ProNCP: A Proactive Network Coding based Protection Protocol

Qiao Xiang
Hongwei Zhang
Jianping Wang
Guoliang Xing

March 16, 2014

1 Preliminary

There has been a lot work done on protection against network failures in both wired and wireless networks. Existing protection techniques can be generally categorized into two classes: 1) proactive protection that sends the same data along two different paths simultaneously, which is also called 1+1 or 1+N protection, and 2) reactive protection that sends the data along one path at the beginning and switch to another path when there is a failure detected, which is also called 1:1 or 1:N protection. It is easy to see that both protection strategies have their own advantages and drawbacks. Proactive protection has zero response time when failures happen while having a higher transmission cost. Reactive protection has a lower transmission cost than proactive protection but requires failure detection mechanism and longer time to take actions.

Different from traditional wired networks, network failures in mission-critical wireless cyber-physical systems usually have the following characteristics:

1. Network failures in WCPS are usually transient (e.g., lower reliability in wireless transmission due to environment change), which means failed nodes and links can function normally after some time;

2. When transient failures happened in WCPS, it is usually not an efficient way to identify and replace the failed hardware because of both the transient nature of these failures and the extra high cost incurred by failure detection and correction operations.

Therefore, an important design principle in building a resilient mission-critical WCPS is to ensure efficient and fast data delivery in the presence of transient network failures by enabling proactive network protections. Making use of the broadcast nature of wireless communication, network coding has promising potentials in network protection because every coded packet contains the same amount of information entropy. Using network coding, every packet is basically equally useful when the destination retrieves the original information.

Recently, there has been some work on providing proactive protection using network coding in mesh networks [2] [8] [12]. However, most of the application scenarios for these work are in optical networks or require some specific routing structure to realize the protection scheme. Therefore, these work cannot be applied to the general scenarios of mission-critical cyber-physical systems. To cope with the requirement of reliable and real-time data delivery in mission-critical WCPS, we extend our solution to minimal cost NC- based routing in Chapter 3 to study the NC-based proactive protection problem in wireless sensor networks. The contribution of this study is as follows:

- We study the minimal cost 1+1 NC-based proactive protection problem. Different from the well-known minimal 2 node-disjoint path problem, we show that this new problem is NP-hard even in a simplified version through a reduction from the 2-partition problem. As a trivial note, we also point out and fix a mistake in the NP-hardness proof of the classic 2 integral network flow problem in [5].

- Motivated by the classic 2 node-disjoint path algorithm and Algorithm **??** we designed in Chapter 3, we propose a heuristic algorithm for the 1+1 NC-based proactive protection problem. This algorithm computes two node-disjoint braids that has a total transmission cost upper bounded by the 2 shortest node-disjoint paths.

- We further design and implement ProNCP, a proactive network coding based protection protocol, on TelosB sensor platforms. We evaluate the performance of ProNCP on our NetEye testbed by comparing it with a benchmark routing protocol (TNDP) that transmits data along 2 node-disjoint paths. Experiment results show that ProNCP performs better than TNDP in terms of reliability, transmission cost and goodput under both no-failure scenario and random transient failure scenarios.

The rest of this chapter is organized as follows: we first present the system model and problem definitions of this study. Then we study the complexity of 1+1 NC-based proactive protection problem and propose a heuristic algorithm. Based on this algorithm, we further implement ProNCP and evaluated its performance on the NetEye testbed. Before we conclude this chapter, we also discuss related work on proactive protection in wireless networks.

2 System model and problem definition

This study shares the similar system model and notations as in Chapter 3. We model a wireless network as a directed graph $G = (V, E)$ with node S as the source and T as the destination. For each node $i \in G$, we use U_i and D_i to denote the set of senders and receivers of i, respectively. And we denote the forwarder set of i as $FS_i \subset D_i$. For each link $i \to j \in E$, we denote ETX_{ij}^x as its expected number of transmission to deliver a packet with length x and $P_{ij}^x = \frac{1}{ETX_{ij}}$ as the corresponding link reliability. Since network coding will not change the packet length during the transmission, we use ETX_{ij} and P_{ij} for simplicity. Then we define $C_{iT}(x)$ as the transmission cost of delivering x linear independent packets from i to T, and $C_{iD_i}(x)$ as the expected number of broadcasts of node i when nodes in D_i collectively receive x linear independent coded packets from i. Assuming S needs to deliver K packets as a batch to T, we define K_i^j as the number of linear independent packets node i receives from node j.

Given a directed graph $G = (V, E)$ and K original packets to be delivered from S to T. We first define the 1+1 proactive protection problem with minimal transmission cost as follows:

Problem Q Given a directed graph $G = (V, E)$ with one source S and one destination T, find two node-disjoint NC-based routing braids B_1 and B_2 such that the total cost of delivering K linear independent packets to T along each braid is minimized.

The transmission objective of Problem **Q** is to deliver 2 copies of each piece of data generated by S to T, which is the same as the 2 node-disjoint path problem. However, the solution to the 2 node-disjoint path problem can only deal with single-node failures. On the contrary, the solution to Problem **Q** will be able to provide robust routing structure for sensor networks against up to F node failures, where $F = min(|V_{B_1}, V_{B_2}|) \geq 1$. Therefore, its solution can protect the network against random transient node failures.

3 1+1 NC-based proactive protection problem

In this section, we study the 1+1 network coding based proactive protection problem in detail. In traditional 1+1 protection schemes, the most common approach is to build 2 node-disjoint paths with the minimal total cost. This problem has been well studied and is solvable in polynomial time [14] [15] [3]. The basic idea of these algorithms is to make use of successive cycling cancellation methods in network flow theory. However, when network coding is introduced into wireless transmission, we will be able to further reduce the transmission cost for single data flow as we have proved in Chapter 3. Therefore, how to construct 2 node-disjoint routing braids with minimal total cost for NC-based transmission becomes an interesting and open problem. To propose the solution to this problem, we first explore its computation complexity.

3.1 Complexity study on problem $\mathbf{Q}$

Though constructing 2 node-disjoint paths with minimal cost for a single data flow can be solved efficiently for survivable networks. It is impossible to transplant the solution idea to construct 2 node-disjoint routing braids with minimal cost for NC-based transmission due to the following reasons:

- In NC-based transmission, the cost of the first hop broadcast does not follow the additive linear law as in traditional network flow theory;
- Routing braid has multiple paths at the second hop such that the traffic load on each path is dynamic depending on its order in the forwarder set instead of being static as in traditional network flow problems.

Towards better understanding the property of problem $\mathbf{Q}$, we study its computational complexity and propose the following theorem.

Theorem 1 *Problem* $\mathbf{Q}$ *is NP-hard.*

Proof To prove this theorem, we first look at Problem $\mathbf{Q}'$, a simpler version of Problem $\mathbf{Q}$ as follows:

Problem $\mathbf{Q}'$ The same as problem $\mathbf{Q}$ except that all the paths from S to T are node-disjoint to each other.

Since we are required to assign each non-terminal node to either braid B_1, braid B_2 or none of them. We are able to build a binary programming model for problem $\mathbf{Q}'$.

$$
\begin{aligned}
\text{Minimize: } C_1 + C_2 \quad = \quad & \frac{1}{1-\prod_{i=1}^{m}(1-x_i \cdot P_{2i-1})} \cdot \sum_{i=1}^{m} \frac{x_i \cdot P_{2i-1} \prod_{j=1}^{i-1}(1-x_j \cdot P_{2j-1})}{P_{2i}} \\
& + \frac{1}{1-\prod_{i=1}^{m}(1-y_i \cdot P_{2i-1})} \cdot \sum_{i=1}^{m} \frac{y_i \cdot P_{2i-1} \prod_{j=1}^{i-1}(1-y_j \cdot P_{2j-1})}{P_{2i}} \\
& + \max\{\frac{1}{1-\prod_{i=1}^{m}(1-x_i \cdot P_{2i-1})}, \frac{1}{1-\prod_{i=1}^{m}(1-y_i \cdot P_{2i-1})}\}
\end{aligned}
$$

such that

$$
\begin{aligned}
& x_i \in \{0,1\} \\
& y_i \in \{0,1\} \\
& x_i + y_i \leq 1 \\
& P_{2i} \geq P_{2(i+1)} \\
& 0 \leq P_{2i} \leq 1 \\
& 0 \leq P_{2i-1} \leq 1 \\
& \text{for } i = 1, 2, \ldots, m,
\end{aligned}
\tag{1}
$$

Although 0-1 programming is generally NP-hard, it does not necessarily result in the NP-hardness of this special class of 0-1 programming. To tackle this class of 0-1 programming, we propose the following lemma about the complexity of Problem $\mathbf{Q}'$:

Lemma 1 *Problem $\mathbf{Q}'$ is NP-hard.*

Proof We prove the NP-hardness of problem $\mathbf{Q}'$ via a reduction from the classic two-partition problem. There are different expressions of the 2-partition problem and we use the following optimization version:

Two-partition problem: Given a finite set A and a weight $w(a)$ for any element $a \in A$, partition set A into two subsets A_1 and A_2 such that the difference between $\sum_{a\in A_1} w(a)$ and $\sum_{b\in A_2} w(b)$ is minimized.

Without loss of generality, we assume that every element in the finite set of the two-partition problem has a positive weight. Given Y, an instance of the two-partition problem with set $X = \{X_1, X_2, \ldots, X_m\}$, we construct Z, an instance of problem $\mathbf{Q'}$ as follows. We first build a topology $S \to \{A_1, A_2, \ldots, A_m\} \to T$. For each $i = \{1, 2, \ldots, m\}$, we define $P_{SA_i} = 1 - 0.1^{w(X_i)}$ and $P_{A_iT} = 1$.

In this constructed instance of $\mathbf{Q'}$, it is straightforward to see that the objective function can be simplified to

$$C_1 + C_2 = 2 + \max\{\frac{1}{1 - \prod_{i=1}^{m}(1 - x_i P_{SA_i})}, \frac{1}{1 - \prod_{i=1}^{m}(1 - y_i P_{SA_i})}\} \tag{2}$$

To minimize Equation 2, the optimal solution must satisfy the following condition,

$$x_i + y_i = 1 \text{ for any } i \tag{3}$$

This means each node A_i must be either assigned to braid 1 or braid 2. This point can be proved through a simple contradiction. Suppose the optimal solution of Z has a node A_x not assigned braid 1 or braid 2. By assigning A_x to the braid that has a higher 1st hop broadcast cost, we can decrease this broadcast cost, which leads to a better solution to Z. Therefore, solving problem $\mathbf{Q'}$ is equivalent to solve the following problem:

$\mathbf{Q'}$ - Partition version: Partition set $\{A_1, A_2, \ldots, A_m\}$ into two subsets S_1 and S_2 such that the difference between $\prod^{A_i\in S_1} 1 - P_{SA_i}$ and $\prod^{A_j\in S_2} 1 - P_{SA_j}$ is minimized.

After a simple mathematical transformation, we can see that

$$\begin{aligned} \prod^{A_i\in S_1} 1 - P_{SA_i} &= 0.1^{\sum^{A_i\in S_1} w(X_i)} \\ \prod^{A_j\in S_2} 1 - P_{SA_j} &= 0.1^{\sum^{A_j\in S_2} w(X_j)} \end{aligned} \tag{4}$$

Through this equation it is readily to see that the partition version of **Z** is equivalent to Y, which means there is an optimal solution to **Z** if and only if there is an optimal solution to Y. From this we claim that there exists a one-to-one mapping from two-partition problem to Q'. Therefore problem Q' is NP-hard.

Having proved the NP-hardness of problem Q', the NP-hardness of problem Q is an immediate outcome.

Having proved Theorem 1, we show that it is impossible to develop a polynomial-time solution to even a simplified version of problem **Q**. This finding motivates us to design an efficient heuristic algorithm to compute good solutions to problem **Q**.

3.2 A finding in the NP-hardness proof for two-commodity integral flow problem

During our work in the complexity study on the problem of finding two node-disjoint routing braids with minimal cost, we find a technical mistake in the NP-hardness proof of two-commodity integral flow (TCIF) problem in the classic paper [5]. In this paper, the authors proposed a reduction from any instance of the satisfiability (SAT) problem to the TCIF problem. For any instance of A of the SAT problem, this paper denotes variables in A as $x_1, x_2, \ldots, x_n$ and the clauses in A as $C_1, C_2, \ldots, C_k$. For each variable x_i, p_i represents the number of positive occurrences of x_i and q_i represents the number of negative occurrences of x_i. A lobe L_i is then constructed for each x_i as shown in Figure 1. After connecting each lobe one by one and adding some extra nodes corresponding all the clauses in instance A. The authors proved that there exists an satisfiable assignment for A if and only if there exists two commodities of integral flow in the reduced instance of the TCIF problem.

However, this proof ignored the case when $p_i = 0$ or $q_i = 0$ for some x_i, which can affect the correctness of this proof. For example, if $p_i = 0$ for some x_i, the constructed lobe L_i has only the lower part. Under this case, when there is a satisfiable assignment which assigns $x_i = 0$ for the SAT instance A, the constructed TCIF instance cannot find two commodities

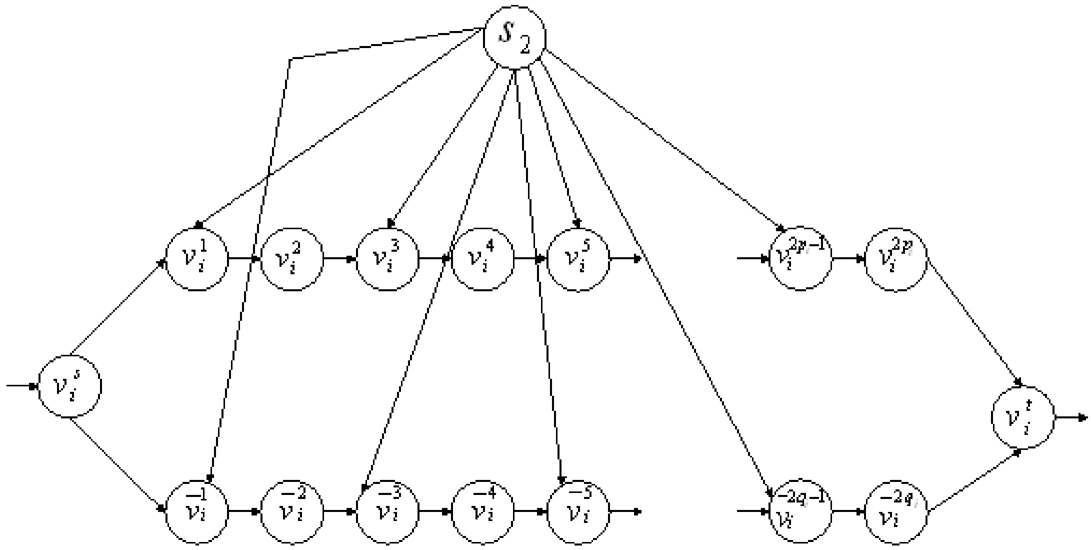

Figure 1: lobe i for variable x_i

of integral flow because each arc has only a capacity of 1 and lobe L_i cannot be used for two commodities of flow. Therefore, it is a lethal mistake for the whole proof.

Though this mistake invalids the whole correctness of this proof, we propose a simple patch to fix it:

EXPatch When $p_i = 0$ or $q_i = 0$ for variable x_i, we add a node v_i^{null} in the upper lobe or a node $\bar{v}_i^{null}$ in the lower lobe as in Figure 2.

Adding **EXPatch** into the NP-hardness proof of TCIF problem, it is readily to verify that the mistake in the original proof is now fixed because there are always the upper part and the lower part in each lobe. Note that this will not affect the proof of ”there exists an satisfiable assignment for an instance of SAT problem if and only if there exists two commodities of integral flow in the reduced instance of TCIF problem“ because there is no link from S_2 to node v_i^{null} or $\bar{v}_i^{null}$ for any i. Therefore, **EXPatch** fixes the mistake in [5] and completes the whole proof.

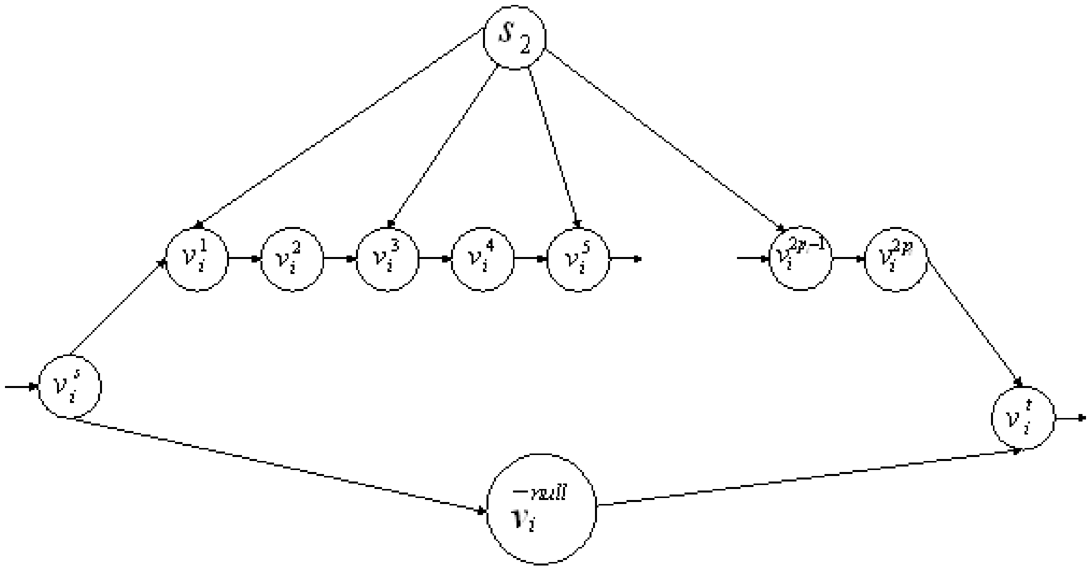

Figure 2: lobe i for variable x_i

4 A heuristic algorithm for Problem Q

Since problem **Q** is NP-hard, in this section we propose a heuristic algorithm to this problem. This algorithm is motivated by both the classic algorithms for k node-disjoint paths with minimal cost [14] [15] [3], our effective load based mathematical framework for measuring the cost of NC-based transmission cost, and our optimal greedy single routing braid algorithm for network coding based routing in Chapter 3.

Algorithms proposed to construct k node-disjoint paths with minimal cost in a given directed graph [14] [15] [3] have a time complexity of $O(k|V|^3)$. In traditional protection studies, these algorithms have been showed to be effective in providing proactive protection to networks against single-node failures. However, by solving problem $\mathbf{Q_0}$ in Chapter 3, we find that the total transmission cost in wireless environment can be further reduced by fully exploring the routing diversity in sensor networks using NC-based routing because the minimal cost of NC-based routing is upper bounded by shortest single path routing in any DAG. Integrating solution ideas behind these two problems together, we propose a heuristic algorithm for problem **Q** that is able to find 2 node-disjoint braids with a total transmission cost upper bounded by two shortest node-disjoint paths and present it as Algorithm 1.

The first step of this heuristic algorithm is finding two node-disjoint paths with minimal total cost using the algorithm proposed in [14] as a reference point. We denote the two routing braids we want to construct as B_1 and B_2 and the two node-disjoint paths with minimal cost we find as $R_1 = S \rightarrow A_1^1 \rightarrow, \ldots, A_m^1 \rightarrow T$ and $R_2 = S \rightarrow A_1^2 \rightarrow, \ldots, A_n^2 \rightarrow T$. And we assign the initial of B_1 and B_2 as:

$$\begin{aligned} B_1 &= \{A_1^1, A_2^1, \ldots, A_m^1\} \\ B_2 &= \{A_1^2, A_2^2, \ldots, A_n^2\} \end{aligned} \tag{5}$$

Without loss of generalness, we assume that the cost of B_1 is larger than or equal to that of B_2, i.e., $C_{B_1} \geq C_{B_2}$. After the initialization of B_1 and B_2, we build an auxiliary graph G_1 by excluding all intermediate nodes in B_2 and all the links attached to these nodes from G. We then use Algorithm **??** to get the optimal single braid on G_1. Denoting the resulting braid as B_{single}^1, we update the first braid as:

$$B_1 = B_{single}^1 \tag{6}$$

With this new B_1, we then perform the same operations to update B_2. We build an auxiliary graph G_2 by excluding all intermediate nodes in B_2 and all the links attached to these nodes from G. Next we run Algorithm **??** again on G_2. Denoting the resulting braid as B_{single}^2, we will be able to update the second braid as:

$$B_2 = B_{single}^2 \tag{7}$$

After these operations, the algorithm stops and we will get two node-disjoint braids with a transmission cost upper bounded by two node-disjoint paths with minimal total cost. The rationale behind this heuristic approach is as follows:

- Instead of randomly dividing nodes into two braids or starting from two randomly paths, starting from two node-disjoint paths with minimal total costs can improve the efficiency of future node assignment process and guarantee the resulting braids have a total transmission cost upper bounded by the two shortest node-disjoint paths;

- Because transient failures are random in WCPS, we allow B_1 to have the priority to select nodes into the braid so that the cost of resulting braids can be balanced. With two node-disjoint braids of equal or balanced cost, the performance of WCPS, including transmission cost and throughput, can stay at a stable level under the existence of random transient failures. This feature is very desirable in modern mission-critical WCPS.

Algorithm 1 A heuristic algorithm for two node-disjoint braids construction

Input: a DAG $G = (V, E)$ with source S and destination T
Construct 2 minimal cost node-disjoint paths $\{R_1, R_2\}$ from S and T, where $C_{R_1} \geq C_{R_2}$
$B_1 = R_1, B_2 = R_2$
$G_1 = G$
for every node V_i in G_1 **do**
    if $V_i \in B_2$ **then**
        Remove V_i and all links attached to V_i from G_1
    end if
end for
Run Algorithm **??** on G_1 and denote the resulting braid as B^1_{single}
$B_1 = B^1_{single}$
$G_2 = G$
for every node V_i in G_2 **do**
    if $V_i \in B_1$ **then**
        Remove V_i and all links attached to V_i from G_2
    end if
end for
Run Algorithm **??** on G_2 and denote the resulting braid as B^2_{single}
$B_2 = B^2_{single}$
Stop and return $\{B_1, B_2\}$

Note: Different from Algorithm **??**, we presented Algorithm 1 as a centralized algorithm. One reason we did this is because the construction of 2 minimal cost node-disjoint paths requires the complete information of the whole graph. The other reason, as we will show in the next section, is that a distributed version of Algorithm 1 would introduce large amounts of communication overhead to the network.

5 Protocol design and implementation

In the last section, we give a description on how to construct two node-disjoint routing braids with low transmission cost from a global perspective. In this section, we present the protocol design of 1+1 proactive NC-based protection (ProNCP) and details of its implementation.

ProNCP is essentially a NC-based routing protocol. It adopts most of EENCR's design principles we presented in Chapter 3, e.g., we implement the random network coding component, the coded feedback scheme and the rate control scheme the same as EENCR. However, we do not adopt the same distance-vector routing engine in EENCR. In EENCR, each node only needs to optimize its forwarder set without considering potential overlapping between the sub-braids of its forwarders and a distance-vector routing engine is sufficient for Algorithm **??**. In ProNCP, on the contrary, to avoid braid overlapping is the most important constraint for braids construction. Therefore, a distance-vector routing engine is insufficient because a sender needs to know the whole graph of the network. A link-state routing component, on the other hand, will introduce high communication overhead and take up too much memory space, and is therefore inapplicable in resource-constrained mission-critical WCPS. To fill this gap, we conduct a long-time sampling test in our testbed to get packet delivery ratio for each link, perform offline computation of Algorithm 1 to get node-disjoint braids for each source, and assign these braids information into the implementation of ProNCP. We leave the design of a low-overhead distributed algorithm for two node-disjoint braids construction as a future research topic. Furthermore, we also add related control schemes in the packet forwarding component to make it fit ProNCP better.

6 Performance evaluation

To characterize the feasibility and effectiveness of network coding in providing proactive protection in mission-critical WCPS, we experimentally evaluate the performance of ProNCP in this section. We first present the experimentation methodology and then the measurement results.

6.1 Methodology

Testbed. We use the *NetEye* wireless sensor network testbed at Wayne State University [1]. The working environment of NetEye is different from that presented in Chapter 2, but the same as that presented in Chapter 3. 130 TelosB motes are deployed in an indoor environment, where every two closest neighboring motes are separated by 2 feet. The layout of the whole testbed is of a grid shape but with some slight variances due to the constraints of the room.

Out of the 130 motes in NetEye, we randomly select 60 motes (with each mote being selected with equal probability) to form a random network for our experimentation. Each of these TelosB motes is equipped with a 3dB signal attenuator and a 2.45GHz monopole antenna. In our measurement study, we set the radio transmission power to be -15dBm (i.e., power level 7 in TinyOS) such that multihop networks can be created. And we use the default MAC protocol provided in TinyOS 2.x.

Protocols studied. To the best of our knowledge, this is the first work to apply network coding against transient node failures in mission-critical WCPS. Some researchers have designed protocols to provide proactive protection using network coding in mesh networks [2] [8] [12]. However, these work cannot be applied to the general scenarios of mission-critical cyber-physical systems because they can only work under the existence of certain routing structures. Given the fact that most of works on routing selection for proactive protection in networks (wired and wireless) are based on the node-disjoint path construction algorithm, we study and compare the performance of the following protocols with the aim to understand the impact of network coding in improving the resilience of mission-critical WCPS against transient node failures,

- *ProNCP*: the 1+1 proactive NC-based protection protocol we propose in this chapter;
- *TNDP*: a routing protocol that sends data along two shortest node-disjoint paths to the receiver.

We implement both protocols in TinyOS 2.x. We choose a batch size of 8 for network coding operation as in Chapter 3. As we explained in the last section, we first conduct a long-time sampling test to get the packet delivery ratio of the whole network. Then we compute both node-disjoint paths and node-disjoint braids offline and assign the results into these two protocols. For TNDP protocol, we define the maximal number of retries for each packet to be 10 if no ACK of this packet was received by the sender/forwarder, this value is the same as what is used in CTP, a shortest single path routing protocol [7].

Performance metrics. For both protocols we study, we evaluate their behavior based on the following metrics:

- *Delivery reliability*: percentage of information elements correctly received by the sink;
- *Delivery cost*: number of transmissions required for delivering an information element from its source to the sink;
- *Goodput*: number of valid information elements received by the sink per second;

Different from the throughput metric used to evaluate the performance of NC-based routing protocols in [4] [10], in this study we use goodput instead. An information element is defined as **valid** if and only if it is linear independent to all packets that are in the same batch and received by the sink.

Topology. We randomly select 60 nodes out of 130 nodes in NetEye to form our experiment topology. From these 60 nodes, we randomly select 10 as source nodes. Each source node periodically generates 40 information elements with an inter-element interval, denoted by Δ_r, uniformly distributed between 500ms and 3s. For ProNCP, every consecutive 8 information elements compose a batch.

Transient node failure model

In our experiments, we deploy a periodic timer for all intermediate nodes in the network. Every time the timer at intermediate node V_i fires, V_i has a probability f to enter a transient failure status, i.e., not able to send or receive any packet. We comparatively study the performance of ProNCP and TNDP under different settings of f:

- $F0$: $f = 0$ for all intermediate nodes in the network; this is to represent the scenario where no node failure happens in the network.
- $F10$ $f = 0.1$ for all intermediate nodes in the network; this is to represent the scenario where intermediate nodes have a 10% chance to stop working for a short period of time.
- $F20$ $f = 0.2$ for all intermediate nodes in the network; this is to represent the scenario where intermediate nodes have a 20% chance to stop working for a short period of time.

6.2 Measurement Results

In what follows, we first present the measurement results for no failure scenario $F0$, then we discuss the case of failure pattern $F10$. In the figures of this section, we present the means and their 95% confidence intervals for the corresponding metrics.

6.2.1 No failure in the network

For the scenario that there is no failure in the network, we run ProNCP and TNDP 5 times each on the selected topology. Figures 3 - 5 show the delivery reliability, delivery cost and goodput of different protocols. In Figure 3, we find that both ProNCP and TNDP achieve a delivery reliability close to 100%. However, the average transmission cost of ProNCP is only 50% of that of TNDP, as shown in Figure 4. This observation is consistent with the design principle of Algorithm 1. By finding the optimal single braid on each auxiliary graph, we are able to significantly reduce the transmission cost of delivering two copies of data from sources to the root.

The reason why TNDP's transmission cost is much higher than ProNCP is because we set a maximal number of retries for each packet when the ACK of this packet is missing. We also try to set this maximal retries a smaller value, e.g. 5 and 8. But the corresponding reliability drops significantly to only 80%. On the contrary, we do not set any maximal number of retries in ProNCP. The number of coded transmissions for each received packet at any node is strictly

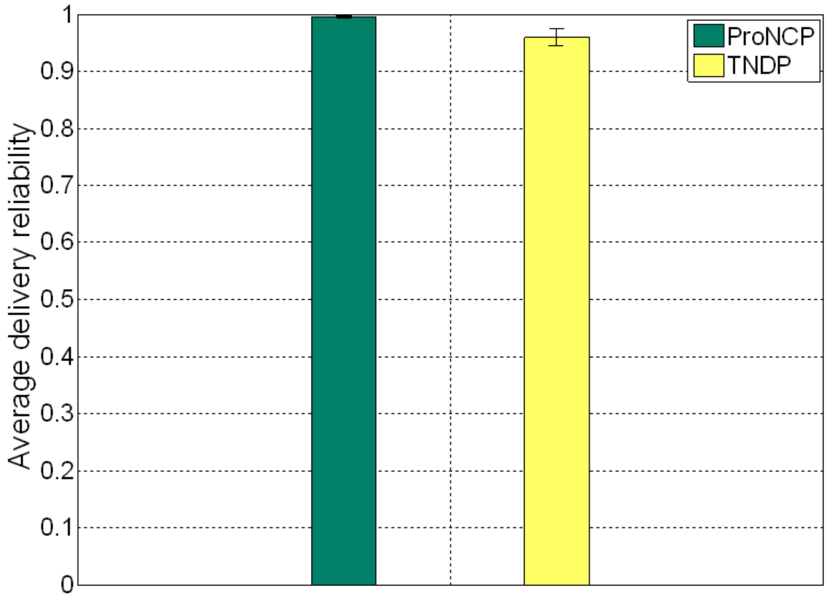

Figure 3: Delivery reliability: 10 sources without failure

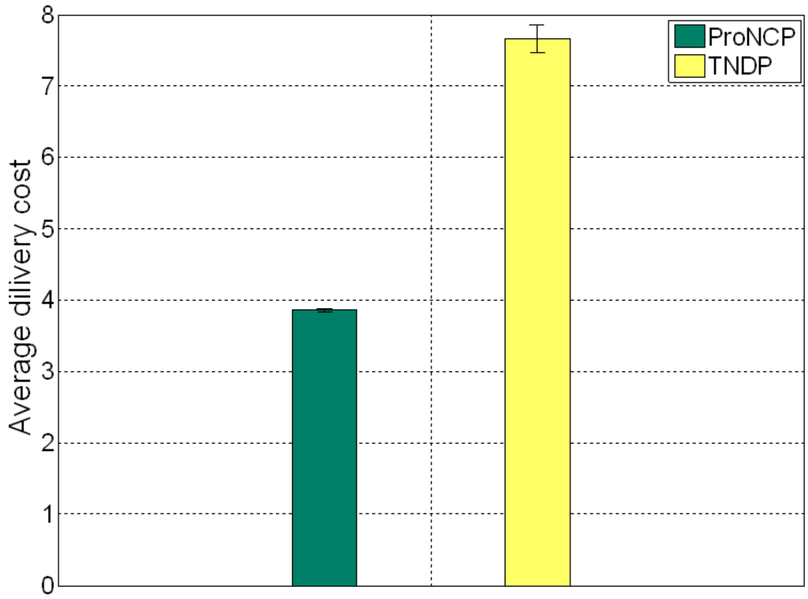

Figure 4: Delivery cost: 10 sources without failure

assigned by the result of Algorithm 1. This further verifies the delivery efficiency of ProNCP over traditional node-disjoint paths algorithm.

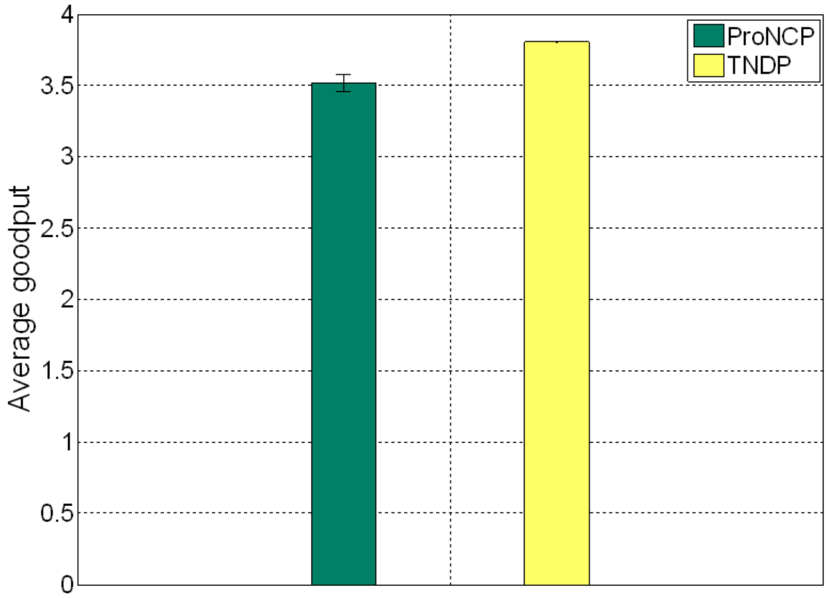

Figure 5: Goodput: 10 sources without failure

In Figure 5, we find that the goodput of TNDP is slightly higher than ProNCP. This characteristic of ProNCP is acceptable. Different from EENCR, senders in ProNCP send two copies of each batch to the root. This proactive protection scheme doubles the traffic load in the whole network, making it more saturated. According to our experiment setting, the goodput of both ProNCP and TNDP are close to the capacity of the whole network.

6.2.2 Random transient node failures in the network

After studying the performance of ProNCP under no failure scenario, we continue to evaluate the performance of ProNCP under the presence of random transient node failure. We run ProNCP and TNDP under each failure model for 10 times. Figures 6 - 8 show the performance of ProNCP and TNDP, including delivery reliability, delivery cost and goodput under both failure models. It is observed in Figure 6 that ProNCP is able to keep the delivery reliability close to 100% under both $F10$ and $F20$ failure models. On the contrary, The delivery reliability of TNDP degrades to 91% under $F10$ model and drops to 80% under $F20$ model. This figure proves that ProNCP is able to provide resilient against transient node failures for

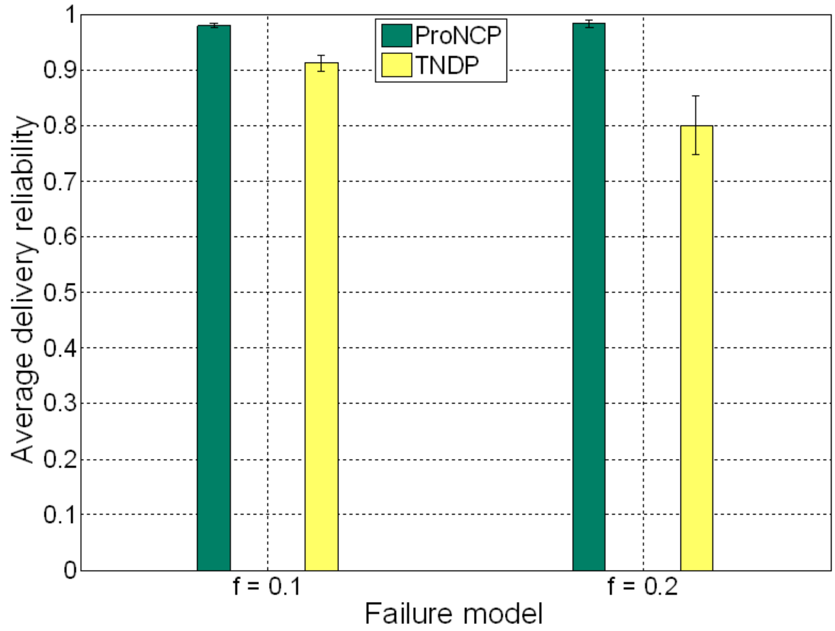

Figure 6: Delivery reliability: 10 sources with failures

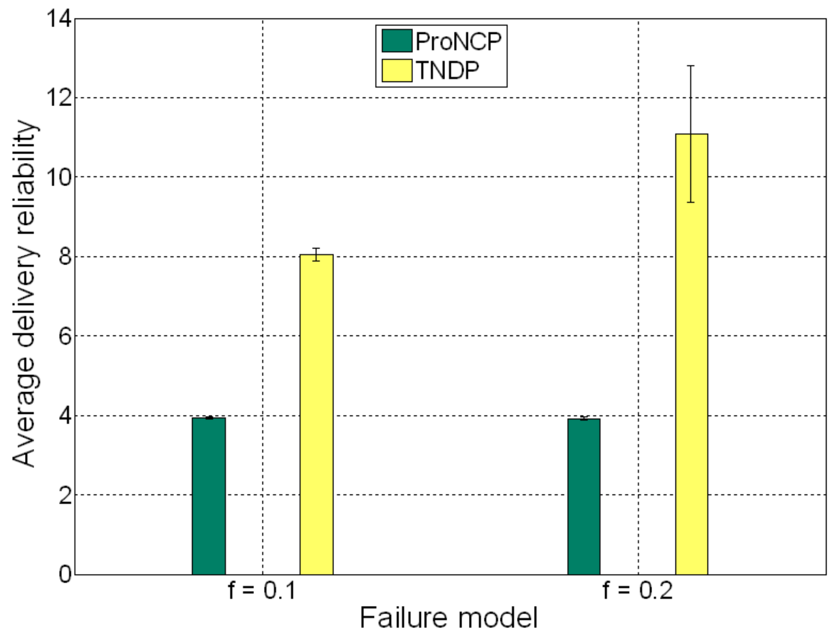

Figure 7: Delivery cost: 10 sources with failures

mission-critical WCPS.

Figure 7 shows that even under the existence of transient node failures, the average transmission cost of ProNCP is kept stable at a very low level. Comparatively, the average transmission cost of $TNDP$ slightly increases in $F10$ case, and drastically increases by 30% while still not able to guarantee data delivery in both failure models. This huge increase of transmission cost in TNDP is because we set the maximal number of retransmissions to be 10 for each packet. Under $F0$ scenario where no transient node failure happened, usually a packet is successfully transmitted over a link before the maximal number of retransmissions is reached. When intermediate nodes randomly enter transient failure status, under which they cannot receive or send packet, other working nodes have to retransmit packets for more times. The higher transient failure probability is, the higher the probability that a node has to keep retransmitting a packet till reaching maximal retries will be. On the contrary, the transmission cost of ProNCP is about the same in both $F10$ and $F20$ compared to the average number of transmissions in $F0$ scenario. This observation proves again the necessity and importance of an optimal algorithm for forwarder set selection in NC-based routing protocols. And it also shows that keep retransmitting under transient node failure cannot bring extra guarantee on reliability but only increase the transmission cost.

Furthermore, the difference between ProNCP's goodput and TNDP's goodput is very little under $F10$ model. And the goodput of ProNCP is even higher than that of TNDP in $F20$. This observation also demonstrates that ProNCP is capable of guaranteeing high data delivery and goodput under various transient node failures.

As a summary, in this section we show that ProNCP is resilient against the dynamics of wireless environment, i.e., transient node failures, in mission-critical WCPS. It is able to provide 1+1 proactive protection to the network with a significant lower transmission cost than the class proactive protection protocol, and maintain a high delivery reliability and goodput under different random transient node failure models.

Related work

There has been a lot work done on protection against node/link failure in both wired and wireless networks. Most existing protection techniques can be categorized into two classes: 1)

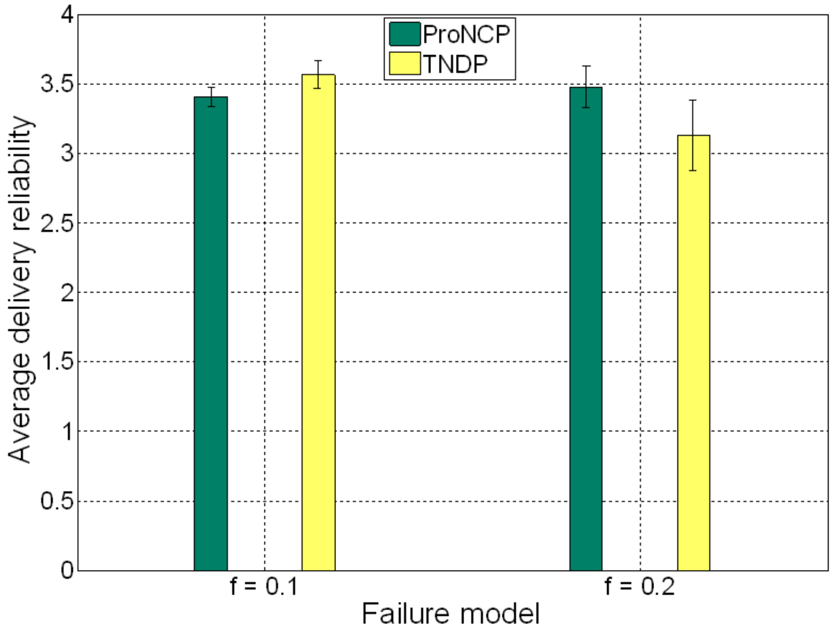

Figure 8: Goodput: 10 sources with failures

proactive protection that sends the same data along two different paths simultaneously, which is also called 1+1 protection. 2) reactive protection that sends the data along one path at the beginning and switch to another path when there is a failure detected, which is also called 1:1 protection. It is straightforward to see that reactive protection has a lower transmission cost than proactive protection while proactive protection needs no response time or failure detection mechanism when failures happened in the network.

In proactive protection, many work focus on constructing node/link disjoint paths such that any single node/link failure will not affect the delivery of data to the destination. Several papers [14] [15] [3] studied disjoint paths in a network and proposed an algorithm to compute k minimum weight node-disjoint paths with a complexity of $O(kN^2)$ where N is the number of nodes in the network. Based on this result, many works have been done. Srinivas *et al.* [13] proposed an algorithm with a complexity of $O(kN^3)$ that controls the transmission power of the source node and compute the corresponding k node-disjoint paths with minimum energy cost in wireless networks. The wireless broadcast nature was considered in this paper for calculating the minimum energy consumption.

Recently, there has been some research on providing protection using network coding. Al-Kofahi *et al.* [2] enhanced the survivability of the information flow between two communicating nodes S and T without compromising the maximum achievable $S-T$ information rate. The authors claimed that most of the links in a network are not bottleneck links, which means that link failures are more likely to affect non-bottleneck links than links in the min-cut. Therefore, they can enhance the survivability of the $S - T$ information flow without reducing the useful $S - T$ rate below the max-flow, if protection is provided to the non-bottleneck links only. The system model of this work is in wired network and the solution cannot provide complete proactive protection to the network.

Kamal *et al.* [8] [12] studied the 1+N protection in the optical network against single link failure. By sending network coded packets on the protection Steiner tree in parallel with the working traffic, the proposed 1+N protocol is able to recover from any single link failure without enduring the delay from switching to the backup path. This problem is strongly NP-hard. And the heuristic solutions proposed in these two papers requires specific routing structure to ensure the protection, which is not realistic in wireless environment.

Braided multipath routing was first proposed in [6]. The major goal of braided multipath routing is to provide reactive protection in networks. After a single path is calculated as the main path, each non-destination node selects another path from itself to the destination. In this way, the data flow can always be switched to another path when there is a failure on the main path. Braided multipath routing can significantly improve the reliability of the network by having a higher connectivity than single path routing [11]. However, it cannot be applied into traditional proactive protection due to high transmission cost.

From the discussion above, we can see that traditional 1+1 protection in wireless network has a low throughput since it does not fully explore the broadcast nature of wireless transmission. Furthermore, packets received by the destination with the same packet number make the transmission redundant, which will increase the transmission cost.

On the contrary, protocols using network coding with opportunistic forwarding [4] [10] [9] have a higher throughput than regular single path routing because any packet received by the destination is not redundant as long as it is linear independent with packets already received by the destination. In the meantime, no node coordination is required between nodes within the same forwarder candidate set. Additionally, network coding with opportunistic forwarding has some implicit proactive protection scheme because the destination can decode all K original packets in the batch as long as it receive any K linear independent packets of this batch.

However, this type of protocols may have high transmission cost caused by no node coordination cost. Furthermore, even though network coding protocols have some implicit proactive protection scheme, they cannot guarantee full proactive protection, i.e., there are cases that one single node failure will lead the destination not receiving K linear independent packets unless it sends retransmission request to the source node.

Having seen both the benefit (the higher throughput and the implicit proactive protection) and drawbacks (high transmission cost and partial protection) brought by wireless network coding, we are motivated to design a network coding protocol for wireless networks in this chapter, such that it can provide full proactive protection against random transient node failures while keeping the high throughput by exploring the broadcast nature of wireless transmission with a low transmission cost.

In [4][10][9], protocols chose nodes with lower delivery cost to the destination into forwarder candidate set. This forwarder selection methods can increase network throughput but increase transmission cost as well because it was originally designed for opportunistic routing. In opportunistic routing, forwarders of the same node are prioritized. A forwarder can only forward the packet it received when no forwarders with higher priority successfully forwarded the packet. In this fashion, network transmission cost can be controlled at a low level. However, in network coding based opportunistic forwarding protocols, every forwarder can forward coded packets when the MAC is ready[4]. This approach did increase the network throughput with no need to design any specific MAC protocol. But if we still adopt the forwarder selection methods designed for opportunistic routing, the transmission cost will be increased.

7 Concluding remarks

NC-based routing has drawn the interests of many researchers in wireless community. Particularly, researchers have been trying to apply this technique into proactive protection for networks. In this section we study how to design energy-efficient network coding based solution in mission-critical wireless cyber-physical systems. Specifically, we study how to provide 1+1 proactive protection in sensor networks. We formally defined the two node-disjoint routing braids problem and prove its NP-hardness via a reduction from 2-partition problem. We then design a heuristic node assignment algorithm to compute two node-disjoint braids with a lower transmission cost than any two node-disjoint paths in the network. Based on this algorithm, we propose ProNCP, a proactive NC-based protection protocol. ProNCP inherits similar modules and components in EENCR, but we add corresponding control schemes to make the implementation satisfy the requirement of proactive protection in mission-critical WCPS.

We evaluate the performance of ProNCP on the NetEye testbed by comparing it with the two shortest node-disjoint paths algorithm (TNDP), the most classic approach in proactive protection. When there is no failure happening in the network, ProNCP is able to achieve a delivery reliability close to 100% with only half of the cost of of TNDP. When intermediate nodes have a probably of randomly entering transient failure state, the delivery reliability of TNDP degrades significantly while ProNCP is still able to maintain a high reliability and a low transmission cost. The resilience of ProNCP shown in the evaluation demonstrates the benefits of network coding in providing proactive protection for mission-critical WCPS. Future work towards this research direction includes the design of a distributed node-disjoint braids construction algorithm with low communication overhead.

References

[1] NetEye testbed. http://neteye.cs.wayne.edu/neteye/home.php.

[2] O.M. Al-Kofahi and A.E. Kamal. Max-flow protection using network coding. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5, 2011.

[3] R. Bhandari. Optimal physical diversity algorithms and survivable networks. In *Computers and Communications, 1997. Proceedings., Second IEEE Symposium on*, pages 433 –441, jul 1997.

[4] Szymon Chachulski, Michael Jennings, Sachin Katti, and Dina Katabi. Trading structure for randomness in wireless opportunistic routing. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '07, pages 169–180, New York, NY, USA, 2007. ACM.

[5] S. Even, A. Itai, and A. Shamir. On the complexity of time table and multi-commodity flow problems. In *Proceedings of the 16th Annual Symposium on Foundations of Computer Science*, pages 184–193, Washington, DC, USA, 1975. IEEE Computer Society.

[6] Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(4):11–25, October 2001.

[7] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. Collection tree protocol. In *ACM SenSys*, 2009.

[8] Ahmed E. Kamal. 1 + n network protection for mesh networks: network coding-based protection using p-cycles. *IEEE/ACM Trans. Netw.*, 18(1):67–80, February 2010.

[9] D. Koutsonikolas, Chih-Chun Wang, and Y.C. Hu. Ccack: Efficient network coding based opportunistic routing through cumulative coded acknowledgments. In *INFOCOM, 2010 Proceedings IEEE*, pages 1 –9, 2010.

[10] Yunfeng Lin, Baochun Li, and Ben Liang. Codeor: Opportunistic routing in wireless mesh networks with segmented network coding. In *Network Protocols, 2008. ICNP 2008. IEEE International Conference on*, pages 13 –22, 2008.

[11] Victoria Manfredi, Robert Hancock, and Jim Kurose. Robust routing in dynamic manets, 2008.

[12] M. Mohandespour and A.E. Kamal. 1+n protection in polynomial time: A heuristic approach. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5, 2010.

[13] Anand Srinivas and Eytan Modiano. Minimum energy disjoint path routing in wireless ad-hoc networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, MobiCom ’03, pages 122–133, New York, NY, USA, 2003. ACM.

[14] J. W. Suurballe. Disjoint paths in a network. *Networks*, 4(2):125–145, 1974.

[15] J. W. Suurballe and R. E. Tarjan. A quick method for finding shortest pairs of disjoint paths. *Networks*, 14(2):325–336, 1984.