In-Network Processing for Mission-Critical Wireless Networked Sensing and Control: A Real-Time, Efficiency, and Resiliency Perspective

Qiao Xiang
Advisor: Dr. Hongwei Zhang
Committee: Dr. Nathan Fisher
Dr. Chengzhong Xu
Dr. Lihao Xu

Department of Computer Science
Wayne State University
Detroit, Michigan 48202, USA
xiangq27@wayne.edu

March 27th, 2014
Outline

1. Introduction
2. Real-time packet packing scheduling
3. Energy-efficient network coding based routing
4. Proactive network coding based protection
5. Conclusion
6. Acknowledgments
Outline

1. Introduction
2. Real-time packet packing scheduling
3. Energy-efficient network coding based routing
4. Proactive network coding based protection
5. Conclusion
6. Acknowledgments
Wireless Sensor Networks
- Highly resource-constrained

In-Network Processing
- Reduce traffic flow $\rightarrow$ resource efficient
- End-to-end QoS are usually not considered

Mission-Critical Wireless Networked Sensing and Control (WNSC)
- Close-loop control
- More emphasis on end-to-end QoS, especially latency, reliability and resiliency
Packet packing
- Application independent INP
- Simple yet useful in practice
  - UWB intra-vehicle control
  - IETF 6LowPAN: high header overhead

Network coding (NC)
- First proposed in wired networks
- Provide benefits on throughput and robustness
- Naturally extended into wireless environment
- Integrated with opportunistic routing, e.g., MORE
Our focus

- Explore system benefits of different INP methods
- Joint optimization between INP and QoS
- Temporal and spatial data flow control in WNSC

Our methodology

- Computational complexity study on different optimization problems
- Light-weight algorithm and protocol design
- Theoretical analysis of different algorithms
- Extensive testbed-based experimental evaluation
Roadmap

1. Real-time packet packing scheduling
2. Energy-efficient NC-based routing
3. Proactive NC-based protection
Outline

1. Introduction
2. Real-time packet packing scheduling
3. Energy-efficient network coding based routing
4. Proactive network coding based protection
5. Conclusion
6. Acknowledgments
Real-time packet packing scheduling

System Model

- A directed collection tree $T = (V, E)$
- Edge $(v_i, v_j) \in E$ with weight $ETX_{v_i,v_j}(l)$
- A set of information elements $X = \{x\}$
- Each element $x : (v_x, l_x, r_x, d_x)$

Problem (P)

- Schedule the transmission of $X$ to $R$
- Minimize the total number of transmissions
- Satisfy the latency constraints of each $x \in X$
Complexity analysis

Problem $P_0$
- Elements are of equal length
- Each node has at most one element

Problem $P_1$
- Elements are of equal length
- Each node generates elements periodically

Problem $P_2$
- Elements are of equal length
- Arbitrary data generating pattern
### Complexity analysis

<table>
<thead>
<tr>
<th>$P_0, P_1, P_2, P$</th>
<th>$K \geq 3$</th>
<th>$K = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
<td>strongly NP-hard</td>
<td>strongly NP-hard</td>
</tr>
<tr>
<td>NP-hard to achieve approximation ratio</td>
<td>$1 + \frac{1}{200N}(1 - \frac{1}{\epsilon})$</td>
<td>$1 + \frac{1}{120N}(1 - \frac{1}{\epsilon})$</td>
</tr>
</tbody>
</table>

$K = \text{Maximal packet length}$

$N = |X|$

Re-aggregation: a packed packet can be dispatched for further packing
\( K \geq 3, \ P_0 \) is NP-hard in tree structures – a reduction from SAT problem

Given a SAT instance with \( n \) clauses and \( m \) variables

For each clause \( j \)

For \( m \) variables
For each variable occurred in clause $j$

Auxiliary elements related to the red ones
Complexity analysis

When $K \geq 3$ and $T$ is a tree, regardless of re-aggregation

- $P_0$ is NP-hard → $P_1$ is NP-hard → $P_2$ is NP-hard → $P$ is NP-hard

When $K \geq 3$ and $T$ is a chain, regardless of re-aggregation

- The reduction from $SAT$ problem still holds*

When $K = 2$ and re-aggregation is not prohibited

- The reduction from $SAT$ problem still holds in both tree and chain structures

When $K = 2$ and re-aggregation is prohibited

- Problem $P$ is equivalent to the maximum weighted matching problem in an interval graph
- Solvable in $O(N^3)$ by Edmond’s Algorithm

*: This solves an open problem in batch process
A utility based online algorithm

When a node receives a packet $\textit{pkt}$ with length $s_f$

- Decisions: to hold or to transmit immediately
- Utility of action: Reduced Amortized Cost
- One-hop locality

$$\text{AC} = \frac{\# \text{ of TX}}{\text{length of data}}$$
A utility based online algorithm

Utility of holding a packet:

\[ AC'_l = \frac{1}{L - s'_f} ETX_{j_R}(L - s'_f) \]

\[ AC_l = \frac{1}{L - s'_f + S_l} ETX_{j_R}(L - s'_f + S_l) \]

Cost without packing

\[ U_l = AC'_l - AC_l \]

Utility of transmitting a packet:

\[ U'_p = \frac{t'_f/tp}{ETX_{p_j R}(s_p)} - \frac{t'_f/tp}{ETX_{p_j R}(L)} \]

\[ = \frac{ETX_{p_j R}(s_p)}{s_p} - \frac{ETX_{p_j R}(L)}{L} \]

\[ U''_p = \frac{L - s'_f}{L - s_p} ETX_{p_j R}(s_p) \]

\[ + \frac{L - s'_f}{L - s_p} ETX_{p_j R}(L) + \frac{I_{mod} ETX_{p_j R}(s_p + l_{mod})}{[L - s'_f/(L - s_p)] s_p + L - s'_f} \]

\[ U_p = \begin{cases} 
U'_p & \text{if } \frac{t'_f}{t_p}(L - s_p) \leq L - s'_f \\
U''_p & \text{otherwise}
\end{cases} \]

Every packet received by parent can get fully packed via \textit{pkt}.
A utility based online algorithm

Decision Rule
- The packet should be immediately transmitted if $U_p > U_l$
- The packet should be held if $U_p \leq U_l$

Competitive Ratio
- Problem $P'$
  - $T$ is a complete tree
  - Leaf nodes generate elements at a common rate

Theorem: For problem $P'$, $tPack$ is $\min\{K, \max_{v_j \in V_{>1}} \frac{2ETX_{v_jR}}{2ETX_{v_jR} - ETX_{p_jR}}\}$-competitive, where $K$ is the maximum number of information elements that can be packed into a single packet, $V_{>1}$ is the set of nodes that are at least two hops away from the sink $R$.
- Example: When $ETX$ is the same for each link, $tPack$ is 2-competitive.
Performance evaluation

Experiment setting up

- Testbed: NetEye, a 130-sensor testbed (State Hall)
- Topology: 120 nodes, half are source nodes, 1 sink node
- Protocols compared: noPacking, simplePacking, spreaded latency, common clock, \( tPack \)
- Traffic patterns: periodic traffic and event traffic
- Metrics: packing ratio, delivery reliability, delivery cost, deadline catching ratio and latency jitter
3-second periodic traffic: packing ratio

![Bar chart showing packing ratio for different latency levels (L1, L3, L5) with different packing methods (noPack, simplePack, SL, CC, tPack).]
3-second periodic traffic: delivery reliability

![Graph showing delivery reliability for different latency levels and packet packing methods.](image)

- noPack
- simplePack
- SL
- CC
- tPack

Median delivery reliability vs. Maximum allowable latency for L1, L3, and L5.
3-second periodic traffic: delivery cost
3-second periodic traffic: deadline catching ratio

![Graph showing the median deadline catching ratio for different maximum allowable latencies](image)

- **SL**
- **CC**
- **tPack**

**Maximum allowable latency**

- L1
- L3
- L5

**Median deadline catching ratio**
3-second periodic traffic: latency jitter

![Bar chart showing latency jitter for different packet packing schemes with varying maximum allowable latency. The chart compares noPack, simplePack, SL, CC, and tPack. The y-axis represents median latency jitter, and the x-axis represents maximum allowable latency (L1, L3, L5).]
6-second periodic traffic
9-second periodic traffic
Event traffic

- Mean packing ratio
- Median delivery reliability
- Mean delivery cost
- Median deadline catching ratio
- Median latency jitter

Qiao Xiang (WSU) Ph.D. Dissertation Defense 03/27/2014 26/77
Remarks

- Impact of INP constraints on problem complexity
- Feasibility of a simple, distributed online algorithm
- System benefits in terms of efficiency and predictable latency
- Temporal data flow control in mission-critical WNSC
Outline

1. Introduction
2. Real-time packet packing scheduling
3. Energy-efficient network coding based routing
4. Proactive network coding based protection
5. Conclusion
6. Acknowledgments
Inter-flow coding vs. intra-flow coding

Inter-flow coding
- Designed for multiple source-destination pairs, i.e., multi-unicast traffic

Intra-flow coding
- Designed for single source-destination pair traffic

In WNSC:
- We focus on multi-hop convergecast traffic, i.e., multiple sources to one destination
  - Inter-flow coding
    - Absence of perfect feedback channel
    - Transform convergecast to multi-unicast traffic is complex
  - Intra-flow coding
    - Easier to transplant to convergecast
Intra-flow network coding: an example

\[ \{1, 2, 3\} = X_1 + 2X_2 + 3X_3 \]
\[ \{1, 1, 1\} = X_1 + X_2 + X_3 \]

K=3
\[ \{1, 0, 0\} = X_1 \]
\[ \{0, 1, 0\} = X_2 \]
\[ \{0, 0, 1\} = X_3 \]

\[ \{2, 3, 4\} = 2X_1 + 3X_2 + 4X_3 \]
\[ \{3, 5, 10\} = 3X_1 + 5X_2 + 10X_3 \]

\[ \{3, 4, 6\} = 3X_1 + 4X_2 + 6X_3 \]

\[ \{2, 3, 5\} = 2X_1 + 3X_2 + 5X_3 \]
\[ \{1, 1, 1\} = X_1 + X_2 + X_3 \]
System model

- A directed graph $G = (V, E)$ with one source $S$ and one destination $T$
- Edge $(i, j) \in E$ with link reliability $P_{ij} = \frac{1}{ETX_{ij}}$
- Node $i$ with transmission cost $C_{iT}$ and a forwarder candidate set $FCS_i$

Problem $Q_0$

- Decide the forwarder set $FS_i$ for each node $i$
- Decide the effective load of each node in $FS_i$ for each node $i$
- Minimize the total transmission cost from $S$ to $T$
A mathematical framework for cost of NC-based routing

### Definition

For a node $j$ in the forwarder candidate set $FCS_i$, the **effective load** $L_j$ is defined as the number of linear independent packets received by $j$ but none of the nodes in $FCS_i$ that has lower transmission cost to the destination.

### How does the framework work?

1. Define the whole forwarder set as a virtual node $V_S$
2. Compute the transmission cost from the $S$ to $V_S$
3. Sort forwarders in non-increasing order of their transmission cost
4. Each forwarder only forwards its effective load with corresponding cost
5. Sum up all transmission cost
An example

\[ P_2 \geq P_4 \geq P_6 \]
Energy-efficient network coding based routing

A mathematical framework for cost of NC-based routing

\[ C_{SDS}(K) = \frac{K}{P_{SV_{DS}}} = \frac{K}{1 - (1 - P_1)(1 - P_3)(1 - P_5)} \]

\[ K_S^A = \frac{KP_1}{1 - (1 - P_1)(1 - P_3)(1 - P_5)} \]

\[ K_S^B = \frac{KP_3}{1 - (1 - P_1)(1 - P_3)(1 - P_5)} \]

\[ K_S^C = \frac{KP_5}{1 - (1 - P_1)(1 - P_3)(1 - P_5)} \]

\[ L_A = K_S^A \]

\[ L_B = K_S'^B = K \frac{K_S^B}{K} (1 - P_1) = K_S^B (1 - P_1) \]

\[ L_C = K_S'^C = K_S^C (1 - P_1)(1 - P_3) \]
\[ C_S(K) = C_{SDS}(K) + C_{AT}(L_A) + C_{BT}(L_B) + C_{CT}(L_C) \]

\[ = \frac{K}{1 - (1 - P_1)(1 - P_3)(1 - P_5)} \]

\[ + \frac{L_A}{P_2} + \frac{L_B}{P_4} + \frac{L_C}{P_6} \]

\[ = \frac{K}{1 - (1 - P_1)(1 - P_3)(1 - P_5)} \]

\[ \cdot \left[ 1 + \frac{P_1}{P_2} + \frac{P_3(1 - P_1)}{P_4} + \frac{P_5(1 - P_1)(1 - P_3)}{P_6} \right] \]
Algorithm 1 Compute the transmission cost of NC-based routing for the current node $S$ with $M$ forwarder candidates

1: Input: current node $S$, $D_S = \{A_1, A_2, \ldots, A_M\}$
2: Output: $C_S(1)$: the expected number of transmissions to deliver 1 packet from $S$ to $T$
3: Sort nodes in $D_S$ by a non-descending order of $C_{A_i}(1)$, where $i = 1, 2, \ldots, M$.
4: Sorted nodes are labeled as $\{A'_1, A'_2, \ldots, A'_M\}$
5: $C_{SDS}(1) = \frac{1}{1 - \prod_{i=1}^{M}(1 - P_{SA'_i})}$
6: $L_{A'_1} = C_{SDS}(1)P_{SA'_1}$
7: $F = 1 - P_{SA'_1}$
8: for $i \rightarrow 2, 3, \ldots, M$ do
9: $L_{A'_i} = C_{SDS}(1)P_{SA'_i}F$
10: $C_{A'_i}(L_{A'_i}) = L_{A'_i}C_{A'_i}(1)$
11: $F = F(1 - P_{SA'_i})$
end for
12: $C_S(1) = C_{SDS}(1) + \sum_{i=1}^{M} C_{A'_i}(L_{A'_i})$
A greedy approach

1. Sort forwarder candidates in non-increasing order of their transmission cost;
2. Select the best candidate remaining into forwarder set;
3. Keep it in the set if the total transmission cost can be reduced, go back to last step;
4. Stop if the total transmission cost cannot be reduced.
Minimize the cost of NC-based routing

**Algorithm 2** Compute the minimal transmission cost of NC-based routing and the corresponding $FS$ for the input node $S$ with $M$ forwarders

1: Input: node $S$, $D_S = \{A_1, A_2, \ldots, A_M\}$, $FS_S = \emptyset$
2: Output: $C_S^*(1)$: the minimal transmission cost to deliver 1 packet from $S$ to $T$
3: Sort nodes in $D_S$ by a non-descending order of $C_{A_i}(1)$, where $i = 1, 2, \ldots, M$.
4: Sorted nodes are labeled as $\{A'_1, A'_2, \ldots, A'_M\}$
5: $FS_S = \{A'_1\}$
6: $C_S^*(1) = \frac{1}{P_{SA'_1}} + C_{A'_1}(1)$
7: for $i \rightarrow 2, 3, \ldots, M$ do
8: Run Algorithm 1 with input $S$ and $D_S = \{A'_1, \ldots, A'_i\}$
9: Get the result as $C_S^{\text{new}}(1)$
10: if $C_S^{\text{new}}(1) > C_S(1)$ then
11: break
12: else
13: $FS_S = FS_S \cup A'_i$
14: $C_S^*(1) = C_S^{\text{new}}(1)$
15: end if
16: end for
Theorem

Given a node $S$ and its forwarder candidate set $D_S = \{A_1, A_2, \ldots, A_M\}$, Algorithm 2 yields the minimal transmission cost to the destination node of NC-based routing and the corresponding forwarder set.

We proved this theorem by contradiction.
Properties of the optimal routing braid

**Theorem**

Given a node $S$ with a candidate set $FCS_S$ of $M$ forwarders, the optimal forwarder set $FS_S$ computed in Algorithm 2 does not always contain node $A^*$ where $A^* \in FCS_S$ and

$$\frac{1}{P_{SA^*}} + C_{A^*} \leq \frac{1}{P_{SA_i}} + C_{A_i} \text{ for any } i \in FCS_S/\{A^*\}.$$  

Shortest single path routing is not always in the optimal braid.
Properties of the optimal routing braid

[Diagram showing a network with nodes S, A_1, A_2, A_3, and T, with labeled edges representing costs.]
The optimal routing braid is \( \{A_1, A_2\} \)

\[
C_{\{A_1, A_2\}} = \frac{1}{1 - (1 - 0.1)(1 - 0.15)} \cdot \left[ 1 + \frac{0.1}{0.4} + \frac{0.15(1 - 0.1)}{0.2} \right]
\]

\[
= \frac{1}{0.235} \cdot \left( 1 + \frac{1}{4} + \frac{0.135}{0.2} \right)
\]

\[
= 8.1915
\]

\[
C_{\{A_1, A_2, A_3\}} = \frac{1}{1 - (1 - 0.1)(1 - 0.15)(1 - 0.9)} \cdot \left[ 1 + \frac{0.1}{0.4} + \frac{0.15(1 - 0.1)}{0.2} + \frac{0.9(1 - 0.1)(1 - 0.15)}{0.1} \right]
\]

\[
= \frac{1}{0.9235} \cdot \left( 1 + \frac{1}{4} + \frac{0.135}{0.2} + \frac{0.6885}{0.1} \right)
\]

\[
= 9.5398 > C_{\{A_1, A_2\}}
\]
Properties of the optimal routing braid

Theorem

Given a node $S$ with a candidate set $FCS_S$ of $M$ forwarders, the optimal transmission cost $C^*_S$ computed in Algorithm 2 is always lower than or equal to $\frac{1}{P_{SA^*} + C_{A^*}}$ where $A^* \in FCS_S$ and $\frac{1}{P_{SA^*}} + C_{A^*} \leq \frac{1}{P_{SA_i}} + C_{A_i}$ for any $i \in FCS_S/\{A^*\}$.

Cost of optimal NC-based routing is upper bounded by shortest single path routing.
EENCR: an energy-efficient NC-based routing protocol

- Routing engine: a distributed implementation of Algorithm 2
- M-NSB: a coded ACK scheme to solve the collective space problem with lower implementation complexity than CCACK
- Rate control: nodes forward a flow after receiving a load-dependent threshold of packets to 1) reduce contention and 2) avoid potential linear dependence between forwarded packets
Experiment setting up

- Testbed: NetEye, a 130-sensor testbed (Maccabees Building)
- Topology: 40 nodes, 10/20 are source nodes, 1 sink node
- Protocols compared: EENCR, CTP, MORE, CodeOR
- Traffic pattern: 3-second periodic traffic
- Metrics: delivery reliability, delivery cost, goodput and routing diversity
10-source: delivery reliability

![Graph showing delivery reliability comparison]

- **EENCR**
- **CTP**
- **MORE**
- **CodeOR**

The graph compares the average delivery reliability of different network coding algorithms. The x-axis represents different source configurations, while the y-axis shows the delivery reliability. The bars indicate the performance of each algorithm, with **EENCR** achieving the highest reliability, followed by **MORE** and **CTP**, and **CodeOR** showing slightly lower reliability.
10-source: delivery cost

![Graph showing delivery cost comparison]

- **EENC**
- **CTP**
- **MORE**
- **CodeOR**

The graph illustrates the delivery cost for different network coding based routing protocols. MORE shows the highest delivery cost, followed by CodeOR, CTP, and EENC.
10-source: goodput

![Bar chart showing average goodput for different network coding methods.]

- EENCR
- CTP
- MORE
- CodeOR

Average goodput

0 1 2 3 4 5 6 7

0 1 2 3 4 5 6 7
10-source: routing diversity

The diagram shows a bar chart comparing the number of forwarders for different routing methods:

- **EENCR**
- **MORE**
- **CodeOR**

The x-axis represents the number of forwarders, ranging from 0 to 6, while the y-axis represents the different routing methods. The chart visually compares the performance of these methods in terms of the number of forwarders required.
Energy-efficient network coding based routing

Performance evaluation

20-source

Qiao Xiang (WSU) Ph.D. Dissertation Defense 03/27/2014
- The first mathematical framework on cost measurement of NC-based routing
- A greedy optimal cost minimization and forwarder set selection algorithm
- System benefits in terms of efficiency and reliability
- **Spatial data flow control in mission-critical WNSC**
Outline

1. Introduction
2. Real-time packet packing scheduling
3. Energy-efficient network coding based routing
4. Proactive network coding based protection
5. Conclusion
6. Acknowledgments
System model

- A directed graph \( G = (V, E) \) with one source \( S \) and one destination \( T \)
- Edge \((i, j) \in E\) with link reliability \( P_{ij} = \frac{1}{ETX_{ij}}\)
- Node \(i\) with transmission cost \( C_{iT}\) and a forwarder candidate set \(FCS_i\)

Problem \(Q\)

- Decide the forwarder set \(FS_i\) for each node \(i\)
- Decide the effective load of each node in \(FS_i\) for each node \(i\)
- Construct 2 node-disjoint routing braids
  - 1+1 protection against transient node failure
- Minimize the total transmission cost from \(S\) to \(T\)

Problem \(Q'\)

- The same as \(Q\) except that in the input graph, all the paths from \(S\) to \(T\) are node-disjoint to each other
Complexity analysis

A 0-1 programming formulation of $Q'$

Minimize: $C_1 + C_2 = \frac{1}{1 - \prod_{i=1}^{m}(1 - x_i \cdot P_{2i-1})} \cdot \sum_{i=1}^{m} x_i \cdot P_{2i-1} \prod_{j=1}^{i-1}(1 - x_j \cdot P_{2j-1})$ $\frac{1}{P_{2i}}$ $\frac{1}{1 - \prod_{i=1}^{m}(1 - y_i \cdot P_{2i-1})} \cdot \sum_{i=1}^{m} y_i \cdot P_{2i-1} \prod_{j=1}^{i-1}(1 - y_j \cdot P_{2j-1})$ $\frac{1}{P_{2i}}$ $+ \max\left\{ \frac{1}{1 - \prod_{i=1}^{m}(1 - x_i \cdot P_{2i-1})}, \frac{1}{1 - \prod_{i=1}^{m}(1 - y_i \cdot P_{2i-1})} \right\}$

such that

$x_i \in \{0, 1\}$

$y_i \in \{0, 1\}$

$x_i + y_i \leq 1$

$P_{2i} \geq P_{2(i+1)}$

$0 \leq P_{2i} \leq 1$

$0 \leq P_{2i-1} \leq 1$

for $i = 1, 2, \ldots, m$,
Lemma

Problem $Q'$ is NP-hard.

We prove this lemma via a reduction from the 2-PARTITION problem.

Theorem

Problem $Q$ is NP-hard.

An immediate result from the NP-hardness of $Q'$. 
A heuristic algorithm for Problem Q

1. Compute 2 node-disjoint paths $R_1$ and $R_2$ with minimal total cost on $G$ ($C_{R_1} \geq C_{R_2}$)
2. Build an auxiliary graph $G_1$ excluding all intermediate nodes of $R_2$ and links connected to these nodes from $G$
3. Compute the first braid $B_1$ as the optimal single braid on $G_1$
4. Build an auxiliary graph $G_2$ excluding all intermediate nodes of $B_1$ and links connected to these nodes from $G$
5. Compute the second braid $B_2$ as the optimal single braid on $G_2$
Finding node-disjoint braids - an example
Finding node-disjoint braids - an example
Finding node-disjoint braids - an example
Finding node-disjoint braids - an example
Finding node-disjoint braids - an example
Finding node-disjoint braids - an example
Finding node-disjoint braids - an example
Algorithm 3 A heuristic algorithm for two node-disjoint braids construction

1: Input: a DAG $G = (V, E)$ with source $S$ and destination $T$
2: Construct 2 minimal cost node-disjoint paths $\{R_1, R_2\}$ from $S$ and $T$, where $C_{R_1} \geq C_{R_2}$
3: $B_1 = R_1, B_2 = R_2$
4: $G_1 = G$
5: for every node $V_i$ in $G_1$ do
6: if $V_i \in B_2$ then
7:   Remove $V_i$ and all links attached to $V_i$ from $G_1$
8: end if
9: end for
10: Run Algorithm 2 on $G_1$ and denote the resulting braid as $B_{single}^1$
11: $B_1 = B_{single}^1$
12: $G_2 = G$
13: for every node $V_i$ in $G_2$ do
14: if $V_i \in B_1$ then
15:   Remove $V_i$ and all links attached to $V_i$ from $G_2$
16: end if
17: end for
18: Run Algorithm 2 on $G_2$ and denote the resulting braid as $B_{single}^2$
19: $B_2 = B_{single}^2$
20: Stop and return $\{B_1, B_2\}$
Performance evaluation

Experiment setting up

- **Testbed:** NetEye, a 130-sensor testbed (Maccabees Building)
- **Topology:** 60 nodes, 10 are source nodes, 1 sink node
- **Protocols compared:** ProNCP, TNDP (two node-disjoint path protection)
  - **Note:** braids and routes are computed offline via sampling results.
- **Traffic pattern:** 3-second periodic traffic
- **Failure model:** probabilistic transient failure on intermediate nodes
- **Metrics:** delivery reliability, delivery cost and goodput
No failure: delivery reliability and goodput

![Graph showing delivery reliability and goodput for ProNCP and TNCP]

- **ProNCP**
- **TNCP**

Average delivery reliability and average goodput are compared between ProNCP and TNCP.
No failure: delivery cost

![Bar chart showing comparison of average delivery cost between ProNCP and TNDP.](chart.png)
Transient failure: delivery reliability

![Bar graph showing delivery reliability for ProNCP and TNDP under different failure models (f = 0.1 and f = 0.2).]
Transient failure: delivery cost

The chart illustrates the average delivery cost for different failure models and two different failure frequencies, f = 0.1 and f = 0.2. The chart compares ProNCP and TNDP models, showing a higher delivery cost for TNDP in both failure scenarios.
Transient failure: goodput

![Graph showing goodput for ProNCP and TNDP under different failure models (f = 0.1 and f = 0.2).]
Remarks

- Complexity study on 1+1 proactive NC-based protection problem
- A heuristic node-disjoint-braids construction algorithm
- System benefits in terms of reliability, delivery cost and stability under transient failures
- Spatial data flow control and protection in mission-critical WNSC
- Future work: design of low-overhead distributed algorithm for node-disjoint braids construction
Conclusion

- Explore optimization of different INP in mission-critical WNSC
- Demonstrate system benefits in terms of reliability, delay, efficiency and resiliency theoretically and experimentally
- Achieve temporal and spatial data flow control
I want to thank my advisor, Dr. Hongwei Zhang, for his tremendous help, full support and patient guidance during my graduate study. I would not achieve anything I have now without you. You are the most important mentor in my life.

I want to thank my dissertation committee, including Dr. Nathan Fisher, Dr. Chengzhong Xu and Dr. Lihao Xu, for all their help and advice in the past few years. I learned a lot from each of you and I really appreciate it.

I want to thank my family, especially my parents, for raising me, supporting me and always having faith in me. I love you, mom and dad.
I want to thank every past and current member of DNC, including Xin Che, Xi Ju, Xiaohui Liu, Chuan Li, Yu Chen, Yuehua Wang, Qing Ai, Pengfei Ren and Ling Wang. It was an unforgettable and pleasant memory working with you guys together. Especially for Xiaohui, we worked closely for a long time. Thank you and good luck with your future career.

I want to thank Jiayu Gong and Xubo Fei. You guys are my best friends in Detroit. We shared a lot of good memory in both work and life.

I want to thank Jianqing Luo. You have been a real role model for me as a graduate student.

I want to thank Ting Yan, Bohuan Wei, Na Zhao and Bo Meng. We started this journey from Nankai years ago. Thank you guys for always having faith in me.
Most importantly, I want to thank my girlfriend, Elita. Thank you for being part of my life when I was in the darkest and most helpless moment of my life. Thank you for rekindling my enthusiasm, ambition and devotion towards computer scientific research. I could not imagine what my life would become without your support. I love you.