

Single-Sink Multicommodity Flow with Side Constraints

F.Bruce Shepherd
Department of Mathematics and Statistics
McGill University
Montreal, QC
email: bruce.shepherd@mcgill.ca

Abstract

In recent years, several new models for network flows have been analyzed, inspired by emerging telecommunication technologies. These include models of *resilient flow*, motivated by the introduction of high capacity optical links, *coloured flow*, motivated by Wavelength-Division-Multiplexed optical networks, *unsplittable flow* motivated by SONET networks, and *confluent flow* motivated by next-hop routing in internet protocol (IP) networks. In each model, the introduction of new side-constraints means that a max-flow min-cut theorem does not necessarily hold, even in the setting where all demands are destined to a common node (sink) in the network. In such cases, one may seek bounds on the “flow-cut gap” for the model. Such approximate max-flow min-cut theorems are a useful measure for bounding the impact of new technology on congestion in networks whose traffic flows obey these side constraints.

1 Introduction

We study several network flow models inspired by recent telecommunication technologies. Rather than presenting the models as questions of independent mathematical interest, as they usually are, we also attempt to explain enough about each technology to see how the model was motivated. We are by no means describing engineering solutions to practical problems, but rather we attempt to illustrate how mathematical theory (in most cases still only partially complete) can shed light on the principles underlying network engineering decisions. We believe the theory ultimately gives important intuition to the practitioner as well. As for our choice of topics, we adopt a view expressed by Ford and Fulkerson in their 1962 book on Network Flows [27], “...we have simply written about mathematics which has interested us, pure or applied.”

We start by stating the Max-Flow Min-Cut Theorem; we do this formally in order to fix some of our notation. We are given a directed graph $D = (V, A)$ where V is the set of n nodes, A is the set of m arcs (sometimes called *links*); we may also be supplied with (usually integer) arc capacities $u(a) \geq 0$. Given a specified *origin* node s , and *destination* node t , one asks how much “flow” can be *routed* from s to t ? For our purposes, an *st flow* is usually defined as an assignment of nonnegative values $f(P)$ to simple directed paths P from s to t . The *value* of the flow is then $\sum_P f(P)$ and the flow is *feasible* if for each arc a , we have: $\sum_{P:a \in P} f(P) \leq u(a)$. A set $S \subseteq V$ is an *st set* if $s \in S, t \notin S$; any such set determines an *st cut* $\delta^+(S)$ consisting of all arcs with their tail in S , and head in $V - S$. The *capacity* of such a cut, denoted by $u(S)$ is then $\sum_{a \in \delta^+(S)} u(a)$. The *Max-Flow Min-Cut Theorem* states that there is a flow of value k if and only if the capacity of each *st cut* is at least k . In Menger [44] the essence of a proof for this theorem is given, although the theorem is often ascribed to Ford and Fulkerson [26], as they incorporated capacities and costs to network flows and studied the topic from an algorithmic perspective. Ford and Fulkerson also studied the *multicommodity flow* case, where many pairs of nodes (s_i, t_i) wish to simultaneously use the network capacity to route their individual flow demands. If this is not a unit demand, we often use $d_{s_i t_i}$ to denote the amount requested¹. We also consider flows in undirected networks, in which case analogous definitions will apply; in that case we refer to a graph $G = (V, E)$ with edge set E , and use $\delta(S)$ to denote the undirected edges in a cut induced by S .

Network flows are the focus of a vast amount of research in combinatorics, optimization, operations research and algorithms – see [2] for a comprehensive introduction. Apart from being mathematically rich however, they also form the basis for practical contributions to a wide range of applications. Many of the applications are quite natural, such as transportation problems. In that setting, *flow paths*, i.e., paths P such that $f(P) > 0$, have a natural interpretation as shipments through a network. However, flow problems can also arise in unexpected contexts as critical subproblems whose solutions are part of a larger (practical or theoretical) agenda. This ubiquity establishes flow in networks as one of the fundamental problems in optimization.

SWITCH-SENSITIVE FLOWS.

Network flow theory makes a key assumption on how the computed flow paths are implemented in practice. Consider the instance displayed in Figure 1(a) with three unit-flow paths indicated

¹We need not consider the case of parallel demands between the same nodes.

by bold, dotted and dashed lines (ignore the labels a_i, b_i for now). The flow depicted is feasible as long as each arc has capacity 1, but in practice there may also be constraints imposed by the nodes. We study several flow models where new constraints are imposed by telecom applications or equipment. In this setting, nodes are typically much more complicated objects. For instance, in a network design process, a node may represent a regional subnetwork or an office. One initially designs the “higher layer”, under the assumption that computed traffic flows can ultimately be implemented in the subnetworks when we “open up” nodes and look inside. This *hierarchical approach* thus defers or *outsources* the work on subnetworks (i.e., inside the nodes) to a second phase.²

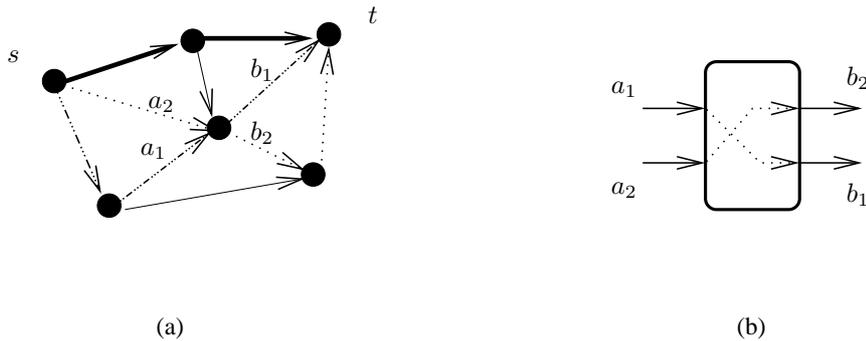


Figure 1:

A “base case” for such a hierarchical design is when each node simply represents a network switch. This approach assumes to some (approximate) extent, that each node is able to *switch*, any two paths that cross it. For instance, in the Figure 1(b), we assume that flow from arc a_1 can be connected to b_1 , and flow from a_2 is connected to b_2 . A *crossbar* is the term used in telecommunication engineering to refer to a switch that is capable of handling any possible demand from its inputs to its outputs, that is, for any matching M from some inputs s_i to some outputs t_i , the node is capable of simultaneously connecting each s_i to its t_i . Crossbars were historically the prevalent architecture within the telephone network and their internal switching networks were quite sophisticated – see Figure 2. Due to the central role played by crossbars, there is a significant literature ([33, 45, 46], cf. also with computer interconnection networks [40]) on the design of switching networks with the crossbar property, and stronger *non-blocking properties* that allow input-output connections to be satisfied dynamically regardless of the order in which requests arrive and leave. The crossbar notion has also figured prominently in theory. One such example dates back to Valiant’s work on circuit complexity lower bounds which led him to so-called super-concentrators possessing a crossbar property. In Section 4 we discuss further applications to the theory of disjoint paths.

The dominance of crossbars probably explains why network flow theory has not focused on more complicated *switch-sensitive* flows. This is in contrast to other well-studied side constraints such as arc gains and losses (*generalized flow*) or delay (*flows over time*). However, modern optical and data network switches are imposing new and sometimes unusual constraints on how traffic can traverse network nodes. One simple example suffices to explain.

²This approach is often reasonable for network design, but it may not be viable for network management where operational end-to-end signals must be established in real time.

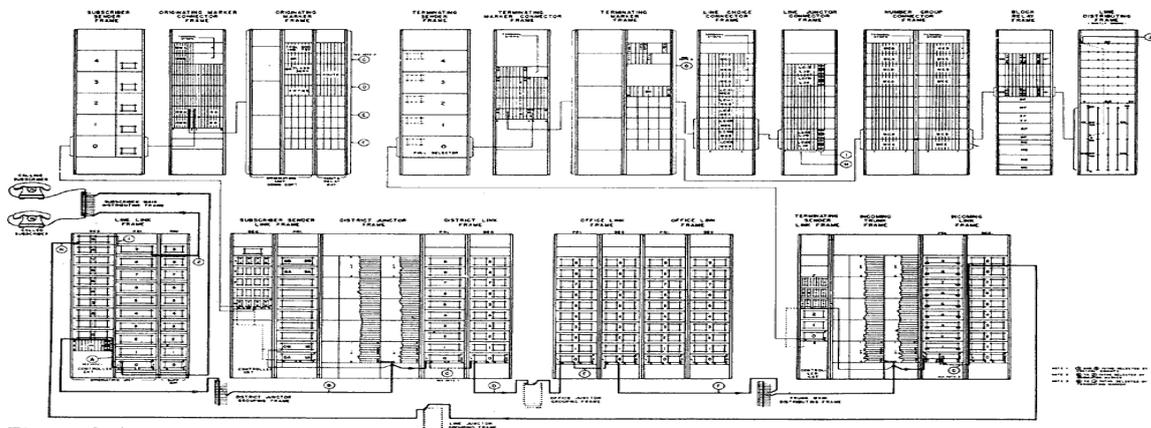
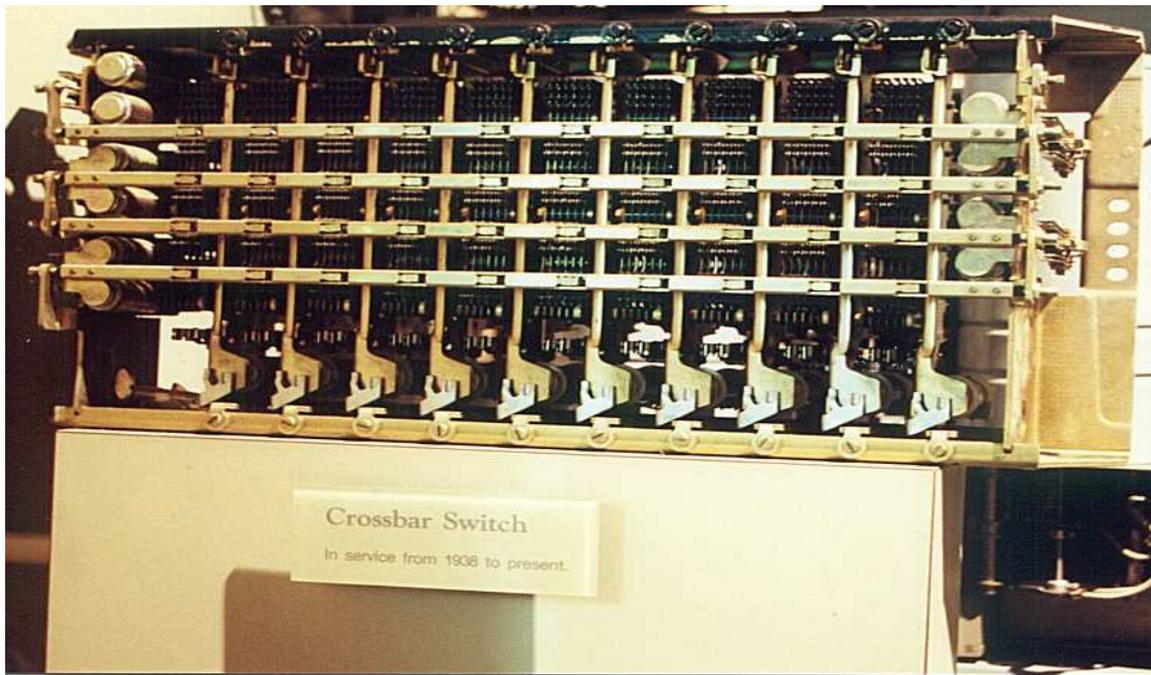


Figure 6-4

Figure 2: “A Crossbar Switch and the call path of a single simple intra-office call (e.g., to a next door neighbor) within a Crossbar No.1 office is diagrammed above. Once your dialed digits were collected, your voice path might be built in a third of a second, and involve the split-second operation of more than ten thousand relays. In practice this was quite a complex process. To create a voice path across a State or across the country required that many additional offices all work in concert to build a specific call path from among the incalculable possibilities; an almost unbelievable achievement of electromechanical computing, executed in fractions of seconds.” *From the Vintage Telephone Equipment Museum: <http://simplethinking.com/photo/phone/vintage/>*

Suppose that all traffic in a network destined to a node t , must obey the following *confluence* property: any flow that enters a node v must exit on a single arc. This is a constraint imposed by early standards for IP (internet protocol) routing. This constraint has a pronounced affect on the basic structure of flow, since any such flow must have a tree-structure – see Figure 5(a). In the next sections we discuss the confluent flow problem and other examples, with an eye to understanding the impact of these constraints from a theoretical vantage point.

FLOW-CUT GAPS AND SINGLE-SINK MULTICOMMODITY FLOW.

We study several examples where network flows must obey additional constraints imposed by a communication technology. In each case, the new technology has certain advantages independent of its affect on network flows. For instance, optical technologies offer huge amounts of cost-effective bandwidth, while IP routers are operationally simple and inexpensive to deploy. Our interest, however, is in the question: *How should we evaluate the impact of new flow constraints on required resources in the (idealized) network?* We adopt two simplifications that allow fairly clean theoretical answers. First, we focus on the commonly used measure of *congestion*. Hence, for us, the “cost” of a new technology will be measured as how much we must uniformly augment the network capacity in order to guarantee a side-constrained flow.

Our second simplification is to focus on the classical single-commodity flow problem. Actually, we consider the extension of single-commodity network flows to so-called *single-sink multicommodity flows* where each node $v \neq t$ in the network has a rational demand $d_{vt} \geq 0$ it wishes to route to the destination (or sink) t . Using such single-sink multicommodity flows as a benchmark for flow problems with side-constraints is inspired by the work of Kleinberg who introduced the Single-Sink Unsplittable Flow Problem.³ For standard flows, a reduction to the single-commodity case is trivial. Add a new node $s \notin V$, and for each $v \in V - t$, add an arc $a_v = (s, v)$ with a capacity of d_{vt} . Evidently, a single-sink multicommodity flow exists if and only if the augmented graph has an st flow of value $\sum_{v \in V-t} d_{vt}$. In many cases, however, constrained network flows are not amenable to this simple reduction to a single commodity flow. For instance, in the *unsplittable network flow* problem, each demand from a node v is required to route its entire flow of d_{vt} on a single path P . While we could add the new node s , we would still need to identify multiple demands between s and t in order to represent the unsplittable flow problem. Hence there is no advantage gained by the reduction; in this sense, the problem, at its heart, is a multicommodity flow problem.

Consider a (unit capacity) single-sink multicommodity flow instance consisting of a digraph $D = (V, A)$ (or undirected graph $G = (V, E)$) and demand pairs s_i, t ; there is a demand of $d_{s_i t}$ from s_i to t . The *density* of a cut $\delta^+(S)$ is defined as $\frac{|\delta^+(S)|}{d(\text{sep}(S))}$; here $\text{sep}(S) = \sum_{s_i \in S} d_{s_i t}$, and it is understood $t \notin S$. Clearly if there is a feasible fractional multicommodity flow, then the density of each cut is at least one, and the *cut condition* is said to hold for the instance. Let Δ denote the density of a minimum density cut. As long as $\Delta \geq 1$, a standard network flow exists. For the problems we discuss, this will not always be the case, and hence we seek $\phi > 0$, the largest value such that there is a (constrained) feasible flow for all *scaled* demand values $\phi d_{s_i t}$. The *flow-cut gap* for an instance is defined as the value $\frac{\Delta}{\phi}$ (there are natural extensions of these definitions to the undirected case and to the case of general multicommodity demands).

³Actually he looked at single *source* flows. Unsplittable flows had been studied earlier by Cosares and Saniee, however Kleinberg coined the actual term “unsplittable”.

Two classical results state that (i) for any single-sink (directed or undirected) multicommodity flow instance the flow-cut gap is 1 [26] (ii) for any undirected multicommodity flow instance the flow-cut gap is $O(\log n)$ and this is tight [41, 42].

For our purposes, the correct interpretation of a flow-cut gap of α is that if you only build your network to satisfy the cut condition, then α is the amount of congestion you can expect on some link in your network. In our presentation we also consider “additive” flow-cut gaps (Theorem 2.3), as well as gaps based on node as opposed to link congestion (Theorem 3.1). In most cases, we use the flow-cut gap to measure the impact or “cost” of forcing our flows to satisfy a new constraint.

2 Optical Networks

2.1 Resilient Flows

The first deployments of optics in telecommunications involved simple point-to-point optical links, such as transatlantic cables. One important side effect of the enormous bandwidth of fibre was that the loss of a single cable could now mean the loss of an unacceptable amount of traffic (think in terms of a city sector versus a few hundred phone calls). The design of *resilient* networks that could withstand failures of its elements, was elevated as an important design objective.

A woeful fact of practical network design is that many initial cost estimates are produced as follows. Suppose we want to carry d_{ij} units of traffic between each node pair i, j . For each i, j find a shortest available $i-j$ path P in the network. Increase the capacity on the links of P by d_{ij} . Repeat. Apart from ignoring existing capacity and upper bounds on link capacity⁴, this approach has a more serious drawback even as a rough cost estimator. Traffic between nodes i, j is completely vulnerable to a single link failure. In this section we study a resilient version of the shortest st path problem that, for instance, could be used in the greedy approach above to give a an cost estimate for a resilient network design. We motivate the ideas first with a simple example.

My first engagements with a network operator were with British Telecom (BT) in the early 1990's. In one consultation, Graham Brightwell and I [10] were asked to examine the following simply resilience problem. One is given n links, between nodes s and t say, where the per-unit cost of reserving bandwidth on link l is c_l . We are also given a threshold T . Find the optimal allocation of capacities to the links such that after deleting any k links, there is still at least T units of capacity between s and t . Linear programming gives a simple answer which we briefly describe.

⁴One is often just “lighting” up equipment that already exists in the network

Consider the case $k = 1$ (the general case is analogous). Let $A = J - I$ be the $n \times n$ matrix (J is the matrix of all 1's) with all 1's except 0's on the diagonal. We require a solution to $\min\{\sum_l c_l x_l : Ax \geq T\mathbf{j}, x \geq 0\}$; \mathbf{j} denotes the column vector of all 1's. Assume $0 \leq c_1 \leq c_2 \leq \dots \leq c_n$. An easy argument shows that any basic solution x^* (of positive and minimum total cost) has the following form: for some integer $p \geq 2$, $x_l^* = \frac{T}{p-1}$ for $l = 1, 2, \dots, p$, and each other component is 0. Furthermore, one can also find the optimal integer solution by a procedure called *up-down rounding*. Take the optimal fractional solution x^* ; without loss of generality $\frac{T}{p-1}$ is not an integer or we are done. We create two integer solutions z^r : one for each choice of $r \in \{\lceil \frac{T}{p-1} \rceil, \lfloor \frac{T}{p-1} \rfloor\}$, as long as $r > 0$. We define $l = \lceil \frac{T}{r} \rceil + k + 1$ and set $z_1^r = z_2^r = \dots = z_{l-1}^r = r$, and $z_l^r = T - (l - 2)r$. All other components of z^r are 0. One may use unimodality of the cost function to establish that one of these two solutions is optimal [5, 11].

This naturally motivates the question of how to plan for resilience in general networks. We now discuss the models proposed in [11]; the above procedure plays an important role in the answer.

In the general setting, how does one plan for resilience. Specifically, how could one define a *shortest "resilient" st path*? The answer depends crucially on how the network responds to a link failure. Two broad restoration categories are: *local restoration* where the endpoints of a link recognize its failure and react to re-route lost traffic, and *endpoint restoration* where traffic between s, t is always restored by the endnodes s, t regardless of where the failed link lies in the network. Both types of restoration exist in the first generation optical networks.

It is yet uncertain what strategies will be adopted in future all-optical networks. A simple and attractive approach is to force traffic between s, t to be routed along disjoint st paths. This scheme is called *diverse path routing* and generalizes the commonly studied $1 + 1$ scheme where capacity is simply reserved on two (edge or node) disjoint paths; hence if any link fails, the other path is still available. This approach is attractive since it allows easy restoration at the endpoints s, t ; some link fails, s and t simply shunt traffic on to one of the nonfailed paths.

If we use such diverse path routing, the resilient network design (or resilient shortest path) problem becomes: find disjoint st paths P_1, P_2, P_3, \dots and capacities u_i for each path P_i , such that if any path fails, the capacity assigned to the remaining paths is at least T (again, the case of allowing $k > 1$ link failures is analogous). Consider such a collection of paths, and let c_i denote the per-unit cost of reserving capacity along the whole path P_i . Suppose the optimal solution assigns positive capacities to a collection of paths P_1, P_2, \dots, P_k , then these paths simply look like direct links connecting s and t . Thus our previous analysis shows that there is an optimal solution that assigns $\frac{T}{k-1}$ capacity to each P_i , and hence the cost of the solution is $\frac{T}{k-1} \sum_i c_i$. It now follows that the best possible choice for the paths P_i is to find a minimum cost st flow of size k . This gives a complete solution for finding the optimal T -resilient diverse path reservation as we now describe.

Find the cheapest pair P_1, P_2 of edge disjoint st paths. Let $C_2 = c(P_1) + c(P_2)$ and x^2 be the solution that assigns T units of capacity to each P_i . For each $k = 3, \dots$, find a minimum cost k -flow $P_1^k, P_2^k, \dots, P_k^k$ between s and t . Set $C_k = \sum_i c(P_i^k)/(k-1)$. If $C_k > C_{k-1}$, then output x^{k-1} , it is the optimal solution. Otherwise, define x^k by assigning $\frac{T}{k-1}$ capacity to each P_i^k .

This simple procedure finds the optimal fractional resilient diverse-path reservation by solving a sequence of minimum cost flow problems. Hence it can be reduced to solving a sequence of shortest path problems, using the successive shortest path algorithm [34]. But what if we want an optimal *integral* solution? One natural heuristic is suggested by the algorithm above. For each computed x^k , apply the up-down rounding procedure. This gives potentially two integral diverse-path reservations $z^{k,+}$ and $z^{k,-}$. Amongst all of these solutions simply choose the cheapest. It turns out that this provides a solution with integral capacities, that is within a factor of $\frac{15}{14}$ of the optimal integral solution. Somewhat bizarrely, this is also best possible!

Theorem 2.1 ([11]). *For any $\epsilon > 0$, it is NP-hard to find an integral diverse-path reservation that is within a multiplicative factor of $\frac{15}{14} - \epsilon$ from the optimal. Moreover, there is a polytime $\frac{15}{14}$ -approximation algorithm for this problem.*

We now discuss an alternative, much weaker, resilient shortest path model that completely ignores any restoration constraints imposed by the network. Indeed, perhaps the most mathematically natural extension of shortest paths that incorporates resilience, is simply to seek arc capacities such that each st cut has capacity at least T , even after a single arc is deleted. This amounts to studying the polyhedron:

$$\mathcal{R}(D, s, t, T) = \{x \in \mathbf{R}^A : x(\delta^+(S) - a) \geq T \text{ for each } st \text{ set } S, \text{ and } a \in \delta^+(S)\}$$

While one may optimize over \mathcal{R} in polytime (indeed it has a simple but large compact formulation), there is no combinatorial method known for this problem. Not surprisingly, there is then not much known about the structure of its extreme points. Several examples are given in [11] - two examples are depicted in Figure 3.

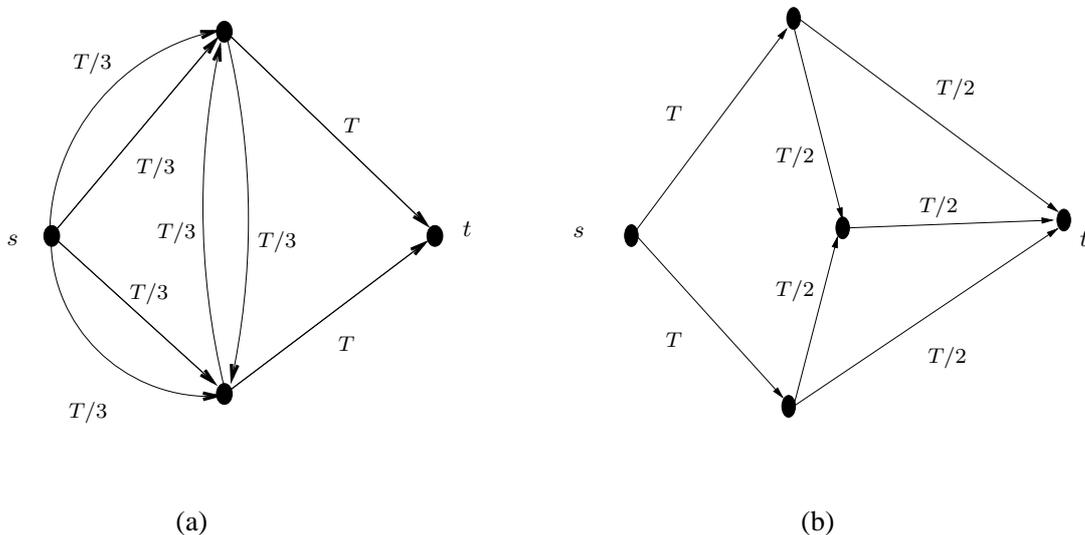


Figure 3: (a) vertex of $\mathcal{R}(D)$ with a cycle in the support; (b) basic solution for both arc-failure and node-failure resilience.

As mentioned, there are many possible, practical models for providing resilience; we discussed one based on diverse paths. The polyhedron \mathcal{R} represents the weakest possible condition one

could impose for resilience. Hence it is a useful benchmark for lower bounds. One can show that its integrality gap is “small”. More significantly for us, it does not yield better than a factor 2 improvement on the cheapest diverse path solution; the proof of this is a fun exercise.

Theorem 2.2 (The Cost of Diverse Path Reservations [11]). *For any nonnegative arc costs c , the cost of reserving T units of capacity on a cheapest pair of disjoint paths, is at most twice the minimum cost solution in $\mathcal{R}(D, s, t, T)$.*

We close by mentioning that in the single-sink flow case with multiple sources, or when the network has arc capacities, the picture is far from complete. The reader is referred to [12] for open problems.

2.2 Unsplittable Flows

The second wave of optics in the telecommunication network began being deployed in the early 1990’s. Instead of just single point to point optical links, the new networks consisted of rings of nodes (switches) with optical links between each pair of neighbouring nodes on the ring. Most importantly, nodes on a ring worked together in terms of detecting network failures and restoring lost traffic across the ring. In America, the standard adopted was called *SONET* (synchronous optical networks) whereas the standard in most of the rest of the world was SDH. In both cases, nodes were able to terminate flow paths (communication paths) whose capacity was from a set of bandwidths prescribed by the standard (e.g., 155, 622 or 2488 Mb/s).

Building the SONET network took place “from scratch” and this was a monumental task. It was also one of the most capital intensive network projects and hence arguably one of the most important network optimizations to have been undertaken. The optimization problem was also enormously complex: where are rings installed? how large is each ring? how is demand routed across multiple rings, before and after network failures? Most American network operators used software developed by Bell Communications Research (Bellcore), called the SONET Toolkit.

In this effort, Cosares and Saniee [21] introduced the following unsplittable flow problem which they called the *Ring Loading Problem*. They are given an undirected ring (cycle) G on nodes $0, 1, 2, \dots, n - 1$ (with edges $i(i + 1)$ for each i) and integer demand matrix $[d_{ij}]$. Routing the demand matrix *unsplittably* means that for each $i < j$, with $d_{ij} > 0$, we choose to route this entire demand either clockwise on the path $i, i + 1, \dots, j$ or counterclockwise on the path $j, j + 1, \dots, i$. We seek the minimum integer L such that there is an unsplittable routing where the maximum load on an edge of G is at most L . They consider the fractional relaxation and define L^* to be the optimal ring load if demands can be fractionally routed, i.e., demands may split their flow across the two directions around the ring. They prove that one may always obtain an unsplittable routing with a load of at most $2L^*$.

Subsequently, Schrijver, Seymour and Winkler (SSW⁵) strengthened the result considerably by showing that an unsplittable routing can be obtained with maximum edge load of $L^* + \frac{3}{2}d_{max}$ ($d_{max} = \max_{i,j} d_{ij}$) - see [51]. They prove this bound via the following result. Consider $2m$ nonnegative reals u_i, v_i for $i = 1, 2, \dots, m$ such that $u_i + v_i \leq 1$. A *signing* of these numbers

⁵The last two authors were also at Bellcore at the time.

is a sequence z_1, \dots, z_m such that for each i , $z_i \in \{v_i, -u_i\}$. Find the minimum value β such that for any given collection of u_i, v_i 's there is a signing such that the following holds for each k :

$$\left| \sum_{i=1}^k z_i - \sum_{i=k+1}^m z_i \right| \leq \beta.$$

They show that one may always obtain an unsplittable ring routing with a maximum edge load of $L^* + \beta d_{max}$. They also prove that $\beta \leq \frac{3}{2}$, thus establishing their bound for the ring loading problem. They actually exhibit an instance of u_i, v_i 's where $\beta > 1$. However, this does not refute the following conjecture which remains open. It also seems temptingly similar to the results we discuss next for single-sink flows.

Conjecture 1 ([51]). *For any instance $G, [d_{i,j}]$ of the ring loading problem, there is an unsplittable routing with maximum edge load at most $L^* + d_{max}$.*

We mention that the algorithm of SSW is also simple to implement and was incorporated in the SONET Toolkit; for a given instance, it was an important subroutine that was solved tens of thousands of times to get quick estimates on the exponentially many possible ring placements.

In 1994, Kleinberg coined the term ‘‘unsplittable’’ flow and initiated the study of unsplittable flows for the single-sink multicommodity case [36] in directed graphs $D = (V, A)$ with arc capacities $u(a)$. By considering a network with two nodes and parallel links, this includes the special case where items can be fractionally packed into bins (of varying capacities) and we wish to integrally pack them without increasing the bin capacities by too much. Kleinberg established that this version of unsplittable flow has a constant flow-cut gap and this was strengthened by Kolliopoulos and Stein [39]. Dinitz, Garg, and Goemans ultimately proved the following tight result.

Theorem 2.3 (The Cost of Unsplittability [22]). *Suppose we have an instance of the single-sink unsplittable flow problem for which there is a feasible fractional flow, then there is an unsplittable flow where the maximum load on each edge e is at most $u(a) + d_{max}$.*

The situation for the minimum cost unsplittable flow is less developed. Goemans ([18], cf. [52]) has conjectured that there is an unsplittable flow using the capacities $(u(a) + d_{max})$ that costs no more than opt , the cost of an optimal fractional flow using capacities $u(a)$. To date, the strongest result known in this direction is due to Skutella [52] who shows the existence of an unsplittable flow of cost opt using capacities $2u(a) + d_{max}$.

2.3 Coloured Flows

In existing SONET networks, optical connections only exist between neighbouring nodes on rings. Incoming information is optically decoded and converted to an electronic signal. Any traffic that is destined for this node is stripped, and new traffic is added before the traffic is rebundled and lasers are then used to transmit a new optical signal. Such an OEO conversion is expensive in terms of delays as well as the equipment need to achieve the individual

tasks. Existing optical networks have also incorporated point-to-point *Wavelength Division Multiplexed* (WDM) fibres or line systems. A WDM fibre is simply one where many different wavelengths (channels) can be transmitted simultaneously. This capability is enabled by installing (moderately expensive) optical terminals within nodes at either end of a link. Typical current WDM fibres may carry anywhere between roughly 4 and 64 channels (although, 1000's of wavelengths have been achieved in laboratory experiments). As the demand for capacity increases, the future optical networks will increasingly take advantage of nodes that can switch paths optically without the need for conversion to electronics, thus producing a genuine all-optical network.

All-optical networking is in its early stages, and it is not entirely clear how future networks will look. These networks will no longer be ring-based as in SONET, but will rather look like general graphs. The most important question is the switching behaviour of network nodes. If nodes are to act as crossbars, as in traditional telecom, then a node should have the ability to make *wavelength translation*; this is a technology that allows a lightpath to change wavelength without conversion to electronics. This point is illustrated in the simple example Figure 4(a) where each link supports two wavelengths, red and blue. There are three paths and no edge is used by more than two paths. If we do not have wavelength translation, then each path must be assigned an end-to-end wavelength (the same wavelength for each of its edges). Obviously this is not possible since each pair of paths intersect in some edge, but we only have two colours. This simple example shows that either we would need at least one node to perform wavelength translation, or more colours per fiber, or we would have to “leave” the optical layer (i.e., convert the signal along some path into electronics, as was done in SONET). Since wavelength translators are extremely expensive, this solution seems unlikely to be adopted as a viable all-optical strategy.

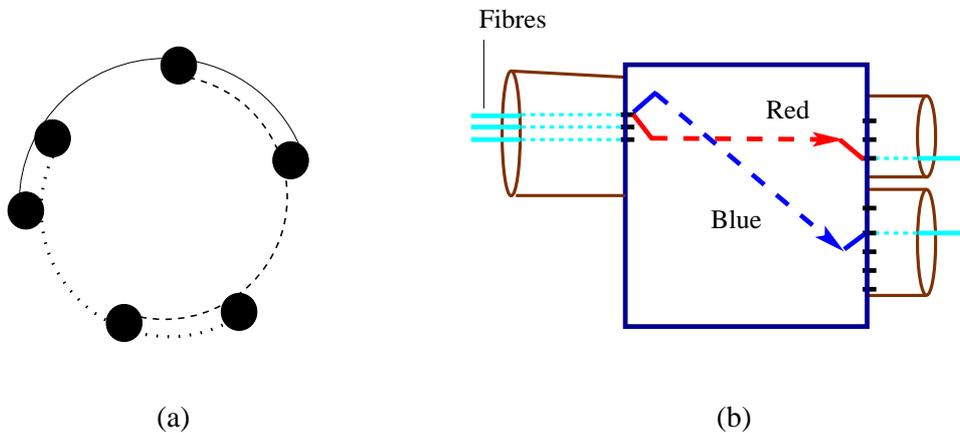


Figure 4: (a) One needs 3 colours for paths (b) A node with spatial switching.

The leading switching alternative are nodes that only do *spatial switching*. This means that a node may connect signals between distinct fibers, as long as the signals are using the same wavelength - see Figure 4(b).⁶ Consider the *minimum fibre* MIN FIB network design problem

⁶Currently there are quite restrictive bounds on the number of fibers that a node is able to terminate. This

that arises in this setting; Winkler and Zhang [55] studied the special case for path graphs, introduced due to its relevance in the design of so-called optical line systems.

We have a directed graph $D = (V, A)$ and for each arc a we may install WDM links each at a cost of $c(a)$, and each supporting λ wavelengths. There is also have a collection of unit demands, given as a set F of ordered node pairs (u, v) . Our goal is to find an optimal set of WDM fibers to activate. That is, we want to install $u(a)$ fibres on each (directed) link, such that the network has enough capacity to route every demand $f \in F$ on a path P_f , and assign this path a wavelength $\lambda_f \in \{1, 2, \dots, \lambda\}$. Since nodes can switch paths between different fibres this problem is potentially much easier than a standard path colouring problem in graphs (which is at least as hard as graph colouring). Indeed we only need to guarantee that the path assignments (colouring) satisfy: $\sum_{f:a \in P_f, \lambda_f = k} 1 \leq u(a)$ for each colour k . Aggregating these constraints, we obtain an obvious necessary condition for the routing P_f and installed capacities $u(a)$:

$$\sum_{f:a \in P_f} 1 \leq \lambda u(a). \quad (1)$$

In general, one would not expect this condition to be sufficient. Surprisingly, one can show that in general networks, a “fractional implies integral” type result holds if we restrict to single-sink multicommodity flow instances. This result can be deduced by seeing that a certain polyhedron has the *integer decomposition property* (IDP) [6]. A polyhedron P has the IDP if for any $x \in P$, and any integer k such that kx is integral, we have $kx = \sum_{i=1}^k x^i$ where each x^i is an integral vector in P . Trivially, any polyhedron with the IDP is integral, i.e., each of its vertices is integral. One can make use of the following characterization of total unimodularity⁷ due to Baum and Trotter [6].

Theorem 2.4 ([6]). *A matrix A is totally unimodular if and only if $\{x : Ax \leq b, x \geq 0\}$ has the integer decomposition property for every integer vector b .*

Consider an arbitrary directed supply digraph $D = (V, A)$ and a single-sink set F of unit demands of the form (s_i, t) for some destination node t and nodes s_1, s_2, \dots, s_k . We wish to determine the number $x(a)$ of fibers to light in each arc a . (The remarks equally apply to the undirected version.) The following flow-gap 1 type result can be established.

Theorem 2.5 (The Cost of Colouring [43]). *If there is a single-sink network integral flow such that the load on any arc a is at most $u(a)\lambda$, then there is a wavelength-assigned such flow that uses at most $u(a)$ fibers on each arc a .*

Proof. Consider the node-arc incidence matrix B for D : for each node v and arc a , $B_{v,a}$ is 1 if a has tail v , -1 if it has head v , and otherwise it is 0. Since B is totally unimodular, we have that $P = \{f : Af \leq u, f \geq 0\}$ has the IDP. Thus if f' is a flow vector whose total load on any arc a is at most $\lambda u(a)$, then $\frac{f'}{\lambda} \in P$, and so by the result of Baum and Trotter, we may decompose f' as $\sum_{i=1}^{\lambda} f^i$ where each f^i is an integral vector in P . As the flow paths determined by any f^i place a load of at most $u(a)$ on any link, there are enough fibers to route these demands on the same wavelength, and the proof is complete. \square

leads to even more complicated network design problems [55, 4, 28, 43].

⁷A $0, +1, -1$ matrix is totally unimodular if each square submatrix has determinant $0, 1$ or -1 .

We now describe an alternative proof based on a simple combinatorial algorithm (see [43]). A flow f is called k -ready (for arc capacities $u(a)$) if the total load $f(a)$ on any arc a is at most $ku(a)$. For each arc a in a k -ready flow f , we define its *requirement* $r(a)$ as: $\max\{0, f(a) - (k-1)u(a)\}$. The main idea is to find a flow f' satisfying some of our demands, so that each arc has $f'(a) \geq r(a)$ and hence we can “reduce” f , roughly stated as $f - f'$, to obtain a new $(k-1)$ -ready flow. If we can choose f' so that $f'(a) \leq u(a)$, then all of the demands routed by f' can use the same wavelength, and hence we may greedily repeat this process to route all demands using k colours on each fibre in the network (i.e., lighting $u(a)$ fibers for each arc a suffices if each fiber has k colours). We now describe this procedure in more detail.

Call an arc a *critical* if $r(a) > 0$. We create an auxiliary digraph as follows. For each arc a , we include a forward auxiliary arc with capacity (upper bound) of $u(a)$ and a lower bound of $r(a)$. For each arc with a load of $(k-1)u(a) - \sigma$ for some $\sigma > 0$ (i.e., its load is less than $(k-1)u(a)$) we include a reverse arc with capacity σ and lower bound of 0. In addition, from each terminal s_i , that sends $d(s_i)$ units of flow to t , we include an arc from t back to s_i with capacity $d(s_i)$. Checking several cases, one deduces that if f is a k -ready flow for u , then in the auxiliary digraph H , we have that for each proper node subset S , the sum of lower bounds on arcs in $\delta_H^+(S)$ is at most the sum of upper bounds on arcs in $\delta_H^-(S)$. Hence Hoffman’s Circulation Conditions [32] imply that there is an integral circulation obeying the lower and upper bounds. Such a circulation corresponds to a routing f' of some of our demands that obeys the capacities $u(a)$. In addition, it is easily checked a $(k-1)$ -ready flow f'' is produced by performing the standard symmetric difference on the flows f and f' .

The circulation f' determines a set of flow paths that can be simultaneously routed on a single wavelength (e.g., k) since we had upper bounds of $u(a)$. We can thus remove these demands, and repeat the process with the $(k-1)$ -ready flow f'' . Note that each circulation problem can be solved simply (and efficiently) by a sequence of shortest path problems, using the successive shortest path algorithm [34].

The previous theorem (and algorithm) also generalizes the so-called p -multi-route flow problem [1]. Indeed multi-route flows correspond to the special case where $u(a) = 1$ for all a . To find a multi-route flow of size say kp , one may compute a k -ready flow (for unit arc capacities) of size kp and then apply the procedure to decompose into the desired k “protected” flows, each of size p . The interested reader is referred to [1] for more background on this problem.

We close by mentioning that the approaches above also give a slight extension of the results for path graphs in [55] to certain directed trees; the key is the total unimodularity of the underlying path-edge incidence matrix. The related problem where one is not given λ , the number of wavelengths per fiber, but wishes to minimize λ is studied for single-sink (multicast) problems in [8]; the above approaches also apply in that setting.

3 Data Networks

The opening of the Internet to commercial interests in the early 1990’s and the advent of browsers led to an explosive growth in terms of data traffic (which now far exceeds the world’s voice traffic). The workhorse for moving this traffic is the IP router. Part of the success of

the Internet is due to the fact that IP routers are relatively inexpensive and operationally simple, compared for instance, to traditional large-scale telecom switches such as the 5ESS whose “five 9’s” reliability is legendary (it was operational 99.999% of the time). Examining the behaviour of such routers inspires the next two flow models we consider.

3.1 Confluent Flows

The strength of IP routing is in the simplicity of its operation which, to a first order of approximation, can be described as follows. A router first builds a *next hop table* that contains an entry for each possible IP address t ; this entry contains the *next hop* for t denoted by $next(t)$. This table is built by means of a protocol (distributed algorithm) which we do not discuss in detail. Once a next hop table is established, upon receiving a packet destined for t , a router simply looks in the table, and forwards the packet to the IP router $next(t)$. It follows that all flow destined to t that passes through v must exit v on a single link – see Figure 5(a). A flow in a digraph with this property is called *confluent* and was introduced by [20]. They proposed that an important indicator of the impact of IP routing on network performance was to measure the degree to which confluent flows force node congestion to increase. It turns out that this congestion could be unbounded. Indeed the example in Figure 5(b) is an instance from [19] which shows that congestion may grow as large as $\Omega(\log n)$ at a node. On the positive side, the congestion is no worse than this.

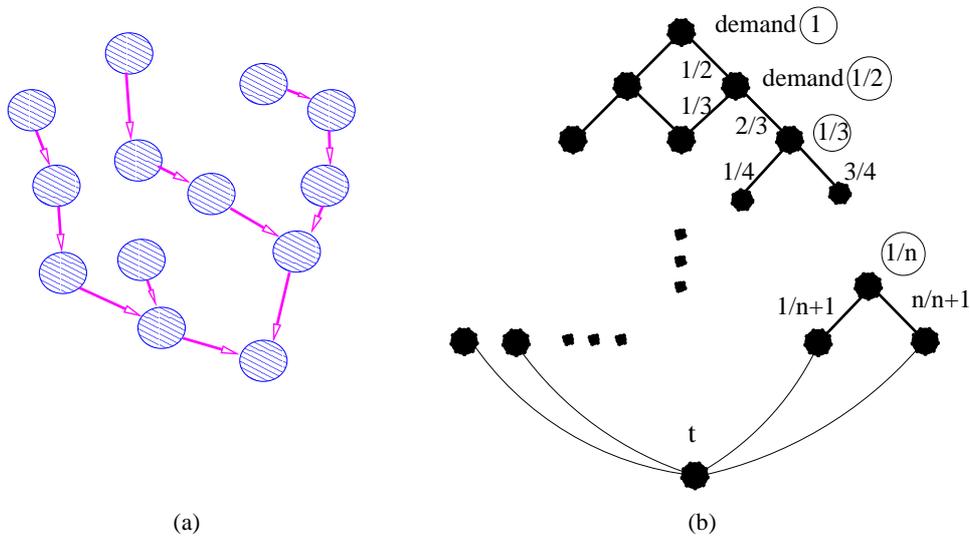


Figure 5: (a) A Confluent Flow (b) An example with $\log n$ congestion. The fractional network flow displayed has a maximum node load of 1; however any confluent flow must route the flow from the top node along a path whose nodes have demands $1, \frac{1}{2}, \frac{1}{3}, \dots$

Theorem 3.1 (The Cost of Confluence [19]). *If a single-sink multicommodity flow instance has a fractional flow with a maximum node load of U , then there is a confluent flow whose maximum node load is $O((\log n)U)$. Moreover, it is NP-hard to determine the optimal node congestion for a confluent flow to within a factor less than $\frac{1}{2} \log n$.*

Several interesting directions remain. For instance, what can one say about instances where the nodes have capacities?⁸ Note that the above result applies only to the uniform capacity case since the bound is in terms of the maximum load U . Another direction is to analyze the behaviour of confluent flows for general multicommodity instances. The algorithm which proves the above theorem is centralized. It would be interesting to understand the ramifications of making a distributed implementation.

3.2 Bifurcated and d -furcated Flows

In our discussion of IP routers, we considered the case where traffic destined to node t must use a single next hop. What happens if we allow a node to send its outgoing traffic on two next hops? Or d next hops? Allowing multiple next hops is a strategy that many network operators allow. We call a single-sink multicommodity flow *bifurcated* (or *d -furcated*) if the support of the flow has the property that the out-degree of any node $v \neq t$ is at most 2 (resp. d). Interestingly, allowing an extra next hop eliminates the unbounded node congestion that occurs with confluent flows. In the following we discuss this result and some of its consequences.

Theorem 3.2 (The Cost of Bifurcation [23]). *If a single-sink multicommodity flow has a fractional flow with a maximum node load of U , then there is a bifurcated flow whose maximum node load is $U + d_{max} \leq 2U$. Moreover, the bound of 2 is best possible.*

In practice, operators may deem it undesirable to split traffic from a common IP address since this leads to the need to perform resequencing of packets at the destination. One common approach is to split traffic using source IP address hashing. In this setting, we then have an additional restriction that requires that traffic from a single source s_i , not be split when it leaves a node. One can handle this refinement as follows. First, apply the methods above to obtain a bifurcated flow. Second, apply Theorem 2.3 to obtain an unsplittable flow.

Theorem 3.3 (Cost of Bifurcation with Unsplit Routes from Source [23]). *If a single-sink multicommodity flow has a fractional flow with a maximum node load of U , then there is a bifurcated flow whose maximum node load is $2U + d_{max}$ (at most $3U$ under the no bottleneck assumption) with the following properties. If a source node s_i routes any demand through a node v , then all of the demand d_i arrives at v , and exits v on a unique arc.*

As one might expect, if the outdegree (number of next hops) d is allowed to grow, the worst case node congestion improves. The same methods used to prove Theorem 3.2 also show that if there is a feasible fractional flow with node congestion one, then there is a d -furcated flow with node congestion at most $1 + \frac{1}{d-1}$, for $d \geq 2$.

Given that an extra next hop ($d = 2$) eliminates unbounded congestion, it is natural to ask what is happening between $d = 1$ (confluence) and $d = 2$ (bifurcation). Again, the proof techniques used for Theorem 3.2 help analyze this question. For $\beta \in [0, 1]$, call a flow β -confluent if at each node v , at least a β fraction of its total flow exits on a single arc, and any remaining flow uses at most one more arc. Thus 1-confluent flows are standard confluent flows, and for any $\beta \leq .5$, a β -confluent flow is just a bifurcated flow. For $\beta \in (.5, 1]$ there is a

⁸A motivation based on interdomain routing is suggested in [54].

spectrum of problems. For $\beta = 1$, the confluent flow problem has unbounded node congestion but this is not the case for any *fixed* $\beta < 1$.

Theorem 3.4 (The Cost of β -Confluence [23]). *Let $\beta \in [\frac{1}{2}, 1)$. Suppose there is a fractional flow for a single-sink multicommodity instance such that for each node v , its load $f(v)$ is at most U . Then there is a β -confluent flow such that for each node v , its load is at most $f(v) + \frac{\beta}{1-\beta}U$.*

Several interesting directions should be explored. As with confluent flows, the situation for general multicommodity flows, non-uniform capacities, costs, or extensions to distributed algorithms have not yet been addressed. It is also natural to ask about a special class of bifurcated flows where each node must split its flow equally along its outgoing arcs. For instance, in practice, this would be the case if an IP router chose its next hop for each packet based on a simple round-robin scheme. We call such flows *halftuent*.

3.3 Network Coding

Consider a single source (we now have a single source s not sink t) multicommodity flow problem, where there are some k destination nodes t_1, t_2, \dots, t_k . The source itself is the location of several bit streams b_1, b_2, \dots, b_q ; the bits in each stream must be transmitted to each destination t_i . Viewed as a flow problem, this gives rise to a commodity for each pair b_j, t_i . Creating a commodity for each pair, however, is wasteful since it does not take advantage of capacity sharing for a fixed stream b_j . Instead one should think of reserving a tree (or rather an arborescence) on which the bits in b_j are routed. This is referred to as a *multicast* routing problem. One thus naturally seeks q disjoint arborescences rooted at s , each of which includes all of the terminals t_i . It turns out, however, that sometimes we may not even need q disjoint trees! We show that this is sometimes the case when our network's nodes are allowed extra processing power (this section is less about adding node constraints to flow, but rather *removing* them!). We obviously require the following *cut condition* be satisfied: if S contains s but not all of the t_i 's, then $|\delta^+(S)| \geq q$.⁹

In traditional *store-and-forward* networks, packets arriving at an internal node are stored, and a copy is forwarded to the next node. Naturally, the number of packets placed on a link between two nodes cannot exceed its channel capacity, and hence this gives rise to a standard integer multicommodity flow problem. Prevailing wisdom had suggested (cf. [56]) that there was no need for data processing within network nodes beyond the capability to replicate packets. In [3] a surprisingly simple example showed that this was not valid. Consider the digraph of Figure 6 where there are two bit streams, denoted by a, b , at node s , to be transmitted to nodes t_1, t_2 .

In the first network, we simply reserve two disjoint arborescences rooted at s , one for each bitstream. In the second network, one easily checks that there does not exist two disjoint arborescences. We are still able, however, to transmit all bits to their destinations t_i , by

⁹The cut condition is not normally sufficient to yield a packing of arborescences, a problem which is NP-hard. There is a sharp characterization, however, when the set of terminals is the set of all nodes V (Edmonds [24]). Namely, there exists q arc-disjoint arborescences rooted at s if and only if the cut condition holds.

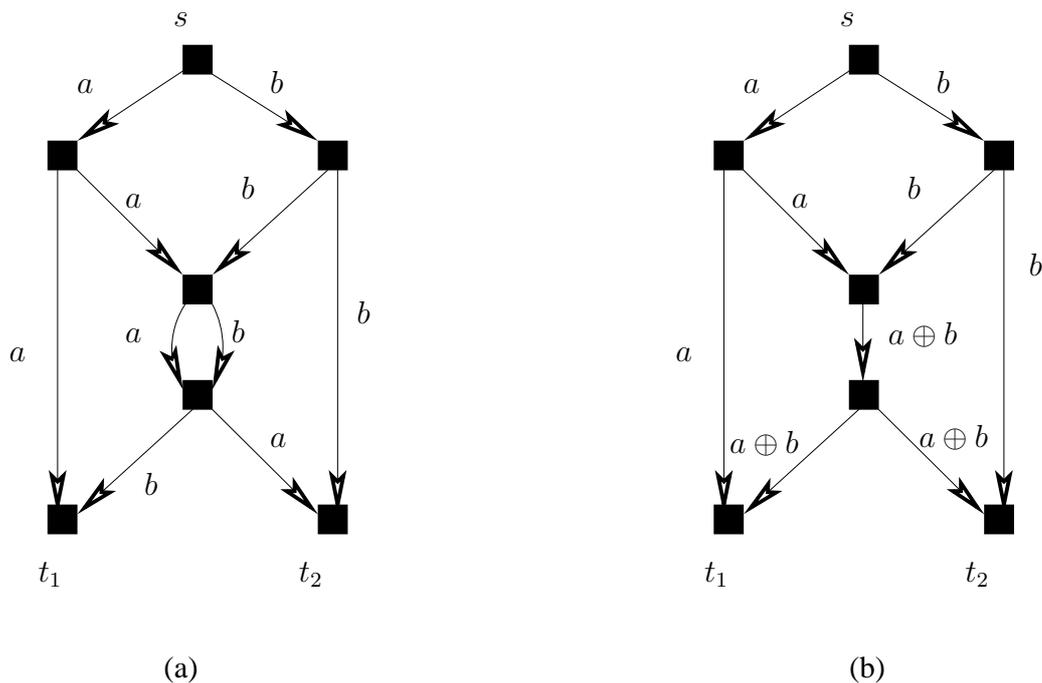


Figure 6: (a) without network coding (b) with network coding.

means of coding (in this case, performing an XOR) at an internal node, and decoding at nodes t_1, t_2 . This example has been proved to extend to any instance where a collection of bitstreams are co-located at a single origin node. Namely, if the cut condition holds then, possibly with coding/decoding at nodes, all streams can be transmitted to the destinations [3]. This surprising result has sparked a huge amount of work in the new area of network coding. There are several interesting open questions in this area. One that seems most relevant to flow theory is whether “coding helps” (as it does in the above example) in the case of undirected multicommodity instances. We have only roughly touched on the topic of network coding; the interested reader is referred to [56, 30] for a more thorough background.

4 Multicommodity Flows via Crossbars and Single-Sink Flows

We close with some comments on how the combination of single-sink flows and crossbars have played a role for general multicommodity flow and disjoint path problems.

Our introductory remarks described traditional network flows as assuming that nodes can behave as crossbars, i.e., that any collection of paths crossing through a node is feasible, as long as arc capacities are satisfied. The problem of how the interconnection of paths is actually achieved within a node is ignored, or delayed to a later subproblem. This concept of “outsourcing” work to crossbars has been used in a number of theoretical routing results. Probably the most celebrated example is the work of Robertson and Seymour [49, 48], and specifically their algorithm for the undirected disjoint path problem with a fixed number of

pairs s_i, t_i . An important scheme in their work is to look for a well-connected subgraph that behaves like a crossbar. The terminals s_i, t_i are then routed to the crossbar along disjoint paths (this amounts to a single-sink flow problem). If the pairs are able to reach the crossbar, then they can be matched up appropriately using the crossbar. The most basic crossbar is simply a single node. Another classical example is a grid since an $n \times n$ grid can act as a crossbar for the nodes (terminals) in any fixed grid row. No matter how we place $n/2$ pairs s_i, t_i on distinct nodes in a row, one may connect them using edge disjoint paths. A clique is another simple example of a crossbar. Routing (outsourcing) to a clique minor is used in the Robertson-Seymour scheme. A similar approach is used by Kleinberg [37] to route a super-constant number of pairs in a half-disjoint fashion. Routing to a single node crossbar is the basis for an $O(\sqrt{n} \log n)$ -approximation for the maximum arc-disjoint path problem in acyclic directed graphs [16]. This is further developed to show that the natural LP relaxation for maximum edge-disjoint paths in undirected graphs is $O(\sqrt{n})$ [17] (and hence $\Theta(\sqrt{n})$). Routing to a grid is used by Chekuri et al. [15] to give a polylogarithmic approximation for the maximum edge-disjoint path problem in planar graphs; an extension for even graphs is given in [38]. Andras Frank [31] indicated that a similar idea of outsourcing to a crossbar appears in early work of Jung on disjoint paths in directed graphs [35].

This same general approach has also been used for (fractional) multicommodity flows. In that setting, a fundamental crossbar is given by the results of Leighton and Rao [41] on uniform (or product) multicommodity flow. They show that for any undirected graph $G = (V, E)$ and any set of *terminals* W , if $|\delta(S)| \geq \min\{|W \cap S|, |(V - S) \cap W|\}$, then there is a *uniform multicommodity flow* with $O(\log n)$ edge congestion. In other words, one can route $\frac{1}{|W|}$ between each pair of nodes in W , and send at most $O(\log n)$ flow on any edge. One deduces that given any matching s_i, t_i on the nodes of W , one may fractionally satisfy these demands using the uniform multicommodity flow above (this is also the essential idea behind Valiant's randomized load balancing). Hence G behaves like a *fractional crossbar* for the terminals W . Routing to such fractional crossbars has been an important ingredient in various places. These include Räcke's [47] breakthrough on low-congestion oblivious routings, and the work [14] on finding a polylogarithmic approximation for the *all-or-nothing multicommodity flow problem*. Both of these also rely on single-sink flows when routing to a fractional crossbar.

Acknowledgments. I am grateful for stimulating discussions and input from colleagues Steve Fortune, David Johnson, Peter Winkler, Gord Wilfong, and Francis Zane. This paper has benefitted from many suggestions and enjoyable collaborations with Chandra Chekuri. I also thank the two anonymous reviewers whose detailed input pointed out numerous ways for improving the article. Finally, I wish to give thanks to Bernhard Korte for supervising my postdoctoral work in Bonn in 1991. His view that there were plenty of practical problems that needed mathematical development apparently rubbed off on me.

References

- [1] C. Aggarwal and J. Orlin. "On Multi-Route Maximum Flows in Networks". *Networks*, **39**, 2002, 43–52.

- [2] R. Ahuja, T. Magnanti and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, 1993.
- [3] R. Ahlswede, N. Cai, S.-Y. R. Li, and R.W. Yeung. “Network information flow”, *IEEE Trans. Inform. Theory*, vol. IT-46, 2000, 1204–1216.
- [4] Elliot Anshelevich and Lisa Zhang, “Path Decomposition Under a New Cost Measure with Applications to Optical Network Design”, *European Symposium on Algorithms (ESA)*, 2004, 28–39.
- [5] J.J. Bartholdi, J.B. Orlin, H.D. Ratliff, “Cyclic scheduling via integer programs with circular ones”, *Oper. Res.* **28** (1980) 1074–1085.
- [6] S. Baum and L. E. Trotter Jr., “Integer rounding and polyhedral decomposition for totally unimodular systems”, *Optimization and Operations Research (Proceedings Workshop Bad Honnef, 1977; R. Henn, B. Korte, W. Oettli, eds.) [Lecture Notes in Economics and Mathematical Systems 157]*, Springer, Berlin, 1978, 15–23.
- [7] B. Beauquier, J.C. Bermond, L. Gargano, P. Hell, S. Perennes U. Vaccaro. “Graph problems arising from wavelength-routing in all-optical networks”. *Theoretical Computer Science*, **233** 1-2, 2000, 165–89.
- [8] B. Beauquier, P. Hell, S. Perennes. “Optimal wavelength-routed multicasting. *Discrete Applied Mathematics* **84** (1998) 15–20.
- [9] D. Bienstock, G. Muratore, “Strong inequalities for capacitated survivable network design problems”, *Math. Programming, Series A*, **89** (2000) 127–147.
- [10] G. Brightwell, F.B. Shepherd, “Consultancy report: Resilience strategy for a single source-destination pair”, *LSE CDAM Report 96-22*, Aug. 1996.
- [11] G. Brightwell, G. Oriolo, B. Shepherd, “Reserving resilient capacity in a network”, *SIAM J. on Discrete Math.* **14** (2001) 524–539.
- [12] G. Brightwell, G. Oriolo, B. Shepherd, “Reserving Resilient Capacity for a Single Commodity with Upper Bound Constraints”, *Networks*, **41** (2), 2003, 87–96.
- [13] Chandra Chekuri and Paul Claisse and Rene Essiambre and Steven Fortune and Dan Kilper and Karun Nithi and Wonsuck Lee and Iraj Saniee and Bruce Shepherd and Gordon Wilfong and Chris White and Lisa Zhang, “Design Tools for Transparent Optical Networks”, *Bell Labs Technical Journal*, vol 11, no. 2, 2006, 129–143.
- [14] C. Chekuri, S. Khanna, and F. B. Shepherd. The All-or-Nothing Multicommodity Flow Problem. *Proc. of the ACM Symposium on Theory of Computing (STOC)*, 2004.
- [15] C. Chekuri, S. Khanna, and F. B. Shepherd. Edge-Disjoint Paths in Planar Graphs. *Proc. of IEEE Foundations of Computer Science (FOCS)*, 2004.
- [16] C. Chekuri and S. Khanna. Edge Disjoint Paths Revisited. *Proc. of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2003.

- [17] C. Chekuri, S. Khanna, and F. B. Shepherd. “An $O(\sqrt{n})$ approximation and integrality gap for Disjoint paths and Unsplittable Flow” *Theory of Computing* 2/7, (2006), 137–146.
- [18] Bellairs Combinatorial Optimization Open Problems, Editors: G. Costain, S. Kennedy, C. Meagher, <http://www.math.mcgill.ca/~bshepherd/Bellairs/bellairs2007.pdf>
- [19] J. Chen, R. Kleinberg, L. Lovasz, R. Rajaraman, R. Sundaram and A. Vetta, “(Almost) tight bounds and existence theorems for confluent flows”, *Proceedings of the 36th ACM Symposium on Theory of Computing (STOC)*, 2004, 529–538.
- [20] J. Chen, R. Rajaraman, and R. Sundaram, “Meet and merge: approximation algorithms for confluent flow”, *Proceedings of the 35th ACM Symposium on Theory of Computing (STOC)*, pp373-382, 2003.
- [21] S. Cosares, I. Saniee, An optimization problem related to balancing loads on SONET rings, *Telecommunications Systems* **3** (1994) 165-181.
- [22] Y. Dinitz, N. Garg and M. Goemans, “On the single-source unsplittable flow problem”, *Combinatorica*, **19**, 1999, 17–41.
- [23] P. Donovan, F.B. Shepherd, A. Vetta, G.T. Wilfong, Degree-constrained network flows, *Proc. of the ACM Symposium on Theory of Computing (STOC)*, 2007.
- [24] J. Edmonds, Edge-disjoint branchings, in: *Combinatorial Algorithms* (Courant Computer Science Symposium 9, Monterey, CA, 1972; R. Rustin ed.), Algorithmics Press, NY 1973, 91–96.
- [25] J. Fong, A. C. Gilbert, S. Kannan, and M. Strauss, “Better alternatives to OSPF routing”, *Algorithmica* (Special issue on network design), **43(1-2)**, 113–131.
- [26] L. R. Ford jr., D. R. Fulkerson, Maximal flow through a network, *Canadian J. Math.*, **8**, 1956, 399-404.
- [27] L. R. Ford jr., D. R. Fulkerson, Flows in Networks, *Princeton University Press, Princeton, NJ*, (1962).
- [28] S. Fortune and W. Sweldens and L. Zhang, “Line system design for DWDM networks”, *Proceedings of the 11th International Telecommunications Network Strategy and Planning Symposium (Networks)*, 2004, 315-320.
- [29] B. Fortz and M. Thorup, “Optimizing OSPF/IS-IS weights in a changing world”, *IEEE Journal on Selected Areas in Communications* (Special Issue on Recent Advances on Fundamentals of Network Management), **20(4)**, 2002, 756–767.
- [30] C. Fragouli and E. Soljanin, “Network Coding Fundamentals”, *Foundations and Trends in Networking. Hanover, MA: NOW Publishers Inc.*, vol. 2, no. 1, 2007, 1–133.
- [31] A. Frank, personal communication, 2005.
- [32] A.J. Hoffman. “Some recent applications of the theory of linear inequalities to extremal combinatorial analysis”. In *Combinatorial Analysis* edited by R. Bellman and M. Hall. AMS, Provident , RI; 1960, 113–128.

- [33] Frank H. Hwang. “The mathematical theory of nonblocking switching networks”, *World Scientific, Series on Applied Math.*, Vol. 11, 1998.
- [34] M. Iri. “A new method of solving transportation-network problems”. *Journal of Operations Research Society of Japan*, **3**, 1960, 27–87.
- [35] H.A. Jung. “Einer Verallgemeinerung des n -fachen Zusammenhangs für Graphen. *Mathematische Annalen*, **187**, 1970, 95–103.
- [36] J. Kleinberg, “Single-source unsplittable flow”, *Proceedings of the 37th Symposium on Foundations of Computer Science (FOCS)*, 1996, 68–77.
- [37] J. M. Kleinberg. Decision algorithms for unsplittable flow and the half-disjoint paths problem. *Proc. of the ACM Symposium on Theory of Computing (STOC)*, 1998, 530–539.
- [38] J. M. Kleinberg. An Approximation Algorithm for the Disjoint Paths Problem in Even-Degree Planar Graphs. *Proc. of the Foundations of Computer Science (FOCS)*, 2005.
- [39] S. Kolliopoulos and C. Stein, “Improved approximation algorithms for unsplittable flow problems”, *Proceedings of the 38th Symposium on Foundations of Computer Science (FOCS)*, 1997, 426–435.
- [40] F. T. Leighton. “Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes”. *Morgan Kaufman Publishing*, 1992.
- [41] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *JACM*, 46(6):787–832, 1999. Preliminary version in *Proc. of the Foundations of Computer Science (FOCS)*, 1988.
- [42] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995. Preliminary version in *Proc. of IEEE Foundations of Computer Science (FOCS)*, 1994.
- [43] A. McGregor and B. Shepherd. Island hopping and path colouring, with applications to WDM network design. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007.
- [44] K. Menger, Zur allgemeinen Kurventheorie, *Fund. Math.*, **10**, 1927, 96-115.
- [45] N. Pippenger. “Telephone Switching Networks”, *AMS Proc. Symp. Appl. Math.*, 26 (1978) 101–133.
- [46] Nicholas Pippenger, Communication networks, Handbook of theoretical computer science (vol. A): algorithms and complexity, *MIT Press, Cambridge, MA*, 1991.
- [47] H. Räcke. Minimizing congestion in general networks. *Proc. of the Foundations of Computer Science (FOCS)*, 2002.
- [48] N. Robertson and P. D. Seymour. Outline of a disjoint paths algorithm. In B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver, Eds., *Paths, Flows and VLSI-Layout*.

- [49] N. Robertson, P.D. Seymour, Graph minors XIII: the disjoint paths problem, *J. Combinatorial Theory B*, **63** (1), 1995, 65-110.
- [50] A. Schrijver, Theory of Linear and Integer Programming, *John Wiley and Sons, NY*, 1986.
- [51] A. Schrijver, P. Seymour, P. Winkler, “The Ring Loading Problem”, *SIAM J. Discrete Math.* **11** (1) (1998) 1-14; reprinted in *SIAM Review* **411**, 1999, 777–791.
- [52] Martin Skutella. “Approximating the single source unsplittable min-cost flow problem”, *Math. Programming Series B*, **91**, 2002, 493–514.
- [53] B. Shepherd and A. Vetta, “Visualizing, finding and packing dijoins”, in D. Avis, A. Hertz, O. Marcotte (eds.), *Graph Theory and Combinatorial Optimization*, Kluwer, 2005, 219–254.
- [54] F.B. Shepherd, G.T. Wilfong, Multilateral Transport Games, *INOC 2005*.
- [55] P. Winkler and L. Zhang, “Wavelength assignment and generalized interval graph coloring”, *Proc. of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2003, 830-831.
- [56] R.R. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang. “Network Coding Theory”, *NOW Publishing*, Boston-Delft 2006.