Cycle-Oriented Distributed Pre-configuration:

*Ring-like Speed with Mesh-like Capacity for Self-Organizing Proactive Network Restoration*

Wayne D. Grover and Demetrios Stamatelakis

grover@trlabs.ca (403 441 3815)  dstamate@edm.trlabs.ca

TRLabs and The University of Alberta

ICC ‘98 June 7-11 Atlanta, Georgia
Outline

- Background and Objectives
- Restoration using Pre-configured Cycles (p-Cycles)
  - Comparison of p-Cycle and SONET Rings
- Optimal Design of Spare Capacity for p-Cycle restoration
  - Self-organization of the p-Cycle configuration
- Significance and Benefits
## Background and Motivation

<table>
<thead>
<tr>
<th>“Ring”</th>
<th>“Mesh”</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. 50 - 60 msec</td>
<td>E. 100 msec - 2 sec. typical</td>
</tr>
<tr>
<td>restoration time</td>
<td></td>
</tr>
<tr>
<td>B. complex capacity</td>
<td>F. simple, exact spare capacity</td>
</tr>
<tr>
<td>design &amp; network</td>
<td>planning</td>
</tr>
<tr>
<td>planning</td>
<td></td>
</tr>
<tr>
<td>C. high total capacity</td>
<td>G. well under 100% spare capacity</td>
</tr>
<tr>
<td>for service provided</td>
<td></td>
</tr>
<tr>
<td>D. simple, low-cost</td>
<td>H. Relatively expensive</td>
</tr>
<tr>
<td>ADMs</td>
<td>DCS nodes</td>
</tr>
</tbody>
</table>
Background: Mesh Restorable Networks

Node X for failure 1
Node X for failure 2
Node X for failure 3
Node X for failure 4

Q. How could you ever have the spare capacity of a mesh network completely pre-connected in advance of any failure?
Restoration using $p$-Cycles

A. Form the spare capacity into a particular set of pre-connected cycles!

A $p$-cycle

A span on the cycle fails - 1 Restoration Path, BLSR-like

A span off the $p$-cycle fails - 2 Restoration Paths, Mesh-like
Optimal Spare Capacity Design for $p$-Cycle Restoration: Method

- **Step 1:** Find set of elementary cycles of the network graph

- **Step 2:** For each cycle, determine $x_{i,j}$: the no. of restoration paths that cycle $i$ contributes for failure $j$.

- **Step 3:** Integer Program:
  
  **Objective:** minimize: total cost of spare capacity.

  **Subject to:**
  
  1. All working links on each span have (simultaneously feasible) access to one or more $p$-cycles.
  2. All $p$-cycles placed are feasible within the span spare capacities assigned
**Optimal Spare Capacity Design: Results**

- “Excess Sparing” = Spare Capacity compared to Optimized Span-Restorable Mesh:

<table>
<thead>
<tr>
<th>Net</th>
<th>Excess Sparing</th>
<th># of unit-capacity $p$-cycles formed</th>
<th># of Unique cycles used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net1</td>
<td>9.09 %</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Net2</td>
<td>3.07 %</td>
<td>88</td>
<td>10</td>
</tr>
<tr>
<td>Net3</td>
<td>0.0 %</td>
<td>250</td>
<td>10</td>
</tr>
<tr>
<td>Net4</td>
<td>2.38 %</td>
<td>2237</td>
<td>27</td>
</tr>
<tr>
<td>Net5</td>
<td>0.0 %</td>
<td>161</td>
<td>39</td>
</tr>
</tbody>
</table>
Comparing the $p$-Cycle Concept to Rings

<table>
<thead>
<tr>
<th>Attribute</th>
<th>$p$-cycles</th>
<th>SONET rings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Modularity</strong></td>
<td>One spare capacity signal unit</td>
<td>OC-n modularity</td>
</tr>
<tr>
<td><strong>Protection Yield</strong></td>
<td>Up to two paths per $p$-cycle</td>
<td>One restoration path per use</td>
</tr>
<tr>
<td><strong>Protection Flexibility</strong></td>
<td>$p$-cycles contribute to restoration of on-cycle and “straddling” failures</td>
<td>Rings only protect spans on the same ring</td>
</tr>
<tr>
<td><strong>Routing and provisioning of working paths</strong></td>
<td>Proceeds without regard to structures formed in the sparing layer</td>
<td>Working path routing must conform to ring systems and limited inter-ring transfer points</td>
</tr>
<tr>
<td><strong>Total Network Redundancy</strong></td>
<td>~ span restorable mesh ($&lt; 100%$)</td>
<td>Over 100% investment in spare capacity. Up to 300%</td>
</tr>
</tbody>
</table>
Comparing $p$-Cycle Concept to Rings

SPARE

WORKING Coverage

BLSR or UPSR

$p$-Cycle
Self-organizing formation of p-cycles
(DCPC Protocol)

**GOAL:** Have the network determine, and continually adapt, its set of p-cycles for maximum failure-readiness

- Based on modified Tandem node rules from Selfhealing Network (SHN) protocol
- Operates autonomously, continuously, in background, on spare links only.
- A “Pro-Active”, *non- real time process* - Somewhat like “distributed pre-planning” use of a restoration algorithm but...
- Determines not only what to do upon failure, but also pre-operates the cross-connections between spares in advance.
- Upon failure, the only real-time action is to make pre-armed traffic substitution connections
- No real time signalling requirement if p-cycles kept under “audit” while in storage
DCPC Tandem Node Broadcast Rules

- Incoming statelets rebroadcast to the largest extent possible
- subject to:  - Competition based on incoming “scores”
  - Re-Broadcast “direction rules”

```
A  F  G
B   E
C  D

“home”  “new nodes”

“been there”

Cycler Node
Tandem Node
Existing Relationship Trail
Permitted
Invalid
```
Tandem Node Competition Rules

- Score is determined by:
  \[
  \text{Score} = \frac{\text{Useful Restoration Paths}}{\text{Links used}}
  \]

Re-Broadcast Competition

- 4 Incoming Statelets
- 3 Broadcast Families
- Statelet Ranking by Score:
  \[s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4\]
Tandem Node Evaluation of Statelet Score

Path evaluation for a Statelet broadcast Arriving at a Tandem Node

Statelet Broadcast Route: C-1-2-T

Case 1: No Paths
Failure of Span T-3

Case 2: 1 Paths
Failure of Span T-2

Case 3: 2 Paths
Failure of Span T-1

Case 4a: 2 paths
Broadcast to a Tandem
Failure of Span T-C

Case 4b: 1 paths
Broadcast to the Cycler
Failure of Span T-C
Tandem Node Formation of “Best” Cycles

How Path count is updated Incrementally for a Statelet Broadcast

- Highest score cycle emerges under Tandem node competition rules

- “Useful” paths \{C-1-2-3-4-C\} = 1 + 1 + (1+2+2) + 1 + 1 = 9

- links used = 5

- “score” = 9/5
DCPC Protocol: Performance

• OPNET Modeler Simulation experiments

• Tested in Stringent Minimal-Capacity Network Designs:

• Results:

<table>
<thead>
<tr>
<th>Network</th>
<th>$p$-cycle Restorability (%)</th>
<th>2-step Restorability (%)</th>
<th>With OC-n Modularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net1</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Net2</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Net3</td>
<td>90.94</td>
<td>97.16</td>
<td>91.53 / 98.49</td>
</tr>
<tr>
<td>Net4</td>
<td>89.16</td>
<td>97.68</td>
<td>100</td>
</tr>
<tr>
<td>Net5</td>
<td>83.75</td>
<td>95.44</td>
<td>95 / 100</td>
</tr>
</tbody>
</table>
The Real-time phase of Restoration with $p$-cycles

Example: 5 links Fail

$p$-cycle 1: restores 1 Path

$p$-cycle 2: restores 2 Paths

$p$-cycle 3: restores 2 Paths

$p$-cycle 4: not used

Total Restoration Paths = 1 + 2 + 2 + 0 = 5
Concluding Discussion: Significance and Benefits

- The $p$-cycle concept offers the prospect of ring-like speed, with mesh-like efficiency.

- The key to the mesh-like efficiency: $p$-cycles protect straddling failures as well as on-span failures.

- $p$-cycles leave the working capacity free to be routed without constraints from protection structures (unlike rings).

- $p$-cycles can be centrally computed or self-organized by the network (DCPC Protocol): Proactive in two senses

  1. Computation of restoration planning is completed Prior to Failure
  2. Restoration Path-formation is also completed Prior to Failure

- $p$-cycles may be kept under constant readiness testing. Each node then knows (a) restoration action, (b) restoration path status, and (c) restoration level to expect, all in advance.

- Possible applications in WDM optical networking