IEEE Transactions on Industrial Informatics, published online on 17 May 2016
DOI: 10.1109/TII.2016.2569523

Efficient Route Update and Maintenance for Reliable Routing in Large-Scale Sensor Networks

Lapas Pradittasnee, Seyit Camtepe, *Member, IEEE,* and Yu-Chu Tian, *Member, IEEE*

***Abstract*—Reliable data transmissions are challenging in industrial wireless sensor networks (WSNs) as channel conditions change over time. Rapid changes in channel conditions require accurate estimation of the routing path performance and timely update of the routing information. However, this is not well fulfilled in existing routing approaches. Addressing this problem, this paper presents combined global and local update processes for efficient route update and maintenance and incorporates them with a hierarchical proactive routing framework. While the global process updates the routing path with a relatively long period, the local process with a shorter period checks potential routing path problems. A theoretical modelling is developed to describe the processes. Through simulations, the presented approach is shown to reduce end-to-end delay up to 30 times for large networks while improving packet reception ratio (PRR) in comparison with hierarchical and proactive routing protocols ROL/NDC, DSDV and DSDV with RPL's Trickle algorithm. Compared with reactive routing protocols AODV and AOMDV, it provides similar PRR while reducing end-to-end delay over 15 times.**

***Index Terms*—Wireless sensor network, routing protocol, hierarchical proactive routing, route update and maintenance, modelling**

I. Introduction

Reliable data transmission is one of the most important issues in industrial wireless sensor networks (WSNs) with fixed sensors. To maintain normal operations of an industrial plant, critical plant measurements such as temperature, vibration and pressure must be received in real-time so that prompt control actions can be taken to prevent any major disruptions [1]. Sensor nodes periodically report sensed data to the controllers, generating periodic data traffic on the network. However, harsh environments in industrial areas have a major impact on the reliability of data transmissions in WSNs. The quality of the network links may change from good to poor, temporarily or permanently, in a small period of time due to noises and interferences [2]. Such sudden changes in channel conditions cannot be remedied at the MAC layer with fixed or slow changing parameters [3], leading to a degradation of the communication performance with more packet drops and increased end-to-end delay. The performance degradation becomes severer when the system needs to wait for a long time to update the routing information. When this happens, only a small number of transmissions and retransmissions can be completed within their time limits [4].

To reduce packet dropout and end-to-end delay, a timely update of the routing information and an accurate estimation of the path quality are essential. They rely on the underlying routing metrics and routing protocol. Some routing metrics have been used to represent the reliability of data transmissions in WSNs, such as packet reception ratio (PRR) [5] and expected transmission count (ETX) [6]. However, if the underlying routing protocol is not designed appropriately, an accurate estimation of those metrics may require a long time. Such a long time will lead to deterioration of the data transmission performance. Unfortunately, existing proactive routing protocols are likely to experience a long route update process, especially in large-scale networks. The industrial WSN environments make this route update process even longer, further degrading the routing performance.

This paper employs a hierarchical proactive routing framework with a two-tier sensor architecture for large-scale industrial WSNs. In this hierarchical framework, the upper-tier nodes establish and maintain multiple routing paths between source-sink pairs, while lower-tier nodes maintain their connections with the upper-tier nodes. Then, to fulfil the requirement of reliable and timely data transmissions, this papers makes two main contributions: 1) two efficient route update and maintenance processes are designed which function on top of the hierarchical proactive routing framework: a global update process and a local update process each uses a different routing metric; and 2) a theoretical model is established to characterize the dynamics of the global and local update processes. With relatively long periods, the global update process evaluates, and updates if needed, routing paths between source-sink pairs. Using PRR as the routing metric, the global update process is conducted in all available routing paths in the upper-tier nodes. These routing paths are constructed based on the route discovery mechanism from Ad Hoc On-demand Multi-path Distance Vector (AOMDV) [7]. With shorter periods, the local update process detects potential problems on the communication links along routing paths. When a sudden change occurs in network performance, it informs the global update process for early path performance evaluation and route update. The local update process uses link quality as the routing metric.

Our routing approach is demonstrated through simulations. It achieves not only an accurate estimation of the routing

Manuscript received September 29, 2015. Accepted for publication May 7, 2016.

L. Pradittasnee, S. Camtepe and Y.-C. Tian are with the School of Electrical Engineering and Computer Science, Queensland University of Technology, GPO Box 2434, Brisbane QLD 4001, Australia. E-mail: lapas@it.kmitl.ac.th, {seyit.camtepe, y.tian}@qut.edu.au

L. Pradittasnee is also with the Faculty of Information Technology, King Mongkut's Institute of Technology, Ladkrabang, 1 Chalongkrung, Ladkrabang, Bangkok 10520, Thailand.

path performance but also a timely routing information update in response to local changes in channel conditions. More specifically, the approach provides better PRR and end-to-end delay than the most popular and comparable proactive routing protocols ROL/NDC, DSDV, and DSDV-Trickle. DSDV-Trickle is a modified DSDV with RPL's Trickle algorithm as its update process. Our approach also reduces the end-to-end delay while maintaining a similar level of PRR in comparison with the most popular reactive routing protocols AODV and its multiple path extension AOMDV. All those results show improved reliability and real-time performance. Furthermore, they are achieved with reduced routing overhead.

The paper is organized as follows. Notations used in the paper are listed in Table I. Section II reviews related work. Section III outlines the hierarchical proactive routing framework for WSNs. The new efficient route update and maintenance processes are presented in Section IV. To characterize the dynamics of the processes, a theoretical model is established in Section V. Section VI evaluates the performance of the presented approach. Finally, Section VII concludes the paper.

II. Related Work and Motivations

While routing is a broad topic, this paper improves routing of large-scale industrial WSNs mainly from two perspectives: route management, and routing metrics incorporating with link quality evaluation. The related work and motivations are discussed from these two perspectives.

A. Route Management

WSN routing protocols are either reactive or proactive. The reactive routing creates a routing path when the source node has a packet to transmit. It generally requires a long time to establish the routing path information. In comparison, the proactive routing creates routing paths at the beginning of the network operations. It then continually updates and maintains the routing information. Thus, it is able to transmit a packet quickly when the packet is ready. However, in proactive routing, each node must maintain a large number of routing paths to all possible destinations. It must also periodically transmit this large amount of routing information to all nodes in the network. Therefore, the implementation of proactive routing in large-scale networks significantly increases the overall routing overhead and resource consumption in each sensor node. As a result, it increases the reaction time when there is a change in network conditions [8], [9].

Fisheye state routing technique [10] offers a solution to the scalability problem in proactive routing. It is similar to the link state algorithm but provides a more effective route distribution process. As a result, each node in the network maintains recent routing information only and thus creates a small amount of routing overhead. Using hierarchical architecture is another well-known approach to solve the scalability problem in proactive routing [11]. Distributing the routing processes into multiple groups of sensor nodes, hierarchical proactive routing defines an effective method to transmit data packets between multiple clusters. Moreover, it can be tuned to meet the requirements of specific applications. For example, ROL/NDC adds the load-balancing technique and thus conserves energy in each sensor node. This helps prolong the network lifetime and reduce the overall end-to-end delay between the source and destination nodes.

IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) uses Trickle algorithm to address the routing overhead problem in proactive routing [12]. Trickle algorithm requires each sensor to transmit a route update packet at the end of each periodic update period U_p. The initial U_p value is set to $U_p = I_{min}$ when the sensor node activates the routing process. The value of U_p is adjusted according to whether or not a consistency or inconsistency is detected between its routing table and the routing information in a route update packet from its neighbour.

- Each time when the node detects the consistency in its routing information, it suppresses the route update transmission for the current U_p period, and doubles the length of the update period $U_p \leftarrow 2U_p$ until the value of U_p reaches its maximum threshold of 2 hours [12].
- Any time when the sensor node detects an inconsistency in its routing information, U_p is reset to $U_p = I_{min}$.

Then, the next route update packet is transmitted at the end of the new periodic update period U_p. The trickle algorithm works more effectively in dense networks. However, it reacts slowly when there is a change in the network. This is due to the fact that only at the end of each periodic update period U_p, can the node send the route update packet. As a result, a delay will likely occur in each of the nodes along a routing path. This causes notably large delays in large-scale networks.

LEACH [13] is a hierarchical routing protocol that forms clusters with two types of sensor nodes: cluster head and cluster member. Cluster heads are randomly selected for a specific period of time. Data is transmitted from cluster members to a cluster head. Then, it is aggregated and sent to the sink from the cluster head. The major drawback of LEACH is the requirement of direct communications between the cluster head and the sink. This creates a severe problem for large-scale sensor networks. Relaxing this constraint, TEEN [14] allows multi-hop transmissions between a cluster head and the sink. However, TEEN requires each node to know the locations of itself and all other nodes. In comparison, our approach in this paper focuses on providing reliable data transmissions without the use of position information.

SEP [15] is also a hierarchical routing protocol. But it has a distinct cluster head selection process. While both LEACH and TEEN assume homogeneous sensor nodes, SEP assumes heterogeneous sensor nodes in the network. Some nodes have better processing capability and larger battery power than others. SEP supports two levels of heterogeneous sensor nodes. Thus, the maximum number of hops from a cluster head to the sink node does not exceed 2 hops. DEEC [16] proposes a different cluster head selection process, which supports more than two levels of heterogeneous sensor nodes. Our paper in this paper does not try to establish hierarchical clusters. Instead, a core network is formed to establish multiple paths between local nodes and the sink.

In large-scale WSNs, the most popular and representative routing protocols include both proactive and reactive protocols.

TABLE I
ROUTE UPDATE AND MAINTENANCE PARAMETERS

Parameters	Description
C_p	Percentage of core nodes in the network: a parameter used to calculate P_c in the periodic core node selection process.
E_i	Initial energy level of node i: a parameter used to calculate P_c in the periodic core node selection process.
E_r	Remaining energy of a node: a parameter used to calculate P_c in the periodic core node selection process.
ETX	Expected Transmission Count: the number of expected transmissions for a packet so that it can be correctly received by the sink.
H_r	The number of received probe packets within the probe period T_l during the local update process.
H_s	The number of sent probe packets within the probe period T_l during the local update process.
I_{min}	Initial value of IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) Trickle algorithm periodic update period U_p.
k_b	The number of received packets through the backup routing path within the global update process periodic update period T_p.
k_m	The number of received packets only through the main routing path within the global update process periodic update period T_p.
k_r	Total number of received packets at the sink within the global update process periodic update period T_p.
$k_s = R \times T_p$	Total number of transmitted packets at the source within the global update process periodic update period T_p.
$l = H_r/H_s$	Local update process link quality metric, which is calculated within every probe period T_l.
l_t	Threshold for the local update process link quality metric $l = H_r/H_s$. If $l < l_t$, the link is considered to be poor.
M_p	The maximum number of packet drops on a routing path before PRR becomes less than PRR_t in a single update period T_p.
N_c	The number of times that a node has become a core node in the previous core node selection process periods T_r.
N_s	Expected periodic update period that the event $PRR < PRR_t$ will likely occur on a routing path for two consecutive periods.
P_c	Probability that a node becomes a core node in the core node selection process.
P_{ei}	Probability that unreliable routing path event $PRR < PRR_t$ occurs within the i^{th} and $(i+1)^{th}$ periodic update periods.
P_l	Probability that unreliable routing path event $PRR < PRR_t$ occurs within a periodic update period T_p.
P_n	Probability that an unreliable routing path event $PRR < PRR_t$ is not detected.
P_t	Core node selection process period. A node becomes a core node if $P_c > P_t$.
$P_u = 1 - P_l$	Probability that the routing path is reliable ($PRR \geq PRR_t$) within a periodic update period T_p.
$PRR = k_r/k_s$	Packet Reception Ratio: the ratio of the number of received packets at the sink to the number of transmitted packets by the source.
PRR_{avg}	Average Packet Reception Ratio for the whole network operation over the total operation time T_t.
PRR_m	Packet Reception Ratio of the main routing path where packets may be retransmitted due to dropped acknowledgements (ACK).
PRR_t	Threshold Packet Reception Ratio. $PRR < PRR_t$ means that the routing path is unreliable.
PRR_o	PRR when $PRR < PRR_t$ on the main routing path is detected; so that, the backup routing path will be activated.
PRR_p	PRR when the event $PRR < PRR_t$ on the main routing path is not detected.
Q_p	Probability that an acknowledgement (ACK) will not be dropped on the main routing path.
R	Rate (packets/sec) at which a source node generates and send packets to a sink over the active routing path(s).
R_e	Error rate ($0 \leq R_e \leq 1$) set on a communication line of a routing path to simulate unreliable routing path.
R_m	The maximum number of retransmissions for packet due to dropped acknowledgements (ACK) on an unreliable routing path.
$T_b = T_t - T_m$	Total amount of time that the backup routing path is used (activated) for source to sink communication.
T_d	Estimated time that an unreliable routing path event $PRR < PRR_t$ on the active routing path will be detected.
T_e	Probe period at which each node on a routing path transmits a probe packet to its neighbours in the local update process.
$T_m = T_d + T_p$.	Total amount of time that only the main routing path is used (activated) for source to sink communication.
T_l	Update period of the local update process. At the end of T_l, each node on a routing path transmits probe packets to its neighbours.
T_p	Update period of the global route update process. At the end of T_p, each sink evaluates the performance of the active routing path.
T_{p0}	Default value for T_p.
T_{pa}	The value of T_p when there is an alert due to the local update process is received.
T_r	Period for the core node selection process, which is initiated at every period T_r.
T_t	Total operation time of the network.
U_p	Update period in IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) Trickle algorithm.

Examples of popular proactive routing protocols are DSDV, DSDV-Trickle and ROL/NDC protocols. AODV and AOMDV protocols are examples of popular reactive routing protocols. Those proactive and reactive routing protocols are comparable with the routing approach presented in this paper in terms of the basic ideas and techniques used in the protocols. Hence, they will be evaluated as benchmark protocols in this paper.

B. Routing Metrics and Link Quality Evaluation

There are two essential requirements for reliable real-time data communications in large-scale WSNs. The first requirement is an accurate performance estimation of the routing path. An appropriate selection of a routing metric determines how well and how complicated the routing performance is characterized for the specific WSN application. The second requirement is a timely update of the routing information according to the routing performance estimation. This is one of the major problems in existing proactive routing and will be further discussed later.

Selecting a routing metric based on specific application requirements is important. It helps establish the best routing path for packet transmissions. Using a single metric is a well-accepted method [6]. Some routing metrics were originally developed for general wired networks, such as hop count and bandwidth. Some other routing metrics, e.g., residual energy and link quality, were designed specifically for WSNs.

The success rate of data transmissions is popularly used to characterize the reliability of WSNs. This is because most industrial applications are time-sensitive and generally require each data packet to be received before its deadline [1], [17]. Delayed packets may result in situations where emergency events are missed out, causing a critical system to malfunction or even fail. A viable approach is to select a path with high success rates, which can provide an acceptable level of reliability even at the cost of increased energy consumption [18]. Such a path will experience fewer retransmissions, implying a smaller end-to-end delay. Typical routing metrics in this category include PRR [5] and ETX [6].

ETX specifies the expected number of transmissions for a packet so that it can be correctly received by the sink. It fully captures the cost of transmission, link reliability, and traffic load on the network [6]. However, to estimate the ETX value for the whole routing path, all ETX values for each of the links on a routing path are required. Therefore, the route update process must provide a method to collect all those ETX values. This increases both routing overhead and processing time, and violates our requirements of small overhead and end-to-end delay for large-scale industrial WSNs.

PRR uses the ratio of the number of received packets at the sink node to the number of transmitted packets from the source node. Given the rate R at which a source node generates packets, PRR for a routing path can be estimated in two ways. First, PRR estimates from each communication link of the routing path can be used to evaluate the PRR value of the routing path. Second, PRR of the routing path can be locally estimated at the sink node without the need of additional information from other nodes. The latter PRR estimation technique has a notable advantage for overhead reduction and end-to-end delay suppression in WSN routing. In addition, data transmissions in the forward direction from the source node to the sink is significantly more important than in the reverse direction in industrial WSNs, particularly for critical measurement data delivery and event-driven real-time control. This is favourable for reliable data transmissions considered in this paper for large-scale industrial WSNs.

However, PRR alone does not provide as reliable data transmission as ETX can do. Because, PRR estimation mechanism considers only the successfully received data packets at the sink. The data packet that is successfully received at the sink may experience a couple of transmission failures when it is forwarded through each link in the routing path. Generally, the MAC layer, such as CSMA/CA, retransmits a packet up to the maximum threshold before it gives up. PRR cannot represent this type of transmission failure. As a result, PRR alone does not fully capture the cost of transmission, link reliability, and network traffic load. Without reliability of data transmissions, measuring timeliness of data transmissions using PRR will become largely devalued in real-time applications.

It is our expectation to develop an ETX-like but lighter and quicker routing mechanism while avoiding ETX's disadvantages. A single routing metric in a conventional route update process has not been found to fulfil this requirement. This motivates our research in this paper to develop two route update processes incorporating with two routing metrics: a global update process with the PRR metric, and a local update process with the link quality metric. Similar to ETX, the link quality metric in the local update process collects global information of the link for its evaluation. This complements the PRR metric in global update process. The two update processes incorporating with their respective routing metrics provide an effective solution for reliable data transmissions with light routing overhead and small end-to-end delay.

A combination of routing metrics is also investigated in industrial WSNs [19]. This results from the fact that the requirements of recent applications have become more complicated. For example, energy consumption and residual energy are combined in [20]. This combination tends to weight energy consumption more heavily than residual energy at the beginning of network operations when all routing paths still have a high level of battery power. More weights are allocated to the residual energy when the residual energy of the routing path falls below a threshold. Such combinations of metrics effectively adapt to the latest status of the network. However, they require a longer time to calculate and thus may not always give good network performance. Furthermore, existing routing protocols with combinations of routing metrics use the same time period to evaluate all these metrics. But those metrics do not have the same sensitivity to the changing environment. This leads to an inaccurate estimation of the routing performance with a long evaluation period, or increased overhead with a short evaluation period.

A long network response time is one of the major problems of proactive routing in large-scale WSNs [21]. Existing proactive routing protocols are mainly based on a simple periodic update process. While being simple, the process requires each node to transmit a route update packet to its neighbours periodically. A large update period is preferable for energy savings and overhead reduction. But it causes a significant delay in response to changes in routing path conditions because a new update will not happen until the current period expires. Reducing the period helps shorten the network response time, but leads to a notable increase in the routing overhead and consequently worsens the overall network performance. This motivates the research of this paper for efficient route update and maintenance processes incorporated with a hierarchical proactive routing framework. A mathematical model is also established to estimate the performance of the two processes. It helps determine the parameters and settings of the processes.

III. A Hierarchical Proactive Routing Framework

Fig. 1 illustrates the hierarchical proactive routing framework for multi-path routing in large-scale industrial WSNs. It has a two-tier structure. The nodes in the upper-tier and lower-tier are called core nodes and local nodes, respectively. The core sensor nodes are responsible for establishing routing paths, estimating the routing performance, updating the routing information, and maintaining multiple routing paths from source nodes to sink nodes. The framework will try to limit the number of core sensor nodes for reliable, timely and lightweight routing.

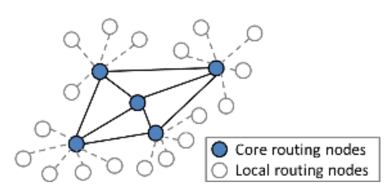

Fig. 1. The hierarchical proactive routing framework.

The core node selection process is initiated at every period T_r. It is adapted from the cluster head selection process in HEED [22]. As the core nodes must establish and maintain multiple routing paths, they likely consume more energy than the local nodes. Hence, the selection process considers the energy level as well as the number of times that the candidate node has previously been selected as the core node. At the beginning of each period T_r, each node calculates the probability that it becomes a core node P_c as:

$$P_c = C_p \times E_r/(N_c \times E_i), \quad (1)$$

where C_p is the percentage of core nodes in the network, N_c is the number of times that this node has become a core node, E_r is the remaining energy of the node and E_i is the initial level of energy of each node. Only the nodes with a value of P_c higher than the threshold value P_t can become the core nodes. To ensure a graceful degradation, P_t can be set initially to a high value and be reduced after each T_r period.

After the core nodes are selected, the remaining nodes become local nodes. The local nodes are responsible for establishment and maintenance of their connections to a closest core node. Each local node broadcasts a control packet to discover core nodes in the nearby area. Then, it establishes a connection with the core node with the highest level of signal strength among the core nodes that responds first. When the connection between the core node and the local node becomes poor for a significant period of time, the local node terminates the current connection. Then, it broadcasts a control packet to request a new connection with other core nodes in the area.

The core sensor nodes are responsible for establishing and maintaining multiple node-disjoint paths for each source-sink pair. The path establishment uses the same process as in Ad Hoc On-demand Multi-path Distance Vector (AOMDV) [7]. In AOMDV, the source node sends route-request to the sink. On receiving the route-request, the sink responds with multiple route-replies. From those route-replies, hop-by-hop reverse sink-source paths are established. They are further used to establish the source-sink routing path. In our hierarchical proactive routing, each of the source-sink pairs establishes two disjoint paths: a main path and a backup path. In the case of a failure or performance drop on certain local links, the source-sink pairs are able to find an unaffected path with a high probability. Once the main and backup paths are established, they are stored, maintained and updated by the core nodes. If an existing routing path is terminated due to continuously unacceptable performance, the same route establishment process is used again to establish a new path.

This paper considers sensor deployment scenarios where for every source-sink node pair it is possible to find at least two node-disjoint paths. However, if some links fails permanently, alternative disjoint paths may become absent. In this case, source-sink pairs may experience high packet dropout and large end-to-end delays. In the worst scenario, they can even get disconnected. Possible mitigation strategies to fix such a problem include removal of noise or interference sources, deployment of new nodes, or physical layer reconfiguration such as selectively increasing the signal levels to establish new links. However, this is beyond the scope of this paper.

The route update and maintenance processes presented in this paper are designed to replace the simple periodic update process widely adopted in proactive routing for WSNs. The simple periodic update process generates a heavy routing overhead in large-scale WSNs. It also experiences a long delay in response to changes in network conditions. In this paper, the new global and local update processes are designed for efficient route update and maintenance in large-scale industrial WSNs. They are implemented in the core routing level of the hierarchical proactive routing. Particularly, in our routing approach, those two update processes are incorporated with multiple routing metrics. PRR and link quality are used in the global and local update process, respectively.

With a relatively long update period, the global update process updates the routing path from the source node to the sink. It executes periodically to determine whether or not the path provides an acceptable level of routing performance. If the level of the performance of the routing path becomes lower than a specific threshold, the current routing path is considered not to be able to deliver data packets within the specific requirements of the application. In this case, the backup routing path gets activated and used along with the current routing path. If the problem disappears within the next period, the backup path is deactivated. Otherwise, if the problem continues within the next period, the faulty routing path gets deactivated and the backup path becomes the new main routing path. As mentioned previously, the route discovery mechanism from AOMDV is adopted in our approach to establish a new backup routing path. If the failure or performance drop affects the new main routing path as well, it gets removed and replaced in a similar way.

In the global update process, PRR is used as the routing metric. It is locally estimated at the sink as follows:

$$PRR = k_r/k_s, \quad (2)$$

where k_r is the number of received packets at the sink, $k_s = R \times T_p$ is the number of transmitted packets by the source node, R is the data rate of the source node, and T_p is the update period. The value of k_r is available locally at the sink. When periodic data traffic generation is considered, which is one of the main types of data traffic in industrial applications [4], [23], the values of R and T_p are also known to the sink.

With a shorter update period, the local update process detects potential problems caused by changes in the channel conditions along the routing path. The routing information from this process is updated more regularly than the global update process. It is not used to determine the routing performance for the global update process. Instead, it is used to notify the sink node to set a new global update period in response to a sudden change in the network conditions. If the local update process detects a link quality degradation, it may cause the global update process to start earlier than usual. It is worth mentioning that a temporary degradation in the link quality may not always cause a performance deterioration sufficient enough to activate the backup routing path. This is due to the fact that the routing performance evaluation at the sink uses local statistics collected over the entire period of the global periodic update process.

The routing metric for the local update process is the link quality l. To enable local update, each node in the routing path transmits a probe packet to its neighbours in every probe period T_e. At the end of each local update period $T_l > T_e$, each node receives H_r probe packets out of total H_s transmitted probe packets. Then, the link quality l is defined as:

$$l = H_r/H_s. \tag{3}$$

Both the global and local update processes incorporating PRR and link quality metrics, respectively, provide an ETX-like but lighter and quicker routing approach. In the global update process, PRR can be evaluated using the local information available at the sink. This is different from ETX, which requires global routing information. Thus, PRR is quicker than ETX in the metric evaluation. Moreover, as the period of the periodic global updates is long, PRR is also lighter than ETX. In the local update process, the way of collecting global routing information for link quality evaluation is similar to that for ETX calculation. Therefore, the link quality metric in the local update process complements the PRR metric used in the global update process.

IV. Efficient Route Update and Maintenance

Maintaining a routing path that can provide a high success rate of data transmissions is one of the most important requirements for reliable data transmissions in industrial WSNs. If the routing path in use experiences a high number of packet drops, it is unlikely that the data packets will be delivered to the sink node on time. This demands quick identification of the poor performance of the routing path. Once the routing path is confirmed to be poorly performing, it should be replaced with an alternative one. The identification and replacement of the poor routing path require appropriate routing metrics and route update processes. For the update processes, a long period for routing metric evaluation will lead to a notable increase in the network response time. On the other hand, reducing the evaluation period will reduce the network response time but it will also introduce more overhead.

This conflicting issue of period selection is solved through designing two route update processes each with a different period: a global update process and a local update process. The two update processes are implemented in the core routing level of the hierarchical proactive routing framework. They work together to maintain multiple routing paths to the sink nodes: a main path and a backup path (Fig. 2). The main path is used to transmit all data packets if no routing problems are detected. If the performance of the main path falls down to an unaccepted level, then the backup path becomes active as well. If the problem on the main path continues, then the current main path is deactivated and the backup path becomes the new main path. If the new main path suffers from the same problem, it gets removed and replaced in a similar way.

The *global route update process* uses PRR as the routing metric. PRR is estimated locally at the sink with no need of additional control packets from other nodes. The sink sends out only one type of control packet: the periodic global update packet. It transmits this control packet to the source node after completing the PRR computation at the end of each T_p period as shown in Fig. 2. The control packet is transmitted through the main path if the path has an acceptable performance or through both the main and backup paths otherwise.

The *local route update process* uses probe packets (transmitted at every T_e period) for estimating the link quality l at the end of each T_l period. To control the overhead, the size of the probe packets is designed to be much smaller than that of the data packets. The header of the probe packet only includes essential routing information, e.g., the address of the sender and the identification of packet types.

The *combined global and local route update processes* enhances the reliability of real-time data transmissions. While the global update provides an accurate estimation of the routing path performance with a relatively long period T_p, it relies on the local update process with a relatively small period T_l to detect any sudden changes in the network conditions. Both route update processes work together to provide an ETX-like but quicker and lighter routing approach.

In the global update process, PRR is estimated in each T_p period. The routing path is considered to be reliable only when the estimated $PRR \geq PRR_t$, where PRR_t is a threshold. If $PRR < PRR_t$, the main path exhibits an unacceptable level of performance. In this case, the backup path will be activated in the next T_p period while the main path is still active. Both paths are used in the next T_p period to ensure acceptable performance of data transmissions. If the main path continues to show $PRR < PRR_t$ for two consecutive T_p periods, then it is terminated and the backup path is promoted to become the main path. In this case, the core routing based on AOMDV calculates a new backup path. The overall process is illustrated in Fig. 3-A.

With a relatively large period T_p, a deterioration of the routing performance of the main path due to changes in network conditions may not be captured in real-time in the global update process. As a result, before the next update, the main path with poor performance is still considered to be well performed though it is actually poor. This affects the PPR and end-to-end delay performance of the WSNs significantly.

This problem is overcome by using the local update process, which detects sudden changes in network conditions. As shown in Fig. 3-B, every node in the main path evaluates the link quality l periodically with period T_l, $T_l < T_p$. If $l < l_t$, where l_t is a threshold, the link is considered to be poor. The node that has been detected with $l < l_t$ notifies the sink and reports the measured link quality with an alert packet (Fig. 4).

Maintenance of reliable data transmissions is achieved through estimating global update period for timely detection and mitigation of poor path performance. The global update period T_p is initially set to be T_{p0} by considering both overhead and performance estimation accuracy [9]. After the sink node receives the alert packet as shown in Fig. 4, a shorter global update period T_{pa} replaces the current period T_{p0}. The shorter global update period T_{pa} aims to update PRR as soon

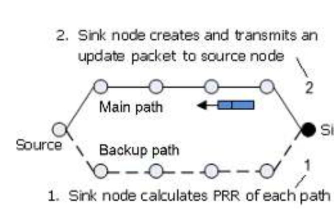

Fig. 2. The route update mechanism of the global update process.

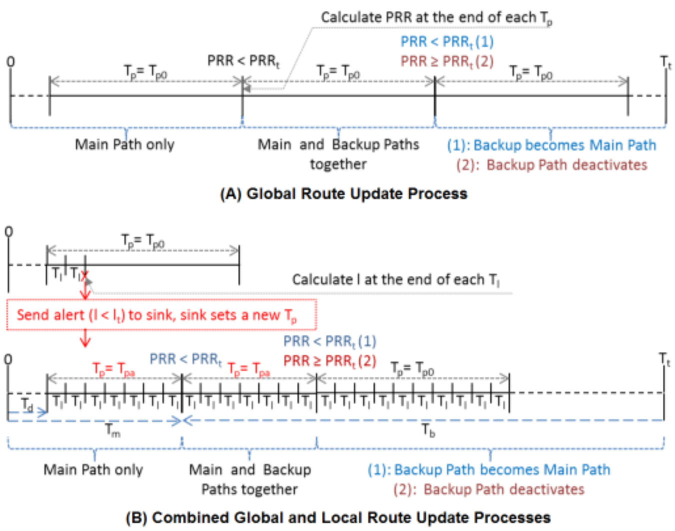

Fig. 3. (A) Timing of the global route update process; and (B) timing of the combined global and local route update processes. T_t: the total network operation time; T_p: global update period; T_l: local update period; T_d: time until the global update period when the sink receives an alert message; $T_m = T_d + T_p$: the operation time when only the main path is used; and $T_b = T_t - T_m$: the operation time of the backup path.

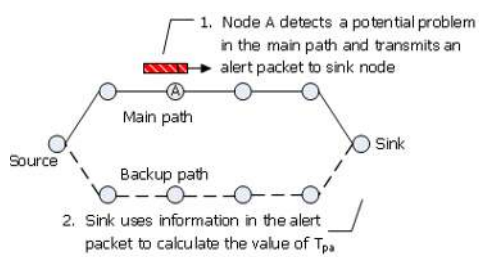

Fig. 4. Alert packet in the local update process.

as the main path is considered to be poor with $PRR < PRR_t$.

$PRR < PRR_t$ occurs when packet drops in the routing path are higher than the maximum number M_p of allowable packet drops in a single T_p period. A lower M_p means a higher PRR_t, which makes the main routing paths less immune to link quality drops. The relation between M_p and PRR_t is:

$$M_p = (1 - PRR_t) \times R \times T_{p0}. \tag{4}$$

With the value of M_p and the smallest reported link quality l, T_{pa} is determined as:

$$T_{pa} = \frac{M_p}{(1 - l) \times R}. \tag{5}$$

After T_{pa} is derived, a new T_p value is set as:

$$T_p = \begin{cases} T_{p0}, & \text{if } T_{pa} \geq T_{p0} \\ T_{pa}, & \text{else} \end{cases} \tag{6}$$

At the end of the new T_p period, the global update process estimates the new value of PRR. If $PRR < PRR_t$, the backup path is activated and $T_p = T_{pa}$ is set as the next global update period. Otherwise, $T_p = T_{p0}$ is set, and only the main path remains active.

V. Theoretical Modelling

This section provides a theoretical modelling of the dynamic behaviours of the efficient route update and maintenance processes presented in this paper. Firstly, the operation time T_m of the main routing path is derived. Then, the operation time T_b of the backup routing path is estimated. Finally, the overall performance of the routing framework is evaluated in terms of average PRR (PRR_{avg}) for the whole network operation period T_t.

A. The Operation Time Variables T_m and T_b

T_m represents the operation time when only the main path is used. Once T_m is estimated, T_b, the operation time of the backup path, can be calculated as $T_b = T_t - T_m$ (Fig. 3). The main path is used until $PRR < PRR_t$ for two consecutive T_p periods after which the main path deactivates and the backup path becomes the main path. $PRR < PRR_t$ occurs when packet drops at the sink are higher than the threshold M_p, which has been estimated in Eq. (4). Hence, the probability P_l that a T_p period will experience $PRR < PRR_t$ can be estimated as $P_l = 1 - P_u$, where P_u is the probability that the current T_p period has $PRR \geq PRR_t$. Using the PRR_p, which is the PRR when the event $PRR < PRR_t$ on the main routing path is not detected, we have:

$$P_u = \sum_{k=0}^{M_p} \binom{R \times T_p}{k} (1 - PRR_p)^k \times PRR_p^{R \times T_p - k}. \tag{7}$$

The probability P_{ei} that two consecutive T_p periods with $PRR < PRR_t$ starts at i^{th} period can be calculated iteratively. Consider the case study in Figs. 5 and 6. The settings are: five T_p periods in a T_t period, 10 transmitted data packets in total at the source node in each T_p period, and threshold $PRR_t = 0.8$. Let '0' represent the event $PRR \geq PRR_t$ and '1' the event $PRR < PRR_t$. The upper part of Fig. 5

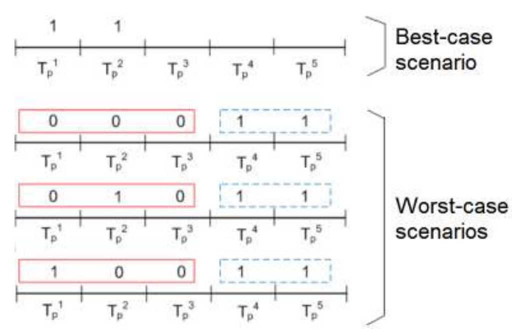

Fig. 5. The best- and worst-case scenarios.

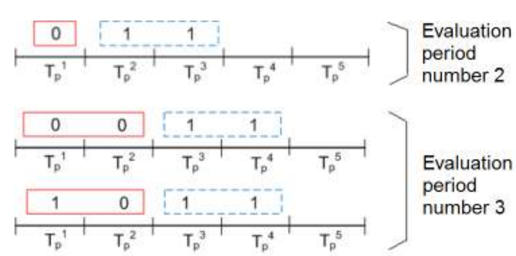

Fig. 6. Evaluation of routing path performance in the second and third periods.

shows the best-case scenario in which the source node can detect the event $PRR < PRR_t$ for two consecutive periods. Assume that the event $PRR < PRR_t$ is detected in the first T_p period. The probability P_{e1} that two consecutive T_p periods with $PRR < PRR_t$ start at the first T_p period is:

$$P_{e1} = P_l^2. \tag{8}$$

Next, the probabilities P_{e2} and P_{e3} that two consecutive T_p periods with $PRR < PRR_t$ start at the second and third T_p periods, respectively, are shown in Fig. 6. They are:

$$P_{e2} = P_u P_l^2,\ P_{e3} = P_u^2 P_l^2 + P_u P_l^3. \tag{9}$$

Finally, the worst-case scenarios are shown in the lower part of Fig. 5. There are multiple possible outcomes but all cases will detect the event that two consecutive T_p periods with $PRR < PRR_t$ start at the forth T_p period. It follows that:

$$P_{e4} = P_u^3 P_l^2 + 2P_u^2 P_l^3. \tag{10}$$

The case study can be generalized for any i^{th} interval by observing that scenarios for P_{e3} can be obtained by appending a '0' event to the beginning of P_{e2} scenarios and by appending a '1' followed by a '0' event to the beginning of P_{e1} scenarios. This is done to generate all possible event combinations except '11' events appearing before the third period. The same applies for P_{e2} and P_{e4} scenarios. In general, P_{ei} for the cases with an arbitrary number of periods can be calculated iteratively as:

$$P_{ei} = P_u P_{e(i-1)} + P_l P_u P_{e(i-2)},\ P_{e0} = 0,\ P_{e1} = P_l^2. \tag{11}$$

Assume that the number of T_p periods in a T_t period is t. Then, the expected update period N_s that $PRR < PRR_t$ will likely occur in two consecutive periods is estimated as:

$$N_s = \textstyle\sum_{i=1}^{t-1} (i \times P_{ei}) / \sum_{i=1}^{t-1} P_{ei}. \tag{12}$$

With the values of T_p and N_s, the time T_d that the source node will detect the event $PRR < PRR_t$ is estimated as:

$$T_d = (N_s - 1) \times T_p. \tag{13}$$

In the next global update period after T_d is expired, the local update process in the main path will detect the unreliable link condition and transmit an alert packet to the sink node (Fig. 4). After the sink node receives the alert packet, it calculates T_{pa} from Eq. (5) and then selects the new value for T_p based on Eq. (6). With this new T_p value, it follows that:

$$T_m = T_p + T_d. \tag{14}$$

After T_m is determined, T_b is calculated from Fig. 3 as:

$$T_b = T_t - T_m. \tag{15}$$

B. The Overall PRR Performance (PRR_{avg})

The overall PRR_{avg} of the route update and maintenance processes is calculated in a similar way to Eq. (2) but with consideration of failure to detect the event $PRR < PRR_t$ [24]. Let PRR_o and PRR_p represent PRR when the event $PRR < PRR_t$ is detected and undetected, respectively. Detection of the event $PRR < PRR_t$ means the backup path will be activated; otherwise keep using the current main path even if the path may become unreliable. Assuming the number of T_p periods in a T_t period is t, the probability P_n that the event $PRR < PRR_t$ is not detected is:

$$P_n = 1 - \textstyle\sum_{i=1}^{t-1} P_{ei}. \tag{16}$$

Then, the overall PRR_{avg} is estimated as:

$$PRR_{avg} = PRR_o \times (1 - P_n) + PRR_p \times P_n. \tag{17}$$

Now, let us estimate PRR_o and PRR_p. The same data transmission model as in [24] based on IEEE 802.15.4 is used in this paper for demonstration. Each data transmission process is completed when the source node transmits a data packet to the sink and then receives an ACK back. If the source node does not receive an ACK within a time limit, it retransmits the same packet until the maximum number of retransmissions (R_m) is reached. Let PRR_m denote the PRR of the routing path based on this data transmission model:

$$PRR_m = 1 - (1 - Q_p)^{R_m + 1}. \tag{18}$$

Thus, the value of PRR_p is estimated based on the number of packets received only through the main path by additionally considering the packet retransmissions:

$$PRR_p = PRR_m \times R \times T_d / k_s. \tag{19}$$

When $PRR < PRR_t$ is detected, both main and backup paths stay active and packets are transmitted through both paths. Moreover, packets that are transmitted through the main path may experience retransmissions. Hence, the value of PRR_o is estimated based on the numbers of packets received through both the main path (k_m) and backup path (k_b):

$$PRR_o = (k_m + k_b)/k_s, \tag{20}$$

where $k_m = PRR_m \times R \times T_d$ and $k_b = R \times T_b$. Substituting Eq. (19) and Eq. (20) into Eq. (17) gives PRR_{avg}.

C. Verification of the Mathematical Modelling

The mathematical modelling is verified by comparing model results with NS2 simulation results for the PRR performance. Two network topologies are investigated with different numbers of nodes: 100, 150 and 200 nodes. Each pair of source and sink nodes has a main path and a backup path.

To simulate an unreliable routing path, one of the links in the main path exhibits error with a rate R_e in data transmissions. The value of R_e varies from 0.2 to 0.9; while once set, it remains constant for the whole network operation period.

The simulation environment is configured as follows. The data link layer is IEEE 802.15.4 with the communication of Two-Ray Ground. The source node generates a Constant Bit Rate (CBR) traffic of data packets of the size of 100 bytes every 10 seconds, i.e., data rate of $R = 0.1$ packet/sec. The radio range of each node is 40 m. Other settings are: $T_t = 6,000$ sec, $T_p = 600$ sec, $PRR_t = 0.8$, and $R_m = 4$.

Model verification results are summarized in Table II. Under different values of R_e, the PRR results derived from the the theoretical modelling are shown in the second column of Table II. Shown in the last two columns of Table II are NS2 simulation results of the PRR performance and corresponding

95% confidence intervals. It is seen from Table II that the PRR results from the theoretical modelling well describe the PRR performance of the path update and maintenance processes. This is evidenced by two observations: 1) the theoretical PPR values match well with those from NS2 simulations, and 2) the theoretical PRR values mostly fall within the 95% confidence interval of the simulation results.

Results in Table II also support the modelling assumption that the scale of the network does not affect the PRR performance for the developed route update and maintenance processes. As shown in Table II, at a given R_e value, the PPR values for different sizes of networks are similar. This is because the propagation delay of transmitting the route update packet to the source node is small in comparison with the evaluation time during which the route update and maintenance processes verify the path condition.

VI. Performance Evaluation

This section evaluates the performance of the route update and maintenance processes implemented in the hierarchical proactive routing framework shown in Fig. 1. The main objective of the route update and maintenance processes is to provide reliable and real-time data transmissions in large-scale industrial WSNs. Therefore, the scalability of our approach is evaluated. Furthermore, PRR and end-to-end delay are used as the performance metrics to characterize the reliability and real-time performance of the route update and maintenance processes. While a higher PRR is required for an improved reliability, a smaller end-to-end delay ensures timeliness.

As proactive routing, our approach in this paper will be compared with both the popularly used proactive routing protocol DSDV and the recent hierarchical proactive routing protocol ROL/NDC. Two types of DSDV are evaluated: DSDV with the simple periodic update process and DSDV with RPL's Trickle algorithm (DSDV-Trickle). Moreover, the presented approach will also be compared with the two popular reactive routing protocols AODV and AOMDV.

A. NS2 Simulation Setup

All selected routing protocols and our routing approach use the same NS2 simulation setup parameters for IEEE 802.15.4

TABLE II
PRR EVALUATION FROM THEORETICAL MODELLING AND SIMULATIONS.

R_e	Theoretical PRR	Results from NS2 Simulation		
		Topology	PRR	95% Confidence Interval
0.3	0.9967	100 nodes	0.9993	[0.9982, 1.0005]
		150 nodes	0.9973	[0.9950, 0.9997]
		200 nodes	0.9980	[0.9958, 1.0003]
0.4	0.9600	100 nodes	0.9686	[0.9605, 0.9769]
		150 nodes	0.9660	[0.9566, 0.9754]
		200 nodes	0.9733	[0.9713, 0.9754]
0.6	0.9517	100 nodes	0.9410	[0.8809, 1.0012]
		150 nodes	0.9433	[0.8727, 1.0140]
		200 nodes	0.9509	[0.9064, 0.9956]
0.8	0.9800	100 nodes	0.9800	[0.9764, 0.9836]
		150 nodes	0.9773	[0.9746, 0.9801]
		200 nodes	0.9773	[0.9739, 0.9808]

data link layer and physical layer: the maximum transmission range of each sensor node is 75 m, the bandwidth of the wireless channel is 250 kbps, each data message is 50 bytes long, and the packet header is fixed to 30 bytes.

Our approach for route update and maintenance presented in this paper constructs routing paths based on the route discovery mechanism from AOMDV. Our approach requires the following additional settings: $PRR_t = 0.8$, $R_m = 4$, $T_t = 6000$ sec, $T_p = 600$ sec, and $T_l = 6$, 15, and 30 sec.

For ROL/NDC, the number of cluster heads is set as 5% of the total number of nodes. Similarly, our approach use the number of core routing nodes equals to 5% of the total number of nodes. This is because the function of the core routing nodes is similar to that of the cluster heads in ROL/NDC.

For DSDV-Trickle, the default values from RPL's RFC [12] are used: I_{min} = 8 ms and I_{max} = 2.3 hours. With a fixed periodic update period, the original DSDV calculates the maximum period that each node must receive at least one route update packet from its neighbours. RPL's Trickle algorithm dynamically adjusts the update period. It uses a neighbour unreachability detection (NUD) algorithm to determine a neighbour is no longer reachable. The NUD algorithm has five states: reachable, stale, delay, probe and unreachable [25]. It requires each node to broadcast or multicast a small probe packet to its neighbour every 4 seconds. When the state changes to delay, the node transmits a solicited probe packet (S-Probe) and requests an ACK from the node at the other end of the link. If the node cannot receive an ACK back after 4 S-Probes, the node marks the link as unreachable.

For DSDV, AODV and AOMDV, all setup parameters are the default NS2 values without any modification.

B. Comparisons with the Proactive Routing Protocols

Experiments are conducted to evaluate the performance of scalability, reliability and timeliness of our routing approach. For scalability, multiple network sizes are tested: 10, 40, 90, 150 and 300 nodes. For reliability and timeliness, permanent communication breakdowns in the network topology are created by randomly selecting a set of sensor nodes in the network to become faulty nodes. A faulty node is a node with its energy level equal to zero J. after a specific period of time. Comparisons of the experimental results for the three proactive routing protocols are shown in Figs. 7 and 8.

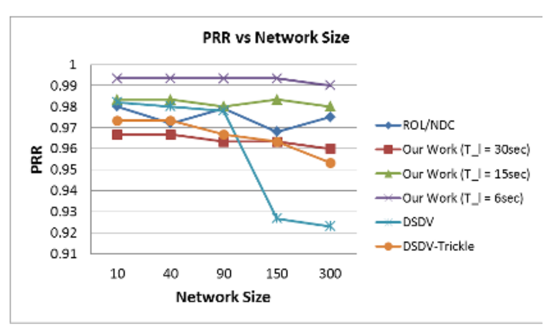

Fig. 7. PRR versus network size.

The first observation from Figs. 7 and 8 is that our approach scales well as the network size increases. Both the PRR and average end-to-end delay do not change much under different network sizes. For ROL/NDC, the PRR performance does not change much with the increase in the network size, but the end-to-end delay performance deteriorates significantly when the network size increases from 150 nodes to 300 nodes. This indicates the poor scalability of the ROL/NDC for large-scale WSNs. DSDV does not scale well in both PRR and end-to-end delay as the network size increases. The two route update processes in our approach with two different update periods, in which one is adjustable dynamically, enable flexible and adaptive route update. The ROL/NDC and DSDV-Trickle have some mechanisms to dynamically adjust the update process, giving a certain degree of scalability. However, the single route update process with a fixed update period in DSDV prevents scalability in large-scale WSNs.

Fig. 7 shows that our approach can be configured for improved PRR performance. With the decrease of the T_l value, the PRR performance of our approach gets better. For example, at $T_l = 6$ sec, the PRR from our approach behaves the best among all protocols. When $T_l = 15$ sec, the PRR performance gets worse but is still comparable with that of ROL/NDC. A bigger T_l means less frequent route updates and thus smaller routing overhead. Therefore, T_l should be tuned to show a good trade-off between the PRR performance and the routing overhead for a specific application.

Our approach with $T_l = 15$ sec and ROL/NDC exhibit comparable PRR performance. This results from their flexible route update processes. Both routing approaches have implemented multipath routing. When the main path becomes poor, an alternative path is activated immediately. Our approach can also adjust the global update period. In comparison, Fig. 7 shows that DSDV and DSDV-Trickle behave worse than our approach and ROL/NDC. Particularly, when the network size becomes large, e.g., 150 and 300 nodes, the PRR performance of DSDV drops significantly. This is due to DSDV's inflexible route update process with a fixed update period. DSDV requires a significant period of time to distribute the new route update information to all participating nodes. Thus, it experiences a long delay in detecting the poor routing path and establishing a new routing path.

For end-to-end delay, Fig. 8 shows that our approach outperforms both ROL/NDC and DSDV for large networks (e.g., 300 nodes). In smaller networks (from 10 to 150 nodes), ROL/NDC behaves with a much higher end-to-end delay, about 10 to 20 times higher, than both DSDV and our approach. This is mainly because ROL/NDC uses TDMA in its MAC protocol. Each node must wait for the beginning of its allocated time slot to transmit the packet. In comparison, both DSDV and our approach use CSMA as the underlying MAC protocol. With CSMA, each node can begin to transmit data packets once the wireless channel becomes available. DSDV-Trickle reacts slowly when there is a change in the network. Each node along the routing path can only send the route update packet at the end of the periodic update period. This will cause notably high delays in large-scale networks. Therefore, the end-to-end delay performance of DSDV-Trickle is not displayed in the plot.

In summary, the experiments have demonstrated that our approach outperforms ROL/NDC, DSDV-Trickle and DSDV in terms of scalability, PRR performance and end-to-end delay.

C. Comparisons with Reactive Routing Protocols

For comparisons with reactive routing protocols, the experimental configurations are the same as those in Section VI-B except for R_e settings. To simulate varying channel conditions in industrial environments, a two-state error model is used with the value of R_e switching between 0 and a non-zero value in the range between 0.2 and 0.9. A good channel condition ($R_e = 0$) remains for 300 sec, and then it changes to a poor condition (non-zero R_e) for the next 1,200 sec. This process repeats in the same manner for the whole network operation.

In comparison with the two popular reactive routing protocols AODV and AOMDV, our proactive routing in this paper shows similar PRR performance, as clearly shown in

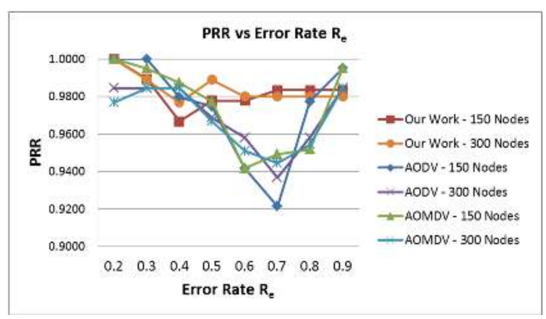

Fig. 9. PRR versus R_e in the two-state R_e model.

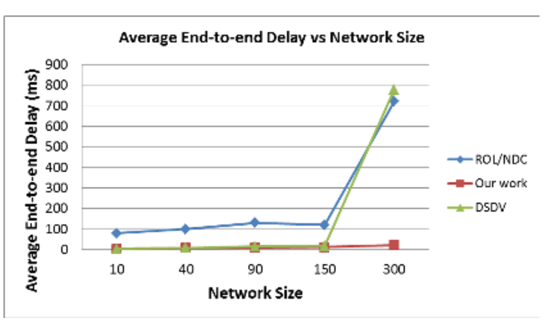

Fig. 8. End-to-end delay versus network size.

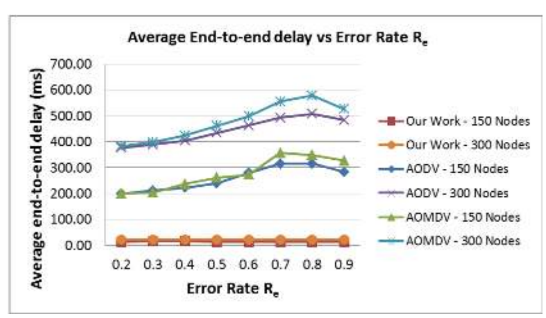

Fig. 10. End-to-end delay versus R_e in the two-state R_e model.

Fig. 9. Reactive routing creates a routing path only when the source node has a packet to transmit. Thus, in general, reactive routing responds to the changes in network conditions more quickly than proactive routing. This avoids using poor links in establishing a routing path. However, in proactive routing, the routing path must be established at the beginning of the network operation. Then, it relies on route update and maintenance processes to maintain the up-to-date routing information. The results in Fig. 9 show that the route update and maintenance processes incorporating with a hierarchical proactive routing in this paper are efficient, which lead to comparable PRR performance to that from AODV and AOMDV.

For end-to-end delay, the route update and maintenance processes incorporating with the proactive routing in this paper exhibits a major advantage over both AODV and AOMDV. As shown in Fig. 10, they maintain the end-to-end delay at a much lower level. For the topology of 300 nodes, the delay from our approach is about 23 ms, compared to over 200ms from AODV and over 30ms from AOMDV. Moreover, the end-to-end delay does not change much in our approach with the increase of R_e for both topologies of 150 and 300 nodes. However, AODV and AOMDV give several times larger end-to-end delay, and the delay tends to increase as R_e or the network size increases. When the routing path becomes poor, AODV and ADMDV must terminate the path and establish a new path. In comparison, our approach simply switches to the backup path. This explains why our approach behaves with much better delay performance.

In summary, the experimental studies show that our approach in this paper not only behaves better than the popular proactive routing protocols ROL/NDC, DSDV and DSDV-Trickle, but also outperforms the popular reactive routing protocols AODV and AODMV.

D. Comparison of Routing Overheads

Our approach in this paper creates a small amount of routing overhead when the network size grows large. The average routing overheads are 288, 481 and 961 packets for network sizes of 50, 100 and 200 nodes, respectively. The small overhead is achieved because our approach only requires the sensor nodes in the core routing level to maintain multiple node-disjoint routing paths. A node in the local routing level is responsible for maintaining a single link to a nearest core routing node. In comparison, due to their on-demand behaviour, AODV and AOMDV creates more routing overheads than our approach. For network sizes of 50, 100 and 200 nodes, AODV creates overheads of 1725, 3331 and 6691 packets, respectively, while AOMDV generates overheads of 837, 1413 and 2862 packets, respectively. In both AODV and AOMDV, the source node must establish a new routing path every time when it has a new packet to transmit to the sink.

For the same network sizes, DSDV introduces the largest routing overhead among all routing protocols investigated: 4349, 14841 and 57456 packets, respectively. DSDV's simple periodic update process requires each node to periodically transmit a route update packet to all its neighbouring nodes. The route update packet includes all routing information in its routing table. As the network size becomes large, the size of the route update packet becomes large too.

In DSDV-Trickle, the total amount of routing overhead can be significantly improved over the original DSDV (2772, 8184 and 19505 packets, respectively, for the same network sizes). However, Each node in DSDV-Trickle is still required to include all its routing information in the route update packet. Therefore, the overall routing overhead is still much higher than that from our approach.

ROL/NDC aims to reduce the total amount of routing overhead, especially the routing overhead from the cluster setup period. Similar to our approach in this paper, ROL/NDC also maintains multiple routing paths. However, ROL/NDC forms a new cluster formation at the beginning of each transmission round. This leads to a notable increase in the total amount of routing overhead as transmission rounds get shorter.

In summary, our approach in this paper behaves with the smallest routing overhead among all proactive and reactive routing protocols investigated in our experiments.

VII. Conclusion

To provide reliable and timely data transmissions for routing process in large-scale industrial WSNs, efficient route update and maintenance processes have been presented in this paper. They are incorporating with a two-tier hierarchical proactive routing framework, in which core nodes establish multiple disjoint routing paths for each source-sink pair. With relatively long global update periods, the global update process evaluates routing paths and updates them as needed using PRR metric. With shorter local update periods, the local update process detects potential problems on the links along the routing paths between source-sink pairs. Then, when required, it informs the global update process for early path performance evaluation and route update. For the presented processes, mathematical models have been developed to estimate the routing path performance theoretically. Simulation studies have been conducted to demonstrate the presented approach with comparisons with the popularly used routing protocols AODV, AOMDV, DSDV, DSDV-Trickle, and ROL/NDC for large-scale industrial WSNs. The results have shown that our approach in this paper: 1) shows good scalability as the network size increases; 2) reduces the end-to-end delay up to 30 times while improving PRR in comparison to proactive routing protocols ROL/NDC, DSDV and DSDV-Trickle; 3) suppresses the end-to-end delay up to 15 times while providing comparable PRR in comparison with reactive routing protocols AODV and AOMDV; and 4) shortens the routing overhead up to 60 times in comparison with all routing protocols investigated in the paper. Therefore, the approach presented in this paper enables reliable and real-time routing for large-scale industrial WSNs.

References

[1] V. Gungor and G. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, Oct 2009.

[2] B. Lu and V. Gungor, "Online and remote motor energy monitoring and fault diagnostics using wireless sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 11, pp. 4651–4659, Nov 2009.

[3] G. Anastasi, M. Conti, and M. Di Francesco, “A comprehensive analysis of the MAC unreliability problem in IEEE 802.15.4 wireless sensor networks,” *IEEE Transactions on Industrial Informatics*, vol. 7, no. 1, pp. 52–65, Feb 2011.

[4] M. Jonsson and K. Kunert, “Towards reliable wireless industrial communication with real-time guarantees,” *IEEE Transactions on Industrial Informatics*, vol. 5, no. 4, pp. 429–442, November 2009.

[5] J. Niu, L. Cheng, Y. Gu, L. Shu, and S. Das, “R3E: Reliable reactive routing enhancement for wireless sensor networks,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 784–794, Feb 2014.

[6] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing,” *Wireless Networks*, vol. 11, pp. 419–434, Jul 2005.

[7] M. K. Marina and S. R. Das, “Ad hoc on-demand multipath distance vector routing,” *Wireless Communications and Mobile Computing*, vol. 6, no. 7, pp. 969–988, 2006.

[8] L. Pradittasnee, Y.-C. Tian, and D. Jayalath, “Efficient route update and maintenance processes for multipath routing in large-scale industrial wireless sensor networks,” in *Australasian Telecomm. Networks and Appl. Conf. (ATNAC)*, Brisbane, Australia, 7-9 Nov 2012, pp. 1–6.

[9] L. Pradittasnee, Y.-C. Tian, D. Jayalath, and S. Camtepe, “An efficient proactive route maintenance process for reliable data transmissions in sensor networks,” in *7th Int. Conf. on Signal Processing and Comm. Systems (ICSPCS)*, Gold Coast, Australia, 16-18 Dec 2013, pp. 1–10.

[10] G. Pei, M. Gerla, and T.-W. Chen, “Fisheye state routing: a routing scheme for ad hoc wireless networks,” in *IEEE Int. Conf. on Communications (ICC)*, vol. 1, 2000, pp. 70–74.

[11] A. Bari, A. Jaekel, J. Jiang, and Y. Xu, “Design of fault tolerant wireless sensor networks satisfying survivability and lifetime requirements,” *Computer Communications*, vol. 35, no. 3, pp. 320 – 333, 2012.

[12] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, “RPL: IPv6 routing protocol for low-power and lossy networks,” *Wireless Communications and Mobile Computing*, vol. 6, no. 7, pp. 969–988, 2012.

[13] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Hawaii Int. Conf. on System Sciences*, Jan. 2000, p. 10 pp. vol.2.

[14] A. Manjeshwar and D. Agrawal, “TEEN: a routing protocol for enhanced efficiency in wireless sensor networks,” in *15th Int. Parallel and Distributed Processing Symposium*, April 2001, pp. 2009–2015.

[15] G. Smaragdakis, I. Matta, and A. Bestavros, “Sep: A stable election protocol for clustered heterogeneous wireless sensor networks,” in *2nd Int. Workshop on Sensor and Actor Network Protocols and Applications (SANPA 2004)*, 2004, pp. 1–11.

[16] L. Qing, Q. Zhu, and M. Wang, “Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks,” *Computer Communications*, vol. 29, no. 12, pp. 2230 – 2237, 2006.

[17] Y.-C. Tian and L. Gui, “QoC elastic scheduling for real-time control systems,” *Real-Time Systems*, vol. 47, no. 6, Dec 2011.

[18] J. Heo, J. Hong, and Y. Cho, “EARQ: Energy aware routing for real-time and reliable communication in wireless industrial sensor networks,” *IEEE Transactions on Industrial Informatics*, vol. 5, no. 1, pp. 3–11, Feb 2009.

[19] Y. Wang, M. C. Vuran, and S. Goddard, “Cross-layer analysis of the end-to-end delay distribution in wireless sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 305–318, Feb 2012.

[20] V. Gungor, C. Sastry, Z. Song, and R. Integlia, “Resource-aware and link quality based routing metric for wireless sensor and actor networks,” in *IEEE International Conference on Communications (ICC)*, Beijing, China, June 2007, pp. 3364 –3369.

[21] H. Shen, Z. Li, and K. Chen, “A scalable and mobility-resilient data search system for large-scale mobile wireless networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1124 – 1134, May 2014.

[22] O. Younis and S. Fahmy, “HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366 – 379, oct.-dec. 2004.

[23] G. Tian and Y.-C. Tian, “Modelling and performance evaluation of the IEEE 802.11 DCF for real-time control,” *Computer Networks*, vol. 56, no. 1, pp. 435–447, January 2012.

[24] J. Wang, Y. Liu, Y. He, W. Dong, and M. Li, “QoF: Towards comprehensive path quality measurement in wireless sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 4, pp. 1003–1013, Apr 2014.

[25] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, “Neighbor discovery for IP version 6 (IPv6),” *Wireless Communications and Mobile Computing*, vol. 6, no. 7, pp. 969–988, 2007.

Lapas Pradittasnee received the Ph.D. degree in information technology from Queensland University of Technology, Brisbane, Australia, in 2014. Since then, he has been with the Faculty of Information Technology, King Mongkut’s Institute of Technology Ladkrabang, Bangkok, Thailand, as a lecturer. His research interests include computer networks, software defined networks and wireless communications with regard to reliable real-time routing protocols.

Seyit Camtepe (S’04 - M’07) received the Ph.D. degree in computer science from Rensselaer Polytechnic Institute, New York, USA, in 2007. From 2007 to 2013, he was with the Technische Universitaet Berlin, Germany, as a Senior Researcher and Research Group Leader in Security. Since 2013, he has been with Queensland University of Technology, Australia, as a Lecturer. His research interests include mobile and wireless communication, pervasive security, and applied and malicious cryptography.

Yu-Chu Tian (M’00) received the Ph.D. degree in computer and software engineering from the University of Sydney, Australia, in 2009, and the Ph.D. degree in Industrial Automation from Zhejiang University, Hangzhou, China, in 1993. He is currently a Professor of Computer Science at the School of Electrical Engineering and Computer Science, Queensland University of Technology, Australia. His current research interests include big data processing and distributed/cloud computing, Networks and communications, real-time computing, and control theory and engineering.