Chapter 4

Reliable Robust and Real Time Communication Protocol (RRRT)
Wireless Sensor Network is deployed with a large number of sensor nodes. These sensor nodes acquire real-time information and transmit the information. When a large number of sensor nodes are active in transmitting the information, there is a possibility of congestion in the data packets. Congestion occurs due to buffer overflow, channel contention, packet collision, reporting rate, many-to-one nature, and dynamically time-varying wireless channel condition. Congestion causes packet loss which in turn decreases network performance and throughput. So in WSNs, it necessitates a congestion control algorithm for lossless data packet transmission, high energy-efficiency, to prolong system lifetime, improve fairness, improve equality of service (QoS) in terms of throughput (or link utilization) and packet loss ratio along with the minimum packet delay. In this chapter, a reliable, robust and real-time (RRRT) protocol is presented to address the need for robust, real-time and reliable event data delivery with minimum energy consumption and with congestion avoidance in WSNs. The RRRT protocol is a novel transport solution that seeks to achieve reliable and timely event detection with minimum possible energy consumption. RRRT protocol is the proposed protocol for robust and self-stabilized MAC layer of the proposed architecture. The RRRT uses a fault-tolerant optimal path (FTOP) [26] for data delivery. FTOP selects a path for each source node to deliver event packets to the sink node. The selected paths are node-disjoint and FTOP is thus fault-tolerant in the sense that event packets can be received by the sink node even if node failures make some paths broken. It includes a combined congestion control mechanism that serves the dual purpose of achieving reliability and conserving energy. The RRRT protocol operation is determined by the current network state based on the delay-constrained event reliability and congestion condition in the network. If the delay-constrained event reliability is lower than required, RRRT adjusts the reporting frequency of source nodes aggressively to reach the desired reliability level as soon as possible. If the reliability is higher than required, then RRRT reduces the reporting frequency conservatively to conserve energy while still
Reliable Robust and Real-Time Communication Protocol (RRRT)

maintaining reliability. This self-configuring nature of RRRT makes it robust for random, dynamic topology in WSNs. Furthermore, ours is the first work that addresses the reliability at both sensor/sub-sink and sub-sink/sub-sink levels. RRRT incorporates adaptive rate-based transmission control and selective acknowledgements (SACK)-based reliability mechanism during sub-sink and sub-sink communication. Performance evaluation via simulation experiments shows that RRRT achieves high performance in terms of reliable event detection, communication latency and energy consumption in WSNs.

4.1 MOTIVATION AND RELATED WORK

Recently, there has been considerable amount of research efforts, which have yielded many promising communication protocols for wireless sensor networks (WSNs) [27] [28] [29] [30]. The common feature of these protocols is that they mainly address the energy-efficient and reliable data communication requirements of WSN. However, in addition to the energy-efficiency and communication reliability, many proposed WSN applications have strict delay bounds and hence mandate timely transport of the event features from the sensor field to the Sub-sink nodes [31]. Consequently, the unique features and application requirements of WSANs call for a real-time and reliable data transport solution. The functionalities and design of such solution for WSANs are the main issues addressed in this work [32]. RRRT is a novel transport solution that seeks to achieve reliable and timely event detection with minimum possible energy consumption and no congestion. It enables the applications to perform right actions in a timely manner by exploiting both the correlation and the collaborative nature of WSNs. Furthermore, RRRT addresses heterogeneous reliability requirements of both sensor/sub-sink and sub-sink/sub-sink communication. More specifically, for sensor-actor communication, unlike traditional end-to-end reliability notions, RRRT defines delay-constrained event reliability notion based on both event-to-action delay bounds and event reliability objectives. On the other hand, for sub-sink/sub-sink communication, it introduces 100% packet-level reliability mechanisms to avoid inaccurate action decisions in the deployment field. In this way, the RRRT protocol simultaneously addresses event transport reliability and timely action performance objectives of WSNs. In general, compared to the existing transport layer proposals in
the related literature, the main contribution of RRRT is that it concurrently provides robust, real-time communication support and addresses heterogeneous transport reliability requirements for typical WSN applications involving reliable event detection and timely action objectives within a certain delay bound. To this end, the notion of delay-constrained event reliability distinguishes RRRT from the other existing transport solutions proposed for wireless ad hoc and sensor networks. To the best of our knowledge, reliable event transport has not been studied from this perspective before and hence RRRT is the first solution attempt simultaneously addressing the robust, real-time and reliable event transport and action performance using fault tolerant optimal path for data delivery in WSNs.

4.2 RRRT PROTOCOL DESIGN PRINCIPLES

Unlike traditional networks, the sensor/sub-sink network paradigm necessitates that the event features are collaboratively estimated within a certain reliability and real-time delay bound. To achieve this objective with maximum resource efficiency, the RRRT protocol exploits both the correlation and the collaborative nature of the network. In the following sections, we first describe the characteristics and challenges of both sensor/sub-sink and sub-sink/sub-sink communication and then based on these characteristics, we discuss the main design components of the RRRT protocol in detail.

4.2.1 Reliable Event Transport

The RRRT protocol is equipped with different reliability functionalities to address heterogeneous requirements of both sensor/sub-sink and sub-sink/sub-sink communication. Next, the main features of these reliability functionalities are described.

4.2.1.1 Sensor/Sub-sink Transport Reliability

In WSNs, sensor/sub-sink transport is characterized by the dense deployment of sensors that continuously observe physical phenomenon. Because of the high density in the network topology, sensor observations are highly correlated in the space
domain. In addition, the nature of the physical phenomenon constitutes the temporal correlation between each consecutive observation of the sensor. Because of these spatial and temporal correlations along with the collaborative nature of the WSNs, sensor/sub-sink transport does not require 100% reliability [28] [29].

The RRRT protocol also considers the new notion of event-to-action delay bound to meet the application-specific deadlines. Based on both event transport reliability and event-to-action delay bound notions, we introduce the following definitions:

1. The observed delay-constrained event reliability ($DR_o$) is the number of received data packets within a certain delay bound at the sub-sink node in a decision interval $i$. In other words, $DR_o$ counts the number of correctly received packets complying with the application-specific delay bounds and the value of $DR_o$ is measured in each decision interval $i$.

2. The desired delay-constrained event reliability ($DR_d$) is the minimum number of data packets required for reliable event detection within a certain application specific delay bound. This lower bound for the reliability level is determined by the application and is based on the physical characteristics of the event signal being tracked.

3. The delay-constrained reliability indicator ($\alpha$) is the ratio of the observed and desired delay-constrained event reliabilities, i.e.,

$$\alpha = \frac{DR_o}{DR_d}$$

4.1

Based on the packets generated by the sensor nodes in the event area, the event features are estimated and $DR_o$ is observed at each decision interval $i$ to determine the necessary action. If the observed delay constrained event reliability is higher than the reliability bound, i.e., $DR_o > DR_d$, then the event is deemed to be reliably detected within a certain delay bound. Otherwise, appropriate action needs to be taken to assure the desired reliability level in sensor/sub-sink communication.
4.2.1.2 **Sub-sink/Sub-sink Transport Reliability**

In WSNs, a reliable and timely sub-sink-sub-sink ad hoc communication is also required to collaboratively perform the right action upon the sensed phenomena [28]. The RRRT protocol simultaneously incorporates adaptive rate-based transmission control and (SACK)-based reliability mechanism to achieve 100% packet reliability in the required ad hoc communication. To achieve this objective, RRRT protocol relies upon new feedback based congestion control mechanisms and probe packets to recover from subsequent losses and selective-acknowledgments (SACK) to detect any holes in the received data stream. These algorithms are shown to be beneficial and effective in recovering from multiple packet losses in one round-trip time (RTT) especially [30].

4.2.2 **Real-Time Event Transport**

To assure accurate and timely action on the sensed phenomena, it is imperative that the event is sensed, transported to the sub-sink node and the required action is performed within a certain delay bound. We call this event-to-action delay, $\delta_{e2a}$, which is specific to application requirements and must be met so that the overall objective of the sensor/sub-sink sub-sink network is achieved. The event-to-action delay, $\delta_{e2a}$ has three main components as outlined below:

1. **Event transport delay $ET_{del}$**: It is mainly defined as the time between when the event occurs and when it is reliably transported to the sub-sink node. In general, it involves the following delay components:
   
   (a) **Buffering delay ($B_{del}$)**: It is the time spent by a data packet in the routing queue of an intermediate forwarding sensor node $i$. It depends on the current network load and transmission rate of each sensor node.
   
   (b) **Channel access delay ($CA_{del}$)**: It is the time spent by the sensor node $i$ to capture the channel for transmission of the data packet generated by the detection of the event. It depends on the channel access scheme in use, node density and the current network load.
(c) Transmission delay \((TEL_{del})\): It is the time spent by the sensor node \(i\) to transmit the data packet over the wireless channel. It can be calculated using transmission rate and the length of the data packet.

(d) Propagation delay \((P_{del})\): It is the propagation latency of the data packet to reach the next hop over the wireless channel. It mainly depends on the distance and channel conditions between the sender and receiver.

2. Event processing delay \((EP_{del})\): This is the processing delay experienced at the sub-sink node when the desired features of event are estimated using the data packets received from the sensor field. This may include a certain decision interval \([30]\) during which the sub-sink node waits to receive adequate samples from the sensor nodes.

3. Action delay \((A_{del})\): The action delay is the time it takes from the instant that event is reliably detected at the sub-sink node to the instant that the actual action is taken. It is composed of the task assignment delay, i.e., time to select the best set of sub-sinks for the task and the action execution delay, i.e., time to actually perform the action. For a timely action it is necessary that the following relation holds:

\[
\delta_{e2a} \geq B_{del} + EP_{del} + A_{del}
\]  

4.3 CONGESTION DETECTION AND CONTROL MECHANISM

In WSNs, because of the memory limitations of the sensor nodes and limited capacity of shared wireless medium, congestion might be experienced in the network. Congestion leads to both waste of communication and energy resources of the sensor nodes and also hampers the event detection reliability because of packet losses \([30]\). Hence, it is mandatory to address the congestion in the sensor field to achieve real-time and reliable event detection and minimize energy consumption. Only the sub-sink node, and not any of the sensor nodes, can determine the delay-constrained reliability indicator, \(\alpha = \frac{DR_s}{DR_d}\) and act accordingly.
In addition, for efficient congestion detection in WSNs, the sensor nodes should be aware of the network channel condition around them, since the communication medium is shared and might be congested with the network traffic among other sensor nodes in the neighborhood [33]. Therefore, because of shared communication medium nature of WSNs, the sensor nodes can experience congestion even if their buffer occupancy is small.

RRRT protocol uses a combined congestion detection mechanism based on both average node delay calculation and local buffer level monitoring of the sensor nodes to accurately detect congestion in the network. Note that the average node delay at the sensor node gives an idea about the contention around the sensor node, i.e., how busy is the surrounding vicinity of the sensor node. To compute the average node delay at the sensor node $i$, the sensor node takes exponential weighted moving average of the elapsed time. In combined congestion detection mechanism of the RRRT protocol, any sensor node whose buffer overflows due to excessive incoming packets or average node delay is above a certain delay threshold value is said to be congested and it informs the congestion situation to the Sub-sink node. More specifically, the Sub-sink node is notified by the upcoming congestion condition in the network by utilizing the Congestion Notification (CN) bit in the header of the event packet transmitted from sensors to the Sub-sink node. Therefore, if the Sub-sink node receives event packets whose CN bit is marked, it infers that congestion is experienced in the last decision interval. In conjunction with the delay-constrained reliability indicator, $\alpha$, the sub-sink node can determine the current network condition and dynamically adjust the reporting frequency of the sensor nodes.

To achieve timely execution of the right action upon the environment, Sub-sink-Sub-sink ad hoc communication must also be efficiently handled. In this respect, congestion control is also imperative for reliable and timely Sub-sink-Sub-sink ad hoc communication. Hence, combined congestion mechanism of the RRRT protocol is also utilized for Sub-sink-Sub-sink ad hoc communication. The details of adaptive rate-based transmission and congestion control algorithms for Sub-sink-Sub-sink ad hoc communication are explained in the next section.
4.4 RRRT PROTOCOL OPERATION FOR SENSOR-SUB-SINK COMMUNICATION

In this section, we describe the RRRT protocol operation during sensor/sub-sink communication. Recall that in the previous sections, based on the delay-constrained event reliability and the event-to-action delay bound notions, we had defined a new delay-constrained reliability indicator \( \alpha = \frac{DR_o}{DR_d} \), i.e., the ratio of observed and desired delay-constrained event reliabilities. To determine proper event reporting frequency update policies, we also define \( T_i \) and \( T_{sa} \), which are the amount of time needed to provide delay-constrained event reliability for a decision interval \( i \) and the application specific sensor/sub-sink communication delay bound, respectively. In conjunction with the congestion notification information (CN bit) and the values of \( f_i, \alpha_i, T_i \) and \( T_{sa} \), the sub-sink node calculates the updated reporting frequency, \( f_{i+1} \), to be broadcast to source nodes in each decision interval. This updating process is repeated until the optimal operating point is found, i.e., adequate reliability and no congestion condition are obtained. In the following sections, we describe the details of the reporting frequency update policies and possible network conditions experienced by the sensor nodes.

4.4.1 Early Reliability and No Congestion Condition

In this condition, the required reliability level specific to application is reached before the sensor/sub-sink communication delay bound, i.e., \( T_i < T_{sa} \) and no congestion is observed in the network, i.e., \( CN = 0 \). However, the observed delay-constrained event reliability, \( DR_o \), is larger than desired delay-constrained event reliability, \( DR_d \). This is because source nodes transmit event data more frequently than required. The most important consequence of this condition is excessive energy consumption of the sensors. Therefore, the reporting frequency should be decreased cautiously to conserve energy. This reduction should be performed cautiously so that the delay-constrained event reliability is always maintained. Therefore, the sub-sink node decreases the reporting frequency in a controlled manner. Intuitively, we try to find a balance between saving energy and maintaining reliability. Hence, the updated reporting frequency can be expressed as follows:
4.4.2 Early Reliability and Congestion Condition

In this condition, the required reliability level specific to application is reached before the sensor/sub-sink communication delay bound, i.e., \( T_i < T_{sa} \), and congestion is observed in the network, i.e., \( CN = 1 \). However, the observed delay-constrained event reliability, \( DR_o \), is larger than the desired delay-constrained event reliability, \( DR_d \). In this situation, the RRRT protocol decreases reporting frequency to avoid congestion and save the limited energy of sensors. This reduction should be in a controlled manner so that the delay-constrained event reliability is always maintained. However, the reporting frequency can be decreased more aggressively than the case where there is no congestion and the observed delay-constrained event reliability, \( DR_o \), is larger than the desired delay-constrained event reliability, \( DR_d \). This is because in this case, we are farther from optimal operating point. Here, we try to avoid congestion as soon as possible. Hence, the updated reporting frequency can be expressed as follows:

\[
f_{i+1} = f_i \frac{T_i}{T_{sa}}
\]

4.4.3 Low Reliability and No Congestion Condition

In this condition, the required reliability level specific to application is not reached before sensor-sub-sink communication delay bound, i.e., \( T_i > T_{sa} \), and no congestion is observed in the network, i.e., \( CN = 0 \). However, the observed delay-constrained event reliability \( DR_o \), is lower than the desired delay-constrained event reliability \( DR_d \). The RRRT protocol can work with any of these routing schemes. Therefore, to achieve required event reliability, we need to increase the data reporting frequencies of source nodes. Here, we exploit the fact that the \( Dr \ vs \ f \) relationship in the absence of congestion, i.e., for \( f < f_{max} \) is linear. In this regard, we use the multiplicative increase strategy to calculate updated reporting frequency, which is expressed as follows:
\[ f_{i+1} = f_i \frac{DR_d}{DR_0} \]  \hspace{1cm} (4.5)

### 4.4.4 Low Reliability and Congestion Condition

In this condition, the required reliability level specific to application is not reached before sensor-sub-sink communication delay bound, i.e., \( T_i > T_{sa} \), and congestion is observed in the network, i.e., \( CN = 1 \). However, the observed delay-constrained event reliability, \( DR_o \), is lower than the desired delay-constrained event reliability, \( DR_d \). This situation is the worst possible case, since desired delay-constrained event reliability is not reached, network congestion is observed and thus, limited energy of sensors is wasted. Hence, the RRRT protocol aggressively reduces reporting frequency to reach optimal reporting frequency as soon as possible. Therefore, to assure sufficient decrease in the reporting frequency, it is exponentially decreased and the new frequency is expressed by:

\[ f_{i+1} = f_i \frac{DR_o}{(DR_d \times X)} \]  \hspace{1cm} (4.6)

Where \( x \) denotes the number of successive decision intervals for which the network has remained in the same situation including the current decision interval, i.e. \( x \geq 1 \). Here, the purpose is to decrease reporting frequency with greater aggression, if a network condition transition is not detected.

### 4.4.5 Adequate Reliability and No Congestion Condition

In this condition, the network is within \( \beta \) tolerance of the optimal operating point, i.e., \( f < f_{max} \) and, \( 1 - \beta \leq \delta_i \leq 1 + \beta \) and no congestion is observed in the network. Hence, the reporting frequency of source nodes is left constant for the next decision interval:

\[ f_{i+1} = f_i \]  \hspace{1cm} (4.7)

Here, our aim is to operate as close to \( \delta_i = 1 \) as possible, while utilizing minimum network resources and meeting event delay bounds. For practical purposes, we define a tolerance level, \( \beta \), for optimal operating point. The entire RRRT protocol operation is presented in the pseudo-algorithm given in Figure 4.1.
\( x=1; \)

**RRRT()**

If (Congestion)
If (\( \delta < 1 \))
/* Low Reliability and Congestion */
\[
f_{i+1} = f_{i} \frac{DR_{o}}{DR_{d} + x};
\]
\( x = x + 1; \)

else if (\( \delta > 1 \))
/* Early Reliability and Congestion */
\( x = 1; \)
\[
f_{i+1} = \min \left( f_{i} \frac{T_{i}}{T_{na}}, f_{i} \frac{T_{i}}{T_{ff}} \right);
\]
end;

else if (No Congestion)
\( x = 1; \)
if (\( \delta < 1 - \beta \))
/* Low Reliability and No Congestion */
\[
f_{i+1} = f_{i} \frac{DR_{d}}{DR_{o}};
\]
else if (\( \delta > 1 + \beta \))
/* Early Reliability and No Congestion */
\[
f_{i+1} = f_{i} \frac{T_{i}}{T_{na};} \text{ end};
\]
else if (\( 1 - \beta \leq \delta \leq 1 + \beta \))
/* Adequate Reliability and No Congestion */
\[
f_{i+1} = f_{i}; \text{ end};
\]
end;
end;

**Figure 4.1:** Algorithm of RRRT
4.5 RRRT PROTOCOL OPERATION FOR SUB-SINK/SUB-SINK COMMUNICATION

In this section, we describe the protocol operation of RRRT during Sub-sink/Sub-sink communication. The protocol operation is composed of two main states: i) Start-up state, ii) Steady state. In Figure 4.2, the RRRT protocol state diagram for Sub-sink/Sub-sink communication is shown.

The operations at each state are described in detail.

1. **Start-Up State**: When establishing a new connection between sender and receiver, the sender transports a probe packet towards the receiver to capture the available transmission rate quickly. Each intermediate node between the sender and receiver intercepts the probe packet and updates the bottleneck delay field of the probe packet, if the current value of delay information is higher than that of the intermediate node. Initially, the delay value of probe packet is assigned zero. Therefore, after one round-trip-time, the sender gets the estimated rate feedback from the receiver, which results in quick
convergence to the available transmission rate. Furthermore, this probing mechanism of startup phase is also applied after route changes.

2. **Steady State**: This state consists of four sub states: i) Increase, ii) Decrease, iii) Hold and iv) Probe. In the following, we describe the RRRT protocol operations in each sub state:

(a) **Increase**: In this state, the sender increases its transmission rate according to the feedback coming from the receiver. Once an increase decision for sender transmission rate is taken, only \( m \) fraction of the difference between transmission rate feedback \( (R_f) \) and sender current transmission rate \( (R_c) \) is performed. The appropriate fraction value \( (m) \) for the transmission rate increase is obtained as follows: If the hop count along the data path is greater than or equal to 4 for that connection, \( m \) is set to 4. Otherwise, if the hop count is less than 4, then \( m \) is set to the actual hop count value along the path. The inherent spatial reuse property of underlying CSMA/CA based MAC protocol requires this normalization in transmission rate.

(b) **Decrease**: In this state, the sender reduces its transmission rate according to the feedback coming from the receiver. Note that the transmission rate is decreased until the minimum transmission rate \( (R_{min}) \) is reached. \( R_{min} \) represents the minimum transmission rate requirement to transfer a certain amount of data within event-to-action delay bound. \( R_{min} \) can be calculated as follows:

\[
R_{min} = \frac{B}{\Delta_{re2a}}
\]

where \( B \) represents the number of packets that should be transmitted to the sub-sink and \( \Delta_{re2a} \) is remaining event-to-action deadline.

(c) **Hold**: In this state, the required transmission rate is reached. Sender does not change the transmission rate unless route failure or congestion occurs in the network.
(d) **Probe:** In this state, the sender sends a probe packet to the receiver so as to monitor the available transmission rate in the network as in start up phase.

Overall, the RRRT protocol dynamically shapes data traffic based on both delay bounds and the current conditions of the network. Note that in the protocol operation, the sender adjusts its transmission rate in response to the rate feedbacks from the receiver, which are sent with the period of $T_{\text{fdbk}}$. To prevent the sender from overflooding the network in case all the feedback packets from the receiver are lost, the RRRT protocol also performs a multiplicative decrease of transmission rate for each feedback periods, in which the sender does not receive feedback from the receiver up to a maximum of two feedback periods. After the second feedback period, if the sender still does not receive any feedback packet, it enters into probe state so as to monitor the available transmission rate in the network. In this respect, the periods of feedback $T_{\text{fdbk}}$ and probe packets $T_p$ should be larger than one round-trip-time (RTT) and small enough to capture the network dynamics.

### 4.6 FAULT TOLERANT PATH (FTOP) FOR DATA DELIVERY

#### 4.6.1 Network Model for FTOP

We represent a WSN as a graph $G (V, E)$, where $V$ is a set of nodes and $E$ is a set of directed edges between nodes. The nodes set $V$ consists of a sensor nodes set and the sink nodes set. The sink nodes set, notated by $V_{\text{SK}}$, represents the set of resource-rich actors responsible for performing powerful actions. The sensor nodes set, notated by $V_{\text{S}}$, represents the set of sensors responsible for transmitting data packets to sink. The nodes of the set $V_{\text{S}}$ are further classified into two categories: source nodes and intermediate nodes. Source nodes are so near the target that they can monitor the target and send out data packets when specific events are detected. Other sensor nodes are intermediate nodes to forward data packets to other intermediate nodes or to sink nodes. It is noted that if a data packet sent from a node $i$ in set $V_{\text{S}}$ can reach a node $j$ in set $V_{\text{S}}$ or $V_{\text{SK}}$, then there is an edge $<i, j>$ in the edge set $E$. The energy consumption ($E$) in the transmission of a data packet between two nodes includes three parts: transmitting energy consumption ($ET$), receiving energy consumption
(ER), and signal propagation energy consumption (ES) i.e., we set \( E = ET + ER + ES \). We assume that ET and ER are a constant of value EM and ES is distance dependent component formulated as \( \gamma d^\alpha \), where \( d \) is the transmission distance, \( \alpha \) is the path loss exponent (usually, \( 1.5 < \alpha < 4.8 \)), and \( \gamma \) is the per unit energy consumption constant.

We assume that each sensor can determine its position by using GPS positioning devices or the like or GPS-free positioning schemes [34], so the transmission distance can be figured out handily. We have that the energy consumption \( E \) is a function of \( d \) of the form \( E(d) = 2EM + \gamma d^\alpha \). It is noted that we can use \( E(d) \) as a cost function associated with an edge of transmission distance \( d \).

### 4.6.2 Problem Definition of FTOP

We are given a graph \( G=(V,E) \) with the vertices set \( V = V_s \cup V_{sk} \), where \( V_s \) represents sensor node set, \( V_{sk} \) represents the sink node set, and \( V_s \) contains the source node subset \( V_{ss} \) and the intermediate node subset \( V_I \), and with the edge set \( E \) consisting of edges \( <x, y> \), where \( x \in V_s \), \( y \in V_{sk} \cup V_{sk} \). Based on edge set \( E \), we define a path set \( PS \) to represent the paths traversable from \( V_{ss} \) to \( V_{sk} \).

The problem to be solved is to find a maximum set \( P \) of disjoint paths in \( PS \), while minimizing the total cost associated with paths in \( P \). Note that two paths \( p_1 \) and \( p_2 \) are said to be disjoint if they have no common nodes except for the end point in the path. Also note that the cost \( C(p) \) associated with a path \( p \) is defined to be:

\[
C(p) = \sum_{e \in \text{edges}(p)} E(\text{distance of } e)
\]

4.8

Where the cost function \( E(d) \) is defined in section 4.1. To be more precise, the problem has two goals. The first goal is of higher priority and is to maximize the number of disjoint paths, i.e.

\[
\text{Maximize } |P|
\]

4.9

The second goal is to minimize the total energy consumption cost, i.e.

\[
\text{Minimize } \sum_{p \in P} E_f(p)
\]

4.10
Figure 4.3 is a simple example to illustrate a WSN for data delivery. The network consists of 7 sensor nodes S1, ..., S7, and 2 sink nodes, SK1 and SK2, where S1 and S2 are source nodes near the target. The directed edges (arcs) represent the reachable relationship, and each edge is associated with a cost of transmission energy consumption.

### 4.6.3 FTOP Scheme

In this section, we present the Optimal Path Planning (FTOP) [26] scheme and analyze its time complexity. In the FTOP scheme, the initial deployment of sensors and sink nodes is first obtained and transformed to be a cost-flow graph. Then, the flow planning algorithm (FPA) determines the flow plan in the graph. Finally, the flow plan is converted into a data delivery path plan and forwarded to the sensor in the network. The FTOP scheme is described as follows.
Fault Tolerant Optimal Path Scheme (FTOP)

1. Each sensor somehow (e.g., by flooding) reports its status to a specific node called coordinator. The report contains node’s identity, reachable neighbors, and costs on edges to neighbors. The coordinator is sink node.

2. The coordinator collects sensors’ reports.

3. The coordinator converts the received report into the cost flow graph, notated by CFG.

4. Determine the fault tolerant optimal flow plan in the CFG based on the Flow Planning Algorithm (FPA) described below.

5. Coordinator converts the flow plan into data delivery path and then forwards to the sensors.

Given the network with source nodes, intermediate nodes, and sink nodes, the Flow Planning Algorithm (FPA) can find a flow plan such that the number of disjoint paths is maximized while the energy consumption cost is minimized. Its procedure is similar to that of the maximum flow minimum cost (Max-Flow Min-Cost) algorithm to determine the fault tolerant optimal path in a given cost-flow graph (CFG). The FPA is described as follows.

4.6.3.1 Flow Planning Algorithm

This section explains the FPA steps in detail.

In step1, a virtual source vertex VS and a virtual target vertex VT are added in CFG.

In step 2, an edge < VS, s> is added from VS to step 3. An edge <sk, VT> is added for each sink node sk with infinite flow capacity and zero cost. Note that VS and VT are virtual nodes added to assist computation, which are not sensor nodes in the network.

In step 4, a node A with multiple flow-in and multiple flow-out edges should be transformed by Node_Transformation( ) procedure to ensure that only one flow
passes through A. By the procedure, A is transformed to be two nodes A_I and A_O connected by an edge with one flow capacity and zero cost as shown in Figure 4.4.

**Flow Planning Algorithm**

*Input*: a cost flow graph $\text{CFG}(V,E)$  
*Output*: Max Flow Plan $\text{FP}$ in $\text{CFG}$

1. Add a virtual Source Vertex $\text{VS}$ and a virtual target vertex $\text{VT}$ in $\text{CFG}$.
2. Add edge $<\text{VS}, s>$ for each source node $s$, and assign the edge with the Zero cost and one flow capacity.
3. Add each edge $<sk, \text{VT}>$ for each sink node $sk$, and assign the edge with the Zero cost and infinite flow capacity.
4. Execute Node_Transformation () procedure for each intermediate node in $\text{CFG}$ with multiple flow in and multiple flow out.
5. Execute Max-Flow Min-Cost Algorithm for $\text{CFG}$ to decide the Maximal Flow Minimum Cost Plan $\text{FP}$.

![Figure 4.3: Illustration of Node-Transformation](image)

In step 5, FPA determines the minimum cost maximum flow plan in graph $\text{CGF}$ within two phases. In the first phase, the maximum flow value from the virtual source to virtual target is determined. In the second phase, FPS selects the minimum cost flow plan based on the result of the first phase. Here, we realize step 5 by two algorithms. In the first phase, we run the Edmonds-Karp algorithm [41] to determine
the maximum flow. Given the graph CFG, the Edmonds-Karp algorithm will find the maximum flow by augmenting the path along the shortest path as far as possible. In the second phase, we run the minimum cost flow (MinCost) algorithm to find the minimum cost flow. Given the CFG and the flow (say F*) found by Edmonds-Karp algorithm, the MinCost algorithm will try to refine the F* in GCF as lower cost as possible. When the MinCost algorithm is unable to further refine F*, it will return this minimum cost flow (F*). To save space, we do not show the details of the Edmonds-Karp algorithm and the minimum cost flow algorithm. The readers are referred to [41] and [46] for the implementations of the Edmonds-Karp algorithm and the MinCost algorithm.

![Figure 4.4: An illustrative example of Flow Planning](image)

We illustrate FPA execution by a flow planning result shown in Figure 4.5, which is a solution of example in Figure 4.3. As shown in Figure 4.3, each edge contains three fields, which are the flow value, capacity, and cost of edge. The flow value field indicates whether the corresponding edge is selected in the solution. To be more precise, if the flow value is 1, then the corresponding edge is selected in the solution. Furthermore, the capacity field means the upper bound of the flow on the edge; the
cost field is the cost associated with the edge. The flow planning algorithm (FPA) in Figure 4.5 contains three disjoint paths, which are: S1→S4→S5→SK1, S2→S6_I→S6_O→SK2, S3→S7→SK1, respectively. They correspond to three disjoint paths with a minimum total cost for the data delivery system of the WSN in Figure 4.5.

4.6.4 Time Complexity Analysis

The time complexity of proposed method is analyzed as follows. We use the variable A to represent the number of sensors and E to represent the number of edges. The core of our scheme is the FPA, which consists of two phases. In the first phase, adding links to virtual source and sink costs $O(1)$ time, and Node Transformation procedure costs $O(A)$ time. In the second phase, FPA finds the maximum flow and the minimum cost maximum flow, which costs $O(A^2E)$ and $O(AE\log(A))$, respectively. To sum up, the time complexity of our proposed scheme is $O(AE^2 + A^2E\log(A))$.

4.7. RRRT Performance Evaluation

4.7.1 Sensor/Sub-sink Communication

To evaluate the performance of the RRRT protocol during sensor-sub-sink communication, we developed an evaluation environment using J-Sim [70]. For sensor/sub-sink communication scenario, the number of sources, sensor/sub-sink delay bound and tolerance level were selected as $n=8$, 1s and $\epsilon=5\%$, respectively. The event radius was fixed at 45m. We run 10 experiments for each simulation configuration. Each data point on the graphs is averaged over 10 simulation runs. Moreover, in this simulation scenario, the sub-sink nodes, which receive data packets from sensors, stop their movements once they start to receive data.

To further investigate RRRT protocol convergence results, we have compared RRRT protocol, ESRT [94], ATP [93], SPEED [35] protocols in terms of convergence time to (Adequate reliability, No congestion) condition and total energy consumption. The reason for comparison with ESRT and ATP is that both of them are based on event transport reliability notion unlike the other transport layer protocols addressing
conventional end-to-end reliability in WSNs. SPEED are a well-known real time communication protocol. As shown in Figure 4.6 and Figure 4.7, the convergence time and total energy consumption of the RRRT protocol are much smaller than those of ESRT, ATP and SPEED for different initial network conditions. This is because ESRT and ATP do not consider application-specific delay bounds while avoiding network congestion and adjusting reporting rate of sensor nodes.

4.7.2 Sub-sink/Sub-sink Communication

For sub-sink/sub-sink communication scenario, the performance of the RRRT protocol is evaluated and compared against ESRT [94], ATP [93] and SPEED [35]. The main performance metrics that we employ to measure the performance of the
RRRT protocol are aggregate throughput and average packet delay. Here, the aggregate throughput reflects the number of packets successfully received at the destination. By average packet delay, we refer to average latency of data packets during sub-sink/sub-sink communication. All the simulations last for 1000 s. We run 10 experiments for each simulation configuration and each data point on the graphs is averaged over 10 simulation runs.

In Figure 4.8, we present the aggregate throughput results of the RRRT protocol and other ad hoc transport protocols, i.e. ATP, ESRT and SPEED. In terms of aggregate throughput, the RRRT protocol outperforms other transport protocols under comparison, since RRRT dynamically shapes data traffic according to the channel condition and intermediate node feedbacks. In Figure 4.9, we also show the average packet delay results of the RRRT and the other transport protocols. As shown in Figure 4.9, for all simulation configurations, the average packet delay values of RRRT are much lower than those of other protocols, since RRRT captures the available bandwidth in the network quickly and does not allow a burst of packet transmissions with explicit congestion notification and rate feedback based mechanisms.

![Figure 4.7: Throughput](image1)
![Figure 4.8: Average Delay](image2)

4.8 CONCLUSIONS

In this chapter, a robust, real-time and reliable transport (RRRT) protocol was proposed to address the communication challenges introduced by the coexistence of sensors nodes in WSNs. The (RRRT) protocol is a novel transport solution that seeks
to achieve reliable and timely event detection with minimum possible energy consumption. It includes a combined congestion control mechanism that serves the dual purpose of achieving reliability and conserving energy. The (RRRT) protocol operation is determined by the current network state based on the delay-constrained event reliability and congestion condition in the network. If the delay-constrained event reliability is lower than required, (RRRT) adjusts the reporting frequency of source nodes aggressively to reach the desired reliability level as soon as possible. If the reliability is higher than required, then (RRRT) reduces the reporting frequency to conserve energy while maintaining reliability. RRRT uses a FTOP to minimize energy consumption. This self-configuring nature of (RRRT) makes it robust for random, dynamic topology in WSNs. Furthermore, ours is the first work that addresses the reliability at both sensor/sub-sink and sub sink/sub-sink level. Performance evaluation via simulation experiments shows that (RRRT) achieves high performance in terms of reliable event detection, communication latency and energy consumption in WSNs.