A Mixed-Integer Approach to Storage Area Network Design using Generic Network Components

Michael O'Sullivan*
Cameron Walker,
Department of Engineering Science, University of Auckland,
Auckland, New Zealand

March 20, 2006

Abstract

With the current trend toward centralised storage, storage area networks are becoming a critical part of commercial computer networks. In this paper we present a formal definition of the storage area network design problem. We summarise an approach from Hewlett-Packard Laboratories for solving this problem that considers all possible network components and uses mixed-integer programming to select a design. We also summarise a preprocessing method that significantly reduces the size of this formulation. We then present a new formulation that uses generic components instead of including all possible components. We modify the preprocessing method to provide an cost function for the generic components. The size of our generic formulation is a significant reduction from the size of the Hewlett-Packard formulation. We compare the two formulations using a small example storage area network design problem.

Key words: Integer programming; network design; storage area networks

1 Introduction

Institutions that handle large amounts of data use storage area networks (SANs) to improve data management. SANs consist of *links*, *hubs* and *switches* (standard components in many computer networks) and allow data to flow between the *hosts*

*Initial research performed as an employee of Hewlett-Packard Laboratories, Palo Alto, California, USA.

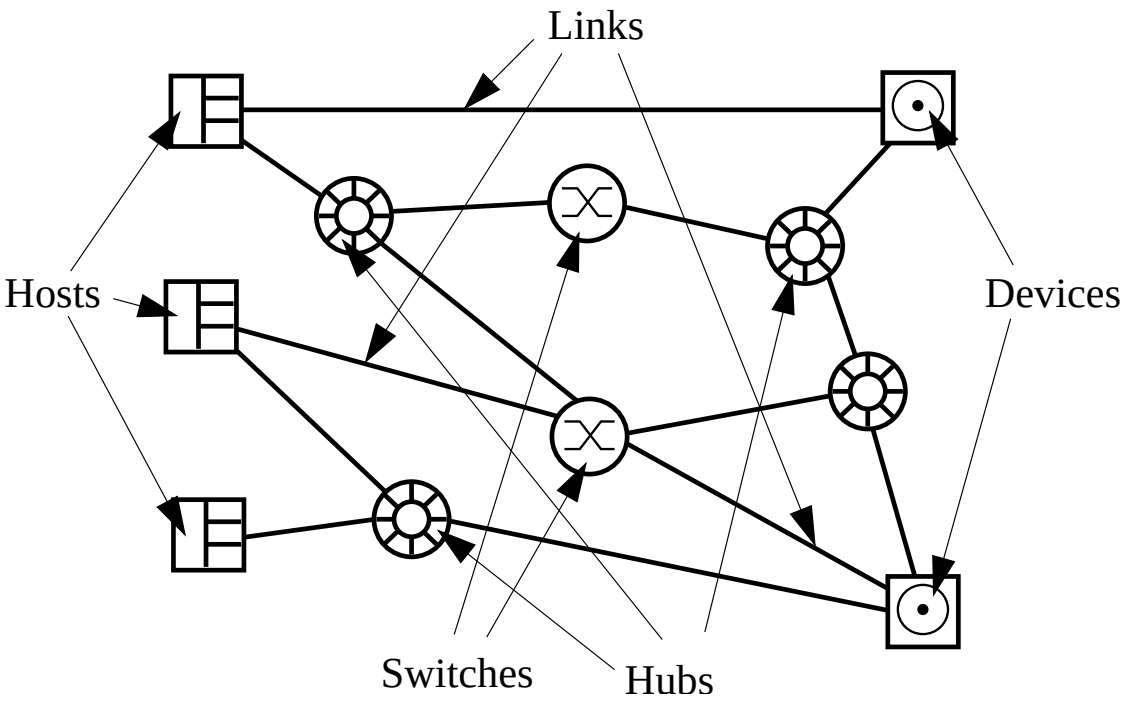


Figure 1: Example SAN

that use the data (client machines and/or servers) and the storage *devices* for the data. Figure 1 shows an example of a SAN.

The SAN design problem (SANDP) requires a configuration of links, hubs and switches to simultaneously support the bandwidth requirements for a set of data *flows*. Each flow starts at a host, moves through the SAN, and finishes at a device. The path that a flow follows must provide enough bandwidth capacity to support the flow's bandwidth requirement. The design of the SAN is restricted by the number of ports and bandwidth capacity of the hosts, devices, links, hubs and switches. The objective of the SANDP is to minimise the cost of the SAN design.

1.1 Background

There are many problems in the network design literature that share common features with the SANDP. If each flow must follow a unique path, then the SANDP extends the non-bifurcated network loading problem ([13], [14], [3], [17], [1]), which is NP–hard. If we allow the flows to be split along multiple paths, the SANDP simplifies to the multicommodity network design problem ([19], [5], [16], [4], [8], [2]), which is also NP–hard, even in the single commodity case ([12]). The SANDP also combines elements of degree-constrained network design ([10], [12], [9]) with node costs ([14]) and node capacities ([11], [23]).

A number of survey papers have been written on the related problem of hub location

([18], [7], [22]). This problem involves the connection of a group of nodes via a network of switches and links, resulting in two subnetworks: the "backbone" and the "tributary". In the SANDP the backbone is a "mesh" of hubs and switches with each tributary being a "star" of links to the hosts and/or devices. Reliability can be incorporated by considering "dual homing" which creates two disjoint paths for each flow. The only work on the mesh/star topology in hub location literature uses heuristics [14, 6] to design the networks. Gavish [14] also uses a Lagrangian relaxation of a classical network design formulation to provide lower bounds for the design cost. However, his formulation does not include many aspects considered in the SANDP including degree limits for nodes and bandwidth capacities on nodes and links.

In Gross and Yellen's comprehensive survey of the graph theory literature [15], Mirchandi and Simchi-Levi [20] discuss communication network design models including capacitated network design with varying bandwidth capacities along links. However, their models also don't include degree limits for nodes and bandwidth capacities on nodes.

The only previous research on the SANDP [24, 21] uses a classical network design mixed-integer programming (MIP) approach. Ward et al. [24] refer to a MIP model without presenting a formulation. O'Sullivan et al. [21] present a preprocessing method for significantly reducing the size of Ward et al.'s formulation. The formulation includes all possible links, multi-hubs (see section 3 for a description of multi-hubs) and switches with binary variables to determine if each component is used or not. The preprocessing method finds the cheapest multi-hub for each combination of number of ports and bandwidth capacity that may be needed in the SAN (for more detail see section 4). This reduces the size of the formulation because all other multi-hubs are removed from consideration.

We present a new MIP formulation that considers *generic* links, multi-hubs and switches. We extend O'Sullivan et al.'s preprocessing method to provide a costing mechanism for these generic components. The size of our new formulation is significantly smaller than that of Ward et al. (even after applying preprocessing to their formulation).

1.2 Summary

In this paper we present a new formulation for the SANDP that puts generic components into the network and uses MIP to determine their properties. We modify an existing preprocessing method to find the optimal component type for any possible set of component properties. Our preprocessing method returns 3 "technology tables" (one for switch types, one for "multi-hub" types and one for link types) that may be used to identify the optimal component type once the component properties have been established. We show how these tables induce (optimal) piecewise-linear cost functions on component properties and embed the technology tables and cost functions within the new generic MIP formulation. When solving the generic MIP

formulation, we focus on finding the best properties for each generic component and the MIP then determines the appropriate component type.

Section 2 contains our description of the SANDP, including the notation used throughout this paper. We summarise the "Hewlett-Packard" MIP formulation [24] for SAN design in section 3. In section 4 we summarise our preprocessing method (slightly modified from that in [21]) for generating technology tables and show how to use the technology tables to significantly reduce the size of the Hewlett-Packard MIP formulation. We model the technology table using mathematical programming in section 5 and embed this model within our generic MIP formulation for the SANDP in section 6. Finally, we give some preliminary results comparing the two formulations in section 7.

2 The SAN Design Problem

The SANDP requires the construction of a minimum-cost network consisting of links, hubs and switches to support a given flow of data between the hosts and devices. First, we briefly discuss each of the SAN entities:

- **hosts** are machines that generate requests for data such as web servers or individual client machines. Hosts have a limited number of ports and each port has a limited bandwidth capacity for data flow;
- **devices** are machines that store data such as disk drives, disk arrays or tape drives. Devices also have a limited number of ports with each port having a limited bandwidth capacity for data flow;
- **links** are cables that transport data. The bandwidth capacity for the data flow through a links depends on the type of cable used. Some examples of cabling technology are SCSI, ethernet and fibre channel. Current SANs often utilise fibre channel cables because of their large bandwidth capacity;
- **hubs** allow for multiple machines (hosts, devices and/or switches) to share data. Data flows into a hub via a link from a machine port, visits all other ports connected to the hub (to determine where it should leave) and then exits via the appropriate port. Since the data must visit every connected port, the bandwidth capacity for data flowing through the hub is restricted not only by the hubs, but also by all incident links and the (external) ports connected to those links. Therefore, there is no need to connect any machine to a hub more than once (since no extra bandwidth capacity is provided). Note that connecting two hubs together only provides extra connectivity (i.e., ports), not extra bandwidth capacity;
- **switches** act as a set of ports with the ability to route flow from one port to another. The only limitation is the number of ports and the bandwidth capacity of each port;

- **flows** are a demand for enough bandwidth capacity to allow data to flow between a specific host and specific device.

To construct a SAN we must select links, hubs and switches from the sets of possible link, hub and switch types, then decide how to configure the chosen components to provide the required connectivity and bandwidth capacity between the hosts and devices.

2.1 Definitions

Let H be the set of hosts and D be the set of devices. For each $h \in H$ define $\eta(h)$ as the number of available ports on h and $\beta(h)$ as the bandwidth capacity of those ports (similarly define $\eta(d)$ and $\beta(d)$ for $d \in D$). Let F be the set of flows and for each $f \in F$ define the required bandwidth capacity $\beta(f)$ for data to flow between the source host $\theta(f)$ and destination device $\delta(f)$.

Let $\mathcal{L}, \mathcal{H}$ and $\mathcal{S}$ be the set of available link, hub and switch types respectively. The parameters of the link, hub and switch types are as follows:

- each link type $t \in \mathcal{L}$ costs $\gamma(t)$ and has a given bandwidth capacity $\beta(t)$;
- each hub type $t \in \mathcal{H}$ costs $\gamma(t)$, has a given number of ports $\eta(t)$ costing $\pi(t)$ each, and bandwidth capacity $\beta(t)$;
- each switch type $t \in \mathcal{S}$ costs $\gamma(t)$, has a given number of ports $\eta(t)$ costing $\pi(t)$, and *port* bandwidth capacity $\beta(t)$.

These definitions will be used throughout the remainder of this paper.

3 The Hewlett-Packard MIP Formulation

Ward et al. give an MIP formulation for solving the SANDP [24, 21]. Their classical network design formulation adds all "possible" components and then selects the optimal network using MIP. However, modelling the behaviour of the hub-to-hub connections presents a challenge. One could group interconnected hubs together into a set, use binary variables to represent membership in a particular set, and give each set the appropriate behaviour. However, the number of possible sets grows exponentially with the number of hubs. Instead, it is easier to generate possible *multi-hub* types $\mathcal{M}$ (that is, one or more hubs connected by links), ban connections between multi-hubs and add all possible multi-hubs. Figure 2 shows the multi-hub (highlighted) from the example SAN (see figure 1) with the *internal* link and *external* links indicated. The internal link is modelled implicitly with its bandwidth capacity (along with the bandwidth capacities of the hubs) determining the bandwith capacity

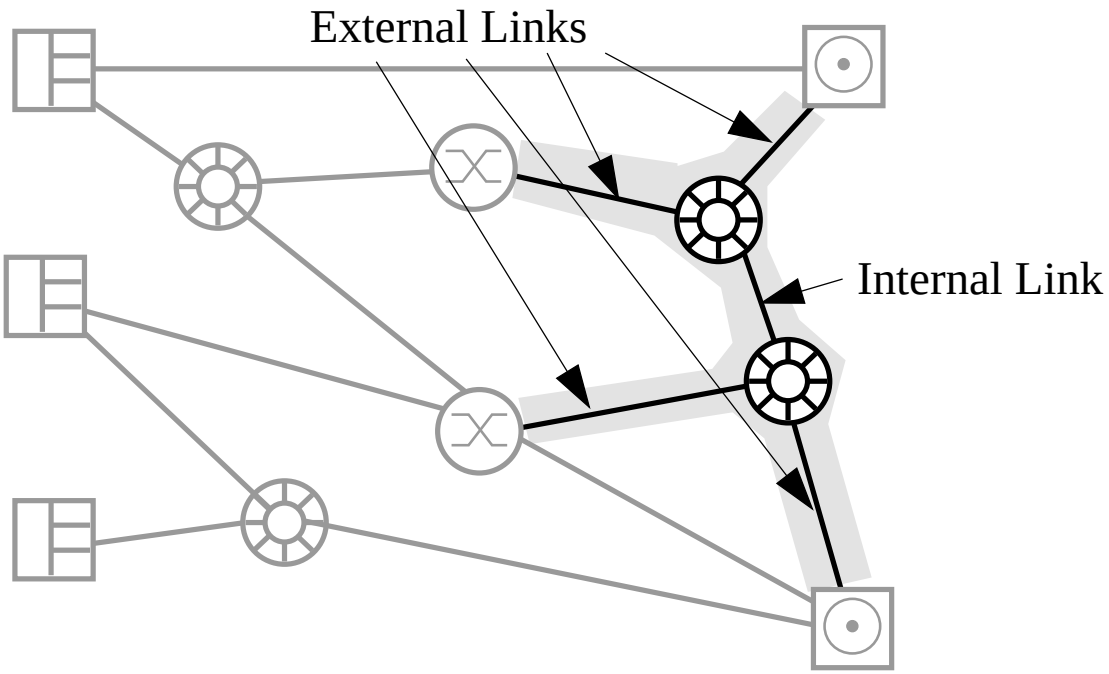


Figure 2: Example Multi-hub

of the multi-hub. The external links are modelled explicitly and their bandwidth capacities limit flow through the multi-hub, see (6).

Each multi-hub type $t \in \mathcal{M}$ costs $\gamma(t)$, has a given number of ports $\eta(t)$ costing $\pi(t)$ each, and bandwidth capacity $\beta(t)$ (note that a multi-hub's *port* bandwidth capacity is the same as its bandwidth capacity). By examining the SANDP link types, hub types and flows we can determine a priori lower and upper bounds on $\eta(t)$ and $\beta(t)$ for any multi-hub $t \in \mathcal{M}$ in an optimal SAN design. We can then enumerate all possible combinations of hub types that provide a multi-hub type with properties within these bounds. For each combination we use the cheapest link type that supports the bandwidth capacity. Thus, given $\mathcal{L}$, $\mathcal{H}$ and F we can determine the set of possible multi-hub types $\mathcal{M}$.

Heuristics developed by Hewlett-Packard Laboratories [24] can provide upper bounds on the cost of a SAN fabric design for any specific problem. This gives a bound on the number of possible components (links, multi-hubs and/or switches) in an optimal SAN. We may also infer upper bounds on the number of links, multi-hubs and switches by examining the SANDP input.

The Hewlett-Packard formulation uses these bounds to add all possible links L, multi-hubs M and switches S into the network. Each link l, multi-hub m and switch s has a given link type $\tau(l)$, multi-hub type $\tau(m)$ and switch type $\tau(s)$ respectively. For brevity we refer to the set of multi-hubs and switches as *nodes* $N \equiv M \cup S$. Each link has a start point $\varphi(l) \in H \cup N$ and an end point $\omega(l) \in N \cup D$. Two multi-hubs may not be linked, so either $\varphi(l)$ or $\omega(l)$ must lie outside M. Also, we

don't allow links to "reflect" each other, so there is no pair of links l and l' with $\varphi(l) = \omega(l')$ and $\omega(l) = \varphi(l')$.

3.1 Decision Variables

Given the set of candidate links L and nodes N, we need to decide which ones to keep. We also need to decide which links will take a flow from its host to its device. Define

$$\begin{aligned} u_n &= 1 \text{ if we keep node } n & v_l &= 1 \text{ if we keep link } l \\ &= 0 \text{ otherwise} & &= 0 \text{ otherwise} \\ a_{fl} &= 1 \text{ if link } l \text{ caries flow } f \text{ forwards} & b_{fl} &= 1 \text{ if link } l \text{ caries flow } f \text{ backwards} \\ &= 0 \text{ otherwise} & &= 0 \text{ otherwise} \end{aligned}$$

Note that links from hosts (to devices) will only carry flow originating at that host (terminating at that device), so we fix $a_{fl} = b_{fl} = 0$ if l starts from a different host, that is $\varphi(l) \in H \setminus \{\theta(f)\}$ (ends at a different device, that is $\omega(l) \in D \setminus \{\delta(f)\}$), and remove these variables from the problem. Also, in the SANDP we only consider flow moving from the hosts to the devices (although in reality data moves both ways along a flow's path), so flow cannot go into hosts or leave devices. Therefore, we can set $b_{fl} = 0$ for any link l leaving a host or entering a device ($\varphi(l) \in H$ or $\omega(l) \in D$). These variables are also removed from the problem.

3.2 Objective Function

Each node has the properties (number of available ports, port cost, bandwidth or port bandwidth, cost) of its (multi-hub or switch) type and each link also has the properties (cost, bandwidth) of its type. For brevity, we omit the type from our notation, i.e., $\gamma(l) \equiv \gamma(\tau(l))$ and $\beta(n) \equiv \beta(\tau(n))$. The total cost of the SAN design is the cost of including the nodes and links

$$Z = \sum_{n \in N} \gamma(n) u_n + \sum_{l \in L} \underbrace{(\pi(\varphi(l)) + \gamma(l) + \pi(\omega(l)))}_{\text{cost of link and incident ports}} v_l \tag{1}$$

In [21] the port cost of multi-hubs is assumed to be 0, but we do not make that assumption here. We show how to deal with non-zero multi-hub port costs when building the technology table (see section 4).

3.3 Flow Constraints

The flow constraints ensure that flow is routed through the SAN in an appropriate manner. Each flow f should only travel along a link l in one direction (otherwise bandwidth would be wasted without moving the flow anywhere)

$$a_{fl} + b_{fl} \leq 1, \;\; f \in F, l \in L \tag{2}$$

Flows first leave their host, then move through nodes in the network until entering the appropriate device (these constraints are similar to flow consevrvation constraints in network flow formulations)

$$\begin{aligned}
\sum_{\substack{l\in L\\ \varphi(l)=h(f)}} a_{fl} &= 1, && f\in F\\
\underbrace{\sum_{\substack{l\in L\\ \varphi(l)=n}} a_{fl} + \sum_{\substack{l\in L\\ \omega(l)=n}} b_{fl}}_{\text{flow going out of } n} &= \underbrace{\sum_{\substack{l\in L\\ \omega(l)=n}} a_{fl} + \sum_{\substack{l\in L\\ \varphi(l)=n}} b_{fl}}_{\text{flow going into } n}, && f\in F, n\in N\\
\sum_{\substack{l\in L\\ \omega(l)=d(f)}} a_{fl} &= 1, && f\in F
\end{aligned} \tag{3}$$

3.4 Component Constraints

Component constraints ensure that components in the SAN obey their specific properties. Hosts, devices and nodes can only connect to as many links as they have available port slots

$$\begin{aligned}
\sum_{\substack{l\in L\\ \varphi(l)=c}} v_l + \sum_{\substack{l\in L\\ \omega(l)=c}} v_l &\le \eta(c), && c\in H\cup D\\
\sum_{\substack{l\in L\\ \varphi(l)=n}} v_l + \sum_{\substack{l\in L\\ \omega(l)=n}} v_l &\le \eta(n)u_n, && n\in N
\end{aligned} \tag{4}$$

The total bandwidth of flows through a multi-hub is restricted by the multi-hub, its ports and all links and ports directly connected to it. We model the multi-hub bandwidth restriction

$$\underbrace{\sum_{f\in F}\beta(f)\Big(\sum_{\substack{l\in L\\ \varphi(l)=m}} a_{fl} + \sum_{\substack{l\in L\\ \omega(l)=m}} b_{fl}\Big)}_{\text{flow out of } m} \le \beta(m)u_m,\ m\in M. \tag{5}$$

By defining the *actual* link bandwidth capacity

$$\hat{\beta}(l) = \min\{\beta(l), \beta(\varphi(l)), \beta(\omega(l))\}$$

we can restrict flow (through the multi-hub m) to be less than the bandwidth capacity of each incident link l and its ports $\varphi(l)$ and $\omega(l)$ (including the port on m)

$$\sum_{f\in F}\beta(f)\Big(\sum_{\substack{k\in L\\ \varphi(k)=m}} a_{fk} + \sum_{\substack{k\in L\\ \omega(k)=m}} b_{fk}\Big) \le \hat{\beta}(l) + \beta(m)(1-v_l),\ l\in L, \varphi(l)=m \text{ or } \omega(l)=m. \tag{6}$$

The bandwidth of flows through a switch is only restricted by the bandwidth capacity of its ports, which is dealt with by the link bandwidth constraints (7).

The total bandwidth of flows through a link is restricted by the actual link bandwidth capacity.

$$\sum_{f \in F} \beta(f)\,(a_{fl} + b_{fl}) \leq \hat{\beta}(l),\ \ l \in L \tag{7}$$

3.5 Example

To demonstrate the Hewlett-Packard MIP formulation (and compare it to our new generic formulation) we will use a small SANDP example. This example problem has 5 hosts and 5 devices with 25 flows between the hosts and devices. The hosts and devices all have 2 port slots available and a port bandwidth capacity of 100 MB/s. The cost of using a port slot on any host is \$600, while there is no cost for port slots on the devices. The bandwidth requirements for the flows are given in Table 1.

Table 1: Flow requirements for example SANDP

Name	Host	Device	Requirement (MB/s)	Name	Host	Device	Requirement (MB/s)
Flow1	Host3	Device1	17	Flow14	Host3	Device4	37
Flow2	Host2	Device2	49	Flow15	Host3	Device5	71
Flow3	Host5	Device3	16	Flow16	Host5	Device4	36
Flow4	Host4	Device4	21	Flow17	Host3	Device3	9
Flow5	Host3	Device5	12	Flow18	Host1	Device4	8
Flow6	Host1	Device1	36	Flow19	Host5	Device1	52
Flow7	Host2	Device2	34	Flow20	Host2	Device2	6
Flow8	Host5	Device3	49	Flow21	Host5	Device1	19
Flow9	Host4	Device4	59	Flow22	Host1	Device3	15
Flow10	Host4	Device5	70	Flow23	Host1	Device2	77
Flow11	Host4	Device1	4	Flow24	Host2	Device5	3
Flow12	Host4	Device2	7	Flow25	Host1	Device5	8
Flow13	Host2	Device3	83				

There are 5 link types, 4 hub types and 3 switch types available to build the SAN. Table 2 contains the properties of these component types.

Even though this is a small problem compared with those encountered in industry (where the number of hosts and devices can reach around 100), the size of the Hewlett-Packard MIP formulation is very large. We know any multi-hub must have at least 3 ports (or we could use a single link) and its minimum bandwidth capacity will be the minimum bandwidth capacity of the links and hubs (in our example this is 30 MB/s). The maximum bandwidth capacity of a multi-hub is the maximum bandwidth capacity of the links because we must carry flow into the multi-hub

Table 2: Example component types for SAN design

Name	Cost ($)	Port Cost ($)	Bandwidth Capacity (MB/s)	Ports Available	Port Bandwidth Capacity (MB/s)
Link1	60	–	30	–	–
Link2	100	–	50	–	–
Link3	120	–	60	–	–
Link4	140	–	70	–	–
Link5	150	–	100	–	–
Hub1	6000	0	50	16	–
Hub2	10000	0	75	10	–
Hub3	11000	0	50	20	–
Hub4	15000	0	100	16	–
Switch1	24000	1000	–	16	100
Switch2	36000	1000	–	32	100
Switch3	30000	500	–	32	75

using some link (in our example this is 100 MB/s). Using this maximum bandwidth capacity we can find the most flows that may be routed through a single multi-hub by solving a knapsack problem (in our example at most 10 flows can be routed through a multi-hub with 100 MB/s bandwidth capacity). Since each flow needs one port to enter a multi-hub and one port to leave the multi-hub the maximum number of ports needed for any multi-hub is twice the maximum number of flows that can be routed through a multi-hub (in our example $2 \times 10 = 20$ ports). Thus we find lower and upper bounds on the number of ports (3 and 20, respectively) and the bandwidth capacity (30 and 100 MB/s, respectively) of any multi-hub. Table 3 shows all possible multi-hub types for our example SANDP (as well as the available switch types).

For our example SANDP, we estimated an upper bound on the optimal SAN design cost by first creating a design that uses the switch type with the most ports and highest port bandwidth capacity (we use the Switch2 switch type for our example). We need at most one port in and one port out for every flow (since switches route flows internally), so we need at most 50 switch ports for our example. This requires 2 Switch2 switches. Solving with these switches available gives a design that costs $56,300 (and uses only 1 Switch2 switch). Using this as an upper bound we add all possible multi-hub and switch types with links of all available types to connect them (remember links are banned between multi-hubs and there is no need for more than one link between a multi-hub and a host, device or switch). Table 3 gives the number of each node type we included in our candidate node set N.

Table 3: Multi-hubs and switches for example SANDP

Name	Cost ($)	Port Cost ($)	Bandwidth Capacity (MB/s)	Ports Available	Port Bandwidth Capacity (MB/s)	Number Included
H1	6000	0	50	16	–	9
H2	10000	0	75	10	–	5
H3	11000	0	50	20	–	5
H4	15000	0	100	16	–	3
H1-H1-L1	12060	0	30	30	–	4
H1-H2-L1	16060	0	30	24	–	3
H1-H3-L1	17060	0	30	34	–	3
H1-H4-L1	21060	0	30	30	–	2
H2-H2-L1	20060	0	30	18	–	2
H2-H3-L1	21060	0	30	28	–	2
H2-H4-L1	25060	0	30	24	–	2
H3-H3-L1	22060	0	30	38	–	2
H3-H4-L1	26060	0	30	34	–	2
H4-H4-L1	30060	0	30	30	–	1
H1-H1-L2	12100	0	50	30	–	4
H1-H2-L2	16100	0	50	24	–	3
H1-H3-L2	17100	0	50	34	–	3
H1-H4-L2	21100	0	50	30	–	2
H2-H2-L2	20100	0	50	18	–	2
H2-H3-L2	21100	0	50	28	–	2
H2-H4-L2	25100	0	50	24	–	2
H3-H3-L2	22100	0	50	38	–	2
H3-H4-L2	26100	0	50	34	–	2
H4-H4-L2	30100	0	50	30	–	1
H1-H1-L3	12120	0	50	30	–	4
H1-H2-L3	16120	0	50	24	–	3
H1-H3-L3	17120	0	50	34	–	3
H1-H4-L3	21120	0	50	30	–	2
H2-H2-L3	20120	0	60	18	–	2
H2-H3-L3	21120	0	50	28	–	2
H2-H4-L3	25120	0	60	24	–	2
H3-H3-L3	22120	0	50	38	–	2
H3-H4-L3	26120	0	50	34	–	2
H4-H4-L3	30120	0	60	30	–	1
H1-H1-L4	12140	0	50	30	–	4
H1-H2-L4	16140	0	50	24	–	3
H1-H3-L4	17140	0	50	34	–	3
H1-H4-L4	21140	0	50	30	–	2
H2-H2-L4	20140	0	70	18	–	2
H2-H3-L4	21140	0	50	28	–	2

continued on next page

Table 3: Multi-hubs and switches for example SANDP

continued from previous page						
Name	Cost ($)	Port Cost ($)	Bandwidth Capacity (MB/s)	Ports Available	Port Bandwidth Capacity (MB/s)	Number Included
H2-H4-L4	25140	0	70	24	–	2
H3-H3-L4	22140	0	50	38	–	2
H3-H4-L4	26140	0	50	34	–	2
H4-H4-L4	30140	0	70	30	–	1
H1-H1-L5	12150	0	50	30	–	4
H1-H2-L5	16150	0	50	24	–	3
H1-H3-L5	17150	0	50	34	–	3
H1-H4-L5	21150	0	50	30	–	2
H2-H2-L5	20150	0	75	18	–	2
H2-H3-L5	21150	0	50	28	–	2
H2-H4-L5	25150	0	75	24	–	2
H3-H3-L5	22150	0	50	38	–	2
H3-H4-L5	26150	0	50	34	–	2
H4-H4-L5	30150	0	100	30	–	1
Switch1	24000	1000	–	16	100	2
Switch2	36000	1000	–	32	100	1
Switch3	30000	500	–	32	75	1

The MIP formulation with these node and link sets contains for this example problem contains 212941 ($\approx 2e^5$) variables and 143753 ($\approx 1.5e^5$)constraints. Analysing the number of variables and constraints in more detail shows that there are: 141 variables for node existence; 11300 for link existence; 201500 for flow across links; 119000 constraints for restricting flow forward and back across a link (2); 3575 for conserving flow around the network (3); 151 for defining connections to hosts, devices and nodes (4); 137 for restricting flow by hub bandwidth capacity (5); 9590 for restricting flow by link connections to hubs (6); and 11300 for restricting flow by link bandwidth capacity (7). This problem takes just over $4\frac{1}{2}$ hours to solve in CPLEX 9.0.0 (AMPL Version 20021031 running on a Compaq nx9110 with 890 MB RAM) giving an optimal design cost of $44,330.

In section 4 we show how to use a preprocessing method to eliminate sub-optimal link types, multi-hub types and switch types from consideration. By eliminating these extraneous component types we can considerably reduce the size of the Hewlett-Packard MIP formulation. We demonstrate this size reduction using our example at the end of section 4.

4 Generating Technology Tables

In this section we discuss a priori generation of technology tables for the components of a SAN. These tables give the optimal component (link, multi-hub or switch) type to use for all possible combinations of component properties. We use these technology tables to reduce the size of the Hewlett-Packard MIP formulation (by only including link, multi-hub and switch types from the tables). We also use the tables to define the piecewise cost functions for the generic components in our new MIP formulation (see section 6).

The cost of a component consists of two parts: a fixed cost for purchasing the component; and a linear cost for using the component properties. Our preprocessing method creates *technology tables* of optimal component types for given component properties. For links, the only relevant property is the bandwidth capacity, so we can create a table that gives the least cost link type for each specified bandwidth. Table 4 shows the appropriate link technology table for our example problem (given in section 3).

Table 4: Example technology table for link types

Bandwidth Capacity (MB/s)	1–30	31–50	51–60	61–70	71–100
Link Type (Cost in $)	Link1 (60)	Link2 (100)	Link3 (120)	Link4 (140)	Link5 (150)

For multi-hubs, we need to connect hosts, device and switches while supplying a specified bandwidth capacity. We can increase the number of possible connections by adding another hub, although this does not increase the bandwidth (due to the hub technology limitations discussed in section 3). The relevant properties are the number of connections and bandwidth capacity the hub supports. O'Sullivan et al. [21] use a knapsack formulation to calculate an upper bound on the number of connections needed for a given bandwidth capacity when the flow requirements are known. Using this bound they calculate the entire table sequentially using small integer programming formulations. However, they assume no cost per connection, which is not true in general. Here, we modified their method to include the costs of the connections (even though they are small compared to the cost of a hub). Each entry in the table corresponds to the least cost multi-hub configuration that supplies the required number of connections and bandwidth capacity. The multi-hub technology table for the example problem is given in Table 5.

Finally, we can create a similar table for switches. However, since connected switches don't affect each other, we only need to create a table for single switch types. The relevant properties for switches are the number of ports on the switch and the bandwidth capacity of the *switch ports*. Table 6 shows the switch technology table for the example problem.

Table 5: Example technology table for hub types

Multi-Hub Configurations		Bandwidth Capacity (MB/s) 1–30	31–50	51–60	61–70	71–75	76–100
Connections	3–10	Hub1	Hub1	Hub2	Hub2	Hub2	Hub4
		Link1	Link2	Link3	Link4	Link5	Link5
	11–16	Hub1	Hub1	Hub4	Hub4	Hub4	Hub4
		Link1	Link2	Link3	Link4	Link5	Link5
	17–18	Hub3	Hub3	2 Hub2	2 Hub2	2 Hub2	2 Hub4
		Link1	Link2	Link3	Link4	Link5	Link5
	19–20	Hub3	Hub3	Hub2	Hub2	Hub2	2 Hub4
		Link1	Link2	Hub4	Hub4	Hub4	Link5
				Link3	Link4	Link5	

Table 6: Example technology table for switch types

Switch Types		Bandwidth Capacity (MB/s) 1–75	76–100
Ports	3–11	Switch1	Switch1
	12–16	Switch3	Switch1
	17–32	Switch3	Switch2

The technology tables presented here give us the optimal components to use for any (necessary) combination of component properties. From these tables we see that the only multi-hubs we need to consider from Table 3 are: H1, H2, H3, H4, H2-H2-L3, H2-H2-L4, H2-H2-L5, H2-H4-L3, H2-H4-L4, H2-H4-L5, H4-H4-L5 (Table 3 gives the number of each of these multi-hub types to consider and we still need to include all links and switches). Table 7 gives a comparison of the size of the Hewlett-Packard MIP formulation for the example problem with and without preprocessing to remove extraneous components. When using preprocessing the size of the problem is reduced by between 25% and 40%. This problem solves in just over 65 minutes in CPLEX 9.0.0 (AMPL Version 20021031 running on a Compaq nx8220 with 512MB RAM) to give the optimal design cost of $44330.

The tables in this section not only describe the best technology available for particular component properties, but they also define a piecewise linear cost function depending on the properties of the components. For links, the cost function is a step function, depending on bandwidth capacity. For multi-hubs, the function moves stepwise in terms of bandwidth capacity, but piecewise linear (possibly discontinuous) in terms of the number of connections. This is similar for switches, with a step function in terms of the port bandwidth capacity and a piecewise linear function in terms of the number of ports on the switch. In section 5 we describe how to model these piecewise cost functions within an MIP formulation, allowing for "generic" components to be included in the formulation.

Table 7: Comparison of MIP formulation size

		Preprocessing?		Decrease
		No	Yes	(% to 1dp)
Variables		212941	78199	36.7
Node Existence		141	39	27.7
Link Existence		11300	4160	36.8
Flow Across Links		201500	74000	36.7
Constraints		143753	50219	34.9
Restricting Flow to One Direction	(2)	119000	42500	35.7
Flow Conservation	(3)	3575	1025	28.7
Define Connections	(4)	151	49	32.5
Hub Bandwidth Capacity	(5)	137	35	25.5
Link Connections to Hub	(6)	9590	2450	25.5
Link Bandwidth Capacity	(7)	11300	4160	36.8

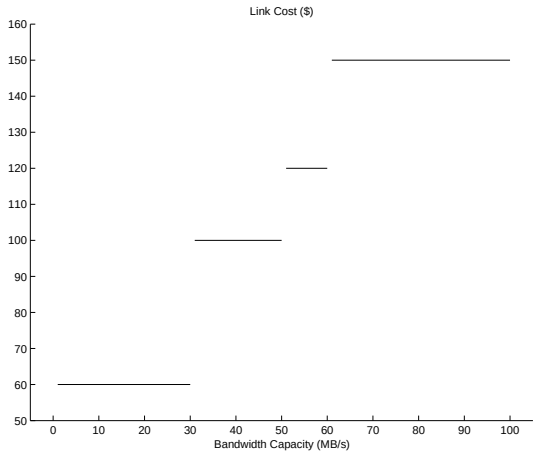
5 Modelling the Cost Function

Once we have generated technology tables for the components (see section 4), we can form the induced piecewise linear cost functions for the different component types. In this section we describe how to determine these functions from the technology tables and show how to model these functions within an MIP formulation.

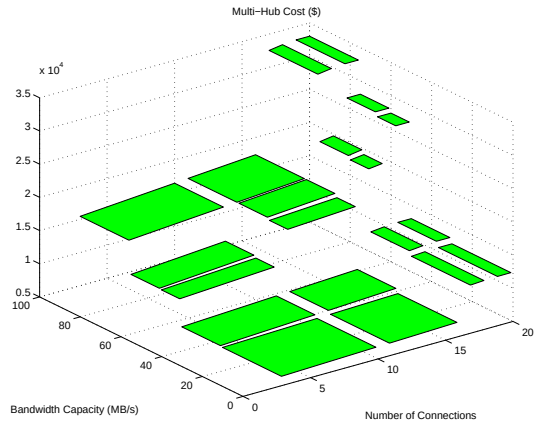
Each linear section of a component cost function has a corresponding section in the function domain. Consider the three different component types, and the corresponding number of properties:

- **links** have 1 component property (bandwidth capacity), so the sections are continuous line segments;
- **multi-hubs** have 2 component properties (number of connections, bandwidth capacity), so the sections are rectangles;
- **switches** also have 2 components (number of ports, port bandwidth capacity), so the sections are rectangles.

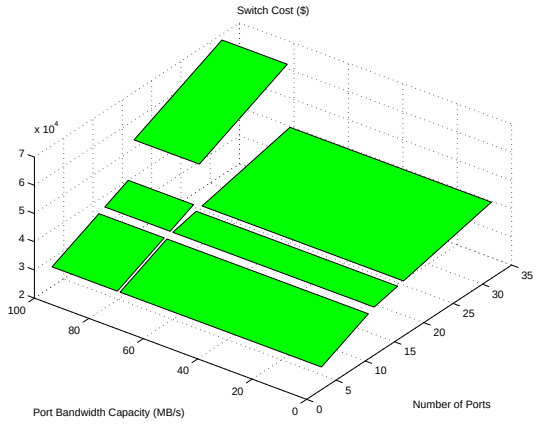
For our example, the component cost functions for links, multi-hubs and switches (corresponding to the respective technology tables – calculated in section 4) are shown in figure 3. Note that the cost of the multi-hubs *includes* the cost of all incident links. In the generic MIP formulation (see section 6) multi-hubs are modelled with all their incident links included, however in the Hewlett-Packard formulation (see section 3) only incident links within the multi-hub are included (links incident to hosts, devices or switches are modelled separately). The technology table from section 4 is valid for both formulation because a single link type is used for each bandwidth capacity range.



(a) Link Curve



(b) Multi-Hub Surface



(c) Switch Surface

Figure 3: Piecewise linear functions for links, multi-hubs and switches

Now that the cost functions are well-defined for the links, multi-hubs and switches, we can formulate variables and constraints to implement them. First, we define a binary variable for each *section* (line segment or rectangle) in the function domain. This is 1 if the component has properties corresponding to that section of the domain, or 0 otherwise. Of course, every component must either exist and fall into exactly one section of the cost function domain, or not exist at all (in which case every variable is 0).

For example, we can define

$$\begin{aligned} z_{li} &= 1 \text{ if link } l \text{ has the properties of section } i; \\ &= 0 \text{ otherwise} \end{aligned}$$

(see figure 4) which means

$$\sum_{i=1}^{4} z_{li} \leq 1 \tag{8}$$

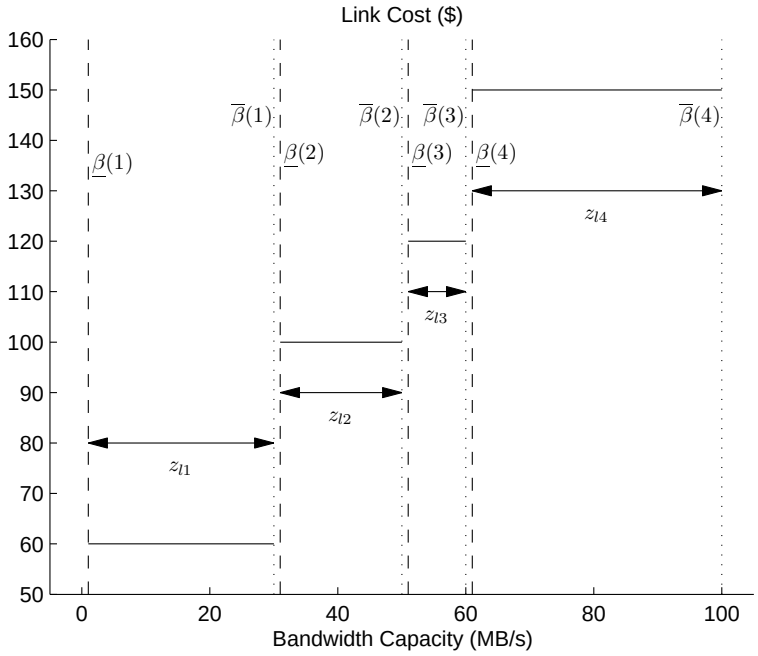


Figure 4: Defining variables for links

We also need to make sure that a component that is priced using a given section of the cost function domain has properties that lie within the restrictions of the properties of that section. The usual master-slave constraints for each of the component properties implement these restrictions. Returning to our example, we need to bound the bandwidth β_l of a link l depending on the section it belongs to (see figure 4)

$$\sum_{i=1}^{4} \underline{\beta}(i) z_{li} \leq \beta_l, \qquad \beta_l \leq \sum_{i=1}^{4} \overline{\beta}(i) z_{li} \quad \text{for each link } l \tag{9}$$

For switches, we need to bound both the number of ports η_s and the port bandwidth β_s of each switch s depending on the section it belongs to (see figure 5)

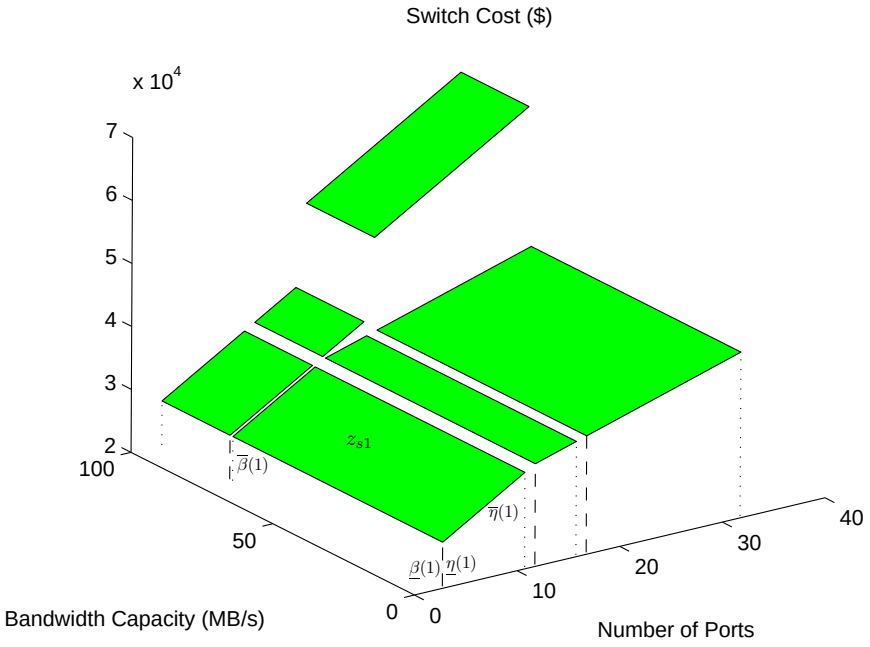


Figure 5: Defining variables for nodes (switches)

$$\begin{array}{ll} \sum_{i \in \text{ sections}} \underline{\eta}(i) z_{si} \leq \eta_s, & \eta_s \leq \sum_{i \in \text{ sections}} \overline{\eta}(i) z_{si} \\ \sum_{i \in \text{ sections}} \underline{\beta}(i) z_{si} \leq \beta_s, & \beta_s \leq \sum_{i \in \text{ sections}} \overline{\beta}(i) z_{si} \end{array} \quad \text{for each switch } s \tag{10}$$

We add these master-slave constraints to our new MIP formulation for the SANDP. This allows us to model generic components, the properties of which are determined in the solution process, rather than adding individual components (and deciding which components to include in the network).

One last consideration needs to be included in our MIP formulation, the contribution of the components to the objective function. We are given a fixed cost for each section of the component cost function domain. There may also be a variable cost for each of the different component properties. For SAN design, there is a fixed cost for each of the different link types used in each section, $\gamma(i)$ for section i, but no variable cost on the bandwidth capacity. For the hubs (switches) there is a fixed cost, no variable cost for the bandwidth (port bandwidth, respectively) capacity, but there is a variable cost on the number of connections (ports, respectively), $\pi(i)$ for section i. However, we are deciding how many connections (ports) the component has, so this could introduce a non-linearity in determining the cost of the connections (ports, respectively), i.e., $\pi(i) z_{mi} \eta_m$. We can remove this non-linearity by introducing new decision variables ν_{mi} (ν_{si}), representing the number

of connections (ports, respectively) of section i used by the multi-hub m (switch s, respectively). Two constraints make sure that these numbers agree with the actual number of connections and the section to which the multi-hub belongs.

$$\begin{array}{rl} \sum_{i\in \text{ sections}} \nu_{mi} & = \eta_m \\ \nu_{mi} & \leq \overline{\eta}(i) z_{mi}, i \in \text{ sections} \end{array} \quad \text{for each multi-hub } m \tag{11}$$

There are similar constraints for switches and the number of ports of each section.

The contribution to the objective function is now given by

$$\begin{array}{lll} \sum_{l\in \text{ links}, i\in \text{ link sections}} & \gamma(i) z_{li} & + \\ \sum_{m\in \text{ multi-hubs}, i\in \text{ multi-hub sections}} & (\gamma(i) z_{mi} & + \ \pi(i)\nu_{mi}) \\ \sum_{s\in \text{ switches}, i\in \text{ switch sections}} & (\gamma(i) z_{si} & + \ \pi(i)\nu_{si}) \end{array}$$

In section 6 we present our new MIP formulation that solves the SANDP using generic components. Rather than deciding what type each of these components is, we use the master-slave constraints described in this section to focus on the properties of the components. We also model the cost of these components using the variables and constraints discussed here.

6 The Generic MIP Formulation

In this section we describe the *generic* (MIP) formulation for solving the SANDP. There are two major concepts within this formulation:

- All components to the network are generic, that is their type is not already determined;
- Multi-hubs are treated as a configuration of hubs with **all** incident links included (so the multi-hub includes any links attached to hosts, devices or switches).

Both these modifications are included to reduce the size of the formulation. By allowing the properties of network components to change, we reduce the number of variables determining inclusion in the network at the expense of introducing variables to define the components. For a problem of the scale encountered in industry this results in a significant reduction in the size of the model. By including all incident links within a multi-hub component, we can also reduce the number of variables required to model the connections of the component. Other constraints ensure that the generic components are "well-behaved", that is, the properties of the component agree with their type, technological restictions are obeyed and flow through the network is conserved and not split. However, as shown later in this section the size of the MIP formulation is signifcantly reduced even for the (relatively) simple example considered in sections 3 and 4. In this section we present the generic formulation followed by a summary of its application to this example.

6.1 Component Definition Constraints

Remember we do not include specific components in our model, rather we include a number of generic components and determine their properties. We assume that there are a priori bounds on the number of components. Let

L be the set of possible (generic) links indexed by $1, 2, \ldots, |L|$,
M be the set of possible (generic) multi-hubs indexed by $1, 2, \ldots, |M|$,
S be the set of possible (generic) switches indexed by $1, 2, \ldots, |S|$,

and the number of sections (in the cost functions' domains) be denoted $\Gamma(L)$, $\Gamma(M)$ and $\Gamma(S)$ for links, multi-hubs and switches, respectively.

We define

$$\begin{array}{lll} u_c & = 1 \text{ if component } c \text{ exists in the network} & \text{for } c \in L \cup M \cup S \\ & = 0 \text{ otherwise} & \\ z_{ci} & = 1 \text{ if component } c \text{ belongs to section } i & \text{for } c \in L, \quad 1 \le i \le \Gamma(L) \\ & = 0 \text{ otherwise} & \qquad\quad c \in M, \quad 1 \le i \le \Gamma(M) \\ & & \qquad\quad c \in S, \quad 1 \le i \le \Gamma(S) \end{array}$$

and restrict a component to belong to exactly one section of the component properties' cost function domain if it exists in the network, cf. (8)

$$\sum_{i=1}^{\Gamma(L)} z_{li} = u_l, l \in L, \quad \sum_{i=1}^{\Gamma(M)} z_{mi} = u_m, m \in M, \quad \sum_{i=1}^{\Gamma(S)} z_{si} = u_s, s \in S, \tag{12}$$

Recall that links have one property (bandwidth capacity), multi-hubs have two properties (number of connections and bandwidth capacity) and switches have two properties (number of ports and port bandwidth capacity). We define variables for the component properties, cf. (9) and (10)

$$\begin{aligned} \beta_l &= \text{ the bandwidth of link } l \in L \\ \eta_m &= \text{ the number of connections on multi-hub } m \in M \\ \beta_m &= \text{ the bandwidth of multi-hub } m \in M \\ \eta_s &= \text{ the number of ports on switch } s \in S \\ \beta_s &= \text{ the port bandwidth of switch } s \in S \end{aligned}$$

that allow us to add in the master-slave constraints to restrict components to the

appropriate sections (of the cost function domain)

$$\begin{array}{lll}
\sum_{i=1}^{\Gamma(L)} \underline{\beta}(L,i) z_{li} \le \beta_l, & \beta_l \le \sum_{i=1}^{\Gamma(L)} \overline{\beta}(L,i) z_{li} & \text{for } l \in L \\
\sum_{i=1}^{\Gamma(M)} \underline{\eta}(M,i) z_{mi} \le \eta_m, & \eta_m \le \sum_{i=1}^{\Gamma(M)} \overline{\eta}(M,i) z_{mi} & \\
\sum_{i=1}^{\Gamma(M)} \underline{\beta}(M,i) z_{mi} \le \beta_m, & \beta_m \le \sum_{i=1}^{\Gamma(M)} \overline{\beta}(M,i) z_{mi} & \text{for } m \in M \\
\sum_{i=1}^{\Gamma(S)} \underline{\eta}(S,i) z_{si} \le \eta_s, & \eta_s \le \sum_{i=1}^{\Gamma(S)} \overline{\eta}(S,i) z_{si} & \\
\sum_{i=1}^{\Gamma(S)} \underline{\beta}(S,i) z_{si} \le \beta_s, & \beta_s \le \sum_{i=1}^{\Gamma(S)} \overline{\beta}(S,i) z_{si} & \text{for } s \in S
\end{array} \tag{13}$$

Note that we added an extra parameter for $\underline{\beta}, \overline{\beta}, \underline{\eta}$ and $\overline{\eta}$ to denote which type of component (link, multi-hub or switch) the bounds refer to.

We must also define

$$\begin{array}{rcll}
\nu_{mi} & = & \text{connections of section } i \text{ on multi-hub } m & \text{for} \quad m \in M, \quad 1 \le i \le \Gamma(M) \\
\nu_{si} & = & \text{ports of section } i \text{ on switch } s & \text{for} \quad s \in S, \quad 1 \le i \le \Gamma(S)
\end{array}$$

and use the constraints

$$\begin{array}{ll}
\nu_{mi} \le \overline{\beta}(M,i) z_{mi}, 1 \le i \le \Gamma(M) & \sum_{i=1}^{\Gamma(M)} \nu_{mi} = \eta_m \quad \text{for } m \in M \\
\nu_{si} \le \overline{\beta}(S,i) z_{si}, 1 \le i \le \Gamma(S) & \sum_{i=1}^{\Gamma(S)} \nu_{si} = \eta_s \quad \text{for } s \in S
\end{array} \tag{14}$$

to properly cost the generic multi-hubs and switches.

6.2 Component Ordering and Anti-symmetry Constraints

We can better control the network composition by defining

$$\begin{array}{rl}
n_L = & \text{the number of links used in the network,} \\
n_M = & \text{the number of multi-hubs used in the network,} \\
n_S = & \text{the number of switches used in the network,}
\end{array}$$

and limiting the number of components (links, multi-hubs and switches)

$$\sum_{l \in L} u_l = n_L, \quad \sum_{m \in M} u_m = n_M, \quad \sum_{s \in S} u_s = n_S \tag{15}$$

We can then remove most of the symmetry in the network design problem by ordering the components

$$\begin{array}{rl} u_l \geq u_{l+1}, & l \in \{1,2,\ldots,|L|-1\}, \\ u_m \geq u_{m+1}, & m \in \{1,2,\ldots,|M|-1\}, \\ u_s \geq u_{s+1}, & s \in \{1,2,\ldots,|S|-1\}, \end{array} \tag{16}$$

and enforcing anti-symmetry constraints on the component properties

$$\begin{array}{lll} & \beta_l \geq \beta_{l+1} & \text{for } l \in \{1,2,\ldots,|L|-1\} \\ \eta_m \geq \eta_{m+1} & \beta_m \geq \beta_{m+1} - \Omega(\eta_m - \eta_{m+1}) & \text{for } m \in \{1,2,\ldots,|M|-1\} \\ \eta_s \geq \eta_{s+1} & \beta_s \geq \beta_{s+1} - \Omega(\eta_s - \eta_{s+1}) & \text{for } s \in \{1,2,\ldots,|S|-1\} \end{array} \tag{17}$$

where Ω is a big-M quantity.

6.3 Defining Flow Paths

We define Φ as the maximum number of connections (links or multi-hubs) on the path for any flow $f \in F$ and n_f as the number of connections on the path for flow $f \in F$. Now we define which connections are being used and which links, hubs and/or switches make up a flow's path

$$\begin{array}{lll} w_{fj} & = 1 \text{ if } f \text{ goes along connection } j & \text{for } f \in F, 1 \leq j \leq \Phi \\ & = 0 \text{ otherwise} & \\ v_{fjc} & = 1 \text{ if } f \text{ goes along link/multi-hub } c \text{ on connection } j & \text{for } f \in F, 1 \leq j \leq \Phi, \\ & = 0 \text{ otherwise} & c \in L \cup M \\ v_{fjs} & = 1 \text{ if } f \text{ goes through switch } s \text{ after connection } j & \text{for } f \in F, 1 \leq j \leq \Phi - 1, \\ & = 0 \text{ otherwise} & s \in S \end{array}$$

and add constraints to control the number of connections on a path as well as ordering the connections

$$\sum_{j=1}^{\Phi} w_{fj} = n_f \qquad \text{for } f \in F \tag{18}$$

$$w_{fj} \geq w_{fj+1} \text{ for } f \in F, 1 \leq j \leq \Phi - 1 \tag{19}$$

and ensure that there is at most one link/multi-hub for each connection and at most one switch at the end of each connection (except the last one)

$$\begin{array}{rll} \sum_{c \in L \cup M} v_{fjc} & = w_{fj} & \text{for } f \in F, 1 \leq j \leq \Phi \\ \sum_{s \in S} v_{fjs} & = w_{fj+1} & \text{for } f \in F, 1 \leq j \leq \Phi - 1 \end{array} \tag{20}$$

Finally, the design will not be optimal if a flow visits a link, hub, or switch more than once, so we restrict this appropriately

$$\sum_{j=1}^{\Phi} v_{fjc} \leq u_c, f \in F, c \in L \cup M, \quad \sum_{j=1}^{\Phi-1} v_{fjs} \leq u_s, f \in F, s \in S \tag{21}$$

To ensure that a flow leaves its host we add the following constraint

$$\sum_{c \in L \cup M} v_{f1c} = 1, f \in F \tag{22}$$

If we want more than one path for a particular flow we can increase the right-hand side of constraint (22). Constraints (21) will then ensure that the paths are *disjoint*, ensuring *diverse protection* reliability.

6.4 Connecting the Network

Using the path information we can determine how the network is connected. We define

$$\begin{aligned} x_{ck} &= 1 \text{ if host/switch/device } c \text{ connects to link/multi-hub } k \\ &= 0 \text{ otherwise} \end{aligned}$$

and make sure that the first connection in a flow's path starts at the host and the last (active) connection ends at the device

$$\begin{array}{ll} x_{\theta(f)k} \geq v_{f1k} & \text{for} \quad f \in F, k \in L \cup M \\ x_{\delta(f)k} \geq v_{fjk} - w_{fj+1}, 1 \leq j \leq \Phi - 1 & \\ x_{\delta(f)k} \geq v_{f\Phi k} & \end{array} \tag{23}$$

Also, make sure that a switch on a path is connected to the links/hubs on the path

$$\begin{array}{ll} x_{sk} \geq v_{fjk} + v_{fjs} - 1 & \text{for} \quad s \in S, k \in L \cup M, f \in F, 1 \leq j \leq \Phi - 1 \\ x_{sk} \geq v_{fj+1k} + v_{fjs} - 1 & \end{array} \tag{24}$$

6.5 Technological Constraints

We need to ensure that the properties of the hosts, devices and components are not being exceeded. We define the number of ports η_c on host/device $c \in H \cup Ds$ and the degree constraints

$$\begin{array}{ll} \sum_{c \in H \cup S \cup D} x_{cl} = 2u_l, l \in L & \sum_{c \in H \cup S \cup D} x_{cm} = \eta_m, m \in M \\ \sum_{k \in L \cup M} x_{ck} = \eta_c \leq \eta(c), c \in H \cup D & \sum_{k \in L \cup M} x_{sk} = \eta_s, s \in S \end{array} \tag{25}$$

ensure that hosts, devices, links, multi-hubs and switches only have the number of ports available.

We also make sure that links and multi-hubs obey their bandwidth capacity restrictions

$$\sum_{f \in F, 1 \leq j \leq N^F} v_{fjk} \beta(f) \leq \beta_k, k \in L \cup M \tag{26}$$

and the port bandwidth capacity of the hosts, devices and/or switches connected to them

$$\begin{aligned} \beta_k &\leq \beta(c) + \Omega\left(1 - x_{ck}\right), c \in H \cup D, k \in L \cup M \\ \beta_k &\leq \beta_s + \Omega\left(1 - x_{sk}\right), s \in S, k \in L \cup M \end{aligned} \tag{27}$$

where Ω is another big-M quantity.

6.6 Example

If we apply our generic MIP formulation to the example from section 3, we can estimate upper bounds on the number of links, multi-hubs and switches. From section 3 we know there will be at most 2 switches and 9 multi-hubs. Since links only exist between host, device and switch ports there will be at most $\lfloor \frac{10+10+2\times32}{2} \rfloor = 42$ links. Therefore, the new formulation has 5433 variables and 18161 constraints. This represents a decrease of 93.1% in the number of variables and 63.8% in the number of constraints from the Hewlett-Packard MIP formulation with our preprocessing method. Unfortunately, the generic formulation is more difficult to solve in CPLEX 9.0.0 than the Hewlett-Packard formulation. After solving for almost 16 hours in CPLEX 9.0.0 (AMPL Version 20021031 running on a Compaq nx8220 with 512MB RAM), no feasible solution had been found.

By examining the structure of the two formulations along with the CPLEX output, we can deduce why the generic formulation is difficult to solve. The Hewlett-Packard formulation has 78199 variables and 50219 constraints (after preprocessing). This is a classical network design formulation where the binary variables represent the inclusion and use of preconfigured components in the SAN, the constraints ensure the components behave "properly". Our generic formulation has 5433 variables (5103 binary and 330 integer) and 18161 constraints. The variables define the components of the network and the constraints ensure the components not only behave properly, but also that they are "built" properly. The linear programming subproblems in the branch-and-bound tree are highly degenerate, possibly as a result of the structure of our generic formulation (more constraints than variables). We expect that solving the dual problem at each branch-and-bound node and using nested Wolfe iterations will signifcantly reduce solution time. Unfortunately, neither of these procedures are currently available in CPLEX 9.0.0.

7 Conclusions and Extensions

This paper presents two MIP formulations for the SANDP. The first formulation (from Hewlett-Packard Laboratories) makes available all possible components for building a SAN. MIP then decides which of these components to use in the design. As the input (number of hosts and devices, number and bandwidth requirements for flows, number of hub, switch and link types) grows, the number of possibilities increases exponentially. Even a small example problem requires hundreds of thousands

of constraints and variables. We summarise a preprocessing method that prunes the possibile network components to leave only those components that would appear in an optimal design. This preprocessing reduces the size of the problem by over 25% and allows CPLEX to solve in a reasonable time (again on a typical PC).

We also present a second formulation that uses generic components and allows MIP to determine the properties of those components. Our new formulation significantly reduces the size of the first formulation with preprocessing (by over 90% in the number of variables and 60% in the number of constraints). While the number of generic components increases as the input grows, the growth is manageable and this formulation may be used for much larger SANDPs. Unfortunately, the new formulation is highly degenerate, so solving using CPLEX is currently difficult. However, by solving the dual subproblem throughout the branch-and-bound tree and using nested Wolfe iterations we expect to solve the generic formulation for large commerical problems in reasonable time. CPLEX does not currently provide these features, so our future research will focus on implementing a branch-and-bound method that solves the dual subproblem at each node and uses nested Wolfe iterations to deal with degeneracy.

References

[1] A. Amiri, A system for the design of packet-switched communication networks with economic tradeoffs, Computer Communications 21 (1998) 1670–1680.

[2] A. Atamtürk, On capacitated network design cut-set polyhedra, Technical report, IEOR Department, University of California at Berkeley, 2000, available at http://ieor.berkeley.edu/ atamturk.

[3] F. Barahona, Network design using cut inequalities, SIAM Journal on Optimization (1996) 823–837.

[4] D. Bienstock, S. Chopra, O. Gunluk, Minimum cost capacity installation for multicommodity network flows, Mathematical Programming 81 (2–1) (1998) 177–199.

[5] D. Bienstock, O. Gunluk, Capacitated network design. Polyhedral structure and computation., INFORMS Journal on Computing 8 (1996) 243–259.

[6] R.B. Boorstyn, H. Frank, Large-scale network topological optimization .

[7] J. Campbell, A survey of hub location, Studies in Locational Analysis 6 (1994) 31–49.

[8] S. Chopra, I. Gilboa, S.T. Sastry, Source sink flows with capacity installation in batches, Discrete Applied Mathematics 85 (1998) 165–192.

[9] C.H. Chu, G. Premkumar, C. Chou, J. Sun, Dynamic degree constrained network design: a genetic algorithm approach, in: Proceedings GECCO–99. Genetic and Evolutionary Computation Conference. Eighth International Conference on Genetic Algorithms (ICGA–99) and the Fourth Annual Genetic Programming Conference (GP–99), volume 1, pp. 141–148.

[10] N. Deo, S. Hakimi, The shortest generalized Hamiltonian tree, in: Proceedings of the 6th Annual Allerton Conference, pp. 879–888.

[11] P. Fetterolf, G. Anandalingam, A Lagrangian relaxation technique for optimizing interconnection of local area networks, Operations Research 40 (1992) 678–688.

[12] M. Garey, D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness, W. H Freeman and Company, San Francisco, CA, 1979.

[13] B. Gavish, A general model for the topological design of computer networks, in: IEEE Global Telecommunications Conference, volume 3, pp. 1584–1588.

[14] B. Gavish, Topological design of computer communication networks – the overall design problem, European Journal of Operational Research 58 (1992) 149–172.

[15] J.L. Gross, J. Yellen (Eds.), Handbook of Graph Theory, Discrete Mathematics and its Applications, CRC Press, 2004.

[16] J. Herrmann, G. Ioannou, I. Minis, J. Proth, A dual ascent approach to the fixed-charge capacitated network design problem, European Journal of Operational Research 95 (1996) 476–90.

[17] J.M. Kleinburg, Single source unsplittable flow, in: Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science, pp. 68–77.

[18] J. Klincewicz, Hub location in backbone/tributary network design: a review, Location Science 6 (1998) 307–335.

[19] T.L. Magnanti, P. Mirchandani, R. Vachani, Modeling and solving the capacitated network loading problem, Operations Research 43 (1995) 142–157.

[20] P. Mirchandi, D. Simchi-Levi, Handbook of Graph Theory, CRC Press, chapter 11.4 Communication Network Design Models, Discrete Mathematics and its Applications, 2004, pp. 1117–1139.

[21] M. O'Sullivan, C. Walker, J. Allard, Improving optimal storage area network design, in: The Proceedings of the 38^{th} Annual Conference, ORSNZ'03, pp. 119–128.

[22] D. Skorin-Kapov, J. Skorin-Kapov, On hub location models, Journal of Computing and Information Technology 3 (1995) 183–192.

[23] V. Sridhar, J. Park, Benders-and-cut algorithm for fixed-charge capacitated network design problem, European Journal of Operational Research 125 (2000) 622–32.

[24] J. Ward, M. O'Sullivan, T. Shahoumian, J. Wilkes, Appia: automatic storage area network design, in: Conference on File and Storage Technology (FAST'02), pp. 203–217.