Successive Survivable Routing for Fault Tolerant Communication Networks

Yu Liu* and D. Tipper *
Department of Information Science and Telecommunications
University of Pittsburgh
Pittsburgh, PA U.S.A

*supported in part by grants from NSF ANIR and DARPA

Motivation

• Why do communications networks need to be survivable?
• Communication Networks are Critical Infrastructure (CI) (PCCIP 1996) the systems, assets and services upon which society and the economy depend
• Communication infrastructure often considered most important CI due to reliance on it by other infrastructures
  – banking and finance, government services
  – power grid SCADA, etc.
• Increasing Impact of Failures
  – Increased bandwidth of links (WDM technology in fiber optic network)
  – Increased societal dependence

Causes of Network Outages

• According to Sprint a link outage in IP backbone every 30 min on average
• Accidents
  – cable cuts, car wreck, etc.
  – According to AT&T 4.39 Cable cuts / year / 1000 miles
• Human errors
  – incorrect maintenance, installation
• Environmental hazards
  – fire, flood, etc.
• Substantive
  – physical, electronic
• Operational disruptions
  – schedule upgrades, maintenance, power outage
• Hardware/Software failures
  – Line card failure, faulty laser, software crash, etc.

Network Survivability

• Definition
  – Ability of the network to support the committed Quality of Services (QoS) continuously in the presence of various failure scenarios
• Survivability Components
  – Analysis: understand failures and system functionality after failures
  – Design: adopt network procedures and architecture to prevent and minimize the impact of failures/attacks on network services.
  – Goal: maintain service for certain scenarios at reasonable cost
• Self - Healing network

Survivable Network Design

• Three steps towards a survivable network
  1. Prevention:
     – Robust equipment and architecture (e.g., backup power supplies)
     – Security (physical, electronic), Intrusion detection, etc.
  2. Topology Design and Capacity Allocation
     – Design network with enough resources in appropriate topology
     – Spare capacity allocation – to recover from failure
  3. Network Management and traffic restoration procedures
     – Detect the failure, and route traffic around failure using the redundant capacity

Survivable Network Design

• Spare Capacity Allocation (SCA) Problem:
  – given working paths and network (or virtual network) topology
  – provision spare capacity and find backup routes for fault tolerance
  – Goal: minimum spare capacity or cost
• Survivable Mesh Networks
  – Consider preplanned protection in mesh networks
  – STM - DCS, ATM - VP, WDM, MPLS, etc.
  – determine routing/capacity allocation for normal demand
  – find location and amount of spare capacity for failure scenarios
  – spare capacity required depends on restoration/survivability technique
Restoration techniques

- Types of restoration schemes
  - link (span) restoration
  - path restoration:
    - Failure dependent (FD), with stub release
    - Failure independent (FID)

Previous Work

- Mesh Network Spare Capacity Design
  - Optimization Techniques
    - Integer Programming Models – NP Hard – scaling problems
    - Heuristics
      - Find feasible solution for fault scenario
  - Drawbacks
    - Ignore Modularity of cost
    - Scalability of optimization techniques
    - Linear variables (difficult to model nonlinear cost)

Present a matrix based formulation and a new heuristic
- fast, near optimal, nonlinear cost and different restoration schemes

SCA Problem

- SCA for Failure Independent Shared Backup Path Restoration
- Notation
  - \( r = 1, 2, \ldots, D \) set of demands (source-destination pairs)
  - \( p = 1, 2, \ldots, P_r \) set of possible paths for demand pair \( r \)
  - \( l = 1, 2, \ldots, L \) set of network links

- Input parameters (constants)
  - \( \alpha_r \) offered traffic load of demand pair \( r \)
  - \( c_l \) unit cost of capacity on link \( l \)
  - \( \delta^r_{p,l} = 1 \) if \( l \) belongs to path \( p \) realizing demand \( r \)
    - \( 0 \), otherwise
  - \( f \) set of link failure scenarios

- Variables
  - \( x^r_p \) flow of demand \( r \) on path \( p \)
  - \( s_l \) spare capacity on link \( l \)

Find \( s_l \) and \( x^r_p \), which

\[
\text{minimize} \quad \sum_{l \in L} c_l \cdot s_l \\
\text{subject to} \quad \sum_{p \in P} x^r_p = 1, \forall r \in D \\
\sum_{r \in D} \sum_{p \in P} \delta^r_{p,l} \cdot x^r_p \leq s_l, \quad \forall l \in L - \{f\}, \forall f, f \in L
\]

SCA Path-flow model

Matrix Based Formulation of SCA

- Matrix Based formulation of Optimization model for FID shared backup path restoration
- Consider path incident matrices \( P \) and \( Q \) for working and backup paths where each matrix has
  - number of rows = number of flows in the network
  - number of columns = number of links in the network
  - row \( i \) in the matrix \( P \) corresponds to the set of links used by flow \( i \)
  - similarly row \( i \) in the matrix \( Q \) corresponds to the set of backup path links used by flow \( i \)
- Relate to spare provision matrix \( G \), and spare capacity reservation \( s \)
  - \( G = Q^T P \), element \( G_{ij} \) gives required spare capacity on link \( j \) when link \( j \) fails
  - \( s = \max(G) \), or \( s \leq G \), spare capacity reservations are the maximum spare capacity for any single link failure

Example

From working and backup paths, \( G = Q^T P \)

From \( G \), \( s = \max(G) \)

Example: when link 2 fails, 

Spare capacity on backup path link 17
Arc-flow model of SCA

\[ \min \quad s = e^T s \quad \text{Total spare capacity} \]
\[ \text{s.t.} \quad s \geq G \quad \text{spare capacity on each link} \]
\[ G = Q^T M P \quad \text{Calculation of spare provision matrix} \]
\[ P + Q \leq 1 \quad \text{Link-disjointed backup paths} \]
\[ Q B^T = D \quad \text{(mod 2)} \quad \text{Flow conservation of backup} \]
\[ Q \text{ is a binary matrix} \]

Decision variable: \( Q, G, s \)

Given: \( P \) – working path link incidence matrix
\( B \) and \( D \) – node-link & flow-node incidence matrices
\( M \) – traffic demand matrix

Another way to find \( G \)

\[ G = \sum_r G_r, \quad \text{where } G_r = q_r^T p_r, \quad p_r \text{ and } q_r \text{ are vectors for working and backup paths of flow } r \]

Approximation algorithm

- Decomposition
  - multi-commodity flow \( \rightarrow \) multiple single flows
- Using shortest path algorithm for each flow to
  - route link-disjointed backup paths
  - using spare provision matrix \( G \) to calculate
  \[ \text{link cost} = \text{incremental spare reservation } v_r; \]
- Flows successively update their backup paths
  \( \rightarrow \) termed: *successive survivable routing* (SSR)

Find spare capacity \( s \)

From \( Q^T \) and \( P \), get \( G \)

From \( G \), get \( s \)

For \( r = 2 \), find \( G_r = q_r^T p_r \)

Link cost and local objective

- **Goal:** Each flow seeks a new backup path with minimal additional reservation
- **Additional reservation as link cost:**
  - Let \( G^+ = (e-p)^T p \)
  - and \( s^+ = \max (G^+ + G) \)
  - \( (e-p) \) assumes that a backup path uses all possible links
  - \( s^+ \) is a temporary spare capacity reservation vector
  - Additional spare reservation \( v_r = s^+ - s \)
  - \( v_r \) tells how much additional spare capacity needed if a link is used on a new backup path
  - Run shortest path with link cost \( v_r \)

Example of link cost

Assume backup path are using all possible links
\( e = (e-p)^T p \)

Find the contribution
\( G^+ = (e-p)^T p \)
SSR flowchart of flow $r$

1. Given $p_r$ and $d_r$
2. Periodically update $G$
3. Calculate $v_r$
4. Update $q_r$ using $v_r$
5. Update $s_r$ and $G$

- On source node of flow $r$:
  - $p_r, q_r$: working and backup path vectors
  - $d_r$: destination node
  - $G, s_r$: spare provision matrix and spare reservation vector
  - $v_r$: incremental spare reservations as link cost
- Stop after no backup path update on the network

Complexity

- Polynomial running time
  - shortest path algorithm for each flow, $O(N^2)$
  - Limited backup path update iterations for each flow
- Polynomial space complexity
  - Advertised information in $O(L^2)$
  - No per-flow based information

Numerical comparison

- Compare different algorithms and bounds
  - RAFT: Resource aggregation fault tolerance
  - SPI: Sharing with partial information
  - SR: Survivable routing (SSR without iteration)
  - SSR: Successive survivable routing
  - SA: Simulated annealing
  - BB: Branch and bound on a path-flow model – optimal
  - LP: Linear programming lower bound
- Metrics:
  - % Redundancy = spare capacity/working capacity,
  - execution time

Experiment networks

Network node degree ranges from 2.31 to 4.4
Consider balanced mesh load case

Redundancy versus Time on Network 3

- SSR, SR, SPI have 64 random cases with different flow orders
- Range of solutions
- Time is the sum of time to compute all 64 cases

Typical SSR results

- RAFT: Resource aggregation fault tolerance
- SPI: Sharing with partial information
- SR: Survivable routing (SSR without iteration)
- SSR: Successive survivable routing
- SA: Simulated annealing
- BB: Branch and bound on a path-flow model – optimal
- LP: Linear programming lower bound
Conclusions

- **Redundancy**
  - $LP \leq BB \leq SA \leq SSR \leq SR \ll SPI \equiv RAFT$
- **Time**
  - $BB > (LP, SA) >> SSR \geq SR \geq SPI \geq RAFT$
- **SSR has best trade-off**
  - Near optimal
  - Fast

Non-linear link cost

- Objective changed to:
  - $\min \sum \phi(s_i)$
  - decision variable: $Q$
- Numerical results using modular link cost
  - Link dimensioning comes in trunks: OC3, OC48, OC192, etc.

Typical Nonlinear Cost Results

- Failure dependent path restoration
  - Each column $G_i$ is decided by column vector $P_i$ and $Q_j$, backup path link adjacency matrix for link $i$ failure
  - $G_i = Q_i^T M P_j$, $1 \leq j \leq L$
  - $G_r = m_r q_r^T p_j$, $1 \leq j \leq L$, $1 \leq r \leq R$
  - $G = \sum G_r$
  - Stub release add $-1$ in the working link locations in $Q_j$ except link $i$

Comparison between restoration schemes

Summary

- Matrix based model reveals the structure of the spare capacity allocation problem
- Approximation – successive survivable routing
  - Partitioning multi-commodity to single flow
  - Using shortest path algorithm with special link cost
  - Update backup paths iteratively
- Extensions for nonlinear cost and different restoration schemes
  - Speed and near optimality of SSR => use in distributed implementation for automatic dynamic preplanning