A Delay-constraint Routing Metric Based on ETX for RPL

Bi Zeng  
School of Computer, Guangdong University of Technology, Guangzhou510006, Guangdong, China

Bo Jiang  
School of Computer, Guangdong University of Technology, Guangzhou510006, Guangdong, China

Xiangyi Cao  
School of Computer, Guangdong University of Technology, Guangzhou510006, Guangdong, China

Jianqi Liu*  
School of Information Engineering, Guangdong Mechanical & Electrical College, Guangzhou510000, Guangdong, China  
*Corresponding author(E-mail: liuqianqi@ieee.org)

Abstract
IPv6 is a popular direction of the Wireless Sensor Networks (WSNs). The Internet Engineering Task Force (IETF) has designated the Routing Protocol for Low power and Lossy networks (RPL) as the standard IPv6 routing protocol for WSNs. Routing metrics are used by RPL to build a Destination Oriented Directed Acyclic Graph (DODAG) as the network topology. And Expected Transmission Count (ETX) is a frequently-used routing metric for RPL. According to ETX, the Objective Function (OF) to be minimized is the expected total number of packet transmissions required to successfully deliver a packet to the ultimate destination. However, it does not take the Real-time data transmission into the account. In this paper, the ETX_D algorithm which based on ETX is proposed not only to keep the characteristics of ETX searching for reliable links but also to reduce the communication delay. Furthermore, combining the proposed algorithm with ant routing algorithm to reduce the delay of the topology rebuilt. ETX_D has been implemented based on ContikiOS and compared with the standard ETX. The experimental results have shown that ETX_D spends less time in transferring the message successfully.

Key words: ETX, RPL, Routing Metric, Duty Cycling Time, ContikiOS.

1. INTRODUCTION
RPL is the standard IPv6 routing protocol designed by IETF for low-power Wireless Sensor Networks (Arun Kumar and Alagumeenaokshi, 2014), such as vehicle ad hoc network (VANET), industrial wireless network and smart home system (Liu et al., 2015; Barcelo et al., 2016). This routing protocol provides several criteria, such as Hop Count, Node Energy and Link Latency, that can be used as routing metrics (Gaddour et al., 2014) to establish network topology and choose the optimal routing path between nodes. In all routing metrics, ETX has become the focus of great concern to researchers. According to ETX routing metric, the objective function to be minimized is the total expected number of packet transmissions required to deliver a packet from the source node to the final destination successfully (Couto et al., 2005). The primary advantage of ETX-based routing protocol is the high-reliability of communication links, and it is very suitable for communication in Low power and Lossy Networks (LLNs). Consequently, many researchers have proposed a number of ETX-based routing metrics, forming the ETX family.

Balakrishnan and Koksal have proposed modified Expected Transmission Count (mETX) in paper (Koksal and Balakrishnan, 2006). The mETX aims at improving the accuracy of the loss estimator function precisely and chooses those links with relatively unfluctuating performance as optimal paths. Jiang has introduced an approach in (Jiang et al., 2011), called Exclusive Expected Transmission Time (EETT). In order to get the maximum throughput, this algorithm tries to find multi-channel routes with minimum interference. EETT is designed for low traffic, multi-radio, and large-scale networks. Inverse Expected Transmission Count (invETX) was proposed by Javaid in (Javaid Bibi and Djouani, 2012). Just like its name, invETX is the inverse of ETX’s value. Reducing the computational complexity of ETX is the priority target of invETX, which indirectly improves the resource utilization and efficiency of the algorithm.

In paper (Comsa and Ivanciu, 2015), Anamaria Comsa and Justin Ivanciu introduced a new objective function, named OF-DELAY. The purpose is to minimize the end-to-end delay in the application-aware routing
for WSNs. In paper (Mohamed and Mohamed, 2015), Belghachi Mohamed and Feham Mohamed designed the QoS_RPL, an objective function which combined the residual energy and the transmission delay based on ant colony optimization. GI-RPL was proposed in another paper (Tian et al., 2013), aimed at getting the minimum value of the average delay towards the DAG root without the condition of retransmission.

Obviously, mETX, EETT and invETX only focused on how to calculate ETX and not took end-to-end transmission delay into consideration. In some application scenarios, such as smart home system, the real-time responsiveness affects the user experience and the transmission delay must be reduced as much as possible. According to OF-DELAY, QoS_RPL and GI-RPL, although reducing latency was considered as the primary objective, they ignored the characteristics of the lossy link in WSNs, resulting in increased packet loss rate.

This presented paper proposes a delay-constraint routing metric based on the standard ETX, which not only retains the characteristics of the ETX searching for reliable links, but also considers the communication delay. It is more suitable for real-time application scenario, such as smart home, to meet the requirements of delay and reliability. Furthermore, we have taken some special situations into account by introducing the ant routing algorithm, such as the reconstruction of the DODAG when a node is dead. Finally, the improved algorithm has been implemented in Contiki Operating System and evaluated on Cooja Simulator in a network topology with 14 nodes.

The rest of the paper is organized as follows: In section 2, some relevant routing metric algorithms and development situation are introduced. In section 3, the achievement of standard ETX and the delay calculation method are presented. Section 4 is devoted to the improved routing algorithm and the network topology design. Section 5 presents the performance evaluation results. Finally, Section 6 concludes the paper and gives some open issues.

2. RELATED WORKS

RPL has been defined in RFC 6550. And this routing protocol provides several criteria defined in RFC 6551, such as Hop Count, Node Energy and Link Latency, that can be used as routing metrics to establish network topology. In all routing metrics, ETX has become one of the most commonly used routing metric.

The original ETX has been proposed in paper (Couto et al., 2005). In this paper, the value of ETX is the expected total number of packet transmissions required to successfully deliver a packet to the ultimate destination, and it is calculated by the counts of broadcast probe. The current stage of ETX research can be divided into two kinds: one is the comparison with the other routing metric; the other is how to make ETX adapt to different networks. There is rare paper caring about the delay of the network using ETX. However, for some WSN systems, a node must be timely response once the controller sends a command to this node.

Delay is another routing metric. But there has not been a standard defined to calculate it. In many papers, all kinds of methods have been proposed to calculate the delay of the RPL. In paper (Pinto et al., 2013), authors have proposed a method to estimate the end-to-end delay. The end-to-end delay included two parts: the packet processing time in nodes, and the transmission time over the link. Although another paper (Pinto et al., 2014) has ameliorated the EEDEM by decreasing the end-to-end delay, they all assumed that the nodes are always powered on. And in paper (Gonizzi, Monica and Ferrari, 2013), authors have proposed a method to calculate the transmission time over the link between two nodes with Duty Cycling Time. But this method was worked in a perfect environment where packets loss rate is non-existent.

3. RPL ROUTING METRIC

In a WSN using RPL, the topology can be built by utilizing one of routing metrics defined in the RPL, which to adapt to different scenarios (Gaddour et al., 2014). Nodes in a WSN build the DODAG base on the RPL routing metric of the OF. This paper only discusses the simple scenario of a WSN having two RPL Instances (Long et al., 2013). Both RPL instances work independently and each instance has only one DODAG. So far, ROLL working group has been designated two OFs, the Objective Function Zero (OF0) in RFC 6552 and the Minimum Rank with Hysteresis Objective Function (MRHOF) in RFC 6719. In OF0, nodes make the DODAG base on the minimum number of hops to the root. Whereas in MRHOF, nodes make the DODAG base on the routing metric such as the ETX. The original ETX algorithm has been implemented in the ContikiOS.

3.1 ETX Routing Metric

There is the definition of ETX in (Couto et al., 2005) and the definition is summed as follows: The ETX of a link is the expected total number of packet transmissions required to successfully deliver a packet from a node to its neighbor. And a route is the sum of each link. For example, a three-hop route and every hop has the perfect link, the ETX of the perfect link is 1, then the ETX of the route is 3; the ETX of a one-hop route with a 50% delivery ratio. The only one hop has the lossy link, the ETX of the lossy link is 2, so the ETX of the route is 2.

Concluded from (Le and Gerla, 2014; Shi, Jin and Song, 2014), the general idea of the work of ETX is as follows: The ETX of a link is calculated by using the forward and reverse delivery ratios of the link. The
The forward delivery ratio $d_f$ is the measured probability that a data packet is successfully arrived at the recipient, and the reverse delivery ratio $d_r$ is the probability that the ACK packet is successfully received. The expected probability that a transmission is successfully received and acknowledged is $d_f \times d_r$. A sender will retransmit a packet if it is not successfully acknowledged. Because each attempt to transmit a packet can be considered a Bernoulli trial, the expected number of transmissions is:

$$ETX = \frac{1}{d_f \times d_r} \quad (1)$$

The delivery ratios $d_f$ and $d_r$ are measured using dedicated link probe packets. Each node broadcasts link probes periodically, and the period is $\tau$. For example, consider a link $(A, B)$ as the Figure 1, A and B are neighbors:

![Figure 1 The link between node A and its neighbor B](image)

Every node remembers the received probes during the last $\omega$ seconds, and allows it to calculate the delivery ratio from the sender at any time $t$ by this formula:

$$r(t) = \frac{\text{count}(t-\omega,t)}{\omega / \tau} \quad (2)$$

The count$(t-\omega,t)$ is the number of probes received during the window $\omega$. And $\omega / \tau$ is the number of probes that should have been received.

Calculation of a link’s ETX requires both $d_f$ and $d_r$. In the case of the link $(A, B)$ in Figure 1, the formula (2) allows $A$ to measure $d_r$. Each probe sent by node $B$ contains the number of probe packets received from $A$ during the last $\omega$ seconds. So when $A$ receives a probe from $B$, $A$ can know the number of successful sent probe packets. As $A$ has counted the number of sent probe packets, so $A$ can measure $d_f$ of link $(A, B)$. Then $A$ can know the ETX of link $(A, B)$. So as $B$.

### 3.2 Delay Calculation

Energy-efficiency is a relevant concern in WSN applications (Capone et al., 2014; Liu et al., 2015). Even in the smart home systems (Iova, Theoleyre and Noel, 2014; Yang et al., 2014 and Chen et al., 2014), every node in WSN is powered by the electricity instead of the battery, hence energy-saving algorithm can save much power. In order to save the power, the simplest method is to power off the node when it does not work. Therefore, a radio duty cycling mechanisms is used in WSNs to save power. In a radio duty cycling mechanisms, nodes in WSNs can have different Duty Cycling Time ($C_T$) and only send and receive messages when the radio is on. Thus, there is a delay during the transmission between two nodes. For a node to send message successfully, the receiver should be awaked. If the two nodes have different cycle time, one node must wait for the other node. So when two nodes have Duty Cycling Time, the packet transmission delay will appear. According to (Gonizzi, Monica and Ferrari, 2013), if packets loss rate is non-existent, the delay of the link $(A, B)$ in Figure 1 can be shown as the Figure 2:

![Figure 2 The delay between node A and its neighbor B](image)

Figure 2 shows the interval between node $A$ and $B$ while $B$ is successfully transmitting a packet to $A$. Such interval is the sum of 4 contributions: (1) the spending time to prepare a send packet; (2) the waiting time till node $A$ wakes up; (3) the backoff time to check channel availability; (4) the time interval to repeat the packet transmission until the receiver wakes up. In an ideal environment, the shortest interval of a transmission is the
sum of the spending time, backoff time and retransmission time in (1), (3), and (4) contributions respectively. Such interval is denoted as Minimum Forwarding Time (MFT). To compute the forwarding delay $D$ between $A$ and $B$, two cases can be distinguished:

1. $A$ and $B$ have the same cycle time ($C_T$). In this case, $C_{(T,A)} = C_{(T,B)}$. And $\Phi_A - \Phi_B$ is constant. Therefore, $D$ is given by formula (3):

$$D = \begin{cases} 
\Phi_A - \Phi_B & \Phi_A - \Phi_B \geq MFT \\
\Phi_A - \Phi_B + C_{T,A} & \Phi_A - \Phi_B < MFT 
\end{cases}$$

(3)

Where $\Phi_B$ is the $B$’s wake-up phase, and $\Phi_A$ is the $A$’s wake-up phase. $B$ sends a packet to $A$, after the time of MFT, the packet is received by $A$. If $\Phi_A - \Phi_B \geq MFT$, $A$ is wake-up after the time of MFT. So $A$ can process the packet. Else if $\Phi_A - \Phi_B < MFT$, $A$ is sleeping after the time of MFT. So it should wait $A$ wakes up again to process the packet.

2. $A$ and $B$ have different cycle time. In this case, $C_{(T,A)} \neq C_{(T,B)}$. As shown in Figure 3:

![Figure 3 The wake-up phase of A and B](image)

Since $A$ can only receive a packet when $B$ starts to send it in the phase $\varphi$, and the start sending time is not a fixed value in $\varphi$. So $\Phi_A - \Phi_B$ is not constant. Therefore, only the average delay can be calculated once a packet is successfully transmitted. In this case, the average delay $D$ can be expressed with formula (4):

$$D = \frac{(C_{T,A} + MFT) + MFT}{2} = \frac{C_{T,A}}{2} + MFT$$

(4)

The average delay $D$ is the average of the maximum delay and minimum delay. The maximum delay is $C_{T,A} + MFT$, the minimum delay is MFT.

4. DESIGN

According to the section 3, we can conclude the limits of each algorithm:

1. ETX does not care about the delay. It is calculated by the periodic probe packets. Nodes send probe periodically and know how many probe packets they should receive during a timespan from each neighbor. Nodes calculate ETX only by the counts of the probe packets that they send or receive but not by the spending time of the probe transmission.

2. The calculated delay of 3.2 is in ideal conditions. In other words, the link is perfect and ETX of the link is 1. But in real application of WSNs, there are collisions and packet loss rate. Therefore, if a packet is sent successfully, it may be sent multiple times and the spent of transmission time will be more than the calculated delay of 3.2.

In an application of WSN, what we care about is how fast the destination node will react once we send a control command. So it’s an important research direction that how to spend less time to complete a successful transmission. This paper proposes the ETX_D to decrease the successful transmission time by the following:

1. ETX is the expected total number of packet transmissions. So a node can get the least number of a transmission.

2. According to subsection 3.2, a node can get the delay when a packet is sent from itself to its parent node.

3. Once a node knows the ETX and delay, it can calculate the delay of the successful transmission. It is discussed in detail in Figure 4.

According to Figure 4, the ETX_D is calculated by the following steps. Figure 5 is the sample topology that assists to explain the process. Node 1 is assumed as the root:

1. All nodes start to work and broadcast probes. Node 1 is the root and set the ETX_D as 0. Node 2, 3 and 4 is waiting the broadcast probes.

2. Node 2 receives the probes from node 1, 3 and 4. Then get the parameters from these probes and calculate the ETX_D. So are the other nodes.
The node is root

Set the ETX_D as 0

Receive the broadcast probe from neighbors

Put ETX_D in probes; Sending broadcast probe periodically

Calculating the ETX_D using the parameters in the received probe

Changing the ETX_D in probes; Sending broadcast probe periodically

Figure 4 The process of ETX_D calculation.

(3) Nodes change the parameters in probes and broadcast probes. For example, node 2 gets the cycle time ($C_T$) and wake-up phase of node 1 from the node 1’s broadcast probes. Then, this node can calculate the delay from itself to node 1. And calculate the ETX by the method of section 3.1. So the ETX_D of node 2 from itself to node 1 can be calculated. The ETX_D of node 3 from itself to node 1 is the same process. Then node 3 puts the value of ETX_D from itself to root node 1 in the probes. Node 2 receives the probes from the node 3 and calculates the ETX_D of self from node 3 to node 1 and compares with the ETX_D of itself to node 1. Finally, node 2 selects the node with a smaller ETX_D value as the parent node.

(4) The other nodes also use the step (2) and step (3) to select the best parent.

Figure 5 The sample topology.

To calculate the real delay when a packet is successfully transmitted, we assume that ETX means the max times that a packet should be transmitted. The real delay may be the ideal delay plus once cycle time, twice the cycle times or the maximum cycle times of a node. And the times is the value of ETX.

4.1 Improved Calculation of Delay

As the max transmission times is ETX, to transmit a data packet successfully, the packet will be sent one time, two times or ETX times. Because it is hard to get the real number of transmission times, we assume that the data packet transmission frequency has equal probability and it is 1/ETX. So, the average delay $D$ over the link (A, B) (shown as Figure 1) can be concluded:

1. A and B have the same cycle time. $ETX_{(A,B)}$ is the predicted times of data transmissions required to send a packet over the link(A,B).

$$
D = \begin{cases} 
\Phi_a - \Phi_b + C_{T_A} \frac{(0+1+2+...+ETX_{A,B}-1)}{ETX_{A,B}} & \Phi_a - \Phi_b \geq MFT \\
\Phi_a - \Phi_b + C_{T_A} \frac{(1+2+...+ETX_{A,B})}{ETX_{A,B}} & \Phi_a - \Phi_b \leq MFT
\end{cases}
$$

(5)
For example. The ETX is 2, \( \Phi_A - \Phi_B \geq MFT \). If a packet from B is sent one time and it is successfully received by A, the delay is \( \Phi_A - \Phi_B \). Else if two times, it should wait at least the time of \( C_T \) when A receives the packet successfully. The successful transmission time can be calculated by formula (3). Then, the real delay is time of successful transmission plus the waiting time, and it is \( \Phi_A - \Phi_B + C_T \). Therefore, the average delay D is \( ((\Phi_A - \Phi_B) + (\Phi_A - \Phi_B + C_T)/ETX \). Because the number of \( \Phi_A - \Phi_B \) is ETX, the final results is \( \Phi_A - \Phi_B + C_T/2 \).

2. A and B have different cycle time. We have to use an average delay as a substitute in this case. And it is composed of two parts. One part is the time spent in \( (ETX-1) \) times transmission, the other part is the time calculated by formula (4), needed for the last successful transmission. Therefore, the ETX_D can be expressed with formula (6):

\[
D = \frac{C_T/2 + MFT + C_T * (ETX_{A,B} - 1 + ETX_{A,B} - 2 + ... + 2 + 1 + 0) \times ETX_{A,B}}{ETX_{A,B}}
\]

Also, the average delay D is the successful transmission time plus the waiting time.

4.2 Topology

According to (Arunkumar and Alagumenaakshi, 2014), a WSN can be divided into many RPL Instances. Each instance has its own RPL Instance ID and its own DODAG. Nodes can be in different RPL Instances at the same time. So the nodes which were chosen as the DODAG root can have different RPL Instance IDs. In this case each of them can make a DODAG with other nodes in the WSN. Roots also can join another DODAG and work as a non-root node.

An example is shown in Figure 6, where node 2 is chosen as the DODAG root. The lines mean the link between two nodes.

4.3 Delay of a Route

In a DODAG, all nodes only have the route from themselves to the root. So a packet that a node wants to transmit to another node should be sent to the DODAG root first. The root determines the next step. Here, ETX_D is defined as the routing delay from a node to the root.

In section 4.1, we can know the delay of a link. Thus the ETX_D can be concluded by the formula (7):

\[
ETX_D = \begin{cases} 
0 & {DODAG \_ ROOT} \\
ETX_D + D_p & {ETX_D + D_p < D_{\text{max}}} \\
D_{\text{max}} & {ETX_D + D_p \geq D_{\text{max}}} 
\end{cases}
\]

Where, the ETX_D is 0 that means this node is the DODAG root. \( ETX_D \) is the delay from a candidate parent to the root and is announced by the candidate parent. The \( D_p \) is the delay of the link between a node and its candidate parent. It can be concluded by formula (5) if two nodes have the same cycle time and by formula (6) in case of different cycle time. For example, N1 is the root, N2 is the child of N1, and N3 is the child of N2. Hence, the ETX_D of N1 is 0, the ETX_D of N2 is the ETX_D of N1 plus the \( D_p \) between N1 and N2, the result is \( 0 + D_p(\text{N1,N2}) \), the \( D_p(\text{N1,N2}) \) is concluded by formula (5) or (6). Then the ETX_D of N3 is the ETX_D
of N2 plus the $D_p$ between N2 and N3, the result is $(0 + D_p(N1,N2) + D_p(N2,N3))$. The cycle time $C_T$ and wake-up phase $\phi$ can be announced by a probe which is used to calculate the ETX. And $D_{max}$ is a maximum threshold.

Summing up the above contents, there are two points to be noted:

1) Once a node is dead, its child has to take much time to change its parent node. It will have an impact on the transmission delay.

2) The real transmission times is unpredictable. The calculation of ETX is a dynamic result. And we have assumed that the ETX is the maximum number of transmissions. The different transmission times are equiprobable. These suppositions will produce certain error.

In order to reduce the error and delay, this paper improves the calculation of ETX_D by drawing an ant routing algorithm and the improved ETX_D is described in 4.4.

### 4.4 Improved ETX_D

The ant routing algorithm can expand a tiny change and quickly select the best route, without spending too much time on training and learning. And these features can be used to save the convergence time once the network topology has changed. According to (Frey, Grose and Gunes, 2014; Cheng and Hou, 2003; Istikmal, 2013), in ant routing algorithm, a node $A$ sends message to another node based on a probability table. Like ETX_A broadcasts the link probes and gets the necessary parameters from other nodes periodically. Then calculate the probability and choose a node $B$ which has the maximum probability as the best parent. And node $B$ must in the $A$’s parent candidate list.

In the probability table, the increasing probability formula is (8):

$$
p'_{n+1}^i = \begin{cases} 
\frac{\Delta p}{1 + \Delta p} p_n^i, & \text{if } p_n^i \leq 1 - \Delta p \\
1 - \frac{\Delta p}{\Delta p} (p_n^i - 1) + 1, & \text{otherwise}
\end{cases} 
$$

(8)

The decreasing probability formula is (9):

$$
p'_{n+1}^i = \begin{cases} 
\frac{1 - \Delta p}{\Delta p} p_n^i, & \text{if } p_n^i \leq \Delta p \\
\frac{\Delta p}{1 - \Delta p} (p_n^i - 1) + 1 & \text{otherwise}
\end{cases} 
$$

(9)

$p_n^i$ is the probability of the node $i$ for being selected as a parent. $p'_{n+1}^i$ is the next time probability calculation. $\Delta p$ is a parameter which is determined by the message in broadcasted link probes.

From formulas (8) and (9), we know that once $\Delta p$ is increased, the change in the probability values will become sharper. The probability values will increase the largest and decrease the least. Therefore we can use $\Delta p$ to determine the best parent. Here we can conclude $\Delta p$ by formula (10):

$$
\Delta p \propto \left[ \frac{D}{D_p} \right]^{\alpha} \left[ \frac{MFT}{D} \right]^{\beta} \left[ \frac{1}{ETX} \right]^{\gamma}
$$

(10)

$D$ is the delay determined by the formula (3) or (4). $D_p$ is the same as used in formula (7). ETX is the link ETX between two nodes. MFT is the Minimum Forwarding Time. Lastly, $\alpha$, $\beta$ and $\gamma$ are the weight of each factor.

Then the improved ETX_D can be calculated by formula (11):

$$
ETX\_D = \begin{cases} 
0 & \text{DODAG\_ROOT} \\
D_p / p + ETX\_D & \text{ETX\_D} < D_{max} \\
D_{max} & \text{ETX\_D} \geq D_{max}
\end{cases}
$$

(11)

Where, the $p$ is the probability of a candidate parent selected as the best parent.

When the calculation result of ETX jitters fast, it means that the count of the transmission times is not a stable value which means that the calculation of delay will not be accurate. For example, there are three nodes $A$, $B$, and $C$. $A$ selects parent from $B$ with the cycle time 0.1s, $C$ with the cycle time 0.4s, the ETX_D of $C$ and $B$ are both 0. The ETX between $A$ and $B$ jitters fast, and its value can be 3 to 4. But according to section 3.1, ETX is an average value. Although the ETX is 4, it may be sent 5 times that the message is sent successfully. The ETX between $A$ and $C$ is stable, which is always 1. So let’s consider a situation that, according to formula (7), $A$ selects $B$ as the best parent because the ETX is 4, and $A$ has originally selected $B$ as the best parent, when the
ETX becomes 4, A will not change the parent because the calculation of delay is still the minimum, which is 0.4s. But when A sends message to B, it should send 5 times. So the real delay is 0.5s. However, if A selects C as the best parent and delay is 0.4s, then by the formula (10), there is a chance that A will select C as the best parent because (1/ETX) is an impact factor without delay. As ∆pof C may be greater than B, the probability of C calculated by formula (8) or (9) will also be greater than the probability of B, therefore A will select C as the best parent by formula (11).

Also, the ETX between A and B may always be 3 and 4. Thus, it is better that A selects B as the best parent. And when the ETX is stable, the Dp and D will be the main influence factors. So how to balance the influence of each factor is the work of α, β and γ.

The improved ETX_D will definitely work in this situation:

Once the ETX between node A and B becomes 4, the delay is calculated as 0.4s. The ETX between A and C is stable 1 and the delay is 0.4s. Thus, node A can choose the stable side (node C) to avoid the situation that the real send times is larger than the calculated ETX. When the ETX between A and B becomes 3, A will select B as the best parent.

5. PERFORMANCE EVALUATION

These two proposed delay-constraint routing metrics (ETX_D and the improved ETX_D) have been implemented in the Contiki-3.0 Operating System and evaluated with the Cooja Simulator (Osterland et al., 2006; Preiss et al., 2015). The performance has been compared with ETX. The tested scenario is composed of 14 TMote Sky nodes. ContikiMAC is a radio duty cycling protocol and we enhance it to support different sleeping periods of the nodes. The network topology is as shown in Figure 6. Node 2 is the root. The cycle time of node 1 and node 2 are 0s (node 1 and node 2 will not poweroff), node 3 to node 6 is 0.1s, 7 to 10 is 0.2s, 11 to 14 is 0.3s. The link quality is lossy ETX ≥ 2.

Figure 7 and Figure 8 are the DODAGs using ETX on the left and ETX_D on the right with formula (7). The DODAG using ETX_D with ant routing algorithm (the improved ETX_D) is almost like the DODAG using ETX_D with formula (7). But the nodes 4, 11, and 12 always change their parents. So there is no stable DODAG and we have not displayed it. The Figure 7 illustrates the stable DODAGs using ETX on the left and using ETX_D with formula (7) on the right. Once node 1 is dead, Figure 8 illustrates the reconstructed DODAGs.

5.1 DODAG Analysis

![Figure 7 DODAG with node 2 as the root. The left is ETX and right is ETX_D](image1)

![Figure 8 DODAG with node 2 as the root when node 1 is not working. The left is ETX and right is ETX_D](image2)
From the Figure 7, we know that the ETX will avoid large throughput because of collisions of data packets. But the delay of ETX_D with formula (7) is still shorter than ETX, even though the data packets can be retransmitted because of the collision. The delay from node 4 to 1 directly is MFT by the formula (6). But from node 4 to node 7 and then to node 1, the delay is greater than MFT calculated by formula (7). That is why node 4 selected node 1 as the best parent. The same, node 11 selected nodes 2 as the best parent.

For the improved ETX_D that is calculated by formula (11), such as node 4 which is one of the children of node 1, because of too many children, there must be more collisions during the transmission from child nodes to node 1. The link probes may not be sent or received successfully. Therefore, the ETX is not a stable value, the $\Delta p$ changed, then the $p$ changed sharper. But the ETX between node 4 and node 7 may be stable. Once the $p$ changed sharper, in formula (11), the calculation of ETX_D maybe larger. Thus sometimes, node 4 from node 7 to node 1 costs less time than to node 1 directly.

From Figure 8, once node 1 is dead, the ETX tends to reduce the network density. So node 7 changes its best parent as node 6, and node 8 changes its parent as node 5. As a result, node 2 has two children, node 5 and node 6 respectively have three children. This can reduce the collision to root. But for ETX_D, the node 2 has four children once node 1 is dead. And this guarantees the network density and also reduces the time delay from nodes to the root. The ETX between node 7 and node 2 (maybe the ETX is 2) is larger than the ETX between node 7 and node 6 (maybe the ETX is 1). But by the calculation of formula (7), the delay from node 7 to node 2 directly is still shorter. This is why node 7 has selected node 2 as the best parent.

What we need to focus on is that the nodes which have to change their parents are often unable to change their parents immediately. The process of changing the parent node will cost much time. For example, the link between node 8 and node 1, once node 1 is dead, by the formula (7), the ETX_D value between node 8 and node 1 is MFT. Node 8 will wait a long time to notice that node 1 is dead and it can’t transmit any packets by node 1 during this time. After a long time, node 8 chose node 5 as its best parent. But in case of the improved ETX_D, once the ETX increase, the probability $p$ decrease sharper. By the calculation of formula (11), once ETX_D between node 8 and node 1 is larger than the ETX_D between node 8 and node 5, node 8 will change its best parent as node 5. Node 8 didn’t have to wait the long time to change its best parent. So this can reduce the delay.

5.2 Delay Comparison

Each node send a packet every one second. During the simulation, each node sent a total of 1000 packets. And we have recorded transmission delay for every packet. The average delay of the 1000 packets has been computed and compared the average delay of each node with the root. In Cooja Simulator, there is a mote output, we can know the time of a packet sent and received. From the mote output, we can count the delay from a node to the root. The Figure 9 is the delay with node 2 as the root. The Figure 10 is the delay with the root of node 2 and node 1 is dead at the halfway. ETX_D1 stands for ETX_D with formula (7) and ETX_D2 stands for the improved ETX_D with formula (11).

![Figure 9 Comparison of delay with node 2 as the root](image-url)
In Figure 7, node 4 selected node 7 as the best parent using ETX. It means that node 4 transmits less times from node 7 to the root node 2 than to node 2 directly. But in Figure 9, node 4 has improved its delay of 12.5% by using the ETX_D and 25% by the improved ETX_D. The above can prove that, although node 4 using ETX_D1 (ETX_D with formula (7)) should retransmit more times than ETX, but the total delay is shorter than ETX. The node 11 is the same. If node 4 selected node 1 as the best parents, node 1 would have more children, the transmissions among their children have more collisions. Then, the successful transmission times always change. Therefore, the ETX is not a stable value and the real transmission times maybe greater than the calculation of ETX. Once the ETX increase, the ETX_D2 increase sharper, like the example has been discussed at the end of section 5.1. Hence sometimes node 4 and 11 using ETX_D2 (ETX_D with formula (11)) will change their parents to node 7 and 5 to avoid the situation of the real transmission being larger than the calculation of ETX.

In Figure 10, once node 1 is dead, nodes 4, 7 and 8 will change their parent. The calculation of the ETX between each node and node 1 will increase. Once the calculated ETX is over a threshold, the nodes will change their parents. Comparing with Figure 9, the delay of node 3, 4, 7, 8, 9 and 10 are increasing because of the process of changing parents. Then, take the node 7 as the example, the ETX_D with formula (7) decreases the delay because node 7 selected node 2 as the best parent but not the node 6. And the ETX_D2 has decreased the delay about 10% further than the ETX_D with formula (7) because ETX_D2 increase sharper. So it took less time that the ETX_D2 of node 7 from node 1 to node 2 becomes over the threshold or larger than the ETX_D2 of node 7 to node 2 directly.

6. CONCLUSIONS

This paper focuses on the ETX routing metric used by RPL. First, we describe ETX routing metric and Delay algorithm respectively, and discuss their characteristics and limitations. Second, we integrate characteristics of the two algorithms and propose ETX_D-based routing protocol, which can get a high-reliability as well as a low latency of links, especially for real-time application scenarios. Furthermore, ant routing algorithm is also introduced to reduce the delay of topology rebuilt. Finally, ETX_D and improved ETX_D have been implemented in ContikiOS and simulated by Cooja Simulator. By compared with ETX, preliminary results have shown that our solution can guarantee high-reliability of links and minimize total delay of a successful transmission.

On the other hand, there are two problems to be solved:

1) How to calculate the real transmission times or reduce the error of the calculation of transmission times. As the ETX is an average value and is not a real transmission times. Therefore, to reduce the error between ETX and real transmission times is still open to investigate.

2) How to design a formula to calculate $\Delta p$ in 4.4 and how to take the value of $\alpha$, $\beta$ and $\gamma$ to reduce the delay is also open for further research.

Finally, it is important to notice here that although the ant routing algorithm can decrease the delay but it also needs more hardware resources.

ACKNOWLEDGEMENTS

This research was supported by Guangdong Natural Science Foundation (No. 2016A030313734), Guangzhou science and technology plan projects (No. 201604020016), Guangdong Province science and technology plan projects (No. 2016B010108004), and Guangdong Province Special Project of Industry-University-Institute Cooperation (No. 2014B090904080).
REFERENCES


