Coloring SONET Rings and Related Things

Joint work with Steve Cosares and Iraj Saniee

Prepared For:
IMA Industrial Affiates

Telcordia Contact:
Tami Carpenter

12/10/99

An SAIC Company

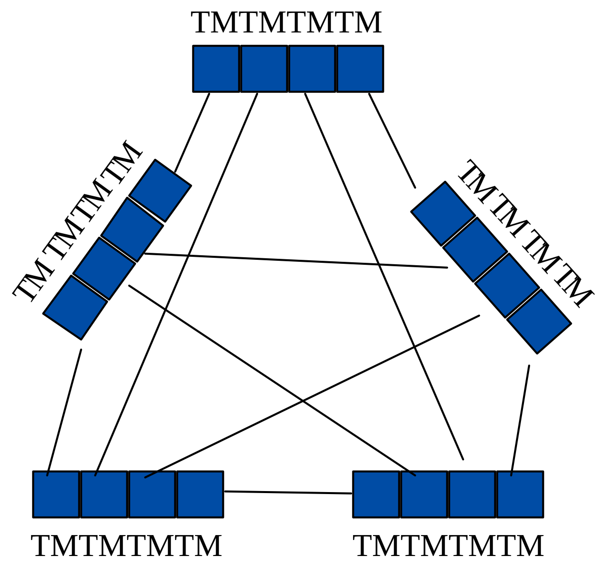
Quick Overview

- **The general setting**
- **The ring sizing problem**
- **Its relations and relaxations**
- **Simple ring sizing heuristics**
- **Performance guarantees**
- **Computational gizmos**
- **Future directions**

Reference: DIMACS tech report 97-02:
Demand Routing and Slotting on Ring Networks
(Site: www.dimacs.rutgers.edu)

BS: Before SONET

- **It was hard to multiplex traffic between different pairs of nodes.**
- **So...a network design might have dedicated connections between every pair of nodes between which there was demand.**
- **If demands needed protection, there would be two such connections. (Diversely routed.)**



Telcordia Technologies
Performance from Experience
Formerly Bellcore

And then there was SONET

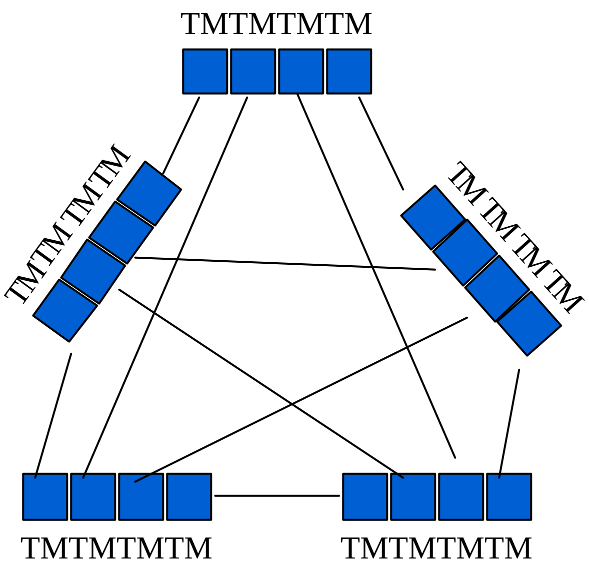
- **Standardized transmission protocols enable multiplexing traffic between different pairs of nodes.**

➡ Higher capacity links with more sharing

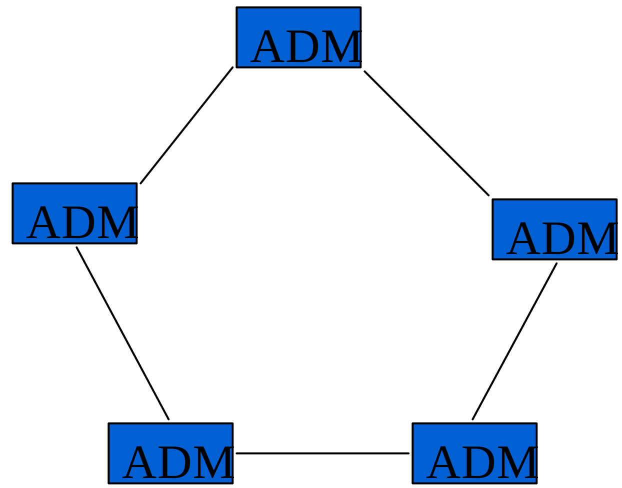
➡ Sparser networks

➡ Even more need to protect demands

Before and After...



Before

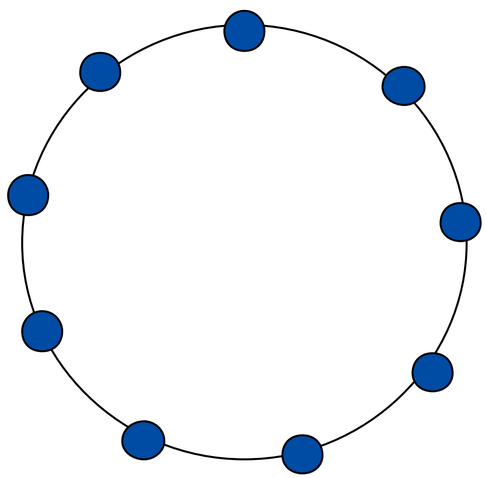


After

Telcordia Technologies
Performance from Experience
Formerly Bellcore

Networks based on Rings

- **Topologically: minimal two-connected subnetwork.**
  - **Efficient use of equipment**
  - **Survivable**
- **SONET technology allows you to:**
  - **Serve and protect demands on rings cost-effecively.**
  - **Assure fast recovery from single failures on the ring.**
- **Rings are a basic building block of SONET networks...**



9 node ring

Telcordia Technologies

Performance from Experience

Formerly Bellcore

Planning a SONET Network

- **Identify nodes in a "community of interest".**
  - Decide which nodes to group on the ring.
- **Find a minimum length cycle connecting these nodes in the network.**
  - This orders the nodes on the ring.
- **Determine the required ring size....**

➡ **Ring Sizing Problem (RSP)**

  - RSP may be solved 1000's of times in a session!

References:

Cosares, Deutsch, Saniee, & Wasem

Cook & Seymour

The Ring Sizing Problem

Given a ring and a set of demands to be routed around the ring….

- **For each demand:**
  - **Determine its routing on the ring.**
  - **All units must route the same way...**
- **For each demand unit:**
  - **Assign it a slot**
  - **Demand units sharing a slot cannot overlap.**
- **Minimize the number of slots used.**

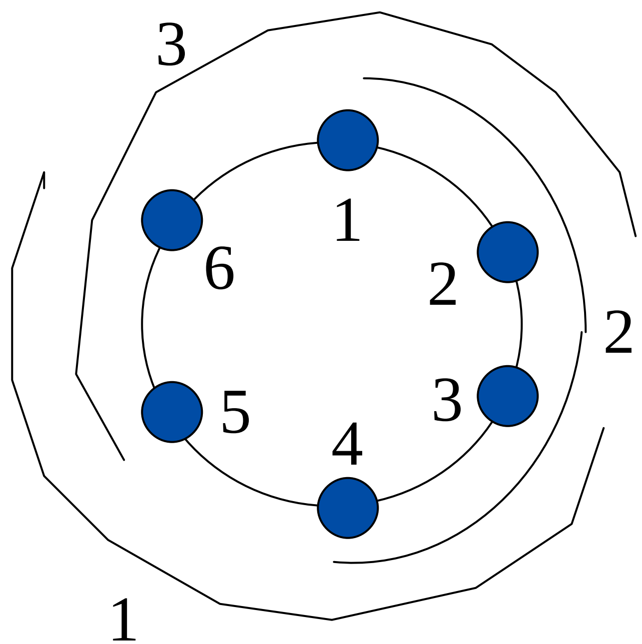
Telcordia™ Technologies
Performance from Experience
Formerly Bellcore

An Example

Demands: 1 ➡ 4 2 units

2 ➡ 5 3 units

3 ➡ 6 1 unit



6 slots required

RSP Formulation

$$\text{minimize: } C^* = \sum_c z^c$$

$$\text{subject to: } \quad x_k + \bar{x}_k = 1 \qquad \forall k$$

$$\sum_c z_k^c = d_k x_k \qquad \forall k$$

$$\sum_c \bar{z}_k^c = d_k \bar{x}_k \qquad \forall k$$

$$\sum_{k:l \in c(k)} z_k^c + \sum_{k:l \in cc(k)} \bar{z}_k^c \le z^c \qquad \forall l, c$$

All variables are binary

Two Relaxations (no slotting)

- Ring Loading Problem

$$\text{minimize } L^*$$
$$\text{s.t.: } x_k + \bar{x}_k = 1 \quad \forall k$$
$$\sum_{k:l \in c(k)} d_k x_k + \sum_{k:l \in cc(k)} d_k \bar{x}_k \le L^* \quad \forall l$$
$$x_k, \bar{x}_k \in \{0,1\}$$

Route to minimize traffic on a link

- LP Relaxation

$$\text{minimize } z^*$$
$$\text{s.t.: } x_k + \bar{x}_k = 1 \quad \forall k$$
$$\sum_{k:l \in c(k)} d_k x_k + \sum_{k:l \in cc(k)} d_k \bar{x}_k \le z^* \quad \forall l$$
$$0 \le x_k, \bar{x}_k \le 1$$

Route to minimize traffic on a link

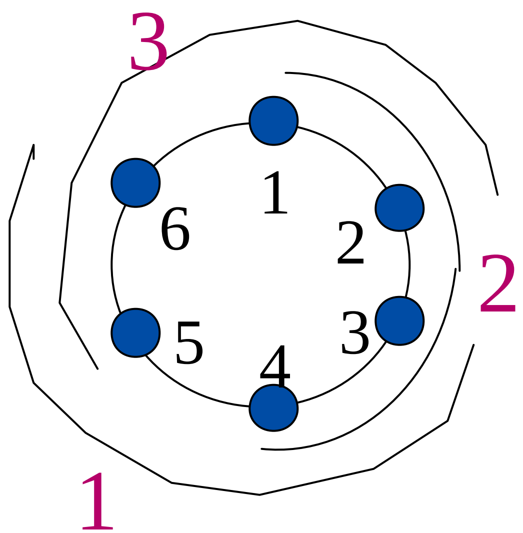
routing may be SPLIT

Telcordia™ Technologies

Performance from Experience

Formerly Bellcore

More Examples



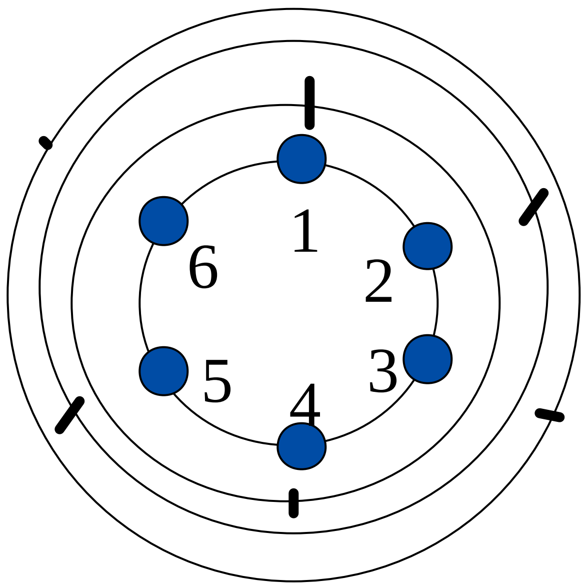
RSP vs RLP

$C^* = 6$

$L^* = 5$

LP Relaxation
Demands split equally

→ $z^* = 3$



Telcordia Technologies
Performance from Experience
Formerly Bellcore

Relaxations without Slotting

- **Ring Loading Problem (NP-Hard)**
  - Cosares and Saniee
  - Schrijver, Seymour & Winkler (SSW)
- **LP Relaxation**
  - Okamura & Seymour
  - Vachani, Shulman, Kubat & Ward (VKSW)
- **"Integer Relaxation" of RLP (Polynomial)**
  - Integer Multi-commodity Flow
  - Frank; Frank, Nishizeki, Saito, Suzuki & Tardos (FNSST)
  - SSW
  - VKSW

Telcordia Technologies — Performance from Experience — Formerly Bellcore

(Not a complete list!!)

Relaxation without one-way-routing

- **"Integer splitting" allowed.**
  - All demands are one unit
- **Demand units must be slotted**.
- The "WDM version" of the problem.
  - Slots = wavelengths.
- NP-Hard: Erlebach & Jansen.
- Better approximation bounds.
  - Kumar
  - Cheng

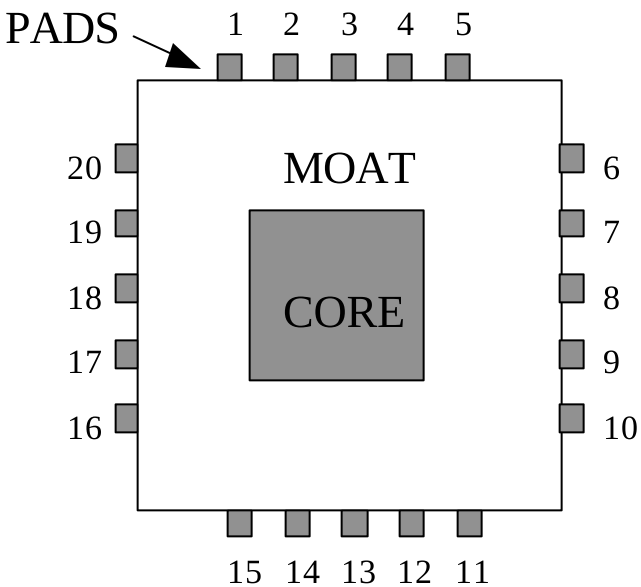
(Not a complete list!!)

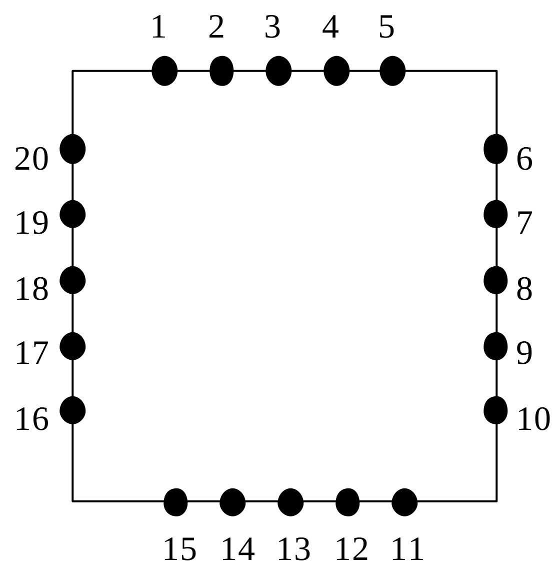
Telcordia™ Technologies
Performance from Experience
Formerly Bellcore

A Generalization from Circuit Design

- Demands may include more than two points on the ring that must be connected
- Demands are UNIT-sized.
- A "demand" is essentially a "wire" that will connect the constituent nodes.
- There are *n* possible routings of a demand with *n* nodes.
- Slotting and no slotting variants:
  - Objective 1: Minimize congestion (overlap) on a link.
    - Minimum Congestion Hypergraph Embedding in a Cycle.
  - Objective 2: Wires must stay within a channel, minimize the number of channels. (Moat Routing)
  - References: Ganley & Cohoon.
  - Both variants are NP-Hard.

A Circuit and its Model as a Cycle





Relaxations of RSP: Lower Bounds

- **We can get lower bounds from relaxations.**

$$z^* \leq L^* \leq C^*$$

- **Okamura & Seymour characterize value of LP:**
  - **Let D(i,j) be the amount of demand separated by removing links i and j.**

$$z^* = \frac{\max_{i,j}\left(D(i,j)\right)}{2}$$

- **With "integer splitting" maximum load on a link is no more than z*+1 (FNSST).**

Heuristics for RSP: Upper Bounds

- **Why heuristics?**
  - Difficult problem that is solved repeatedly.
- **We consider "two-phased" heuristics:**
  - Route THEN slot.
  - Easier to think about.
  - Components are known entities.
  - More flexible. (Everyone has a routing method…)
- **For these heuristics:**

$$z^* \le C^* \le C \le 2z^*$$

- **Methods are very simple!**

“Favorite” Routing Methods

- **“Cost-based” routing:**
  - Each link has a non-negative cost and we route to minimize the cost.
  - “Min hop” is a popular special case.
  - “One way” routing is a simple special case.
- **“Unsplitting” the LP solution.**
- **Routing from the Ring Loading Problem,**
  - Either optimal or heuristic.

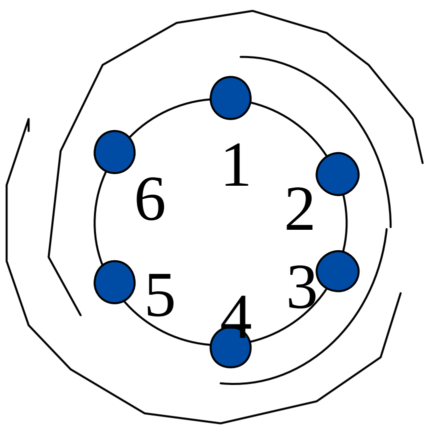
Telcordia™ Technologies

Performance from Experience

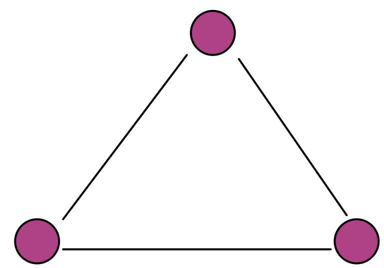
Formerly Bellcore

Slotting a Given Routing

Slotting ⇔ Coloring the vertices of a Circular Arc Graph (which is NP-Hard)



Circular arcs

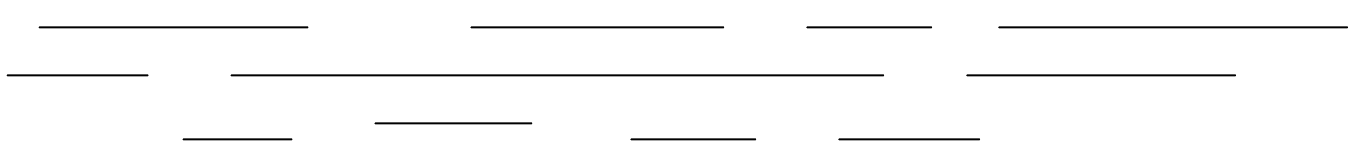


Circular arc graph

To demonstrate bounds for RSP heuristics we need only a very simple slotting heuristic

Interval Graphs

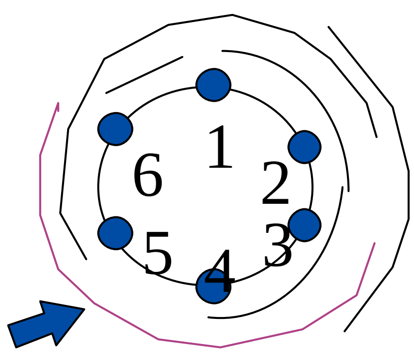
- A simple special case....
- Overlap graph associated with a collection of interval on a line



- Nice properties:
  - Chromatic number = max overlap (slots = load)
  - Linear time coloring.

A Slotting Heuristic

- **Tucker's Algorithm**:
  - **Select the point on the ring overlapped by the least demand. Assign each of these demand units to their own slot.**
  - **The remaining (unslotted) demands form an interval graph, so slot them optimally.**



TA uses 4 slots

Telcordia Technologies
Performance from Experience
Formerly Bellcore

Bounds for Two-phase Heuristics

- **Cost*-based routing + Tucker's Algorithm (TA)**
  - uses $\leq$ 2z* slots
- **Unsplitting the LP solution + TA**
  - uses $\leq$ 2z* slots
- **Optimal load routing + TA**
  - uses $\leq$ 2L* slots
  - optimal slotting is no better!
- All bounds are tight.

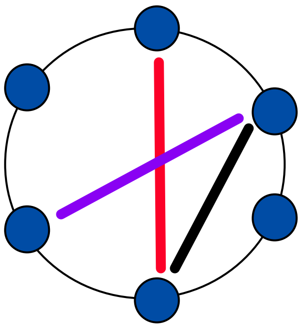
Telcordia Technologies
Performance from Experience
Formerly Bellcore

Simplest Case: One-way Routing (Or Edge-Avoidance Routing...)

- **Route all demand clockwise.**
- **This requires $\leq$ 2z* slots.**
  - Observe: This routing does not use link (n,1).
  - Consider the optimal LP routing. It routes at most z* units on (n,1). Reverse the routing of these demands, and we get the clockwise routing and have increased the load on no link by more than z*.
  - Now we have an interval graph with a maximum overlap of no more than 2z*.
  - Interval graph coloring gives the slotting.
- **Clockwise routing trick gives 2-approximation for MCHEC and Moat Routing.** (Carpenter, Cosares, Ganley & Saniee)

Parallel Routings

- **Represent each demand as a chord:**



  - »**Two demands are crossing if their chords intersect.**
  - »**Two demands are parallel if they do not.**

- **A routing is parallel if no edge between two parallel demands has load from each of them.**

Telcordia Technologies
Performance from Experience
Formerly Bellcore

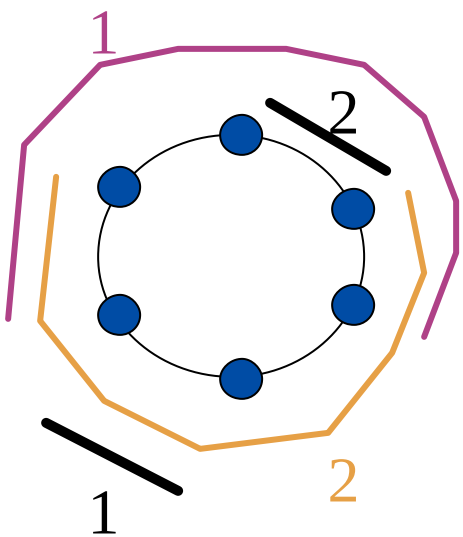
Examples of Parallel Routings

- **Edge-avoidance routing**
- **Cost-based routing**
  - With positive cost or appropriate tie-breaking rule
- **A parallel optimal LP solution**
  - Can be obtained from any LP solution. (SSW)
- **A parallel optimal solution to the "integer relaxation"**
- **Any unsplitting of a parallel routing**

- **Parallel Routings require no more than 2z* slots**

Optimal: Perhaps NOT Parallel

Optimal solutions for RSP or RLP may not be parallel



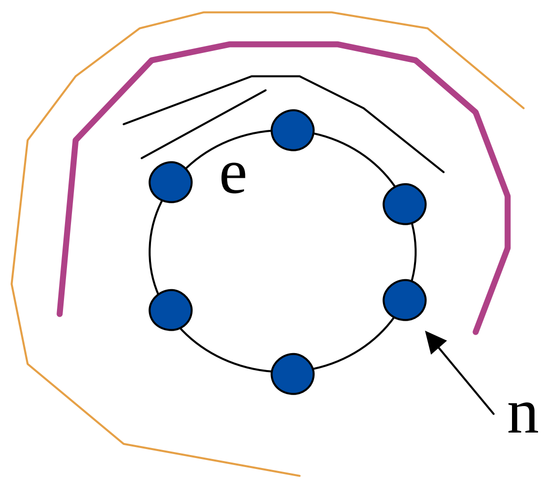
C* = L* = 3

Parallel routing can be no better than 4.

Telcordia Technologies
Performance from Experience
Formerly Bellcore

Slotting Parallel Routings

- **Helpful Lemma: In a parallel routing, for any edge e there is a node n such that no demand routes over both n and e.**



Telcordia Technologies
Performance from Experience
Formerly Bellcore

Slotting Parallel Routings...

- **Given a parallel routing R, let e be the edge with maximum load, L.**
- **Let *l* load on the least loaded point.**
- **TA slots using at most L+*l* slots.**

We can show that L+*l* ≤ 2z*:

- **Reverse the direction of all demands crossing e.**
- **The load at n increases by L and is at least L+*l*.**
- **This routing avoids e, so no load exceeds 2z***
- **so..... 2z* ≥ load at n ≥ L + *l***

Telcordia Technologies

Performance from Experience

Formerly Bellcore

How Good is the Optimal RLP Routing?

- **An “optimal” two-phased approach is theoretically no better than a naïve one:**
  - **Find the optimal solution to RLP**
  - **Find the optimal slotting of this routing**
  - **The result guarantees no better than twice optimal.**
- **We can construct an example for which the optimal load routing requires asymptotically twice as many slots as the optimal slot routing.**

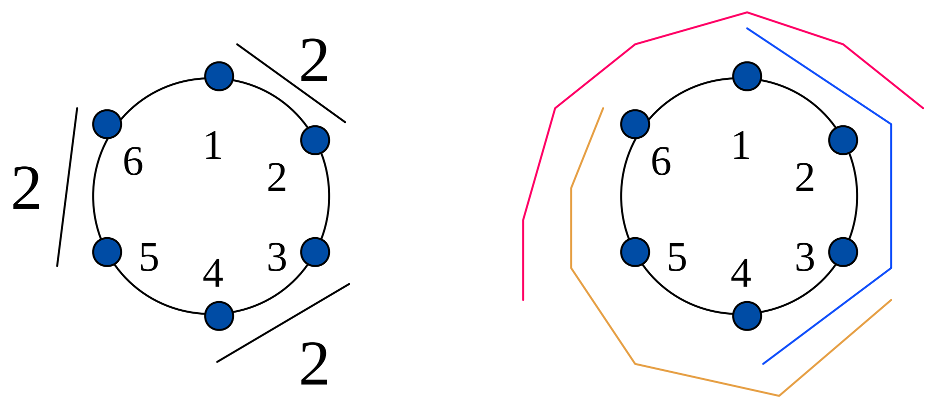
Telcordia™ Technologies
Performance from Experience
Formerly Bellcore

“Bad Cases” for “Optimal” Two-Phase

- **Ring of size 2n, where n is odd.**
- **Demands:**
  - **“short demands”:**

    2 units between nodes i and i+1 for all odd i.
  - **“long demands”:**

    2 unit-sized demands between nodes i and i+n, i<n



Short demands   Long demands (2 each)

Some Observations

- **z^* = n+1 because cutting the ring in half separates all of the long demands and one short demand. (2n+2 units.)**
- **This is a lower bound on both link load and the number of slots.**
- **If both units of the long demands route in the same direction, then we need at least 2n slots.**

Telcordia™ Technologies
Performance from Experience
Formerly Bellcore

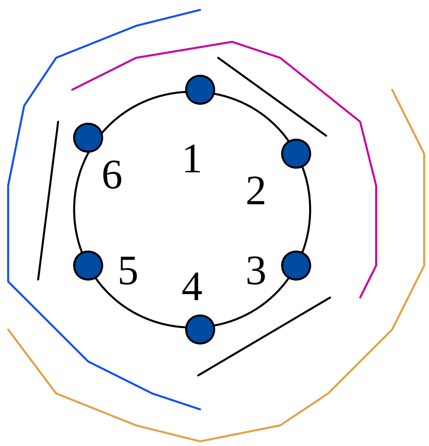
The Optimal Load (RLP) Routing

- **The short demands route along (i,i+1)**
- **Both units of each long demand route to overlap as few short demands as possible.**
- **The maximum link load is n+1, so this is optimal for RLP.**
- **This is a unique optimal**
- **It requires 2n slots.**

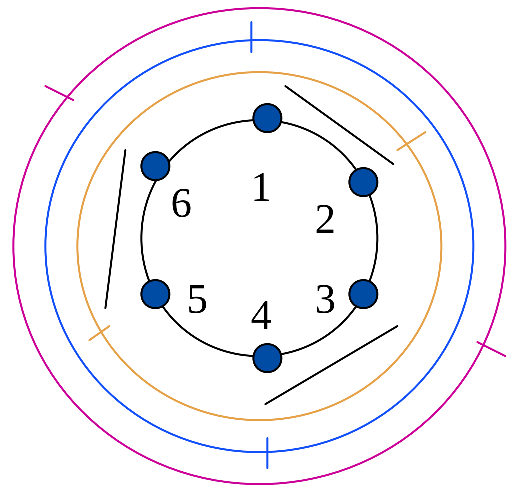
Telcordia Technologies
Performance from Experience
Formerly Bellcore

A Better Routing

- **Route the short demands (i,i+1)**
- **Route one unit of each long demand in each direction**
- **This requires n+2 slots.**



Optimal load routing -- all arcs are 2 demand units



Optimal slot routing-- short arcs are 2 units

Telcordia™ Technologies
Performance from Experience
Formerly Bellcore

To Recap....

- **Edge-avoidance routing $\leq$ 2z* slots**
- **Cost-based routing $\leq$ 2z* slots**
- **LP unsplitting $\leq$ 2z* slots**
- **Optimal Load routing $\leq$ 2L* slots**
- **All $\leq$ 2C* slots**

Telcordia Technologies
Performance from Experience
Formerly Bellcore

A “Practical” Version of Tucker’s Algorithm

- **We must use enough slots to slots demands that overlap a point.**
- **Don’t place demands that “straddle” a point alone in a slot:**
  - **Find an independent set that includes a straddling demand plus demands from the interval graph.**
- **Use the fact that demands terminate at a small number of nodes.**
  - **Iterate: try “cutting” the ring at each node and use the best solution.**

Telcordia Technologies

Performance from Experience

Formerly Bellcore

A Two-phased Heuristic

- **Find a routing** using Cosares & Saniee dual ascent procedure.
  - (An iterative cost-based approach.)

- **Find a slotting**:
- At each node:
  - For each “straddler” d:
    - Find a maximum independent set that includes d.
    - Slot these demands together.
  - Slot the remaining interval graph.
- Use the best solution...

Telcordia Technologies
Performance from Experience
Formerly Bellcore

More Integrated Heuristics

- Consider routing and slotting together.
- Begin with the CAG that contains arcs for both possible routings for each demand.
  - Iteratively find maximum independent sets.
  - Route & slot the demands in the set.
  - Remove arc corresponding to alternate route & continue.
  - MIS variations: max weight independent set, MIS containing a particular demand.
- Observation: the maximum weight set of crossing demands gives a lower bound on the number of slots required.
  - This is the maximum weight clique in the circle graph.
  - Begin by finding independent sets that include these….

Sample Results

Z*	L*	C*	optslot(L*)	Two-phase	MIS	2z*
34	38	38	45	38	40	68
44.5	48	48	48	48	53	89
28.5	29	29	29	31	32	57
29	29	29	29	30	32	58
44.5	50		62	56	62	89
30	30	30	32	30	32	60
32	32	32	32	32	34	64
38	40		45	44	49	76
44.5	49		52	50	58	89
29	29	29	33	30	29	58

10 node ring
20 demands
Weibull(3,20)

Continuing Work

- More thorough examination of heuristics.
- Complete development of “integrated” approaches.
- Solving the integer program directly.
  - Might be able to exploit the modularity of real problems.
  - “Integer-split” version (with slotting) might have more structure to exploit.
- Better understanding of the LP relaxation for the “circuit design” problems.
  - Can we characterize the LP solution as Okamura and Seymour did for the case when demands have two endpoints?