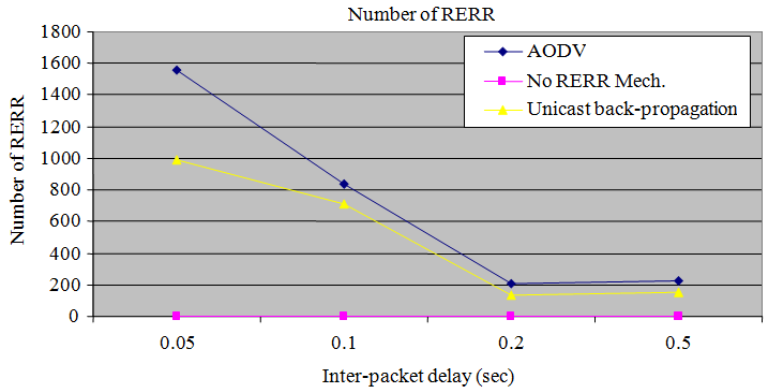

Fig. 11. RERR traffic overhead comparison of UBP and AODV

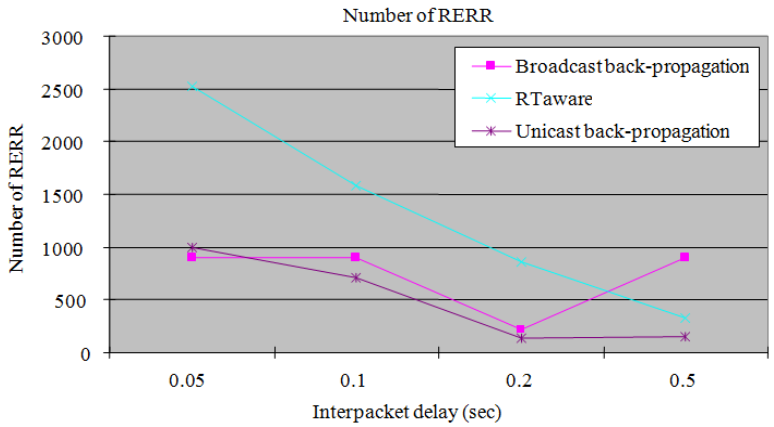

Fig. 12. RERR traffic overhead comparison

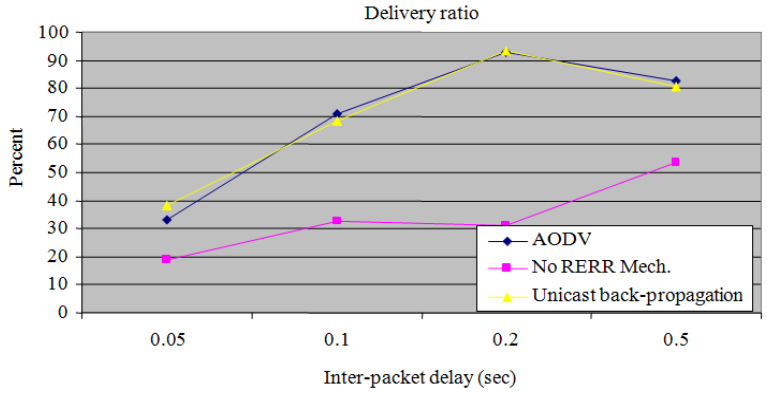

Fig. 13. Delivery ratio comparison of AODV and UBP

Goodput

UBP provides better goodput as the header can be further compressed; resulting into more bytes available for payload. Goodput can be defined as the payload bytes delivered from source to the destination in a given time. As Fig. 8 shows, the delivery ratio of UBP is similar to that of AODV, despite the fact that RERR is delivered without having originator's address. In case, no mechanism exists for RERR delivery, the delivery ration reduces to almost 50% of the case when UBP is used. Figure 15 confirms that UBP shows up to 10% better goodput than AODV. Figure 16 shows that the performance of BBP and RTABP is not better than UBP.

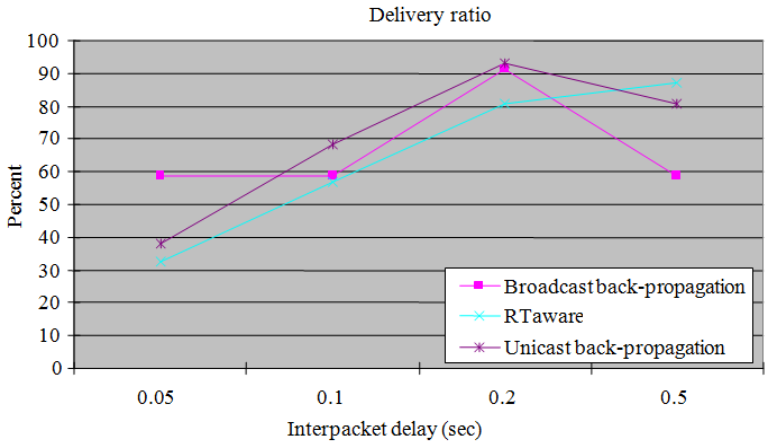

Fig. 14. Delivery ratio comparison between UBP, BBP and RTABP

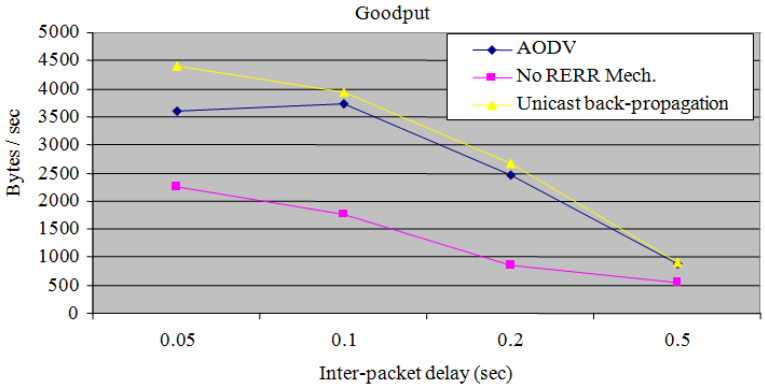

Fig. 15. Goodput comparison of UBP and AODV

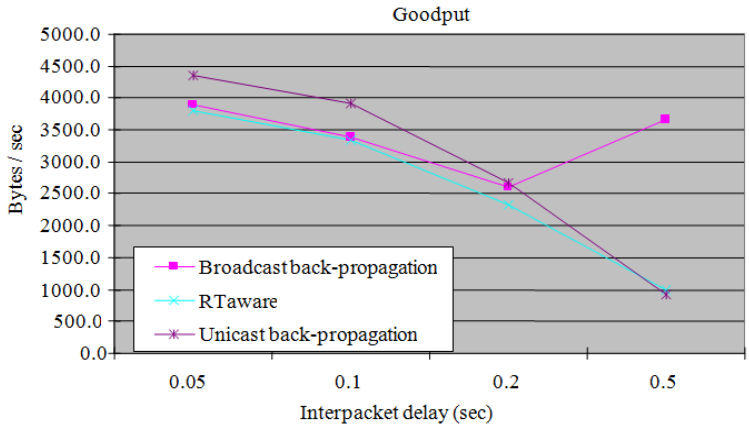

Fig. 16. Goodput comparison between UBP, BBP and RTABP

End-to-End Average Delay

The average end-to-end delivery time for UBP is higher than AODV for low data rates, but the performance improves for higher data rates. When the data rate is low, the propagation of RERR takes more time because each notification to the previous nodes is delayed, which increases the recovery time. This increase in recovery time increases the average delivery time. On the other hand, RERR delivery time is low for higher data rates, which improves the end to end delay. Figure 17 shows this phenomenon.

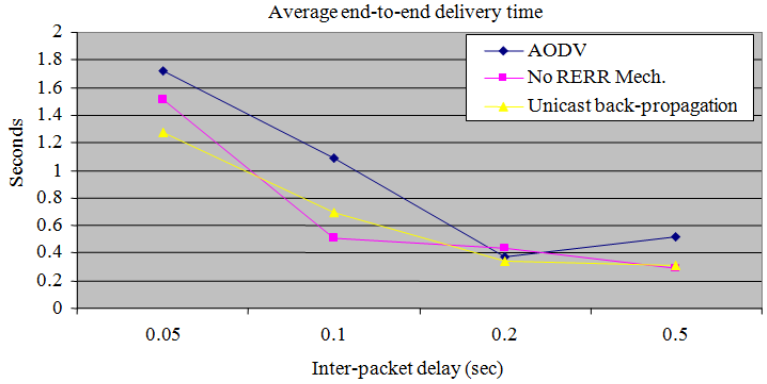

Fig. 17. E2E delivery time comparison of UBP and AODV

Conclusion

The transmission of IPv6 over LoWPAN has many challenges that are evolving into advantages including but not limited to the definition of new packet formats and header compression mechanisms. Our proposed mechanism (UBP) propagates RERR message without using originator's address. The results show that UBP mechanism generates less RERR overhead yet shows better diagnostic performance as AODV. UBP can be used to compress the adaptation layer header by a maximum of 64 bits-yielding approximately 6% more space in 6LoWAN link-layer frame.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

IEEE LoWPAN, 2003. Standard 802.15.4-2003. IEEE LoWPAN.

6LoWPAN Working Group, 2005. 6LoWPAN working group. 6LoWPAN.

Kushalnagar, N., G. Montenegro and C. Schumacher, 2007. IPv6 over LoWPANs: Overview, assumptions, problem statement and goals. RFC 4919.

Montenegro, G., N. Kushalnagar, J. Hui and D. Culler, 2007. Transmission of IPv6 packets over IEEE 802.15.4 Networks, RFC 4944.

Perkins, C.E. and E.M. Royer, 2000. The Ad Hoc On-Demand Distance Vector Protocol. In: Ad Hoc Networking, Perkins, C.E., (Ed.), Addison-Wesley, ISBN-10: 9780321579072. pp: 173-219.

Chakeres, I. and K.B. Luke, 2002. AODVjr, AODV simplified. ACM SIGMOBILE Mobile Comput. Commun. Rev., 6: 100-101. DOI: 10.1145/581291.581309

Salom, R.., P. Kaspar, T. Blecha, J. Freisleben and J. Bartovsky *et al*., 2012. Implementation of AODV routing protocol in sensor wireless networks. Proceedings of hte 20th Telecommunications Forum (TELFOR), Nov. 20-22, IEEE Xplore Press, Belgrade, pp: 194-197. DOI: 10.1109/TELFOR.2012.6419181

Kim, K., S.D. Park, G. Montenegro and S. Yoo, 2007. 6lowpan Ad Hoc On-demand Distance Vector Routing (LOAD), draft-danial-6lowpan-load-adhoc-routing-03. text. Ajou University.