applied sciences

MDPI

Article

An Efficient Network Coding-Based Fault-Tolerant Mechanism in WBAN for Smart Healthcare Monitoring Systems

Yuhuai Peng [1,2], Xiaojie Wang [3,*], Lei Guo [1,2], Yichun Wang [3] and Qingxu Deng [1,2,*]

[1] Key Laboratory of Medical Image Computing of Ministry of Education, Northeastern University, Shenyang 110819, China; pengyuhuai@cse.neu.edu.cn (Y.P.); guolei@cse.neu.edu.cn (L.G.)
[2] School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China
[3] School of Software, Dalian University of Technology, Dalian 116620, China; wangych1996@126.com
[*] Correspondence: wangxj1988@mail.dlut.edu.cn (X.W.); dengqx@mail.neu.edu.cn (Q.D.); Tel.: +86-138-898-62359 (X.W.); +86-138-988-86668 (Q.D.)

Received: 12 July 2017; Accepted: 7 August 2017; Published: 10 August 2017

Abstract: As a key technology in smart healthcare monitoring systems, wireless body area networks (WBANs) can pre-embed sensors and sinks on body surface or inside bodies for collecting different vital signs parameters, such as human Electrocardiograph (ECG), Electroencephalograph (EEG), Electromyogram (EMG), body temperature, blood pressure, blood sugar, blood oxygen, etc. Using real-time online healthcare, patients can be tracked and monitored in normal or emergency conditions at their homes, hospital rooms, and in Intensive Care Units (ICUs). In particular, the reliability and effectiveness of the packets transmission will be directly related to the timely rescue of critically ill patients with life-threatening injuries. However, traditional fault-tolerant schemes either have the deficiency of underutilised resources or react too slowly to failures. In future healthcare systems, the medical Internet of Things (IoT) for real-time monitoring can integrate sensor networks, cloud computing, and big data techniques to address these problems. It can collect and send patient's vital parameter signal and safety monitoring information to intelligent terminals and enhance transmission reliability and efficiency. Therefore, this paper presents a design in healthcare monitoring systems for a proactive reliable data transmission mechanism with resilience requirements in a many-to-one stream model. This Network Coding-based Fault-tolerant Mechanism (NCFM) first proposes a greedy grouping algorithm to divide the topology into small logical units; it then constructs a spanning tree based on random linear network coding to generate linearly independent coding combinations. Numerical results indicate that this transmission scheme works better than traditional methods in reducing the probability of packet loss, the resource redundant rate, and average delay, and can increase the effective throughput rate.

Keywords: wireless body area network (WBAN); real-time monitoring; network coding; fault-tolerant; smart healthcare

1. Introduction

Smart healthcare [1] has attracted more and more worldwide attention due to its advantage of flexibility, mobility, and ease of constant monitoring of the patient. Smart healthcare, an integration of various information technologies including Internet of Things (IoTs) [2], cloud computing, and big data processing, aims at building a remote disease prevention and care platform. As a great driving force of the Fourth Industrial Revolution, the IoT is playing an ever more important role in Intelligent Manufacturing, the Smart Grid, Smart Cities, Vehicular Ad Hoc Networks (VANETs), Smart Healthcare, etc. The notion of the IoT can be regarded as an extension of communication networks and the Internet;

it employs sensing technology and intelligent devices to perceive and recognise the physical world, conducts calculations, processes and mines knowledge through network transmission interconnections. Further, IoT realises information exchanges and seamless connections between things and things and persons and things to effect real-time control, precise management, and scientific decision-making. In future networks, the IoT will be able to integrate sensors, wireless communication, embedded computing, cloud computing, big data, etc., and to apply intelligent terminals, mobile computing and the ubiquitous network to all aspects in people's daily lives. Therefore, IoT is a key to solve growing global demand for satisfactory medical services.

As the core technology of medical IoT, wireless body area network (WBAN) can be used for real-time patients monitoring and home healthcare, has wide application prospects, and has huge market potential. WBAN can automatically collect human Electrocardiograph (ECG), Electroencephalograph (EEG), Electromyogram (EMG), body temperature, blood pressure, blood sugar, blood oxygen, and other life sign parameters from the pre-deployed sensors on the patient's body surface or inside their body. This in in order to achieve real-time, convenient, all-weather healthcare monitoring, thus providing a flexible and effective means for real-time healthcare monitoring of patients inside and outside the hospital room, and in Intensive Care Units (ICUs). In the recent years, the emergence of wireless sensor networks in healthcare systems has significantly increased, mainly in the areas of remote health monitoring, medical data access, and communication with caregivers in emergency situations. Using WSN, we can easily design a simple but efficient system to monitor the conditions of patient continuously. Patients can be tracked and monitored in normal or in emergency conditions at their homes, hospital rooms, and in ICUs.

In most countries, soaring medical expenses and the declining availability of medical services have become the main challenges to be addressed by medical and health services at present. Because of the large global population, overall medical resources are relatively scarce. Furthermore, high-speed economic development will also bring about problems such as population aging, rapid growth of chronic diseases, and high sub-health ratio. Therefore, smart healthcare becomes an effective way to fundamentally solve the problem of inadequate and overly expensive medical services by using information technology to change the existing medical service mode and improve the utilization of medical resources.

Figure 1 depicts the convergence architecture of smart healthcare. Due to the inherent nature of a wireless medium, sensors and link failures often occur because of energy depletion, channel fading, faulty configuration, malicious attacks, etc., in the harsh emergency ward environment. A traditional fault-tolerant scheme with a resource-hungry backup path would require a large amount of redundancy. Additionally, traditional reactive fault-tolerant mechanisms provide protection only after errors occur, resulting in longer delays. Moreover, taking into account the complexity of the smart healthcare environment, previous schemes cannot be applied directly to healthcare monitoring systems. Therefore, this article describes the design and implementation of a proactive, reliable data transmission mechanism with fault-tolerant capacity based on the principle of random linear network coding. This fault-tolerant scheme includes two aspects: the grouping of greedy nodes and the construction of coding trees. As long as the gateway receives a sufficient number of linear independent encoding combinations, original packets can be recovered quickly, even when some errors occur. Numerical results have shown that this resilience transmission mechanism works better than previous methods in increasing the ratio of successful deliveries and the effective throughput rate, and in reducing the degree of resource redundancy and end-to-end delay.

The remainder of this article is organised as follows. Research progress in fault-tolerance is reviewed in Section 2. The network model is then formulated in Section 3. In Section 4, details are presented of how we can generate a spanning tree based on network coding theory. The source-grouping strategy is described in Section 5. Simulation results are discussed in Section 6. Finally, we conclude the paper in Section 7.

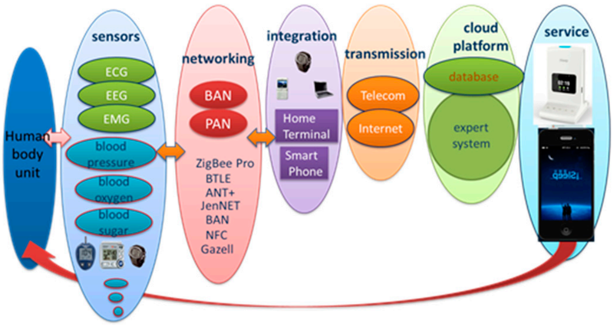


Figure 1. The convergence architecture of smart healthcare.

2. Related Work

During the past few decades, there has been much research on the fault-tolerant problem in network optimization and algorithm design. How to provide a fault-tolerant guarantee is one of the most important challenges. Researchers have drawn attention to this active field and proposed several solutions.

Sterbenz et al. in [3] explored the resilience problem in current communication networks and systematically designed an Internet architecture with survivability as a design consideration. In [4], the authors presented a flow assignment method for computer network design by combining routing and survivability aspects. The authors of [5] investigated the problem of a base station failure in a wireless cellular mobile communication system. In [6], the effects of failures on user mobility and survivability issues in phone/personal communications services networks in a wireless environment were addressed. The authors in [7] took survivability requirements and capacity constraints in wireless access networks into account and designed multi-period optimization. Reddy et al. in [8] designed a novel routing protocol to solve the survivability protection problem against multi-path failures within a minimum packet redundancy restriction. References [9–13] focus on fault-tolerant issues while considering node placement, topology control, and the routing algorithm. Regarding the design of survivability mechanisms in wireless multi-hop networks, researchers have conducted exploratory work with various optimization objectives during the past few years. The authors in [14] studied the maximal network-covering problem in wireless networks. Using binary integer programming methods, the authors solved the maximal-covering problem and offered reliable services provisioning, meeting the network connectivity requirement. This approach was expected to help network planners to deploy Wireless Multi-hop Networks (WMNs) with lower installation costs. The authors in [15] designed a novel scheme to establish a reliable network infrastructure based on the ear decomposition method. The generated topology can guarantee full coverage to all mesh clients and tolerate an error in a single mesh node. The authors of [16] studied the issue of mesh gateway deployment and network topology control, presenting two schemes to address the requirements for delay-tolerance and survivability in backbone wireless mesh networks. The network survival mechanisms in [16] can offer network planners a number of feasible compromise solutions. However, these static proactive network survivability mechanisms lack good flexibility and adaptability. Bisti et al. in [17] presented a new routing protection algorithm to increase network survivability against node or link failures in a wireless multi-hop environment. The method in [17] permits network elements to react to local failures quickly using a proactive backup route. With the joint optimization of scheduling, routing, and channel assignment, the authors of [18] studied the network recovery problem in a multi-interface multi-channel wireless mesh scenario. Their scheme can adjust routing and channel assignments to avoid network congestion. To some degree, these reactive schemes can recover the fault path to ensure packet transmission is uninterrupted. However, longer delays prevent these schemes from being practical.

Al-Kofahi et al. in [19] addressed the survivability problem under a many-to-one traffic pattern and provided sufficient theoretical proof and analysis. They also proposed a network protection mechanism based on network coding to overcome the inadequacies of previous methods. In [20], Misra et al. studied the fault-tolerance routing problem and developed a learning-automaton-based adaptive fault-tolerant routing scheme in an IOT environment. To improve bandwidth efficiency, Wang et al. in [21] designed a network-coding-based flow allocation and rate-control scheme to tackle the multicast optimization problem for multimedia services in an IOT environment. In [22], Qiu et al. investigated the survivability network topology design problem, and developed a novel model by analysing the roles of different network elements in a heterogeneous wireless sensor network scenario.

Otto et al. in [23] estimate the actual transmission performance of the WBAN prototype system, and analyzes the factors that may cause the network transmission reliability to decline. Latre et al. in [24] evaluated the reliability of the Cascading Information retrieval by Controlling Access with Distributed slot Assignment (CIADIA) protocol, and proposed the improved mechanism based on CIADIA to further improve transmission reliability. Zhou et al. in [25] studied the adaptive resource scheduling mechanism in WBAN to ensure reliable data transmission. Wu et al. in [26] proposed a channel reservation mechanism that can improve the reliability of transmission in a non-ideal WBAN channel environment. However, WBAN system [27] requirements for transmission reliability, energy efficiency, and latency are higher than any previous communication system. If only a single performance index is optimized, it will lead to other performance degradation, which is not conducive to the reliable and efficient transmission of medical services.

3. Network Model

Our network model in Figure 2 consists of three types of nodes: a gateway, sinks, and sensors in a Wireless Body Sensor Network (WBSN) for smart healthcare. We can use a unidirectional connected graph to represent the network topology. Here we use $G = (V, E)$ to indicate backbone network topology of a WBSN, where E refers to the set of wireless links, and V refers to the set of network elements, including a gateway, sinks, and sensor nodes. Network traffic generated from rock-bottom sensors will converge the wireless gateway T. The gateway can be connected to the external wired Internet. In our network model, we mainly investigate a many-to-one traffic model from various sensor nodes to an external gateway. Depending on the number of hops to the gateway node, sink nodes are layered in many levels. It is assumed that nodes in WBSNs are equipped with multi-interface multi-channel functions. In this way, every network element can exchange information with its neighbours at the same time. We assume that every sensor user U_k must be connected to at least two sink nodes L_k for connectivity. We indicate all sensor elements by U_s, in which $U_s = \{S_1, S_2, ..., S_n\}, U_s \subset V, |U_s| = n$. We assume the min-cut between L_s and T to be h. This assumption means that the degree of connection between sink nodes and gateway nodes is very rich. It is assumed that all native or encoding packets are the same size for encoding efficiency. It is assumed that gateways can perform a decoding operation to release original information. According to the well-known Gauss Elimination (GE) method [28], destination nodes have a higher probability of recovering their desired packets by solving groups of linear equations, if the information they receive is sufficient for GE solutions.

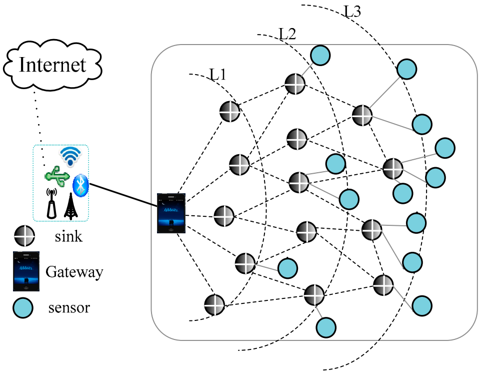


Figure 2. Network model.

4. Design of Network Coding Tree Algorithm

In our model, it is assumed that all sink nodes can encode and decode packets freely, while the rock-bottom sensor nodes cannot. When a particular sink node receives original packets from associated sensor nodes, it attempts to encode the native information into a single message. The sink node will then deliver those encoded messages. An elastic transmission mechanism following the principle of random linear coding [29,30] is presented to ensure that the entire sink node information encoding will be not linearly dependent. This fault-tolerant mechanism consists of two parts: the construction of a network coding tree and the design of a greedy grouping algorithm.

The construction of the spanning tree used for illustration of the coding relationship focuses on generating a tree that can associate all rock-bottom sensor nodes U_s, while all leaf nodes on the tree are located among their associated sink nodes L_s. With this method, destination nodes can construct linearly independent combinations according to the connection relationships of network nodes. The five-step algorithm to establish the spanning tree is given in detail below.

In the first step, a depth-first search (DFS) or breadth-first search (BFS) strategy can be employed to scan the full logical topology established by a WSN. In this way, a spanning tree can be produced with the root node deployed at node L_s.

In the second step, it is assumed that the leaf elements in this spanning tree include some sensor node U_s. If there exist some sensors which are leaf nodes in coding tree, these nodes are denoted as u.

In the third step, the algorithm constantly scans the entire network topology until it finds a neighbour node L_s of u. We indicate this sink node as x. Note that node u cannot be a child node of node x.

In the fourth step, the algorithm pollards this coding spanning tree to ensure its leaf elements do not include any U_s elements. In the fifth step, the algorithm continues to pollard unnecessary L_s nodes if leaf nodes contain any U_s elements.

The fourth step of this algorithm can benefit from a spanning-tree construction proposed in the literature [31,32]. It can be proven that the polynomial time-complexity analysis of establishing this spanning tree will be $O(|V|)$, while the time complexities of both the BFS and DFS algorithms are $O(|E|)$. The pseudo-code of the network coding algorithm is given in Table 1 below.

Table 1. Pseudo-Code of Network Coding Tree Algorithm.

Algorithm for Constructing Coding Tree

```
1: Use the DFS (depth-first search) or BFS (breadth-first search) algorithm to search the entire network
topology to generate a node-rooted tree rooted at $L_s$.
2: If there exist sensors which are leaf nodes in tree, denote it as $u$.
3: End if
4: Keep on search until finding one $u'$ neighbor node who is not $u'$ parent node, denote it as $x$.
5: connect the node $u$ and $x$ to construct a ring, denote it as $C$.
6: Traverse all the nodes in ring $C$ until finding one sensor v whose $L_s$ neighbor node w in not in ring $C$.
7: IF node $u$ and v are already connected, then Cut the loop directly in front or behind $v$.
8: Else
9: Cut the ring on both sides of v, Repeat steps 4–6 until tree trimming ends.
10: End if
11: keep on Pruning trees until There is no leaf node in the tree.
12: The algorithm ends.
```

In order to better explain the implementation of the coding tree, below is a concrete implementation example. The example network topology is shown in Figure 3. Figure 4a depicts the result of DFS search with C as the root node. Using the steps 2–6, we can get the result in Figure 4b. Figure 4c,d depict cutting off the corresponding edge and trimming off excess leaf nodes.

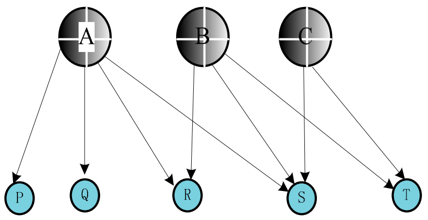


Figure 3. Example network topology for coding tree.

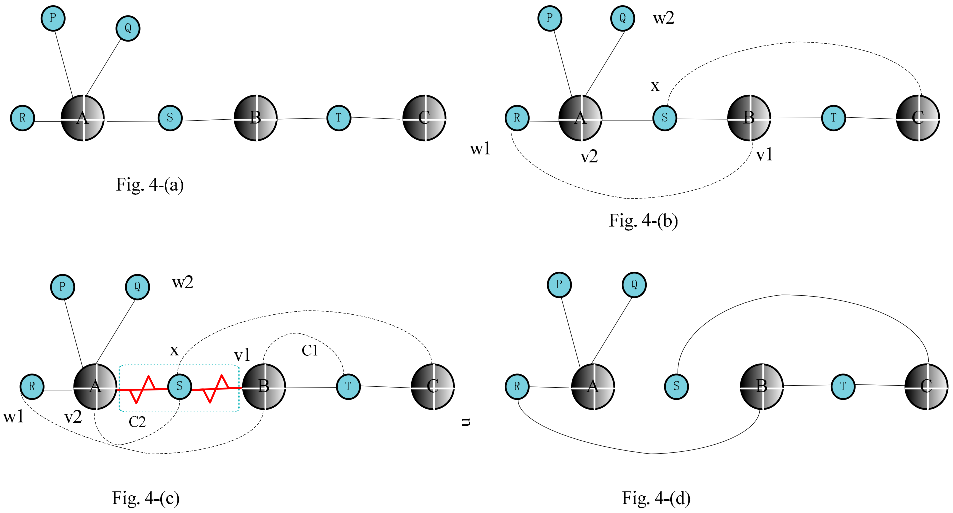


Fig. 4-(a)

Fig. 4-(b)

Fig. 4-(c)

Fig. 4-(d)

Figure 4. Coding tree algorithm implementation process. (**a**) Tree-construction; (**b**) Result of two iterations; (**c**) Result of tree-modification; (**d**) Tree establishment.

Here, it is noted that the condition under which this coding tree algorithm can work well must satisfy $n \prec h$. This condition requires that the full scale of sensor elements in the coding tree must be not more than the scale of their associated sink nodes. However, in actual WSNs, one sink node often associates with multiple sensor elements. If the full scale of source sensors is larger than the full scale of encoded sinks ($n \geq h$), the construction of the network spanning tree following the preset perfect full logical topology would no longer work well. In this situation, the elastic transmission mechanism must follow another grouping strategy to meet the original conditions. This greedy grouping strategy can divide the n sensor nodes, covered by h sink nodes, into groups and ensure that the full network topology constructed by the grouping strategy can continue to employ the coding tree algorithm in Section 4.

The key data structures and their functions are summarised in Table 2.

Table 2. Data Structure and Function of Simulation Program.

Names of Data Structures	Functions
FindTree	Look for DFS_tree
ModTree	Prune DFS_tree and allocate encoding coefficient
Greedy_Group	Logically group the topology that does not meet the coding algorithm condition
GaloisField	Galois field, where encoding is done
nc_node	Define network coding node class
nc_generate_xy	Generate the node coordinates, by which connectivity degree of nodes can be determine
test_connectivity	Measure the connectivity of network topology
nc_independence	Judge whether the received encoding combinations is linearly independent
nc_peform_transmission	Select the node to send packets, and do the encoding operation
nc_update	Judge whether the received encoding combinations is linearly independent, and done the decoding operation

5. Design of Greedy Grouping Algorithm

Without loss of generality, this paper considers the case of a multi-channel multi-interface WSN with a single gateway. The network topology is layered, based on the hops between each sink node and the gateway. Each level of sinks is associated with a group of sensors that continuously monitor mining status and periodically collect data in the smart healthcare environment. It is assumed that L_i and U_i respectively denote the ith level of the sink and sensor sets. Sink nodes normally converse over two orthogonal radio channels. One channel communicates between the sinks to construct the underlying infrastructure, while the other is responsible for communications between the sinks and sensors. This ensures sinks and sensors work simultaneously without mutual interference. The levels of sink nodes occupy the wireless channel using Time Division Multiple Address (TDMA). Each layer of sink nodes is allocated a special time slot for packet transmission. This means that data transmission in each level occurs at the same time. Within each time slot, sensors first deliver data to the relevant level of sinks; the sinks then begin to re-transmit the corresponding data. Typically, each set of sensors and its associated level of sinks is active only during specific time slots assigned to it. The n source nodes represent sensor nodes in the active layer in our network model. If each sensor generates a separate data unit, there will be a total of n data units (from source nodes) that need to be forwarded to the gateway. Here, the binary linear network coding technique is employed to effectively prevent a link failure phenomenon. In our proposal, the minimum number of paths is adopted to provide as much fault-tolerance as possible.

If network latency t is taken into account, the output packet combination can be formulated as in Formula (1):

$$Y_j(t+1) = \sum_{i=1}^{L} g_{ij} X_i(t) + \sum f_j Y(j)(t) \tag{1}$$

In the network model, it is assumed that sensor nodes collect status information periodically. The sink nodes preserve sufficient storage space to accommodate native packets, to encode these original packets and to transmit them. For simplicity, we ignore the delay caused by the encoding operation in our network model. Therefore, here we can calculate the simplified representation of output packets in Formula (2) as follows:

$$Y_j = \sum_{i=1}^{L} g_{ij} X_i \tag{2}$$

For random linear network coding, in Formulas (1) and (2), X_i and Y_j represent the input and output packets, where i and j denote the incoming and output links, respectively. In particular, we use g_{ij} to represent the coding vector generated randomly, L to represent the number of incoming packets, and $Y(j)$ to represent the native packets generated by the sink nodes themselves.

Due to the harsh wireless communication environment in smart healthcare, packet corruption occurs often. Based on the principle of random linear network coding, coded packets are mathematically random linear combinations of the native packets. Based on the GE theory, destination nodes can recover corrupted packets with higher probability. Note that the encoding vector g_{ij} must be recorded in the packet header to assist the destination nodes in the decoding process. Because of the broadcast nature of wireless media, sink nodes can overhear the encoding combinations generated by neighbours. They can then encode them together and forward them. In random linear network coding for our proposal, the coding node encodes the incoming packets with the global encoding vector embedded in the header. The extra fields (or bytes) in messages, as well as storage overhead for coding functions, can also incur a few expenses, compared with traditional methods. Here, note the coding conditions in our proposal target WSNs with omni-directional antenna, in which the sensor and sink nodes are resource-constrained. This means that the memory requirements must be slightly higher than a delay-bandwidth product. In addition, our scheme uses a dynamic cache update scheme, which periodically scans expired packets, so the cache space is not very large. While a sink node can overhear the encoding combinations generated by every sink node in the WBAN, this can create a lot of coding opportunities for sinks. At the expense of this overhead, the proposed fault-tolerant mechanism performs better than traditional schemes in reducing packet loss ratio, end-to-end delay, and resource redundancy degree, and in increasing useful throughput ratio. Therefore, it is worthwhile to pay a small amount of overhead.

A set of sensor nodes that dispatch messages can be considered a group. We can assign a label to a group with a sequence number ID. The decoding process will begin when the destination nodes have received enough native messages with the same sequence number ID. In random linear network coding, the packet information and encoding vectors are mixed at each coding node according to a local coding method. Therefore, the destination node does not need to obtain the knowledge of the whole network topology and global encoding information to recover the native packets. If the rank of matrix made up of global encoding vectors is equal to the number of native packets in each round, the native packets can be released at last. Therefore, as long as the number of linearly independent packets received by destination node is equal to or larger than the number of native packets, all of the native packets can be recovered with high probability through the GE method for solving linear equations over finite fields.

The greedy grouping will first select the candidate sensors with the largest number of neighbour nodes and put them into a grouping labelled as group 1. This selection method can ensure that many connections with L_s nodes can be kept to produce more coding opportunities. The selection method continues putting sensors into this group with the same sequence number until the accumulated degree of this set is not equal to $h-1$. When group 1 is complete, a subsequent series of groups will be constructed in sequence using this grouping strategy until all sensors are covered, when the greedy grouping algorithm ends. This greedy grouping strategy chooses the source sensors with the most neighbour nodes at every grouping stage. In this way, our greedy algorithm particularly tries to create coding opportunities without regarding other factors that may influence network performance. Therefore, this grouping scheme satisfies the optimal requirement from the local perspective. Globally, however, the optimal grouping combinations are not necessarily achieved [33,34].The pseudo-code of the network coding algorithm is given in Table 3 below.

Table 3. Pseudo-Code of Greedy Grouping Algorithm.

Algorithm for Constructing Greedy Grouping

```
Define the variable
Stores the scheduled source node, $SchdSet = \phi$;
Initialize the encoding group number, $coding\_lable = 0$;
Initialize the cluster number, $cluster\_id = 0$;
Start grouping
Initialize source node index in group, $index = 0$;
Calculate the minimum cut $h$ between $L_s$ and destination node
Define a boolean variable Found, If the source node joins the current cluster, it is true, $Found = TRUE$;
$While(|SchdSet|<|U_s|)\ \ do$
    $x = \phi$;
    $if(index > h-2||Found == FALSE)\ \ then$
        $cluster\_id++$;
        $index++$;
    End if
    $coding\_lable++$;
    x={choose the sensor u with maximum connectivity degree}, $u \notin SchdSet$
    $if(x == \phi)\ \ then$
        $Found = FALSE$;
    Else
        $SchdSet = SchdSet \cup x$;
        $Schedule[cluster\_id][index] = x$;
        Complete a node grouping, jump to the next one
    End if
End While
```

Finally, the destination nodes will judge if they have received k linearly independent packet combinations. If so, the k original packets can be solved with the GE method. It can be proven that the computational complexity of this greedy grouping algorithm is (k^3). The proof is omitted here because of space. If the wireless channel environment is too unfavourable, the destination nodes will not receive sufficient packets to satisfy the GE condition. In this harsh environment, the source nodes will start there transmission of lost packets. If the destination nodes do not receive enough required information, they will request their neighbour nodes to re-transmit the associated information. The neighbour nodes will also continue to forward lost packets until the GE solution condition is satisfied. For simplicity, it is assumed here that a one-hop retransmission will usually satisfy the GE solution requirement.

In order to better explain the grouping process, Figure 5 gives an example. In Figure 5, there are four source nodes: S1, S2, S3, and S4, and the maximum number of link-disjoint paths between the sinks and the gateway is three. Therefore, the maximum number of source nodes in feasible grouping is 2. Using the greedy grouping algorithm in Table 3, we can derive an output result SET1 = {{S1,S2}, {S3,S4}}. In contrast to the other two grouping results SET2 = {{S1,S2}, {S1,S3}, {S1,S4}} and SET3 = {{S1,S3}, {S2,S4}}, we found that the optimal solution is SET1. Because this scheme obtains a fairness index more than SET2, and consumes less bandwidth resources than SET3. In this greedy strategy, each packet uses only three links to forward data to the minimum cutting edge, while four links in SET3.

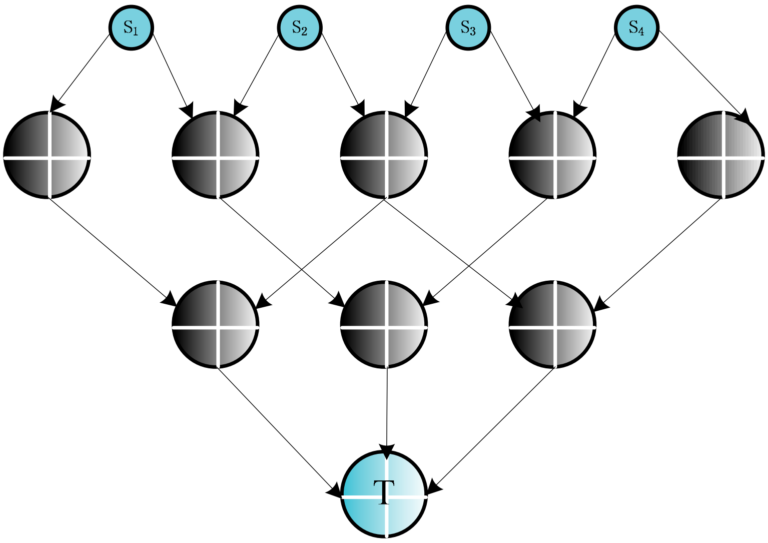


Figure 5. Illustration for grouping algorithm.

6. Numerical Results and Analysis

Extensive simulations have been conducted to compare the transmission reliability and efficiency, resource consumption, and latency performance in the elastic transmission mechanism [35,36] with a resource redundancy protection mechanism and backup path protection mechanism for validation of its effectiveness. We defined four reliability and efficiency performance indicators that we measured by continuing to increase the network traffic and link failure probability. We built a simulation framework in the Windows-environment-based C++language, with the network topology depicted in Figure 6.

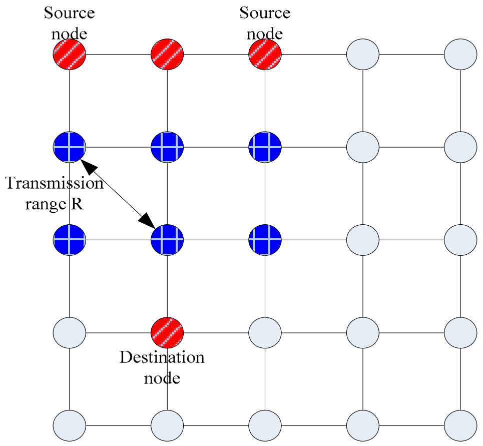
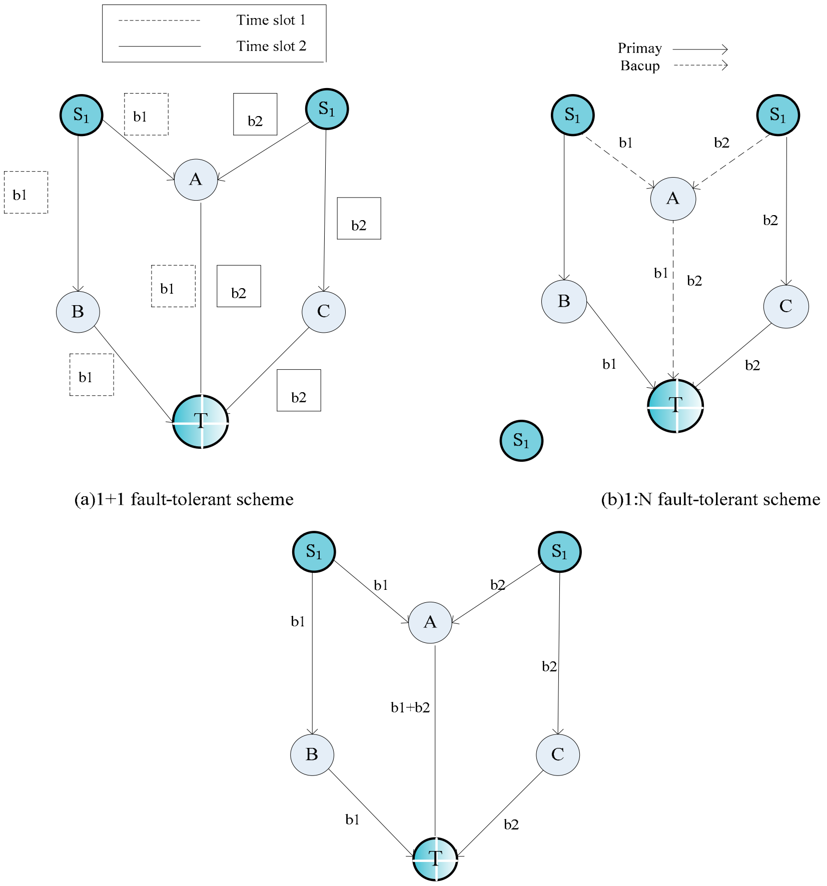


Figure 6. Network topology for simulation experiments.

Figure 7 shows the data transfer arrow signs of three methods, in which the source nodes S1 and S2 need to transmit the packets b1 and b2 to the destination node *T*, respectively.



(a)1+1 fault-tolerant scheme

(b)1:N fault-tolerant scheme

(c)Network coding based fault-tolerant mechanism

Figure 7. Packet transmission for three fault-tolerance schemes.

In the process of simulation, it is assumed that the probability of link failure is independent at each hop. The source node sends a steady stream of traffic to the destination node. When each generation of packets is received by destination nodes, they record packet information including packet size, time stamp, encoding operation, decoding operation, and other statistics. The sensor and sink nodes also calculate the transmitting times of packets.

In our simulation configuration, we adopt the following method: the traffic flow is established between each pair of source and destination nodes. This traffic flow would generate new packets according to the assumed statistical distribution (e.g., Poisson distribution), meanwhile, it would specify the size of packet and the average rate of generating packets. In this way, we can configure the dynamic packets. The main simulation parameter values are listed in Table 4 below.

Table 4. Simulation Parameters Settings.

Parameters Value
Attenuation model two ray
Channel capacity 2 Mbit/s
Signal transmission range 250 m
Signal interference range 550 m
Packet size 512 Byte
Output queue type FIFO (First-In First-Out)
Cache capacity 50 packets
energy model generic radio energy model
Simulation area 1000 m × 1000 m

A. Performance Indicators

In our simulation experiments, we adopted four performance indicators:

- Successful delivery ratio

We define the Successful Delivery Ratio (*SDR*) as the percentage of network traffic received at destination nodes relative to the network traffic generated at sources. We use this *SDR* indicator to weigh the probability of successful packet transmission. A larger *SDR* means a higher probability of successful delivery. The *SDR* is computed using Formula (3):

$$SDR = \frac{\sum_{d_k \in (S,D)} P_r}{\sum_{s_k \in (S,D)} P_s} \tag{3}$$

- Resource redundancy degree

We define the Resource Redundancy Degree (*RRD*) as the percentage of redundant network traffic relative to the network traffic generated at sources. These copied packets consist of two parts: duplicates and linearly dependent packets. This *RRD* indicator can be used to measure the cost of the fault-tolerant mechanism. The *RRD* can be computed using Formula (4):

$$RRD = \frac{\sum\limits_{i=1}^{n} R^{i}_{packet\_useless}}{\sum\limits_{j=1}^{n} R^{j}_{packet\_emitted}} \tag{4}$$

- End-to-end delay

We define End-to-end delay (*ED*) as the sum of the latencies for every message received by the destination nodes. This *ED* indicator can be used to weigh the timeliness of the fault-tolerant mechanism. Because the network coding operation incurs some latency, the *ED* indicator consists of four parts: encoding latency $d_{encoding}$, decoding latency $d_{decoding}$, transmission latency $d_{transmission}$, and waiting latency d_{wait}. The ED can be calculated using Formulas (5) and (6):

$$ED = \frac{1}{n}\sum_{j=1}^{n} D_j \tag{5}$$

$$D = d_{wait} + d_{transmission} + d_{encoding} + d_{decoding} \tag{6}$$

- Useful throughput ratio

We define the Useful Throughput Ratio (UTR) as the percentage of the scale of effective network information relative to the scale of full network information received by destination nodes. We use this UTR indicator to weigh the transmission efficiency of the fault-tolerant mechanism. Therefore, a larger UTR value means more effective delivery. This UTR indicator can be computed using Formula (7):

$$UTR = \frac{\sum_{d_i \in (S,D)} P_{useful}}{\sum_{d_j \in (S,D)} P_{received}} \tag{7}$$

B. Simulation Results and Analysis

To compare the advantages and disadvantages of the DFS and BFS schemes, some simulation experiments were conducted employing these two search algorithms in the network coding tree algorithm. The range of sink nodes in WSNs could be changed between $|U_s| + 1$ and $2\ |U_s|$. Additionally, we continued to increase the number of sensor nodes, keeping the number of sink nodes at twice as many. The experiments were executed 100 times to obtain average values. Figure 8 depicts the number of sink nodes needed while increasing the number of sensor nodes. We observe that the number of leaves generated using the BFS method is larger than that generated by the DFS method. Although global optimization was not achieved, the introduction of the BFS search algorithm was more beneficial than DFS in generating more leaf combinations in the network coding tree.

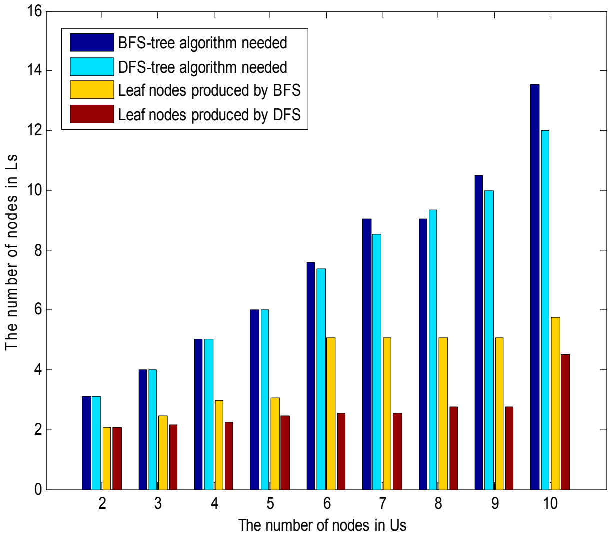


Figure 8. Comparison of the total number of generated leaves nodes between breadth-first search (BFS) and depth-first search (DFS) algorithms.

Figure 9 presents the probability of successful transmission as link failure probability changes. It can be observed that the SDR values of these three fault-tolerant mechanisms shrink as the link failure probability increases. In particular, the SDR values drop abruptly to zero when the link failure probability exceeds 0.3. This is because the NCFM, 1:N and 1 + 1 fault-tolerant mechanisms cannot receive correct packets in harsh channel conditions when the maximum number of retransmissions is reached. In addition, we observe that the *SDR* of the 1 + 1 scheme is more or less the same as that of the 1:N scheme. This is because we reserve just one backup path to retransmit the lost packet in the 1:N scheme; the difference between the 1 + 1 and 1:N schemes can be shown from the analysis of *RRD* and *ED*. Therefore, we conclude that the fault-tolerant capacity of the NCFM algorithm is stronger than those of the 1 + 1 and 1:N schemes. Additionally, the NCFM algorithm has an evident advantage over 1:N and 1 + 1 when the link failure probability is lower than 0.3.

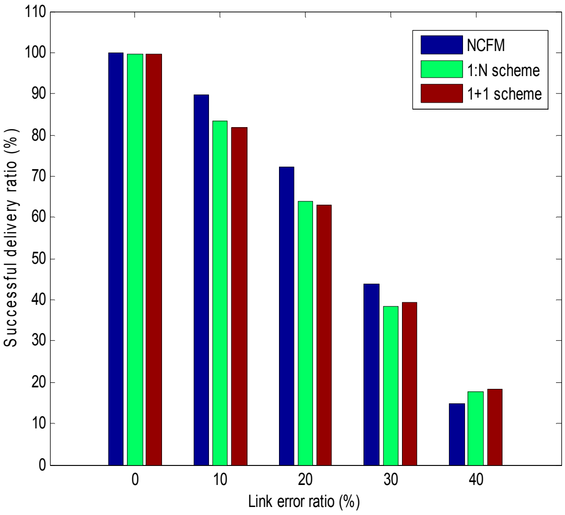


Figure 9. Comparison of successful delivery ratios.

Figure 10 presents the average latency performance analysis for the three fault-tolerant mechanisms as the link failure probability increases. The end-to-end delay curve of the elastic transmission mechanism varies with those of the 1:N and 1 + 1 protection mechanisms if the link failure probability is zero in perfect channel conditions. When the link failure probability rises, the average latency curve of the elastic transmission mechanism will become better than those of the 1:N and 1 + 1 protection mechanisms. This is because the random linear network coding operation can reduce the number of packet transmissions or retransmission times as much as possible when the maximum transmissions bottleneck is reached. Here it is assumed that a node can finish only one packet delivery each time. The 1:N fault-tolerant mechanism must continue to switch the backup path for the retransmission of lost packets, which results in longer *ED*. For the 1 + 1 scheme, packet delivery will be postponed when transmissions fail on both the working and backup paths. In particular, the average latency curves of all schemes will rise rapidly if the wireless link quality degrades. In this situation, it is likely that the destination node cannot receive sufficient packets for decoding with the NCFM algorithm. Therefore, the *ED* performance will fail when the link failure probability is greater than 40%.

Figure 11 shows the average end-to-end delay analysis for the three fault-tolerant mechanisms as the network traffic continues to increase. The simulation experiments were conducted assuming that the link failure probability was 8%. As increased network traffic is injected into the WSN, the average end-to-end delay of the three fault-tolerant mechanisms shows an upward trend. As the number of source packets continues to increase, the *ED* benefit achieved by NCFM becomes gradually evident. The reason is that NCFM exploits the encoding function to cut down the wireless resource redundancy with more information. Under some *SDR* levels, the destination can still receive sufficient packets to restore the original packets. Therefore, this proactive NCFM algorithm outperforms the 1:N and 1 + 1 protection mechanisms. The 1:N method in lossy networks can guarantee the reliability of data transmission to some degree, but the latency caused by path switching and packet retransmission is longer than those of the other two fault-tolerant mechanisms. Therefore, the conclusion can be drawn that the 1:N fault-tolerant mechanism is not suitable if network users have strict real-time demands. The proactive NCFM method can make full use of limited bandwidth resources to supply fast recovery for delay-sensitive traffic.

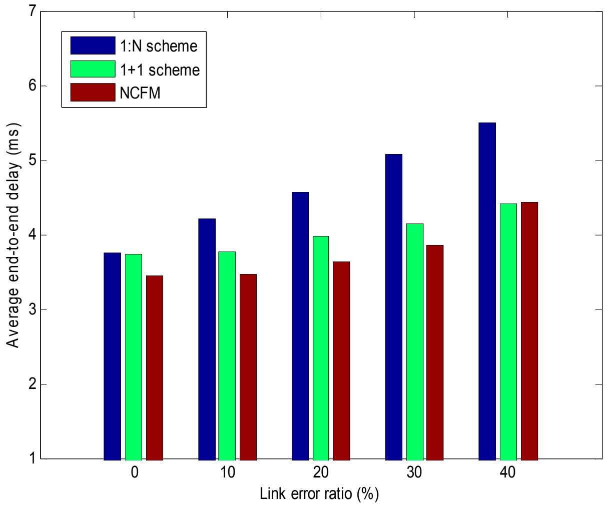


Figure 10. Comparison of average end-to-end delay.

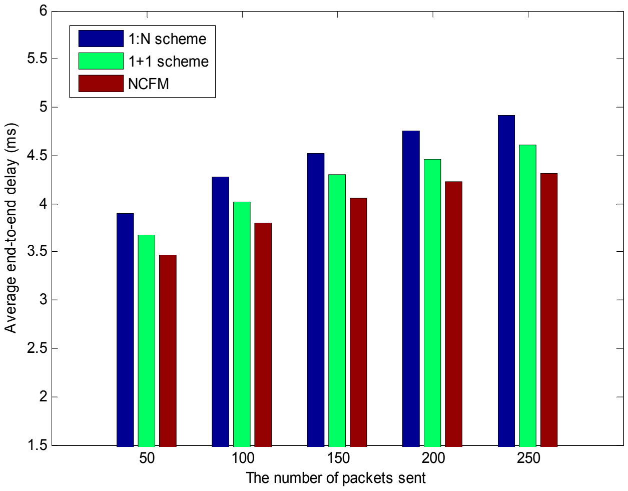


Figure 11. Comparison of ED with the number of packets sent forNetwork Coding-based Fault-tolerant Mechanism (NCFM), 1 + 1 and 1:N.

Figure 12 displays the analysis of the degrees of resource redundancy for the three fault-tolerant mechanisms when the link failure probability is increased. As the link failure probability increases, the degrees of resource redundancy continue to rise, as shown in Figure 12. With the deterioration of channel conditions, packet loss frequently occurs in certain nodes. Partial nodes need to resend the missing packets to ensure that all destination nodes can receive their packets error-free. A large number of retransmissions can cause many redundant packets in WSNs. As depicted in Figure 12, the cost of an elastic transmission mechanism and resource redundancy protection mechanism in terms of the degree of resource redundancy seems to be less than that of a backup path protection mechanism. This is because many different packets are encouraged to mix together into one packet before transmission. This operation can decrease the number of packet transmissions or retransmissions as much as possible, and improve wireless resource utilization. Although some packets are lost, the destination can still exploit the received encoded packets to recover the original packets without retransmission. In particular, the *RRD* of the 1:N method rises sharply after the link error rate reaches 0.1. This is because poor channel quality causes much retransmission, resulting in network congestion, which incurs many more redundant packets. To a certain extent, *RRD* is similar to the energy consumption efficiency. We can conclude that the elastic transmission mechanism is more economical than the 1 + 1 and 1:N mechanisms. Although NCFM increases the data processing burden of nodes, energy consumption in the computer hardware is much smaller than that needed to process packet transmission, so this part of energy consumption can be ignored.

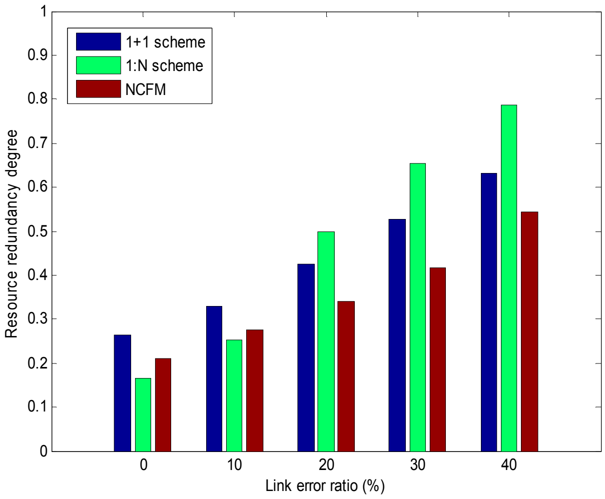


Figure 12. Comparison of resource redundancy degrees.

Figure 13 presents an analysis of the degree of resource redundancy for the three fault-tolerant mechanisms as the network traffic continues to increase. The simulation experiments ran with a link failure probability of 10%. As the injected network traffic grows, the *RRD*s of the three fault-tolerant mechanisms trend steadily upward. When the link failure probability is relatively low, it can be observed that the expense of NCFM in the degree of resource redundancy is smaller than that of the 1:N and 1 + 1 protection mechanisms. The reason for this is similar to the analysis for Figure 12. Therefore, we can conclude that for identical link error rates, the NCFM is the most economical fault-tolerant scheme for network resources.

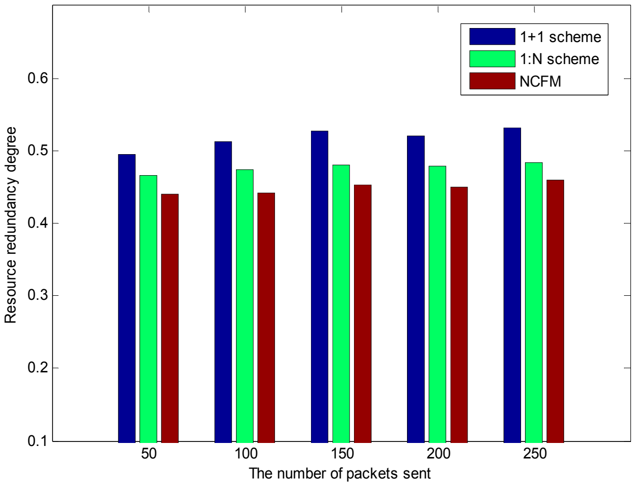


Figure 13. Comparison of RRD with the number of packets sent for NCFM, 1 + 1 and 1:N.

Figure 14 shows the analysis of the useful throughput ratio for the three fault-tolerant mechanisms. The simulations were conducted with a link failure probability of 8%. It can be observed that the useful throughput ratio performance of NCFM is much better than that of the 1:N and 1 + 1 protection mechanisms. From the comparison results in Figure 14, it can be concluded that NCFM can proactively offer fault-tolerant functions for native packets only if the network coding condition is satisfied. Compared with the traditional 1:N and 1 + 1 fault-tolerant mechanisms, we observe that NCFM has higher transmission efficiency, lower network expenses, and a more reliable fault-tolerant capacity.

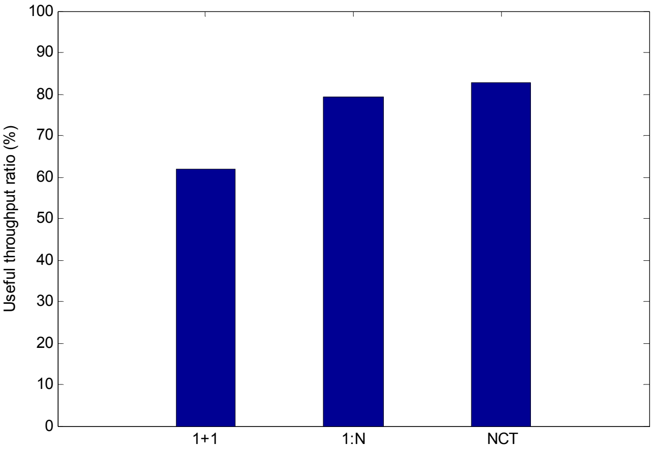


Figure 14. Comparison of useful throughput ratio.

7. Conclusions

To tackle the fault-tolerant problem in a many-to-one stream model in a smart healthcare monitoring scenario, we have designed a proactive resilient data transmission mechanism with the principle of random linear network coding. This fault-tolerant scheme includes two aspects: the greedy grouping of nodes and coding tree construction. As long as the gateway receives a sufficient number of linearly independent encoding combinations, original packets can be recovered quickly even when some errors occur. Additionally, a scalable simulation framework with encoding and decoding functions has been devised to verify the performance of three different fault-tolerant mechanisms. Based on this platform, a number of experiments have been conducted comparing transmission reliability, costs of resource consumption, latency performance, and transmission efficiency. Numerical statistics have shown that this elastic transmission mechanism outperforms traditional network protection mechanisms in improving the successful delivery ratio and the useful throughput ratio, and reduces the degree of resource redundancy and average end-to-end delay. The performance analysis offers some guidelines for effectively and efficiently designing survivability protection methods under different traffic patterns. In the future, this proactive network-coding-based elastic-transmission mechanism will be implemented in a real-time healthcare monitoring systems.

Acknowledgments: The preliminary work of this paper was presented at the International Conference on Computer Science and Software Engineering (ICCSSE) 2013. The work was supported by the National Natural Science Foundation of China (61501105, 61471109, 61471110), the National Key Research and Development Program of China (Grant No. 2016YFC0801607), the China Postdoctoral Science Foundation (2013M541243), the Fundamental Research Funds for the Central Universities (N150404018, N130304001, 150404015, N150401002, N161608001), and the Postdoctoral Science Foundation of Northeast University (20140319).

Author Contributions: Y.P. performed the experimental part, modelled, analyzed data and reported the results. X.W. and Y.W. contributed to the experimental part and result analysis. Q.D. and L.G. guided the whole research and supported the structure of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, C.; Lai, C.; Lai, Y.; Wu, Z.; Chao, H. An inferential real-time falling posture reconstruction for Internet of healthcare things. *J. Netw. Comput. Appl.* **2017**, *89*, 86–95. [CrossRef]
2. Ning, Z.; Xia, F.; Hu, X.; Chen, Z.; Obaidat, M. Social-oriented Adaptive Transmission in Opportunistic Internet of Smartphones. *IEEE Trans. Ind. Inform.* **2017**, *13*, 810–820. [CrossRef]
3. Ning, Z.; Xia, F.; Ullah, N.; Kong, X.; Hu, X. Vehicular Social Networks: Enabling Smart Mobility. *IEEE Commun. Mag.* **2017**, *55*, 49–55. [CrossRef]
4. Pierre, S.; Beaubrun, R. Integrating routing and survivability in fault-tolerant computer network design. *Comput. Commun.* **2000**, *23*, 317–327. [CrossRef]
5. Chu, K.-C.; Lin, F.Y.-S. Survivability and performance optimization of mobile wireless communication networks in the event of base station failure. *Comput. Electr. Eng.* **2006**, *32*, 50–64. [CrossRef]
6. Tipper, D.; Dahlberg, T.; Shin, H.; Charnsripinyo, C. Providing fault tolerance in wireless access networks. *IEEE Commun. Mag.* **2002**, *40*, 58–64. [CrossRef]
7. Bose, I.; Eryarsoy, E.; He, L. Multi-period design of survivable wireless access networks under capacity constraints. *Decis. Support Syst.* **2005**, *38*, 529–538. [CrossRef]
8. Reddy, T.; Sriram, S.; Manoj, B.S.; Murthy, C. MuSeQoR: Multi-path failure-tolerant security-aware QoS routing in ad hoc wireless networks. *Comput. Netw.* **2006**, *50*, 1349–1383. [CrossRef]
9. Qiu, T.; Zhao, A.; Xia, F.; Si, W.; Wu, D. ROSE: Robustness Strategy for Scale-Free Wireless Sensor Networks. *IEEE/ACM Trans. Netw.* **2017**, *1–16*. [CrossRef]
10. Chen, X.; Kim, Y.-A.; Wang, B.; Wei, W.; Shi, Z.; Song, Y. Fault-tolerant monitor placement for out-of-band wireless sensor network monitoring. *Ad Hoc Netw.* **2012**, *10*, 62–74. [CrossRef]
11. Boukerche, A.; NelemPazzi, R.W.; Araujo, R.B. Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments. *J. Parallel Distrib. Comput.* **2006**, *66*, 586–599. [CrossRef]
12. Ning, Z.; Liu, L.; Xia, F.; Jedari, B.; Lee, I.; Zhang, W. CAIS: A Copy Adjustable Incentive Scheme in Community-based Socially-Aware Networking. *IEEE Trans. Veh. Technol.* **2017**, *66*, 3406–3419. [CrossRef]
13. Qiu, T.; Chen, N.; Li, K.; Qiao, D.; Fu, Z. Heterogeneous ad hoc networks: Architectures, advances and challenges. *Ad Hoc Netw.* **2017**, *55*, 143–152. [CrossRef]
14. Lee, G.; Murray, A.T. Maximal covering with network survivability requirements in wireless mesh networks, Computers. *Environ. Urban Syst.* **2010**, *34*, 49–57. [CrossRef]
15. Benyamina, D.; Hafid, A.; Gendreau, M.; Maureira, J.C. On the design of reliable wireless mesh network infrastructure with QoS constraints. *Comput. Netw.* **2011**, *55*, 1631–1647. [CrossRef]
16. Hsu, C.-Y.; Wu, J.-L.C.; Wang, S.-T.; Hong, C.-Y. Survivable and delay-guaranteed backbone wireless mesh network design. *J. Parallel Distrib. Comput.* **2008**, *68*, 306–320. [CrossRef]
17. Bisti, L.; Lenzini, L.; Mingozzi, E.; Vallati, C.; Erta, A.; Malesci, U. Improved network resilience of wireless mesh networks using MPLS and Fast Re-Routing techniques. *Ad Hoc Netw.* **2011**, *9*, 1448–1460. [CrossRef]
18. Jiang, S.; Xue, Y. Providing survivability against jamming attack for multi-radio multi-channel wireless mesh networks. *J. Netw. Comput. Appl.* **2011**, *34*, 443–454. [CrossRef]
19. Al-Kofahi, O.; Kamal, A. Network coding-based protection of many-to-one wireless flows. *IEEE J. Sel. Areas Commun.* **2009**, *27*, 797–813. [CrossRef]
20. Misra, S.; Gupta, A.; Krishna, P.V.; Agarwal, H.; Obaidat, M.S. An adaptive learning approach for fault-tolerant routing in Internet of Things. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2012), Paris, France, 1–4 April 2012; pp. 815–819.
21. Wang, J.; Liu, Z.; Shen, Y.; Chen, H.; Zheng, L.; Qiu, H.; Shu, S. A distributed algorithm for inter-layer network coding-based multimedia multicast in Internet of Things. *Comput. Electr. Eng.* **2015**. [CrossRef]
22. Qiu, T.; Liu, X.; Feng, L.; Zhou, Y.; Zheng, K. An efficient tree-based self-organizing protocol for internet of things. *IEEE Access* **2016**, *4*, 3535–3546. [CrossRef]
23. Otto, C.; Milenkovic, A.; Sanders, C.; Jovanov, E. System architecture of a wireless body area sensor network for ubiquitous health monitoring. *J. Mob. Multimedia* **2006**, *1*, 307–326.
24. Latre, B.; Braem, B.; Moerman, I.; Blondia, C.; Reusens, E.; Joseph, W.; Demeester, P. A low-delay protocol for multi-hop wireless body area networks. In Proceedings of the Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services, Philadelphia, PA, USA, 6–10 August 2007; pp. 1–8.

25. Zhou, G.; Lu, J.; Wan, C.-Y.; Yarvis, M.D.; Stankovic, J.A. Body QoS: Adaptive and radio-agnostic QoS for body sensor networks. In Proceedings of the IEEE 27th Conference on Computer Communications (INFOCOM 2008), Phoenix, AZ, USA, 13–18 April 2008; pp. 565–573.
26. Wu, G.; Ren, J.; Xia, F.; Xu, Z. An adaptive fault-tolerant communication scheme for body sensor networks. *Sensors* **2010**, *10*, 9590–9608. [CrossRef] [PubMed]
27. Qiu, T.; Qiao, R.; Wu, D. EABS: An Event-Aware Backpressure Scheduling Scheme for Emergency Internet of Things. *IEEE Trans. Mob. Comput.* **2017**. [CrossRef]
28. Lipschutz, S.; Lipson, M. *Schaum's Outlines of Linear Algebra*, Tata McGraw-hill edition; The McGraw-Hill Companies, Inc.: Delhi, India, 2001; pp. 69–80.
29. Li, S.Y.; Yeung, R.W.; Cai, N. Linear network coding. *IEEE Trans. Inf. Theory* **2003**, *49*, 371–381. [CrossRef]
30. Ahlswede, R.; Cai, N.; Li, S.; Yeung, R. Network information flow. *IEEE Trans. Inf. Theory* **2000**, *46*, 1204–1216. [CrossRef]
31. Hou, W.; Ning, Z.; Guo, L. Temporal, Functional and Spatial Big Data Computing Framework for Large-Scale Smart Grid. *IEEE Trans. Emerg. Top. Comput.* **2017**. [CrossRef]
32. Guo, L.; Ning, Z.; Song, Q.; Huang, F.; Jamalipour, A. Joint Encoding and Grouping Multiple Node Pairs for Physical-Layer Network Coding with Low-Complexity Algorithm. *IEEE Trans. Veh. Technol.* **2017**. [CrossRef]
33. Qiu, T.; Liu, X.; Han, M.; Li, M.; Zhang, Y. SRTS: A Self-Recoverable Time Synchronization for Sensor Networks of Healthcare IoT. *Comput. Netw.* **2017**. [CrossRef]
34. Park, K.; Park, J.; Lee, J. An IoT System for Remote Monitoring of Patients at Home. *Appl. Sci.* **2017**, *7*. [CrossRef]
35. Aly, S.A.; Kamal, A.E.; Al-Kofahi, O.M. Network protection codes: Providing self-healing in autonomic networks using network coding. *Comput. Netw.* **2012**, *56*, 99–111. [CrossRef]
36. Rouayheb, S.; Sprintson, A.; Georghiades, C. Robust Network codes for unicast connections: A case study. *IEEE/ACM Trans. Netw.* **2011**, *19*, 644–656. [CrossRef]