Node-Capacitated Multicommodity Routing for a PON Ring

A. Frank
Eötvös University and Ericsson Traffic Laboratory
Budapest, Hungary

F.B. Shepherd
Bell Labs, Lucent Technologies
Murray Hill, NJ

V. Tandon
Ciena Europe Ltd.
High Holborn, London, UK

Z. Vegh
Eötvös University
Budapest, Hungary

November 12, 1998

Abstract
We consider a routing problem which arises in the development of an algorithm which assigns routes in real-time to incoming path requests (calls) between pairs of nodes in a passive optical local network. For the application addressed, this led to an integral multicommodity flow problem on a ring. In the version of this problem where capacities are on the edges, a characterization for the existence of a solution followed from work of Okamura-Seymour and Frank. The application discussed here, however, leads to capacities being on the nodes of the ring. We give necessary and sufficient cut conditions for this fractional flow problem in the case where there are capacities on the nodes (and/or edges). This yields a polynomial time algorithm to determine whether such a flow exists. We also see that if we increase the capacity of each link by one, then we may always find an integral routing. Finally, we give simulation results for an on-line ring routing algorithm which was suggested by the preceding results. These showed that blocking in a network can be delayed by departing from a strict shortest path routing regime, even in the case of wavelength routing. This suggests that linear programming cut conditions could be used as measures for how not to route calls in real-time.

Keywords: *Multicommodity flow, ring networks, cut conditions, on-line routing.*

1 Description of a Routing Problem in Passive Optical Networks

One method of accessing the vast bandwidth of optical fibre is through the use of Wavelength Division Multiplexing (WDM) techniques [6]. Within future telecommunications access networks, Passive Optical Networks (PON) may prove a cost-effective and efficient method of delivering a wide variety of services to customers over fibre [1]. In such a network, many users may be connected to each PON and pairs of users are dynamically allocated a pair of wavelengths for the duration of a call or connection between them. Wavelengths assigned to a call on a chosen route will conflict with other wavelengths only if they are at the same frequency; hence paths and wavelengths must be chosen to avoid this colour clashing. This is because each node of the network is equipped with a single laser for each of the transmitted frequencies and data transmitted by a single laser can be 'received' throughout the local network by any laser tuned to that frequency. It is this passive nature of PONs which makes the equipment inexpensive relative to other optical technologies.

Bidirectional ring topologies have been shown to provide reliability and flexibility - see [2], [3]. The number of wavelengths available on a ring of PONs, however, is still far less than the number of users. [1] A future optical access network based on bidirectional rings of PONs and utilising WDM has been proposed in [14] where wavelength reuse has been used as a means of overcoming this limitation. Wavelength reuse is achieved by the inclusion of simple blocking filters, *absorbers*, in the network which can be used to localize a wavelength to a desired region thereby allowing its reuse elsewhere in the network - see Figure 1. The resulting system architecture is called a *λ-PON* ([15]).

We consider the problem of satisfying on-line path requests in a λ-PON (capacitated ring network) where each node of the ring thus corresponds to a PON which can transmit several wavelengths passively through the ring. The passive property and the fact that each wavelength can be used at a PON for a single call, means that node capacities, as opposed to traditional edge capacities, are inherent to the development of an on-line algorithm.

The complication of node capacities led us to first study a static problem which is paired more directly to the situation where wavelength conversion is available. Namely, we characterize those static path requests which can be satisfied (routed) without violating the inherent node capacities. This would be precisely the problem of interest if the network manager could "start from scratch" each time a new path request was received (i.e., if it were allowed to re-route calls in progress).

More precisely, we consider a (ring) graph whose nodes $(0, 1, 2, \ldots, n-1)$ correspond to the PONS and a traffic matrix whose entries $\hat{d}_{ij}$ represent the number of path requests between users attached to PON i and users attached to PON j. Note that a request from PON i to PON j could either be routed along $i, i+1, \ldots, j$ or $j, j+1, \ldots i$. If, say, the former was chosen, then some wavelength L would be used by each of the PONS in $[i, j]$. Here, we use $[i, j]$ to denote the set $\{i, i+1, \ldots, j-1, j\}$. Thus each of these PONS will have one less available wavelength. We consider the problem of determining whether a traffic matrix can be satisfied subject to not violating integer node capacities $c(i)$ on the nodes of the ring. The reason for allowing varying node capacities $c(i)$ is in aid of our goal to

[1] Although potentially much larger than the number of available wavelengths in larger optical networks.

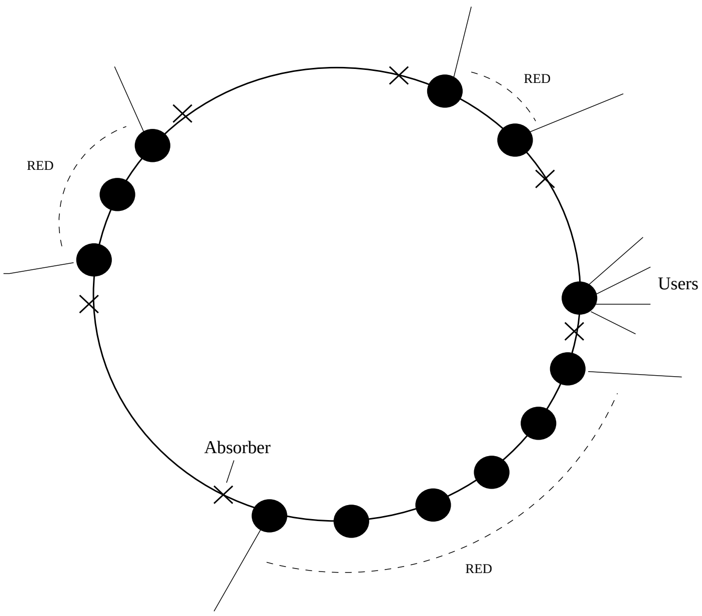


Figure 1:

find an on-line routing algorithm; as a network becomes more congested, the net effect is that the PONS will have differing capacities available. We discuss this further in Section 3 where an approach is proposed for on-line algorithms which uses *cut conditions* for such static problems. Several algorithms are proposed and simulation results, for a crude version of one of them, are presented.

1.1 Overview of Static Routing Problems

A *static routing problem* is determined by a network which is capacitated in some fashion and a list of *demands* which need to be satisfied between pairs of nodes in the network. We refer to the capacitated network as the *supply* graph $G = (V, E)$ where $V = \{0, 1, 2, \ldots, n-1\}$. The list of demands is usually specified as a traffic matrix. Generally each demand represents a request for some type of "connection" in the supply graph; this may be a path, a cycle, a tree and so on. In the application discussed, each demand between a pair of nodes i and j actually represents a request for two paths, the idea being a path from i to j as well as a return path for j to i. Moreover, the limited area of the local network examined, enables us to route the two paths for a particular demand along distinct paths in the ring. This together with the fact that the supply graph is undirected allows us to represent each such demand as a request for two paths. We model this by a simple undirected *demand graph* $H = (V, F)$ together with a non-negative integer demand function $h : F \to \mathbb{Z}_+$. Thus $h(ij)$ represents the number of paths requested between i and j; for the PON problem discussed, this is necessarily even but for most of the remainder we do not assume this.

We denote by $\mathcal{P}$ the collection of all paths in G and by $\mathcal{P}_{ij}$ the set of all paths which join the nodes i, j. A *routing* in G is an assignment $x : \mathcal{P} \rightarrow \mathbb{Z}_+$ of integers to the paths of G. A routing is said to *fulfil* h if

$$\sum(x(P) : P \in \mathcal{P}_{ij}) = h(ij). \tag{1}$$

At this point we have said nothing about the capacity of the network and so a routing may or may not be *feasible*. The network G may be supplied with edge capacities (usually integer) $u : E \rightarrow \mathbb{R}_+$ or node capacities $\hat{c} : V \rightarrow \mathbb{R}_+$.

A routing x *satisfies the edge capacities* if

$$\sum(x(P) : P \in \mathcal{P}, e \text{ is an edge of } P) \leq u(e) \tag{2}$$

holds for every edge $e \in E$. Similarly x *satisfies the node capacities* if

$$\sum(x(P) : P \in \mathcal{P}, v \text{ is a node of } P) \leq \hat{c}(v) \tag{3}$$

holds for every node $v \in V$.

There has been considerable work on edge-capacitated routing problems, e.g., [10], [4], [12], [16], [7], - see also [8]. Note that by subdividing edges, one may view the edge-capacitated routing problem as a special case of node-capacitated routing. Indeed in the latter, we may include both edge and node capacities.

Note also that by using multiple edges, one can view edge-capacitated routing as an edge-disjoint paths problem (i.e., with $0, 1$ capacities). This standard trick for reducing to $0, 1$ capacities does not work in the case of node capacities. In fact, in contrast with the body of work for edge-capacitated network routing, there are fewer positive results addressing node-capacitated routing problems, prime exceptions include results of Menger and Ford, Fulkerson (i.e., single source-destination pair) and the work of Robertson and Seymour [11] on the disjoint paths problem (i.e., $0, 1$ node capacities) - see also [13]. Negative results for the disjoint paths problem include NP-completeness for undirected grid graphs [5] and when the union of the supply and demand graphs is planar [9] (the latter holds also for edge-disjoint paths).

2 Node-Capacitated Routing in a Ring Network

2.1 Feasibility

We consider the case where our supply $G = (V, E)$ is a cycle (or circuit or ring), $V = \{0, 1, \ldots, n-1, n = 0\}$ and $E = \{e_0, \ldots, e_{n-1}\}$ where $e_i = i(i+1)$, $i = 0, \ldots, n$ (arithmetic is modulo n). We sometimes identify i with v_i to make explicit that we are referring to a node. We use (i, j) to mean a directed arc from i to j and $[i, j]$ to denote the set of nodes $\{i, i+1, \ldots, j\}$. We use $P(i, j)$ to denote the path in G: $i, i+1, \ldots, j$. For a vector $x : \mathcal{P} \rightarrow \mathbb{R}$, we use the shorthand $x(ij)$ for the value $x(P(i, j))$.

Decision Problem: FEASIBLE TRAFFIC DEMAND (F-TRAFFIC)

Input: A cycle supply graph $G = (V, E)$; demand $H = (V, F)$ with $h : F \rightarrow \mathbb{Z}_+$; an integer capacity $\hat{c}(i)$ for each node $i \in V$

Question: Is there a routing which fulfils h and satisfies the node capacities?

Our approach is to relax the integrality condition on our routings. Namely, a *fractional routing* is an assignment $x : \mathcal{P} \rightarrow \mathbb{R}_+$ to the paths of G. Using this notation, the remainder of the definitions from the preceding section immediately translate to conditions on such fractional routings. In particular, we will search for fractional routings which fulfil h and which satisfy the node capacities. We will also see that by increasing each capacity by at most one, we may find an integral routing.

We start with a simplifying observation for the node-capacitated model. For any node k let $d_k = \sum_{f=kj \in F} h(f)$ be the total number of path requests which involve node k. Note that the only freedom we have in choosing paths P to fulfil a demand edge ij is in the orientation around the cycle G. Any such path will use exactly one unit of capacity at each of the nodes i and j. In particular, we must have $\hat{c}(k) \geq d_k$ for each node k and $c(k) = \hat{c}(k) - d_k$, called the *through capacity* of k, is the amount of capacity remaining at node k which can be used for paths satisfying demands which do not involve k. Let $I(P)$ denote the set of *internal nodes* of a path P. Then our original problem is equivalent to finding a routing x which fulfils h and for which

$$\sum(x(P) : v \in I(P)) \leq c(v) \tag{4}$$

for each node v. For the remainder of the paper, we assume that we are given nonnegative through-capacities and are asked to find a path assignment which satisfies the (through) node capacity constraints (4).

For any function $g : S \rightarrow \mathbb{R}$, we use the notation $g(X) := \sum(g(s) : s \in X), X \subseteq S$. For a number γ, let $\gamma^+ := \max(0, \gamma)$. For a function g, let g^+ be defined by $g^+(s) := (g(s)^+)$. For any subset X of the nodes of some graph, we let $\delta(X)$ denote the edges with one endpoint in X and the other in $V - X$. We sometimes shorten $g(\delta(X))$ to $d_g(X)$ and we denote $d_h([j+1, i])$ by $L(e_i, e_j)$ which is just the sum of the demands which *cross* the edge-cut $\{e_i, e_j\}$; this is also called the *load* of cut $\{e_i, e_j\}$.

The Okamura-Seymour Theorem (OS), when specialized to a ring G, asserts that there is a routing satisfying the edge-capacity constraints if and only if the edge-cut condition holds, that is,

$$d_u(X) \geq d_h(X) \tag{5}$$

for every subset $X \subseteq V$. Moreover, if u is integer-valued then there is a half-integral routing x. Note that it suffices to assume (5) for subsets X inducing a subpath of G. Hence the edge-cut condition is equivalent to requiring that

$$u(e_i) + u(e_j) \geq L_h(e_i, e_j) \text{ for every } 0 \leq i < j < n \tag{6}$$

(We also note that for rings the OS theorem is easy to prove.)

By a *double-cut* we mean a function $y : V \rightarrow \{0, 1, 2\}$.
For demand edge $f \in F$ and double-cut y, let $s(f, y) := \min[y((u, v)), y((v, u))]$.
Double-cut condition

$$\sum(h(f)s(f, y) : f \in F) \leq \sum(y(v)c(v) : v \in V) \tag{DC}$$

Theorem 1 *Given nonnegative through capacities c. There is a solution to the fractional node-capacitated routing problem if and only if (DC) holds for every double-cut y. Moreover, if (DC) holds, then there there exists a half-integral routing, and by increasing each capacity by one we may find an integral routing.*

The necessity is straightforward. The proof of sufficiency is based on the following approach. We will show first that (DC) implies the existence of a (half-integer) edge-capacity function $u : E \to \mathbb{Z}/2$ so that the edge-cut condition holds and so that any solution to the edge-capacitated routing problem will automatically be a solution to the node-capacitated problem. This in turn implies the existence of a half-integral routing by the Okamura-Seymour Theorem.

Before continuing with the proofs, we give an example which shows the simpler cut conditions of OS (two non-zero coefficients, both of which are 1) will not suffice for node-capacitated routing. Namely, consider a ring on the nodes $0, 1, 2, 3, 4, 5$ with all node through capacities equal to 1 but for node 3 which has capacity 0. Suppose now we are given two unit demands between nodes $0, 3$ and between nodes $2, 4$. One may verify that all "2-node-cut" conditions (akin to OS) are satisfied. Indeed, all double-cuts conditions are satisfied for any $0, 1$-valued y. On the other hand, let y be the vector with $y_1 = y_5 = 1$, $y_3 = 2$ and all other components 0. Then the right hand side for the double-cut constraint is 2, whereas the left hand side is $s(03, y) + s(24, y) = 1 + 2 = 3$. One easily finds examples in this way which give necessary double-cut constraints with an arbitrarily large number of positive components.

Theorem 2 *Let $u : E \to \mathbb{R}$ be a function satisfying edge-cut condition (6) and*

$$u(e_{i-1}) + u(e_i) \leq d_h(v_i) + 2c(v_i) \tag{7}$$

for every $v_i \in V$. Suppose that a routing $x : \mathcal{P} \to \mathbb{R}_+$ fulfils the demand h and satisfies the edge-capacity constraints with respect to the edge-capacity function u^+. Then x also satisfies the node-capacity constraint (3). Moreover, if there is an edge $e \in E$ with $u(e) \leq 0$, then x is integer-valued.

Proof: Let us consider an arbitrary node i. If $u(e_i) \leq 0$, then $u^+(e_i) = 0$ and hence $x(P) = 0$ for every path P using e_i. Therefore $x(P) = 0$ for every path P using i as an inner node, that is, in this case (3) is satisfied for i. The same argument works when $u(e_{i-1}) \leq 0$.

In the remaining case $u(e_i) > 0$ and $u(e_{i-1}) > 0$. Let $\alpha := \sum(x(P) : P$ is a path ending at i and using $e_i)$. Let $\beta := \sum(x(P) : P$ is a path ending at i and using $e_{i-1})$. Then $\alpha + \beta = d_h(i)$. Let $\gamma := \sum(x(P) : P \in \mathcal{P}, i$ is an inner node of $P)$. We have $\gamma + \alpha \leq u^+(e_i) = u(e_i)$ and $\gamma + \beta \leq u^+(e_{i-1}) = (e_{i-1})$. By combining these with (7) we obtain $\alpha + \beta + 2\gamma \leq u(e_{i-1}) + u(e_i) \leq d_h(i) + 2c(i) = \alpha + \beta + 2c(i)$, from which $\gamma \leq c(i)$ follows.

Let $u(e) \leq 0$ for some $e \in E$, that is, $u^+(e) = 0$, Let f be any demand edge and P_1, P_2 the paths connecting its end nodes. If P_1 contains e, then $x(P_1)$ must be zero and hence $x(P_2) = h(f)$, an integer. □

Theorem 3 *Let c be a (half-) integer vector of node through capacities. If (DC) holds for every double-cut y, then there exists a half-integral function $u : E \to \mathbf{Z/2}$ (i.e., $2u$ is integer-valued) satisfying (7) and (6). Moreover, if the range of h consists of even integers, then u may be chosen to be integer.*

Proof: We will define an appropriate directed graph $D = (U, A)$ with costs on its edges in which every element of V determines two edges and every edge-cut of G (consisting of a pair of edges) determines two edges. If there is a negative circuit in D, this will correspond to a double cut violating (DC). If there is no negative circuit in D, then there is a feasible potential and this will define a required u satisfying (7) and (6).

To be more specific, subdivide each edge e_i by two nodes s_i and t_i and let $U := \{s_i, t_i : i = 0, \ldots, n-1\}$. Imagine $\ldots t_{i-1}, i, s_i, t_i, i+1, s_{i+1} \ldots$ clockwise. Let A_1 consist of the following directed edges:

$$t_{i-1}s_i \text{ and } t_i s_{i-1}$$

for $i = 0, \ldots, n-1$. Let A_2 consist of the following directed edges:

$$s_i t_j \text{ and } s_j t_i$$

for $0 \leq i < j < n$. (In particular, $s_i t_{i-1}, s_{i-1} t_i \in A_2$ and $t_{i-1} s_i, t_i s_{i-1} \in A_1$). Let $A := A_1 \cup A_2$ and define a cost-function $w : A \to \mathbb{Z}$, as follows.

$$w(t_{i-1}s_i) := w(t_i s_{i-1}) := d_h(i) + 2c(i)$$

for $i = 0, \ldots, n-1$ and

$$w(s_i t_j) := w(s_j t_i) := -L(e_i, e_j)$$

for $0 \leq i < j < n$. We will use the notation $L(s_i t_j) := L(s_j t_i) := L(e_i, e_j)$.

It is well-known that either there is a directed circuit of negative total weight or else there is an integer-valued feasible potential, and with the help of a shortest path algorithm one can efficiently decide which alternative holds. Let us consider the two possibilities.

CASE 1. There is no negative circuit in $D := (U, A)$ with respect to w, that is, there is an integer-valued function $\pi : U \to \mathbb{Z}$ for which

$$\pi(v) - \pi(u) \leq w(uv) \text{ for every } uv \in A.$$

Such a function is called a *feasible potential.* For $i = 0, \ldots, n-1$ let us define

$$u(e_i) := (\pi(s_i) - \pi(t_i))/2.$$

.

Claim 4 *u satisfies (7).*

Proof. For $t_{i-1}s_i \in A_1$ and $t_i s_{i-1} \in A_1$ we have

$$\pi(s_i) - \pi(t_{i-1}) \le d_h(i) + 2c(i)$$

and

$$\pi(s_{i-1}) - \pi(t_i) \le d_h(i) + 2c(i).$$

By adding up and subdividing by 2, we obtain $u(e_i) + u(e_{i-1}) \le d_h(i) + 2c(i)$, as required for (7). $\square$

Claim 5 *u satisfies (6).*

Proof. For $s_i t_j \in A_2$ and $s_j t_i \in A_2$ we have

$$\pi(t_j) - \pi(s_i) \le -L(e_i, e_j)$$

and

$$\pi(t_i) - \pi(s_j) \le -L(e_i, e_j).$$

By adding up and dividing by two, we obtain $-u(e_i) - u(e_j) \le -L(e_i, e_j)$ which is just (6). $\square$

That is, if Case 1 occurs, then we have obtained a half-integer-valued u satisfying (3) and (7).

CASE 2. There is a negative circuit C in $D := (U, A)$ with respect to w. For each $k = 1, 2$, let C_k denote the set of edges of C belonging to A_k. Let $y(i)$ $(i = 0, \ldots, n-1)$ be the number of edges among $t_{i-1}s_i$ and $t_i s_{i-1}$ (C_1 edges) which belong to C. Hence $y(i)$ is $0, 1$, or 2, that is, y is a double-cut.

Lemma 6 *The double-cut condition (DC) for y is violated.*

Proof. For a demand edge $f = ij$, let $\alpha(f)$ denote the number of C_2-edges "crossing" f.

Claim 7

$$\alpha(f) \le 2s(f, y) + y(i) + y(j).$$

Proof. For convenience, think of f as being vertical and i above j. By symmetry, we may suppose that $y(P(i,j)) \le y(P(j,i))$, that is, $s(f,y) = y(P(i,j))$. Then $s(f,y)$ is the number of C_1-edges on the right hand side of f.

Let α_1 denote the number of C_2-edges crossing f whose head is t_i and hence crosses *to the right* or t_{i-1} (crosses *to the left*). Then clearly $\alpha_1 \le y(i)$ since each t_k has out-degree one.

Similarly, let α_2 denote the number of C_2-edges crossing f whose head is t_j or t_{j-1}. Then $\alpha_2 \leq y(j)$.

Let α_3 denote the number of C_2-edges crossing f to the right and whose head is different from t_i and t_{j-1}. Then, clearly, $\alpha_3 \leq y(P(i,j)) = s(f,y)$. Similarly, let α_4 denote the number of C_2-edges crossing f to the left and whose tail is different from s_i and s_{j-1}. Then, since each s_k has in-degree one, we also have $\alpha_4 \leq y(P(i,j)) = s(f,y)$.

By adding up we obtain: $\alpha(f) = \alpha_1 + \alpha_2 + \alpha_3 + a_4 \leq 2s(f,y) + y(i) + y(j)$, as required. □

For any arc $e \in C_2$ between s_i and t_j say, let $L(e)$ denote $L(e_i, e_j) = -w_e$.

Claim 8

$$\sum(L(e) : e \in C_2) \leq 2\sum(h(f)s(f,y) : f \in F) + \sum y(i)d_h(i) : i \in \{0,\dots,n-1\}). \qquad (8)$$

Proof. Recall that for a demand edge $f \in F$ $\alpha(f)$ denotes the number of C_2-edges crossing f. Then $\sum(L(e) : e \in C_2) = \sum(\alpha(f)h(f) : f \in F)$.

By Claim 7, we have $\sum(\alpha(f)h(f) : f \in F) \leq \sum(h(f)[2s(f,y) + y(i) + y(j)] : f = ij \in F) = 2\sum(h(f)s(f,y) : f \in F) + \sum(y(i)d_h(i) : i \in \{0,\dots,n-1\})$, as required. □

Since C is a negative circuit, that is, $w(C) < 0$, we have $\sum(-w_e : e \in C_2) > \sum(w(e) : e \in C_1)$. This can be re-written as:

$$\sum(L(e) : e \in C_2) > \sum_{i=0}^{n-1} y(i)[d_h(i) + 2c(i)].) \qquad (9)$$

By combining (8)) and (9), we obtain $2\sum(y(i)c(i) : i = 0,\dots,n-1) < 2\sum(h(f)s(f,y) : f \in F)$, that is, the double-cut condition is indeed violated for y. □□

Summing up, we have shown that if D has no negative circuit, then there is a half-integral u satisfying (7) and (6), and hence by (OS) and Theorem 2, there is a fractional routing for the node-capacitated problem. Otherwise, there is a negative circuit, and there is double-cut for which (DC) is violated. This completes the proof of Theorem 3 and the first half of the main theorem. □□□

Theorem 9 *If there is a half-integral solution y to (7) and (6) for which $y(e) \geq 0$ for every $e \in E$, then there is an integer-valued solution $z \geq 0$ to (7) and (6), as well, for which $|y(e) - z(e)| \leq 1/2$ for every $e \in E$.*

Proof: Suppose indirectly that we have a counter-example for which $2y(E)$, a positive integer, is as small as possible. By the OS Theorem, there is a routing $x : \mathcal{P} \to \mathbb{R}_+$ fulfilling demand h and satisfying the edge-capacity constraint with respect to y.

We claim some properties.

(PX1) *$y(e) > 0$ for every edge $e \in E$.* If $y(e) = 0$ for some edge $e \in E$, then x is integer-valued by Theorem 3 and hence $z(e_i) := \sum(x(P) : P$ uses $e_i)$ satisfies the requirements.

(PX2) *For every $e \in E$, $y(e)$ is non-integer.* For suppose without loss of generality that $y(e_{n-1})$ is integral but $y(e_0)$ is not. Contract e_{n-1} to obtain a new instance (h', y', c') in the obvious way except that we alter $c'(v_0) = c(v_0) + c(v_{n-1})$. One checks that y' satisfies (7) and (6). Since this instance is smaller than (h, y, c) we may apply Theorem 9. Hence there exists a desired z' for y'. One now checks that the vector z, with $z(e_i) = z'(e_i)$ for each $i \neq n-1$ and $z(e_{n-1}) = y(e_{n-1})$ satisfies the Theorem, a contradiction.

(PX3) *Every $e \in E$ belongs to a tight cut and to a tight constraint of the form (7).* If e does not belong to a tight cut, we can decrease $y(e)$ by a half. An appropriate z for the resulting vector, will also satisfy the theorem's conditions for y. If e does not participate in a tight constraint (7), then we can increase $y(e)$ by one half. If this results in a counterexample for Theorem 9, then we may contract e as in (PX2) to obtain a counterexample smaller than y. Otherwise, we obtain the desired vector z as in the first case.

(PX4) $d_h(v_i) > 0$ *for every* $1 \leq i \leq n$. If $d_h(v_i) = 0$ for some i, then $y(e_i) = y(e_{i-1})$ since each edge belongs to a tight cut. "Eliminate" v_i by contracting e_i. If there is a required vector z' for the contracted problem, then z will be good for the original where $z(e_i) := z'(e_{i-1})$ and $z(e_j) := z'(e_j)$ $(j \neq i)$. To see this, one has to check only that $(*)\ \ z(e_{i-1}) + z(e_i) \leq d_h(v_i) + 2c(v_i) = 2c(v_i)$. Since $y(e_i)$ is not an integer, then $y(e_i) + y(e_{i-1}) = 2y(e_i)$ is an odd number, and hence $y(e_i) + y(e_{i-1}) \leq 2c(v_i) - 1$. We have $z(e_{i-1}) + z(e_i) = 2z(e_i) \leq 2(y(e_i) + 1/2) \leq 2c(v_i)$.

(PX5) $y(P) < 1$ *for every* $P \in \mathcal{P}$. Suppose that $y(P) \geq 1$ for a path P connecting the end-nodes of a demand edge f. Consider demand function h' and edge-capacity y' where h' arises from h by reducing $h(f)$ by one and y' arises from y by reducing $y(e)$ by 1 for each edge e of P. Also c' arises from c by reducing each internal node of P by 1 (one can verify that this results in a nonnegative vector). Again we can apply Theorem 9 and obtain an integer-valued solution z' to (7) and (6) with respect to h', c'. Define z by increasing $z'(e)$ by 1 for each edge e of P. This z satisfies (7) and (6), a contradiction.

(PX6) *If $h(f) \geq 1$ for some demand edge f and $C := \{e_i, e_j\}$ is a tight cut, then f crosses C.* For otherwise, one of the two paths P_1 and P_2 connecting the end-nodes of f, say P_1, uses both e_i and e_j. Since C is tight, we must have $y(P_1) = 0$ and hence $y(P_2) = h(f) \geq 1$, contradicting (PX5).

(PX7) *Every two (distinct) tight cuts are crossing, in particular, every edge $e \in E$ is contained in at most one tight cut.* Indeed, for two non-crossing cuts C_1 and C_2 there is a node v which is "between" the two cuts. By (PX4) there is a demand edge f ending at v and f can cross at most one of the cuts C_1 and C_2, contradicting (PX6).

We can conclude that every edge of G belongs to exactly one tight cut, and the tight cuts are pairwise crossing. It follows that $|V|$ is even, each opposite pair of nodes form a demand edge whose demand is one and there are no other demand edges. Furthermore each opposite pair of edges of G form a tight cut and these are all the tight cuts.

Let $k := |V|/2$.

We claim that each $y(e_i)$ has the same value. If this were not true, then there would be two consecutive edges with different y-values and we may assume that these edges are e_0 and e_1. We also may assume that $y(e_0) < y(e_1)$. Since these are half-integers but not integers, we have $y(e_0) \leq y(e_1) - 1$. But this, along with the tightness of cut $\{e_1, e_{k+1}\}$ would imply that the cut $\{e_0, e_{k+1}\}$ is also tight, contradicting the property that only opposite pairs of edges form tight cuts.

We have obtained that there is a positive integer t so that $y(e_i) = (2t-1)/2$ for every index i, $1 \leq i \leq n$. Since the load of each cut of opposite pair of edges is k and these cuts are tight, one has $k = 2t - 1$. Therefore $|V|$ is even but not divisible by 4. We now define z by $z(e_i) := y(e_i) + \frac{(-1)^i}{2}$ for each i. Since only the cuts formed by opposite edges are tight, z satisfies the edge-cut condition (6) and it also satisfies (7).

We can conclude that there is an integer-valued solution z to (6) and (7) contradicting the indirect assumption that no such solution exists. This contradiction completes the proof of Theorem 9. $\square$

We remark that the condition $|y(e) - z(e)| \leq \frac{1}{2}$ on our vector z implies that we may find (in linear time) the integral vector z by "fixing" any non-integral $y(e)$ by increasing or decreasing it by $\frac{1}{2}$. This is followed by a sequence of alterations implied by any violated tight cuts.

Let us return to the proof of (the non-trivial direction of) the Main Theorem. By Theorem 3 there is a half-integral capacity function y satisfying (6) and (7). If y has a negative component, then by Theorem 2.1 we have an integer-valued routing satisfying the node-capacity constraints. If y is non-negative, then by Theorem 9 there is an integer-valued function z satisfying (6) and (7).

By applying Okamura-Seymour with respect to h as demand and z as an edge-capacity function we obtain that there is a half-integer routing x satisfying the edge-capacity constraints. By Theorem 2.1 this x satisfies the node-capacity constraints.

Finally, suppose that x is a feasible routing for our problem and consider those demands ij which are *split* under x, i.e., $x(ij), x(ji)$ are both positive. Suppose first that there exists a pair of noncrossing split demands ij, kl where these nodes appear on the ring in clockwise order i, j, k, l. Then for sufficiently small $\epsilon > 0$ we may reduce $x(ji), x(lk)$ by ϵ and increase $x(ij), x(kl)$ by ϵ to obtain a new feasible routing. In fact, choosing ϵ as large as possible, the new routing will have one less split demand. We may repeat this process until we have a (still half-integral) routing for which any pair of split demands is crossing. Assume that these demands are $u_i u_{k+i}$ (subscripts mod $2k$) where the nodes $u_0, u_1, \ldots, u_{2k-1}$ occur in this order around the ring. Finally let x' be a solution obtained from x as follows. For each $i \in \{0, 1, \ldots, 2k-1\}$ increase $x(u_i u_{i+k})$ by $(-1/2)^i$ and $x(u_{i+k} u_i)$ by $(-1/2)^{i+1}$. The load of any edge under this new solution is changed by at most one and the proof of the theorem is complete. $\square$

3 Dynamic Routing Algorithms and Cut Conditions

At this point we consider the application of a static theorem such as Theorem 1 to the design of an ORA. Its main use is a criterion for measuring the vulnerability of a network. For a given set of requests which are currently in progress and have already been assigned paths, the *reduced capacity vector* $\tilde{c}$ of the network, consists of the available capacities at each node.

Note that for a given load $\hat{d}$ and reduced capacities $\tilde{c}$, we can solve the node-capacitated routing problem if and only if there is no violated doubled-cut inequality for some μ. Thus the robustness of a network relative to a current traffic load and set of reduced (through) capacities should measure how close we are to having one of these inequalities being violated. In other words, the vulnerable capacities should correspond to nodes in the support of some $0,1,2$ vector μ for which the excess capacity $\epsilon(\mu)$ is small. Moreover, a vector with $1^T\mu$ large, is relatively more vulnerable since it has proportionally more nodes which could bring down its excess capacity. Thus a possible measure of the robustness of a network, relative to target demand $\hat{d}$ and reduced through capacities $\tilde{c}$, is:

$$\rho(d,\tilde{c}) = \min\{\frac{\epsilon(\mu)}{1^T\mu} : \mu \in \{0,1,2\}^{V_G}\} \tag{10}$$

This suggests that we chose our route for an incoming request to be in the direction which leaves the network with a higher robustness for demand loads which are likely in the future.

One approach is to then choose a *load prototype* $\hat{d}$ which can be thought of as a typical (or worst-case) load which we are likely to encounter. As a new request is received, the algorithm will try first the path in the direction which leaves the robustness larger, (and then if it cannot satisfy the call, it tries the other direction).

The aim of this method is to delay blocking as long as possible. Our simulations indicate that such algorithms are not as efficient if a significant level of blocking has already occurred. In this case a shortest path method is preferable. Thus in the algorithm below, we use a variable `bp-cutoff` which is the point beyond which the algorithm should revert to simply choosing a shortest path in its initial attempt to satisfy the request.

DOUBLED-CUT ALGORITHM

[1] WAIT for event: If a request is terminated go to **[2]**;
If request between i,j is initiated go to **[3]**.

[2] If the terminated request is along the path $P(i,j)$ update the reduced capacities along this path by increasing each node's capacity by one. Goto 1.

[3] If blocking is at least `bp-cutoff` Goto 4. Otherwise calculate the robustness parameter twice for the same load prototype $\hat{d}$. Once with reduced capacities decreased by one along $P(i,j)$ and the other with reduced capacities decreased along $P(j,i)$. Choose the path assignment for the request to be the one whose corresponding robustness was larger. If this fails, try the other path. Goto 5.

[4] Choose the path assignment for the request to be the one whose length was shorter. If this fails, then try to assign the longer path.

[5] Update the reduced capacity corresponding to the path selected in **[3]** or **[4]**.
Go to **[1]**

We remark that the success of this algorithm is obviously dependent on a good choice of the prototype $\hat{d}$. If the distribution of path requests is uniform, then the expected distance on the cycle between a demand pair, will be $n/4$. One reasonable prototype is then $\hat{d}_{i(i+\lfloor n/4 \rfloor)} = 1$ for each $i = 0,\ldots n-1$ and 0 for each other ij. We may also wish to alter this demand to ensure that $\rho(d,\tilde{c})$ is nonnegative.

In this paper, we have not discussed a good combinatorial algorithm for solving (10). An easier ORA to implement, but keeping in this paradigm, is to choose the next route so as to avoid the occurrence of a violated 2-cut inequality. This is achieved by replacing Steps **[3]**,**[4]** with a new parameter $\rho_2(d, \tilde{c})$ which is defined as the same minimum as in (10) except that only sets of size one or two are considered.

We implemented versions which are simpler still. Namely, we choose the route for the new request ij to be the one whose minimum reduced capacity for an internal node is larger (such 'widest-shortest' algorithms have been considered in a number of previous applications). If this is the same for both paths, then several tie-breaking rules have been tried: (i) choose the shorter path or (ii) choose the path with fewer nodes of minimum reduced capacity or (iii) choose the path whose second smallest through-capacity is smaller. A final approach is given below; this compares the minimum reduced capacities for each of the two possible paths relative to their lengths.

Weighted 1-cut algorithm

[1] WAIT for event: If a request is terminated go to **[2]**;
If request is initiated go to **[3]**.

[2] If the terminated request is along the path $[i, j]$ update the reduced capacities along this path by increasing each node's capacity by one. Goto 1.

[3] If a request between ij is requested, then let `minreduced[l], minreduced[s]` be the minimum reduced capacity of an internal node on the longer and shorter paths between i and j; also let `len[l],len[s]` be the lengths of the two paths.

[4] If the blocking in the network is at most `bp-cutoff`, and

$$\texttt{minreduced}[\texttt{l}] - \texttt{minreduced}[\texttt{s}] > \texttt{len}[\texttt{l}] - \texttt{len}[\texttt{s}]$$

then try to make the path assignment on the longer path and then on the shorter path.
Otherwise first consider the shorter path and then the longer.

[5] Update the reduced capacity corresponding to the path selected in **[4]**.
Go to **[1]**

The above algorithms aim to use the capacity of the network so as to avoid blocking as long as possible. During the peak periods when blocking is high, our algorithm returns to the shortest path approach [2] since we can no longer afford to waste capacity in avoiding blocking - it has already occurred! We have implemented the weighted 1-cut algorithm and have found that blocking occurs noticeably later than for the shortest path approach which was previously used (as well as any of the traditional widest-shortest algorithms tried).

The following figures - Figure 2, Figure 3 - represent comparisons of the weighted 1-cut and shortest path and clockwise algorithms. The simulation tool OPNET was used to run the comparisons. The network considered consisted of 24 PONS each with 256 pendant users. The Poisson distribution (with interarrival time 450 seconds) was assumed for path requests. Holding times (mean 20 seconds) were assumed to be negative exponentially distributed. Path requests were considered equally likely amongst all node pairs.

[2]First try shortest path, then the secondary path.

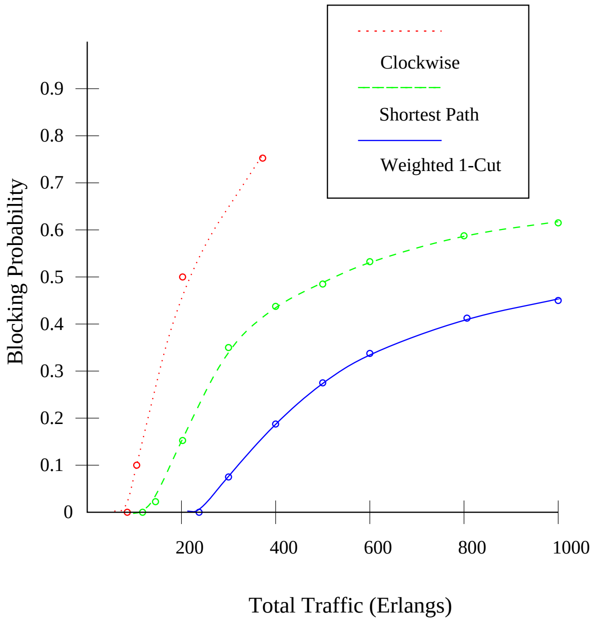


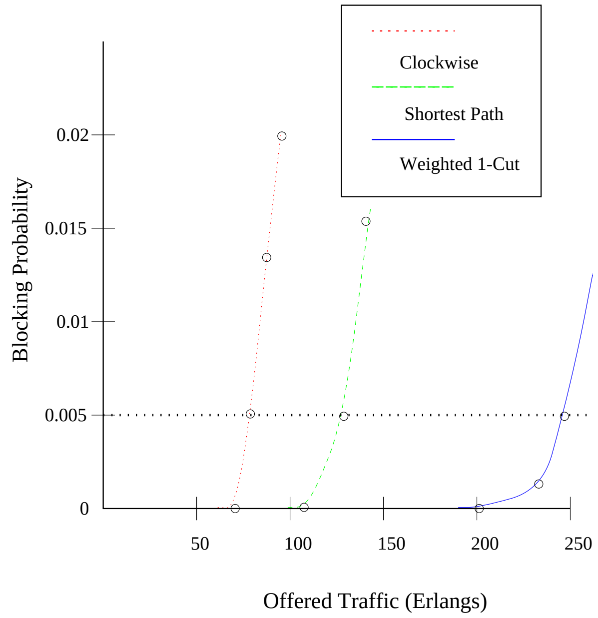
Figure 2:

The algorithm and results reported here are for the case where wavelength conversion is available at the nodes. In the case of wavelength routing, the use of the WEIGHTED 1-CUT algorithm performed similarly to the shortest path and clockwise algorithms - see [15] for the full array of simulation results. In this case, for the ring networks examined, it was seen that in order to achieve similar blocking levels under the wavelength routing regime, the network required 20 percent extra capacity at the nodes. This is far less than the theoretical bound found in [16] which states that any routing achievable with wavelength conversion can also be achieved without converters if the capacities are doubled (their results are for link based routing, but they also hold in the node-capacitated version).

Another possible use of a cut condition such as Theorem A is in lower bounding the best possible blocking probability which can be provided by an ORA. This can be done as follows. If our system ever develops a load $\hat{d}$ which violates some doubled-cut inequality, then regardless of which on-line routing algorithm is used, there will be NO feasible path assignment. Thus the blocking probability of any ORA is at least as large as the probability that some violated cut inequality occurs. For certain distributions, such bounds may be analytically derived.

4 Conclusions

We give necessary and sufficient conditions to be able to find a node-capacitated path assignment for a load (set of path requests) on a ring network. These give a hint to the structure of traffic congestion which leads to blocked requests. Initial implementation shows that the



Expanded Version of Figure

Figure 3:

onset of blocking during congested periods can be delayed by using such cut conditions as a benchmark for how not to route calls in real time. It would be interesting to develop a full-blown version of these algorithms based on a solution technique for (10).

Acknowledgements: Much of the work was performed while the second author was at the Centre for Discrete and Applicable Mathematics at the London School of Economics. The work originated from a BT Fellowship at BT Laboratories, Martlesham Heath, carried out under the supervision of Dr. I. Hawker. Part of the present collaboration was spawned at the DONET annual workshop in London, 1996. DONET is funded under the EU Human Capital and Mobility Program.

References

[1] A.R. Beaumont, I.R. Cade, P.K. Fitch, P. Jenkins, D. Payne, Passive Optical Network For Telecommunications, *EFOC/LAN 1991*, London, June 1991, 481 - 423.

[2] S. J. Clendening, Ring Networks To The Rescue, *Telephone Engineer and Management*, (1992) 50-55.

[3] A.F. Elrefaie, Multiwavelength Survivable Ring Network Architectures, *IEEE International Conference on Communications (ICC)*, Geneva, Switzerland May 23-26, 1993, 1245 - 1251.

[4] A. Frank, Edge-disjoint paths in planar graphs, *J. Combinatorial Theory B*, **39**, (1985), 164-178.

[5] M.R. Kramer, J. van Leeuwen, The Complexity of Wire-Routing and Finding the Minimum Area Layouts for Arbitrary VLSI Circuits in *F.P. Preparata: Advances in Computing Research 2: VLSI Theory*, JAI Press, London (1984) 129-146.

[6] *IEEE Journal of Selected Areas in Communications Special Edition on WDM*, August, 1990 Vol. 8 No 6.

[7] S. Khanna, A Polynomial Time Approximation Scheme for the SONET Ring Loading Problem, *Bell Labs Technical Journal,* Spring, 1997 36-41.

[8] B. Korte, L. Lovász, H.J. Prömel, A. Schrijver Eds., *Paths, Flows and VLSI-Layout*, Springer, Berlin (1990).

[9] M. Middendorf, F. Pfeiffer, On the Complexity of the Disjoint Paths Problem, *Combinatorica* **13** (1), (1993) 97-107.

[10] H. Okamura, P. Seymour, Multicommodity flows in planar graphs, *J. Combinatorial Theory B*, **31**, (1981) 75-81.

[11] N. Robertson, P.D. Seymour, Graph Minors XIII: The Disjoint Paths Problem, *J. Combinatorial Theory B* (1986).

[12] A. Schrijver, P. Seymour, P. Winkler, The Ring Loading Problem, *SIAM J. Discrete Math.* **11** (1) (1998) 1-14.

[13] A. Schrijver, Homotopic Routing Methods, in [8] 329-371.

[14] V. Tandon, M.Wilby, C.Todd, F.Burton, A Novel Upgrade Path For Transparent Optical Networks Based On Wavelength Reuse, *IEEE INFOCOM 95*, Boston, April, 1995.

[15] V. Tandon, LambdaPON: The dynamically reconfigurable Wavelength Multiplexed Passive Optical Network, *University College London*, December, 1995.

[16] G. Wilfong, P. Winkler, Ring Routing in WDM Networks, *Proceedings SODA*, (1998), 333-341.