Load-Optimal Local Fast Rerouting for Resilient Networks

Yvonne-Anne Pignolet, ABB Corporate Research
Stefan Schmid, University of Aalborg
Gilles Trédan, LAAS-CNRS, Toulouse
Motivation

- Critical infrastructure has high availability requirements
- Industrial systems are more and more connected
- Hard real-time requirements

⇒ How to provide dependability guarantee despite link failures in networks?
⇒ Possible without communication between nodes? And low load?
Local Fast Failover

Traffic demand: {1,2,3}->6

Local failover @1: Does not know failures downstream!
Local Fast Failover

Traffic demand: {1,2,3} -> 6

Failover matrix:
flow 1->6: 2,3,4,5,…

Local failover @1:
Reroute to 2!
Local Fast Failover

Traffic demand: \{1,2,3\} -> 6

Failover matrix: flow 1->6: 2,3,4,5,…

Local failover @1: Reroute to 2!

But also from 2: 6 not reachable. Next: 3.
Local Fast Failover

Failover matrix:
flow 1→6: 2,3,4,5,…
flow 2→6: 3,4,5,…
flow 3→6: 4,5,…

Traffic demand: 
{1,2,3}→6

Max load: 
3 😞
Local Fast Failover

Failover matrix:
flow 1->6: 2,5, ...
flow 2->6: 3,4,5,...
flow 3->6: 4,5,...

Statically defined, no global knowledge and no communication!

A better solution: load 2 😊
Local Fast Failover

Failover matrix:
- flow 1->6: 2,5, ...
- flow 2->6: 3,4,5, ...
- flow 3->6: 4,5, ...

For load balance the prefixes should differ

A better solution: load 2 😊
Find a failover matrix $M$ that needs many link failures for a high load

**Problem statement**

Row $i$ used for flow $i$, each row is a permutation, source and destination are ignored

1: Upon receiving a packet of flow $i$ at node $v$
2: If $v \neq$ destination:
3: If $(v, \text{destination})$ available: forward to $d$
4: $j = \text{index of } v \text{ in } \text{ith row}, \quad /*m_{i,j} = v*/$
5: While $m_{i,j} = \text{source or (v,m_{i,j}) unavailable}$
6: $j = j+1$
7: Forward to $m_{i,j}$
Good and bad news  [BS,Opodis 2013]

Lower bound:
High load unavoidable even in well-connected residual networks:
# failures $\varphi$ can lead to load at least $\sqrt{\varphi}$, even in highly connected networks

Example: All-to-One Traffic

Upper bound:
Load $\sqrt{\varphi}$ generated with a failover matrix where each row is a random permutation needs at least Omega($\varphi / \log n$) failures.
Deterministic Failover Matrices

Construction goal: Low intersection in short prefixes!
**Latin Squares with Low Intersection**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>4</th>
<th>12</th>
<th>9</th>
<th>3</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>12</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>...</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>13</td>
<td>8</td>
<td>11</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>4</td>
<td>x</td>
<td>9</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>4</td>
</tr>
</tbody>
</table>

If \( k < \sqrt{n} \) a latin square failover matrix where the intersection of two \( k \)-prefixes is at most 1 has load \( \varphi < k \) with \( \Omega(\varphi^2) \).

Can we construct such matrices?
Symmetric Balanced Incomplete Block Designs

An \((n, k, \lambda)\)-BIBD consists of

- Set \(X\) with \(n\) elements \(\{1, \ldots, n\}\)
- Blocks \(A_1, \ldots, A_n\), containing \(k\) elements of \(X\)
- \(|A_i \cap A_j| = \lambda\)

Hall’s Marriage Theorem:

A \(d\)-regular graph contains \(d\) disjoint perfect matchings

\(\begin{array}{c}
\text{(7,3,1)-BIBD} \\
1,2,7 \quad 1,3,6 \\
2,5,6 \quad 2,4,3 \\
1,4,5 \quad 3,5,7 \\
4,7,6
\end{array}\)

3-regular bipartite graph
Construction of $k$-prefix with low intersection

1, 4, 13
12, 3, 4
9, 1, 2
8, 6, 12

1 4 13 9 3
3 12 4 5 6
2 1 9 7 8
12 8 6 3 13

$n-k$
Results

Theory:
Deterministic BIBD-Failover Matrix achieves asymptotically optimal load

Experiments:
All-to-one routing and random failures  Permutation routing and random failures
Conclusion and Future Work

- Deterministic failover with **load guarantees**
  applying latin squares, BIBDs, matchings

- BIBDs are a tool that can probably be used in many other contexts

Next:
Algorithms and improved bounds for sparse communication networks