Computer Networks xxx (2009) xxx–xxx

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Efficient and scalable provisioning of always-on multicast streaming services

Meeyoung Cha [a,*], W. Art Chaovalitwongse [b], Jennifer Yates [c], Aman Shaikh [c], Sue Moon [d]

[a] MPI-SWS, Germany
[b] Rutgers University, USA
[c] AT&T Labs – Research, USA
[d] KAIST, Korea

ARTICLE INFO

Article history:
Received 30 May 2008
Received in revised form 24 June 2009
Accepted 13 July 2009
Available online xxxx
Responsible Editor: Qian Zhang

Keywords:
Network design
Multicast
Shared Risk Link Group (SRLG)
Integer Linear Programming (ILP)
Scalability
Heuristic algorithms
Capital expense

ABSTRACT

There is a growing need for large-scale distribution of realtime multicast data such as Internet TV channels and scientific and financial data. Internet Service Providers (ISPs) face an urgent challenge in supporting these services; they need to design multicast routing paths that are reliable, cost-effective, and scalable. To meet the realtime constraint, the routing paths need to be robust against a single IP router or link failure, as well as multiple such failures due to sharing fiber spans (SRLGs). Several algorithms have been proposed to solve this problem in the past. However, they are not suitable for today's large networks, because either they do not find a feasible solution at all or if they do, they take a significant amount of time to arrive at high-quality solutions.

In this paper, we present a new Integer Linear Programming (ILP) model for designing a cost-effective and robust multicast infrastructure. Our ILP model is extremely efficient and can be extended to produce quality-guaranteed network paths. We develop two heuristic algorithms for solving the ILP. Our algorithms can guarantee to find high-quality, feasible solutions even for very large networks. We evaluate the proposed algorithms using topologies of four operational backbones and demonstrate their scalability. We also compare the capital expenditure of the resulting multicast designs with existing approaches. The evaluation not only confirms the efficacy of our algorithms, but also shows that they outperform existing schemes significantly.

1. Introduction

Large-scale, always-on streaming services that have been traditionally offered in dedicated networks are now supported in the Internet via multicast (or one-to-many communication) technologies. One example is Internet TV (IPTV)—the distribution of television and video signals over an IP network [25,12,44]. Internet Service Providers (ISPs) worldwide are building backbone infrastructures for IPTV. According to Parks Associates, the number of IPTV subscribers, which was 10.85 million in 2007, will grow to 60 million by 2011. Another important application for robust multicast technologies is the distribution of financial, scientific, and corporate data [35,9,32]. Migrating from NASDAQ's dedicated access lines, the New York Stock Exchange now distributes the stock price update ticker over an IP network [7]. As a third example, CERN, the European Organization for Nuclear Research, is streaming hundreds of Gbps of high-energy physics data to its participating laboratories [1]. These examples demonstrate the opportunities for utilizing a large-scale static multicast infrastructure.

Despite several emerging opportunities, designing multicast infrastructure for large-scale, always-on streaming services is very challenging for the following two reasons:

* Corresponding author. Tel.: +49 681 9325 663; fax: +49 681 9325 229.
E-mail address: mcha@mpi-sws.org (M. Cha).

1389-1286/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2009.07.005

Please cite this article in press as: M. Cha et al., Efficient and scalable provisioning of always-on multicast streaming services, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.07.005

(1) Service availability: Customers expect highly available service even under frequent network faults and disruptions. While networks recover from failures through re-convergence of their routing protocols, the re-converge is not rapid enough to meet the realtime demands of interactive services [26,8]. Complex multiple link failures in multicast applications complicate the situation further.
(2) Financial viability: Network components incur operations costs (e.g., leasing cost, maintenance cost) and such costs should be minimized in network designs to be competitive in the market. However, finding a cost-effective design is nontrivial and it is often derived as a very hard optimization problem.

In this paper, we address the problem of provisioning a robust multicast backbone infrastructure in a cost-effective manner. We assume that multicast routing is implemented in the IP layer, where IP multicast streams are groomed and transmitted in the optical layer, as a light tree or a light path. We propose a technique that is pre-planned (i.e., pre-configuring the backup routes for rapid fault recovery) and dedicated (i.e., reserving available capacity required for recovery). A pre-planned and dedicated approach is known to guarantee available bandwidth at the time of failure and the fault recovery is much faster than calculating for backup paths on the fly [20].

We assume that locations of the multicast sources and destinations are predetermined, which is very common in practice. Given two sources and a set of destinations, our goal is to design multicast routing paths in the backbone such that, even in the presence of failures, there is at least one viable path from any one of the sources to every destination. For example in IPTV, national TV head-ends (sources) stream hundreds of TV channels continuously to data centers in major cities (destinations). We assume there exist two sources that send out redundant multicast streams to protect against catastrophic failure of a source. Our model also applies to the case of a single source by replicating the source. There are many ways to control the coordination between the two sources. For the purpose of this work, we assume that each destination is automatically connected to the nearest operating source using the IP anycast technology [2,5].

One salient feature of our approach is the protection against Shared Risk Link Group (SRLG) failures. An SRLG is a group of links that can potentially fail together due to a single cause [38,3,40]. Multiple disjoint links in the IP or even optical layer may share a common fate of failure due to sharing some common risk (e.g., conduit, right-of-way), as illustrated in Fig. 1. Our multicast tree design protects against these complex yet realistic SRLG failures.

Another important feature of our approach is network resource efficiency. We use two active paths connecting two sources to every destination such that these two paths guard the network paths from any single failure. This is similar to a 1 + 1 path protection technique, where two disjoint paths are used between a source and a destination. However, our technique also protects against a source failure by using only two paths per destination (see highlighted paths in Fig. 1), while a 1 + 1 path protection would require a total of four paths per destination. Unlike the existing approaches, our technique aggressively minimizes the network cost by allowing the two multicast trees from each of the sources to overlap. In Section 4 we demonstrate that our approach, compared to the popular Active Path First approach [18,29], can reduce costs by up to 30% in real topologies. The cost reduction comes from our strategy of jointly considering two trees from both sources as opposed to constructing one tree at a time (as in APF).

We develop an off-line optimization framework for constructing multicast routing paths based on Integer Linear Programming (ILP) and also take into account two realistic operations constraints into our ILP model. First, we take into account the quality of provisioned paths such as latency limits and the number of SRLGs per path. Services such as stock tickers and sports channels in IPTV have very tight latency requirements, for example, sport channels in IPTV and stock ticker [9]. The ability to fine-tune service configuration is useful for service providers in meeting such challenges; for instance, they can meet the guaranteed performance in service layer agreement (SLA). Second, we address the scalability challenge. Like all other NP-hard problems, there does not exists polynomial time algorithms to solve the multicast problem considered. Solving this problem to optimality becomes extremely time consuming for large networks—especially in a tier-1 backbone of over a hundred of nodes. Toward this end, we develop two heuristic algorithms to find high-quality, feasible solutions.

The rest of this paper is organized as follows. Section 2 reviews related work and describes the motivation of our work. In Section 3, we present our ILP optimization

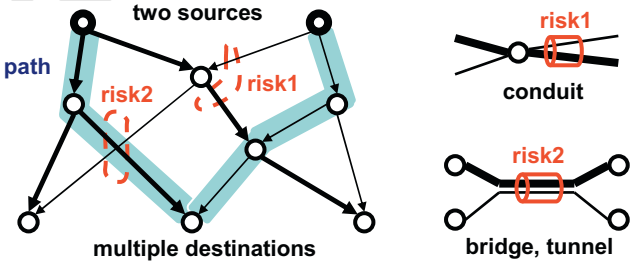


Fig. 1. Service architecture of the multicast provisioning problem with examples of SRLGs. Two diverse links may be placed into the same conduit or bridge at the physical layer and are subject to a single point of failure.

Please cite this article in press as: M. Cha et al., Efficient and scalable provisioning of always-on multicast streaming services, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.07.005

framework and analyze its complexity. We then give performance-aware extensions and also present two heuristic algorithms for large-scale problems. Section 4 evaluates the proposed algorithms using real network topologies. We also demonstrate the trade-off between reliability and cost. Finally, we conclude and discuss issues for further work in Section 5.

2. Background and motivation

This section briefly reviews the related work and introduces the motivation for our work with an illustrative example.

2.1. Path protection routing

Robustness against failures is crucial for service provision. Typically, fault recovery can operate on-demand (i.e., computing backup routes at the time of a failure) or pre-planned (i.e., pre-configuring the backup routes) [20,37]. Because on-demand recovery of multicast requires long latency [15], we consider a *pre-planned* recovery in this paper. In a pre-planned recovery, alternate disjoint backup routes are set up for predefined failures of a node, a link [23,31], or a path [28,4,43]. We consider *path* protection scheme, where a backup path is set from the source to the destination that is independent (no overlapping links or nodes) of the primary active path. Finally, the backup capacity can be dedicated or shared amongst multiple other applications. Sharing the backup capacity does not guarantee a successful restoration as there may not be enough available capacity at the time of a failure. For high level of availability, we *dedicate* the capacity required for both primary and backup paths.

2.2. Shared Risk Link Group (SRLG)

In this paper we consider protection against a realistic Shared Risk Link Group (SRLG) failure. An SRLG is a group of links that can potentially fail together due to a single cause [38]. SRLG consideration is an important practical concern, because it is a dominant form of failure and a single fiber cut can interrupt all the traffic in the fiber, which typically corresponds to a huge loss in traffic [21].

We describe how an SRLG failure is different from a link failure with an illustrative example. The toy example in Fig. 2 shows two sources, $s1$ and $s2$, and three destinations, $d1$, $d2$ and $d3$. Thick and thin arrowed lines represent multicast routing paths from $s1$ and $s2$, respectively. Consider what happens if multiple simultaneous links fail due to a common risk, *risk1*. In this scenario, in the link-diverse design in Fig. 2a the node $d1$ becomes disconnected from the network. In the SRLG-diverse design in Fig. 2b, on the other hand, every destination is still connected to at least one of the sources. Such robustness usually comes at an increased routing cost. Assuming each link has a cost of 1, the total cost of link-diverse routing is 8, while that of SRLG-diverse routing is 9.

Protection against an SRLG not only increases the network routing cost, but finding such paths is also not easy. Ellinas et al. first showed that if an arbitrary set of links can belong to a common SRLG, then the problem of finding SRLG-diverse paths is *NP-complete* [17]. A number of ILP-based approaches and efficient heuristic algorithms have been proposed to solve this NP-complete problem [45,6]. Amongst them, Li et al. [28] formulated the problem so that the total spare capacity allocated in the network is minimized. Shen et al. [40] assumed three classes of SRLG-diverse protection (dedicated, shared, and unprotected) and formulated the problem as a revenue maximization problem for given request connections. One common element in these research is that their transport is unicast (one-to-one). In contrast, we tackle the problem of multicast (one-to-many) communication.

2.3. Provisioning multicast routing paths

Multicast routing paths can be built *statically* or *dynamically*. In a static multicast tree, the locations of the sources and destinations and the traffic demand (i.e., the size and the number of multicast groups) are predetermined. Therefore, a fixed multicast tree is used for content distribution. For instance, nationwide IPTV service and stock ticker distribution utilize a static multicast tree. In contrast, a dynamic multicast tree is often used when traffic

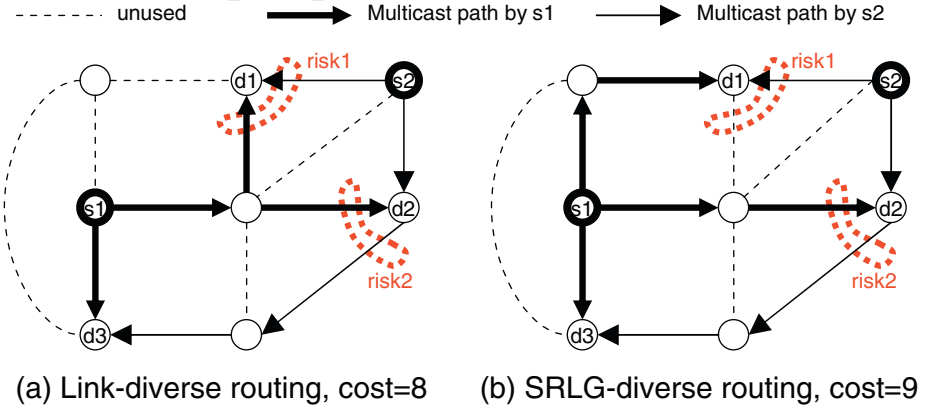


Fig. 2. Comparison of link-diverse and SRLG-diverse routing.

Please cite this article in press as: M. Cha et al., Efficient and scalable provisioning of always-on multicast streaming services, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.07.005

pattern changes over time. For instance, online games and video conferences receive requests from differing sets of participating nodes and users over time.

In this paper, we are motivated by large-scale deployments of nationwide IPTV service, where sources and destinations are fixed in the backbone. Therefore, we consider a protection approach for static multicast traffic. We mention that incorporating dynamic traffic patterns in our framework is beyond the scope of this work and we leave this problem as a future work.

Below, we first review protection approaches for dynamic multicast traffic. Although these approaches are not directly applicable to our work, they bring insight into the key strategies and technologies for provisioning multicast routing paths. We then review protection approaches for static multicast traffic, which is the focus of this paper.

2.3.1. Provisioning dynamic multicast traffic

Most of the existing protection approaches for dynamic multicast traffic can be classified into (i) path-based protection, (ii) tree-based protection, (iii) segment-based protection, (iv) ring-based protection, and (v) cycle-based protection. We briefly discuss each of the protection approaches.

In path-based protection, the goal is to identify a backup path that is disjoint for each "primary" or working path. Singhal et al. proposed optimal path pair protection, a technique to dynamically construct resilient backup paths, while allowing as much dynamic traffic as possible (i.e., minimizing the blocking probability) [43,42].

In tree-based protection, the goal is to find a backup tree that does not share any resources with the primary multicast tree. Singhal and Mukherjee constructed mathematical models to find link-disjoint backup trees at a globally optimum cost [41]. Li et al. also proposed a variant of a shortest path tree algorithm to find SRLG-diverse trees for newly arriving traffic [29]. In computing the protection tree, the links metrics are altered to avoid any common SRLG, as well as to promote sharing of backup resources.

In segment-based protection, working paths are divided into overlapping segments (or sub-paths) and the goal is to protect against each segment failure [10]. Liao et al. proposed a protection algorithm that dynamically adjusts the link cost of the network to establish a primary multicast tree that has link-disjoint backup segments [30]. A backup segment can efficiently share resource with its working tree or with other backup segments.

In ring-based protection, the two segments in the working path and the backup path form a ring structure, so that traffic can be switched to the backup path at the time of a failure. Ring-based methods are fast in restoration (less than 50 ms), because only the end nodes of the failed link take action [20]. Rahman and Ellinas proposed a multicast protection algorithm that is based on a link-disjoint tree or the collapsed ring protection method [36]. Hwang et al. also proposed a ring protection technique in which they find disjoint subtrees of a multicast tree and reserve spare capacity for rapid local recovery [23]. Leelarusmee et al. conducted a comparison study and demonstrated that ring-based protection strategies offer extra advantages in recovery times [27].

Finally, cycle-based protection considers the use of the pre-configured protection cycle (*p*-cycle) [20], which offers fast restoration speed and high efficiency in resource utilization. *p*-cycle technology was first investigating in the context of multicast traffic by Zhang and Zhong [46]. They demonstrated that *p*-cycle based designs are capacity efficient and achieve blocking performance comparable to optimal path pair technique [48,47].

2.3.2. Provisioning static multicast traffic

We now review related work on provisioning static multicast trees. Previous research on resilient multicast routing proposed the use of redundant trees to protect against failures of a node or a link [36,24]. For example, Médard et al. present an algorithm which creates node- or link-redundant trees, by utilizing cyclic paths in the network [33]. A well-known approach to find link-redundant trees is called the *Active Path First (APF)* [18]. APF works as follows. First, construct a minimum-cost steiner tree including one source and all the destinations. Next, remove all the links used in the first tree. Finally, construct a second steiner tree that includes the remaining source and all the destinations, based on the remaining topology. Naturally, the two trees themselves are link-diverse.

Unlike the APF approach, in this paper, we allow the two multicast trees to overlap. This strategy allows us to aggressively minimize the routing cost. We also consider SRLG failures and protect against *multiple* simultaneous failures due to a single fiber cut. In our analysis section, we will compare the routing cost of our design with a representative implementation of the APF designs [16,29,18]. An early version of this work appeared in IEEE Global Internet Symposium, 2006 [11]. In this paper, we newly provide a mathematical framework towards provisioning multicast routing paths that is quality-aware and scalable to large-scale networks.

3. Routing optimization framework

In this section, we (a) present our optimization framework to find multicast routing paths, (b) discuss the complexity of the proposed framework, (c) provide path quality-aware extension and (d) develop two heuristic algorithms for large networks.

3.1. Mathematical programming model: baseline ILP model

The goal of our backbone provisioning is to find a routing from sources to destinations such that each destination has at least one viable path to any one of the sources under failure of a node, a link, or an SRLG. We propose a lightweight protection scheme that finds two SRLG-diverse paths connecting each of the two sources to every destination (i.e., each SRLG can be present in at most one of the two paths to each destination).

Our technique is based on Integer Linear Programming (ILP). We introduce the following notations for the parameters.

- G: The network infrastructure.
- V: The set of network nodes in G.

Please cite this article in press as: M. Cha et al., Efficient and scalable provisioning of always-on multicast streaming services, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.07.005

- E: The set of duplex links that connect the nodes in G.
- S: Two source nodes in V.
- D: The set of destination nodes in V.
- B: The set of SRLGs; each SRLG $b \in B$ is a set of links in E, i.e., $b \subset E$.
- $c_{i,j}$: The cost associated with a link $(i,j) \in E$.

Given $G(V,E), S, D, B$ and $c_{i,j}$, our goal is to find SRLG-diverse paths connecting each destination $d \in D$ with the two sources S such that the cost associated with links used in routing is minimized. We assume that locations of D and S are predetermined by the service provider and that sources always have data for multicast.

We introduce the following decision variables to formally formulate the ILP of the multicast provisioning problem.

$Y^s_{i,j} = 1$, *if* link (i,j) is used by any path rooted at source node $s \in S$; 0, *otherwise*.
$X^s_{i,j,d} = 1$, *if* link (i,j) is used by the path from source node $s \in S$ to destination $d \in D$; 0,*otherwise*.
$Z^s_{b,d} = 1$, *if* any link in a path from source $s \in S$ to destination $d \in D$ belongs to an SRLG $b \in B$; 0, *otherwise*.

The *objective function* is formed to minimize the total capital expenditures (CAPEX), which is the sum of the cost of the links used by the two multicast sources:

$$\min \sum_{s \in S} \sum_{(i,j) \in E} Y^s_{i,j} c_{i,j}. \quad (1)$$

The following *logical constraints* ensure that a link *must be* selected (to incur the link cost) when it is included in any path:

$$Y^s_{i,j} \geqslant X^s_{i,j,d} \quad \forall (i,j) \in E,\ \forall s \in S,\ \forall d \in D. \quad (2)$$

The following *multi-commodity flow constraints* ensure the flow conservation at each node which allows each destination to have a flow path from each of the sources:

$$\underset{\{j|(i,j)\in E\}}{m} X^s_{i,j,d} - \sum_{\{j|(j,i)\in E\}} X^s_{j,i,d} = \sigma^s_{i,d} \quad \forall i \in V,\ \forall s \in S,\ \forall d \in D, \quad (3)$$

$$\text{where} \quad \sigma^s_{i,d} = \begin{cases} 1 & \text{if } i = s, \\ -1 & \text{if } i = d, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

The parameters $\sigma^s_{i,d}$ denote the net flow capacity generated, carried, or arriving at node i for a path from source s to destination d. More precisely, $\sigma^s_{i,d}$ has the value of 1 if node i is the source; -1 if node i is the destination (sink); and 0, otherwise.

The following *SRLG-diverse constraints* ensure that, for each destination, an SRLG can be used by the paths from *only* one of the two sources. In other words, these constraints allow that all the paths from the same source can have common SRLGs, while ensuring that the paths from the two sources to each destination are SRLG-diverse

$$Z^s_{b,d} \geqslant X^s_{i,j,d} \quad \forall (i,j) \in b,\ \forall s \in S,\ \forall d \in D,\ \forall b \in B, \quad (5)$$

$$\sum_{s \in S} Z^s_{b,d} \leqslant 1 \quad \forall d \in D,\ \forall b \in B. \quad (6)$$

Finally, the following constraints state that $X^s_{i,j,d}, Y^s_{i,j}$, and $Z^s_{b,d}$ are boolean variables

$$X^s_{i,j,d}, Y^s_{i,j}, Z^s_{b,d} \in \{0,1\} \quad \forall s \in S,\ \forall d \in D,\ \forall (i,j) \in E,\ \forall b \in B. \quad (7)$$

The space complexity of the above ILP formulation is mainly determined by the number of variables. In our case, complexity is $O(|V| \cdot |E| + |V| \cdot |B|)$, since it has $O(2 \cdot |V| \cdot |E|)$ variables for $X^s_{i,j,d}$ and $O(2 \cdot |V| \cdot |B|)$ variables for $Z^s_{b,d}$. The space complexity of ILP in terms of variables should not be compared directly to the time complexity.

3.2. Complexity issues

We show our problem is NP-hard using a reduction from a known NP-hard problem. We define the following decision problem variation of our problem.

Problem 1. Given a mesh network topology $G(V,E)$ with arbitrary SRLGs with two source nodes $s \in S$, a set of destination nodes $d \in D$, and link costs c_{ij}, along with a provisioning cost goal of τ, does there exist a set of feasible SRLG-diverse paths from the two sources s to each destination d such that the total cost of those paths is no more than τ?

Theorem 1. *Problem 1 is NP-complete.*

Proof. First, Problem 1 clearly belongs to NP, as a solution can be encoded as two different $O(n)$-length paths for each destination, and the cost summation that at least one path survives from a single failure can be evaluated in $O(n)$ time. Next, we note that Problem 1 is a generalization of the problem of finding SRLG-diverse paths between a single source and a single destination in a given graph (a.k.a. SRG-Diverse Routing) proved by Ellinas et al. [17]. As the SRG-Diverse Routing problem has been shown to be NP-complete, we execute the following transformation from any arbitrary solution of SRG-Diverse Routing to Problem 1, such that the SRG-Diverse Routing instance is a yes-instance if and only if the transformed Problem 1 instance is a yes-instance. Let us add two nodes ($|S|$) in the graph and two links connecting each of the two new nodes with the source with the costs of 0. Assume that the two new links do not share any SRLGs. We then add $|D|$ nodes and $2 * |D|$ edges. Each of these nodes is connected to the destination node by 2 edges that do not share any SRLGs. The new SRG-Diverse Routing problem is equivalent to Problem 1. Fig. 3 illustrates the case for $|D| = 3$. □

As a simple consequence of Theorem 1, we have the following corollary.

Corollary 1. *The multicast provisioning problem with SRLG constraints is strongly NP-hard.*

Next, we provide two important extensions for our optimization framework. The first extension incorporates the need to control the quality of the provisioned paths. The second extension is the efficient decomposition algorithms that address scalability challenge for large networks.

Please cite this article in press as: M. Cha et al., Efficient and scalable provisioning of always-on multicast streaming services, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.07.005

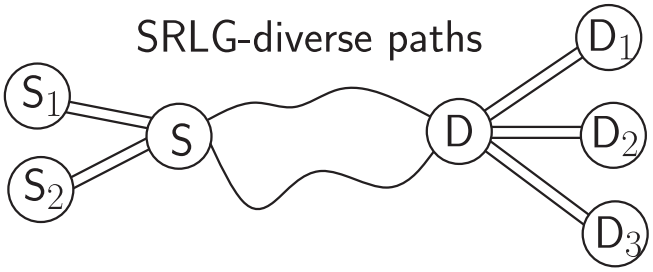


Fig. 3. A helpful example where two new nodes S_1, S_2 are attached to the existing source node S, and three new nodes D_1, D_2, D_3 are attached to the existing destination node D.

3.3. Path quality-aware realistic considerations

The baseline ILP allows a service provider to minimize the total cost of provisioning, but does not consider the quality of the paths. Here, we provide extensions to the baseline ILP that allows service providers to control quality of the individual paths (e.g., latency, hop counts, per-path bound of SRLGs). We use latency as an example metric to describe our extension, which is useful in real-time streaming with tight latency requirements such as stock ticker applications [9]. To guarantee real-time delivery of traffic, we impose a constraint on latency in the following two ways: (a) setting an upper bound on the latency for *all* paths and (b) for *any one* of the two paths per destination. The latter can be used to guarantee low-latency primary paths, while allowing for lower quality backup paths. A variant optimization goal would be to focus on minimizing the maximum delay, which can be incorporated as a secondary objective in the objective function. In this paper, however, we set an upper bound on the delay to comply with the service layer agreement (SLA) for real-time applications.

As a first extension of the ILP model, we impose an upper bound on the latency of both paths for every destination. To achieve this, let us denote the delay of each link $(i,j) \in E$ as $d(i,j)$, and denote the latency threshold by θ_{all}. We introduce the following constraints to the baseline ILP to ensure that the sum of distances for every path is within the threshold[1]

$$\sum_{(i,j)\in E} X^s_{i,j,d} d(i,j) \leqslant \theta_{\text{all}} \quad \forall s \in S,\ \forall d \in D. \tag{8}$$

For the latter case of having a low-latency primary path and lower quality backup, for each destination, we impose a latency upper bound, θ_{shorter}, on only one of the two paths. To achieve this, we introduce a new boolean variable P^s_d, where $P^s_d = 1$ if the path from source s to destination d violates the latency constraint; and 0 otherwise. Eq. (9) states that both paths cannot violate the delay constraint at the same time. We then modify Eq. (8) and substitute θ_{all} with $(\theta_{\text{shorter}} + P^s_d \alpha)$ as in Eq. (10). By adjusting θ_{shorter}, we can control the latency limit of a solution. The value of α represents the slack delay between the two paths. Since the goal of the second scheme is to impose a latency upper bound on one path, we usually set α to a large value, which could be the diameter of the network, or the maximum shortest distance between any two nodes within the network. The second scheme is a generalization of the first scheme

$$\sum_{s\in S} P^s_d \leqslant 1 \quad \forall d \in D, \tag{9}$$

$$\sum_{(i,j)\in E} X^s_{i,j,d} d(i,j) \leqslant \theta_{\text{shorter}} + P^s_d \alpha \quad \forall s \in S,\ \forall d \in D. \tag{10}$$

This framework can be used to incorporate other path quality metrics.

3.4. Scalable solution approaches: heuristic algorithms

Running ILP exact algorithms becomes extremely time consuming for large networks. Here, we present scalable heuristic algorithms for the multicast provisioning problem. Our approach is based on a decomposition (or divide-and-conquer) technique [13], where a large-scale provisioning problem is divided into smaller subproblems. The proposed algorithms are very useful when the ILP exact algorithm fails to find a feasible solution within resource limits (computational time or memory). Unlike existing algorithms, our algorithms overcome trap scenarios and can be parallelized easily. In the following, we present two algorithms: Greedy Local (GL) and Improved Greedy Local (IGL).

3.4.1. Greedy Local (GL)

The main idea of the *Greedy Local (GL)* algorithm is to divide the problem into $|D|$ subproblems, each containing the same set of nodes, links, SRLGs, and the two sources, but a single destination. The procedure of the GL algorithm is as follows:

(1) For each destination in D, run the baseline ILP and find two SRLG-diverse path pairs.
(2) Consolidate the paths found from the subgraphs as a solution.

We provide a generic pseudo code for *GL* algorithm in Fig. 4. The algorithm starts with initialization in line 1. Lines 2 and 3 describe the procedure for solving the baseline ILP for each destination d. Lines 4–6 examine if the ILP has produced a solution path pair $\left(P^1_d, P^2_d\right)$ from the two

[1] One might think that we can incorporate delay of a link into the cost and thus cast the problem as the basic ILP of the previous section. However there is a glitch. The basic ILP minimizes the overall cost of the design, but does not guarantee anything about the per-path cost.

sources s^1, s^2 to the destination d, and stores the result in $\mathcal{P}$ and D^*. Line 7 terminates the procedure.

Solving an individual two-source, one-destination problem is still NP-hard. However, the GL algorithm can drastically reduce the number of decision variables by a factor of $|D|$, which make the subproblems much easier to solve. Although one needs to solve $|D|$ subproblems, the computation is very efficient even for large networks because the term $|D|$ grows exponentially in space in the baseline ILP problem. The space complexity for each subproblem is $O(|E| + |B|)$. More importantly, this algorithm can be parallelized.

Procedure GL
1 $\mathcal{P} \leftarrow \emptyset$ and $D^* \leftarrow \emptyset$;
2 **foreach** $d \in D$
3 Solve ILP for a subgraph $S \times \{d\}$;
4 **if** solution paths found (p_d^1, p_d^2)
5 $\mathcal{P} \leftarrow \mathcal{P} \cup \{p_d^1, p_d^2\}$;
6 $D^* \leftarrow D^* \cup \{d\}$;
7 **return** $\mathcal{P}$ and D^*;

Fig. 4. Pseudo code of *Greedy Local (GL)* algorithm.

Two multicast examples with two sources and three destinations in Fig. 5a and b illustrate the differences between the baseline ILP and GL algorithm. In Fig. 5a, the same multicast traffic received at $d1$ from $s1$ is further delivered to $d2$ and $d3$. Likewise, the traffic received at $d3$ from $s2$ is also shared by $d2$ and $d1$. On the other hand, the GL algorithm in Fig. 5b leads to much less sharing of traffic, as SRLG-diverse path pairs for each destination are chosen without considering the paths chosen for the other destinations. As a result, the total design cost increases from 8 to 12.

One advantage of the GL algorithm is its capability to quickly check if there exists any feasible solution for the multicast provisioning problem. It is easy to prove that a feasible solution exists for the base ILP model if and only if there exists a feasible solution to all of the $|D|$ subgraph problems. At the end of this section, we provide a formal proof of this proposition. If SRLG-diverse paths do not exist for a given destination (due to pathological topology), we may exclude such destinations for multicast deployment or allow SRLG-overlapping paths. We give a new ILP model in the Appendix A that jointly minimizes the cost of design and the number of SRLG-overlapping links.

3.4.2. Improved Greedy Local (IGL)

We propose an enhancement of the GL algorithm, which we call *Improved Greedy Local (IGL)* algorithm, from

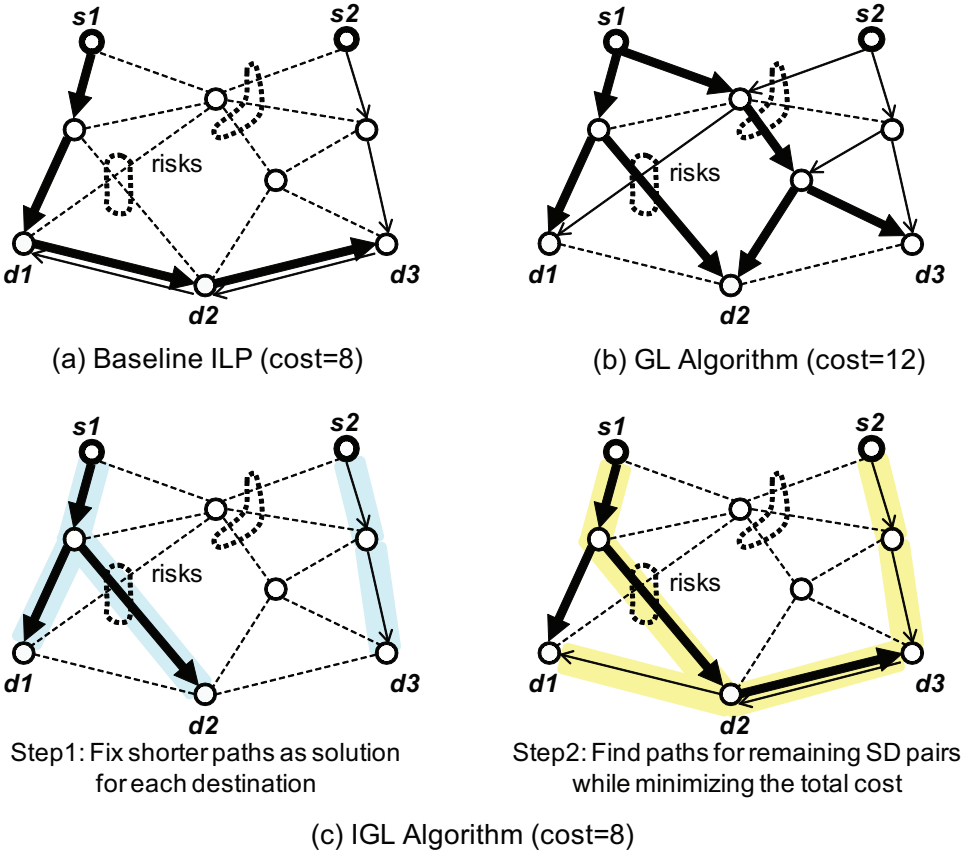


Fig. 5. Illustration of algorithms. Depicted solution is one instance of a solution.

Please cite this article in press as: M. Cha et al., Efficient and scalable provisioning of always-on multicast streaming services, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.07.005

the observation that per-destination path selection in *GL* is oblivious to each other. The IGL algorithm takes paths explored in the GL algorithm as input and runs in two steps:

(1) Given SRLG-diverse path pairs found by the GL algorithm, fix the shorter path for each destination as the final solution.
(2) Given a partial solution where each destination is connected to only one of the sources, find the second set of paths for all destinations jointly by solving the baseline ILP.

Overall, the IGL algorithm requires running the GL algorithm once, then solving the baseline ILP where half of the paths are already solved.

In the first step, we can fix the shorter path for each destination by setting the upper and lower bounds of $X^s_{i,j,d}$ variables to 1 for the links used in the shorter paths. This technique can be viewed as adding more constraints to the ILP and is also referred to as *integer cut constraints*. By utilizing SRLG-diverse path pairs found by *GL*, the first step guarantees that there exists at least one SRLG-diverse path counterpart for every destination. In the second step, these path counterparts are searched jointly to aggressively minimize the total cost. Fig. 5c illustrates the procedure for the IGL algorithm. After the first step, shorter paths for each destination are fixed as a solution; $d1$ and $d2$ are connected to $s1$, and $d3$ to $s2$. At second step, the baseline ILP connects destinations with the remaining source, while minimizing the total cost. For instance, the path from $s2$ to $d1$ is chosen such that the cost of multicast tree for $s2$ is minimized.

We provide a generic pseudo code for the IGL algorithm in Fig. 6. After initialization (line 1), we solve the GL algorithm and obtain the solution paths $\mathscr{P}_{GL}$ and the set of valid destinations with solution D_{GL} (line 2). Lines 3 and 4 describe the procedure for setting the integer cut constraint: for each destination d, we add a shorter path p_d^{short} from the solution set $\mathscr{P}_{GL}$ to a constraint $\mathscr{C}$. Then, we solve the baseline ILP with the fixed set of paths $\mathscr{C}$ in line 5. Lines 6–9 examine if the ILP has produced a solution for each destination d and store the result in $\mathscr{P}$ and D^*. Line 10 terminates the procedure.

Procedure IGL
1 $\mathcal{P} \leftarrow \emptyset$, $D^* \leftarrow \emptyset$ and $\mathcal{C} \leftarrow \emptyset$;
2 Run procedure GL and obtain $(\mathcal{P}_{GL}, D_{GL})$;
3 **foreach** $d \in D_{GL}$
4 $\quad \mathcal{C} \leftarrow \mathcal{C} \cup \{p_d^{short} \mid (p_d^{short}, p_d^{long}) \in \mathcal{P}_{GL}, |p_d^{short}| \leq |p_d^{long}|\}$;
5 Solve ILP with $S \times D_{GL}$ and $\mathcal{C}$;
6 **foreach** $d \in D_{GL}$
7 $\quad$ **if** solution paths found (p_d^{short}, p_d^{new})
8 $\quad\quad \mathcal{P} \leftarrow \mathcal{P} \cup \{p_d^{short}, p_d^{new}\}$;
9 $\quad\quad D^* \leftarrow D^* \cup \{d\}$;
10 **return** $\mathcal{P}$ and D^*;

Fig. 6. Pseudo code of *Improved Greedy Local (IGL)* algorithm.

The upper bound design cost provided by the *IGL* algorithm is at least as tight as the worst-case cost provided by *GL* algorithm. Compared to the baseline ILP problem, the number of integer variables is reduced by $|D|$ in the first step, and by half in the second step in the *IGL* algorithm.

Below, we show that there exists a feasible solution for the base ILP model, if and only if there exists a feasible solution to all of the $|D|$ subgraph problems.

Proposition 1. *There exists a feasible solution to the baseline ILP model, if and only if there exists a feasible solution to all of the $|D|$ subgraph problems in the GL and IGL algorithms.*

Proof. (Necessity condition.) We can prove this by contradiction. Assume that there exists a feasible solution to all of the $|D|$ subgraph problems but the baseline ILP model is infeasible. If all $|D|$ subgraph problems are feasible, there must exist two SRLG-diverse paths from the two sources to each of the D destinations. This would contradict the assumption that the baseline ILP model is infeasible.

(Sufficiency condition.) We likewise prove this by contradiction. Assume that there exists a feasible solution to the baseline ILP model but one (or more) subgraph problem is infeasible. If there exists a feasible solution to the baseline ILP model, there must exist two multicast trees with SRLG-diverse paths (subtree) to each of the $|D|$ destinations. Thus, all of the $|D|$ subgraph problems must be feasible. □

4. Empirical analysis

In this section, we first evaluate the performance of the baseline ILP model and the path performance-aware extensions through simulation experiments. We then compare the heuristic algorithms GL and IGL. The experiments show that (a) the baseline ILP provides robustness against SRLG failures at a small additional cost; (b) path performance-aware extensions allow us to find good operational points in terms of cost; and (c) the proposed heuristic algorithms overcome traps and scales well as the problem size increases. Our analysis is based on four realistic networks coupled with their SRLG information. We use the GAMS tool [19] to model the ILP formulation and use CPLEX [14] to solve all programs on a desktop with a 2.8 GHz Intel Pentium 4 processor and 1 GB memory. Below we first describe our assumptions about the network infrastructure configuration.

4.1. Network infrastructure settings

Table 1 summarizes the network topologies used in the evaluation. *Net1* and *Net2* are operational tier-1 backbones located across the US, and *US-NET* and *Italian network* are the topologies published in [40]. Each topology is listed with its name and the number of nodes and bi-directional links. When there are parallel links, we make the graph simple (*i.e.*, a graph with at most a single edge between any pair of nodes) by adding new nodes and links. For Net1 and Net2, we use the information about real fiber span to identify SRLGs as follows. First, we associate a un-

Please cite this article in press as: M. Cha et al., Efficient and scalable provisioning of always-on multicast streaming services, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.07.005

Table 1
Summary of dataset.

Topology	# nodes	# links	# SRLGs	# destinations	Run-time (s)
Net1	178	392	212	33	800
Net2	38	68	41	10	2
US-Net	24	43	32	10	1
Italian	21	36	28	10	1

ique SRLG with each link, interface, and fiber spans that comprise the link. Then, we remove those SRLGs which are strict subsets of other SRLGs. To identify SRLGs for US-NET and Italian network, we use the risk relationships amongst the links as given in [40].

We determined the locations of sources and destinations as follows. For Net1 and Net2, we considered 40 largest cities in the US as potential service recipients and identified the nodes in the topology that are located geographically closest to those potential recipients. These nodes were used as destinations. Some of the top-40 cities were mapped to a single node in the Net1 and Net2 topologies. In this way, we identified 33 and 10 destinations in Net1 and Net2, respectively. The two locations of the sources were chosen strategically by the tier-1 ISP that owns Net1 and Net2; one is located in the West coast of the US and the other in the East coast. For US-NET and Italian network, we randomly selected 10 destinations and two sources. In all the cases, sources and destinations formed disjoint sets.

Now we describe the link cost function, $c_{i,j}$, used in our evaluation. For US-NET and Italian networks, we use the link costs specified in [40]. For Net1 and Net2, the cost of each individual link is approximated by the leasing cost for using the link for service. The leasing cost is calculated as the sum of costs associated with the ports at the two ends of the link and a cost that is proportional to the distance of the link. The values for these port-related and route-mile related components vary depending on the technology being used in the network (e.g., port and link type, link capacity, vendor), as well as the bandwidth used for each link. We use the actual cost values based on the price listing for optical cross-connect ports and unprotected OC-48 links. This information was provided by the tier-1 ISP from which we also obtained the Net1 and Net2 topologies.

The bandwidth required for the links depends on the (anticipated) traffic load of the application. In our case, we assume the application is IPTV and use 1 Gbps as the bandwidth requirement [12]. 1 Gbps typically equates to 200 TV channels or IPTV multicast groups when we assume each channel is 5 Mbps.

4.2. Evaluation of the ILP model

The ILP model in the previous section can be directly implemented and solved via a commercial ILP solver such as CPLEX [14]. We call this approach *Exact*.

We first evaluate *Exact* by comparing it with two other designs with a relaxed survivability constraint. The first one, which we will refer to as Src-Div, is derived by simply constructing two minimum cost multicast trees from each of the sources independently. In the second design, which we refer to as Link-Div, two multicast trees are constructed from each of the sources simultaneously, while the routing paths are constrained to be link-diverse and the total cost is minimized. The ILP formulation resembles that of our SRLG-diverse paths, except that Eqs. (5) and (6) are replaced by,

$$\sum_{s\in S} X^s_{i,j,d} \leqslant 1 \quad \forall (i,j) \in E,\ \forall d \in D. \tag{11}$$

To evaluate the risk due to single SRLG failures, we report the number of critical SRLGs whose failure disconnects one or more receiver(s) from both sources in each design (Table 2). We also report the number of unreliable receivers that are subject to service interruption under SRLG failure. Recall that there are a total of 30 receivers in Net1 and 10 receivers in the other networks, and 212 distinct SRLGs in Net1, 41 in Net2, and 32 and 28 in US-NET and Italian network. As expected, the number of critical SRLGs and the number of unreliable receivers are zero for both SRLG-Div and Heuristic. Src-Div and Link-Div, however, do not have this property. For example in the Src-Div design in Net1, there are 36 critical SRLGs whose failure will disconnect at least one receiver from both sources. In Link-Div of the Italian network, four unreliable receivers are subject to the risk of disconnection under certain critical SRLG failures.

4.3. Guarantee of individual path quality

Finally, we apply the new extensions that incorporate performance metrics for the selected paths (see Section 3.3). Due to lack of space, we only show the result for Net2. However, similar results were obtained for the other topologies. Fig. 7a shows the relative cost of the solutions when we set an upper bound θ_{all} on the delay of all paths. Fig. 7b shows that when we have an upper bound on only one of the two paths for each destination, θ_{shorter}. The *y*-axis is normalized such that the solution without any delay upper bound is 100%, for each of the figures. The *x*-axis shows the upper bound on delay in milliseconds. We assume 1 ms of propagation delay for a 100 mile distance as suggested in [22,39]; however, the effective propagation delay can be larger when one takes intermediate switching and routing equipment into account.

Both Fig. 7a and b show similar trade-offs between the cost and path quality. When the delay limit is below a certain value (less than 25 ms in Fig. 7a), it is infeasible to find any solution. We are able to find the first feasible solution

Table 2
Risk analysis across designs upon SRLG failure.

Topology	SRLG-Div, Heuristic		Src-Div		Link-Div	
	u.D	c.risk	u.D	c.risk	u.D	c.risk
Net1	0	0	27	36	19	17
Net2	0	0	6	11	1	1
US-NET	0	0	4	7	1	1
Italian	0	0	7	7	4	3

Notation: u.D denotes unreliable receivers; and c. risk denotes number of critical SRLGs of risk.

Please cite this article in press as: M. Cha et al., Efficient and scalable provisioning of always-on multicast streaming services, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.07.005

with some cost increase (to 117%) compared to the no-constraint case. More economical solutions become available, as we relax the constraint and accept paths with higher delays as solutions. The two rightmost data points in the figure represent part of the curve where we have reached the optimal cost, and increasing the upper bound on delay becomes meaningless. It is interesting that the curves have a step-like shape. This shows the *step-wise inverse relationship* between cost and the delay limit. This is a very useful result because an uphill step boundary in the trade-off curve provides a good point for service providers to select in terms of cost-quality trade-off. We point out that our results conform to a recent theoretical study on the trade-off curves for QoS routing [34].

4.4. Performance evaluation

We first evaluate the performance of the proposed ILP-based heuristic algorithms (GL and IGL). These two algorithms break one ILP optimization task into several sub-problems, therefore the results are not strictly optimal. Yet, these algorithms are scalable and can be solved in a large problem space. Therefore, we compare GL and IGL against *Exact* and Active Path First (APF) and analyze the trade-off relationship between the cost increase and scalability. For this comparison, we limit the problem size so that all systems can obtain a solution.

4.4.1. Cost comparison

In this cost comparison, we were not able to find any feasible solution for APF using a naive approach based on a single steiner tree from each of the sources for all four topologies. To overcome this, we use a variant of the trap avoidance scheme in [16]. Instead of exhaustively searching for k minimum-cost steiner trees, we construct k random trees. We set a lower bound on the acceptable total cost. We then try to find a steiner tree whose cost is no more than the bound. We then gradually increase the lower bound until we find k trees from a given source. We try out 50 trees from each of the sources and check if a secondary tree can be constructed.

Fig. 8 shows the relative costs across all designs. The y-axis is normalized such that the cost of *Exact* is 100% for each topology. Note that all four designs can survive in the event of any single SRLG failure. We make the following observations. First, for the Net1 topology, the

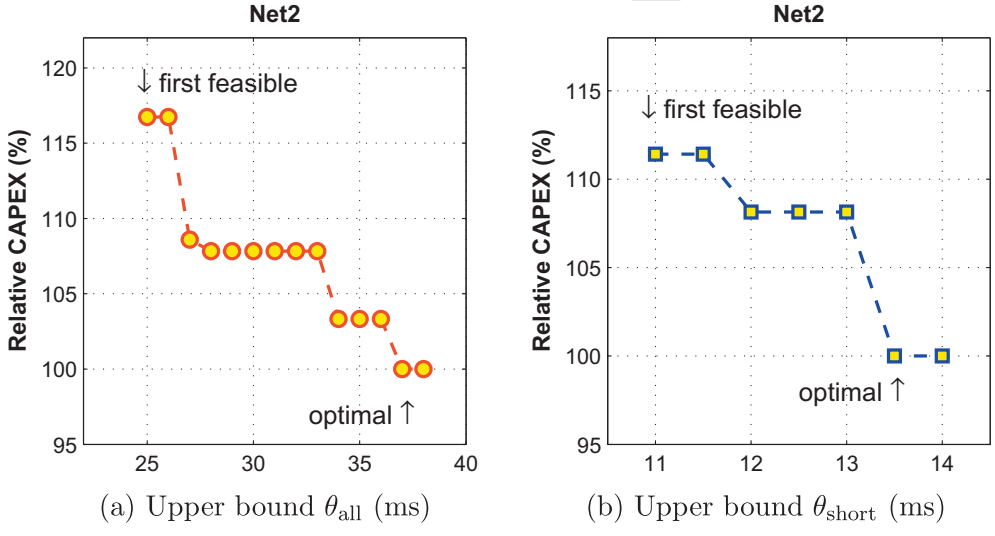


Fig. 7. Trade-off curves between CAPEX and delay limit.

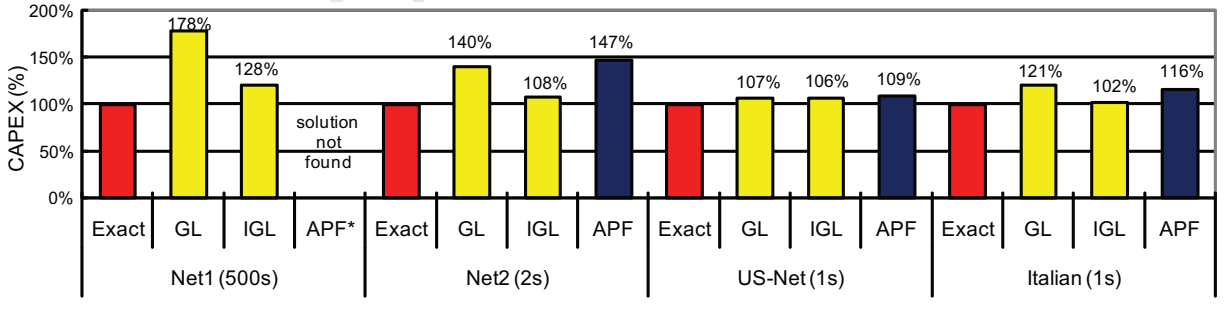


Fig. 8. CAPEX comparison for four network topologies across various provisioning algorithms: ILP Exact Approach (Exact), Greedy Local (GL), Improved Greedy Local (IGL), and Active Path First (APF). The CPU times required for solving *Exact* are given next to the topology names.

Please cite this article in press as: M. Cha et al., Efficient and scalable provisioning of always-on multicast streaming services, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.07.005

Table 3
The number of distinct intermediate nodes (V^*), links (E^*), SRLGs (B^*), and total route-mile ($Dist^*$) used in designs.

Topology	Algorithm	$\|V^*\|$	$\|E^*\|$	$\|B^*\|$ (overlap)	$Dist^*$
Net1	*Exact*	51	86	83 (25)	11,844 (100%)
	GL	60	103	107 (34)	27,310 (231%)
	IGL	56	94	104 (23)	14,959 (126%)
	APF	Solution not found			
Net2	*Exact*	14	23	21 (11)	7657 (100%)
	GL	16	24	28 (6)	11752 (153%)
	IGL	15	24	23 (9)	8387 (110%)
	APF	20	31	33 (0)	11990 (153%)

In $|B^*|$ field, *overlap* shows the distinct number of SRLGs that appear in routing paths of both of the sources.

existing APF approach fails to find any solution. This shows the strength of the proposed approaches in trap avoidance and finding solutions even for very large networks. Second, *Exact* provides the lowest cost solution. This demonstrates the advantage of the *Exact* approach of jointly considering two trees from both sources as opposed to constructing one tree at a time (as in APF). The advantage in terms of design cost reduction is nearly 30% ($\approx 1-100/147$) for the Net2 topology. Third, comparing the two scalable heuristic algorithms, *IGL* provides a more economical solution than *GL* (with a cost difference between 2% and 28%).

4.4.2. Solution size comparison

In Table 3, we list the number of distinct intermediate nodes, links, and SRLGs, as well as the total route-mile used in designs for Net1 and Net2. A small number of intermediate nodes reflects a compact design, while a small number of links and small total route-mile directly reflect savings on the port and distance related costs respectively. Table 3 shows that *Exact* performs best for all the size related metrics, and *IGL* outperform *GL* and APF. In Net1, we observe that the solution size ($|V^*|$ and $|E^*|$) of *GL* is slightly larger than the solution size of a more expensive design, *IGL*. A typical IP router port cost is approximately three times higher than the cost of a same capacity optical port. Therefore, if each node is a router, it will be more desirable to use fewer links than to obtain a smaller route-mile.

Interestingly, the number of intermediate SRLGs is *smaller* for more economical solutions. This is a highly desired result since fewer SRLGs mean less exposure to

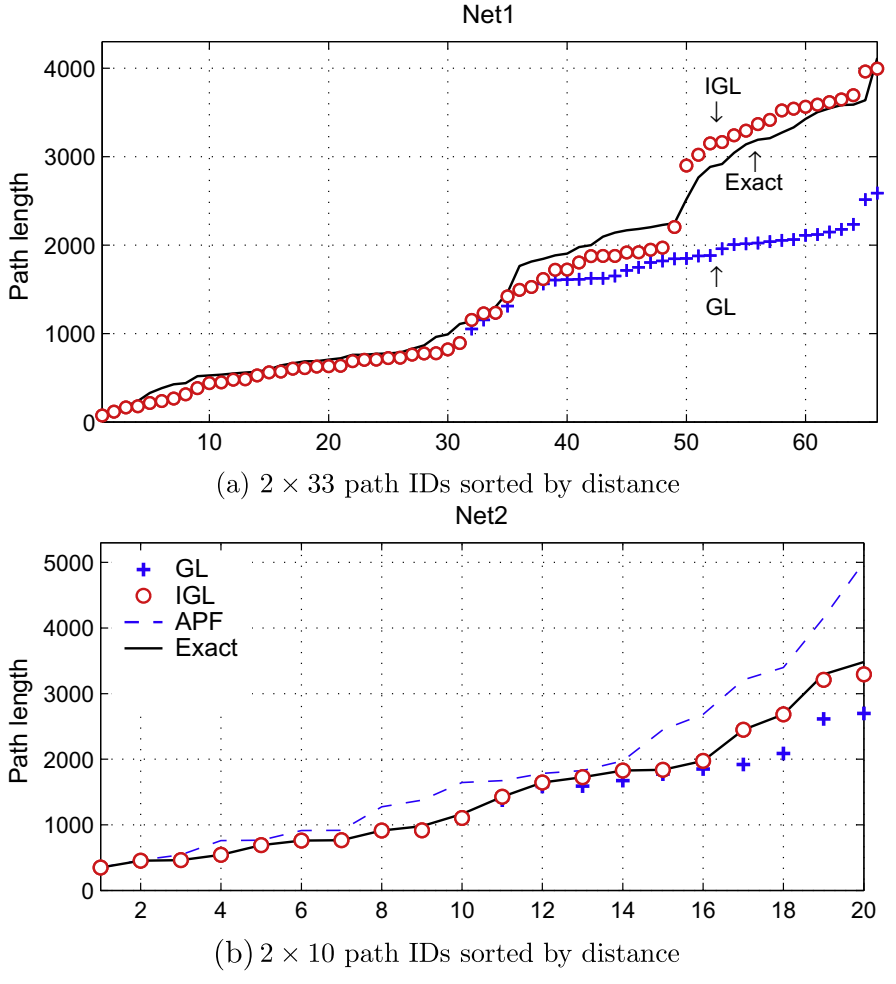
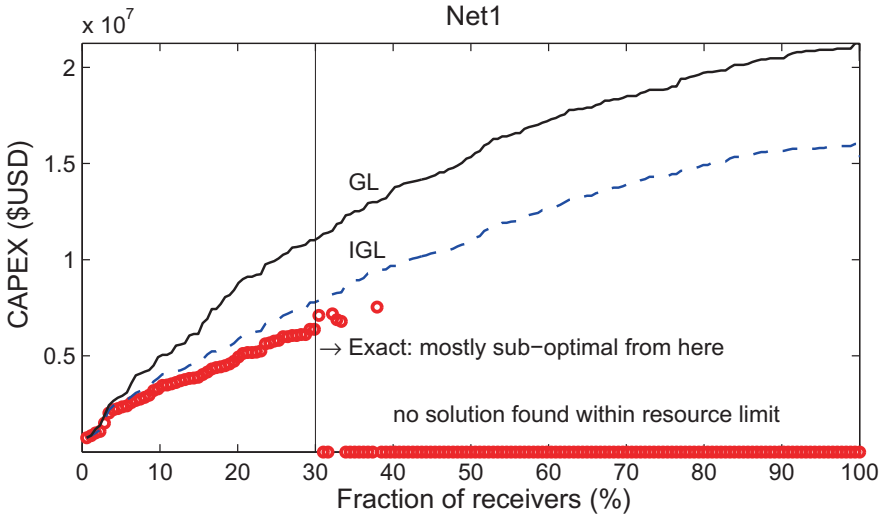
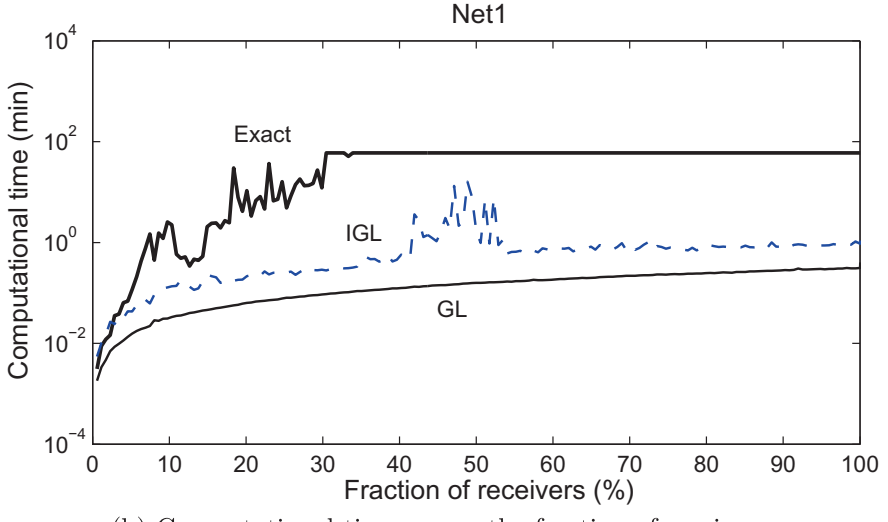


Fig. 9. Path length distribution of all source–destination pairs.

Please cite this article in press as: M. Cha et al., Efficient and scalable provisioning of always-on multicast streaming services, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.07.005



(a) CAPEX versus the fraction of receivers



(b) Computational time versus the fraction of receivers

Fig. 10. Scalability analysis of the solution approaches.

potential risks, provided the probability of failure is identical amongst SRLGs.

4.4.3. Path length comparison

While the total route-mile (Dist*) in Table 3 gives us a good estimate for the overall size of the provisioned networks, it does not tell us much about the route-mile of the individual paths. In this section, we examine this quantity for all the algorithms. Fig. 9 shows the distribution of path lengths, $\sum_{(i,j)\in E} X^s_{i,j,d} d(i,j)$, for all source–destination pairs, (s,d). For example, there are 2×33 paths for Net1. Paths are sorted on the x-axis according to their lengths. The plot shows that *GL* consistently provides the lowest path length. This is because path selection is locally optimized for each destination. We observe that *IGL* deviates from *GL* for $x > 30$ for Net1 and $x > 14$ for Net2. Because *IGL* shares half of the paths with *GL*, the other half are chosen to minimize the total cost. In fact, this is what allows IGL to reduce costs. We observed similar results for the US-Net and Italian networks.

4.5. Scalability analysis

We evaluate the scalability of the proposed algorithms by examining how computation time and total cost vary as we increase the problem size. We show the result for only Net1, since for other topologies, *Exact* produces solutions in less than a minute. First, we apply *GL* to all potential destinations to determine the "valid" destinations, D^*, that can have SRLG-diverse paths to the two sources. We find that all but two destinations belong to D^*. After determining D^*, we randomly pick one node each time and add it to a receiver group, thereby gradually increasing the number of receivers. For each receiver group, we apply all the provisioning algorithms and compute the total cost of their solutions and CPU time required to determine

Please cite this article in press as: M. Cha et al., Efficient and scalable provisioning of always-on multicast streaming services, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.07.005

these solutions. Fig. 10 shows two plots, one depicting the total cost of solutions and the other showing the CPU time required to solve each problem. The *x*-axis is normalized from 0% to 100%, such that 100% represents the case when the number of receivers is $|D^*|$. We do not plot a line for APF, since it does not produce solutions for all the cases.

Fig. 10a shows that the cost increases as the number of receivers increases for *Exact*, *GL*, and *IGL*. An important observation is that *Exact* does not scale well, compared to other algorithms that provide solutions for all $|D^*|$ cases. As we increase the number of receivers, we reach a point where the ILP solver reports a sub-optimal solution (i.e., the best integer solution found by CPLEX within its means).[2] While sub-optimal solutions are still useful because they are feasible and satisfy SRLG constraints, *Exact* may not produce any solution due to CPU and/or memory limitations as the problem size increases. In these simulations, we set the limit on CPU time to 3600 s such that the ILP solver halts when it finds an optimal integer solution or when it reaches the time limit. As can be seen from the figure, *Exact* cannot find a solution for the fraction of receivers was greater than (cost of zero in the plot means no solution). This clearly demonstrates the efficacy of our efficient and scalable algorithms.

Fig. 10b plots the computation time versus the normalized number of receivers. The *y*-axis is on a log scale and the unit is minutes. Here, the CPU time for *Exact* increases significantly as the number of receiver grows, which points to the increasing difficulty of solving the NP-hard problems. On the other hand, *GL* and *IGL* scale much better as the problem size increases. A detailed look at these algorithms show that CPU time increases roughly linearly with the number of receivers. Note that in this paper we have considered scalability only in terms of the number of receivers. One can perform similar analyses in terms of network size, its topological structure, the number of sources, and the number of links per SRLG. We leave this analysis as part of our future work.

5. Concluding remarks

In this paper, we investigated the off-line multicast provisioning problem for always-on streaming services in the backbone network. In particular, we focused on reliable multicast in the optical layer, which guarantees available bandwidth at the time of failure and fast fault recovery. We have presented two decomposition algorithms based on an ILP formulation and important performance-aware extensions to the ILP formulation. The decomposition algorithms divided the multicast provisioning problem into small subproblems, where each subproblem was solved efficiently while overcoming trap scenarios. In addition, the proposed extensions took into account the quality of the provisioned paths such as latency limits, the number of intermediate nodes and links, and per-path upper bound of SRLGs. In general, our numerical results showed that the decomposition algorithms produce economical solutions, overcome trap scenarios, and scalable to handle large networks (with a few hundred nodes, fiber spans, and multicast receivers). The main contributions and findings are as follows:

(1) We presented a new ILP model to design efficient multicast infrastructure for modern streaming services. Our model is robust to SRLG failures and outperforms the state-of-the-art, reducing costs by 30%.
(2) We extended the model to incorporate path performance requirements (e.g., shorter latency) at increased cost.
(3) We developed two scalable heuristic algorithms for large networks and demonstrated that these algorithms overcome trap scenarios, unlike existing approaches.
(4) We evaluated the algorithms using real network topologies.

There are a number of directions that we would like to extend our work in the future. First, a natural extension of our work is to consider different provisioning scenarios with multiple sources. In terrestrial multicast services, adding more sources provides higher redundancy. However, adding an extra source facility increases the design cost significantly. We are also interested in understanding the trade-off between the number of additional sources and the flexibility in design that the additional sources may provide. Another interesting investigation is to find the optimal number of sources, and the best locations for these sources. In addition, we are interested in a scenario when no SRLG-diverse paths can be found due to topological constraints (i.e., destinations in $D \setminus D^*$). In such a case, it might still be beneficial to support maximally SRLG-disjoint paths. We provide one possible method for finding such paths in the Appendix A. Third, we would like to develop an online provisioning approach, where the provisioned path can quickly adapt when a random link failure happens. We hope our work expedites the development of always-on multicast streaming applications by saving costs and improving quality.

Appendix A

In real life, new demands for IPTV services arise frequently (*e.g.*, additional fiber links, business locations, increased bandwidth), which causes the network topology of IPTV services to change over time. In such cases, the service providers need to incorporate the dynamics of a given infrastructure in their routing strategies. As the number of links and destination nodes increases, the current backbone network topology might not be able to support the SRLG-diverse constraint.

In this appendix, we provide an adaptive IP model that can handle this situation, which jointly minimizes the cost of design and the number of SRLG-overlapping links. Basically, the model will give an alternative solution that has a minimum number of SRLG-overlapping links with respect to the communication cost and the overlapping cost (the cost of resiliency reduction). The basic idea of the adaptive

[2] ILP solvers typically utilize state-of-the-art techniques to find the upper bound and prove the optimality of the solution. Therefore, we are able to verify if a solution instance is optimal or sub-optimal.

Please cite this article in press as: M. Cha et al., Efficient and scalable provisioning of always-on multicast streaming services, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.07.005

IP model is as follows. First, we relax the SRLG-constraint by introducing a surplus 0–1 variable W_d^b in Eq. (A.5), where $W_d^b = 1$ if a SRLG $b \in B$ is used in both paths to the destination $d \in D$ (the paths are no longer SRLG-diverse), and $W_d^s = 0$ otherwise. We penalize the number of overlapping SRLGs over all destinations by a weight parameter λ, which is incorporated into the objective function. The proposed adaptive IP model is given by

$$\min \sum_{s\in S} \sum_{(i,j)\in E} Y_{i,j}^s c_{i,j} + \lambda \sum_{d\in D} \sum_{b\in B} W_d^b \tag{A.1}$$

such that

$$Y_{i,j}^s \geqslant X_{i,j,d}^s \quad \forall (i,j) \in E,\ \forall s \in S,\ \forall d \in D, \tag{A.2}$$

$$\sum_{\{j|(i,j)\in E\}} X_{i,j,d}^s - \sum_{\{j|(j,i)\in E\}} X_{j,i,d}^s = \sigma_{i,d}^s \quad \forall i \in V,\ \forall s \in S,\ \forall d \in D, \tag{A.3}$$

$$Z_{b,d}^s \geqslant X_{i,j,d}^s \quad \forall (i,j) \in b,\ \forall s \in S,\ \forall d \in D,\ \forall b \in B, \tag{A.4}$$

$$\sum_{s\in S} Z_{b,d}^s \leqslant 1 + W_d^b \quad \forall d \in D,\ \forall b \in B, \tag{A.5}$$

$$X_{i,j,d}^s, Y_{i,j}^s, Z_{b,d}^s, W_d^b \in \{0,1\} \quad \forall s \in S,\ \forall d \in D,\ \forall (i,j) \in E,\ \forall b \in B, \tag{A.6}$$

where $$\sigma_{i,d}^s = \begin{cases} 1 & \text{if } i = s, \\ -1 & \text{if } i = d, \\ 0 & \text{otherwise.} \end{cases} \tag{A.7}$$

References

[1] High-Energy Physics Team Sets Data-Transfer World Records, Cern Courier, January 2009, <http://cerncourier.com/cws/article/cern/37317>.
[2] J. Abley, K. Lindgvist, RFC 4786: Operation of Anycast Services, 2006.
[3] M. Alicherry, R. Bhatia, I. Saniee, S. Sengupta, SRLG-diversity aware protection routing in optical mesh networks, in: Proceedings of the National Fiber Optic Engineers Conference, March 2005.
[4] R. Andersen, F. Chung, A. Sen, G. Xue, On disjoint path pairs with wavelength continuity constraint in WDM networks, in: Proceedings of the IEEE INFOCOM, March 2004.
[5] H. Ballani, P. Francis, Towards a global IP Anycast service, in: Proceedings of the ACM SIGCOMM, August 2005.
[6] R. Bhandari, Survivable Networks: Algorithms for Diverse Routing, Kluwer Academic Publishers, 1999.
[7] K.P. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, Y. Minsky, Bimodal multicast, ACM Transactions on Computer System 17 (2) (1999) 41–88.
[8] C. Boutremans, G. Iannaccone, C. Diot, Impact of link failures on VoIP performance, in: Proceedings of the ACM NOSSDAV, May 2002.
[9] K.P. Briman, A review of experiences with reliable multicast, Software – Practice and Experience 29 (9) (1999) 741–774.
[10] J. Cao, L. Guo, H. Yu, L. Li, A novel shared segment protection algorithm for multicast sessions in mesh WDM networks, ETRI Journal 28 (3) (2006) 329–336.
[11] M. Cha, W.A. Chaovalitwongse, Z. Ge, J. Yates, S. Moon, Path protection routing with SRLG constraints to support IPTV in WDM mesh networks, in: Proceedings of the IEEE Global Internet Symposium, May 2006.
[12] M. Cha, G. Choudhury, J. Yates, A. Shaikh, S. Moon, Case study: resilient backbone network design for IPTV services, in: Proceedings of the International Workshop on Internet Protocol TV Services over World Wide Web, May 2006.
[13] T.H. Cormen, E. Leiserson, Charles, R.L. Rivest, Introduction to Algorithms, MIT Press, 1990.
[14] CPLEX. ILOG Inc., <http://www.ilog.com/>.
[15] J.-H. Cui, M. Faloutsos, M. Gerla, An architecture for scalable efficient and fast fault-tolerant multicast provisioning, IEEE Network Magazine 18 (2) (2004) 26–34.
[16] D. Dunn, W. Grover, M. MacGregor, Comparison of *K*-shortest paths and maximum flow routing for network facility restoration, IEEE Journal on Selected Areas in Communications 12 (1) (1994) 88–99.
[17] G. Ellinas, E. Bouillet, R. Ramamurthy, J.-F. Labourdette, S. Chaudhuri, K. Bala, Routing and restoration architectures in mesh optical networks, Optical Networks Magazine 4 (1) (2003) 91–106.
[18] A. Fei, J. Cui, M. Gerla, D. Cavendish. A "Dual-Tree" scheme for fault-tolerant multicast, in: Proceedings of the IEEE International Conference on Communications, June 2001.
[19] GAMS, <http://www.gams.com/>.
[20] W.D. Grover, Mesh-Based Survivable Networks, Prentice-Hall, 2003.
[21] J.Q. Hu, Diverse routing in optical mesh networks, IEEE/ACM Transactions on Networking 51 (3) (2003) 489–494.
[22] J.P. Hughes, W.R. Franta, Geographic extension of HIPPI channels via high speed SONET, IEEE Network Magazine 8 (3) (1994) 42–53.
[23] I.-S. Hwang, R.-Y. Cheng, W.-D. Tseng, A novel dynamic multiple ring-based local restoration for point-to-multipoint multicast traffic in WDM mesh networks, Photonic Network Communications 14 (1) (2007) 23–33.
[24] A. Jukan, AoS-based Wavelength Routing in Multi-Service WDM Networks, Springer-Verlag, Wien, Germany, 2001.
[25] K. Kerpez, D. Waring, G. Lapiotis, J.B. Lyles, R. Vaidyanathan, IPTV service assurance, IEEE Communications Magazine 44 (9) (2006) 166–172.
[26] C. Labovitz, A. Ahuja, A. Bose, F. Jahanian, Delayed internet routing convergence, IEEE ACM Transactions on Networking 9 (3) (2001) 293–306.
[27] P. Leelarusmee, C. Boworntummarat, L. Wuttisittikulkij, Design and analysis of five protection schemes for preplanned recovery in multicast WDM networks, in: Proceedings of the IEEE Sarnoff. Symposium on Advances in Wired and Wireless Communication, 2004.
[28] G. Li, C. Kalmanek, R. Doverspike, Fiber span failure protection in mesh optical networks, Optical Networks Magazine 3 (3) (2002).
[29] Y. Li, Y. Jin, L. Li, L.M. Li, On finding the multicast protection tree considering SRLG in WDM optical networks, ETRI Journal 28 (4) (2006) 517–520.
[30] L. Liao, L. Li, S. Wang, Multicast protection scheme in survivable WDM optical networks, Journal of Network and Computer Applications 31 (3) (2008) 303–316.
[31] H. Luo, L.M. Li, H. Yu, S. Wang, Achieving shared protection for dynamic multicast sessions in survivable mesh WDM networks, IEEE Journal on Selected Areas in Communications 25 (9) (2007) 83–95.
[32] N.F. Maxemchuk, D.H. Shur, An Internet multicast system for the stock market, ACM Transactions on Computer System 19 (3) (2001) 384–412.
[33] M. Medard, S.G. Finn, R.A. Barry, Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs, IEEE/ACM Transactions on Networking 7 (5) (1999) 641–652.
[34] P.V. Mieghem, L. Vandenberghe, Trade-off curves for QOS routing, in: Procedings of the IEEE INFOCOM, April 2006.
[35] R. Piantoni, C. Stancescu, Implementing the Swiss exchange trading system, in: Proceedings of the Symposium on Fault-Tolerant Computing, June 1997.
[36] T. Rahman, G. Ellinas, Protection of multicast sessions in WDM mesh optical networks, in: Proceedings of the IEEE Optical Fiber Communications (OFC), March 2005.
[37] R. Ramaswami, K. Sivarajan, Optical Networks: A practical Perspective, Morgan Kaufmann, 1998.
[38] P. Sebos, J. Yates, G. Hjalmtysson, A. Greenberg. Auto-discovery of shared risk link groups, in: Proceedings of the IEEE Optical Fiber Communication (OFC), March 2001.
[39] C. Semeria, J.W. Stewart, Supporting Differentiated Service Classes in Large IP Networks, Whitepaper, Juniper Networks, 2001.
[40] L. Shen, X. Yang, B. Ramamurthy, Shared risk link group (SRLG)-diverse path provisioning under hybrid service level agreements in wavelength-routed optical mesh networks, IEEE/ACM Transactions on Networking 13 (1) (2005) 918–931.
[41] N.K. Singhal, B. Mukherjee, Protecting multicast sessions in WDM optical mesh networks, IEEE/OSA Journal of Lightwave Technology 21 (4) (2003) 884–892.
[42] N.K. Singhal, L.H. Sahasrabuddhe, B. Mukherjee, Optimal multicasting of multiple light-trees of different bandwidth granularities in a WDM mesh network with sparse splitting capabilities, IEEE/ACM Transactions on Networking 14 (5) (2003) 1104–1117.
[43] N.K. Singhal, L.H. Sahasrabuddhe, B. Mukherjee, Provisioning of survivable multicast sessions against single link failures in optical

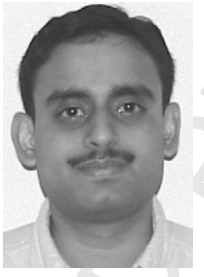
Please cite this article in press as: M. Cha et al., Efficient and scalable provisioning of always-on multicast streaming services, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.07.005

WDM mesh networks, IEEE/OSA Journal of Lightwave Technology 21 (11) (2003) 2587–2594.
[44] S. Vedantham, S.-H. Kim, D. Kataria, Carrier-grade ethernet challenges for IPTV deployment, IEEE Communications Magazine 44 (7) (2006) 24–31.
[45] H. Zang, C. Ou, B. Mukherjee, Path-protection routing and wavelength assignment (RWA) in WDM mesh networks under duct-layer constraints, IEEE/ACM Transactions on Networking 11 (2) (2003) 248–258.
[46] F. Zhang, W.-D. Zhong, Performance evaluation of optimal multicast protection approaches for combined node and link failure recovery, IEEE/OSA Journal of Lightwave Technology 26 (19) (2008) 3298–3306.
[47] F. Zhang, W.-D. Zhong, Performance evaluation of *p*-cycle based protection methods for provisioning of dynamic multicast sessions in mesh WDM networks, Photonic Network Communications 16 (2) (2008) 127–138.
[48] F. Zhang, W.-D. Zhong, *p*-Cycle based tree protection of optical multicast traffic for combined link and node failure recovery in WDM mesh networks, IEEE Communications Letter 13 (1) (2009) 40–42.

Meeyoung Cha is a post-doctoral researcher at Max Planck Institute for Software Systems (MPISWS). She received a Ph.D. degree in Computer Science from KAIST in 2008. Her research interests are in the design and analysis of large-scale networked systems. Her recent work has focussed on multimedia streaming systems and online social networks. She won the best paper award at ACM IMC 2007 for her work characterizing the YouTube workload.

W. Art Chaovalitwongse is an Assistant Professor of Industrial and Systems Engineering, Rutgers University. He received a B.S. degree in Telecommunication Engineering from King Mongkut Institute of Technology Ladkrabang, Thailand, in 1999 and M.S. and Ph.D. degrees in Industrial and Systems Engineering from University of Florida in 2000 and 2003. He previously worked as a Post-Doctoral Associate in the NIH-funded Brain Dynamics Laboratory, Brain Institute and in the departments of Neuroscience and Industrial and Systems Engineering at University of Florida. Before joining Rutgers, he worked for one year at the Corporate Strategic Research, ExxonMobil Research and Engineering, where he managed research in developing efficient mathematical models and novel statistical data analyses for upstream and downstream business operations.

Jennifer Yates is a Technical Specialist in IP Network Management and Performance group at AT&T Labs – Research. She has worked on issues relating to IP control of optical networks, IP and optical integration, and IP network management and performance. She is extremely active both within the Research community and within AT&T – the author of numerous papers, standards contributions, patent applications and an editor of Transactions on Networking. She received the Ph.D. degree from the University of Melbourne in 1998.



Aman Shaikh is a Technical Specialist at AT&T Labs (Research). He obtained his Ph.D. and M.S. in Computer Engineering from the University of California, Santa Cruz in 2003 and 2000 respectively. He also holds a B.E. (HONS) in Computer Science and an M.Sc. (HONS) in Mathematics from the Birla Institute of Technology and Science, Pilani, India. His current research interests include IP routing, and network management and operations. He has published several research and technical papers in these areas.

Sue Moon received her B.S. and M.S. from Seoul National University, Seoul, Korea, in 1988 and 1990, respectively, all in computer engineering. She received a Ph.D. degree in computer science from the University of Massachusetts at Amherst in 2000. From 1999 to 2003, she worked in the IPMON project at Sprint ATL in Burlingame, California. In August of 2003, she joined KAIST as an assistant professor and now teaches in Daejeon, Korea. Her research interests are in network performance measurement and monitoring of diverse network types and their security and anomalous aspects.