Title	Analysis of topology aggregation techniques for QoS routing
Author(s)	Uludag, S; Lui, KS; Nahrstedt, K; Brewster, G
Citation	Acm Computing Surveys, 2007, v. 39 n. 3, article no. 7
Issued Date	2007
URL	http://hdl.handle.net/10722/73844
Rights	ACM Computing Surveys. Copyright © Association for Computing Machinery, Inc.

Analysis of Topology Aggregation Techniques for QoS Routing

SULEYMAN ULUDAG

University of Michigan - Flint

KING-SHAN LUI

University of Hong Kong

KLARA NAHRSTEDT

University of Illinois Urbana-Champaign

and

GREGORY BREWSTER

DePaul University

We study and compare topology aggregation techniques used in QoS routing. Topology Aggregation (TA) is defined as a set of techniques that abstract or summarize the state information about the network topology to be exchanged, processed and maintained by network nodes for routing purposes. Due to scalability, aggregation techniques have been an integral part of some routing protocols. However, TA has not been studied extensively except under a rather limited context. Under the continuing growth of the Internet, scalability issues of QoS routing have been gaining more importance. Therefore, we survey the current TA techniques, provide methodology to classify, evaluate, and compare their complexities and efficiencies.

1. INTRODUCTION

1.1 Preliminaries

Soon after the conception of the principles of packet-switching data networks in the early 1960s [Baran 1964], [Kleinrock 1961] and [Davies 1965], delivery of packets to
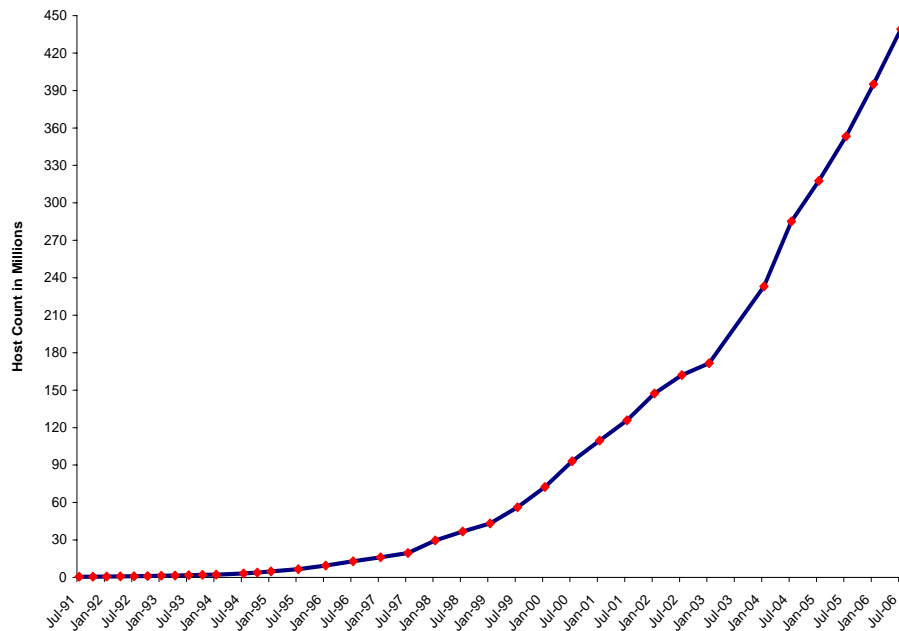
Fig. 1. Internet Domain Survey Host Count History as of July 2006. Source:Internet Systems Consortium (www.isc.org).

their intended destinations, or *routing*, became one of the most vital elements of network designs. Routing is realized by means of routing protocols. Algorithms and closely intertwined set of functions compose routing protocols. With the expected, but often times mutually opposing, requirements of accuracy, simplicity, optimality, efficiency and scalability, routing still retains its central importance in the packet switching data networks today. It is not an exaggeration to state that its significance is increasing due to factors such as the following:

(1) The ever-increasing transmission rates of networks, as well as the emerging new applications, result in new challenges. Quality of Service (QoS) has already been fueling the demand for better and more efficient routing infrastructure.

(2) New operating environments, such as wireless, sensor and Ad Hoc networking, necessitate adaptation of the principles of the legacy routing and/or invention of new ones. These new operating environments range from the more tangible wireless and Ad Hoc networks to more distant and esoteric ones, such as *Interplanetary Internet (http://www.ipnsig.org/)* [Burleigh et al. 2003].

(3) Even though the rate of increase in the number of Internet hosts, domain names and users have somewhat diminished as a result of the recent slump in the world economy, their overall figures still make the routing task more daunting than ever. Figure 1 shows a plot of the number of hosts on the Internet as of July of 2006.

All of these factors keep the research on legacy routing and QoS routing as vital [Clark et al. 2002] and have led to many studies. Systematic taxonomy and

analysis of QoS routing (routing in short henceforth) are given in [Chen and Nahrstedt 1998], [Mieghem et al. 2003] and [Younis and Fahmy 2003]. In what follows we narrow our focus to scalability dimension by means of topology aggregation within the general routing area.

The routing function provides connectivity among a set of participating nodes. In order to deliver packets to the intended destinations, state information about the network must be known by the routing protocols. Some of these state information components are static, such as the capacity of a link in terms of data transmission rate, while some others are dynamic, such as the available (instantaneous) capacity, delay, etc. It is this dynamic aspect of the network conditions that requires the exchange, processing, maintenance and storage of state information at each node.

The scalability requirement for routing addresses the performance of the network with respect to routing as the spatial or temporal characteristics of the network state information change. IRTF Routing Research Group's (RRG) Future Domain Routing (FDR) Scalability Research Subgroup (RR-FS) (http://rr-fs.caida.org/) has been recently established to cope with this problem from the perspective of the distributed computation theory. The objective of any scalable routing technique is to embed the scaling notion into every step of the design process and to ensure a predictable and acceptable level of performance. Example causes of the potential growth of state information are (a) increase in network nodes that participate in routing, (b) increase in offered traffic load, and (c) addition of new users with similar or more rigid performance expectations. Reducing such spatial state information has received some attention in the research community in the past. One noteworthy technique to deal with scalability has been *Topology Aggregation (TA)*. Broadly speaking, aggregation involves studying constituent micro processes of macro systems in order to represent the latter by a fraction of the complete information from the former with the greatest accuracy possible. Representative and tractable characterization and modeling of systems have been invaluable for this problem. TA, as it is used in the networking field and especially with respect to routing, refers to the abstraction or summarization of the state information to be exchanged, processed and maintained by network nodes. The objective is to optimize the performance of the routing and hence the overall system and to prevent operation degradation.

Suppose that the bottommost level I of Figure 2 represents the actual physical topology of a network. TA techniques aim at transforming that physical topology into a more succinct representation, such as level II or even level III, so that routing algorithms may run on that compressed or aggregated topology information with the least possible deviation from the optimum had it been run on the actual physical topology (level I) in Figure 2. In graph-theoretic terms, TA schemes are sometimes referred to as *graph compaction* techniques.

1.2 Organization

This article surveys the previously proposed TA techniques, provides methodology, taxonomies, and compares and contrasts them. Section 2 presents the relevance of the TA techniques and the motivation for this study. Section 3.1 reviews the hierarchical network architectures and structures that facilitate or utilize such aggregation techniques. The existence of hierarchy, explicit or implicit, in all routing architec-
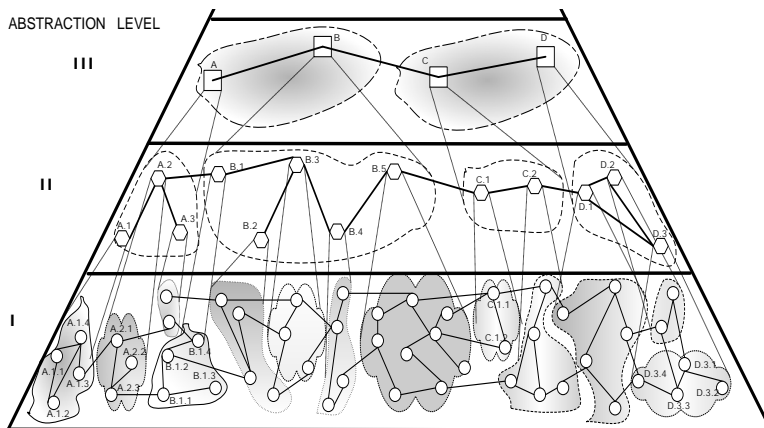
Fig. 2. Topology Aggregation in the sense of a pyramidal hierarchical network with Abstraction Level I, II and III.

tures, including the current Internet routing, is emphasized. The network model, notations, assumptions and definitions are outlined in Section 3.2. We present two important classifications of the TA techniques in Section 4. Structural taxonomy of TA techniques, based on Figure 9, deals with the logical layout or formation of nodes and is explained in Section 4.2 with more emphasis on the more commonly used categories, such as Full-Mesh, Simple Compaction, Tree-based and Star-based. Section 4.3 presents the classification of how a representative (termed as *epitome*) QoS parameter value is picked when there are more than one path between two nodes. Section 4.3 ends with a comparison and discussion. An important part of any TA scheme is its re-aggregation policy, which is studied with a classification of available options in Section 5. Section 6 contains the concluding remarks.

2.   MOTIVATION FOR TOPOLOGY AGGREGATION

Several trends have emerged concerning QoS routing on the Internet. The individual ASes that make up the Internet have become more densely interconnected, as opposed to the tree structure as envisioned by the design [Doria and Davies 2006]. This topological change has been partly propelled by the ever-decreasing costs of data communications and partly by the resilience sought by the customers through multi-homing. What is further fueling the change is the proliferation of new services being requested and, thereby, constraints[1] required by the customers. As a result, the number of registered ASes, the BGP FIB[2] size and the total advertised IP address space are on the rise [Doria and Davies 2006; Huston 2006]. From the

---

[1]Constraints are imposed by the QoS and more recently by the Traffic Engineering (TE) requirements of the operational IP networks. IETF TE Working Group defines TE in RFC3272 as *that aspect of Internet network engineering dealing with the issues of performance evaluation and performance optimization of operational IP networks* [Awduche et al. 2002].

[2]Forwarding Information Base (FIB) is defined in RFC1812 [Baker 1995] as the table containing the information necessary to forward IP Datagrams. This table contains the state information such as the interface identifier and next hop information for the reachable destinations.

perspective of the routing architectures and algorithms, all of the above boil down to more state information to be maintained greater processing power requirements and more bandwidth to be exchanged for the routing updates. The routing table entries, the information per routing entry, the size and frequency of the routing update packets are some examples of growing state information variables. These overhead factors are putting a strain on the scalability properties of the Internet routing infrastructure [Huston 2001; Doria et al. 2006].

In addition to the problem of scalability, security and commercial confidentiality of the internal layouts of the ASes and domains or subnetworks within the ASes are considered to be essential requirements of the future generation of routing architectures and protocols [Doria and Davies 2006].

Topology aggregation has been proposed as a solution for problems similar to the above, but under different design paradigms [Castineyra et al. 1996; ATM Forum 2002]. The first such proposal, the Nimrod Architecture [Castineyra et al. 1996], was one of the candidates for IPng (or IP next generation, later renamed as IPv6) but was eliminated from the process because it was deemed to require *too much of a research effort* [Bradner and Mankin 1995]. A second approach, which is generally considered to have been inspired by Nimrod, is the ATM PNNI specification. Contrary to the expectations of the many involved in the ATM standardization process, ATM failed to dethrone the IP-based Internet as the infrastructure of future communications networks. This prevented PNNI and its techniques, including TA, from deployment and further testing.

However, there seems to be a renewed interest in topology aggregation techniques recently. For instance, *Map Abstraction* is another term used to refer to the same concept by the Internet Research Task Force (IRTF) Routing Research Group in their effort to lay out the fundamental requirements of the future routing protocols[3]. Further, many major players of the current Internet's design principles have been contemplating fresh approaches, not originally articulated, to carry the Internet into the future. *NewArch* (http://www.isi.edu/newarch/), an initiative to propose a new architecture for future generation Internet, had one of the projects listed as *Map or Abstraction Routing* (very similar to the definition of topology aggregation in this study) in [Clark et al. 2002]. A new, alternate interdomain routing approach is NIRA [Yang 2003] whose Topology Information Propagation Protocol *summarizes*, or aggregates, and propagates a domain's interconnections with its neighbors. Considering these new initiatives and the active research in the field of inter-domain routing, we provide a methodology and framework to evaluate and classify existing topology aggregation techniques. We believe that the holistic view on the TA area as well as the analysis of individual algorithms will be useful for future routing algorithms and protocols.

---

[3]The following quotation verbatim from [Doria and Davies 2006] states one of the topology requirements of future domain routing protocols: *Routers MUST, where appropriate, be able to construct abstractions of the topology that represent an aggregation of the topological features of some area of the topology.* More information about IRTF RRG can be found at http://psg.com/˜avri/irtf/rrg-page.html

3. NETWORK MODEL AND NOTATION

In this section, we briefly survey the routing architectures. Our objective is to show that all routing architectures, including the currently used ones in practice, have the notion of hierarchy, a *sine qua non* of TA techniques. Furthermore, we present the network model of the routing architecture, notations of the physical and logical topology, QoS parameter types, and the computation of path QoS values.

3.1 Hierarchy and Network Routing Architectures

Hierarchy is considered to be one of the key routing design principles for scalability [Yu 2000]. A hierarchically organized network is one whose physical and/or logical layout follows a well-defined structure with multiple levels of abstraction. The main motivation behind it is the principle of information hiding to reduce the state information for scalability purposes. Additional benefits, especially when separately administered domains need to exchange traffic in large public data communications networks, are better network security and the concealing of details of the network's topology, which are usually considered by the owners of the network service providers to be proprietary information.

The downside of hierarchical design is the potential inaccuracy of the state information maintained. For example, it has long been theoretically known that TA may increase average packet path length in the network [Kleinrock and Kamoun 1977] due to the lack of complete information to calculate the optimal path. In practice, Internet Service Providers (ISPs) employ multi-homing and peering to minimize the negative impact of longer path lengths. Multi-homing and peering enable the ISPs to bypass the pure hierarchical paths and make shortcuts to higher layers of the hierarchy or to the peers to reduce the paths lengths. We give more details on the Internet Hierarchy in Section 3.1.1.

Figure 2 depicts a hierarchical network design with three abstraction levels. At each hierarchy level k (where $k \in \{2, ..., m\}$ with m being the *depth* of the hierarchy), aggregated or summarized topology state information about levels *1* to *k-1* is kept instead of the full-blown view of the network structure. In turn, at each level l (where $l \in \{1, ..., m - 1\}$), state information for levels *1* to *l* is aggregated before it is presented to layer $l + 1$. The main motivation behind these efforts is based on the observation that as the network size N increases, the cost of routing becomes prohibitively expensive; in particular, more storage for routing tables, more processing power and line capacity for increased routing state updates are needed. Hierarchical clustering schemes are proposed as a solution to this problem. The main idea, for any node, is to keep more complete routing information about network nodes in terms of a nearness criteria, such as hop distance, and less detailed or aggregated information for the nodes further away from it. Thus, it follows a *pyramidal* structure with more information aggregation in the upper levels in the hierarchy. Figure 3 shows the view of node $A.1.2$ for Figure 2's network in which each cloud is assumed to represent a cluster. Node $A.1.2$ only maintains complete (or more complete) information about the nodes within its cloud (i.e. $A.1$). Summarized, aggregated or abstracted routing information is maintained for the rest of the network.

The seminal work for hierarchical networks from queuing-theoretic perspective
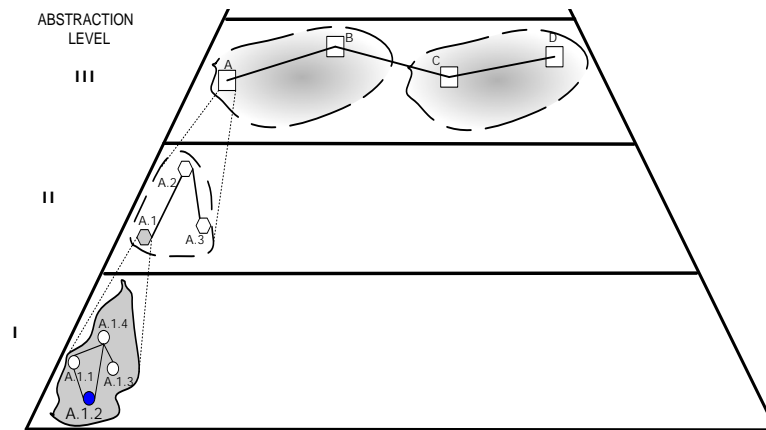
Fig. 3.   Node A.1.2's view of the network topology for Figure 2's network.

was carried out by Kamoun and Kleinrock [1977; 1979; 1980]. Many others followed up with different architectures: Adaptive Hierarchical Routing Protocol (AHRP) [Tsai et al. 1989], Landmark Hierarchy [Tsuchiya 1988], Scalable Inter-Domain Routing Architecture(SIDRA) [Estrin et al. 1992], Inter-Domain Policy Routing (IDPR) [Streenstrup 1993], Viewserver Hierarchy [Alaettinoglu and Shankar 1995], Nimrod Routing Architecture [Castineyra et al. 1996], ATM PNNI [ATM Forum 1996; ATM Forum 2002], Area-based Link-Vector Algorithms [Behrens and Garcia-Luna-Aceves 1998]. Even the most commonly used routing protocols today make use of some sort of hierarchy as part of their critical functionality, such as areas in OSPF [Moy 1998], levels in IS-IS [ISO/IEC10589 2002], confederations and route reflectors in BGP [Rekhter and Li 1995]. It is this hierarchical infrastructure, either explicit or implicit, that the Topology Aggregation techniques exploit to lay the ground for scalable routing.

   3.1.1 *Internet Hierarchical Topology.* The Internet topology has been shown to have a hierarchical structure. The Internet is made up of Autonomous Systems (AS). An AS is a collection of networks, routers and links that fall under the authority of a single organization and exposes a single, cohesive policy to outside networks. Figure 4 shows the time series of the continuous growth of the AS numbers for the past decade. The structure of the Internet topology and its hierarchy can be inferred [Subramanian et al. 2002; Cai et al. 2004; Ge et al. 2001] from the customer-provider and peer-to-peer relationships even though there is no central authority that dictates or maintains it. We briefly summarize the classification of ASes from [Subramanian et al. 2002] to show the intrinsic hierarchy of the Internet topology:

—Layer 0 (Dense Core): These ASes form the innermost core of the Internet hierarchy. They have no upstream providers and form a clique with the others at this level through settlement-free agreements. There are less than 20 ASes in this category. Examples include AT&T, AOL, Sprint, etc.

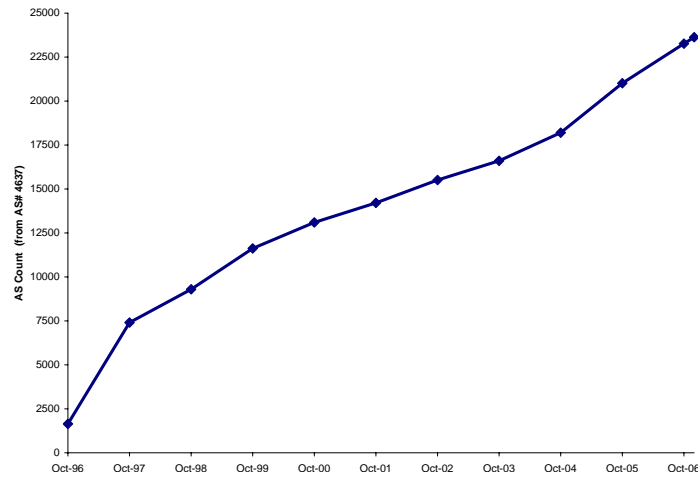—Layer 1 (Transit Core): These are the large national providers and customers of

Fig. 4. The growth of AS allocations over the past decade.Source: http://bgp.potaroo.net/as4637/

the dense core ASes. They also peer with each other. Examples include UUNet Europe, Singapore Telecommunications. When the number of ASes was 10915 (from the date of the routing table dumps in ([Subramanian et al. 2002]), Transit Core ASes numbered 129.

—Layer 2 (Outer Core): These are the regional ISPs with a few customer ASes. They also peer with other such regional ISPs. Examples are Turkish Telecom, Minnesota Regional Network, etc. 897 ASes out of 10915 were in this category in late 1999.

—Layer 3 (Small Regional ISPs): These are the ASes with one or more customers. 971 such ASes were identified out of 10915 ASes.

—Layer 4 (Stub ASes): These are the origins or the sinks of the traffic. They do not carry transit traffic. 8898 ASes were in this category in the aforementioned study.

3.2   Notation and Definitions

The network is modeled as a hierarchical topology. Notation is given in a complete form, and, where necessary, some simplifications are provided to reduce clutter. Without loss of generality, it would suffice to restrict our model to a two-level hierarchy in this study. Unless stated otherwise, the network models are undirected.

A set of *domains*[4] constitutes an inter-network (called an internet). Let $I(D, L)$ tuple denote a connected internet, where D is the set of domains that compose the internet, $D = \{G_i \mid G_i = (V_i, E_i), \text{ where } 1 \leq i \leq |D|\}$, and L is the set of directed, inter-domain links that connect the domains, $L = \{l_{xy}^{ij} \mid \forall \ G_i, G_j \in D$

---

[4]Roughly defined, a domain is a set of network nodes (or routers) that exchange routing update messages by means of a common interior routing protocol. In this survey, it might be an *area* in an OSPF network, an *Autonomous System* in the Internet, a *Peer Group* in ATM PNNI specification or just simply a *subnetwork*.
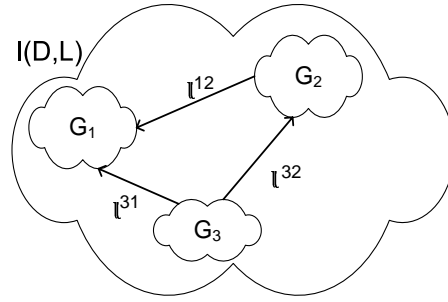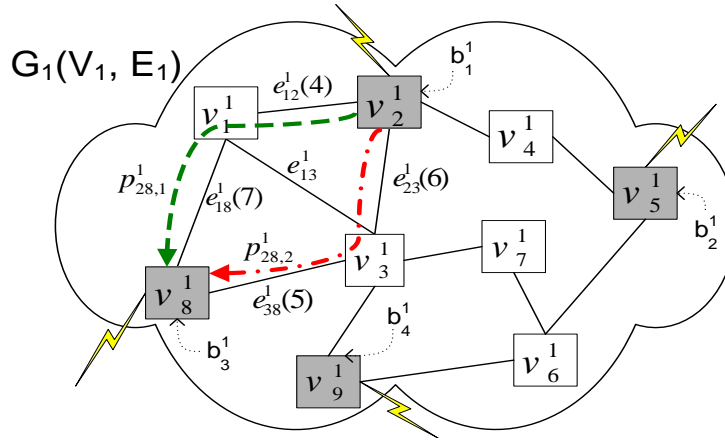
Fig. 5.   An example of the internet model.

Fig. 6.   An example of the internet domain model.

and are connected via border nodes b_x^i and b_y^j}. A border node is defined as the edge node that makes connections incoming from or outgoing to other domains and denoted by b_x^i as the xth border node of the domain i. An example $I(D, L)$ is illustrated in Figure 5.

Each domain is connected and modeled as a tuple $G(V, E)$, where V is the set of vertexes or nodes and E is the set of directed edges or links in the domain. An example domain using the notation stated below in Section 3.2.1 is depicted in Figure 6 with only a subset of the components marked to simplify the illustration. Let $|D|, |L|, |V_i|, |E_i|$ refer to the number of domains, inter-domain links, vertexes in domain G_i and intra-domain links in G_i, respectively. The set of border nodes of domain G_i is denoted by $B_i \subseteq V_i$, which are connected to other domain border nodes via some inter-domain links.

3.2.1   *Notation of Physical Elements.* The following are the definitions and notations of the *Physical* Topology:

—$v_j^i \rightarrow$ physical node j of domain G_i.

—$V_i = \{v_1^i, v_2^i, v_3^i, \ldots, v_{|V_i|}^i\} \to$ the set of all nodes or vertexes in domain G_i. Wherever the domain i under consideration is apparent from the context, the superscript or subscript signifying the domain is dropped. For example, V_i becomes V when domain i is obvious. In Figure 6, $V_1 = \{v_1^1, v_2^1, v_3^1, v_4^1, v_5^1, v_6^1, v_7^1, v_8^1\}$ or simply $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$.

—$e_{jk}^i \to$ physical intra-domain link from node v_j to v_k in domain G_i.

—$E_i = \{e_{jk}^i \mid \forall v_j, v_k \in V_i$ which are connected$\} \to$ the set of all links in domain G_i.

—$B_i = \{b_1^i, b_2^i, b_3^i, \ldots, b_{|B_i|}^i\} \to$ the set of all border nodes in G_i. In Figure 6, the border nodes are denoted by shadowed squares: $B_1 = \{v_2, v_5, v_8, v_9\}$ or $B_1 = \{b_1^i, b_2^i, b_3^i, b_4^i\}$, where mapping from v_j to b_k^i is done in ascending order of the node numbers in V.

—$|B_i| \to$ is the number of border nodes in G_i.

—$P_{jk}^i = \{p_{jk,1}^i, p_{jk,2}^i, \ldots, p_{jk,|P_{jk}^i|}^i\} \to$ the set of all paths from node v_j to node v_k in domain G_i. In Figure 6, two paths from v_2 to v_8 are depicted among other paths of the set P_{28}^1, that is $\{p_{28,1}^1, p_{28,2}^1\} \subset P_{28}^1$.

—$|P_{jk}^i| \to$ the number of distinct paths from node v_j to node v_k in domain G_i.

—$p_{jk,s}^i[n] \to$ the nth link of the sth path from node v_j to node v_k in domain G_i. In Figure 6, $p_{28,1}^1[1] = e_{12}^1$.

—$p_{jk,s}^i = \{p_{jk,s}^i[1], p_{jk,s}^i[2], \ldots, p_{jk,s}^i[|p_{jk,s}^i|]\} \to$ the set of all links of the sth path from node v_j to node v_k in domain G_i. In Figure 6, the first path from node v_2 to node v_8 has two links, i.e. $p_{28,1}^1 = \{p_{28,1}^1[1], p_{28,1}^1[2]\} = \{e_{12}^1, e_{18}^1\}$.

—$|p_{jk,s}^i| \to$ the number of links of the sth path from node v_j to node v_k in domain G_i.

—$P_i = \{P_{jk}^i \mid \forall v_j, v_k \in V_i\} \to$ the set of all physical paths in G_i.

—$Q = \{q^1, q^2, q^3, \ldots, q^{|Q|}\} \to$ the set of all QoS parameters associated with links. $|Q| = m$ is the total number of QoS parameters.

—$q_{e_{jk}^i}^r \to r$th QoS parameter of link e_{jk}^i, where $1 \le r \le m = |Q|$.

3.2.2 *Notation of Logical Elements.* $I'(D', L')$ denotes the transformed reproduction of an internet $I(D, L)$ to reduce the state information needed to represent and convey it compactly. Likewise, a subgraph $G_i'(V_i', E_i')$ is a transformation or reduction of graph $G_i(V_i, E_i)$. A simple transformed logical representation of the physical topology of Figure 6 is illustrated in Figure 7.

We will use the following notations and definitions for the transformed topology, called *logical* topology:

—$D' = \{G_1', G_2', G_3', \ldots, G_{|D|}'\} \to$ the set of domains in the logical topology.

—$L' = \{l_{xy}^{ij}{}' \mid \forall G_i', G_j' \in D$ and connected via border nodes $b_x^i{}'$ and $b_y^j{}'\} \to$ the set of all logical inter-domain links.

—$v_j^i{}' \to$ logical node j of domain G_i'.

—$V_i' = \{v_1^i{}', v_2^i{}', v_3^i{}', \ldots, v_{|V_i|}^i{}'\} \to$ the set of all logical nodes in G_i'. In Figure 7, $V_1' = \{v_1^1{}', v_2^1{}', v_3^1{}', v_4^1{}', v_5^1{}'\}$. Note that by general practice the first four nodes
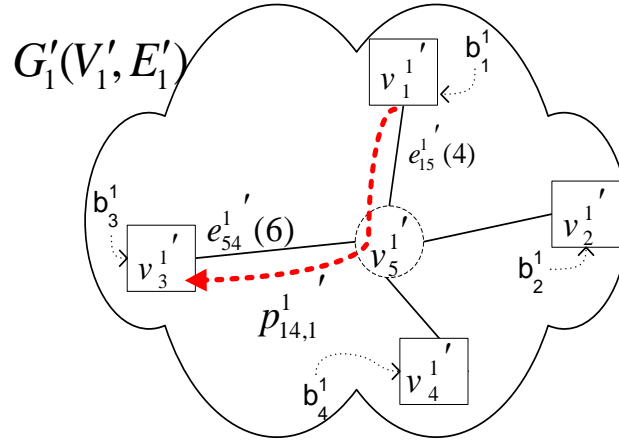
Fig. 7. A transformed graph of Figure 6 by means of a simple star topology. Note that the transformed or logical topology consists of border nodes of the physical topology shown in Figure 6 and a fictitious node $v_5^{1\prime}$.

of $V_1{}'$ are the border nodes from Figure 6, i.e. $v_1^{1\prime} = v_2^1 = b_1, v_2^{1\prime} = v_5^1 = b_2, v_3^{1\prime} = v_9^1 = b_3, v_4^{1\prime} = v_8^1 = b_4$. The last node $(v_5^{1\prime})$ is the fictitious node of the transformed star topology.

—$e_{jk}^{i\ \prime} \rightarrow$ logical intra-domain link from node $v_j{}'$ to node $v_k{}'$ in domain $G_i{}'$.

—$E_i{}' = \{e_{jk}^{i\ \prime} \mid \forall v_j', v_k' \in V_i{}' \text{ which are connected}\} \rightarrow$ the set of all logical intra-domain links in $G_i{}'$.

—$p_{jk}^{i\ \prime} = \{p_{jk,1}^{i\ \prime}, p_{jk,2}^{i\ \prime}, \dots, p_{jk,|p_{jk}^i|}^{i\ \prime}\} \rightarrow$ the set of all logical paths in transformed topology from node $v_j{}'$ to node $v_k{}'$ in domain $G_i{}'$. In Figure 7, there is only one path from $v_1{}'$ to node $v_4{}'$ due to the peculiar features of the star topology, i.e. $p_{14}^{1\ \prime} = \{p_{14,1}^{1\ \prime}\}$ which is denoted by dashed line in the figure.

—$|p_{jk}^{i\ \prime}| \rightarrow$ the number of distinct paths from node $v_j{}'$ to node $v_k{}'$ in domain $G_i{}'$.

—$|p_{jk,r}^{i\ \prime}| \rightarrow$ the number of links of rth path from node $v_j{}'$ to node $v_k{}'$ in domain $G_i{}'$.

—$p_{jk,r}^{i\ \prime} = \{p_{jk,r}^{i\ \prime}[1], p_{jk,r}^{i\ \prime}[2], \dots, p_{jk,r}^{i\ \prime}[|p_{jk,r}^{i\ \prime}|]\} \rightarrow$ the set of all links of the rth path from node $v_j{}'$ to node $v_k{}'$ in domain $G_i{}'$. In Figure 7, the path from node $v_1{}'$ to node $v_4{}'$ has two links, i.e. $p_{14,1}^{1\ \prime} = \{p_{14,1}^{1\ \prime}[1], p_{14,1}^{1\ \prime}[2]\} = \{e_{15}^{1\ \prime}, e_{54}^{1\ \prime}\}$.

—$P_i{}' = \{p_{jk}^{i\ \prime} \mid \forall v_j', v_k' \in V_i{}'\} \rightarrow$ the set of all logical paths in $G_i{}'$.

—$Q' = \{q^{1\prime}, q^{2\prime}, q^{3\prime}, \dots, q^{|Q'|\prime}\} \rightarrow$ the set of all QoS parameters associated with the logical links.

—$q_{e_{jk}^i}^{r\ \prime} \rightarrow r$th QoS parameter of link $\quad e_{jk}^{i\ \prime}, \quad$ where $\quad 1 \le r \le |Q'|$.

3.2.3 *QoS Parameter Computation.* The QoS parameter of a path is computed by means of the QoS parameters of the individual links that form the path. Path computation for the three most common QoS parameter types is given below:

(1) If the QoS parameter is *restrictive*[5], then the minimum (or maximum) value of the links that compose the path number s between node v_j and v_k determines the overall end-to-end QoS parameter for the corresponding path:

$$q^r_{p^i_{jk,s}} = min(max)\{q^r_{p^i_{jk,s}[t]} \quad | \quad 1 \le t \le |p^i_{jk,s}|\}$$

for all restrictive QoS parameters r. Bandwidth is an example of a restrictive QoS parameter. In Figure 6, for the purpose of illustration, let us assume that there are only two paths from node v_2 to node v_8 and that the numbers next to the link notations in the parentheses denote the bandwidth as QoS parameter 1, i.e. $q^1_{p^1_{28,1}} = min\{q^1_{p^1_{28,1}[1]}, q^1_{p^1_{28,1}[2]}\} = min\{q^1_{e_{12}}, q^1_{e_{18}}\} = min\{4,7\} = 4$. Similarly, $q^1_{p^1_{28,2}} = min\{6,5\} = 5$.

(2) If the QoS parameter is *additive*[6], then the sum of the QoS values of all the links that constitute the path number s between node v_j and v_k determines the end-to-end QoS parameter of the path:

$$q^r_{p^i_{jk,s}} = \sum_{t=1}^{|p^i_{jk,s}|} q^r_{p^i_{jk,s}[t]}$$

for all additive QoS parameters r. An example of an additive QoS parameter is delay. For example, if the QoS parameter associated with the links in Figure 6 represents delay then

$$q^1_{p^1_{28,1}} = \sum_{t=1}^{|p^1_{28,1}|} q^1_{p^1_{28,1}[t]} = q^1_{p^1_{28,1}[1]} + q^1_{p^1_{28,1}[2]} = 4 + 7 = 11$$

By the same token, $q^1_{p^1_{28,2}} = 6 + 5 = 11$.

(3) If the QoS parameter is multiplicative, then the end-to-end parameter is computed as the product of the individual link parameters that make up the path number s between node v_j and v_k:

$$q^r_{p^i_{jk,s}} = (1 - \prod_{t=1}^{|p^i_{jk,s}|} (1 - q^r_{p^i_{jk,s}[t]}))$$

for all multiplicative QoS parameters r. Packet loss (or packet delivery) ratio is a multiplicative QoS parameter. For example, if in Figure 6 we assume that the edges are labeled with packet loss ratios in percentages, then we

---

[5]There are different terms used in the literature to refer to the restrictive parameter. Link constraint [Chen and Nahrstedt 1998], link attribute [Lee 1995b], bottleneck, non-additive [Lee 1995a] and concave [Wang and Crowcroft 1996] all refer to the same phenomenon. We adopt restrictive [Bauer et al. 2000] to refer to it.
[6]Similarly, convex, path attribute [Lee 1995b], path constraint [Chen and Nahrstedt 1998] and additive [Wang and Crowcroft 1996] are used interchangeably in the literature.
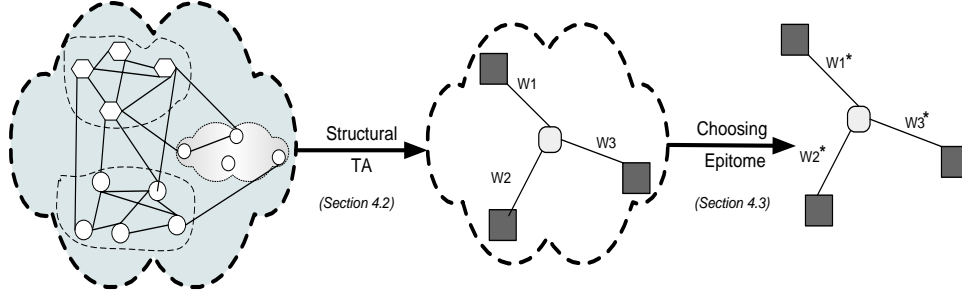
Fig. 8.   Methodology of TA Taxonomies: A Bird's-eye View.

calculate QoS path parameter as follows:

$$q^1_{p^1_{28,1}} = (1 - \prod_{t=1}^{|p^1_{28,1}|} (1 - q^1_{p^1_{28,1}[t]})) \ = \ (1 - [(1 - q^1_{p^1_{28,1}[1]}) \cdot (1 - q^1_{p^1_{28,1}[2]})])$$

$$= \ (1 - [(1 - 0.04) \cdot (1 - 0.07)]) = 10.72\%$$

Applying the same approach to $q^1_{p^1_{28,2}}$, we get $q^1_{p^1_{28,2}} = 10.7\%$.

Choosing the epitome (the most representative) of the QoS parameter values between two nodes, when there are multiple paths with different values, is represented by an *amalgamation* function:

$$q^r_{p^i_{jk}} = \coprod q^r_{p^i_{jk,s}}, \quad \forall \ \ \text{paths} \ \ s.$$

The epitome of QoS parameter r for connecting nodes j and k of domain i is determined by $\coprod()$. A survey of various amalgamation functions $\coprod()$ is presented in details in Section 4.3.

4.   TOPOLOGY AGGREGATION AND ITS TAXONOMIES

Topology aggregation (TA) may be defined as a series of actions that summarizes or abstracts the topological details of the components of a (sub)network to reduce the size of the state information as used by the routing algorithms. It usually involves a compact and succinct portrayal of the underlying (sub)network in terms of the constituent network nodes and/or the transmission links. The expected result of TA, then, is reduced processing power requirements, lower communications overhead via smaller and less frequent updates, and decreased requirements for storage at network nodes. With TA, the routing nodes will need to disseminate smaller updates to other nodes in the network and each will need to consider less voluminous data as input to the routing algorithms. In this section, we start with the discussion of the methodology of taxonomies for TA followed by two different and critical taxonomies of TA techniques with comprehensive comparisons at the end of the corresponding subsections. A macro level example is explained afterwards.

4.1   Methodology of TA Taxonomies

The methodology used to classify, analyze and discuss the TA techniques is depicted

in Figure 8. High complexity with explicit or implicit hierarchy is inevitable in many of today's networks, such as the Internet, first responder networks, military and civil surveillance networks, power networks, etc. The leftmost cloud in Figure 8 represents the aforementioned complexity in physical layouts. Structural TA (Section 4.2) transforms the layout of the physical topology and generates a simplified, logical topology. The middle cloud of Figure 8 shows one possible complexity reduction by means of a star representation. Usually, this step involves a tradeoff between inaccuracy of representation and information reduction; i.e. the more the structural TA reduces the topological information the higher the inaccuracy results. Assignment of weights, albeit suboptimal, may be carried out at this step as well.

The second TA step is about picking the optimal value for the link weights (QoS Parameter values) in the reduced topology as depicted by the rightmost portion of Figure 8. Taxonomy on picking the optimal weight or choosing an epitome for QoS parameters is discussed in more details in Section 4.3.

The two taxonomies are not necessarily carried out in sequence or together. When just a topological reduction is needed without optimal link weights, only the Structural TA may be sufficient. On the other hand, choosing the optimum value between two nodes in the network may not necessarily require the Structural TA. When both topological reduction *and* optimal link weights are demanded a structural TA technique and an epitome selection scheme need to be implemented in sequence as shown in Figure 8.

4.2 Structural Taxonomy of TA

TA is a spatial abstraction or reduction since it involves bringing down the physical size of the state information to be maintained. Structural classification of TA scheme is concerned with the layout of the logical representation of the nodes after aggregation. We depict the structural taxonomy of TA techniques in Figure 9. *Nodal Abatement* simply refers to considering only the border nodes for inter-domain routing and disregarding the other ones. *Link abatement* refers to disregarding the parallel inter-domain links. We will simply refer to the combination of these two techniques as *Topology Transformation*. This results in a more compact representation of the underlying network with little or no information lost. From the graph theoretical perspective, we would like to transform the subnetwork $G_i(V_i, E_i)$ to $G_i'(V_i', E_i')$, where $|V_i'| < |V_i|$, $B_i \subseteq V_i'$ and $|E_i'| < |E_i|$. We exemplify the Topology Transformation techniques on a very simple subnetwork shown in Figure 10[7]. The topology consists of 8 nodes, 4 of which are border nodes, as denoted by shaded circles. The total number of QoS parameters is $|Q| = m = 2$ and ordered pair (q^1, q^2) denotes the restrictive and additive QoS parameters, respectively. In what follows, description and example of the major techniques from Figure 9 are presented first, followed by the complexity comparison.

4.2.1 *Full Mesh (FM).* We start off with the Full Mesh since many other TA techniques use it as the first step. The FM provides full connectivity among the border nodes. It captures the details of the topology at the expense of more spatial

---

[7]Without loss of generality, our example is an undirected graph, i.e. $\forall e_{jk}^i \in E_i$ and $\forall q^r \in Q$, $q_{e_{jk}^i}^r = q_{e_{kj}^i}^r$, to reduce clutter.
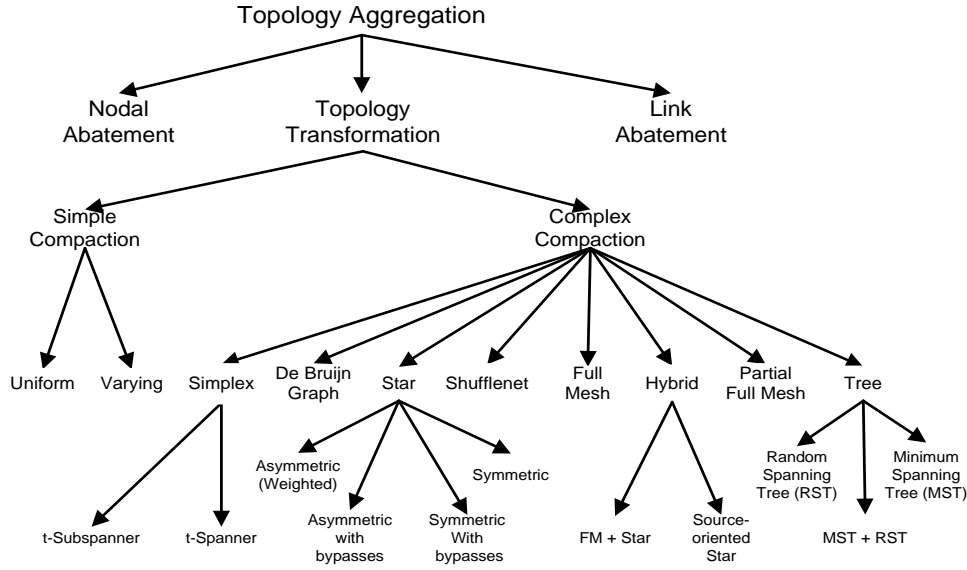
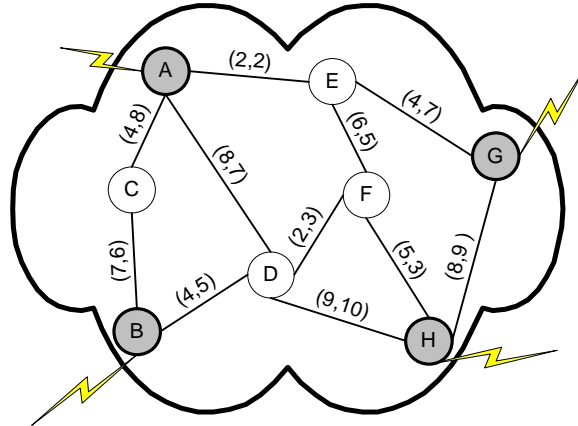Fig. 9.   Structural Taxonomy of topology aggregation techniques.

Fig. 10. A simple subnetwork to illustrate the topology transformation schemes. Link QoS parameters (q^1, q^2) denote the restrictive (e.g. bandwidth) and additive (e.g. delay) QoS parameters, respectively.

and temporal complexity. Figure 11 is the FM representation of Figure 10. FM is not an efficient technique by itself due to its $O(|B|^2)$ space complexity. Nevertheless, it is usually the first step of many TA techniques. FM is studied in many studies, such as [Awerbuch et al. 1998], [Guo and Matta 1998], [Iliadis 2000].

4.2.2   *Simple Compaction.* *Simple Compaction* basically collapses the whole subnetwork G_i into a single node. In the *Uniform Simple Compaction* [Guo and Matta
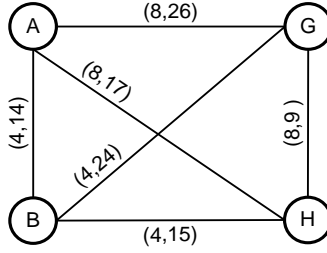
Fig. 11. FM representations of Figure 10. (Details and alternative methods of computing QoS parameters are discussed in Section 4.3.)
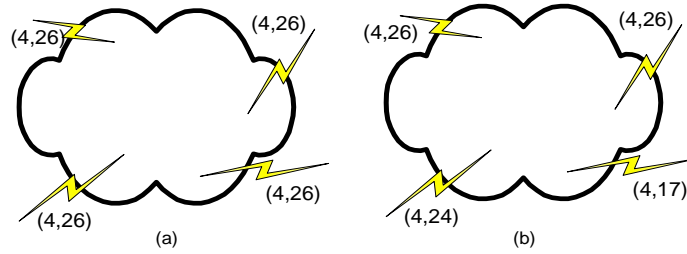

Fig. 12. Advertised vectors of Figure 10 under (a) Uniform Simple Compaction, (b) Varying Simple Compaction.

1998], each border node advertises the same QoS parameter vector[8] to all other sub-networks, whereas in the *Varying Simple Compaction*([Chang and Hwang 2001], Sarangan[2001; 2002; 2004]), the advertised vector may vary from one border node to another. Figure 12(a) shows a possible aggregated topology of Figure 10 based on Uniform Simple Compaction by using the FM from Figure 11 with the worst restrictive parameter (minimum weight of any path that connects two border nodes) of 4 and worst additive parameter (maximum weight between any two border nodes) of 26 as advertised values. Figure 12(b) is a possible aggregated topology of Figure 10 based on Varying Simple Compaction by using the FM from Figure 11 with the worst additive parameter from *each* border node separately as advertised values. For example, node H has three links in the FM in Figure 11 $((4, 15), (8, 17), (8, 9))$ and the worst additive value out of H $(4, 17)$ as shown in Figure 12(b). Note that, in Figure 12, only the additive parameter is allowed to vary although the other parameter or both may be allowed to vary as well. We elaborate on the alternative methods of choosing the epitome of these QoS parameters in Section 4.3. Obviously, the Simple Compaction approach suffers from inaccurate representations, since uniformity across the domain or through the border node is assumed, which is rarely valid.

Next, we study the *Complex Compaction*, which comprises a set of more sophis-

---

[8]Note that the advertised QoS parameter vector represents the associated metric for traversing the domain or the subnetwork that is perceived as a *single node* by others.
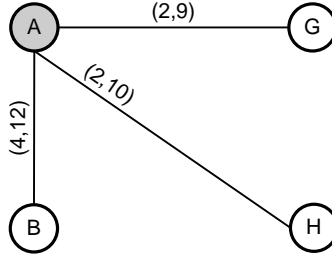
Fig. 13.   Partial FM representation of Figure 10 based on additive parameter minimization.

ticated, yet more accurate, representations of the subnetwork.

4.2.3   *Partial FM.* The *Partial FM*, introduced in [Korkmaz and Krunz 1999; 2000][9], is based on an idea in [Behrens and Garcia-Luna-Aceves 1998; Garcia-Luna-Aceves and Behrens 1995] to reduce the overhead of the FM. The basic idea stipulates that each border node only advertises the *relevant* topology information to the outside. Figure 13 shows the advertised topology of Node A about the subnetwork of our example in Figure 10. The only information, that outside subnetworks need to know, are (a) the number of border nodes in the domain, and (b) QoS parameters to pass through the domain, i.e. QoS parameter to reach other border nodes via Node A.

4.2.4   *Tree-based TA.* *Tree* category from our classification scheme in Figure 9 is another graph compaction method to transform the topology information into a more succinct form. For all of the techniques under tree category, the first step is to transform the topology into a full mesh of the border nodes.

(1) *Spanning Tree (ST)* is a tree representation of the topology that covers all the border nodes without forming a loop. An ST of nodes in B contains exactly $|B| - 1$ links. Thus, the spatial complexity of the topology is reduced to $O(|B|)$ from $O(|B|^2)$. An ST may be constructed based on maximizing a restrictive parameter among the border nodes[10], as shown in Figure 14(a) or on minimizing an additive parameter, as shown on Figure 14(b). The former is called a *restrictive-parameter based Maximum Weight ST*, while the latter is *additive-parameter based Minimum Weight ST*. We use the abbreviation *MST* to refer to either unless there is ambiguity, in which case we will use the full name. Our discussion here assumed undirected (symmetric) graph where link weights are the same on both directions. More detailed studies of MST on undirected graphs can be found in [Awerbuch et al. 1998] and [Lee 1995a]. Detailed treatment of MST on directed graph is provided in [Awerbuch and Shavitt 2001].

(2) *Random ST (RST)* is a spanning tree constructed without regard to maximizing or minimizing any of the QoS parameters. An example is shown in Figure 14(c).

---

[9]The term the authors used in [Korkmaz and Krunz 2000] is *source-oriented*. We use Partial FM to refer to it in this paper.

[10]We aim to choose the paths with the maximum restrictive parameter. For example, in Figure 14(a), the maximum restrictive parameter between nodes A and G is 8 based on Figure 19(a).
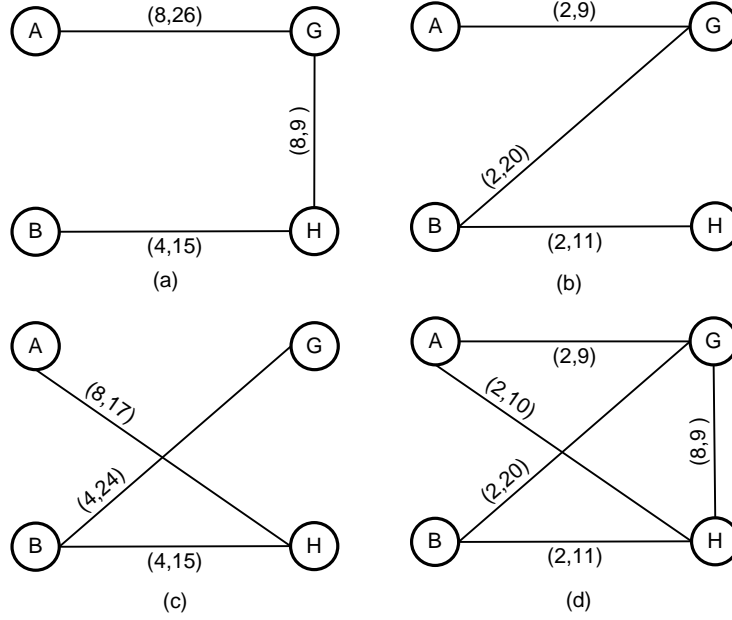
Fig. 14. Spanning Trees of Figure 10. (a) Maximum Weight ST based on the restrictive parameter, (b) Minimum Weight ST based on the additive parameter, (c) RST, (d) MST+RST (Union of (a) and (c)).

The running time complexity is $O(E + B)$, which is better than $O(E * logB)$ of MST[Cormen et al. 2001]. Simulation-based comparative studies of RST and other tree-based TA schemes are presented in Awerbuch[1998; 2001].

(3) The *MST and RST* combination, proposed in [Awerbuch et al. 1998; Awerbuch and Shavitt 2001], is simply a union of the constituent elements, as shown in Figure 14(d). The simulations in [Awerbuch and Shavitt 2001] showed good performance in terms of worst-pair distortion costs compared to other alternatives, where distortion $= \max_{i,j \in V} \frac{q_{p_{ij}}}{q'_{p_{ij}}}$, $q_{p_{ij}}$ is the minimum-cost path from node i to j in the network, while $q'_{p_{ij}}$ is the minimum-cost path from i to j in the aggregated topology.

4.2.5 *Simplex TA.* There are two categories under the *simplex* category: t-spanner and t-subspanner.

(1) *t-spanner*, first introduced in [Chew 1986; Peleg and Ullman 1987], is a spanning subgraph $G'(V, E')$ of graph $G(V, E)$ such that $E' \subseteq E$ and $\forall v_j, v_k \in V, q^r_{p_{jk,s}} \leq q^r_{p_{jk,s}}{}' \leq t \cdot q^r_{p_{jk,s}}$ with respect to a chosen QoS parameter r. The value of t is referred to as the *stretch factor* (i.e. worst-pair distortion) in the literature. The optimal value for the stretch factor is $t = 1$. Extensions based on Minimum Weight ST algorithms, such as Kruskal's [Althöfer et al. 1993], Prim's or Sollin's [Ahuja et al. 1993], can be used to find a t-spanner of a graph. A recent, improved algorithm for spanner graphs can be found in [Baswana and Sen
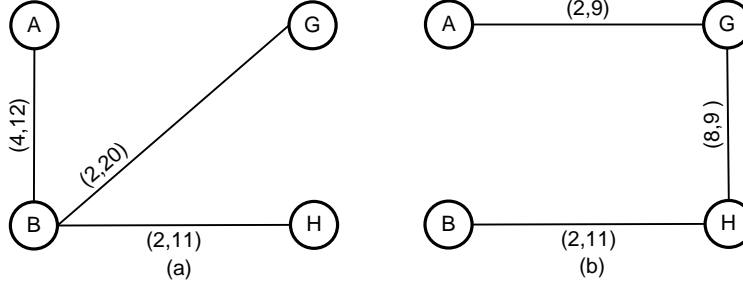
Fig. 15. (a) t-spanner of FM based on additive parameter minimization (t=32/9), (b) t-subspanner of the original topology for the border nodes (t=29/12). Both are based on Figure 10.

2003]. A 32/9-spanner of the example topology is depicted in Figure 15(a) which is based on the FM from Figure 19(b). $t = 32/9$ because worst-pair distortion between Figure 19(b) (the actual topology) and Figure 15(a) (the aggregated topology) for any path additive value is 32/9, i.e. the additive cost between A and G is 32 in aggregated topology versus 9 in the actual topology. Many instances of t-spanner problems are intractable [Cai 1994] even for a single parameter case. To the best of the authors' knowledge, there is no published work regarding the multiple parameter t-spanner because of the high complexity of the problem. Detailed treatment of t-spanner of single parameter can be found in Awerbuch[1998; 2001].

(2) *t-subspanner* is introduced in [Lee 1999] and may be considered as a generalization of the t-spanner. [Lee 1999] defines t-subspanner as a spanning subgraph $G'(V', E')$ of graph $G(V, E)$ such that $E' \subseteq E$, $V' \subseteq V$ and $\forall e_{jk} \in E'$, $q^r_{p^i_{jk}} \leq q^r_{p^i_{jk}}{}' \leq t \cdot q^r_{p^i_{jk}}$ with respect to a chosen QoS parameter r. Note that when $V' = B$, the solution of the t-subspanner approach and the t-spanner of the FM of the same subnetwork are identical. Figure 15(b) shows a t-subspanner, based on the FM of Figure 19(b), where $t = 29/12$ since the worst-pair distortion is between nodes A and B; 29 in the subspanner versus 12 in the actual topology. Minimum Equivalent Subspanner (MES) is a t-subspanner with minimum number of links, where t=1. Figure 16 is an example of MES. MES produces an aggregated topology with identical values among any nodes since $t = 1$. In effect, it tries to eliminate redundant links without changing the cost of paths under aggregated topology from the actual values in the original network. [Lee 1999] provides two algorithms, based on Dijkstra and Floyd-Marshall shortest path algorithms [Ahuja et al. 1993] to find the MES.

4.2.6 *Star-based TA.* Another complex compaction category is *Star*, as recommended by the ATM's PNNI[ATM Forum 2002]. In all our examples below, we assume that star formation is based on only one of the QoS parameters. We discuss the available options to consider more than one parameters in the decision process in Section 4.3. There are 4 kinds of star-based aggregations:

(1) *Symmetric Star* transforms the topology into a logical star with a fictitious nucleus to which each node is connected by an identical link QoS parameter.
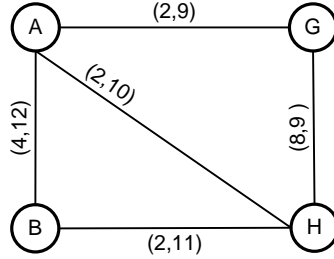
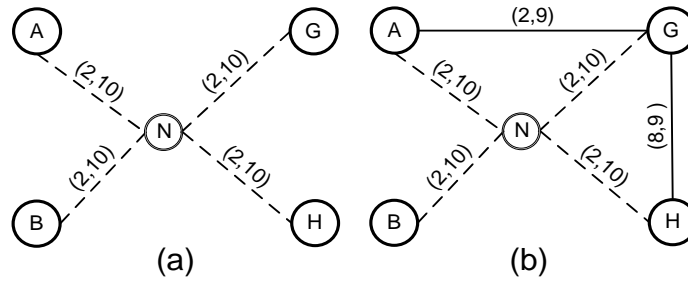Fig. 16.   MES of Figure 10 based on additive parameter minimization.

Fig. 17. Symmetric Star (a) without bypasses, (b) with bypasses. Both are based on Figure 10, with respect to the additive metric.

Figure 17(a) shows an aggregated topology as a symmetric star without by-passes (explained below). Note that the star is symmetric only with respect to the additive metric, but not with respect to the the restrictive metric in Figure 17(a). The logical links that connect border nodes to the nucleus are generally referred to as *spokes*. We address the different methods to determine the QoS parameters to be associated with spokes in Section 4.3. [Awerbuch et al. 1998] and [Guo and Matta 1998] study symmetric star without bypasses.

(2) *Symmetric Star with Bypasses* is a similar structure to the symmetric star, but with the addition of *bypasses*. A bypass or an exception is a direct connection between two border nodes. A symmetric star with only identical QoS parameter values will result in an inaccurate representation of the network unless the underlying topology is very close to uniformity in terms of the QoS parameter values. If a connection between two border nodes via the fictitious nucleus grossly deviates from its real FM value, then a bypass is introduced. The existence of a bypass will ensure that a more realistic QoS parameter value will result. A symmetric star with bypasses is shown in Figure 17(b). A more detailed discussion of symmetric star with bypasses is in [Iliadis 2000] and [Lee 1999].

(3) *Asymmetric (Weighted) Star* is a star whose spokes can take on different QoS parameter values. The asymmetry reflects the underlying heterogeneity of the physical topology. It may be termed as *weighted* to take different criteria into
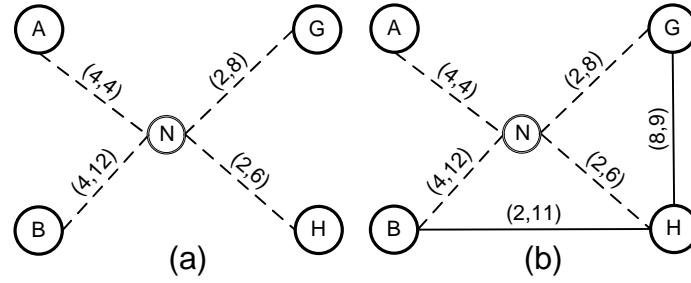
Fig. 18. Asymmetric (Weighted) Star (a) without bypasses, (b) with bypasses. Both are based on Figure 10.

account, such as administrative policies. Figure 18(a) is a depiction of an asymmetric star. An example implementation of an asymmetric star without bypasses is given in [Hao and Zegura 2000], where weights are assigned to the spokes based on the total traffic going through them.

(4) *Asymmetric (Weighted) Star with bypasses* is identical in its main objective to that of the symmetric star with bypasses, i.e. to reduce the inaccuracy by means of bypasses. Figure 18(b) is an asymmetric star with bypasses. Asymmetric star with bypasses is explained in [Iwata et al. 1998] and [Liu et al. 2000].

4.2.7   *Others.* The *De Bruijn Graph* [Yoo et al. 2000] and *Shufflenet* [Yoo et al. 2001] topology aggregation schemes are based on the techniques from [Mukherjee 1997] to represent the FM with better accuracy than a star but with less complexity than the FM itself. Details can be found in [Yoo et al. 2004].

The last category of complex compaction techniques from Figure 9 is the *hybrid* category, which combines more than one of the above techniques. An example hybrid TA method is introduced in [Hao and Zegura 2000] which differentiates the characteristics of the QoS parameters in terms of their expected frequency of change. It asserts that hop count changes less frequently than the available bandwidth and, hence, the former should be advertised less frequently in full-mesh representation, whereas the latter should be periodically advertised in star representation.

4.2.8   *Comparison Table of Structural TA Classes.* In this section, we compare the structural TA classes in Table I. The *Time Complexity* column in Table I gives the typical run-time for the corresponding TA algorithm. Note that FM representation is a required step in all of these topology aggregation techniques. Second relevant dimension of complexity is denoted as *Decode* to specify the time needed to decompress the aggregated topology information to be used by the routing subsystem. The third and last column of complexity is *Spatial Complexity* which is directly proportional to the size of aggregated information to be kept and advertised to the other nodes. Thus, communication overhead is a function of the spatial complexity. Note that the spatial complexity and the precision of the aggregated information are directly related; that is, all other things being equal, reduction in spatial complexity compromises the precision of the aggregated topology. A more detailed comparative analysis is given in [Uludag et al. 2005].

Table I. Comparison of main structural TA types in terms of complexity, QoS parameter type and imprecision.

	Complexity			QoS [a] Metric	Exact? [d]
	Time	Decode [b]	Spatial [c]		
FM	$O(VE^2)$	-	$O(B^2)$	A/R	Y
Simple Cmpct.	$O(VE^2)$	-	$O(1)$	A/R	N
Partial FM	$O(VE^2)$	$O(1)/O(B)/-$	$O(B)$	A/R	Y
MST	$O(VE^2 + ElogB)$	$O(B)/O(B)/O(B^2)$	$O(B)$	A/R	Y/N[e]
RST	$O(VE^2 + E)$	$O(B)/O(B)/O(B^2)$	$O(B)$	A/R	N
t-spanner	$O(VE^2 + B^2)^f$	$O(B)/O(B)/O(B^2)$	$O(B^{1+1/t})$	A	N
1-subspanner	$O(VlogV + E)/O(V^3)$ [g]	$O(B)/O(B)/O(B^2)$	$O(B^2)$	A	Y
Star	$O(BE^2 + BEVlogV)$ [h]	$O(1)/O(B)/O(B^2)$	$O(B)$	R and A	N

To simplify, E, B, V, are used instead of $|E|$, $|B|$, $|V|$, respectively.
[a] Whether the QoS parameter is restrictive (R), additive (A), or R/A for either
[b] Decoding is needed only when the TA category is not FM or Simple. It is presented as Time to decode one logical link / all logical links related to a single border node / all among boder nodes
[c] Spatial complexity of logical structure.
[d] Y if the logial value used is the same as at least one physical path that it is representing, N otherwise.
[e] The representation is exact if the parameter is restrictive; otherwise, for the additive case, it only assures an upper bound [Lee 1995a].
[f] Many instances of t-spanners are known to be NP-Complete, see [Cai 1994; Fekete and Kremer 2001].
[g] $O(VlogV + E)$ is the complexity for distributed mechanism and $O(V^3)$ is the complexity for centralized mechanism.
[h] For different types of star, the time complexity might be slightly different but in all cases FM dominates.

4.3 Choosing an Epitome for QoS Parameters

Structural TA techniques discussed in Section 4.2 often need to choose a representative path value or *epitome* among multiple paths between nodes. An *Amalgamation Function*, $\coprod$, performs this task, as introduced briefly in Section 3.2. For example, in Figure 10, P_{AB} is the set of all paths that connect v_A to v_B, such as $p_{AB,1} = \{e_{AC}, e_{CB}\}$, $p_{AB,2} = \{e_{AD}, e_{DB}\}$, $p_{AB,3} = \{e_{AE}, e_{EF}, e_{FD}, e_{DB}\}, \ldots$. One of these paths should be selected as the *epitome* with the most representative QoS parameter values in the TA process. This decision is at the heart of the TA process and has a direct effect on the resulting inaccuracies. The difficulty is compounded in the presence of multiple QoS parameters. For instance, Figure 19(a) is the FM representation of our topology of Figure 10 based on the maximization of the restrictive QoS parameter and 19(b) is the FM representation based on the minimization of the additive parameter. In each case, we either use the restrictive or the additive parameter to find the best path and its corresponding value between the border nodes. For example, the path with the maximum restrictive parameter from A to G is $p_{AG,1} = \{e_{AD}, e_{DH}, e_{HG}\}$. Thus, in Figure 19(a), $q^1_{p_{AG}} = 8$ and the corresponding additive parameter of the path p_{AG} is $q^2_{p_{AG}} = 26$. Similarly, the path with the best or minimum additive parameter from A to G is $p_{AG,2} = \{e_{AE}, e_{EG}\}$ and thus in Figure 19(b), link from A to G has $(q^1_{p_{AG}}, q^2_{p_{AG}}) = (2, 9)$.

The classification of choosing a path epitome for QoS parameters is depicted
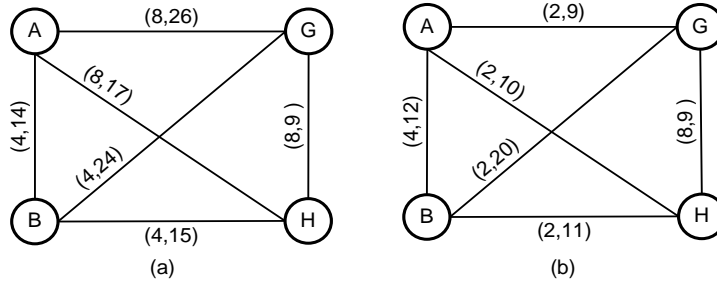
Fig. 19. FM representations of Figure 10. (a) FM based on the restrictive parameter maximization. (b) FM based on the additive parameter minimization.
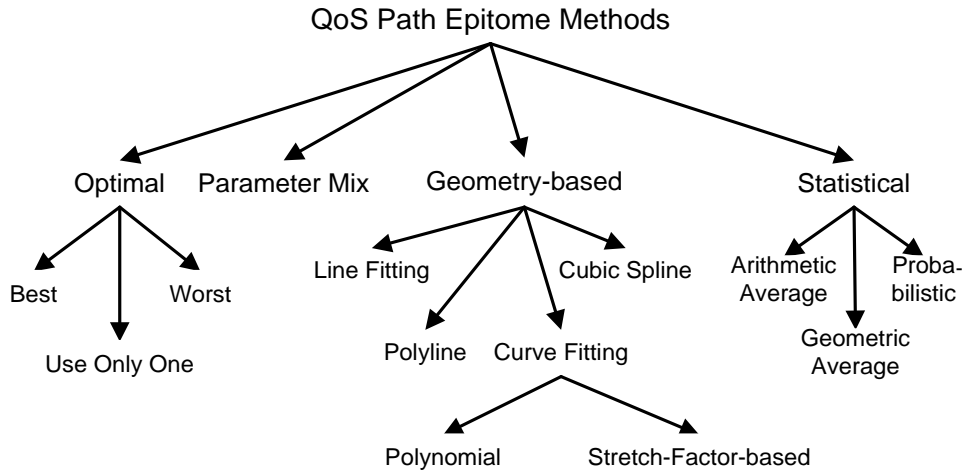

Fig. 20.   The Taxonomy of choosing an epitome for path QoS Parameters.

in Figure 20. Brief discussions of each epitome are provided next, followed by a comparison.

4.3.1 *Optimal Selection.* This is the simplest and most commonly used selection method. *Best Selection* chooses the most favorite (optimal or close-to-optimal) QoS parameter out of the paths under consideration. For the restrictive case, it may be either the maximum or the minimum; for the additive case the lowest, and for the multiplicative metric the smallest product of the individual links. The definition of the *worst* is just the opposite of the *best*. Only a single QoS parameter can be considered by the *worst* and *best* functions. If there are multiple QoS metrics, then we can restrict the decision to be based on one parameter only, *Use Only One* in Figure 20, and use the corresponding values of that path for the other parameters. The other extreme is to use separate representations for each parameter and find representative paths for each q^r. However, it is still not clear how to combine them at the time of routing or forwarding.

4.3.2   *Parameter Mix Selection.* When the number of QoS parameters is greater than one, then the decision gets harder. One option for *parameter mix selection* is to normalize each QoS parameter and use a weighted combination of all normalized parameters as a joint, single QoS metric. It might be a linear, exponential, logarithmic or any other combination. Yet, the interactions of the QoS parameters are not very well understood and this is not an easy decision to make. For example, an aggregated topology based on bandwidth may earmark path $p^i_{jk,1}$ with a bandwidth of 5 and path $p^i_{jk,2}$ with a delay of 9 separately as optimal choices. When a request for a path with 4 units of bandwidth *and* 8 units of delay comes in it is a very hard decision to pick a path that will satisfy both delay and bandwidth requirements *simultaneously*.

4.3.3   *Geometry-based.* Geometry-based representation and geometric concepts turn out to be very useful particularly for one additive and one restrictive parameter[11] (i.e. delay-bandwidth) studies. When representing a set of path parameters on the Cartesian plane, an area that represents the QoS supported by the set of paths can be defined. Suppose that the QoS parameter of a path is $(3, 2)$ (delay is 3 units and bandwidth 2 units). On the Cartesian plane, any request that is below and right to $(3, 2)$ can be supported by this path. That is, the area below and right to a point represents the QoS parameters supported by the underlying path of the point. The area supported by a set of paths is just the union of the areas supported by individual paths in the set. Interestingly, it turns out that, in many cases, not all the paths are needed to describe that area geometrically. We illustrate this by an example.

EXAMPLE 1. *Let X and Y be two border nodes, similar to nodes A and B in Figure 10. Suppose that there are 7 alternate paths between them whose (A, R) pairs (such as delay and bandwidth, respectively) are given by $(3, 2), (4, 2), (4, 5), (7, 2),$ $(5, 3), (6, 5), (7, 8)$. We represent these paths as points on the Cartesian Plane as shown in Figure 21 and the shaded area represents the QoS supported by these paths. The area is a staircase and can be defined by using the points on convex corners only, which are $(3, 2), (4, 5),$ and $(7, 8)$.*

The set of convex corner points is called *Efficient Frontier* (i.e. staircase function) in [Bauer et al. 2000]. More formally, $Q = \{q^1, q^2\}$ is the set of QoS parameter values where q^1 is additive and q^2 is restrictive. Let p_{jk} be a path connecting node j to k. If there exists no other path p'_{jk} such that $q^1_{p'_{jk}} > q^1_{p_{jk}}$ and $q^2_{p'_{jk}} < q^2_{p_{jk}}$, then the path p_{jk} is an element of the efficient frontier from node j to k. The algebraic representation of all the paths sitting on the efficient frontier among all border nodes is referred to as the *transition matrix* by some studies, such as [Iliadis 2000; Bauer et al. 2000].

There are four reported approaches in the geometry-based category; line fitting, curve fitting, polyline and cubic spline:

---

[11]Two additive parameter case can also be represented by geometric representation. However, there are no studies of two additive metric case for TA due to the NP-Completeness of the corresponding path finding problem (Shortest Weight-Constrained Path in [Garey and Johnson 1979].) even though good approximations exist in the literature.
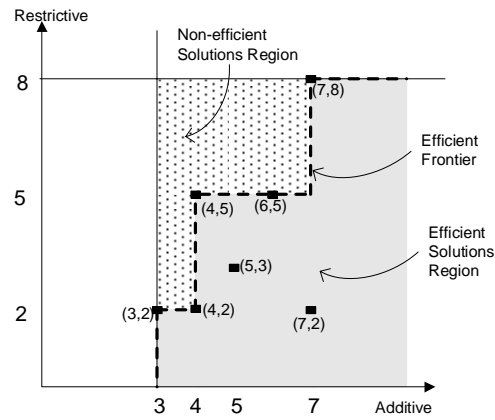
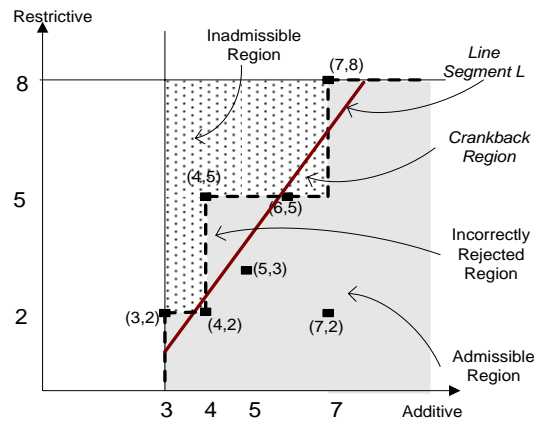Fig. 21.   The geometric representation of the Efficient Frontier for Example 1.

Fig. 22. The geometric representation of Line Segment TA method of [Lui et al. 2000] for Example 1.

Line Fitting — Proposed by [Lui et al. 2004], the basic idea is to use a line segment to represent the efficient frontier on the Cartesian plane for a logical link in the FM. The line segment is found by using linear regression. A sample line segment for the Example 1 is given in Figure 22. Whatever request falls below the line segment is rejected and above is admitted as routable. Obviously, those requests that are below the line segment but outside the dotted staircase function will be cranked back as they are admitted although they are in the inadmissible region. Similarly, all requests that fall in the region above the line segment but below the staircase function are rejected although they are routable. The FM represented by line segments is then transformed into an asymmetric star with bypasses topology by ensuring that the QoS parameters of reaching from node v_i to v_j in star representation via the nucleus is as close to the QoS value in the FM
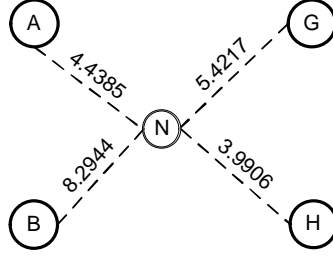
Fig. 23. The asymmetric star representation of Figure 19(b) based on additive metric only by using Liu[2000]'s least-square approximation method.

representation as possible. In order to find the QoS parameter values of logical links to and from the fictitious nucleus of the asymmetric star topology, arithmetic *join* and *split* operations are defined for line segments. The former operation adds the QoS parameters of spokes while the latter breaks the FM values into spoke values. Possible extensions are to study two different two-parameter type combinations such as two additive parameters and to consider more than two parameters.

Similar idea, but only for one additive parameter is analyzed in [Liu et al. 2000]. Asymmetric star with bypasses is the TA method used. Least Square Approximation (LSA) and Maximum Deviation Minimization (MDM) algorithms are compared to find the spoke values. The basic idea is to find the spoke values whose deviation from the actual optimal path values between border nodes is minimized. In other words, the objective function of LSA is

$$F(q_1', q_2', \cdots, q_{|B|}') = \min \sum_{j<k} (\frac{q_j' + q_k'}{q_{jk}} - 1)^2$$

where q_j' represents the QoS parameter value of the spoke from border node j and q_{jk} is the optimal QoS parameter between border node j and k in the original network. For example, if the LSA method was to be used to form the asymmetric star of Figure 19(b) based on additive parameter only, we would get the one depicted in Figure 23. Least-square algorithm can be found in [Cormen et al. 2001]. The objective function of the MDM, on the other hand, minimizes the maximum relative deviation:

$$F(q_1', q_2', \cdots, q_{|B|}') = \min \left(\max_{1 \leq j,k \leq |B|, j \neq k} \frac{|q_j' + q_k' - q_{jk}|}{q_{jk}} \right)$$

Simulation results indicate that LSA outperforms MDM. Even though the LSA is inexpensive in terms of time complexity, when bypasses need to be added, the proposed algorithm for LSA dominates and time complexity is increased to $O(B^2)$.

Curve Fitting — There are two alternative approaches for the curve fitting technique:

The first one, *stretch-factor* based, is introduced in [Korkmaz and Krunz 2000; 1999]. The authors define a new parameter called, the *stretch factor*, which measures the deviation of the parameters of a path from the best parameters possible. The stretch factor of a path p that goes from border node j to k, denoted s_factor_p,
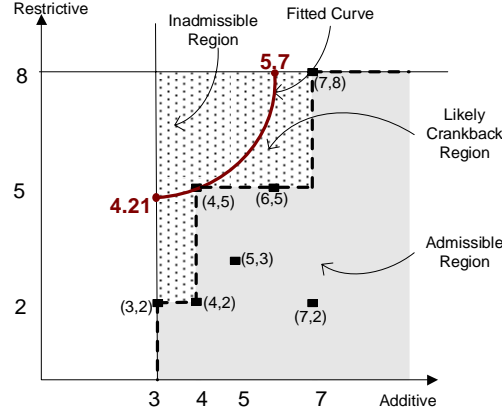
Fig. 24. The geometric representation of Korkmaz2000 approach [Korkmaz and Krunz 2000] for a two-parameter subnetwork model on Example 1.

is defined as follows:

$$s\_factor_p = \frac{Best_{q^r}}{q_p^r} + \frac{q_p^a}{Best_{q^a}}$$

$$Best_{q^r} = max\left\{ q_p^r \quad | \quad \forall p_{jk}, \quad \forall \quad \text{restrictive parameters} \quad r \right\}$$

$$Best_{q^a} = min\left\{ q_p^a \quad | \quad \forall p_{jk}, \quad \forall \quad \text{additive parameters} \quad a \right\}$$

When one restrictive and one additive parameters exist in the system the stretch factor becomes

$$s\_factor_p = \frac{Best_{q^r}}{q_p^r} + \frac{q_p^a}{Best_{q^a}}$$

Refer to Example 1, the best possible delay and bandwidth of the paths are 3 and 8, respectively. Therefore, the stretch factor of (4, 5) is $4/3 + 8/5 = 2.9$.

The authors provide two strategies to assign parameters to logical links. In the first approach, the path with the smallest stretch factor is selected to represent the QoS parameter values between the border nodes. In the second approach, a logical link is represented by the best possible value of each parameter and the minimum stretch factor among the paths. For example, the first approach will use (4, 5), which has the smallest stretch factor among the paths, to represent the paths in Example 1 while the second approach will use (3, 8) and stretch factor 2.9 for the logical link. Omitting the details for brevity, the idea in [Korkmaz and Krunz 2000; 1999] can be illustrated geometrically for Example 1 in Figure 24. It effectively contracts the region in which non-routable requests are admitted. Unlike the second curve fitting approach by [Tang and Chen 2004], to be explained below, Korkmaz-Krunz approach can accommodate more than two QoS parameters.

The second curve fitting method, proposed in [Tang and Chen 2004] fits a least-square polynomial of degree n given m data points in Cartesian plane. When $n = 1$ the approach is identical to Lui [2004]'s. Refer to Example 1, if we decide to use
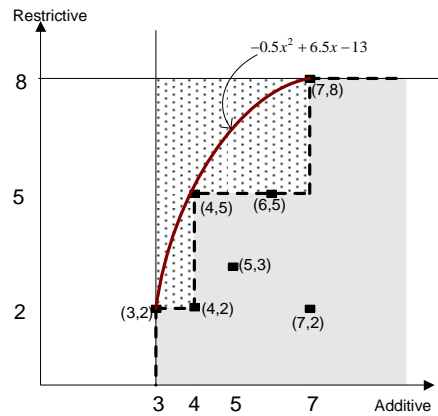
Fig. 25. A Quadratic polynomial fit by least-square method for the data set of Example 1, presented in Tang[2004].
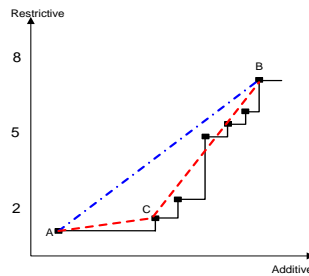

Fig. 26.   Polyline approximation of Tang[2004].

a quadratic function for polynomial curve, then the least-square method generates $-0.5x^2 + 6.5x - 13$ as plotted in Figure 25.

Even though Tang's approach cannot use more than two parameters, its precision is reported to be better than Korkmaz's approach in terms of minimizing the area between the staircase function and the curve.

Polyline — [Tang and Chen 2004] also describes a polyline approach that represents the staircase using several line segments instead of one. It takes advantage of the fact that not all representative points contribute identically to the imprecision of approximation. For example, out of the seven representative points in Figure 26, A, B and C are more relevant than the others for they shape more substantially the overall behavior of the staircase function. The goal is to find a number of line segments that minimize the area between the staircase and the lines. Due to the running time complexity of finding the optimal set of lines, the authors provide a heuristic to choose the line segments once the number of segments used is chosen.

Cubic Spline — The cubic spline approach, also suggested by [Tang and Chen 2004], is similar to polynomial curve fitting; g piecewise cubic polynomials approximate the data set which is broken up into g even ranges in terms of the restrictive

parameter.

The main objective of the geometric techniques discussed above is to minimize the *area* between the staircase function and the approximating geometric representation. Calculation of this area is stated in [Tang and Chen 2004].

4.3.4   *Statistical.* Some statistical tools may also be employed to choose an epitome for QoS parameters under multiple paths. *Arithmetic Average* is the sum of the QoS parameter values of paths divided by the number of paths, i.e. $q_P^{arith} = \frac{1}{|P_{ij}|} \sum\limits_{\forall p \in P_{ij}} q_p$.

Geometric Average is the product of the QoS parameter values of paths raised to a power equal to the reciprocal of the number of paths, i.e. $q_P^{geo} = \left(\prod\limits_{\forall p \in P_{ij}} q_p \right)^{\frac{1}{|P_{ij}|}}$.

Both are only applicable to single-restrictive/additive and single-multiplicative cases.

A *Probabilistic* approach has been proposed in [Ghosh and Acharya 2001] and [Sarangan et al. 2004]. The basic idea is to associate reliability values with QoS parameter availability. The approach is an attempt to accommodate the intrinsic network resource variations by means of a probabilistic model. Each router keeps a time series of their outgoing links to record QoS resource availability. Relative frequency of occurrences of these values are then used to compute an empirical probability distribution to be associated with each link. Choosing a representative QoS parameter when there are multiple paths with different parameters is accomplished by using a Goodness-of-Fit test. Kullback-Leibler distance test finds the least distance estimate of the probability distribution among the alternates. The discrete random variable of the probability distribution of each link serves as the QoS parameters in routing information disseminations. This is the only probabilistic TA proposal to the best of our knowledge. The paper only assumed a FM TA method. How and under what circumstances should the aggregation algorithm rerun is addressed. It would be interesting to see the behavior of this approach for TA methods other than FM of Figure 9. Another extension might be to address more than one QoS parameters.

4.3.5   *Comparison Table of the Epitome Selection Algorithms.* Table II shows the possible alternatives for choosing the QoS parameter value (i.e. epitome) to use in the aggregated topology when more than one path with different QoS parameter values exist in the physical topology. The first column is the method by which a decision is made among many alternative paths with respect to the QoS parameter. The second and the third columns show the number and the type of the QoS parameters, respectively.

4.4   A Macro Level Example for TA Taxonomies

To sum up the taxonomies presented earlier, we use the [Lui et al. 2004]'s proposal as an example: The complicated topology of the underlying physical and hierarchical network is first transformed into a Full-Mesh of border nodes. Then, the FM is reduced to an asymmetric star with bypasses. These first two steps basically form the Structural TA phase, as explained in Section 4.2. As for assigning QoS parameter values to the spokes of the star structure, line fitting in geometric category

Table II.   Methods to choose a representative QoS parameter(s) among many alternative paths.

Selection Method		Number of QoS Parameters	Type of QoS Parameters
Optimal		Single	A/R
Parameter Mix		Multiple	Any Combination
Geometric	Line	Double	R and A
Geometric	Polyline	Double	R and A
Geometric	Spline	Double	R and A
Geometric	Curve	Double/Multiple	R and A
Statistical	Arithmetic Average	Single	R/A
Statistical	Geometric Average	Single	R/A
Statistical	Probabilistic	Single	R/A

(Section 4.3) is used by means of a least-square algorithm.

4.5   Simulation Considerations

Simulation is the most common tool to gauge the performance of the TA techniques. The precision or the amount of lost information due to TA has been the focus of the simulations. Two important metrics in this respect are *Success Ratio*, the percentage of successfully admitted requests out of all attempts, and *Crankback Ratio*, the ratio or percentage of unsuccessful ones over all attempted requests.

5.   RE-AGGREGATION MECHANISMS

In this section, we consider the impact of re-aggregation choices on the performance of the system. Due to network dynamics, i.e. change in the topology and the QoS parameters, it is inevitable that the aggregated state information becomes stale after the initial invocation of the aggregation algorithm. In order to disseminate more up-to-date topology information, a re-aggregation policy should also be specified as part of a holistic TA scheme. Further, initial aggregation is only conducted once while re-aggregation needs to be carried out continuously. Thus, re-aggregation is a very critical component of TA schemes. Yet, only a very limited number of studies have taken re-aggregation into consideration.

Figure 27 shows the categorization of the decisions with respect to the re-aggregation task. The first decision is to set a policy as to when to trigger the re-aggregation. The classification on triggering methods in Figure 27, and explained below, are mostly based on the one in [Apostolopoulos et al. 1998]. *Threshold based* policy triggers re-aggregation when the relative difference between the current and previously advertised aggregated information exceeds a certain threshold, usually a percentage value. The simulation in [Chang and Hwang 2001] concludes that threshold based re-aggregation policy with hysteresis produces the most accurate aggregated information. *Class based* policy breaks up the QoS parameter into classes and triggers when a class boundary is crossed. Fixed-size classes or exponentially distributed
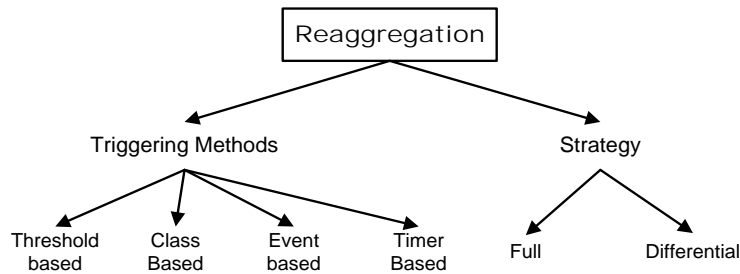
Fig. 27.   Classification of segregation mechanisms.

ranges may be used. [Awerbuch et al. 1998] used a logarithmic update (i.e. exponentially distributed class sizes) re-aggregation policy and the simulation in their paper confirmed its effectiveness. The authors in [Awerbuch et al. 1998] give an example of a link with a data rate capacity of 16 with re-aggregation triggers when link utilization reaches the values of 8, 12, 14, 15, and 16. *Timer based* ([Hao and Zegura 2000]) invokes the aggregation algorithm periodically to ensure a uniform spacing between successive executions. *Event based* ([Iliadis and Scotton 1999]) triggers re-aggregation after topological changes, such as links or routers going up or down.

Another aspect of re-aggregation is how to do it when the time for re-aggregation comes. The easiest option is to run the full aggregation algorithm. However, the computational overhead may be prohibitive or at least expensive. The TA technique used in [Iliadis and Scotton 1999] (transition matrix with a graph coloring algorithm) is amenable to partial or differential updates under some conditions when re-aggregation is necessary without full-blown invocation of the algorithm. Another differential update is proposed for a linear programming solution of TA mechanism specific to ATM PNNI standard in [Iwata et al. 1998].

Since the overall system performance is highly dependent upon the re-aggregation choice, it is highly suggested to avoid choosing any TA mechanism without considering its tradeoff with respect to the re-aggregation policy options.

6. CONCLUSION

We have analyzed and compared topology aggregation techniques from the literature. Our focus was to provide a methodology and its taxonomies to evaluate these techniques with respect to their implications on QoS routing. We believe that some part(s) of the TA techniques or a holistic approach with TA in mind would likely to contribute positively to scalability features of the QoS routing protocols and algorithms. Especially, interdomain routing in the Internet looks a very good candidate for TA techniques as the size in different dimensions, such as host count, AS count, etc., continue to expand unabated. A clear understanding of these techniques is essential to allow us to tackle the very difficult problems posed by the continuing increase in complexity in Internet interdomain and inter-AS routing. Further, the domain of application for TA techniques is not limited to the Internet. Whenever structural information and/or complexity reduction is needed on a network with

implicit or explicit hierarchy in the physical layout, structural TA techniques may be used. Optimal link or path weight (or QoS parameter) representation under multiple alternatives may utilize one of the epitome selection algorithms presented in our study.

Overall, the key is the tradeoff between performance and accuracy of results. More studies of these tradeoffs are needed to shed light to determine the optimal point of balance.

ACKNOWLEDGMENTS

REFERENCES

Ahuja, R. K., Magnanti, T., and Orlin, J. 1993. *Network Flows: Theory, Algorithms and Applications*, First ed. Prentice Hall.

Alaettinoglu, C. and Shankar, A. U. 1995. The Viewserver Hierarchy for Interdomain Routing: Protocols and Evaluation. *IEEE Journal on Selected Areas in Communications 13,* 8 (October), 1396–1410.

Althöfer, I., Das, G., Dobkin, D., Joseph, D., and Soares, J. 1993. On sparse spanners of weighted graphs. *Discrete & Computational Geometry 9,* 1, 81–100.

Apostolopoulos, G., Guerin, R., Kamat, S., and Tripathi, S. K. 1998. Quality of service based routing: A performance perspective. In *SIGCOMM.* 17–28.

ATM Forum. 1996. Private Network-Network Interface Specification version 1.0.

ATM Forum. 2002. Private Network-to-Network Interface Specification version 1.1. *ftp://ftp.atmforum.com/pub/approved-specs/af-pnni-0055.002.pdf* .

Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and Xiao, X. 2002. Overview and Principles of Internet Traffic Engineering. *Request for Comments (RFC 3272).*

Awerbuch, B., Du, Y., Khan, B., and Shavitt, Y. 1998. Routing through networks with hierarchical topology aggregation. Tech. Rep. 98-16, DIMACS. March.

Awerbuch, B. and Shavitt, Y. 2001. Topology aggregation for directed graphs. *IEEE/ACM Transactions on Networking 9,* 1 (February).

Baker, F. 1995. Requirements for IP Version 4 Routers. *Request for Comments (RFC 1812).*

Baran, P. 1964. On Distributed Communications. *RAND Memorandum* . A series of eleven papers for work started in 1961. http://www.rand.org/publications/RM/baran.list.html.

Baswana, S. and Sen, S. 2003. A simple linear time algorithm for computing a $(2k-1)$-spanner of $O(n^{1+1/k})$ size in weighted graphs. In *Automata, languages and programming.* Lecture Notes in Comput. Sci., vol. 2719. Springer, Berlin, 384–396.

Bauer, D., Daigle, J., Iliadis, I., and Scotton, P. 2000. Efficient frontier formulation for additive and restrictive metrics in hierarchical networks. *IEEE ICC*, 1353–1359.

Behrens, J. and Garcia-Luna-Aceves, J. J. 1998. Hierarchical Routing Using Link Vectors. *IEEE INFOCOM '98 San Francisco, CA*, 702–710.

Bradner, S. and Mankin, A. 1995. The Recommendation for the IP Next Generation Protocol. *Request for Comments (RFC 1752).*

Burleigh, S., Hooke, A., Torgerson, L., Fall, K., Cerf, V., Durst, B., Scott, K., and Weiss, H. 2003. Delay-Tolerant Networking: An Approach to Interplanetary Internet. *IEEE Communications Magazine*, 128–136.

Cai, L. 1994. NP-completeness of minimum spanner problems. *Discrete Applied Mathematics. Combinatorial Algorithms, Optimization and Computer Science 48,* 2, 187–194.

CAI, S., GAO, L., GONG, W., AND XU, W.-Q. 2004. On generating internet hierarchical topology. In *Proc. of IEEE Conference on Decision and Control*.

CASTINEYRA, I., CHIAPPA, N., AND STEENSTRUP, M. 1996. The Nimrod routing architecture. *RFC1992*.

CHANG, B. AND HWANG, R. 2001. Efficient Hierarchical QoS Routing in ATM Networks. *Computer Communications 24,* 16 (October), 1648–1660.

CHEN, S. AND NAHRSTEDT, K. 1998. An Overview of Quality of Service Routing for Next-Generation High-Speed Networks: Problems and Solutions. *IEEE Network Magazine*, 64–79.

CHEW, P. 1986. There is a planar graph almost as good as the complete graph. In *Proceedings of the second annual symposium on Computational geometry*. ACM Press, 169–177.

CLARK, D., BELLOVIN, S., BRADEN, B., CHIAPPA, N., FABER, T., FALK, A., HANDLEY, M., SHENKER, S., SOLLINS, K., AND WROCLAWSKI, J. 2002. NewArch: A New Architecture for an Internet. *Report to Next Generation Internet PI Meeting*. Work In Progress.

CORMEN, T., LEISERSON, C., RIVEST, R., AND STEIN, C. 2001. *Introduction to Algorithms*, Second ed. McGraw Hill/MIT Press, Cambridge, MA.

DAVIES, D. 1965. Packet Switching Proposal. *Proposal and Prototype Implementation at UK National Physical Laboratories*. Proposal and Prototype Implementation of packet-switching concepts.

DORIA, A. AND DAVIES, E. 2006. Analysis of IDR requirements and History. *Internet Draft, <draft-irtf-routing-history-03.txt>*. Work In Progress.

DORIA, A., DAVIES, E., AND KASTENHOLZ, F. 2006. Requirements for Inter Domain Routing. *Internet Draft, <draft-irtf-routing-reqs-06.txt>*. Work In Progress.

ESTRIN, D., REKHTER, Y., AND HOTZ, S. 1992. Scalable Inter-Domain Routing Architecture. *ACM SIGCOMM '92 Baltimore, MD*, 40–52.

FEKETE, S. P. AND KREMER, J. 2001. Tree spanners in planar graphs. *Discrete Appl. Math. 108,* 1-2, 85–103. International Workshop on Graph-Theoretic Concepts in Computer Science (Smolenice Castle, 1998).

GARCIA-LUNA-ACEVES, J. AND BEHRENS, J. 1995. Distributed, Scalable Routing Based on Vectors of Link States. *IEEE JOURNAL on Selected Areas in Communication 13,* 8 (October), 1383–1395.

GAREY, M. R. AND JOHNSON, D. S. 1979. *Computers and intractability : a guide to the theory of NP-completeness*. W. H. Freeman, San Francisco.

GE, Z., FIGUEIREDO, D. R., JAISWAL, S., AND GAO, L. 2001. Hierarchical structure of the logical Internet graph. In *Proc. SPIE Vol. 4526, p. 208-222, Scalability and Traffic Control in IP Networks, Sonia Fahmy; Kihong Park; Eds.*, S. Fahmy and K. Park, Eds. 208–222.

GHOSH, D. AND ACHARYA, R. 2001. A probabilistic scheme for hierarchical qos routing. *IEEE ICON*, 416–421.

GUO, L. AND MATTA, I. 1998. On state aggregation for scalable qos routing. *IEEE Proceedings of the ATM Workshop '98*, 306–314.

HAO, F. AND ZEGURA, E. W. 2000. On scalable QoS routing: Performance evaluation of topology aggregation. *INFOCOM*, 147–156.

HUSTON, G. 2001. Commentary on Inter-Domain Routing in the Internet. *Request for Comments (RFC 3221)*.

HUSTON, G. 2006. BGP Table Statistics and Reports. *http://bgp.potaroo.net/*.

ILIADIS, I. 2000. Optimal pnni complex node represenations for restrictive costs and minimal path computation time. *IEEE/ACM Transactions on Networking 8,* 4 (August).

ILIADIS, I. AND SCOTTON, P. 1999. Transition matrix generation for complex node representation. *IEEE ATM Workshop '99, Kochi, Japan*.

ISO/IEC10589. November 2002. IS-IS, Intermediate system to Intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473, also republished as rfc1142). *ISO/IEC 10589, http://isotc.iso.ch/livelink/livelink/ fetch/2000/2489/Ittf_Home/PubliclyAvailableStandards/s018673e.pdf*.

IWATA, A., SUZUKI, H., IZMAILOV, R., AND SENGUPTA, B. 1998. QoS Aggregation Algorithms in Hierarchical ATM Networks. *IEEE ICC'98, Atlanta*.

KAMOUN, F. AND KLEINROCK, L. 1979. Stochastic Performance Evaluation of Hierarchical Routing for Large Networks. *Computer Networks 3,* 5 (November), 337–353.

KLEINROCK, L. 1961. Information Flow in Large Communication Nets. *MIT Proposal for PhD Thesis* . His thesis was published in 1962. Available: http://www.lk.cs.ucla.edu/LK/Bib/REPORT/PhD/.

KLEINROCK, L. AND KAMOUN, F. 1977. Hierarchical routing for large networks: Performance evaluation and optimization. *Computer Networks 1,* 3 (January), 155–174.

KLEINROCK, L. AND KAMOUN, F. 1980. Optimal Clustering Structures for Hierarchical Topological Design of Large Computer Networks. *Networks 10,* 3 (Fall), 221–248.

KORKMAZ, T. AND KRUNZ, M. 1999. Source-oriented topology aggregation with multiple qos parameters in hierarchical atm networks. *7th IWQoS, London*.

KORKMAZ, T. AND KRUNZ, M. 2000. Source-oriented topology aggregation with multiple QoS parameters in hierarchical networks. *ACM Transactions on Modeling and Computer Simulation 10,* 4 (October), 295–325.

LEE, W. C. 1995a. Spanning tree method for link state aggregation in large communication networks. *INFOCOM*, 297–302.

LEE, W. C. 1995b. Topology aggregation for hierarchical routing in atm networks. *ACM SIGCOMM Computer Communications Review, Special Issue on ATM 25,* 2 (April), 82–92.

LEE, W. C. 1999. Minimum equivalent subspanner algorithms for topology aggregation in atm networks. *ICATM '99. 2nd International Conference on ATM*, 351–359.

LIU, J., NIU, Z., AND ZHENG, J. 2000. Parameter Dimensioning Algorithms of the PNNI Complex Node Model with Bypasses. *IEICE Transaction on Communications E83-B,* 37 (March), 638–645.

LUI, K., NAHRSTEDT, K., AND CHEN, S. 2000. Hierarchical QoS routing in delay-bandwidth sensitive networks. *Proceedings of IEEE LCN, Tampa, FL*, 176–189.

LUI, K.-S., NAHRSTEDT, K., AND CHEN, S. 2004. Routing with topology aggregation in delay-bandwidth sensitive networks. *IEEE/ACM Trans. Netw. 12,* 1, 17–29.

MIEGHEM, P. V., KUIPERS, F., KORKMAZ, T., KRUNZ, M., CURADO, M., MONTEIRO, E., MASIP-BRUIN, X., SOLE-PARETA, J., AND SANCHEZ-LOPEZ, S. 2003. *Quality of Service Routing.* Chapter 3 in Quality of Future Internet Services, EU-COST 263 Final Report, edited by Smirnov et al. in Springer LNCS 2856, pp. 80-117.

MOY, J. April 1998. Open Shortest Path First Version 2 (OSPFv2). *Request for Comments 2328, http://www.ietf.org/rfc/rfc2328.txt*.

MUKHERJEE, B. 1997. *Optical Communication Networks*, First ed. McGraw Hill, .

PELEG, D. AND ULLMAN, J. D. 1987. An optimal synchronizer for the hypercube. In *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing.* ACM Press, 77–85.

REKHTER, Y. AND LI, T. 1995. A Border Gateway Protocol 4 (BGP-4). *Request for Comments (RFC 1771)*.

SARANGAN, V. AND ACHARYA, R. 2001. Using network flows in hierarchical qos routing. *IEEE Ninth International Conference On Networks (ICON)*, 422–427.

SARANGAN, V., GHOSH, D., AND ACHARYA, R. 2002. "state aggreagtion using network flows for stochastic networks". *IEEE Globel Communications Conference GLOBECOM 2002"*, 2430–2434.

SARANGAN, V., GHOSH, D., AND ACHARYA, R. 2004. Performance Analysis of Capacity-aware State Aggregation for Inter-domain QoS Routing. In *IEEE Global Telecommunications Conference, GLOBECOM 2004.* Dallas, Texas.

STREENSTRUP, M. 1993. An Architecture for Inter-Domain Policy Routing. *Request for Comments (RFC 1478)*.

SUBRAMANIAN, L., AGARWAL, S., REXFORD, J., AND KATZ, R. H. 2002. Characterizing the internet hierarchy from multiple vantage points. In *Proc. of IEEE INFOCOM 2002, New York, NY*.

Tang, Y. and Chen, S. 2004. QoS information approximation for aggregated networks. In *IEEE International Conference on Communications, ICC 2004*. Paris, France, 2107–2111.

Tsai, W., Ramamoorthy, C., Tsai, W., and Nishguchi, O. 1989. An Adaptive Hierarchical Routing Protocol. *IEEE Transactions on Computers 30,* 8 (August).

Tsuchiya, P. F. 1988. The Landmark hierarchy: A new hierarchy for routing in very large networks. *Proceedings of ACM SIGCOMM 1988*, 128–134.

Uludag, S., Lui, K., Nahrstedt, K., and Brewster, G. 2005. Comparative Analysis of Topology Aggregation Techniques and Approaches for the Scalability of QoS Routing. Tech. Rep. TR-05-010, DePaul University, School of Computer Science.

Wang, Z. and Crowcroft, J. 1996. QoS Routing for Supporting Multimedia Applications. *IEEE JSAC 14,* 7 (Sep.), 1228–1234.

Yang, X. 2003. Nira: a new internet routing architecture. In *FDNA '03: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*. ACM Press, New York, NY, USA, 301–312.

Yoo, Y., Ahn, S., and Kim, C. 2000. Link state aggregation using a shufflenet in atm pnni networks. *IEEE GLOBECOM, San Francisco, CA*, 481–486.

Yoo, Y., Ahn, S., and Kim, C. 2001. Topology Aggregation Schemes for Asymmetric Link Parameters in ATM Networks. Tech. Rep. SNU-CE-TR-2001-1, Seoul National University.

Yoo, Y., Ahn, S., and Kim, C. 2004. Topology aggregation schemes for asymmetric link state information. *Journal of Communications and Networks 6,* 1 (March), 46–59.

Younis, O. and Fahmy, S. 2003. Constraint-based routing in the internet: Basic principles and recent research. *IEEE Communications Surveys & Tutorials 5,* 1.

Yu, J. 2000. Scalable Routing Design Principles. *Request for Comments (RFC 2791)*.