Comparative Analysis of Topology Aggregation Techniques and Approaches for the Scalability of QoS Routing

Suleyman Uludag*
School of Computer Science
DePaul University
suludag@cs.depaul.edu

King-Shan Lui
Department of Electrical
and Electronic Engineering
University of Hong Kong
kslui@eee.hku.hk

Klara Nahrstedt
Department of Computer Science
University of Illinois Urbana-Champaign
klara@cs.uiuc.edu

Greg Brewster
School of Computer Science
DePaul University
brewster@cs.depaul.edu

May 12, 2005

Abstract

In this survey, we study and compare topology aggregation techniques pertaining to QoS routing. Aggregation techniques have been an explicit by design or implicit part of many routing protocols, including the currently deployed ones on the Internet due to scalability. Topology Aggregation, defined as those techniques to abstract or summarize the sate information to be exchanged, processed and maintained by network nodes for routing purposes, have not been studied extensively except under a rather limited context. Under the continuing growth of the Internet, scalability issues of routing, and QoS routing in particular, have been gaining more importance than ever. With this in mind, we are surveying the TA techniques from the literature. Many of the techniques of TA, if not in its entirety, seems to be relevant to current and future IP networks, especially when the very active research area of interdomain routing is considered.

*Contact Author

Contents

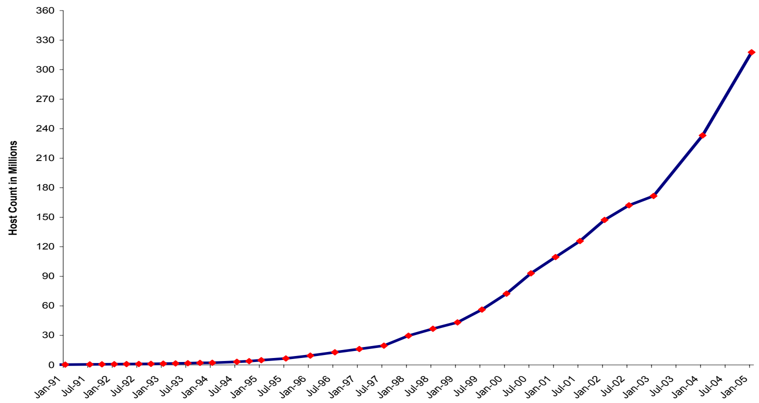

Figure 1: Internet Domain Survey Host Count History as of January 2005. Source:Internet Systems Consortium (www.isc.org).

1 Introduction

Soon after the conception of the principles of packet-switching data networks in the early 1960s [1], [2] and [3], delivery of packets to 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 (InterPlaNet)*[1] [4].

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 January of 2005.

All of these factors keep the research on legacy routing and QoS routing as a vital area [5]. This has led to many studies. A very good and systematic taxonomy and analysis of QoS routing[2] is given in [6]. In what follows we narrow our focus to scalability by means of topology aggregation within the general QoS 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.

[1] See http://www.ipnsig.org/

[2] Routing will refer to both legacy and QoS routing henceforth.

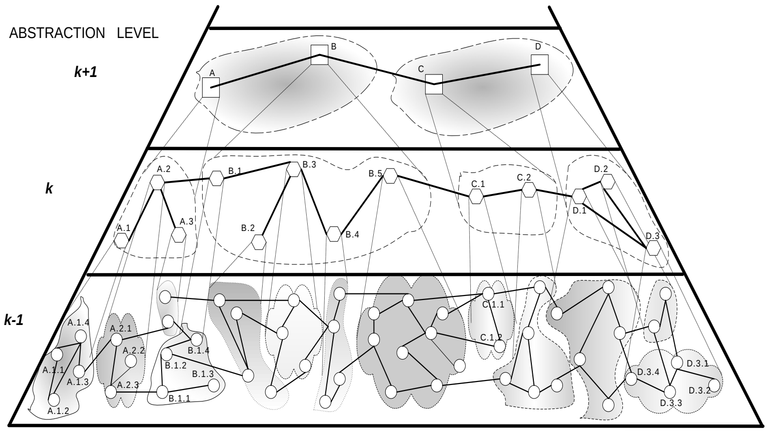

Figure 2: Topology Aggregation in the sense of a pyramidal hierarchical network with Abstraction Level $k-1$, k and $k+1$.

It is this dynamic aspect of the network conditions that requires the exchange, processing,maintenance and storage of state information at each node[3].

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)[4] 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 ($k-1$) 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 k or even level $k+1$, 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 $k-1$) in Figure 2. In graph-theoretic terms, TA schemes are sometimes referred to as *graph compaction* techniques.

This article surveys the previously proposed TA techniques, compares and contrasts them. Section 2 presents the relevance of TA techniques. Section 3 reviews the hierarchical network architectures and structures that facilitate or utilize such aggregation techniques. The network model, notation, assumptions and definitions are outlined in Section 4. The definition, taxonomy and description of TA techniques are given in Section 5 together with the methods about choosing an epitome for QoS parameters when the QoS parameter

[3]Distributed and centralized routing would not obviously require same level of overhead burden on each node. Without loss of generality, we disregard it here temporarily since it only affects the magnitude of the overhead; however, a complete elimination of this overhead is not possible.

[4]http://rr-fs.caida.org/

values of the paths connecting the same nodes differ. Comparison of TA techniques and their synopses are provided in Section 6. Section 7 concludes the discussion.

2 Motivation for Topology Aggregation

Several trends have emerged concerning QoS routing on the Internet. The individual ASs that make up the Internet have become more densely interconnected, as opposed to the tree structure as envisioned by the design [7]. 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[5] required by the customers. As a result, the number of registered ASs, the BGP FIB[6] size and the total advertised IP address space are on the rise [7, 10]. From the perspective of the routing architectures and algorithms, all of the above boil down to more state information to be maintained[7], greater processing power requirements and more bandwidth needed to exchange the routing updates. These overhead factors are putting a strain on the scalability properties of the Internet routing infrastructure [11, 12]. In addition to the problem of scalability, security and commercial confidentiality of the internal layouts of the ASs and domains or subnetworks within the ASs are considered to be essential requirements of the future generation of routing architectures and protocols [7]. Topology aggregation has been proposed as a solution for problems similar to the above, but under different design paradigms [13, 14]. The first such proposal, the Nimrod Architecture [13], 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* [15]. 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[8]. 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. One such newly surfacing idea is the realization of the inevitability of aggregation and thereby *Map/ Abstraction Routing*, which is listed as one of the current projects by the NewArch initiative [5]. Considering these new initiatives and the active research in the field of inter-domain routing, we survey the literature on topology aggregation techniques with a comparison at the end. We believe that the holistic approach to TA as well as the individual algorithms will be be useful for future routing algorithms and protocols, especially for the interdomain routing.

3 Hierarchical Network Routing Architectures

In this section, we briefly survey the routing architectures; those with an explicit hierarchy built-in by design and those with hierarchy either added *ad hoc* or implicitly. Our objective is to show that all routing

[5]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* [8].

[6]Forwarding Information Base (FIB) is defined in RFC1812 [9] as the table containing the information necessary to forward IP Datagrams. It is the table that contains the state information such as the interface identifier and next hop information for the reachable destinations.

[7]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.

[8]The following quotation verbatim from [7] 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.*

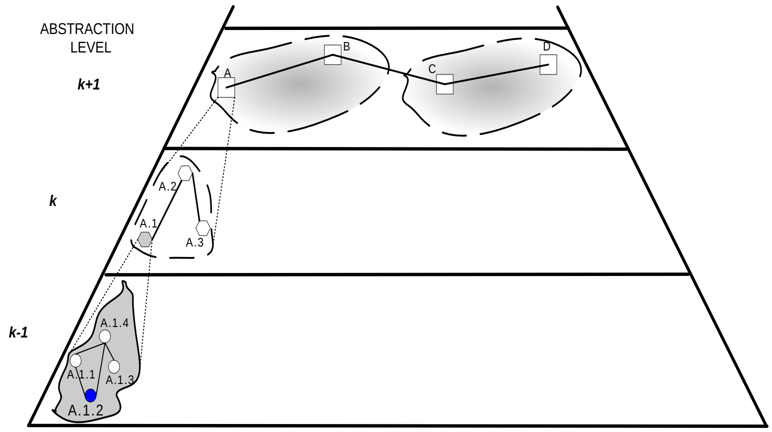

Figure 3: Node A.1.2's view of the network topology for Figure 2's network.

architectures, including the currently used ones in practice, have the notion of hierarchy; *sine qua non* of TA techniques.

Hierarchy is considered to be one of the key routing design principles for scalability [16]. A hierarchically organized network is one whose physical and/or logical layout follows a well-defined structure with multiple levels of abstraction from 1 to m. The main motivation behind it is the principle of information hiding and thereby reducing 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 [17] due to the lack of complete information to calculate the optimal path. Multi-homing and peerings that violate the hierarchy are techniques used in practice by Internet Service Providers (ISPs) to minimize the negative impact of longer path lengths. Figure 2 depicts a hierarchical network design with three of its abstraction levels shown. At each hierarchy level k (where $k \in \{2, ..., m\}$ and m is the *depth* of the hierarchy), aggregated or summarized topology state information about levels *1* to *k-1* is used. 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 for this problem. The main idea, for any node, is to keep more complete routing information about network nodes in terms of a nearness criteria[9] 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. $A.1$). All the rest of the information it maintains for routing purposes is aggregated even about the nodes at its level such as $A.2$, $A.3$, etc. The seminal work for hierarchical networks from queuing-theoretic perspective was carried out by Kamoun and Kleinrock [17, 18, 19]. Many others followed up with different architectures: Adaptive Hierarchical Routing Protocol (AHRP)[20], Landmark Hierarchy[21], Scalable Inter-Domain Routing Architecture(SIDRA)[22], Inter-Domain Policy Routing (IDPR)[23], Viewserver Hierarchy[24], Nimrod Routing Architecture[13], ATM PNNI[25, 14], Area-based Link-Vector Algorithms[26]. Even the most commonly used routing protocols

[9]The most common nearness criteria or metric is a hop distance.

today make use of some sort of hierarchy as part of their critical functionality, such as areas in OSPF[27], levels in IS-IS[28], confederations and route reflectors in BGP[29]. Topology Aggregation techniques exploit the hierarchical infrastructure to lay the ground for scalable routing.

4 Network Model and Notation

In this section, our network model and its corresponding notations are presented. 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.

A set of *domains*[10] constitutes an internet. Let $I(D, L)$ tuple denote a connected internet, where D is

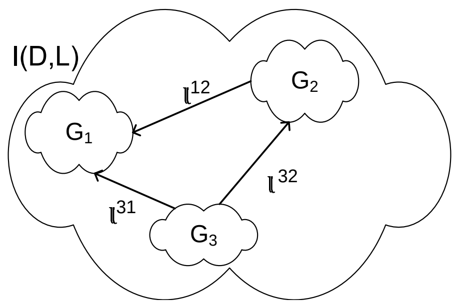

Figure 4: An example internet model.

the set of domains that compose the internet, $D = \{G_i \quad | \quad G_i = (V_i, E_i), \quad \text{where} \quad 1 \leq i \leq |D|\}$, and L is the set of directed, inter-domain links that connect the domains, $L = \{l_{xy}^{ij} \quad | \quad \forall \quad G_i, G_j \in D$ and are connected via border nodes b_x^i and $b_y^j\}$[11]. 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 4.

Each domain is connected and modeled as a tuple $G(V, E)$, where V is the set of vertexes[12] and E is the set of directed edges[13] in the domain. An example domain is depicted in Figure 5 with only a subset of the components marked to bring down the jumble.

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. $B_i \subseteq V_i$ is the set of border nodes of domain G_i which are connected to other domain border nodes via some inter-domain links.

The following are the definitions and notations of the *Physical* Topology:

- $V_i = \{v_1^i, v_2^i, v_3^i, \ldots, v_{|V_i|}^i\} \rightarrow$ 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 5, $V_1 = \{v_1^1, v_2^1, v_3^1, v_4^1, v_5^1, v_6^1, v_7^1, v_8^1\}$ or simply $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$.

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

[10] 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*.

[11] Wherever appropriate, the superscripts or subscripts may be omitted, such as l^{ij} or l_{xy}.

[12] Vertex and node will be used interchangeably for the rest of the paper

[13] The terms edge and intra-domain links are also used interchangeably

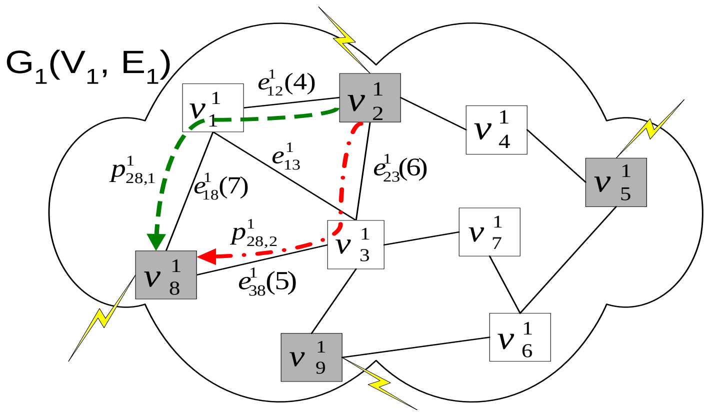

Figure 5: An example internet domain model.

- $e^i_{jk} \rightarrow$ physical intra-domain link from node v_j to v_k in domain G_i

- $E_i = \{e^i_{jk} \quad | \quad \forall v_j, v_k \in V_i \quad \text{which are connected}\} \rightarrow$ the set of all links in domain G_i

- $B_i = \{b^i_1, b^i_2, b^i_3, \ldots, b^i_{|B_i|}\} \rightarrow$ the set of all border nodes in G_i. In Figure 5, the border nodes are denoted by shadowed squares: $B_1 = \{v_2, v_5, v_8, v_9\}$ or $B_1 = \{b_1, b_2, b_3, b_4\}$, where mapping from v_i to b_j is done in ascending order of the node numbers in V.

- $B_i| \rightarrow$ is the number of border nodes in G_i.

- $P^i_{jk} = \{p^i_{jk,1}, p^i_{jk,2}, \ldots, p^i_{jk,|P^i_{jk}|}\} \rightarrow$ the set of all paths from node v_j to node v_k in domain G_i. In Figure 5, two paths from v_2 to v_8 are depicted among other paths of the set p^1_{28}, that is $\{p^1_{28,1}, p^1_{28,2}\} \in p^1_{28}$.

- $|P^i_{jk}| \rightarrow$ the number of distinct paths from node v_j to node v_k in domain G_i

- $p^i_{jk,s}[n] \rightarrow$ the nth link of the sth path from node v_j to node v_k in domain G_i. In Figure 5, $p^1_{28,1}[1] = e^1_{21}$.

- $p^i_{jk,s} = \{p^i_{jk,s}[1], p^i_{jk,s}[2], \ldots, p^i_{jk,s}[|p^i_{jk,s}|]\} \rightarrow$ the set of all links of the sth path from node v_j to node v_k in domain G_i. In Figure 5, the first path from node v_2 to node v_8 has two links, i.e. $p^1_{28,1} = \{p^1_{28,1}[1], p^1_{28,1}[2]\} = \{e^1_{21}, e^1_{18}\}$.

- $|p^i_{jk,s}| \rightarrow$ the number of links of the sth path from node v_j to node v_k in domain G_i

- $P_i = \{P^i_{jk} \quad | \quad \forall v_j, v_k \in V_i\} \rightarrow$ the set of all physical paths in G_i

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

- $q^r_{e^i_{jk}} \rightarrow r$th QoS parameter of link e^i_{jk}, where $1 \leq r \leq m = |Q|$
  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 are given below:

  1. If the QoS parameter is *restrictive*[14] then the minimum (or maximum) value of the links that compose the path number s between node v_j and v_k determine the overall end-to-end QoS parameter for the corresponding path:

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

     for all restrictive QoS parameters r. Bandwidth is an example of a restrictive QoS parameter. In Figure 5, for the simplicity reason of the 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*[15] 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 5 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 5 we assume that the edges are labeled with packet loss ratios in percentages then we calculate QoS path parameter as follows:

     $$\begin{aligned} q^1_{p^1_{28,1}} = (1 - \prod_{t=1}^{|p^1_{28,1}|} (1 - q^1_{p^1_{28,1}[t]})) &= (1 - [(1 - q^1_{p^1_{28,1}[1]}) \cdot (1 - q^1_{p^1_{28,1}[2]})]) \\ &= (1 - [(1 - 0.04) \cdot (1 - 0.07)]) = 10.72\% \end{aligned}$$

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

[14] There are different terms used in the literature to refer to the restrictive parameter. Link constraint [6], link attribute [30], bottleneck, non-additive [31] and concave [32] all refer to the same phenomenon. We adopt restrictive [33] to refer to it.

[15] Similarly, convex, path attribute [30], path constraint [6] and additive [32] are used interchangeably in the literature.

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:

$$q^r_{p^i_{jk}} = \coprod (\forall q^r_{p^i_{jk,s}})$$

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 5.2.

$I'(D', L')$ 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 5 is illustrated in Figure 6.

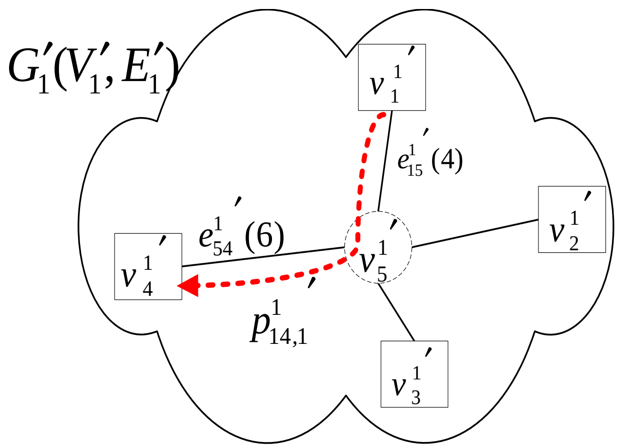

Figure 6: A transformed graph of Figure 5 by means of a simple star topology. Note that the transformed or logical topology consists of the border nodes of the physical topology of Figure 5 and a fictitious node $v^1_5{}'$
.

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

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

- $L' = \{l^{ij}_{xy}{}' | \quad \forall G_i{}', G_j{}' \in D$ and connected via border nodes $b^i_x{}'$ and $b^j_y{}'\} \rightarrow$ the set of all logical inter-domain links

- $v^i_j{}' \rightarrow$ logical node j of domain G_i

- $V_i{}' = \{v^i_1{}', v^i_2{}', v^i_3{}', \ldots, v^i_{|V_i|}{}'\} \rightarrow$ the set of all logical nodes (vertexes) in $G_i{}'$. In Figure 6, $V_1{}' = \{v^1_1{}', v^1_2{}', v^1_3{}', v^1_4{}', v^1_5{}'\}$. Note that the first four nodes of $V_1{}'$ are the border nodes from Figure 5, i.e. $v^1_1{}' = v^1_2 = b1, v^1_2{}' = v^1_5 = b2, v^1_3{}' = v^1_9 = b3, v^1_4{}' = v^1_8 = b4$. The last node $(v^1_5{}')$ is the fictitious node of the transformed star topology.

- $e^i_{jk}{}' \rightarrow$ logical intra-domain link from node $v_j{}'$ to node $v_k{}'$ in domain G_i

- ${E_i}' = \{{e^i_{jk}}' \quad | \quad \forall {v_j}', {v_k}' \in {V_i}' \quad \text{which are connected}\} \rightarrow$ the set of all logical intra-domain links in G_i

- $|{p^i_{jk}}'| \rightarrow$ the number of distinct paths from node ${v_j}'$ to node ${v_k}'$ in domain ${G_i}'$

- ${p^i_{jk}}' = \{{p^i_{jk,1}}', {p^i_{jk,2}}', \ldots, {p^i_{jk,|p^i_{jk}|}}'\} \rightarrow$ the set of all paths in transformed topology from node ${v_j}'$ to node ${v_k}'$ in domain ${G_i}'$. In Figure 6, there is only one path from ${v_1}'$ to node ${v_4}'$ due to the peculiar features of the star topology, i.e. ${p^1_{14}}' = \{{p^1_{14,1}}'\}$ which is denoted by dashed line in the figure.

- $|{p^i_{jk,r}}'| \rightarrow$ the number of links of rth path from node ${v_j}'$ to node ${v_k}'$ in domain ${G_i}'$

- $p^i_{jk,r}[n] \rightarrow$ the nth link of the rth path from node v_j to node v_k in domain G_i. In Figure 5, $p^1_{28,1}[1] = e^1_{12}$.

- ${p^i_{jk,r}}' = \{{p^i_{jk,r}}'[1], {p^i_{jk,r}}'[2], \ldots, {p^i_{jk,r}}'[|{p^i_{jk,r}}'|]\} \rightarrow$ the set of all links of the rth path from node ${v_j}'$ to node ${v_k}'$ in domain ${G_i}'$. In Figure 6, the path from node ${v_1}'$ to node ${v_4}'$ has two links, i.e. ${p^1_{14,1}}' = \{{p^1_{14,1}}'[1], {p^1_{28,1}}'[2]\} = \{{e^1_{15}}', {e^1_{54}}'\}$.

- ${P_i}' = \{{p^i_{jk}}' \quad | \quad \forall {v_j}', {v_k}' \in {V_i}'\} \rightarrow$ the set of all logical paths in ${G_i}'$

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

- ${q^r_{e^i_{jk}}}' \rightarrow r$th QoS parameter of link $\quad {e^i_{jk}}', \quad$ where $\quad 1 \leq r \leq |Q'|$

5 Topology Aggregation

Topology aggregation (TA) may be defined as a series of actions taken to summarize or to abstract 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. A survey of such techniques is presented in the subsequent subsections followed by comparison and complexity analysis.

5.1 Techniques

TA is a spatial abstraction or reduction since it involves bringing down the physical size of the state information to be maintained. Figure 7 depicts a classification of the TA techniques. *Nodal Abatement* simply refers to considering only the border nodes for inter-domain routing and disregarding the other ones. *Link abatement* is 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 $|B_i| = |V'_i| < |V_i|$ and $|E'_i| < |E_i|$. We exemplify the Topology Transformation techniques on a very simple subnetwork shown in Figure 8[16]. The topology

[16]Without loss of generality, our example is an undirected graph, i.e. $\forall e^i_{jk} \in E_i$ and $\forall q^r \in Q$, $q^r_{e^i_{jk}} = q^r_{e^i_{kj}}$, to reduce clutter.

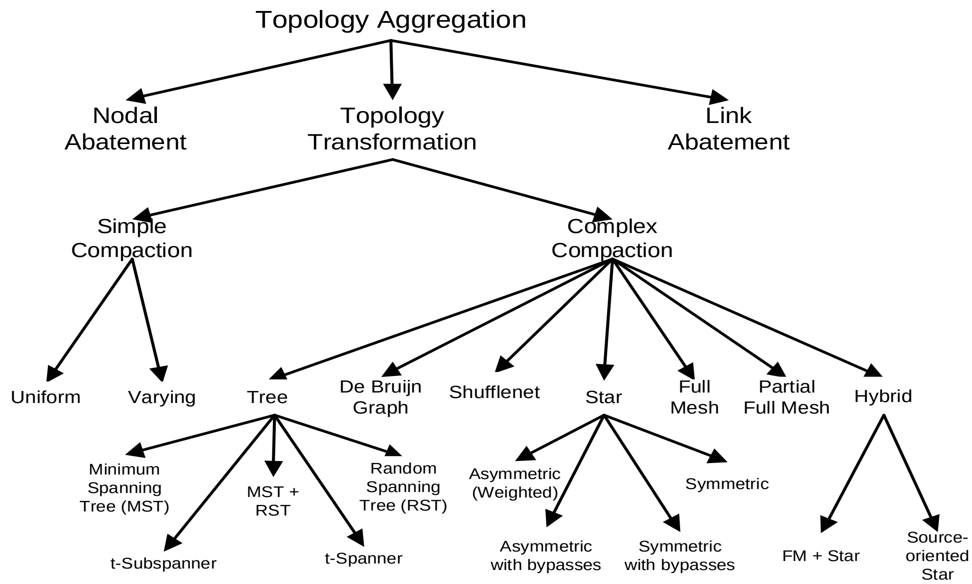

Figure 7: Classification of topology aggregation techniques.

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.

The *Full Mesh (FM)*[17] is the full connectivity among the border nodes. It captures the details of the topology at the expense of more spatial and temporal complexity. Figure 9(a) is the FM representation of our topology based on the maximization of the restrictive QoS parameter and 9(b) is the FM representation based on the minimization of the additive parameter. In each, 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 $A-D-H-G$. Thus, in Figure 9(a), $q^1_{p_{AG}} = 8$ and the corresponding additive parameter of the path $A-D-H-G$ is $q^2_{p_{AG}} = 26$. Similarly, the path with the best or minimum additive parameter from A to G is $A-E-G$ and thus in Figure 9(b), link from A to G has $(q^1_{p_{AG}}, q^2_{p_{AG}}) = (2, 9)$. 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.

Simple Compaction basically collapses the whole subnetwork G_i into a single node. In the *Uniform Simple Compaction*, each border node advertises the same QoS parameter vector[18] to all other subnetworks, whereas in the *Varying Simple Compaction*, the advertised vector may vary from one border node to another. Figure 10(a) shows a possible aggregated topology of Figure 8 based on Uniform Simple Compaction by using the FM from Figure 9(a) with the worst restrictive 4 and worst additive 26 as advertised values. Figure 10(b) is a possible aggregated topology of Figure 8 based on Varying Simple Compaction by using the FM from Figure 9(a) 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 9(a) $((4, 15), (8, 17), (8, 9))$ and the worst additive value out of H $(4, 17)$ as shown in Figure 10(b). Note that, in Figure 10, 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 5.2. Obviously, the Simple Compaction approach suffers from inaccurate representations, since uniformity across the domain or through the border

[17]We start off with the Full Mesh since many other TA techniques use it as the first step and it will help us explain our subsequent examples.

[18]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

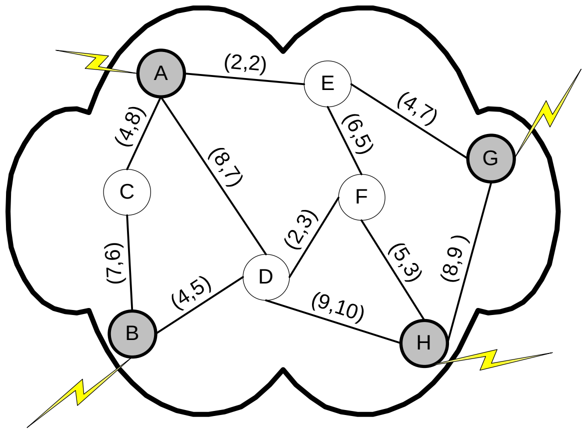

Figure 8: A simple subnetwork to illustrate the topology transformation schemes. Link QoS parameters (q^1, q^2) denotes the restrictive (e.g. bandwidth) and additive (e.g. delay) QoS parameters, respectively.

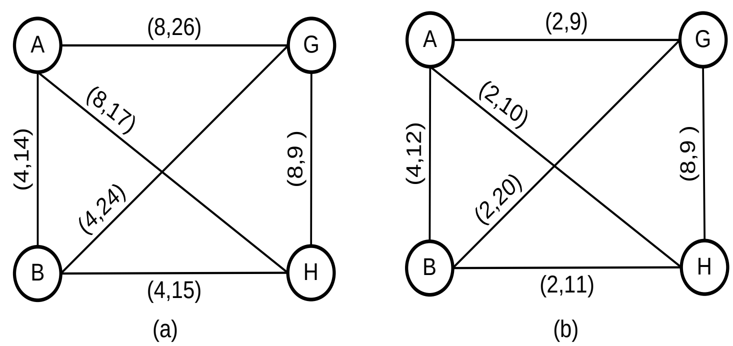

Figure 9: FM representations of Figure 8. (a) FM based on the restrictive parameter maximization. (b) FM based on the additive parameter minimization.

node is assumed, which is rarely valid.

The *Complex Compaction* is a set of more sophisticated, yet more accurate, representations of the subnetwork.

The *Partial FM*, introduced in [34, 35][19], is based on an idea in [26, 36] 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 11 shows the advertised topology of Node A about the subnetwork. 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.

Tree category from our classification scheme in Figure 7 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 required 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

[19] The term the authors used in [35] is *source-oriented*. We use Partial FM to refer to it in this paper.

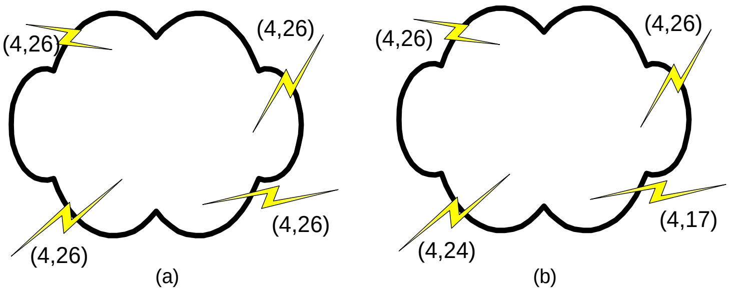

Figure 10: Advertised vectors of Figure 8 under (a) Uniform Simple Compaction, (b) Varying Simple Compaction.

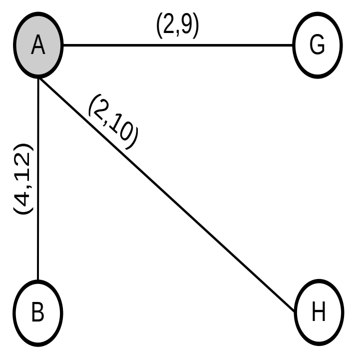

Figure 11: Partial FM representation of Figure 8.

a restrictive parameter among the border nodes [20], as shown in Figure 12(a) or on minimizing an additive parameter, as shown on Figure 12(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.

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 12(c). The running time complexity is $O(E+V)$, which is better than $O(ElogV)$ of MST[37].

3. The *MST and RST* combination, proposed in [38, 39], is simply a union of the constituent elements, as shown in Figure 12(d). The simulations in [39] showed good performance in terms of worst-pair distortion costs compared to other alternatives, where distortion $= \max_{i,j \in V} \frac{q_{p_{ij}}}{q'_{p_{ij}}}$ where $q_{p_{ij}}$ 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. *t-spanner*, first introduced in [40, 41], 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 chosen QoS parameter r. The value of t is referred to as *stretch factor* [21] in the literature. The optimal value for the stretch factor is $t = 1$. Extensions based on Minimum Weight ST algorithms, such as Kruskal's [42], Prim's or Sollin's [43], can be used to find a t-spanner of a graph. A recent, improved algorithm for spanner graphs can

[20] We aim to choose the paths with the maximum restrictive parameter. For example, in Figure 12(a), the maximum restrictive parameter between nodes A and G is 8 based on Figure 9(a)

[21] Note that worst-pair distortion defined earlier is the same the stretch factor

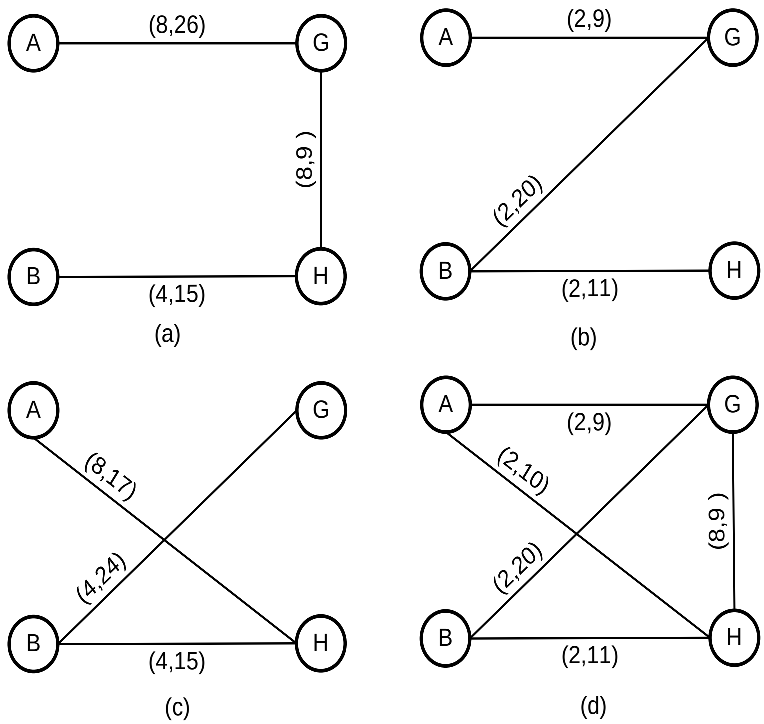

Figure 12: Spanning Trees of Figure 8. (a) Maximum Weight ST based on the restrictive parameter, (b) Minimum Weight ST based on the additive parameter, (c) RST, (d) MST+RST (Union of (a) and (c)).

be found in [44]. A t-spanner, where $t = 32/9$, of the example topology is depicted in Figure 13(a) which is based on the FM from Figure 9(b). It is (32/9)-spanner because worst-pair distortion between Figure 9(b) (the actual topology) and Figure 13(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 [45] 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.

5. *t-subspanner* is introduced in [46] and may be considered as a generalization of the t-spanner. [46] defines t-subspanner as a spanning subgraph $G'(V', E')$ of graph $G(V, E)$ such that $E' \subseteq E$, $V' \subseteq V$ and $\forall e_{jk} \in E$, $q^r_{p^i_{jk,s}} \leq q^r_{p^i_{jk,s}}{}' \leq t \cdot q^r_{p^i_{jk,s}}$ 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. The advantage of the t-subspanner is that it can directly be applied to a full subnetwork without requiring the intermediary step of constructing the FM as the t-spanner does. [46] provides two algorithms, based on Dijkstra and Floyd-Marshall shortest path algorithms [43]. Figure 13(b) shows a (29/12)-subspanner again based on the FM of Figure 9(b) since the worst-pair distortion is between nodes A and B; 29 in the subspanner versus 12 in the actual. Minimum Equivalent Subspanner (MES) is a t-subspanner with minimum number of links where t=1. Figure 14 is an example 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.

The *De Bruijn Graph* [47] and *Shufflenet* [48] topology aggregation schemes are based on the techniques in [49] to represent the FM with better accuracy than a star (discussed below) but with less complexity than the FM itself. Shufflenet organizes the underlying topology logically as a matrix with p^k rows and k columns

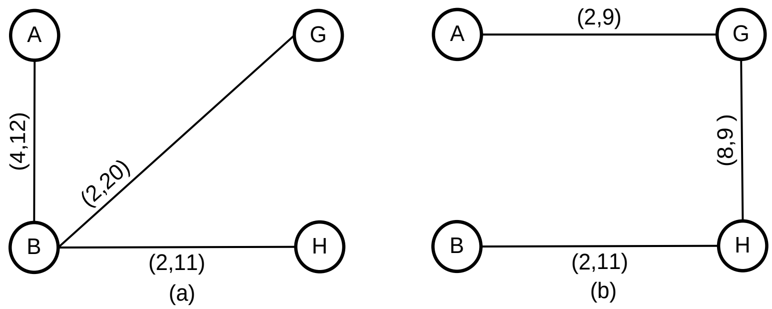

Figure 13: (a) t-spanner of FM based on additive parameter minimization (t=32/9), (b) t-subspanner of the original topology for the border nodes (t=29/12). Both are based on Figure 8.

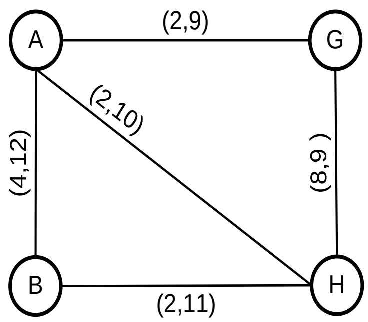

Figure 14: MES of Figure 8.

by picking the parameters p and k such that $|B| \leq k \cdot p^k$ [22]. Each cell of the matrix corresponds uniquely to a physical node. The shufflenet denoted by (p, k) has total number of nodes equal to $N = k \cdot p^k$ and total number of links $p \cdot N$. The unique identifier of each node is given by $(n-1)p^k + l - 1$, where l and n are the row and column numbers, respectively. Each node i has p outgoing links to nodes in the next column identified by $(i \bmod p^{k-1})p + j$, where $j = 1, 2, \ldots, p$. Finally, the nodes in the last column are connected to the first-column nodes in a wrap-around fashion. Figure 15(a) shows the FM of all 8 ($8 \leq 2 \cdot 2^2$) nodes of the network[23]. Figure 15(b) is a transformation of the subnetwork into the shufflenet. A heuristic algorithm is provided by the authors to transform the network into a shufflenet or a De Bruijn graph in terms of a restrictive (bandwidth), additive (hop count) or both. Figure 15(c) depicts the aggregation based on the de Bruijn graph. A de Bruijn graph is identified by two integers, Δ and D, where the total number of border nodes is less than or equal to Δ^D and the total number of links is given by $N\Delta = \Delta^{D+1}$. Each node is identified by a unique number whose format is $a_1 a_2 \ldots a_D$, $a_i \in 0, 1, 2, \ldots, \Delta - 1$. Each node has Δ directed outgoing edges to nodes whose identifiers are given by $b_1 b_2 \ldots b_D$ where $b_i = a_{i+1} \forall i = 1, 2, \ldots, D-1$. The drawback of both shufflenet and de Bruijn graphs are the increased delay (the average hop count rises) and the need to have certain number of nodes for optimum aggregation. For instance, if the number of border nodes is 10 then p and k should be chosen as 2 and 3, respectively, giving $n = 18$, which is greater than number of border nodes needed.

Another complex compaction category is *Star*, as used by the ATM's PNNI [14]. There are 4 kinds of

[22] It is recommended in [50] that when the number of border nodes is not so large a small k is preferred, otherwise a small p is preferred. Further, both p and k should be greater than or equal to 2.

[23] Both shufflenet and de Bruijn aggregations are based on the FM of the border nodes, not all the nodes, as shown in Figure 15. The reason why we chose the total number of nodes, but not the border nodes alone, is to illustrate the concept since with four number of border of nodes it would not be very clear to explain the techniques.

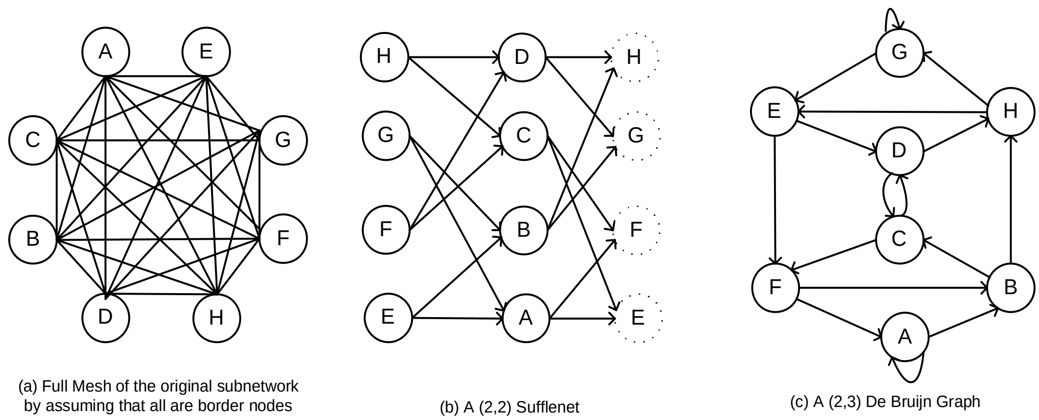

Figure 15: Shufflenet and de Bruijn Graph based aggregations of Figure 8.

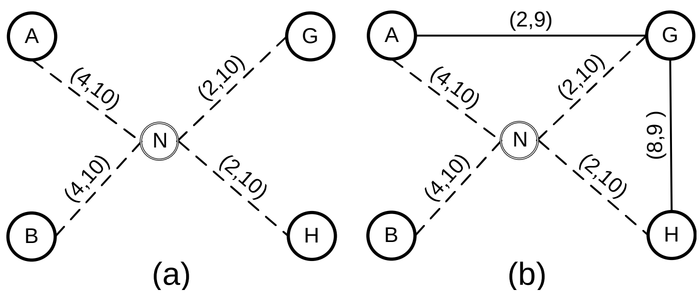

Figure 16: Symmetric Star (a) without bypasses, (b) with bypasses. Both are based on Figure 8, with respect to the additive metric.

star-based aggregations[24]:

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. Figure 16(a) shows an aggregated topology as a symmetric star without bypasses (explained below) based on the additive metric q^2. Note that the star is symmetric only with respect to the additive metric, but not to the the restrictive metric in Figure 16. 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 associate with spokes in Section 5.2.

2. *Symmetric Star with Bypasses* is similar to the symmetric star with the addition of *bypasses*. A bypass or an exception is a direct connection between two border nodes whose connection via the fictitious nucleus grossly deviates from its real FM value. 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 distribution of the QoS parameter values. To cope with and to reduce the inaccuracy, bypasses are inserted. The existence of a bypass will ensure that a more realistic QoS parameter values will result. A symmetric star with bypasses is shown in Figure 16(b).

3. *Asymmetric (Weighted) Star* is a star whose spokes can take on different QoS parameter values. The

[24]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 5.2.

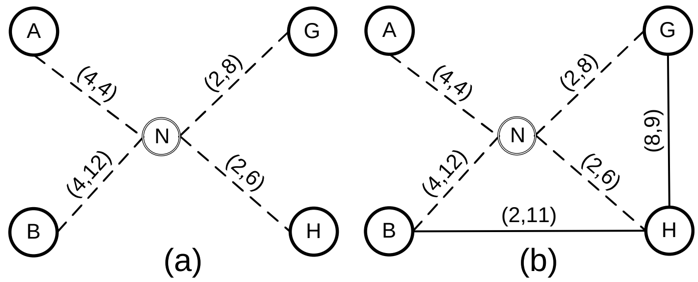

Figure 17: Asymmetric (Weighted) Star (a) without bypasses, (b) with bypasses. Both are based on Figure 8.

asymmetry reflects the underlying heterogeneity of the physical topology. It may be termed as *weighted* to take different criteria into account, such as administrative policies. Figure 17(a) is a depiction of an asymmetric star.

4. The main objective of the *Asymmetric (Weighted) Star with bypasses* is identical to that of the symmetric star with bypasses, i.e. to reduce the inaccuracy. Figure 17(b) is an asymmetric star with bypasses.

The last category of complex compaction we discuss is the *hybrid*, which combines more than one of the above techniques. The hybrid aggregation introduced in [51] differentiates the characteristics of the QoS parameters in terms of their likely change frequencies. 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. Another hybrid scheme, named *Source-oriented Star* by the authors in [34, 35], is simply a union of the symmetric or asymmetric star without bypasses and the partial FM discussed above. The partial FM serves as the bypasses and hence one can consider this similar in essence to the star with bypasses.

5.2 Choosing an Epitome for QoS Parameters

Topology aggregation often requires choosing among multiple paths between nodes. An *Amalgamation Function* performs this task, as introduced briefly in Section 4. For example, in Figure 8, 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. Table 1 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(s). The next three columns show the number of QoS parameters associated with the links. The last three are the types of the QoS parameter, as described in Section 4, that the corresponding methods can use.

- *Best* 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.

- *Arithmetic Average* is the sum of the link QoS parameter values of a path divided by the number of links on the path, i.e. $q_p^{arith} = \frac{1}{|p_{ij}|} \sum_{\forall e \in p_{ij}} q_e$. *Geometric Average* is the product of the link QoS parameter

Representation Method	Number of QoS Parameters			Type of QoS Parameters		
	Single	Double	Multiple	Restrictive	Additive	Multiplic.
Best	√	-	-	√	√	√
Worst	√	-	-	√	√	√
Arithmetic Average	√	-	-	√	√	-
Weighted Average	√	-	-	√	√	√
Geometric Average	√	-	-	-	-	√
Parameter Mix	-	√	√	√	√	√
Use only one	-	√	√	√	√	√
Line Fitting	√	√	-	√	√	√
Curve Fitting	-	√	-	√	√	-
Cubic Splines	-	√	-	√	√	√
Polyline Fitting	-	√	-	√	√	√
Probabilistic	√	√	√	√	√	√

Table 1: Methods to choose a representative QoS parameter(s) among many alternative paths.

values of a path raised to a power equal to the reciprocal of the number of links on the path, i.e. $q_p^{geo} = \sqrt[|p_{ij}|]{\prod_{\forall e \in p_{ij}} q_e}$. Both are only applicable to single-restrictive/additive and single-multiplicative cases. *Weighted Average* can be used for any single QoS parameter types.

- When the number of QoS parameters is greater than one the decision gets harder. One option, i.e. *parameter mix* 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. We can restrict the decision to be based on one parameter only (*Use Only One* in Table 1), choose the representative path value accordingly and find the values of other QoS parameters along the chosen path to advertise. 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.

- The next four options are geometric-representation based; *line fitting* [52], *curve fitting* [35, 34, 53], *cubic splines* [53] and *Polyline Fitting* [53]. Detailed discussions of these techniques are presented in Section 6.2.

- A *Probabilistic* approach has been proposed in [54] and [55]. The basic idea is to associate reliability values with QoS parameter availability. This time series data is then converted into a discrete random variable based on the relative frequency of occurrence to represent the likelihood of finding a QoS resource. This TA scheme then uses the Kullback-Leibler distance as a measure of goodness-of-Fit to choose the Representative value and the path. This is the only probabilistic TA proposal to the best of our knowledge. Again the details are in Section 6.2.

6 Comparison and Synopses of TA techniques

6.1 Comparison of TA Techniques

In this subsection, we compare various TA techniques from the literature in terms of several criteria as shown in Table 2 and Table 3. Table 2 compares the TA techniques alone while Table 3 distinguishes them from the routing perspective including the simulation setting. In both of these tables, the first column provides

Table 2: Comparison of the Topology Aggregation techniques

	Category [a]	Strategy [b]	QoS Parameter: Prob? [c]	QoS Parameter: Sym? [c]	QoS Parameter: Type [d]	Selection Criteria [e]	Precision [f]	Complexity: Time [g]	Complexity: Decode [h]	Reagg. [i]
Awerbuch[38]	MST	C	N	Y	A	Best	N	$O(B^2)$	$O(B)/O(B)/O(B^2)$	Full
Awerbuch[39]	t-spanner	C	N	N	A	Best	N	$O(B^2)$	$O(B)/O(B)/O(B^2)$	Full
Bauer[33]	FM	D/C	N	Y	R and A	Best	Y	$O(B^5)$	NA	Full
Ghosh[54]	FM	D	Y	N	R and A	Prob.	Y	$O(kEB^2)^j$	NA	Partial
Guo[56]	FM,star,simple	D	N	Y	R and A	Worst	Y/N	$O(B^2)/O(B)$	$O(1)/O(B)/O(B^2)$	Full
Sarangan[55]	Vary Simple	D	Y	N	R	Irrelevant	N	$O(VE^2 + nElog(nE))^k$	NA	Full
Iliadis[57]	FM	D	N	Y	R	Best	Y	$O(Elog(E))$	NA	Partial
Iliadis[58]	Sym Star byp.	D	N	Y	R	Best	N	$O(Elog(E) + B)$	$O(1)/O(B)/O(B^2)$	Partial
Chang[59]	Vary Simple	D	N	N	R and A	Best	Y	$O(B(VlogV + E))$	NA	Full
Lee[31, 30]	MST	C	N	Y	R or A	Best	Y/N[l]	$O(B^2(VlogV + E))^m$	$O(B)/O(B)/O(B^2)$	Full
Lee[46]	1-subspanner	D/C	N	N	A	Best	Y	$O(VlogV + E)/O(V^3)^n$	$O(B)/O(B)/O(B^2)$	Full
Lui[52]	Asym Star Byp.	C	N	N	R and A	Line	N	$O(BE^2 + BEVlogV)$	$O(1)/O(B)/O(B^2)$	Full
Korkmaz[35]	Src-ori. Star	C	N	N	multiple	Curve	N	$O(B^2(K + E)(VlogV + E))^o$	$O(1)/O(B)/O(B^2)$	Full
Iwata[60]	Asym star byp.	C	N	Y	R/A	Weighted	N	$O(B^2VE^2)^p$	$O(1)/O(B)/O(B^2)$	Partial
Hao[51]	Hyb./asym.star[q]	D/C	N	Y	R	Best/Ave.	Y/N	NA[q]	NA	Full
Liu[61]	Asym star byp.	C	N	Y	R and A	Best	N	$O(B^2)$	$O(1)/O(B)/O(B^2)$	Full
Yoo[50]	Shufflenet/deBruijn	C	N	Y/N	R	Least dist.	N	$O(B^2 + V^2logN)/O(B^2logV)$	$(\frac{B^3}{pk^2}/O(B^2)^r$	Full
Tang[53]	NA[s]	D	N	Y	R and A	Lines[t]	N	$O(Ek)^t$	NA	Full

E, B are the number of links and the border nodes, respectively, instead of the $|E|$ and $|B|$ to simplify the notation.
[a] Based on the classification presented in Figure 7
[b] D for distributed; C for centralized; D/C for both.
[c] *Prob.* and *Sym* indicate whether the QoS parameter is defined as a random variable and whether it is symmetric, respectively.
[d] Whether the QoS parameter is restrictive (R), additive (A) or multiplicative (M), R/A for either
[e] Based on the classification from Table 1
[f] Y means exact, i.e. if the value used is the same as at least one physical path that it is representing, N is for otherwise or imprecise.
[g] For Centralized algorithms, time needed for a node to compute the whole TA, while for Distributed, time needed for a border node to compute its portion
[h] Decoding is needed only when the TA category is not FM or Simple. *NA* stands for decoding is not needed; otherwise, decoding complexity is presented as Time to decode one logical link / all logical links related to a single border node / all among border nodes
[i] Whether the re-aggregation can be done partially without having to consider the unchanged. If not then full reaggregation
[j] The random variable is discrete with k number of bins
[k] Each link is represented by a pmf of length n
[l] The representation is exact if the parameter is restrictive; otherwise, for the additive case, it only assures an upper bound.
[m] The running time is not provided in the paper. Finding MST takes $O(B^2logB)$ and is dominated by the time to find the FM, which is $O(B^2(VlogV + E))$.
[n] $O(VlogV + E)$ is the complexity for distributed mechanism and $O(V^3)$ is the complexity for centralized mechanism.
[o] K is the number of QoS parameters being considered.
[p] FM dominates the polynomial-time linear-programming formulation. However, the LPF may take exponential time for some input [37].
[q] No algorithms are given for hybrid (FM+star) and weighted star; they should be similar to FM+sym star and asym star, respectively, though.
[r] Only decoding complexity of the whole subnetwork; p and k are shufflenet design parameters.
[s] Only how to compactly represent 2 QoS parameters are discussed; no TA category is mentioned.
[t] Multiple line segments (polyline) are fitted to the data; k is the number corner points on the polyline.

Table 3: Comparison of the Topology Aggregation techniques from routing perspective

	Spatial Complexity		Path Selection	Update Trigger [c]	Simulation Settings	
	Advertise [a]	Inter-AS [b]			Net. Dyn. [d]	Eval. Metric [e]
Awerbuch[38]	$O(B)$	$O(D*B)$	Hierar. Src.	Threshold	Y	SR, CR
Awerbuch[39]	$O(B)$	$O(D*B)$	-	-	No Sim.	Distortion
Bauer[33]	$O(B^2)$	NA	NA	NA	No Sim.	No Sim.
Ghosh[54]	$O(kB^2)^f$	$O(kDB^2+L)$	MRP[g]	Periodic	Y	SR
Guo[56]	$O(B^2)/O(B)/O(1)$	$O(D*B^2)/O(D*B)/O(D)$	Prob.[h]	Periodic	Y	3 metrics[i]
Sarangan[55]	$O(1)$	$O(D+L)$	shortest path	Periodic	Y	SR
Iliadis[57]	$O(B^2)$	NA	NA	Event-based	No Sim.	No Sim.
Iliadis[58]	$O(B)$	NA	NA	Event-based	No Sim.	No Sim.
Chang[59]	$O(1)$	$O(DB+L)$	Min.Cost	Event-based	Y	Weighted BR
Lee[31, 30]	$O(B)$	NA	NA	NA	No Sim.	No Sim.
Lee[46]	$O(B^2)$	NA	NA	NA	No Sim.	No Sim.
Lui[52]	$O(B)$	$O(DB+L)$	LSRA	Periodic	N	Imprecision, CR, SR
Korkmaz[35]	$O(B)$	$O(DB+L)$	CBDRA[j]	Periodic	Y	CR, SR
Iwata[60]	$O(B)$	$O(DB)$	Hierar. Src.[k]	Periodic	N	BR
Hao[51]	$O(B^2)/O(B)$	$O(DB^2)/O(DB)$	Several[l]	Periodic	Y	BR
Liu[61]	$O(B)$	$O(DB+L)$	shortest	Periodic	N	SR, CR
Yoo[50]	$O(pB)/O(\Delta B)$ [m]	$O(pBD+L)/O(\Delta BD+L)$	NA	Periodic	N	SR
Tang[53]	NA	NA	NA	NA	No Sim.	No Sim.

[a] Spatial complexity of the TA information a border node advertises
[b] Spatial complexity of a border node's routing table for inter-domain routing. *NA* means no routing detail is provided in the paper.
[c] It is the event that triggers the TA algorithm to be re-run
[d] *Y* is for considering the network dynamics, such as traffic load, for routing in the simulation, *N* otherwise
[e] *SR* for success or admission ratio, *CR* for crankback ratio, *BR* blocking or rejection ratio, *No Sim.* for no simulation.
[f] The random variable is discrete with k number of bins
[g] [62]
[h] Based on the utilization (or residual bandwidth) and hop count, links are assigned *weights*. These weights are used probabilistically to select a path to route the connection by using such routing schemes as in [63].
[i] The total number of connections setup, average blocking probability and total amount of BW in use.
[j] [64]
[k] The routing algorithm is PNNI-compliant and specified in [65]
[l] Widest-shortest [66], shortest-widest [32], competitive call [67], shortest-distance [68] and min-hop routing.
[m] p and k are shufflenet design parameters, Δ is a de Bruijn graph design parameter.

the citation for the TA Method. Some of these citations had more than one TA technique introduced or analyzed. For these cases, we chose to either include the one that the authors picked as the best or included them all or selected one most unique in the paper. The first column of the Table 2 is the category of the TA technique with respect to the taxonomy in Figure 7. The second column, *Strategy*, specifies whether the TA is designed to run in a distributed or centralized fashion. The next three columns are about how the QoS parameters are defined; whether or not as random variables, symmetric (i.e. if $q^r_{e^i_{jk}} = q^r_{e^i_{kj}}$ equality holds true) and restrictive, additive or both (as defined in Section 4), respectively. *Selection Criteria* classifies the TA techniques with respect to the Table 1 representation methods. *Precision* indicates whether the logical graph G' preserves the QoS parameter values of the underlying physical network G or some loss of precision occurs. *Time Complexity* is the running time of the TA algorithm while *Decode Complexity* is the time needed to decode the logical representation for the actual, physical one. The last column, *Reaggr.* shows whether a partial run of the TA algorithm is possible when re-aggregation is inevitable due to changes in G.

The first two column of Table 3 are about the spatial complexity of the TA technique; the former, *Advertise*, gives the spatial complexity of TA information a border node disseminates to other domains, and the latter, *Inter-AS*, presents the spatial complexity of a border node's routing table for the inter-domain routing. *Path Selection* basically considers the QoS routing protocol used in the simulations. *Update Trigger* is the policy or event that initiates the TA algorithm to be re-run. The last two columns are for the simulation settings. *Net Dyn* specifies whether a static network is assumed or any of the inherent network variables are considered, such as change in the network resources, congestion levels, etc. The last one, *Eval Metrics*, lists the metric(s) used in the simulations to compare the various algorithms or techniques considered.

6.2 Synopses of the TA Techniques

In addition to the comparative analysis of the TA methods depicted in Table 2 and Table 3, below we provide synopses of the TA methods:

Awerbuch[38] — The performance of different TA techniques (star, spanning trees, FM, spanner) are evaluated under hierarchical source routing. MST, similar to Figure 12b, scores the best in terms of throughput (realized connections over attempted) and crankback ratio (average number of crankbacks per realized connection). Better performance is observed in the simulations when link costs are inverse-exponentially proportional to the residual bandwidth. Another contribution is the threshold-triggered rerun policy for re-aggregation to reduce the cost of TA. Unlike the exact representation of MST under restrictive parameter as reported in [31], imprecision is inevitable under additive parameter.

Awerbuch[39] — TA for asymmetric networks (or directed graphs) is the main topic of this approach. A complicated, Random Bartal Trees [69] based theoretical analysis is given for TA. However, to the authors' confession, the complexity was very high in terms of running time, and the simulation results favored MST and RST combinations as the better TA method, corroborating their previous work in [38].

Bauer[33] — Only logical node characterization under one additive (hop count) and one restrictive parameter is studied without any routing algorithms. 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} is 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 path p_{jk} is an element of the minimal set from node j to k.The geometric depiction of the minimal set produces a boundary which is called an *Efficient Frontier*[25] by the authors, see Figure 18 and Example 1.

Example 1. *Let X and Y be two border nodes. Suppose that there are 7 alternate paths between them whose (A, R) pairs are given by* $(3, 2), (4, 2), (4, 5), (7, 2), (5, 3), (6, 5), (7, 8)$. *As can be readily seen in Figure 18, the Efficient Frontier is the set of convex corner points of ordered pairs of QoS parameters on the Cartesian Plane.*

The algebraic representation of all the efficient paths among all border nodes is referred to as a *transition matrix.* The algorithm to generate the efficient frontier is elegant but time complexity of $O(B^5)$ is prohibitive.

[25]Staircase function is the term used by many others in place of the efficient frontier as we show later.

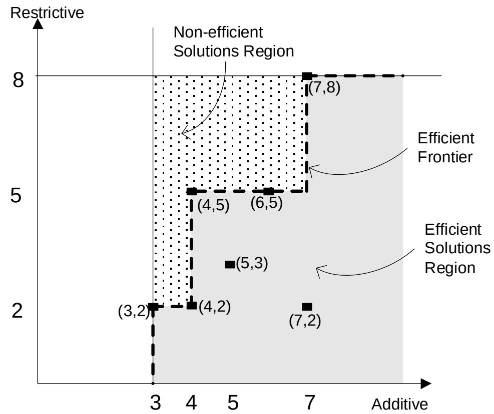

Figure 18: The geometric representation of Efficient Frontier for Example 1.

Ghosh2001 [54] — The approach in [54] 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 dissemination's. 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 7. Another extension might be to address multiple QoS parameter case n the subnetwork.

Guo[56] — Simple Compaction, Full-Mesh and star TA schemes are compared. The unique analysis of the paper is that it considers the traffic load distribution, i.e. uniform and skewed (with hotspots) loads are taken into consideration. Hop count and utilization (or available bandwidth) are advertised as part of the routing information updates. The network nodes then convert these into weights and carry out a probabilistic routing. The conclusions indicate that under uniform load both FM and Star outperform the Simple Compaction. However, under skewed load, which is a more realistic scenario in today's highly volatile and fluctuant networks, even Simple Compaction performs as well as or better than the FM and Star. This is a counter argument to a common belief that aggregation always lead to imprecision.

Sarangan [70, 71, 55] — The method in [70] can be applied only for aggregating bandwidth. The authors point out that advertising only the maximum path bandwidth is not sufficient and suggest to advertise also the total traffic a domain can accept. This total traffic is basically the maximum flow a domain can forward using different paths simultaneously. For example, refer to the simple domain in Figure 8, border node A would advertise the total traffic that it can transmit through itself is 14 (4+8+2). These 14 units cannot exit the network through the same border but can do so through all the borders together. The maximum flow is a well-known graph-theoretic problem. More formal definition of max-flow is beyond the scope of this survey and can be found in [37] and [43]. The classical Ford-Fulkerson algorithm is then used to calculate the max-flow a subnetwork supports via a particular border node. Maximum (or widest) bandwidth and the max-flow values are then advertised as part of the aggregated information. The TA method used is varying simple compaction. The basic approach is augmented in [71] with a stochastic model and the performance is analyzed in [55].

Iliadis[57, 58] — A graph-coloring technique[26], implemented by means of the Kruskal-Prim MST algo-

[26]The definition of the graph coloring in the paper seems to be different than the ones available in the literature as the author

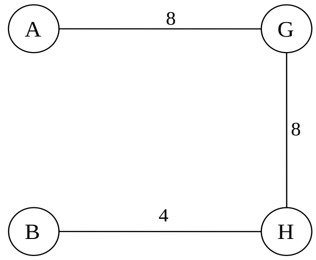

Figure 19: Bandwidth Spanning Tree of Figure 8.

rithm, is used in [57] to compute the *transition matrix*, identically defined as in [33]. The transition matrix is $|B|x|B|$ matrix whose cells state the best QoS parameter among the border nodes. The algorithm provided is not proved. Even the numerical example provided fails to reach the optimal solution. Symmetric star with bypasses for restrictive, single QoS parameter case is used in [58]. The algorithm provided is proved, the upper bound on the number of bypasses increases fast and is approaching $O(|B|)$. Additive parameter or multiple parameter scenarios are not addressed.

Chang[59] — Chang et al. proposes a varying simple compaction TA mechanism that the QoS parameter of transiting a domain via a certain border node is the parameter of the best path from that border node to any other border node within the same domain. The QoS parameters advertised in the simulations are delay, cost, and bandwidth. However, the path that has best delay may be different from the path that has the best cost. For example, refer to Figure 8, border node A would advertise the transit delay to be 9, which comes from the path $A \rightarrow E \rightarrow G$. On the other hand, A would advertise the maximum bandwidth allowed to be 8, which is derived from the path $A \rightarrow D \rightarrow H$. The paper also studies the performance of using different link cost functions and update policies (see the second column of the Table 3). They have compared Markov Decision Process (MDP)based link cost function and Competitive-on-Line (COL) [67]. In order to reduce the size and the frequency of the routing information advertisements, constant-timer based, network state based and cost based triggering schemes are compared. They introduce the cost-based update with hysteresis approach in which the routing information advertisement and the re-aggregation is triggered when the link cost, defined as an exponential function of the residual bandwidth, changes by more than a predefined threshold. They use *Fractional Reward Loss* as the performance evaluation metric, that may be viewed as weighted blocking probability for the connections.

Lee[31, 30] — [30] gives a tutorial on topology aggregation and Lee studies spanning tree (ST) approaches to TA in [31]. The QoS parameters considered are bandwidth and delay, but the method can be applied to bandwidth or delay alone. The first step is to build a FM among the border nodes. The bandwidth and delay parameters of a logical link are those of a single physical path that goes between the related border nodes. This physical path can be the maximum bandwidth path. The issue of picking this physical path is outside the scope of the papers. After the FM is formed, a maximum-weight spanning tree is found based on the FM for each parameter. That is, there are two spanning trees: one for bandwidth and one for delay. Refer to the simple subnetwork in Figure 8, Figure 9(a) is the FM based on maximum bandwidth paths. The spanning trees of bandwidth and delay are shown in Figures 19 and 20, respectively. To decode a logical link in the FM from the spanning tree representation, one has to identify the path the connects the related border nodes in the spanning trees. The bandwidth of the logical link is the bandwidth of the path in the bandwidth spanning tree. The delay of the logical link is found in the same way, but on the delay spanning tree. That is, the delay of the logical path is the delay of the minimum delay edge on the delay spanning tree path. Figure 21 is the decoded FM of the spanning trees in Figures 19 and 20. It can be observed from the figures that the spanning tree provides an exact representation of the FM for bandwidth (restrictive) parameter but not for delay (additive) parameter. Network is assumed to be symmetric; asymmetry may

allows adjacent nodes to have the same color.

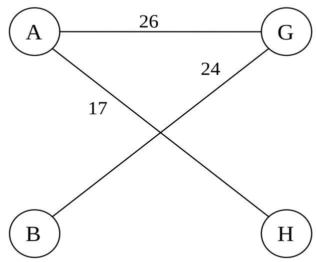

Figure 20: Delay Spanning Tree of Figure 8.

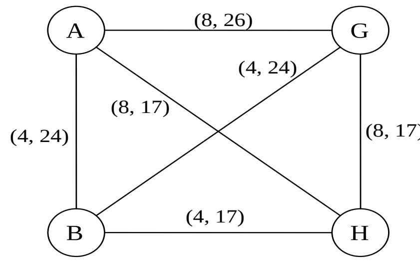

Figure 21: Decoded FM of Figures 19 and 20.

defeat the whole purpose of TA by bringing the complexity to a similar level of the FM. There is neither routing protocol nor any simulation results provided in the papers.

Lee1999 [46] — This paper addresses the case of an asymmetric network for the single-parameter TA. The proposed method is based on *1-subspanner*, called as Minimum Equivalent Subspanner (MES). An example MES was provided in Section 5.1. Modified Floyd-Marshall [43] (centralized) and modified Dijkstra [72] (distributed) are given for MES. The difference between a 1-spanner and 1-subspanner is that the latter directly works on the original topology with border and non-border nodes to find a MES of the border nodes while the former first requires a FM of border nodes. This is an elimination of cumbersome step. However, no routing algorithm is given nor any performance evaluation simulation is conducted. Further, a symmetric star with bypasses TA method is shown to be extracted from the MES by invoking a Steiner 1-spanner algorithm. There are no details of the Steiner algorithm which is likely to be an NP-hard problem.

Korkmaz[34, 35] — The concepts from LVA [36] and ALVA [26] are adopted for the single-parameter TA. That is, not all the information of a FM representation is relevant to the neighboring subnetworks. Instead, different relevant portions of the FM is advertised through different border nodes. The authors term this a *source-oriented* approach. The resulting aggregated topology of a source-oriented approach is called *quasi-star*. Three TA methods are given in the paper: (1) Unified quasi-star (as defined above), (2) Source-oriented simple node (Uniform Varying Compaction in Figure 7) and (3) Source-oriented star (a union of asymmetric star and quasi-star of border nodes). The simple node and the stars are constructed based on the FM. For simple node, the parameter of the worst path from a given border node to every other border node is used. For example, Figure 22 is the FM of Figure 8 when only delay is considered. The parameter that border node A would advertise is $max\{9, 10, 12\} = 12$. A star is formed by taking the average approach. The parameter of a spoke is the average of the parameters of logical links that are related to the spoke. Refer to Figure 22, the parameter of the spoke from A to the nucleus is (9 + 10 + 12) / 3 = 10 1/3 while the parameter of the spoke from the nucleus to H is (9 + 10 + 11) / 3 = 10. The authors also study how to assign parameters to logical links in a FM when there are multiple QoS parameters. The authors define a new parameter called *stretch factor* which measures the deviation of the parameters of a path from the best parameters possible.

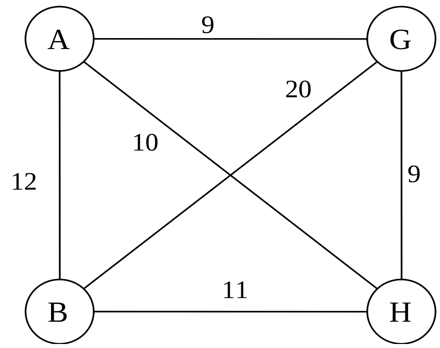

Figure 22: FM of Figure 8 when delay is considered

The stretch factor of a path p that goes from border node j to k, denoted s_factor_p, is defined as follows:

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

$$Best_{q^r} = max\left\{q_p^r \quad | \quad 1 \leq p \leq |p_{jk}|, \quad \forall \quad \text{restrictive parameter} \quad r\right\}$$

$$Best_{q^a} = min\left\{q_p^a \quad | \quad 1 \leq p \leq |p_{jk}|, \quad \forall \quad \text{additive parameter} \quad a\right\}$$

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 links 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.

No routing algorithm is provided in the papers but mechanisms for calculating the stretch factor of a path and checking whether a path is likely to meet the QoS requirement of a request are developed. For simplicity, the details of the mechanisms are not described here. The idea can be illustrated geometrically for Example 1. For the Example 1 with one restrictive and one additive parameters, geometrical representation is nothing more than a curve fitting as illustrated in Figure 23. It effectively contracts the region in which non-routable requests are admitted. Only limited network dynamics are considered for the simulation and for the re-aggregation policy in [35].

Lui[52] — This approach brings forth a solution for networks similar to the one addressed by Bauer in [33]. The network with one additive and one restrictive parameter is termed here as Bandwidth-Delay Sensitive Networks. 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 using linear regression. A sample line segment for the Example 1 is given in Figure 24. 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 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

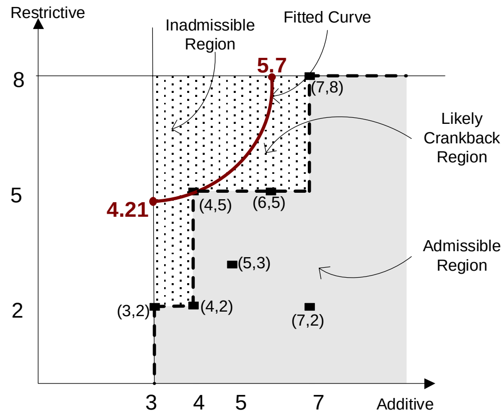

Figure 23: The geometric representation of Korkmaz2000 approach [35] for a two-parameter subnetwork model on the Example 1.

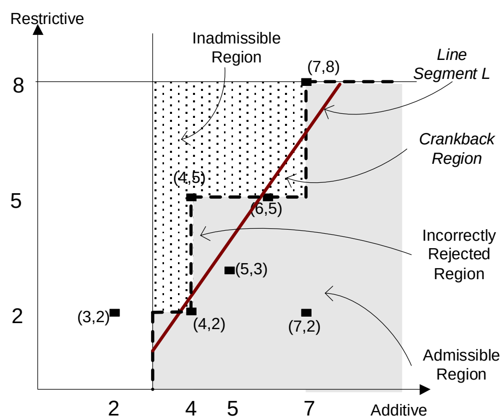

Figure 24: The geometric representation of Line Segment TA method of [73] for the Example 1.

asymmetric star with bypasses topology. In order to find the QoS parameter values of logical links to and from the fictitious nucleus, arithmetic *join* and *split* operations are defined for line segments. Routing algorithms that are tailored for the line segments are developed in the papers. Simulations are performed on static networks and matters of re-aggregation are not discussed in depth in the papers. Possible extensions are to study two different two-parameter type combinations such as two additive parameters and to consider more than two parameters. Further, it would be interesting how the TA approach behaves under different update triggers and when the network dynamics are considered.

Iwata[60] — The TA method proposed generates an asymmetrical star with bypasses by using a Linear Programming Formulation (LPF) with FM as the initial step. Several LPFs for different QoS parameters defined by the PNNI are solved separately one parameter at a time. The simulations compare different TA schemes by using the blocking probability as the only performance evaluation metric.

Hao[51] — Full-mesh and star TA methods are compared with respect to routing update policy and QoS routing algorithms. Similar to the Guo1998 [56], one main conclusion is that TA does not always result degraded routing performance even if imprecision is increased. As for the routing update policy, the simulation reveals that as the update interval goes up the performance gap between the FM and star

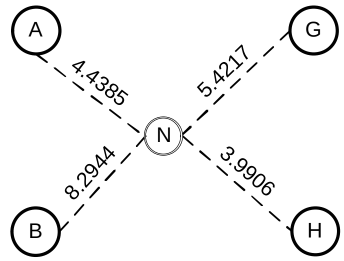

Figure 25: The asymmetric star representation of Figure 9b based on additive metric only by using Liu[61]'s least-square approximation method.

narrows. In other words, while the routing performance of the FM is better than star when the update interval is short and hence the accuracy is high, star performs very close to FM when the update interval widens and makes topological information available to the network nodes more imprecise. Two TA methods are proposed: (1) The *Hybrid* recognizes that not all QoS parameters are created equal; some would live with less frequent updates than the others. Discriminating among the QoS parameters in terms of the frequency of their update and the level of detail when updates are generated is shown to produce good results. The example used in the paper is the hop count versus available bandwidth. The former can be advertised less frequently in a FM fashion while the latter is to be updated more frequently in star fashion. The downside is that both FM and star aggregations need to be maintained by the border nodes, defeating the very purpose of complexity reduction. (2) *Weighted Star* is an asymmetric star that assigns weights to the spokes based on the total traffic going through them. In a way, this approach is similar to the max-flow approach by [70] except that no stochastic model is assumed in this.

Liu[61] — Asymmetric star with bypasses is the TA method used. Least Square Approximation (LSA) and Maximum Deviation Minimization (MDM) are compared 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 is

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

where ${q_j}'$ represents the QoS parameter value of the spoke form 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 9b based on additive parameter only, we would get the one depicted in Figure 25. 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 dominates and time complexity is increased to $O(B^2)$.

Yoo[47, 48] — Two methods from Optical Communications Networks [49]are adopted to aggregate topology information in subnetworks. The first is *Shufflenet* and the other is *de Bruijn Graph* technique. Mapping algorithms to both of these representations are provided in the paper. A common problem with both is that the delay as measured in terms of the hop count increases. Also, these techniques are inherently not conducive to more than one QoS parameter. Figure 15 shows numerical examples of both Shufflenet and de Bruijn techniques of TA.

Tang[53] — The main focus of this proposal is how to improve the accuracy of TA techniques with respect to choosing epitome of QoS parameters as discussed in 5.2 under one restrictive and one additive QoS parameter. Three separate approaches are introduced: polynomial curves, cubic splines and polylines. The polynomial curve approach fits a least-square polynomial of degree n given m data points in Cartesian plane. When $n = 1$ the approach is identical to Lui [52]'s.

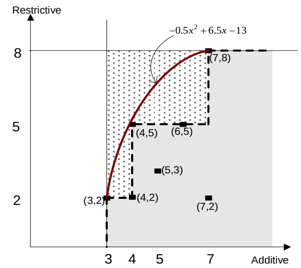

Figure 26: A Quadratic polynomial fit by least-square method for the data set of Example 1, presented in Tang[53].

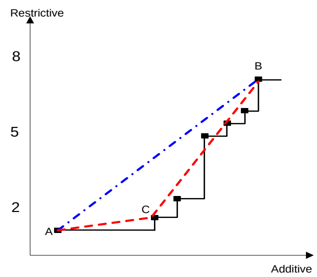

Figure 27: Polyline approximation of Tang[53].

Example 2. *Let us consider the same setting as Example 1. The representative points are* $(3, 2), (4, 5), (7, 8)$. *If we decide to use a quadratic function for polynomial curve the least-square method generates* $-0.5x^2 + 6.5x - 13$ *as plotted in Figure 26.*

The cubic spline approach 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 polyline approximation uses k piecewise line segments to approximate the data set, again by means of the least-square method. The polyline 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 27, A, B and C are more relevant than the others for they shape more substantially the overall behavior of the staircase function. Once the value of g is chosen, the authors provide a heuristic to choose the line segments to reduce the running time complexity as finding the optimum set of lines minimizing $\Delta(\kappa, \kappa')$ (see below for definition) would be costly.

The three approaches with line segment of Lui[52] and curve fitting of Korkmaz[35] are compared in terms of minimizing the area ($\Delta(\kappa, \kappa')$) between the staircase function[27] and the approximation line or curve:

$$\begin{aligned}\Delta(\kappa, \kappa') &= \Delta_+(\kappa, \kappa') + \Delta_-(\kappa, \kappa') \\ &= \int_{-\infty}^{\infty} \max(\kappa - \kappa', 0) + \int_{-\infty}^{\infty} \max(\kappa' - \kappa, 0)\end{aligned}$$

where κ is the staircase function of the additive parameter in terms of the restrictive parameter and κ' is the function of the approximation curve for κ. Note that the $\Delta_-(\kappa, \kappa')$ region is marked as *Incorrectly Rejected Region* and $\Delta_+(\kappa, \kappa')$ as *Crankback Region* in Figure 24.

[27] The staircase function is the term used for the efficient frontier of Figure 18.

7 Conclusion

We have analyzed and compared topology aggregation techniques from the literature. Our focus was to evaluate these techniques with respect to their implications on QoS routing as there seem to be very few studies topology of aggregation techniques that consider their efficacy on QoS routing protocols. 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. The key is the tradeoff between performance and accuracy of routing results. Much more study of these tradeoffs is needed to shed light to determine the optimal point of balance.

References

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

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

[3] D. Davies, "Packet Switching Proposal," *Proposal and Prototype Implementation at UK National Physical Laboratories*, 1965. Proposal and Prototype Implementation of packet-switching concepts.

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

[5] D. Clark, S. Bellovin, B. Braden, N. Chiappa, T. Faber, A. Falk, M. Handley, S. Shenker, K. Sollins, and J. Wroclawski, "NewArch: A New Architecture for an Internet," *Report to Next Generation Internet PI Meeting*, January 2002. Work In Progress.

[6] S. Chen and K. Nahrstedt, "An Overview of Quality of Service Routing for Next-Generation High-Speed Networks: Problems and Solutions," *IEEE Network Magazine*, pp. 64–79, November/December 1998.

[7] A. Doria and E. Davies, "Analysis of IDR requirements and History," *Internet Draft, <draft-irtf-routing-history-01.txt>*, July 2004. Work In Progress.

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

[9] F. Baker, "Requirements for IP Version 4 Routers," *Request for Comments (RFC 1812)*, June 1995.

[10] G. Huston, "BGP Table Statistics and Reports," *http://bgp.potaroo.net/*, 2003.

[11] G. Huston, "Commentary on Inter-Domain Routing in the Internet," *Request for Comments (RFC 3221)*, December 2001.

[12] A. Doria, E. Davies, and F. Kastenholz, "Requirements for Inter Domain Routing," *Internet Draft, <draft-irtf-routing-reqs-03.txt>*, July 2004. Work In Progress.

[13] I. Castineyra, N. Chiappa, and M. Steenstrup, "The Nimrod routing architecture," *RFC1992*, August 1996.

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

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

[16] J. Yu, "Scalable Routing Design Principles," *Request for Comments (RFC 2791)*, July 2000.

[17] L. Kleinrock and F. Kamoun, "Hierarchical routing for large networks: Performance evaluation and optimization," *Computer Networks*, vol. 1, pp. 155–174, January 1977.

[18] F. Kamoun and L. Kleinrock, "Stochastic Performance Evaluation of Hierarchical Routing for Large Networks," *Computer Networks*, vol. 3, pp. 337–353, November 1979.

[19] L. Kleinrock and F. Kamoun, "Optimal Clustering Structures for Hierarchical Topological Design of Large Computer Networks," *Networks*, vol. 10, pp. 221–248, Fall 1980.

[20] W. Tsai, C. Ramamoorthy, W. Tsai, and O. Nishguchi, "An Adaptive Hierarchical Routing Protocol," *IEEE Transactions on Computers*, vol. 30, August 1989.

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

[22] D. Estrin, Y. Rekhter, and S. Hotz, "Scalable Inter-Domain Routing Architecture," *ACM SIGCOMM '92 Baltimore, MD*, pp. 40–52, August 1992.

[23] M. Streenstrup, "An Architecture for Inter-Domain Policy Routing," *Request for Comments (RFC 1478)*, June 1993.

[24] C. Alaettinoglu and A. U. Shankar, "The Viewserver Hierarchy for Interdomain Routing: Protocols and Evaluation," *IEEE JOURNAL on Selected Areas in Communications*, vol. 13, pp. 1396–1410, October 1995.

[25] ATM Forum, "Private Network-Network Interface Specification version 1.0," March 1996.

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

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

[28] I. 10589, "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*, November 2002.

[29] Y. Rekhter and T. L. and, "A Border Gateway Protocol 4 (BGP-4)," *Request for Comments (RFC 1771)*, March 1995.

[30] W. C. Lee, "Topology aggregation for hierarchical routing in atm networks," *ACM SIGCOMM Computer Communications Review, Special Issue on ATM*, vol. 25, pp. 82–92, April 1995.

[31] W. C. Lee, "Spanning tree method for link state aggregation in large communication networks," *INFOCOM*, pp. 297–302, 1995.

[32] Z. Wang and J. Crowcroft, "Quality-of-Service Routing for Supporting Multimedia Applications," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 1228–1234, September 1996.

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

[34] T. Korkmaz and M. Krunz, "Source-oriented topology aggregation with multiple qos parameters in hierarchical atm networks," *7th IWQoS, London*, May 1999.

[35] T. Korkmaz and M. Krunz, "Source-oriented topology aggregation with multiple QoS parameters in hierarchical networks," *ACM Transactions on Modeling and Computer Simulation*, vol. 10, pp. 295–325, October 2000.

[36] J. Garcia-Luna-Aceves and J. Behrens, "Distributed, Scalable Routing Based on Vectors of Link States," *IEEE JOURNAL on Selected Areas in Communication*, vol. 13, pp. 1383–1395, October 1995.

[37] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA: McGraw Hill/MIT Press, Second ed., May 2001.

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

[39] B. Awerbuch and Y. Shavitt, "Topology aggregation for directed graphs," *IEEE/ACM Transactions on Networking*, vol. 9, February 2001.

[40] P. Chew, "There is a planar graph almost as good as the complete graph," in *Proceedings of the second annual symposium on Computational geometry*, pp. 169–177, ACM Press, 1986.

[41] D. Peleg and J. D. Ullman, "An optimal synchronizer for the hypercube," in *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, pp. 77–85, ACM Press, 1987.

[42] I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares, "On sparse spanners of weighted graphs," *Discrete & Computational Geometry*, vol. 9, no. 1, pp. 81–100, 1993.

[43] R. K. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms and Applications*. Upper Saddle River, NJ: Prentice Hall, First ed., 1993.

[44] S. Baswana and S. Sen, "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*, vol. 2719 of *Lecture Notes in Comput. Sci.*, pp. 384–396, Berlin: Springer, 2003.

[45] L. Cai, "NP-completeness of minimum spanner problems," *Discrete Applied Mathematics. Combinatorial Algorithms, Optimization and Computer Science*, vol. 48, no. 2, pp. 187–194, 1994.

[46] W. C. Lee, "Minimum equivalent subspanner algorithms for topology aggregation in atm networks," *ICATM '99. 2nd International Conference on ATM*, pp. 351–359, June 1999.

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

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

[49] B. Mukherjee, *Optical Communication Networks*. : McGraw Hill, First ed., January 1997.

[50] Y. Yoo, S. Ahn, and C. Kim, "Topology aggregation schemes for asymmetric link state information," *Journal of Communications and Networks*, vol. 6, pp. 46–59, March 2004.

[51] F. Hao and E. W. Zegura, "On scalable QoS routing: Performance evaluation of topology aggregation," *INFOCOM*, pp. 147–156, 2000.

[52] K.-S. Lui, K. Nahrstedt, and S. Chen, "Routing with topology aggregation in delay-bandwidth sensitive networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 1, pp. 17–29, 2004.

[53] Y. Tang and S. Chen, "QoS information approximation for aggregated networks," in *IEEE International Conference on Communications, ICC 2004*, (Paris, France), pp. 2107–2111, June 2004.

[54] D. Ghosh and R. Acharya, "A probabilistic scheme for hierarchical qos routing," *IEEE Ninth International Conference On Networks (ICON)*, pp. 416–421, October 2001.

[55] D. G. V. Sarangan and R. Acharya, "Performance Analysis of Capacity-aware State Aggregation for Inter-domain QoS Routing," in *IEEE Global Telecommunications Conference, GLOBECOM 2004*, (Dallas, Texas), December 2004.

[56] L. Guo and I. Matta, "On state aggregation for scalable qos routing," *IEEE Proceedings of the ATM Workshop '98*, pp. 306–314, May 1998.

[57] I. Iliadis and P. Scotton, "Transition matrix generation for complex node representation," *IEEE ATM Workshop '99, Kochi, Japan*, May 1999.

[58] I. Iliadis, "Optimal pnni complex node represenations for restrictive costs and minimal path computation time," *IEEE/ACM Transactions on Networking*, vol. 8, August 2000.

[59] B. Chang and R. Hwang, "Efficient Hierarchical QoS Routing in ATM Networks," *Computer Communications*, vol. 24, pp. 1648–1660, October 2001.

[60] A. Iwata, H. Suzuki, R. Izmailov, and B. Sengupta, "QoS Aggregation Algorithms in Hierarchical ATM Networks," *IEEE ICC'98, Atlanta*, June 1998.

[61] J. Liu, Z. Niu, and J. Zheng, "Parameter Dimensioning Algorithms of the PNNI Complex Node Model with Bypasses," *IEICE Transaction on Communications*, vol. E83-B, pp. 638–645, March 2000.

[62] R. Guerin and A. Orda, "QoS Routing in Networks with Inaccurate Information: Theory and Algorithms," *IEEE/ACM ToN*, vol. 7, June 1999.

[63] S. Bahk and M. E. Zarki, "Dynamic Multi-path Routing and How it Compares with other dynamic Rouitng Algorithms for High Speed Wide Area Networks," *ACM SIGCOMM '92*, September 1992.

[64] Z. Wang and J. Crowcroft, "Bandwidth-Delay Based Routing Algorithms," *Proceedings of the IEEE GLOBECOM '95*, vol. 3, pp. 2129–2133, 1995.

[65] A. Iwata, R. Izmailov, D.-S. Lee, B. Sengupta, G. Ramamurthy, and H. Suzuki, "ATM Routing Algorithms with Multiple QoS Requirements for Multimedia Networking," *IEICE Transactions on Communications*, vol. E79-B, August 1996.

[66] R. Guerin and A. Orda and D. Williams, "QoS Routing mechanisms and OSPF Extensions," *IEEE GLOBECOM '97 Phoenix, AZ*, vol. 3, pp. 1903–1908, November 1997.

[67] R. Gawlick, A. Kamath, S. Plotkin, and K. Ramakrishnan, "Routing and Admission Control in General Topology Networks," Tech. Rep. CS-TR-95-1548, Stanford University, 1995.

[68] Q. Ma, P. Steenkiste, and Z. H., "Scalable Inter-Domain Routing Architecture," *ACM SIGCOMM '96*, pp. 206–2167, August 1996.

[69] Y. Bartal, "Probabilistic Approximation of Metric Space and its algorthmic Applications," *Proceedings of 37th Annu. IEEE Symp. Foundations of Computer Science*, pp. 184–193, October 1996.

[70] V. Sarangan and R. Acharya, "Using network flows in hierarchical qos routing," *IEEE Ninth International Conference On Networks (ICON)*, pp. 422–427, October 2001.

[71] V. Sarangan, D. Ghosh, and R. Acharya, “”state aggreagtion using network flows for stochastic networks”,” *IEEE Globel Communications Conference GLOBECOM 2002”*, pp. 2430–2434, November 2002.

[72] E. Dijkstra, “A Note on Two Problems in connexion with Graphs,” *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[73] K. Lui, K. Nahrstedt, and S. Chen, “Hierarchical QoS routing in delay-bandwidth sensitive networks,” *Proceedings of IEEE LCN, Tampa, FL*, pp. 176–189, 2000.