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

Outline

Introduction

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

Introduction

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

Introduction

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

Introduction

Roadmap

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

Outline

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

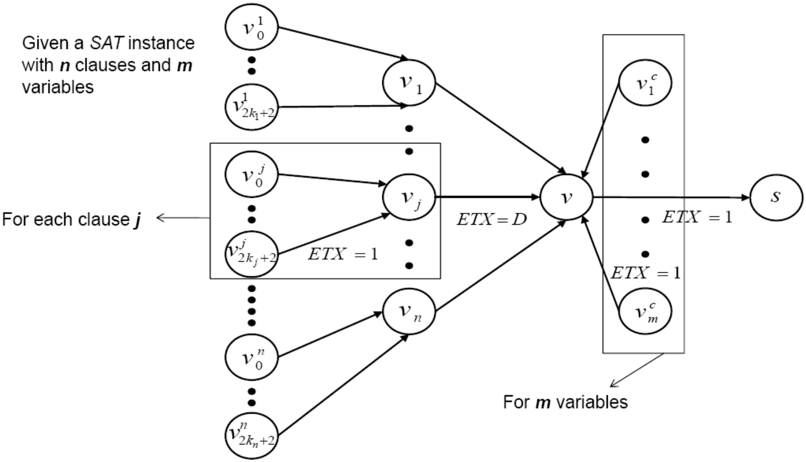
P_0, P_1, P_2, P	$K \geq 3$	$K = 2$	
		re-aggregation is not prohibited	re-aggregation is prohibited
Complexity	strongly NP-hard	strongly NP-hard	$O(N^3)$
NP-hard to achieve approximation ratio	$1+\frac{1}{200N}(1-\frac{1}{\epsilon})$	$1+\frac{1}{120N}(1-\frac{1}{\epsilon})$	

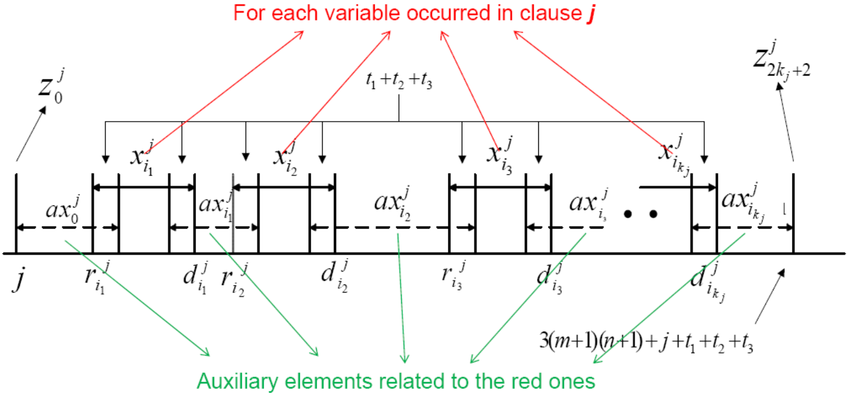
$K =$ Maximal packet length $N = |X|$

Re-aggregation: a packed packet can be dispatched for further packing

Complexity analysis

$K \geq 3$, P_0 is NP-hard in tree structures – a reduction from SAT problem




For each variable occurred in clause j
$t_1+t_2+t_3$
z_0^j
$z_{2k_j+2}^j$
$x_{i_1}^j$
$x_{i_2}^j$
$x_{i_3}^j$
$x_{i_{k_j}}^j$
ax_0^j
$ax_{i_1}^j$
$ax_{i_2}^j$
$ax_{i_3}^j$
$ax_{i_{k_j}}^j$
j
$r_{i_1}^j$
$d_{i_1}^j$
$r_{i_2}^j$
$d_{i_2}^j$
$r_{i_3}^j$
$d_{i_3}^j$
$d_{i_{k_j}}^j$
$3(m+1)(n+1)+j+t_1+t_2+t_3$
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 $\rightarrow$ P_1 is NP-hard $\rightarrow$ P_2 is NP-hard $\rightarrow$ 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 *pkt* with length s_f

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

$$AC = \frac{\#\ \text{of TX}}{\text{length of data}}$$

A utility based online algorithm

Utility of holding a packet:

Cost with packing

$$AC'_l = \frac{1}{L - s'_f} ETX_{jR}(L - s'_f) \qquad AC_l = \frac{1}{L - s'_f + S_l} ETX_{jR}(L - s'_f + S_l)$$

Cost without packing

$$U_l = AC'_l - AC_l$$

Utility of transmitting a packet:

$$U'_p = \frac{\frac{t'_f}{t_p} ETX_{p_jR}(s_p)}{\frac{t'_f}{t_p} s_p} - \frac{\frac{t'_f}{t_p} ETX_{p_jR}(L)}{\frac{t'_f}{t_p} L} = \frac{ETX_{p_jR}(s_p)}{s_p} - \frac{ETX_{p_jR}(L)}{L}$$

$$U''_p = \frac{\lceil \frac{L-s'_f}{L-s_p} \rceil ETX_{p_jR}(s_p)}{\lceil \frac{L-s'_f}{L-s_p} \rceil s_p} - \frac{\lfloor \frac{L-s'_f}{L-s_p} \rfloor ETX_{p_jR}(L) + I_{mod} ETX_{p_jR}(s_p + l_{mod})}{\lceil \frac{L-s'_f}{L-s_p} \rceil s_p + L - s'_f}$$

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

Every packet received by parent can get fully packed via ***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

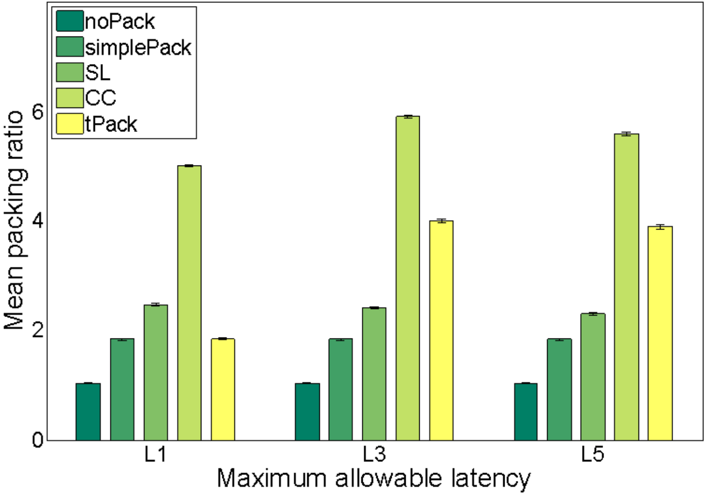
- Problem P'
  - T is a complete tree
  - Leaf nodes generate elements at a common rate
- Theorem: For problem P', *tPack* is $min\{K, max_{v_j \in V_{>1}} \frac{2ETX_{v_j R}}{2ETX_{v_j R} - ETX_{p_j R}}\}$-competitive, where K is the maximum number of information elements that can be packed into a single 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

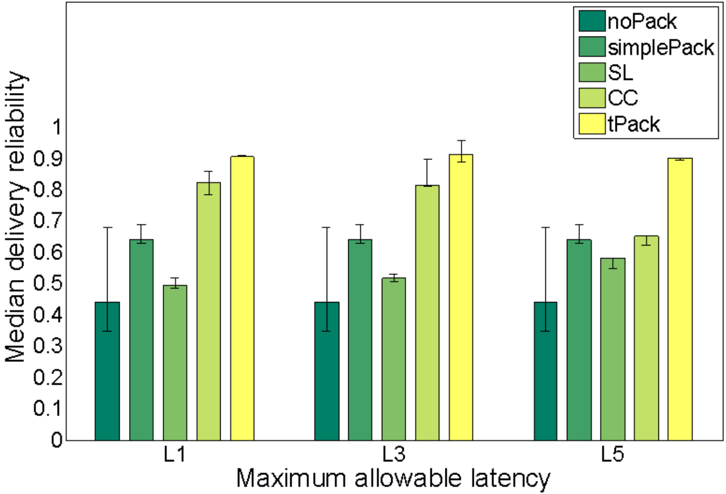
Experiment setting up

- 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

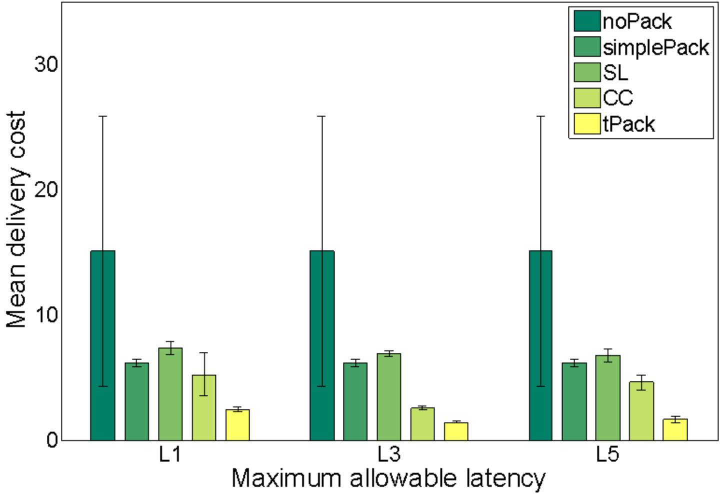
3-second periodic traffic: packing ratio



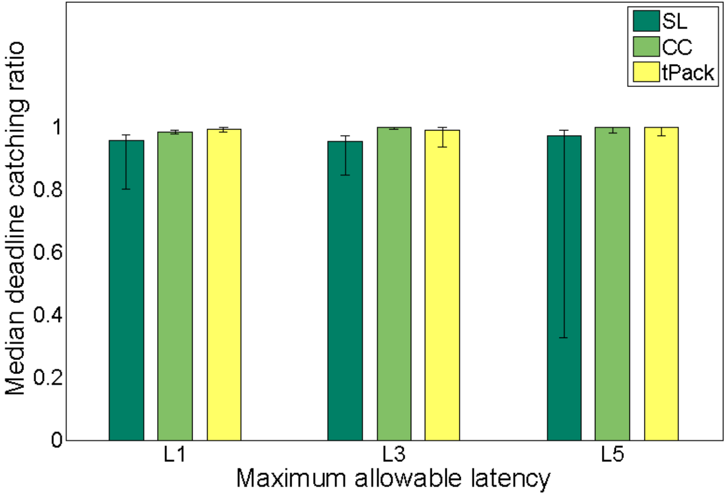
3-second periodic traffic: delivery reliability



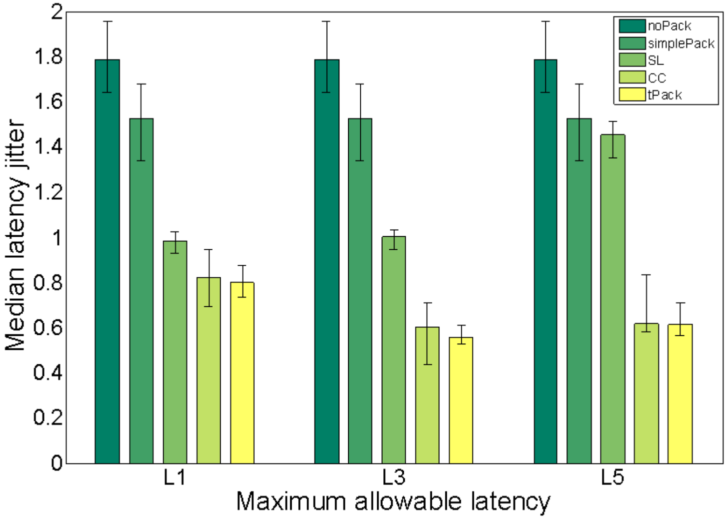
3-second periodic traffic: delivery cost



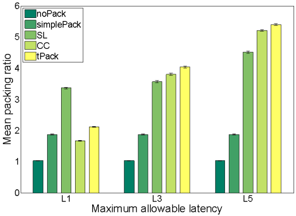
3-second periodic traffic: deadline catching ratio

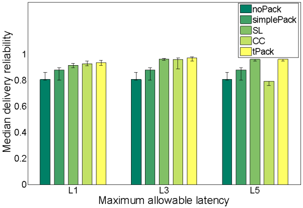


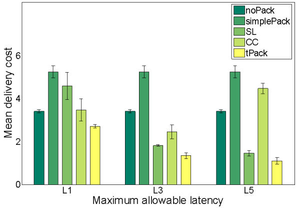
3-second periodic traffic: latency jitter

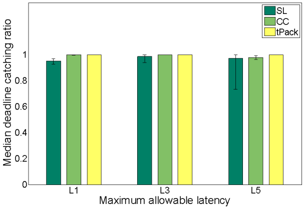


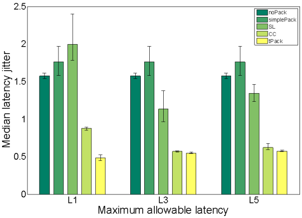
6-second periodic traffic



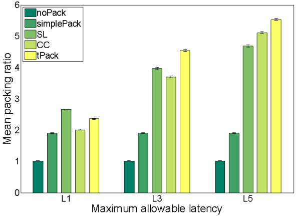


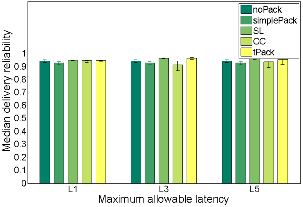


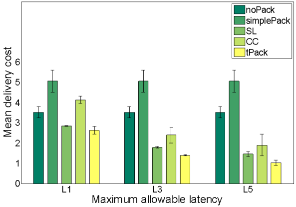


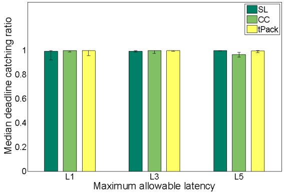


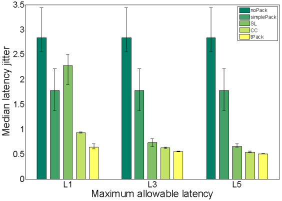
9-second periodic traffic



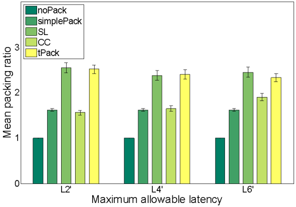


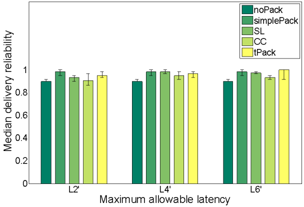


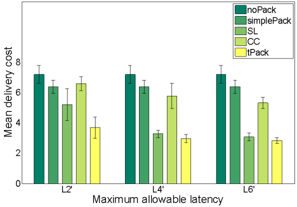


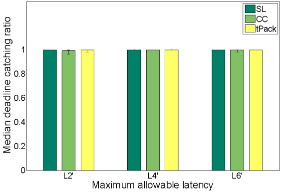


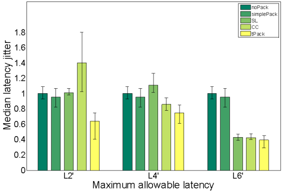
Event traffic











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

Inter-flow coding vs. intra-flow coding

Inter-flow coding

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

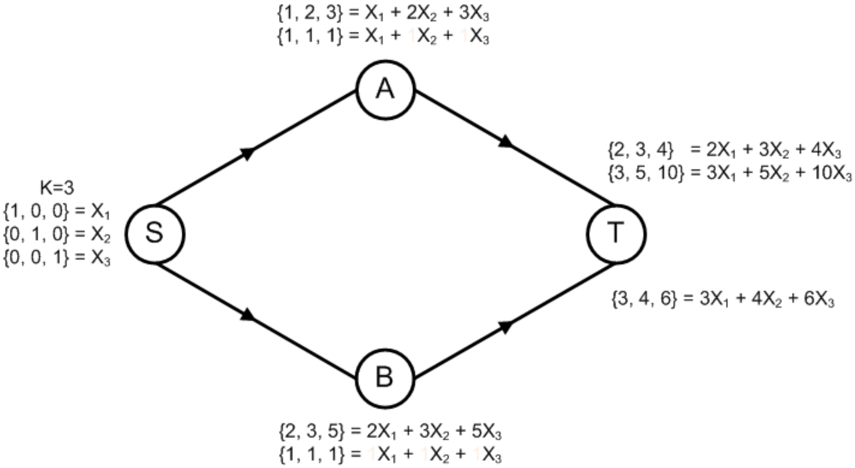
Intra-flow coding

- 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



Energy-efficient network coding based routing

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

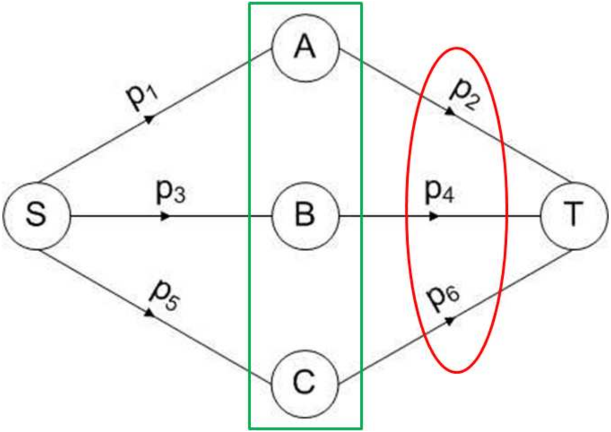
Definition

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



$$P_2 \geq P_4 \geq P_6$$

$$C_{SD_S}(K) = \frac{K}{P_{SV_{D_S}}} = \frac{K}{1-(1-P_1)(1-P_3)(1-P_5)}$$

$$K_A^S = \frac{KP_1}{1-(1-P_1)(1-P_3)(1-P_5)}$$

$$K_B^S = \frac{KP_3}{1-(1-P_1)(1-P_3)(1-P_5)}$$

$$K_C^S = \frac{KP_5}{1-(1-P_1)(1-P_3)(1-P_5)}$$

$$L_A = K_A^S$$

$$L_B = K_B^{S'} = K\frac{K_B^S}{K}(1-P_1) = K_B^S(1-P_1)$$

$$L_C = K_C^{S'} = K_C^S(1-P_1)(1-P_3)$$

$$
\begin{aligned}
C_S(K) &= C_{SD_S}(K) + C_{AT}(L_A) + C_{BT}(L_B) + C_{CT}(L_C) \\
&= \frac{K}{1-(1-P_1)(1-P_3)(1-P_5)} \\
&\quad + \frac{L_A}{P_2} + \frac{L_B}{P_4} + \frac{L_C}{P_6} \\
&= \frac{K}{1-(1-P_1)(1-P_3)(1-P_5)} \\
&\quad \cdot [1 + \frac{P_1}{P_2} + \frac{P_3(1-P_1)}{P_4} + \frac{P_5(1-P_1)(1-P_3)}{P_6}]
\end{aligned}
$$

Algorithm 1 Compute the transmission cost of NC-based routing for the current node S with M forwarder candidates

Input: current node S, $D_S = \{A_1, A_2, \ldots, A_M\}$

Output: $C_S(1)$: the expected number of transmissions to deliver 1 packet from S to T

Sort nodes in D_S by a non-descending order of $C_{A_i}(1)$, where $i = 1, 2, \ldots, M$.

Sorted nodes are labeled as $\{A'_1, A'_2, \ldots, A'_M\}$

$C_{SD_S}(1) = \frac{1}{1-\prod_{i=1}^{M}(1-P_{SA'_i})}$

$L_{A'_1} = C_{SD_S}(1)P_{SA'_1}$

$F = 1 - P_{SA'_1}$

for $i \to 2, 3, \ldots, M$ **do**

$\quad L_{A'_i} = C_{SD_S}(1)P_{SA'_i}F$

$\quad C_{A'_i}(L_{A'_i}) = L_{A'_i}C_{A'_i}(1)$

$\quad F = F(1 - P_{SA'_i})$

end for

$C_S(1) = C_{SD_S}(1) + \sum_{i=1}^{M} C_{A'_i}(L_{A'_i})$

Minimize the cost of NC-based routing

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

Input: node S, $D_S = \{A_1, A_2, \ldots, A_M\}$, $FS_S = \emptyset$
Output: $C_S^*(1)$: the minimal transmission cost to deliver 1 packet from S to T
Sort nodes in D_S by a non-descending order of $C_{A_i}(1)$, where $i = 1, 2, \ldots, M$.
Sorted nodes are labeled as $\{A_1', A_2', \ldots, A_M'\}$
$FS_S = \{A_1'\}$
$C_S^*(1) = \frac{1}{P_{SA_1'}} + C_{A_1'}(1)$
for $i \rightarrow 2, 3, \ldots, M$ **do**
    Run Algorithm 1 with input S and $D_S = \{A_1', \ldots, A_i'\}$
    Get the result as $C_S^{new}(1)$
    if $C_S^{new}(1) > C_S(1)$ **then**
        break
    else
        $FS_S = FS_S \cup A_i'$
        $C_S^*(1) = C_S^{new}(1)$
    end if
end for

Theorem of optimality

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.

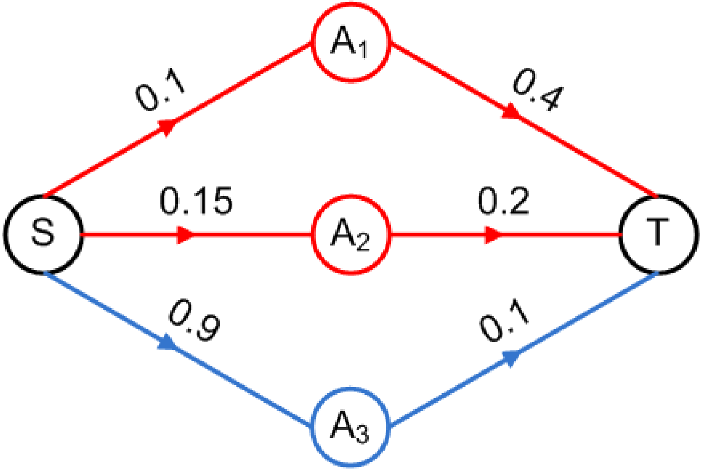
Properties of the optimal routing braid

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

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

Properties of the optimal routing braid



The optimal routing braid is $\{A_1, A_2\}$

$$
\begin{aligned}
C_{\{A_1,A_2\}} &= \frac{1}{1-(1-0.1)(1-0.15)} \cdot [1 + \frac{0.1}{0.4} + \frac{0.15(1-0.1)}{0.2}] \\
&= \frac{1}{0.235} \cdot (1 + \frac{1}{4} + \frac{0.135}{0.2}) \\
&= 8.1915 \\
C_{\{A_1,A_2,A_3\}} &= \frac{1}{1-(1-0.1)(1-0.15)(1-0.9)} \\
&\quad \cdot [1 + \frac{0.1}{0.4} + \frac{0.15(1-0.1)}{0.2} + \frac{0.9(1-0.1)(1-0.15)}{0.1}] \\
&= \frac{1}{0.9235} \cdot (1 + \frac{1}{4} + \frac{0.135}{0.2} + \frac{0.6885}{0.1}) \\
&= 9.5398 > C_{\{A_1,A_2\}}
\end{aligned}
$$

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.

Protocol design and implementation

EENCR: an energy-efficient NC-based routing protocol

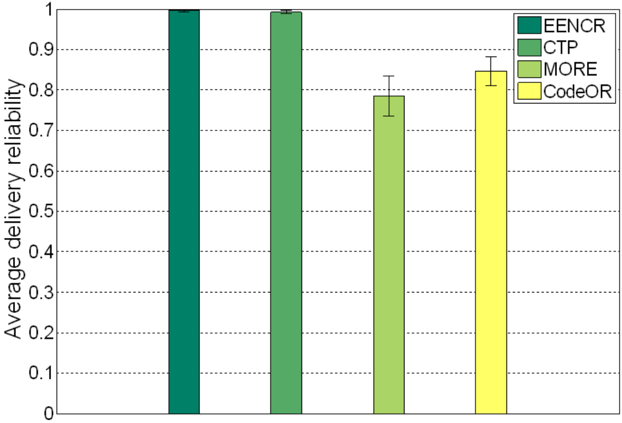
- Routing engine: a distributed implementation of Algorithm 2
- M-NSB: a coded ACK scheme to solve the collective space problem with lower implementation complexity than CCACK
- Rate control: nodes forward a flow after receiving a load-dependent threshold of packets to 1) reduce contention and 2) avoid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

Performance evaluation

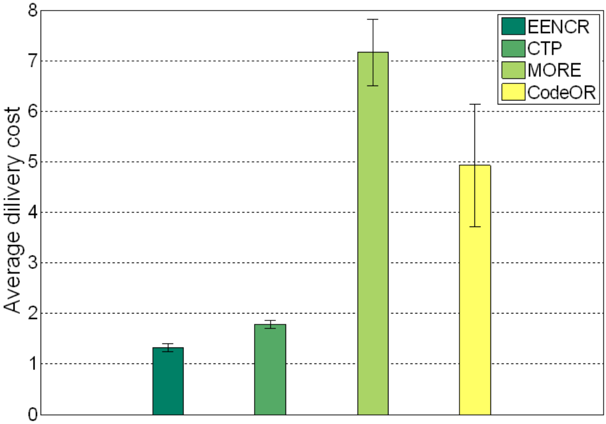
Experiment setting up

- 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

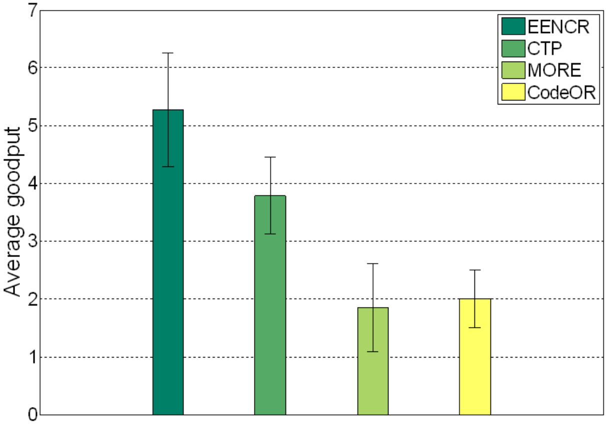
10-source: delivery reliability



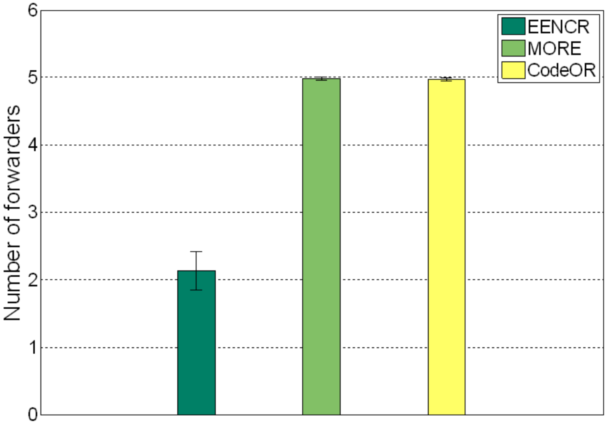
10-source: delivery cost



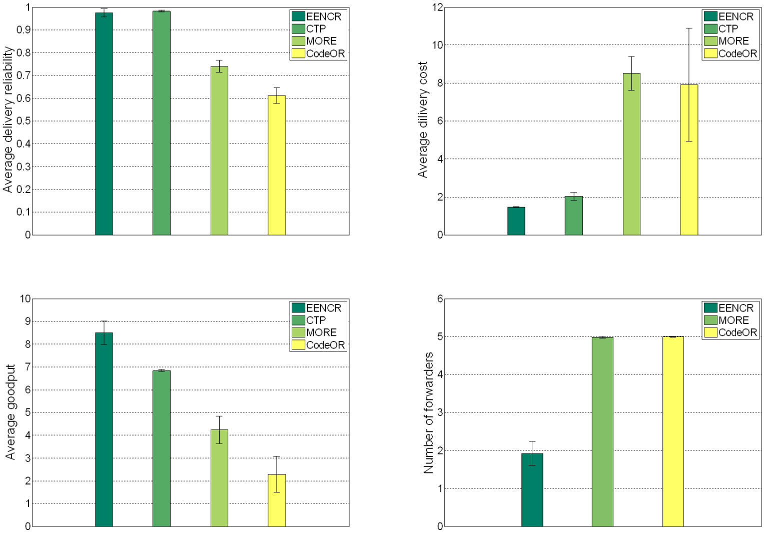
10-source: goodput



10-source: routing diversity



20-source



Average delivery reliability
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
EENCR
CTP
MORE
CodeOR
Average dilivery cost
0
2
4
6
8
10
12
EENCR
CTP
MORE
CodeOR
Average goodput
0
1
2
3
4
5
6
7
8
9
10
EENCR
CTP
MORE
CodeOR
Number of forwarders
0
1
2
3
4
5
6
EENCR
MORE
CodeOR

Remarks

- 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

Proactive network coding based protection

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 $\mathbf{Q}$

- 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 $\mathbf{Q}'$

- The same as $\mathbf{Q}$ 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 $\mathbf{Q}'$

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

such that

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

Complexity analysis

Lemma

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

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

Theorem

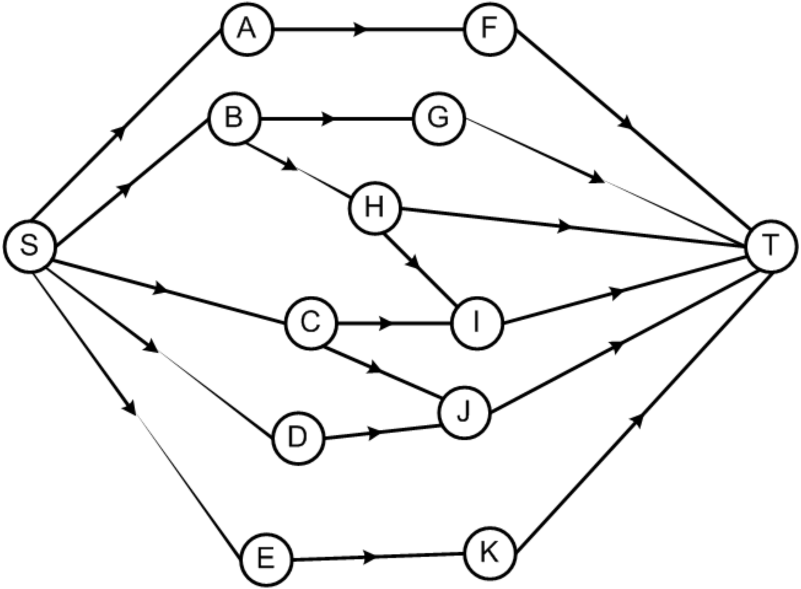
Problem $\mathbf{Q}$ *is NP-hard.*

An immediate result from the NP-hardness of $\mathbf{Q}'$.

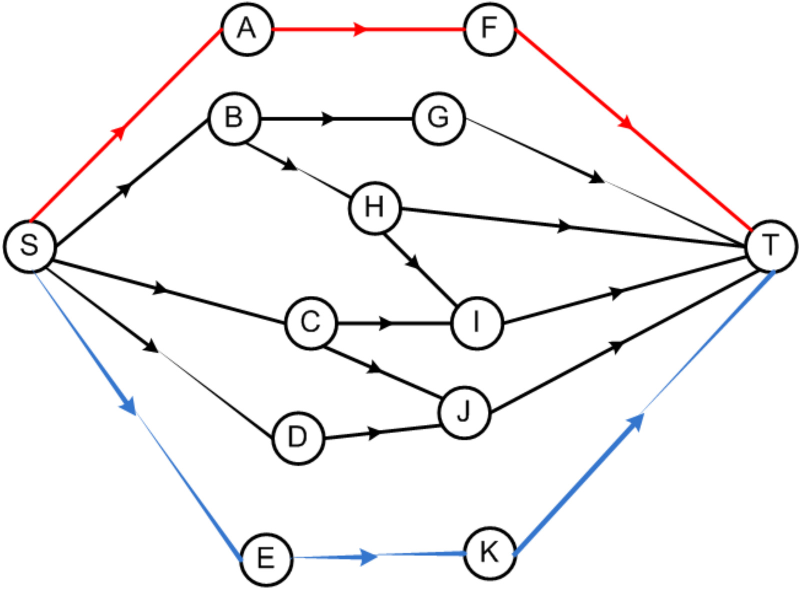
A heuristic algorithm for Problem Q

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

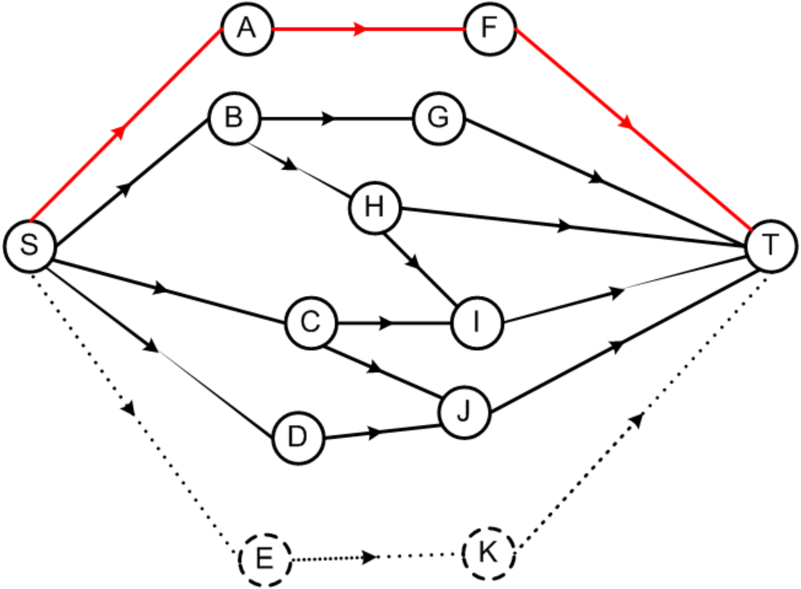
Finding node-disjoint braids - an example



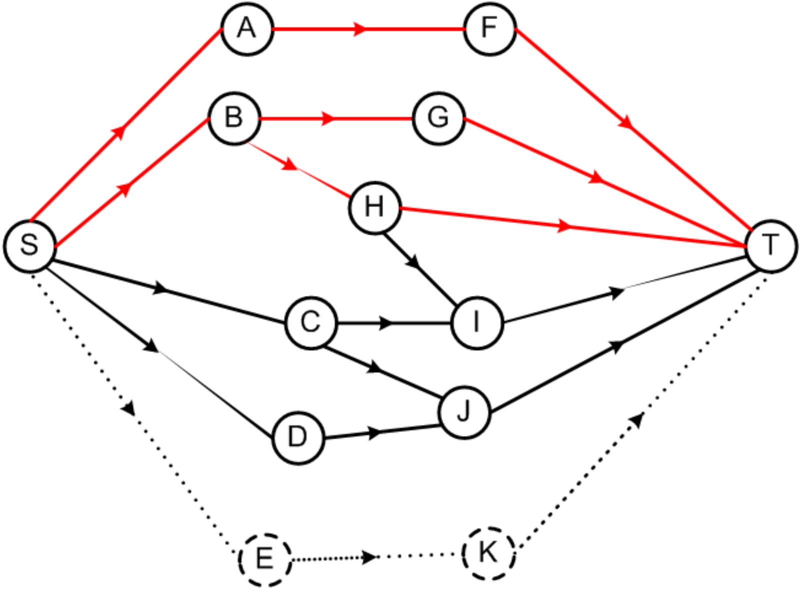
Finding node-disjoint braids - an example



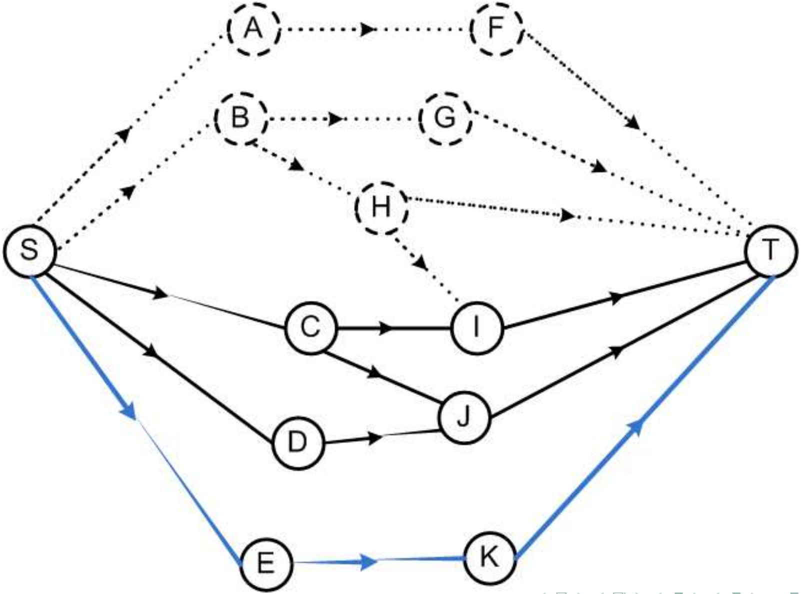
Finding node-disjoint braids - an example



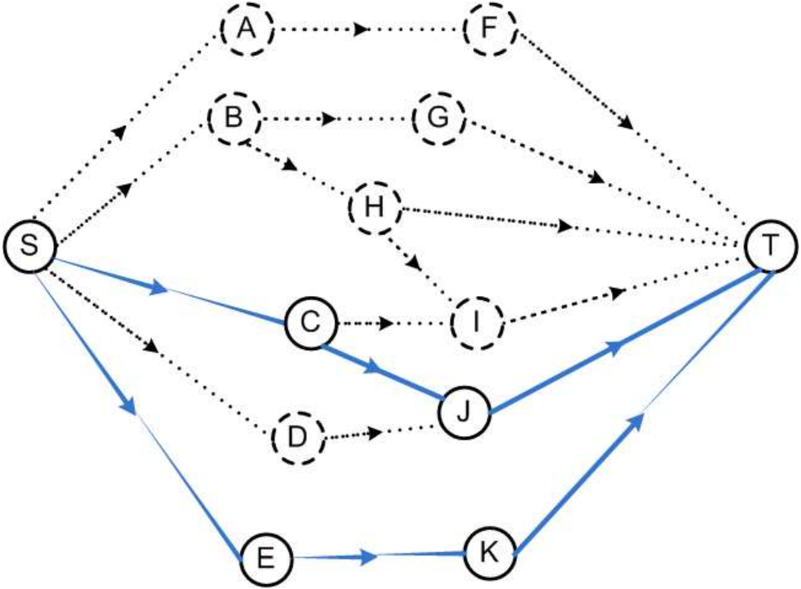
Finding node-disjoint braids - an example



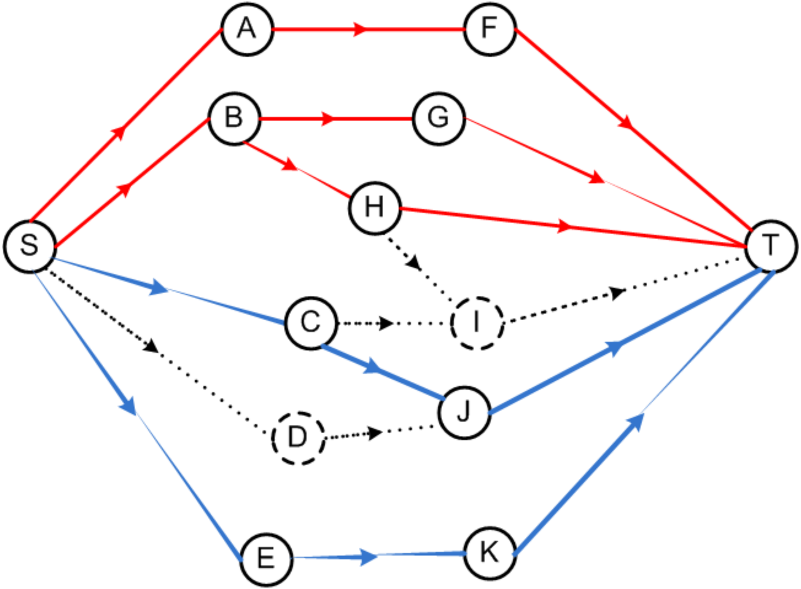
Finding node-disjoint braids - an example



Finding node-disjoint braids - an example



Finding node-disjoint braids - an example



Algorithm 3 A heuristic algorithm for two node-disjoint braids construction

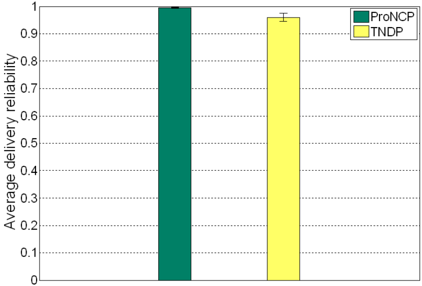
Input: a DAG $G = (V, E)$ with source S and destination T
Construct 2 minimal cost node-disjoint paths $\{R_1, R_2\}$ from S and T, where $C_{R_1} \geq C_{R_2}$
$B_1 = R_1, B_2 = R_2$
$G_1 = G$
for every node V_i in G_1 **do**
    if $V_i \in B_2$ **then**
        Remove V_i and all links attached to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

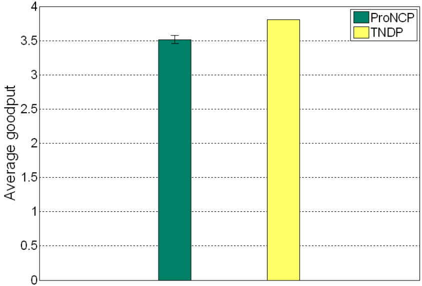
Performance evaluation

Experiment setting up

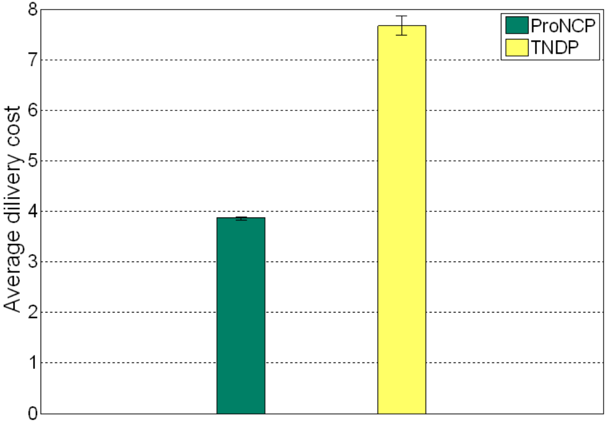
- Testbed: NetEye, a 130-sensor testbed (Maccabees Building)
- Topology: 60 nodes, 10 are source nodes, 1 sink node
- Protocols compared: *ProNCP*, TNDP (two node-disjoint path protection)
  Note: braids and routes are computed offline via sampling results.
- Traffic pattern: 3-second periodic traffic
- Failure model: probabilistic transient failure on intermediate nodes
- Metrics: delivery reliability, delivery cost and goodput

No failure: delivery reliability and goodput

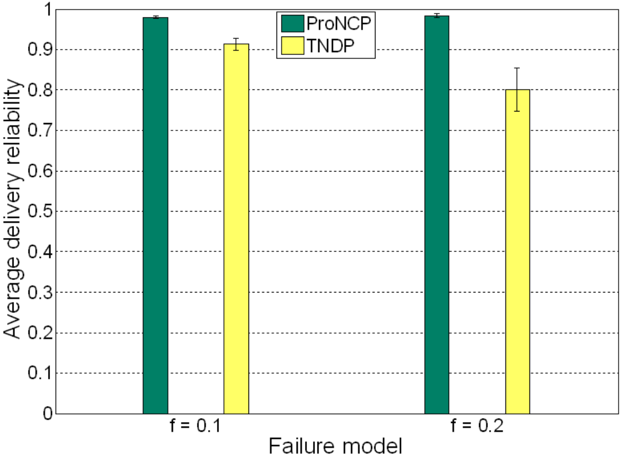




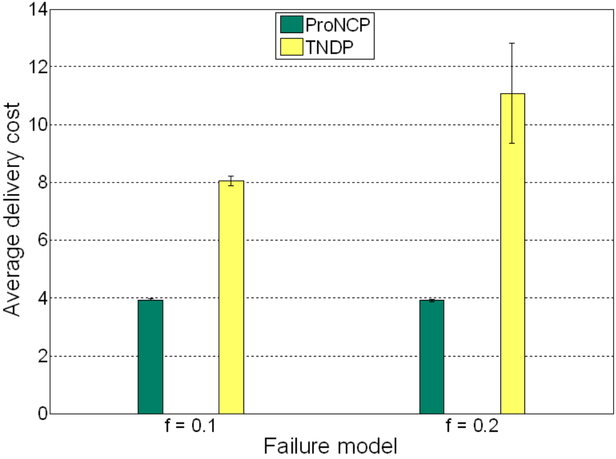
No failure: delivery cost



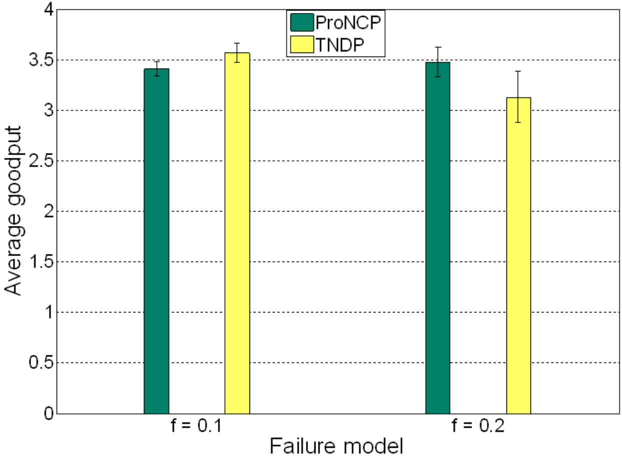
Transient failure: delivery reliability



Transient failure: delivery cost



Transient failure: goodput



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

Outline

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

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**

Acknowledgments

- 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.

Acknowledgments

- 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.

Acknowledgments

- 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.