

Designing Low-Capacity Backup Networks for Fast Restoration

ABSTRACT

There are two basic approaches to allocate protection resources for fast restoration. The first allocates resources upon the arrival of each connection request; yet, it incurs significant set-up time and is often capacity-inefficient. The second approach allocates protection resources during the network configuration phase; therefore, it needs to accommodate *any* possible arrival pattern of connection requests, hence calling for substantial *over-provisioning* of resources. In this study we establish a novel protection approach that overcomes all the above drawbacks.

During the network configuration phase, we construct an (additional) low-capacity *backup network*. Upon a failure, traffic is rerouted through a bypass in the backup network. We establish that, with proper design, backup networks induce *minor* capacity overhead. We further impose several design requirements (e.g., hop-count limits) on backup networks and their induced bypasses, and prove that, commonly, they also incur minor overhead. Our approach offers additional benefits, most notably: traffic demands can be routed in an *unprotected* fashion, using standard routing schemes; moreover, upon a failure, control effort and congestion on the (primary) network are *small* and *localized* since affected traffic is immediately rerouted through the backup network. Motivated by these findings, we design efficient algorithms for the construction of backup networks.

Keywords

Fast-Restoration, Network Design, Graph-Theory.

1. INTRODUCTION

Transmission capabilities have increased to rates of 10 Gbit/s and beyond [10]. With this increase, any failure may lead to a vast amount of data loss. Accordingly, fast restoration has become a central requirement in the design of high-capacity networks e.g., optical mesh networks. It has been recognized that, for many practical settings, the speed and capacity of the involved links do not allow to activate restoration mechanisms *after* the failure. Thus, *protection resources* must be allocated in advance i.e., *before* a failure occurs [13].

There are two basic approaches to allocate protection resources. In the first approach, resources are allocated *on demand* i.e., upon the arrival of every bandwidth request, thus incurring a significant overhead in terms of connection set-up time. Consequently, this approach presents a clear tradeoff between the efficiency of the resulting solution in terms of capacity usage and the time needed to compute it; furthermore, its corresponding solutions are usually based on *only* partial (or no) information regarding the network state

and future connection requests. A different approach is to pre-allocate the protection resources *during the configuration phase* of the network. While this approach enables to perform computations offline, it requires allocating protection resources for *any* potential pattern of connection requests; hence, it usually calls for a substantial *over-provisioning* of protection resources.

In this study we introduce a novel protection approach, which overcomes all the above drawbacks and incurs a negligible toll of protection resources. The proposed approach is based on allocating *dedicated resources* to be used *exclusively* for handling failures. In essence, given a *primary network* that is used in normal operation mode to route demands (in an unprotected manner), we propose to establish a (low-capacity) *backup network* that can protect against *any single failure* experienced by the primary network; i.e., upon a failure of any primary link $e=(u,v)$, the traffic on e is rerouted from u to v through bypass paths that exclusively belong to the backup network. We formulate this notion as a network design problem with the objective of minimizing the total spare capacity of the backup network. The following example illustrates this idea.

Example 1: Considering Fig. 1, the solid lines represent the connectivity in a given (unprotected) primary network. Assume that the network is undirected and the capacity of all links (in each direction) is 1 except for the (bold) links (a,f) and (f,e) that have a capacity of 5. The dashed lines with indicated capacities represent a backup network. It is easy to verify that this backup network indeed provides protection against any single link failure in the primary network. For example, upon a failure of the (unit capacity) link (b,e), it is possible to reroute (exclusively over the backup network) one flow unit through the bypass path (b,c,d,e). Similarly, when link (a,f) that has a capacity 5 fails, it is possible to send one flow unit over the bypass path (a,e,f) and 4 flow units over the bypass path (backup link) (a,f). Note that bypass paths that protect

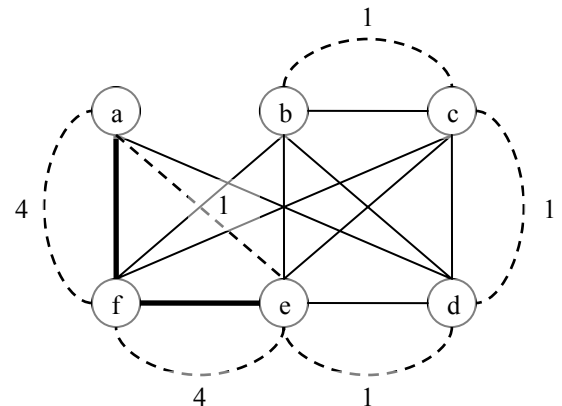


Fig. 1: A primary network and a corresponding backup network.

different links in this primary network can intersect each other and share the same amount of bandwidth to achieve efficiency. For example, the links (b,e) and (c,e) are protected by the bypasses (b,c,d,e) and (c,d,e) , respectively, with the shared links (c,d) and (d,e) .

Our analysis clearly indicates that the proposed approach, of establishing backup networks for primary networks, induces very small overhead in terms of extra capacity. We independently prove this claim for two different types of well-established models of real-world networks, namely Waxman networks [15] and Power-Law networks [4]. In essence, our results indicate that, in order to construct a backup network for an N -node primary network, it is sufficient to increase the total capacity allocated to the primary network by a factor of $O(\frac{1}{N})$ for Waxman networks and by a factor of $O(\frac{1}{\ln N})$ for Power-Law networks. Hence, the analysis for both (independent) models share the following (identical) conclusion: *at a minor price of extra capacity, it is possible to construct backup networks that fully protect against any single link failure.*

The above approach proposes other major advantages. First, the use of backup networks enables traffic demands to be routed in an *unprotected* fashion through any standard (and simple) routing scheme (e.g., shortest path algorithms); hence, their use can substantially simplify the routing process while still providing (transparent) protection to all connections being routed over the primary network. Moreover, upon a failure, the induced control effort is *small* and *localized*¹. In particular, unlike common restoration schemes that (upon a failure) divert all traffic to backup paths and thus may suddenly overload distant links on the (same) network [5], with our approach *no hot spots* are created on the *primary network*, since affected traffic is immediately rerouted to the backup network.² Finally, the relative long-time scale associated with the pre-placement of spare capacity into the network enables the employment of *offline* construction algorithms (with a complete knowledge of the network) that are capable of maximizing the capacity sharing among the bypass paths, thus *optimizing resource utilization*.

Motivated by these major benefits, we consider several (independent) design requirements that are important for the efficient deployment of backup networks. First, each link in the primary network should often be protected by a bypass path (in the backup network) with a *bounded hop-count*. Indeed, imposing a small hop-count limit on each bypass path is essential for supporting QoS requirements; for example, it has been noted [1] that the queuing delay in congested networks increases *exponentially* with the number of hops. Another example of the importance of such hop-

count limits is provided by optical networks, where the signal quality is deteriorated as it travels over multiple hops.

Another fundamental design requirement for backup networks considers the *number of bypass paths* that protect each primary link (i.e., among how many bypasses the rerouted traffic is split). It is often important to bound this number due to several reasons: first, splitting traffic over multiple bypass paths can cause packets to arrive out of order, thus increasing delivery latency and buffering requirements [14]; second, the complexity of a scheme that distributes traffic among multiple bypasses considerably increases with the number of paths [12]; third, often there is a limit on the number of explicit bypass paths (such as label-switched paths in MPLS) that can be set up between a pair of nodes [12].

Finally, we address a design requirement that considers the topology of the resulting backup network; specifically, this requirement restricts each backup topology to be a *subgraph* of the primary topology. With this requirement, the construction of the backup network is much easier for practical purposes. For example, when the backup network is a subgraph of the primary network, the duct systems that contain the communication cables of the primary network can be used to thread all communication cables of the backup network, thus avoiding an extensive and expensive digging process. In fact, with this requirement, it is possible to avoid hardware installation altogether. Indeed, when the backup network is a subgraph of the primary network, it is possible to allocate a fraction of the bandwidth of each primary link for protection purposes (i.e., for the backup network); hence, the backup network is defined over the (available) primary network infrastructure and no hardware installation is required.³

Each of these design requirements levies a toll in terms of the required backup capacity. Accordingly, we turn to consider the extra capacity that must be allocated to the backup network due to the imposition of each combination of the above design constraints. Specifically, we quantify the increase in extra capacity as follows. Given a set of (one or more) design constraints, what is the worst-case ratio between the *minimum capacity* that needs to be allocated to a backup network that *satisfies* this set of constraints and the *minimum capacity* that needs to be allocated to a backup network that *has no constraints to satisfy*? For all possible combinations of constraints we provide (rigorous) upper and lower bounds on this worst-case ratio. Table 1 summarizes the corresponding results for N -node networks.

The results summarized in Table 1 provide insights as well as important design rules for the efficient construction of backup networks. First, all combinations of design constraints save two (namely, the combinations that include

¹ I.e., portions of the network far away from the failed link do not need to be aware of the failure.

² Note that the mechanisms that switch the traffic from the primary network to the backup network are essentially simple and can be easily implemented (in hardware) by configuring all bypasses in advance.

³ Yet, it should be noted that defining the backup network over the primary network infrastructure may expose dependencies between certain failures on the primary network and failures on the backup network.

Hop-Count Limit ¹	Unsplittable Routing ²	Subgraph Constraint	Constraints
			Price
No	No	No	1 (by definition)
No	No + Yes	Yes	At least $2 \cdot (1 - \frac{1}{N})$; at most 2
No + Yes	Yes	No	Exactly $2 \cdot (1 - \frac{1}{N})$
Yes	No	No	At most $2 \cdot (1 - \frac{1}{N})$
Yes	No + Yes	Yes	$\Omega(N)$

Table 1: The increase in extra capacity due to each combination of design constraints (ratio with respect to the unconstrained case).

both the hop-count and the subgraph constraints) increase the extra capacity by a factor of at most 2; thus, their enforcement provides important performance benefits, and, at the same time, it incurs only a small cost in terms of extra capacity; in particular, although we have shown in our analysis that backup networks induce minor overhead only for the unconstrained case, this overhead remains small also when the corresponding combinations of design constraints are imposed.³ Yet, when the hop-count limits and the subgraph constraints are *concurrently* imposed, the extra capacity is (dramatically) increased by a factor of $\Omega(N)$; hence, since such an increase is usually prohibitive, only one out of the two design constraints should be considered. Also, it is interesting to note that the cost incurred by imposing the requirement to support unsplittable routing at the backup network is *equal* to that incurred by satisfying that requirement *and* the requirement for small hop-counts; hence, when the requirement to support unsplittable routing is imposed, hop-count limits can be also imposed at no cost.

For the above design constraints, we design several polynomial running time algorithms that aim at minimizing the capacity allocated for the backup networks while satisfying a given set of constraints. Specifically, we design two types of algorithms. The first imposes the requirement to support unsplittable routing at the backup network while the other allows traffic to be split among several bypasses (i.e., the unsplittable routing requirement is not imposed). For the splittable case, we present a *polynomial* running time algorithm that *optimally* solves the problem while considering either one or both of the other design constraints (namely, the hop-count limits and the subgraph constraint). For the unsplittable case, we present two algorithms that approximate the optimal solution by a factor of at most 2. The first approximation is designed to meet the subgraph constraint while running in a time complexity of $O(NM)$ for

M -link N -node networks, and the other is designed to meet the hop-count limit while running in a (linear) time complexity of $O(N)$. Finally, we show how to modify some of the proposed schemes in order to construct backup networks that protect against *correlated failures*.

The rest of this paper is organized as follows. In Section 2, we introduce some terminology and formulate the model. In Section 3, we establish important design rules for backup networks and investigate the increase in allocated capacity induced by the requirement to satisfy the above design constraints. In Section 4, we show how to construct a backup network for any given primary network while satisfying a given set of constraints; moreover, we extend the single-link failure assumption and consider the construction of backup networks that provide protection against multiple (correlated) failures. In Section 5, we consider two well-known models of real-world networks and analytically show that backup networks induce very small overhead in terms of extra capacity. Finally, Section 6 summarizes the results and discusses future research directions.

2. MODEL

We are given a primary network $G(V, E)$ with a capacity c_e for each link $e \in E$. Let $N = |V|$ and $M = |E|$. As commonly done in studies on survivability and on optical networks, we assume that the network is *undirected*.⁴ The goal is to construct a backup network $G^b(V, E^b)$ that protects the traffic carried by each link in $G(V, E)$. To that end, we have to find a set of links $E^b \subseteq V \times V$ and backup capacities $\{c_e^b\}$ for these

links so that $G^b(V, E^b)$ can be used to reroute the traffic of any primary link $e \in E$ once it fails. As explained, we consider three types of restrictions on the backup networks. The first is a hop count restriction h imposed on each bypass path in the backup network. The second restriction is on the topology of the backup network; specifically, this restriction limits the backup network to be a subgraph of the primary network. The third restriction is imposed on the number of bypass paths that protects each primary link. Referring to the third constraint, we formulate the following types of backup networks. *Unsplittable backup networks* have a bypass path p between the end-nodes of each (primary) link $e \in E$ such that p can carry all the traffic of e once it fails i.e., the capacity of p (denoted by $c(p)$) is at least c_e . *Splittable backup networks* have a collection of bypass paths $P(e)$ between the end-nodes of each (primary) link $e \in E$ such that the total capacity of all protection paths in $P(e)$ is at least the capacity of e i.e., $c_e \leq \sum_{p \in P(e)} c(p)$. Finally, among all feasible backup networks that satisfy the above restrictions, we wish to construct one with minimum total capacity i.e., $\sum_{e \in E^b} c_e$ is minimized.

¹ The hop count limit is set to (the tight value) 2 i.e., all bypass paths in the backup network must consist of at most 2 links.

² The number of bypass paths is restricted to (the tight value) 1, i.e., upon a failure on any primary link $e = (u, v)$, the traffic is rerouted from u to v over a *single* bypass path.

³ Indeed, each combination involve only $O(1)$ increase in extra capacity.

⁴ In optical networks, adjacent nodes are usually connected by a pair of (identical) fibers carrying information in opposite directions [8]. Therefore, undirected graphs efficiently model real optical networks and thus have been the focus of many studies on survivability, e.g., [3], [16], [9], [7].

In the basic version of the problem, we focus on the single link failure model i.e., at any given time there exists at most one failed link. Note that this assumption enables to construct backup networks that are significantly more efficient in terms of capacity, since bypass paths in the backup network for two *different* (primary) links can intersect each other and *share* the same amount of capacity. In Section 4, we extend the single link failure assumption and consider backup networks that protect against *correlated failures*.

3. DESIGN RULES FOR BACKUP NETWORKS

In this section we investigate the increase in the total capacity of backup networks when the design constraints mentioned in the Introduction are imposed. This investigation enables to understand some fundamental tradeoffs in backup networks, and it provides important design rules for their efficient construction. To that end, we quantify the increase in the total capacity allocated for the backup network due to each *combination* of the following design constraints: (i) imposing a hop-count limit of 2 on each bypass path; (ii) supporting unsplittable routing at the backup network (i.e., upon any failure of a primary link, there is at least one bypass path that can reroute all affected traffic); (iii) restricting the topology of the backup network to be a subgraph of the topology of the primary network. For convenience, we represent any combination of constraints as a triplet of binary indicators (H, U, S) with roles and possible values as follows.

- Indicator H takes the value H^+ if the 2-hop count limit is imposed, and the value H^- otherwise.
- Indicator U takes the value U^+ if it is required to support unsplittable routing, and the value U^- otherwise.
- Indicator S takes the value S^+ if the subgraph constraint is imposed, and the value S^- otherwise.

Then, for any combination of constraints (H, U, S) , we denote by $\rho(H, U, S)$ the worst-case ratio between the *minimum capacity* allocated for a backup network that *satisfies the combination* (H, U, S) and the *minimum capacity* allocated for a backup network that *has no imposed restrictions* (i.e., (H^-, U^-, S^-)).

We outline the organization and the main results obtained in this section. In Subsection 3.1, we establish a lower bound on the minimum capacity of *any* backup network; this lower bound is essential for the evaluation of $\rho(H, U, S)$ for each combination (H, U, S) . In Subsection 3.2, we prove that protection capacity is increased (in the worst case) by a factor of *at most* $2(1 - \frac{1}{N})$ when hop count limits are imposed and by a factor of *exactly* $2(1 - \frac{1}{N})$ when unsplittable routing must be supported (i.e., $\rho(H^+, U^-, S^-) \leq 2 \cdot (1 - \frac{1}{N})$ and $\rho(H^-, U^+, S^-) = 2 \cdot (1 - \frac{1}{N})$). Somewhat surprisingly, we establish that when both constraints are concurrently imposed (i.e., both H and U are set) the protection capacity is still increased by a factor of exactly

$2(1 - \frac{1}{N})$; hence, in unsplittable backup networks, the hop-count limits can be imposed at no price. In Subsection 3.3, we show that restricting the backup network to be a subgraph of the primary network increases the protection capacity by a factor of at most 2 and at least $2(1 - \frac{1}{N})$, both for the splittable and unsplittable cases. Finally, in Subsection 3.4, we consider the increase in the protection capacity when the backup network must concurrently satisfy both the hop count limit and the subgraph constraint; although each of them increases the protection capacity only by a (small) constant factor, we show that when both constraints must be satisfied (*concurrently*), the protection capacity dramatically increases by a factor of $\Omega(N)$.

3.1 A Tight Lower Bound on Minimum Capacity

In this subsection, we establish a lower bound on the minimum capacity of unrestricted backup networks i.e., the case where (H^-, U^-, S^-) . This lower bound is used in Subsections 3.2 and 3.3 to bound from above the ratio $\rho(H, U, S)$. In addition, we provide a simple example that demonstrates that this lower bound is tight.

Lemma 1: Given a primary network $G(V, E)$, denote for each $v \in V$ the maximum capacity of a link that is incident on v by $C(v)$, i.e., $C(v) = \max_{(v, u) \in E} \{c(v, u)\}$ where $c(v, u)$ is the capacity

of the link (v, u) . Then, $\frac{1}{2} \cdot \sum_{v \in V} C(v)$ is a lower bound on the minimum capacity of any unrestricted¹ backup network.

Proof: Consider any unrestricted backup network $G^b(V, E^b)$ for the primary network $G(V, E)$. Let $E^b(v)$ be the collection of all links that are incident on v in $G^b(V, E^b)$. Note that, for each $v \in V$, the total protection capacity of all links incident on v in $G^b(V, E^b)$ must be at least $C(v)$ i.e., $\sum_{e \in E^b(v)} c_e \geq C(v)$

for each $v \in V$. Indeed, otherwise the link with maximum capacity in $G(V, E)$ that is incident on a node $v \in V$ is not protected. Thus, we conclude

$$\sum_{v \in V} \sum_{e \in E^b(v)} c_e \geq \sum_{v \in V} C(v). \quad (1)$$

Since the network $G^b(V, E^b)$ is undirected, each link $e = (v, u)$ in E^b belongs to both $E^b(v)$ and $E^b(u)$; hence, $2 \cdot \sum_{e \in E^b} c_e = \sum_{v \in V} \sum_{e \in E^b(v)} c_e$. This, together with (1), proves that

$\sum_{e \in E^b} c_e \geq \frac{1}{2} \cdot \sum_{v \in V} C(v)$ i.e., the minimum capacity of every unrestricted backup network is at least $\frac{1}{2} \cdot \sum_{v \in V} C(v)$. ■

As the following simple example shows, the above lower bound is tight. Consider a network that consists of a pair of nodes u, v that are connected by a single link e . Obviously, for this case, the optimal backup network constitutes of a single parallel link to e with a capacity of c_e ; since by

¹ I.e., the case (H^-, U^-, S^-) .

definition $C(u)=C(v)=c_e$, it follows that $\frac{1}{2} \cdot \sum_{v \in V'} C(v) = c_e$; hence, the total capacity of this (optimal) backup network is exactly $\frac{1}{2} \cdot \sum_{v \in V'} C(v)$.

3.2 The Price of Hop-Count Restrictions and Unsplittable Routing is Small

In this subsection we investigate the increase in the protection capacity when the backup network must support unsplittable routing and/or satisfy hop count restrictions on each bypass path. Obviously, imposing each of the two design requirements substantially reduces the set of feasible solutions. Therefore, one could expect that the enforcement of any of these constraints results in a severe increase in the capacity needed for protection. Yet, in the following, we prove that designing backup networks that are restricted to reroute traffic unsplittably over bypass paths with a hop limit $h=2$ never increases the protection capacity by a factor of more than $2(1-\frac{1}{N})$. Specifically, we show that each of the ratios $\rho(H^+, U^+, S^-)$ and $\rho(H^+, U^+, S^-)$ is equal to $2(1-\frac{1}{N})$. For the case where only the hop count limit is imposed, we show that $\rho(H^+, U^+, S^-) \leq 2(1-\frac{1}{N})$. The analysis for all these ratios is based on the following network construction.

QoS-Backup Network <Primary Network $G(V, E)$ >

1. Initialize the set of links in the backup network to be empty i.e., $E^b \leftarrow \emptyset$.
2. Let s be a node in V incident to a link with maximum capacity i.e., $C(s) = \text{Max}_{v \in V'} \{C(v)\}$.
3. For each node $u \in V' \setminus \{s\}$, add to the set E^b a link between nodes s and u with a capacity $C(u)$.
4. Return the network $G^b(V, E^b)$.

Fig. 2: Procedure QoS-Backup Network

Theorem 1: The minimum protection capacity of an *unsplittable* backup network with a *hop limit* $h=2$ is larger by a factor of at most $2 \cdot (1-\frac{1}{N})$ than the minimum protection capacity of a *splittable* backup network that has *no hop count limits* i.e., $\rho(H^+, U^+, S^-) \leq 2 \cdot (1-\frac{1}{N})$.

Proof: Given a primary network $G(V, E)$, in order to prove the theorem it is sufficient to show that Procedure QoS-Backup Network constructs an unsplittable backup network that consists of solely 2-hop protection paths and has a total capacity of at most $(1-\frac{1}{N}) \cdot \sum_{v \in V'} C(v)$; this, together with Lemma 1 (that establishes a lower bound of $\frac{1}{2} \cdot \sum_{v \in V'} C(v)$ on the minimum capacity of any splittable backup network), proves the theorem. Let $G^b(V, E^b)$ be the backup network returned by Procedure QoS-Backup Network when applied on the primary network $G(V, E)$. First, it is easy to see that,

by construction, each link $(u, v) \in E$ has a 2-hop bypass path (u, s, v) in $G^b(V, E^b)$; hence, each primary link in $G(V, E)$ is protected by a 2-hop path in $G^b(V, E^b)$. Next, note that the total capacity of the links in $G^b(V, E^b)$ is $\sum_{u \in V' \setminus \{s\}} C(u) = \sum_{v \in V'} C(v) - C(s)$; since $C(s) = \max_{v \in V'} \{C(v)\} \geq \frac{1}{N} \cdot \sum_{v \in V'} C(v)$, it holds that the total capacity of $G^b(V, E^b)$ is $\sum_{u \in V' \setminus \{s\}} C(u) = \sum_{v \in V'} C(v) - C(s) \leq \sum_{v \in V'} C(v) - \frac{1}{N} \cdot \sum_{v \in V'} C(v) = (1-\frac{1}{N}) \cdot \sum_{v \in V'} C(v)$; thus, according to Lemma 1, the capacity of $G^b(V, E^b)$ is larger by a factor of at most $2 \cdot (1-\frac{1}{N})$ than the minimum possible capacity for splittable backup networks. It is left to be shown that each link $e \in E$ has a protection path p in $G^b(V, E^b)$ that can carry all the traffic of e i.e., $c_e \leq c(p)$, thus establishing that $G^b(V, E^b)$ is an unsplittable backup network for $G(V, E)$. To that end, note that, for each link $(u, v) \in E$, the bypass path (u, s, v) in $G^b(V, E^b)$ has a capacity of $\min\{C(u), C(v)\}$; indeed, by construction, the capacity of $(u, s) \in E^b$ and $(s, v) \in E^b$ is $C(u)$ and $C(v)$, respectively. Therefore, since by definition of $C(\cdot)$, the value of both $C(u)$ and $C(v)$ is at least the capacity value of the link $(u, v) \in E$, it follows that the capacity $\min\{C(u), C(v)\}$ (of the bypass path (u, s, v)) is at least the capacity of the link (u, v) . ■

As mentioned in the Introduction, splitting the rerouted traffic over a small number of bypass paths, each with a bounded hop-count, is essential for supporting QoS-sensitive applications. Note that, in the worst case, the rerouted traffic can be split among $\Omega(M)$ bypass paths, each with $\Omega(N)$ links. For such cases, Theorem 1 suggests a useful design rule that enables to trade the amount of capacity needed for protection with the quality of the rerouted traffic. Note that this tradeoff is very effective, as it involves $O(M)$ improvement in the split ratio and $O(N)$ improvement in the hop-count of the bypass paths, at a price of just $O(1)$ increase in the (minimum) protection capacity.

We proceed to present other important insights and design rules for the efficient construction of unsplittable backup networks with hop count limits. To that end, we first establish the following corollary that stems directly from the upper bound on $\rho(H^+, U^+, S^-)$ (Theorem 1) and the obvious relations $\rho(H^+, U^+, S^-) \leq \rho(H^+, U^+, S^-)$, $\rho(H^+, U^+, S^-) \leq \rho(H^+, U^+, S^-)$.

Corollary 1: The ratios $\rho(H^+, U^+, S^-)$, $\rho(H^+, U^+, S^-)$ and $\rho(H^+, U^+, S^-)$ are all bounded from above by $2 \cdot (1-\frac{1}{N})$.

We now show that the upper bound obtained in Corollary 1 for the ratios $\rho(H^+, U^+, S^-)$ and $\rho(H^+, U^+, S^-)$ is tight. To that end, it is sufficient to establish a lower bound of $2 \cdot (1-\frac{1}{N})$ for these ratios. We first focus on establishing a lower bound for the cost incurred by the requirement to reroute traffic unsplittably i.e., a lower bound for $\rho(H^+, U^+, S^-)$. More specifically, we present an example of optimal unsplittable and splittable backup networks that share the same primary network and differ from each other in the allocated capacity

by a factor of $2 \cdot (1 - \frac{1}{N})$. The example is illustrated through Fig. 3. The solid lines represent a primary network (with a topology of an N -node ring), and the dashed lines represent an unsplittable backup network in (a) and a splittable backup network in (b). We assume that the capacities of all links in the primary network are equal to 1. Moreover, it is assumed that each link has a capacity of 1 in the backup network of (a) and a capacity of $\frac{1}{2}$ in the backup network of (b).

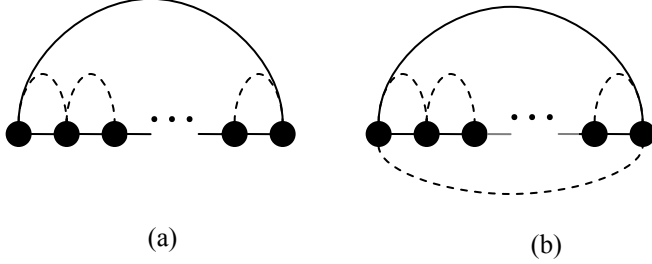


Fig. 3: The price of unsplittable routing is exactly $2 \cdot (1 - \frac{1}{N})$.
(a) Optimal unsplittable backup network with a capacity of $N-1$.
(b) Optimal splittable backup network with a capacity of $\frac{N}{2}$.

First note that any *unsplittable* backup network must consist, for each node $v \in V$, of at least one link with a unit capacity that connects v to some other node in $V \setminus \{v\}$. Indeed, if a node v is connected to all its neighbors in the backup network only by links with a capacity smaller than 1, it cannot reroute the traffic unsplittably upon any failure on the links that are incident to v in the *primary* network. This, together with the fact that every backup network for the given primary network must consist of at least $N-1$ links¹, establishes that the total capacity of any unsplittable backup network must be of at least $N-1$. In particular, the unsplittable backup network presented in (a) is optimal. Consider now the splittable backup network presented in (b). Note that this backup network is a *ring*. Therefore, upon a failure of any primary link, it is possible to split the traffic evenly and reroute each half in an opposite direction. Thus, since the capacity values of all links are equal to $\frac{1}{2}$ in the backup network and equal to 1 in the primary network, the backup network is feasible as it protects against any failure on the primary network. Note that the total capacity allocated for this backup network is $\frac{N}{2}$; hence, the total capacity of any optimal splittable backup network (for the given primary network) is at most $\frac{N}{2}$. Thus, it holds that $\rho(H^-, U^+, S^-) \geq \frac{N-1}{\frac{N}{2}} = 2 \cdot (1 - \frac{1}{N})$.

Finally, note that, since $\rho(H^+, U^+, S^+) \geq \rho(H^-, U^+, S^-)$, the lower bound obtained for $\rho(H^-, U^+, S^-)$ also applies for $\rho(H^+, U^+, S^+)$ i.e., $\rho(H^+, U^+, S^+) \geq 2 \cdot (1 - \frac{1}{N})$. Note that for both ratios, this lower bound coincides with the upper bound established in Corollary 1. Hence, the price of unsplittable routing is *exactly* $2 \cdot (1 - \frac{1}{N})$ and is not affected by imposing (additional)

¹ Indeed, otherwise the backup network is not connected; and, clearly, if the primary network has a connected topology then the backup network must also be connected.

hop-count limits on the bypass paths i.e., $\rho(H^-, U^+, S^-) = \rho(H^+, U^+, S^+) = 2 \cdot (1 - \frac{1}{N})$. We summarize this discussion with the following corollary.

Corollary 2: Each of the ratios $\rho(H^-, U^+, S^-)$ and $\rho(H^+, U^+, S^+)$ is exactly $2 \cdot (1 - \frac{1}{N})$.

3.3 The Subgraph Constraint has a Small Price

As mentioned in the Introduction, from a practical viewpoint, it may be much easier to construct a backup network with a topology that is a subgraph of the primary network. Accordingly, in this subsection we investigate the increase in protection capacity when the topology of the backup network is required to be a subgraph of the topology of the primary network. Specifically, we show that, due to this "subgraph constraint", the total capacity allocated for the backup network increases (in the worst case) by a factor that is between $2 \cdot (1 - \frac{1}{N})$ and 2 for both splittable and unsplittable backup networks i.e., $\rho(H^-, U^-, S^+), \rho(H^-, U^+, S^+) \in [2 \cdot (1 - \frac{1}{N}), 2]$.

The proof of the upper bound (Theorem 2) is based on the network construction specified in Fig. 4. More specifically, given a primary network $G(V, E)$, we prove that the following procedure constructs a backup network that is a subgraph of $G(V, E)$ and has a capacity of at most twice that of an optimal backup network (with arbitrary topology and no hop-counts).

Subgraph Backup Network <Primary Network $G(V, E)$ >

1. Initialization

- Let e_1, e_2, \dots, e_M denote a non-decreasing ordering of the links of E according to their capacities.
- Initialize the set of links in the backup network to be empty i.e., $E^b \leftarrow \emptyset$.
- Define an index i and initialize it to zero i.e., $i \leftarrow 1$.

2. While $i \leq M$

- Consider the link $e_i = (u_i, v_i)$ in the primary network $G(V, E)$.
- If e_i does not create a cycle with the (current) links of E^b , add to E^b a link between nodes u_i and v_i with capacity c_e (i.e., the same capacity as in the primary network).
- $i \leftarrow i + 1$

3. Return the network $G^b(V, E^b)$.

Fig. 4: Procedure Subgraph Backup Network

Theorem 2: Restricting the backup network to be a subgraph of the primary network increases the minimum protection capacity by a factor of at most 2 both for the splittable and for the unsplittable case i.e., $\rho(H^-, U^-, S^+) \leq 2$ and $\rho(H^-, U^+, S^+) \leq 2$.

Proof: Given a primary network $G(V, E)$, in order to prove the theorem it is sufficient to show the existence of an unsplittable backup network $G^b(V, E^b)$ that is a subgraph of $G(V, E)$ and has a total capacity of at most $\sum_{v \in V} C(v)$. Indeed, by Lemma 1, every splittable backup network has a lower bound of $\frac{1}{2} \cdot \sum_{v \in V} C(v)$ on the minimum protection capacity; hence, since the minimum protection capacity in the splittable case is not larger than in the unsplittable case, the existence of such a backup network (i.e., an unsplittable backup network that satisfies the subgraph requirement and has a total capacity of at most $\sum_{v \in V} C(v)$) proves the theorem both for the splittable and unsplittable cases. Accordingly, in the following we show that the network $G^b(V, E^b)$, returned by Procedure Subgraph Backup Network (Fig. 4), when applied on the primary network $G(V, E)$, satisfies all the following: (i) $G^b(V, E^b)$ is an unsplittable backup network for $G(V, E)$; (ii) $G^b(V, E^b)$ is a subgraph of $G(V, E)$; (iii) $G^b(V, E^b)$ has a total capacity of at most $\sum_{v \in V} C(v)$. Hence, proving (i), (ii) and (iii) establishes the theorem.

We first show that $G^b(V, E^b)$ is an unsplittable backup network for the primary network $G(V, E)$. To that end, we have to show (by definition) that each link $e \in E$ has a protection path p in $G^b(V, E^b)$ that can carry all the traffic of e , i.e., $c_e \leq c(p)$. Note that when a link $e \in E$ between nodes u and v is considered in the construction of $G^b(V, E^b)$ (as per Fig. 4), either a parallel link to e with a capacity c_e is added to the backup network $G^b(V, E^b)$ or there already is a path in $G^b(V, E^b)$ between u and v . Obviously, in the first case, the parallel link that was added to $G^b(V, E^b)$ has a capacity c_e that can support the traffic of the link e . In the second case, link e was considered after a path p from u to v was formed in $G^b(V, E^b)$; hence, since the links are considered by decreasing capacity values, the capacity of each link along p in $G^b(V, E^b)$ is not smaller than c_e ; therefore, the path p can support the traffic carried by e .

We turn to show that the total capacity of $G^b(V, E^b)$ is at most $\sum_{v \in V} C(v)$. To that end, we show the existence of a one-to-one mapping $f: E^b \rightarrow V$ in the network $G^b(V, E^b)$. Specifically, we map each link $(u, v) \in E^b$ into one of its end-nodes (i.e., either u or v) such that no two links in E^b are mapped into the same node in V . Since links are added to the backup network with capacity values identical to those in the primary network, and since $C(v)$ is the maximum capacity of a link incident on node v , such a mapping enables to allocate for each link $e \in E^b$ an exclusive node $v \in V$ such that $c_e \leq C(v)$; obviously, this proves the theorem, since it implies that $\sum_{e \in E^b} c_e \leq \sum_{v \in V} C(v)$.

We turn to specify the mapping. We partition the set of nodes V into two disjoint sets S and T such that $S \cup T = V$. Initially, we pick an arbitrary node $s \in V$ and assign $S = \{s\}$ and $T = V \setminus \{s\}$. At each step we choose a link $e = (u, v)$ in E^b , which crosses the cut (S, T) (i.e., $u \in S$, $v \in T$ and $e \in E^b$) and map (u, v)

into the node $v \in T$; then, we move the node v from the set T to the set S . The process is repeated until there is no link in the cut (S, T) . Obviously, it follows by construction that at the end of the process each link (u, v) that was considered by the process is mapped into one of its end nodes (either u or v). Moreover, once a link (u, v) is mapped to v , the process removes v from the set T ; hence, since the process maps links *only* to nodes in T , each node is always associated with at most one link i.e., there is no pair of links that are mapped to the same node. Therefore, it is left to be shown that at the end of the process *all* the links in E^b are mapped to some node in V . To that end, it is sufficient to show that each link is considered by the process at least once.

To that end, we first show that $G^b(V, E^b)$ is connected. Recall that we assume that $G(V, E)$ is connected. Hence, for each pair of nodes $u, v \in V$, there exists a path in $G(V, E)$ that connects u and v ; hence, since we have shown that $G^b(V, E^b)$ has a path that connects the end nodes of each link in E ,¹ there is also a path between every pair of nodes $u, v \in V$ in $G^b(V, E^b)$; hence, $G^b(V, E^b)$ is connected. Now assume by way of contradiction that there exists a link in E^b that has not been considered by the process and denote by $A \subseteq E^b$ the set of all such links. Since $G^b(V, E^b)$ is connected, it follows that, for each link $(u, v) \in A$, either u or v (or both) must be connected to at least one node in $V \setminus \{u, v\}$ by a link from E^b . Among the links in A , choose a link $e = (u, v)$ that is closest in terms of hop count to a link that has already been considered by the process; by the selection of (u, v) there must exist a node $w \in V \setminus \{u, v\}$ such that either the link (v, w) or the link (u, w) belongs to E^b and it has been considered by the process. Without loss of generality, assume that link (u, w) has been considered by the process. Therefore, by construction, when the process ends, node u is in the subset S . On the other hand, since by construction $G^b(V, E^b)$ is acyclic, it is impossible that a link incident on the node v has been considered by the process, since this implies that there exists in $G^b(V, E^b)$ a path between s and v and also a path between s and u that, together with the link (u, v) , form a cycle in $G^b(V, E^b)$.² Hence, node v is in T and the link (u, v) crosses the cut (S, T) ; thus, by construction, it should have been considered by the process. Obviously, this contradicts the selection of the link (u, v) . We thus conclude that all the links in E^b are considered by the process at least once.

It remains to be shown that the topology of $G^b(V, E^b)$ is a subgraph of $G(V, E)$. By construction, Procedure Subgraph Backup Network defines a link between a pair of nodes in $G^b(V, E^b)$ *only* if the pair of nodes is connected by a link in

¹ Specifically, we have shown at the beginning of the proof that $G^b(V, E^b)$ is an unsplittable backup network for $G(V, E)$. Hence, it must have at least one path that connects the end nodes of each link in E .

² By construction, the process considers only links from E^b ; moreover, it is easy to see that all considered links define a connected component. Therefore, there must exist in $G^b(V, E^b)$ a path between s and v and also a path between s and u if the process has considered a link incident on the node v and a link incident on the node u .

the primary network $G(V, E)$; hence, $G^b(V, E^b)$ is a subgraph of $G(V, E)$, thus completing the proof. ■

The property established in theorem 2 should be considered in cases where building a completely independent infrastructure for the backup network is too costly (or impossible). Specifically, in such cases, Theorem 2 suggests an important design rule that makes the construction of the backup network substantially easier at the price of increasing the total protection capacity by a factor of at most 2. We now turn to show that this result is almost tight for both the splittable and the unsplittable cases. Specifically, we show that each of the ratios $\rho(H^-, U^-, S^+)$ and $\rho(H^+, U^+, S^+)$ is larger than $2 \cdot (1 - \frac{1}{N})$. To that end, we present an example of a primary network with two corresponding backup networks that due to the subgraph constraint differ from each other in the allocated capacity by a factor of $2 \cdot (1 - \frac{1}{N})$. The example is illustrated in Fig. 5. The solid lines (in (a) and (b)) represent a primary network with a topology of an N -node path graph, the dashed lines in (a) represent a backup network that satisfies the subgraph constraint, and the dashed lines in (b) represent a backup network with an arbitrary topology. We assume that the capacities of all links in the primary network are equal to 1. Moreover it is assumed that each link has a capacity of 1 in the backup network of (a) and a capacity of $\frac{1}{2}$ in the backup network of (b).

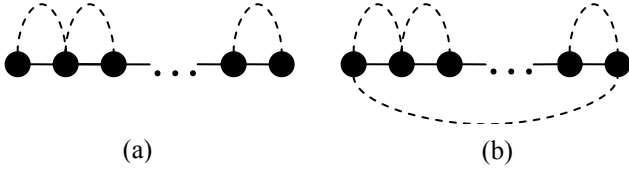


Fig. 5: The subgraph constraint has a price of at least $2 \cdot (1 - \frac{1}{N})$.
(a) Optimal backup network satisfying the subgraph constraint.
(b) A backup network with arbitrary topology.

We first bound from below the ratio $\rho(H^-, U^-, S^+)$. To that end, note that when the subgraph constraint is imposed, the backup and the primary networks must have identical topologies (i.e., the backup network must consist of a parallel link to each primary link).¹ Therefore, the topology of the backup network is a path graph and there is only one path between each pair of nodes. Thus, upon a failure of any primary link e , the rerouted traffic cannot be split and is carried unsplittably over the link that is parallel to e in the backup network. Therefore, since the capacity of each primary link is 1, the capacity of the backup links must be of at least 1; hence, the total capacity of any backup network

¹ Indeed, assume by way of contradiction the existence of a backup network with a topology that is *not* a duplicate of the given primary network but satisfies the subgraph constraint; clearly, this backup network is disconnected as it contains less than $N-1$ links. Yet, as mentioned, any backup network of a connected primary network must also be connected.

must be of at least $N-1$, and in particular the backup network presented in (a) is optimal. On the other hand, the total capacity of any (unrestricted) optimal backup network for the given primary network is at most $\frac{N}{2}$. Indeed, the backup

network presented in (b) is both feasible² and allocated a backup capacity of $\frac{N}{2}$ units. Thus, it holds that

$$\rho(H^-, U^-, S^+) \geq \frac{N-1}{\frac{N}{2}} = 2 \cdot (1 - \frac{1}{N}); \text{ also, since } \rho(H^-, U^-, S^+) \geq \rho(H^-, U^-, S^+)$$

must hold, it also holds that $\rho(H^-, U^-, S^+) \geq 2 \cdot (1 - \frac{1}{N})$. These lower bounds, together with the upper bounds of Theorem 2, establish the following corollary.

Corollary 3: The ratios $\rho(H^-, U^-, S^+)$ and $\rho(H^+, U^+, S^+)$ are bounded from below by $2 \cdot (1 - \frac{1}{N})$ and from above by 2.

3.4 A Prohibitive Cost for Hop-Count Limits Combined with Subgraph Constraints

In the previous subsections we have shown that the minimum protection capacity increases by a factor of at most 2 when either the backup network must satisfy hop-count limits or the backup network is required to be a subgraph of the primary network; similar efficient guarantees are obtained when each of these constraints is combined with a requirement to support unsplittable routing over the backup network. Yet, in contrast to these positive results, in this subsection we show that when the hop-count limit *and* the subgraph constraint are concurrently imposed, the protection capacity can increase by a factor as large as $\Omega(N)$ i.e., both $\rho(H^-, U^-, S^+) = \Omega(N)$ and $\rho(H^+, U^+, S^+) = \Omega(N)$ hold. Obviously, this dramatic increase in protection capacity is prohibitive for practical purposes and implies that *only one* (i.e., the more significant) of the two constraints should be imposed.

We begin with the ratio $\rho(H^+, U^+, S^+)$. Consider the primary network $G(V, E)$ presented in Fig. 6. Assume that all link capacities are equal to 1. Denote by V_1 the upper set of nodes and by V_2 the lower set of nodes. The network is a complete bipartite graph i.e., there is a link between each pair of nodes $u \in V_1$ and $v \in V_2$ but not between any pair of nodes that are both in V_1 or in V_2 . Let $G^b(V, E^b)$ be a splittable backup network for the primary network $G(V, E)$ that satisfies both a hop-count limit $h=2$ and the subgraph constraint. We first

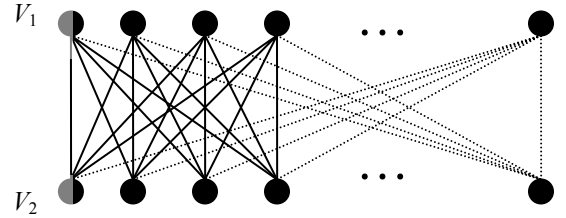


Fig. 6: A primary network with a topology of a complete bipartite graph.

² Indeed, the backup network is a ring with link capacities equal to $\frac{1}{2}$; hence, upon a failure of any primary link, it is possible to split the traffic evenly and reroute each half in the opposite direction.

show that $G^b(V, E^b)$ and $G(V, E)$ must share the same topology, i.e., $G^b(V, E^b)$ must consist of a parallel link for each primary link $e \in E$.

Assume, by way of contradiction, that $G(V, E)$ and $G^b(V, E^b)$ do not share identical topologies. Hence, there must exist a link in the primary network with no corresponding parallel link in the backup network i.e., a link $(u, v) \in E$ such that $(u, v) \notin E^b$. Upon a failure of the link $(u, v) \in E$, the backup network must provide an alternative 2-hop bypass path between the nodes u and v . Yet, it is easy to see that such a bypass cannot exist in the backup network. Indeed, there is no common neighbor v_k to any two nodes v_i, v_j in $G(V, E)$ that are connected by a link; hence, any path between u and v that does not include the direct link (u, v) must have a hop count larger than 2 in the primary network $G(V, E)$. Thus, since $G^b(V, E^b)$ is a subgraph of $G(V, E)$, the bypass for the link (u, v) in $G^b(V, E^b)$ must be larger than 2. Obviously, this contradicts the assumption that $G^b(V, E^b)$ is a backup network that consists of bypasses with at most 2 hops. Thus, we conclude that $G^b(V, E^b)$ and $G(V, E)$ must share identical topologies.

We now employ the fact that $G^b(V, E^b)$ and $G(V, E)$ share the same topology to show that the total capacity of $G^b(V, E^b)$ is at least $\frac{N^2}{4}$. To that end, note that upon a failure of any primary link e , only the link that is parallel to e in the backup network satisfies the hop-count restriction $h=2$; indeed, we have already shown that all other bypass paths violate this hop-count limit. Hence, upon a failure of the link e only the link parallel to e (in the backup network) is employed. Therefore, since the capacity of each primary link is 1, the capacity value of each parallel link (i.e., a link in the backup network) must be of at least 1; hence, $\sum_{e^b \in E^b} c_e \geq |E^b|$.

However, since $G^b(V, E^b)$ and $G(V, E)$ share the same topology it holds that $|E^b| = |E|$; hence, $\sum_{e^b \in E^b} c_e \geq |E^b| = |E| = \frac{N^2}{4}$.

Literally, any splittable backup network for $G(V, E)$ that concurrently satisfies the hop-count limit and the subgraph constraint must be allocated with a capacity of at least $\frac{N^2}{4}$.

On the other hand, it is easy to see that when no restrictions on the backup network are imposed, any spanning tree (defined over V) with link capacities equal to one, is a feasible backup network for $G(V, E)$; since the total capacity in such a case is $N-1$, the total capacity of any (unrestricted) optimal backup network for $G(V, E)$ is at most $N-1$. Thus, we conclude that $\rho(H^+, U^-, S^+) = \frac{N^2}{N-1} = \Omega(N)$ i.e., when neither the hop-count limits nor the subgraph requirements are considered, the capacity of the backup network can decrease by a factor of $\Omega(N)$. This lower bound, together with the fact that $\rho(H^+, U^+, S^+) \geq \rho(H^+, U^-, S^+)$, establish the following corollary.

Corollary 4: The ratios $\rho(H^+, U^-, S^+)$ and $\rho(H^+, U^+, S^+)$ are bounded from below by $\Omega(N)$.

4. CONSTRUCTING BACKUP NETWORKS

In this section we show how to construct backup networks for any given primary network. We first present *optimal* construction algorithms for backup networks that satisfy the design constraints mentioned in the Introduction. Then, we consider the computational complexity of each algorithm. For the cases where the computational complexity is prohibitive, we present *constant* approximation algorithms that construct the backup networks in *polynomial* time; we also present some numerical results that show that the running time of all the proposed optimal algorithms is feasible for practical purposes. Finally, we show how to modify the proposed algorithms in order to construct backup networks that protect against *correlated* link failures.

4.1. Optimal Algorithms

In this section we formulate linear and integer programs, the solution of each identifies an optimal backup network for any given primary network. For ease of presentation, we transform each undirected link in the primary network $G(V, E)$ into two directed links with opposite directions that have each the same capacity as the original (undirected) link.¹ Denote by $f_e(e)$ the total flow rerouted over the backup link $e^b = (w_1, w_2) \in E^b$ upon a failure on the link e ; let c_{e^b} denote the capacity of the backup link $e^b \in E^b$. Upon a failure on a primary link $e = (u, v) \in E$, the flow $f_e(e)$ carried over the backup link $e^b = (w_1, w_2) \in E^b$ is a composition of flows that are rerouted from u to w_1 through bypass paths of different hop counts. Let $f_{e^b}^h(e)$ be the total flow over $e^b = (w_1, w_2) \in E^b$ that is rerouted from u to w_1 through bypasses (from u to w_1) with a hop-count of exactly h upon a failure on the primary link $e = (u, v) \in E$.

If the topology of the backup network must be a subgraph of the primary network $G(V, E)$, we set $E^b \leftarrow E$ (yet, it is possible to assign zero capacities to backup links, i.e., $c_{e^b} = 0$ for some $e^b \in E^b$). On the other hand, when there is no restriction on the resulting topology (i.e., the subgraph constraint is not imposed and arbitrary topologies are allowed) we set $E^b \leftarrow V \times V$. For each $v \in V$, we denote by $O(v)$ the set of all links in E^b that emanate from v , and by $I(v)$ the set of all links in E^b that enter that node, namely $O(v) = \{v \rightarrow l \mid v \rightarrow l \in E^b\}$ and $I(v) = \{w \rightarrow v \mid w \rightarrow v \in E^b\}$. Finally, for every link $e = u \rightarrow v$, let

¹ Recall that an undirected link with a capacity c_e represents two directed links such that each of the links can transfer at most c_e flow units. Therefore, the total capacity of all links in the undirected representation and in the directed representation differs by a factor of 2. In particular, minimizing the total capacity of all links in the directed representation also minimizes the total capacity in the undirected representation.

$s_e=u$ and $t_e=v$. Then, a corresponding linear program whose solution is an optimal splittable backup network that satisfies the hop-count and subgraph constraints can be formulated using the variables $\{\{f_{e^b}^h(e)\}, \{c_{e^b}\}\}$, as specified in Fig. 7.

<p>Program Backup Network ($G(V,E), E^b, H$)</p> <p>Parameters:</p> <p>$G(V,E)$ – the primary network; E^b – the connectivity of the backup network; H – the hop-count limit.</p> <p>Minimize $\sum_{e^b \in E^b} c_{e^b}$ (1)</p> <p>Subject to:</p> $\sum_{e^b \in O(v)} f_{e^b}^h(e) - \sum_{e^b \in I(v)} f_{e^b}^{h-1}(e) = 0 \quad \begin{matrix} \forall e \in E, \forall h \in [0, H], \\ \forall v \in V \setminus \{s_e, t_e\} \end{matrix} \quad (2)$ $\sum_{e^b \in O(s_e)} f_{e^b}^0(e) \geq c_e \quad \forall e \in E \quad (3)$ $\sum_{e^b \in I(t_e)} \sum_{h=0}^H f_{e^b}^h(e) \geq c_e \quad \forall e \in E \quad (4)$ $\sum_{h=0}^H f_{e^b}^h(e) \leq c_{e^b} \quad \forall e \in E, \forall e^b \in E^b \quad (5)$ $f_{e^b}^h(e) = 0 \quad \forall e^b \in E^b, \forall e \in E, h \notin [0, H-1] \quad (6)$ $f_{e^b}^h(e) \geq 0 \quad \forall e \in E, \forall e^b \in E^b, h \in [0, H] \quad (7)$ $c_{e^b} \geq 0 \quad \forall e^b \in E^b \quad (8)$ $f_{e_1^b}^h(e) = f_{e_2^b}^h(e) \quad \begin{matrix} \forall e \in E, \forall h \in [0, H], \forall e_1^b, e_2^b \in E^b \\ \text{s.t. } s_{e_1} = t_{e_2} \text{ and } t_{e_1} = s_{e_2} \end{matrix} \quad (9)$	
---	--

Fig. 7: Constructing backup networks with minimum capacity.

The objective function (1) minimizes the total capacity allocated to the backup network $G^b(V, E^b)$. Constraint (2) is the nodal flow conservation constraint of the backup flow (i.e., flow on the backup network). More specifically, equation (2) states that upon a failure of any primary link $e=s_e \rightarrow t_e$, the total backup flow that enters into any node $v \in V \setminus \{s_e, t_e\}$ and has traversed paths (from s_e to v) of hop-count $h-1$ must equal the total backup flow emanating out of that node through paths of hop count h . Equations (3) and (4) ensure that, for each primary link $e \in E$, the backup network reserves at least a total capacity of c_e along the paths that connect the end-nodes of e ; specifically, constraint (3) emits flow from the end-node s_e of each primary link $e=s_e \rightarrow t_e$ a backup flow of at least c_e flow units; similarly, constraint (4) absorbs

at the end node t_e of each primary link $e=s_e \rightarrow t_e$, a total backup flow of at least c_e flow units. Equation (5) ensures that, upon a failure of any primary link $e \in E$, the total flow rerouted over each backup link $e^b \in E^b$ is at most c_{e^b} .

Expression (6) rules out non-feasible flows that violate the hop restriction, and Expressions (7) and (8) restrict all variables to be non-negative. Finally, Expression (9) restricts the solution to be symmetrical; hence, it rules out all non-feasible solutions that are not feasible for the (original) *undirected* network.

Note that, since Equation (7) allows the variables $\{f_{e^b}^h(e)\}$ to take *any* non-negative value, the rerouted flow upon a failure of any link $e \in E$ can be *split* among several paths; hence, the solution (that consists of the variables $\{c_{e^b}\}$) constitutes an optimal *splittable* backup network for the given hop-count restriction H . All that is needed in order to transfer the backup flow *unsplittably* is to modify Equation (7) so that each variable $f_{e^b}^h(e)$ would take either the value 0 or the value c_e for each $e \in E$, $e^b \in E^b$ and $h \in [0, H]$. Note that, by doing so, we obtain an *integer program* that constructs an *optimal unsplittable* backup network while satisfying the given design constraints.

Both the linear program (that corresponds to the splittable case) and the integer program (that corresponds to the unsplittable case) can be solved by commercial software tools such as CPLEX or MOSEK [3]. We now consider the running time of each of these programs. To that end, it is important to note that the number of variables and constraints in Program Backup Network is polynomial in the network size. Indeed, the hop count restriction H is at most $N-1$; therefore, the number of variables $\{\{f_{e^b}^h(e)\}, \{c_{e^b}\}\}$ and the number of constraints is in the order of $N^3 \cdot M$. Thus, since the complexity incurred by solving a linear program is polynomial in the number of constraints and the number of variables [6], Program Backup Network has a polynomial running time for the splittable case. On the other hand, for the unsplittable case Program Backup Network is an integer program that has no polynomial solution in the general case.

4.2. Fast Approximation Algorithms

In the previous subsection, we have shown that the running time of the proposed optimal construction schemes is polynomial in the input for the splittable case but may be intractable for the unsplittable case. Fortunately, in this subsection, we observe that Procedure QoS-Backup Network (Fig. 2) and Procedure Subgraph Backup Network (Fig. 4) can be used as alternative approximations for the construction of unsplittable backup networks. Specifically, these approximations establish unsplittable backup networks that satisfy either the subgraph constraint or the hop-count constraint; moreover, we show that both of them are operated in *low polynomial time* and produce backup networks with a total capacity of *at most twice the optimum*.

Procedure QoS-Backup Network and Procedure Subgraph Backup Network are shown to efficiently approximate the (corresponding) optimal solutions in the proof of Theorems 1 and 2. Specifically, while both are shown to return an unsplittable backup network with a capacity of at most twice the optimum¹, Procedure QoS-Backup Network is guaranteed to return a backup network that consists of solely 2-hop protection paths and Procedure Subgraph Backup Network is guaranteed to return a backup network that satisfies the subgraph constraint. Moreover, it is easy to see that the execution times of Procedure QoS-Backup Network and Procedure Subgraph Backup Network are $O(N)$ and $O(MN)$, respectively. Thus, these procedures can consider either the hop-count constraint or the subgraph constraint (but not both²), while providing attractive performance guarantees on both running time and allocated protection capacity.

4.3. Construction Algorithms for Backup Networks: Slow & Optimal or Fast & Suboptimal?

Limiting the running time of network algorithms that are frequently executed (e.g., routing protocols, scheduling and switching algorithms, web services, etc.) is usually of major practical importance. Accordingly, in such contexts heuristics and approximations improve their running time at the price of deteriorating the quality of the returned solution. Yet, in our context such a compromise is usually not required. First, the *single* computation of an optimal network is often followed by an installation and configuration process that can last for days. At the same time, the consequences of a poorly designed network may impose a prohibitive toll (as opposed, e.g., to an occasional packet that is not sent along the best path or scheduled in the best time slot). Therefore, unless the computation time of the construction algorithm is infeasible for practical purposes (e.g., days), the quality of the solution may well be favored over the computation time; hence, the optimal construction algorithms (specified in subsection 4.1), would be favored over the constant approximation schemes (specified in subsection 4.2). In the following, we indicate that Program Backup Network (specified in subsection 4.1) can be applied to establish optimal backup networks (for typical primary networks) in the order of minutes.

¹ Both procedures have shown to return an unsplittable backup network with a capacity of at most $\sum_{v \in V'} C(v)$; This, together with Lemma 1 that establishes a lower bound of $\frac{1}{2} \sum_{v \in V'} C(v)$ on the minimum capacity of *any* backup network, prove that the procedures return a solution that is within a factor of 2 away from the optimum.

² We have shown in Section 3.3 that the increase in the required protection capacity when both constraints are concurrently satisfied is prohibitive for practical purposes. Thus, although the procedures can satisfy only one of the two constraints, a unifying scheme for both constraints is of limited interest from a practical viewpoint.

Following the lines of [4], we generated³ random topologies with N nodes and invoked Program Backup Network over each topology. Then, we measured the time needed to construct an unsplittable backup network with a hop-count limit $H=2$; the program was implemented in Matlab 6.0 and run in a 2Ghz Intel 4 machine. Finally, we averaged the measurements over 150 runs for each N , $N \in \{10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70\}$. In Fig. 8 we depict the average running time versus network size (number of nodes). Note that the running time is increased *linearly* with the number of nodes; hence, from practical point of view, it scales very efficiently with N . Moreover, for networks with less than 70 nodes the backup network is constructed in a time smaller than a minute.

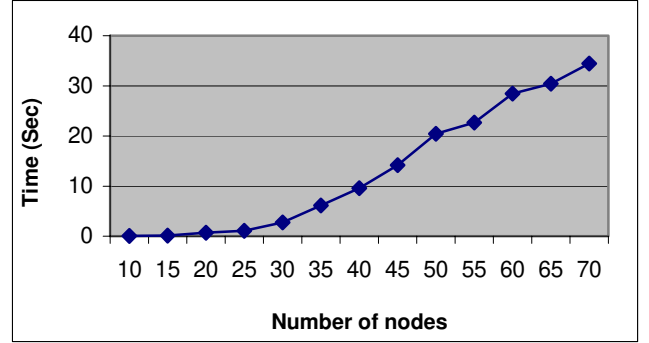


Fig. 8: Running time increases linearly with network size

4.4. Coping with Correlated Failures

It has been reported [11] that nearly 30% of all link failures are correlated. Accordingly, in this subsection we extend our work to a framework where a failure of one primary link affects other primary links. To that end, we assign to each primary link $e \in E$ a *failure correlated set* $F(e) \subseteq E$ such that, upon a failure of the link e , all links in $F(e)$ fail. Then, given a primary network $G(V, E)$ and a correlated set $F(e)$ for each $e \in E$, our goal is to design an optimal backup network that considers the design constraints such that, upon a failure of a link $e \in E$, the backup network provides protection against the failures of *all* links in $F(e) \cup \{e\}$.

Our solution for the above problem is to extend Program Backup Network (specified in Fig. 7) to deal with correlated failures. To that end, note that when no correlation among the failures exists (and only single failures can take place), the bypass paths can intersect each other and *share* the same amount of capacity, provided they are used for the failures of *different* primary links in the network. On the other hand, for correlated failures each backup link $e^b \in E^b$ must be able to carry the backup flows induced by *all* link failures in the correlated set $F(e)$. Hence, if a set of primary links (say, e_1, e_2, \dots, e_k) concurrently fail and their associated backup flows cross the same link $e^b \in E^b$ then, it is required that

³The construction is specified in Section 5 under the Power-Law topology model.

$f_{e^b} = f_{e^b}(e_1) + f_{e^b}(e_2) + \dots + f_{e^b}(e_k) \leq c_{e^b}$. Thus, we only have to modify Equation (5) in Program Backup Network that restricts the backup capacities $\{c_{e^b}\}$ to be large enough to carry the rerouted traffic caused by *single* failures to be large enough to carry the rerouted traffic caused by correlated failures. More specifically, by replacing Equation (5) with the restriction $\sum_{e' \in F(e) \cup \{e\}} f_{e^b}(e') = \sum_{e' \in F(e) \cup \{e\}} \sum_{h=0}^H f_{e^b}^h(e') \leq c_{e^b}$ for each primary link $e \in E$ and backup link $e^b \in E^b$, every backup link e^b has sufficient capacity c_{e^b} to carry all backup flows induced by the failure of the correlated set $F(e)$ and the link e ; hence, the resulting backup network can protect against correlated failures while minimizing the allocated protection capacity.

5. HOW EFFECTIVE IS THE PROPOSED APPROACH?

In this section, we show that the total capacity that must be allocated to protect a primary network is minor compared to the capacity of the primary network. To that end, we evaluate the ratio between the total capacity of a primary network and the total capacity of a corresponding *optimal* backup network. Clearly, a large value for this ratio indicates that, by proper design, it is possible to allocate a single unit of protection capacity against a large amount of capacity in the primary network, thus implying that the proposed approach induces only a small overhead in terms of capacity.

Given a primary network $G(V, E)$ and the corresponding *optimal unsplittable* backup network $G^b(V, E^b)$, let $\rho(G)$ be the ratio between the total capacity of $G^b(V, E^b)$ and the total capacity of $G(V, E)$ i.e.,

$$\rho(G) \triangleq \frac{\sum_{e \in E} c_e}{\sum_{e \in E^b} c_e}.$$

In the following we analytically show that, for both Waxman topologies [15] and Power-Law topologies [4], the expected value of this ratio is large. Note that, since the total protection capacity of splittable backup networks is never larger (and is usually smaller) than that of unsplittable backup networks, the bound $\rho(G)$ holds also for the splittable case.

We first present our analysis for Waxman topologies [15]. In this model, a pair of nodes is located at the diagonally opposite corners of a square area of unit dimension. Then, $N-2$ nodes are uniformly spread over the square. Finally, a link between each two nodes u and v , is introduced with the following probability, which depends on the distance $\delta(u, v)$ between them:

$$p(u, v) = \alpha \cdot \exp\left[\frac{-\delta(u, v)}{\beta \cdot \sqrt{2}}\right], \quad (*)$$

where α and β are parameters in the range $(0, 1]$.

We assume (for both the Waxman and the Power-Law models) that the capacities of all links in the constructed network have the same order of magnitude, which is the typical case. Indeed, in optical networks, there are a few standard sizes of bandwidth for optical links, each with a transmission capability between several hundreds of Mbit/sec to a few Gbit/sec¹.

Theorem 3: Assuming a Waxman topology, the expected ratio between the total capacity of a primary network and that of the corresponding optimal backup network is $\Omega(N)$ i.e., $E[\rho(G)] = \Omega(N)$.

Proof: In the proof of Theorem 1, we have shown that Procedure QoS-Backup Network constructs, for any given primary network $G(V, E)$, an unsplittable backup network with a total capacity of at most $\sum_{v \in V} C(v)$.² In particular, for any given primary network $G(V, E)$, it holds that

$$\rho(G) \geq \frac{\sum_{e \in E} c_e}{\sum_{v \in V} C(v)}; \quad \text{thus, since } \sum_{e \in E} c_e \geq M \cdot c_{\min} \quad \text{and} \quad \sum_{v \in V} C(v) \leq N \cdot c_{\max} \quad \text{for } M = |E|, \quad c_{\min} \triangleq \min_{e \in E} \{c_e\} \quad \text{and} \quad c_{\max} \triangleq \max_{e \in E} \{c_e\},$$

each primary network $G(V, E)$ must satisfy

$$\rho(G) \geq \frac{\sum_{e \in E} c_e}{\sum_{v \in V} C(v)} \geq \frac{M \cdot c_{\min}}{N \cdot c_{\max}}. \quad (2)$$

Assume now that $G(V, E)$ is a primary network of a Waxman topology. In the following, we employ (2) to show that the expected value of $\rho(G)$ is $\Omega(N)$. To that end, define for each pair of nodes $v_i, v_j \in V$, an indicator $I_{i,j}$, which is 1 if link (v_i, v_j) exists in $G(V, E)$, and is 0 otherwise; note that, by definition, $P\{I_{i,j}=1\} = p(v_i, v_j)$, where the latter is given by (*). Then, the total number of links M can be written as a sum of indicators i.e., $M = \sum_{(v_i, v_j) \in V \times V} I_{i,j}$. Therefore, $E[M] = E\left[\sum_{(v_i, v_j) \in V \times V} I_{i,j}\right] = \sum_{(v_i, v_j) \in V \times V} E[I_{i,j}] = \sum_{(v_i, v_j) \in V \times V} [0 \cdot P\{I_{i,j}=0\} + 1 \cdot P\{I_{i,j}=1\}] = \sum_{(v_i, v_j) \in V \times V} P\{I_{i,j}=1\}$.

Finally, note that, since the distance between any two nodes in the square area is at most $\sqrt{2}$, it holds that $P\{I_{i,j}=1\} = p(v_i, v_j) = \alpha \cdot \exp\left[\frac{-\delta(v_i, v_j)}{\beta \cdot \sqrt{2}}\right] \geq \alpha \cdot e^{-\frac{1}{\beta}}$ for each $v_i, v_j \in V$. Hence,

$$E[M] = \sum_{(v_i, v_j) \in V \times V} P\{I_{i,j}=1\} \geq N \cdot (N-1) \cdot \alpha \cdot e^{-\frac{1}{\beta}}. \quad (3)$$

We are finally ready to derive the expected value of $\rho(G)$. Since N is a predefined parameter in the construction of the network, it holds according to (2) that

¹ E.g., most links between backbone routers are either OC48 or OC192 (with line rates of 2.48 Gbit/s and 9.95 Gbit/s, respectively) whereas the links between access routers and backbone routers are either OC3 or OC12 (with line rates of 155.52 Mbit/s and 622 Mbit/s, respectively). As such, all link capacities in a typical optical network have the same order of magnitude.

² Recall that $C(v)$ denotes the maximum capacity of a link incident on v .

$$E[\rho(G)] \geq E\left[\frac{M \cdot c_{\min}}{N \cdot c_{\max}}\right] = \frac{1}{N} \cdot E\left[M \cdot \frac{c_{\min}}{c_{\max}}\right]. \quad (4)$$

Moreover, since we assume that all link capacities have the same order of magnitude, there exists some positive constant k such that $\frac{c_{\min}}{c_{\max}} \geq k$; hence, it holds according to (4) that,

$$E[\rho(G)] \geq \frac{1}{N} \cdot E\left[M \cdot \frac{c_{\min}}{c_{\max}}\right] \geq \frac{1}{N} \cdot E[M \cdot k] = \frac{k}{N} \cdot E[M].$$

This, together with (3) and the fact that k , α and β are all independent of N , establish that

$$E[\rho(G)] \geq \frac{k}{N} \cdot E[M] \geq \frac{k}{N} \cdot \left[N \cdot (N-1) \cdot \alpha \cdot e^{-\frac{1}{\beta}}\right] = \Omega(N).$$

Thus, the theorem is established. ■

We turn to present our analysis for the Power-Law topology model [4], in which the node degrees follow a power-law distribution. Specifically, the probability $p(d)$ of having a node with a degree d is proportional to the value of that degree raised by some negative constant power i.e., $p(d) \propto d^{-\alpha}$ for some $\alpha > 0$. It was observed in [4] that a typical value corresponding to practical network topologies is $\alpha \approx 2$. Accordingly, we assume in our analysis that $p(d) \propto d^{-2}$. We now present the expected ratio $\rho(G)$ for this model.

Theorem 4: Assuming a Power-Law topology, the expected ratio between the total capacity of a primary network and that of the corresponding optimal backup network is $\Omega(\ln N)$ i.e., $E[\rho(G)] = \Omega(\ln N)$.

Proof: Let d_v denote the degree of node $v \in V$ in the primary network $G(V, E)$. Then, the total number of links M (in the primary network) can be written as $M = \frac{1}{2} \cdot \sum_{v \in V} d_v$. Thus, it holds that the expected value of M is bounded by

$$E[M] = E\left[\frac{1}{2} \cdot \sum_{v \in V} d_v\right] = \frac{1}{2} \cdot \sum_{v \in V} E[d_v]. \quad (5)$$

Next, since all node degrees are distributed identically according to the power-law distribution $p(d) \propto d^{-2}$, it follows that $p(d_v = n) \propto n^{-2}$ for each $v \in V$, and let β be a positive constant such that $p(d_v = n) = \beta \cdot n^{-2}$. Hence, it holds that

$$E[d_v] = \sum_{n=1}^N n \cdot p(d_v = n) = \sum_{n=1}^N n \cdot (\beta \cdot n^{-2}) = \beta \cdot \sum_{n=1}^N \frac{1}{n} \text{ for each } v \in V.$$

Since $\frac{1}{n}$ is a monotonically decreasing function, it holds that $\sum_{n=1}^N \frac{1}{n} \geq \int_1^{N+1} \frac{1}{x} \cdot dx$ (see for example [2]); hence, it

$$\text{holds that } E[d_v] = \beta \cdot \sum_{n=1}^N \frac{1}{n} \geq \beta \cdot \int_1^{N+1} \frac{1}{x} \cdot dx = \beta \cdot \ln(N+1) \text{ for each } v \in V.$$

In particular, according to (5) it follows that,

$$E[M] = \frac{1}{2} \cdot \sum_{v \in V} E[d_v] \geq \frac{1}{2} \cdot N \cdot \beta \cdot \ln(N+1). \quad (6)$$

We are finally ready to upper-bound the expected value of $\rho(G)$. We have shown in (2) that, for any given primary network $G(V, E)$, it holds that $\rho(G) \geq \frac{M \cdot c_{\min}}{N \cdot c_{\max}}$. Therefore, since N is a predefined parameter and c_{\min} , c_{\max} are in the

same order of magnitude there exists some positive constant k where $\frac{c_{\min}}{c_{\max}} \geq k$ such that,

$$E[\rho(G)] \geq E\left[\frac{M \cdot c_{\min}}{N \cdot c_{\max}}\right] \geq E\left[\frac{M}{N} \cdot k\right] = \frac{k}{N} \cdot E[M].$$

This, together with (6) and the fact that β and k are independent of N establishes that,

$$E[\rho(G)] \geq \frac{k}{N} \cdot E[M] \geq \frac{k}{N} \cdot \left[\frac{1}{2} \cdot N \cdot \beta \cdot \ln(N+1)\right] = \Omega(\ln N).$$

Thus, the theorem is established. ■

Remark: Theorems 3 and 4 readily extend to hold also for unsplittable backup networks that satisfy either a hop count-limit of h ($h \geq 2$) over the bypass paths or unsplittable backup networks that satisfy the subgraph constraint. Indeed, according to Section 3, these constraints involve only $O(1)$ increase in the (minimum) protection capacity.

The above results demonstrate the major benefit in the construction of a dedicated backup network for a primary network. Specifically, they indicate that, at a minor price of extra capacity, it is possible to allocate to each primary link a single 2-hop bypass path, thus enabling an efficient mechanism for protection that also satisfies rigid QoS requirements¹ for the rerouted traffic.

6. CONCLUSIONS

Fast restoration schemes allocate resources for protection either on demand (i.e., upon the arrival of a connection request) or during the configuration phase of the network. While the first approach incurs considerable overhead in terms of connection set-up time, the second approach requires allocating protection resources for *any* possible pattern of connection requests hence usually calls for substantial over-provisioning. The restoration approach described in this study overcomes both drawbacks. Indeed, it computes and allocates all protection resources during the configuration phase, and at the same time it consumes only a small amount of protection resources. Another fundamental advantage of the proposed approach is its ability to recover from failures without overloading other links on the primary network. This turns to be of a major practical importance when considering the findings of [5], according to which 80% of link overloads are due to *link failures*. Moreover, the proposed approach frees the primary network from any restoration considerations, e.g., it allows employing standard and simple routing algorithms.

Motivated by these results, we considered three major design requirements, namely: (i) bypass paths should have a bounded hop-count; (ii) rerouted traffic should be split among a bounded number of bypasses; and (iii) the topology of the backup network should be a subgraph of the primary network. We rigorously quantified the increase in protection capacity due to each combination of the above constraints. The obtained results provide several important insights and

¹ Specifically, the rerouted traffic is carried unsplittably and exclusively over 2-hop protection paths.

design rules, namely: (i) from a practical viewpoint, it may be unfeasible to concurrently incorporate the subgraph and hop-count constraints; (ii) all other combinations of design constraints can be efficiently incorporated in the backup network at a small price; and (iii) when unsplittable routing must be supported, hop-count limits can be incorporated at no price.

We have also considered several efficient construction algorithms for backup networks. Specifically, for each possible combination of design constraints, we presented a corresponding optimal construction algorithm. For the cases where the computational complexity of the optimal algorithm is significant, we established alternative *constant* approximation schemes with low polynomial running time. Finally, we showed how to modify the optimal construction algorithms to protect against correlated failures.

Several important directions for future research follow from our study. First, throughout our study we focused on the *worst-case* ratio $\rho(H, U, S)$ for quantifying the increase in protection capacity for a combination of design constraints (H, U, S) . While we have shown that $\rho(H^+, U^+, S^+) = \Omega(N)$ and $\rho(H^-, U^-, S^-) = \Omega(N)$, nothing is known yet about the value of these ratios in typical cases; in particular, preliminary simulation results indicate that these ratios are substantially smaller in practice. Next, while our study mainly focused on the single link failure model, it may be interesting to investigate properties of backup networks that provide protection against multiple (uncorrelated) failures; in particular, it is important to investigate whether such backup networks still induce tolerable overhead in terms of protection capacity. Finally, we would like to incorporate design considerations that reflect specific application requirements; for example, when traffic is rerouted over multiple bypass paths, it is essential in delay-sensitive applications (e.g., voice or video) to limit the delay differences (i.e., the delay-jitter) between the bypasses.

REFERENCES

- [1] A. Charny and J. Y. LeBoudec, *Delay Bounds in a Network with Aggregate Scheduling*, First International Workshop on Quality of Future Internet Services, Berlin Germany, Sep. 2000.
- [2] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*, Cambridge, MA: The MIT Press, 2001.
- [3] CPLEX, *CPLEX 9.0 manual*, ILOG, <http://www.ilog.com>, 2005.
- [4] M. Faloutsos, P. Faloutsos, and C. Faloutsos, *On Power-law Relationships of the Internet Topology*, in Proc. ACM SIGCOMM, Cambridge, MA, September 1999.
- [5] S. Iyer, S. Bhattacharyya, N. Taft, and C. Diot, *An approach to alleviate link overload as observed on an IP backbone*, in Proc. IEEE INFOCOM, San Francisco, CA, USA, Mar. 2003.
- [6] N. Karmarkar, *A New Polynomial-Time Algorithm For Linear Programming*, *Combinatorica*, vol. 4, pp. 373-395, 1984.
- [7] A. Konak and A. E. Smith, *Designing Resilient Networks Using A Hybrid Genetic Algorithm Approach*, In proc. Genetic And Evolutionary Computation Conference (GECCO), Washington D.C., June 2005.
- [8] R. Krishnaswamy and K. N. Sivarajan, *Design of logical topologies: a linear formulation for wavelength-routed optical networks with no wavelength changers*, *IEEE/ACM Transaction on Networking*, April. 2001.
- [9] Y. Liu, D. Tipper and P. Siripongwutikorn, *Approximating Optimal Spare Capacity Allocation by Successive Survivable Routing*, *IEEE/ACM Transaction on Networking*, Feb. 2005.

- [10] G. Maier, A. Pattavina, S. De Patre, and M. Martinelli, *Optical Network Survivability: Protection Techniques in the WDM Layer*, *Photonic Networks Communications*, vol. 4, no. 3-4, July-Dec. 2002.
- [11] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. N. Chuah and C. Diot, *Characterization of Failures in an IP Backbone* in Proc. IEEE INFOCOM, Hong-Kong, March 2004.
- [12] S. Nelakuditi and Zhi-Li Zhang, *On Selection of Paths for Multipath Routing*, In proc. IWQoS, Karlsruhe, Germany, 2001.
- [13] A. Orda and A. Sprintson, *Efficient Algorithms for Computing Disjoint QoS Paths*, in Proc. IEEE INFOCOM, Hong Kong, March 2004.
- [14] D. Thaler and C. Hopps, *Multipath Issues in Unicast and Multicast Next-Hop Selection*, IETF RFC 2991, 2000.
- [15] B. M. Waxman, *Routing of Multipoint Connections*, *IEEE Journal on Selected Areas in Communications*, 6:1617-1622, 1988.
- [16] D. Xu, Y. Chen, Y. Xiong, C. Qiao, H. He, *On Finding Disjoint Paths in Single and Dual link Cost Networks*, IEEE INFOCOM, Hong-Kong, March 2004.