QoS Information Approximation for Aggregated Networks

Yong Tang and Shigang Chen
Department of Computer & Information Science & Engineering
University of Florida
Gainesville, Florida 32611–6120
Email: {yt1, sgchen}@cise.ufl.edu

***Abstract*— Many important network functions (e.g., QoS provision, admission control, traffic engineering, resource management) rely on the availability and the accuracy of the network state information. However, it is impractical to maintain the complete state information of a large internetwork at a single location. Instead, a large network is often hierarchically structured, with each domain advertising its aggregated state. To achieve scalability, the amount of information after aggregation should be minimized. To improve accuracy, the aggregation method must be carefully selected. This paper gives a unified account of state aggregation based on the concept of service curves. The aggregation of network state is modeled as a recursive process of service curve transformation. New approximation methods based on polynomial curves, cubic splines and polylines are proposed, and their scalability/accuracy tradeoffs are studied. Our simulations show that these new methods approximate the network state far more accurate than the existing methods. In particular, the polylines achieve the best scalability/accuracy tradeoff.**

I. INTRODUCTION

The availability and the accuracy of the network state information is essential to many important network functions. With the explosively growing of the Internet in recent years maintaining the up-to-date state information of a large network has been a difficult task. To solve the scalability problem, the current networks are structured hierarchically as a collection of domains, which may be further decomposed into sub-domains recursively. At each hierarchical level, the state information can be aggregated and only the aggregated information is advertised externally. The state aggregation has to achieve two goals. (1) The aggregated information about a domain should be as accurate as possible. (2) The size of the aggregated state should be reduced as much as possible. While the performance of the network function is decided by the provided information, it is equally important to compact the state information so that the scalability is achieved. As a result, a tradeoff has to be made. In addition to the above issues, the network aggregation often deals with multi-dimensional state. In this paper, although we focus two dimensions, *delay* and *bandwidth*, our approaches can be easily extended to multi-dimensions with one restrictive (bandwidth) and multiple additive (delay, cost, ... etc.) parameters.

The aggregated state of a domain does not contain detailed information about the internal structure of the domain, but only the information describing end-to-end properties between border nodes [1], [2], [3], [4]. Consider a domain with two border nodes, a and b, in Figure 1. The delay and the bandwidth of each physical path from a to b represent a point on the delay-bandwidth plane, and the points of all physical paths from a to b define a *service staircase*, which outlines the area of supported services (the shaded area). Any service request whose bandwidth and delay requirements fall in the area can be admitted. There are up to $|E|$, which is the number of physical links in the domain, corner points on the staircase [4]. The goal of aggregation is to approximate the service staircase with limited space independent of the domain size. We should point out here that the routing process does not need the approximation curve completely lie within the supported service area. The router will first reject all connection requests lies below the approximation curve. The actual path will be probed and the resource will be reserved after that. The flow will be rejected as well if the resource requirement can not be satisfied in the actually path.

The work of aggregating a delay-bandwidth sensitive network can be traced back to [1], where Lee transformed the network to a spanning tree among border nodes. While the paper provides a distortion-free aggregation for bandwidth, the distortion of the delay can be large. Iwata et al. [2] proposed an aggregation approach for networks of six parameters. The distortion can be minimized by a linear programming approach. Korkmaz and Krunz [3] proposed to approximate multi-dimensional QoS state by three values: the minimum delay min_d, the maximum bandwidth max_b, and the smallest stretch factor min_s among all paths between two border nodes. The stretch factor of a path measures how much the delay and the bandwidth of the path deviate from the best delay and the best bandwidth of all paths. It essentially defines a hyperbola curve that resides below the staircase (Figure 1). A connection request is accepted if the delay requirement is larger than min_d, the bandwidth requirement is smaller than max_b, and its stretch factor is smaller than min_s. Lui et al. [4] chose a line segment to approximate the staircase. The line segment is determined by the least square method that minimizes the least-square error, i.e., the summation of the squares of the distances from the corder points to the line. The paper demonstrated by simulation that line segments work better than stretch factors on average. It dose not discuss the general case of the aggregation algebra, however, only

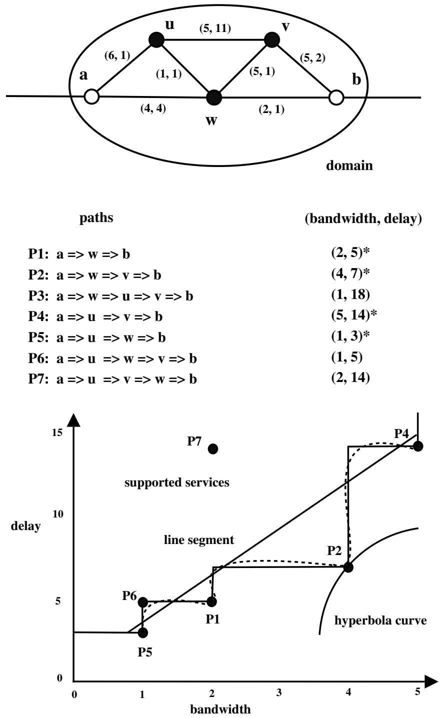

Fig. 1. service staircase and its approximates for (bandwidth, delay)

approximation has been provided.

This paper gives a unified account of state aggregation based on the concept of service curves. We propose three curves, *polynomial curves*, *piecewise cubic splines*, and *polylines* with far better accuracy compared with Korkmaz-Krunz's algorithm [3] and Line Segment algorithm [4]. The algebra and the algorithms related to the state information aggregation has been discussed and the performance of our methods is demonstrated by our simulations.

II. Aggregation Model

We use a tuple (V, B, E) to represent a domain in a hierarchically structured network. V is the set of nodes in the domain, $B \subset V$ is the set of border nodes that have links connecting to nodes outside of the domain, and E is the set of directed physical links among the nodes in V. Each link (u, v) is associated with a pair of QoS parameters $(w_{u,v}, d_{u,v})$, where $w_{u,v}$ is the bandwidth and $d_{u,v}$ is the delay of the link.

A service request with a bandwidth requirement of w and a delay requirement of d can be satisfied by a link (u, v) if $w \leq w_{u,v}$ and $d \geq d_{u,v}$. Therefore, the *supported services*, denoted as $s_{u,v}$, are defined as the set of all requests that are satisfied by link (u, v). Formally, we have

$$s_{u,v} = \{(w, d) \mid w \leq w_{u,v} \wedge d \geq d_{u,v}, w \in \mathbf{R}^+, d \in \mathbf{R}^+\}$$

which is represented by the upper region above the staircase in Figure 1. The bandwidth and the delay of a physical path p are defined as $w_p = \min_{(u,v)\in p} w_{u,v}$ and $d_p = \sum_{(u,v)\in p} d_{u,v}$. Similar to the supported service of a link (u, v), we define the supported services of path p as s_p.

Given two nodes a and b, there may exist multiple physical paths between them. The set of all physical paths $\{p_1, p_2, ..., p_l\}$ from a to b can be represented by a *logical link* (a, b) in the aggregated topology. In this case, a single pair (bandwidth, delay) is not enough to represent the state information any more. It should be characterized by a *service staircase* (Figure 1). The convex-corner points, which uniquely define the shape of the staircase, are called *representatives* (e.g., p_5, p_1, p_2, and p_4 in Figure 1). The set of representatives for a logical link (a, b) is denoted as $\gamma_{a,b}$, which can be calculated in time $O(|E|^2 + |E||N|\log|N|)$ [4]. Given the non-decreasing nature of a staircase, the representatives can be arranged in an ascending order: $\gamma_{a,b} = \{(w_i, d_i)\}$, such that $w_i < w_j \ \wedge \ d_i < d_j, \ \forall 1 \leq i < j \leq I$, where I is the size of $\gamma_{a,b}$ and $I \leq |E|$. If we consider the delay as a function (κ) of the bandwidth, the service staircase can be defined as

$$\kappa(w) \quad = \quad \begin{cases} d_1 & ; w \leq w_1 \\ d_i & ; w_{i-1} < w \leq w_i, \ \forall 1 < i \leq I \\ \infty & ; w > w_l \end{cases} \qquad (1)$$

The supported services of link (a, b) can be expressed as

$$s_{a,b} = \{(w, d) \mid d \geq \kappa(w), w \in \mathbf{R}^+\}$$

With the assumption that certain service curves are used, we define an algebra for network state aggregation based on service curves. Given a large hierarchically structured network, after the state information of the physical links is aggregated into physical paths and then logical links, each domain is reduced to a much smaller topology consisting of only border nodes. As a domain advertises its aggregated state, only the service curves between the border nodes are seen from the outside of the domain. The same aggregation process can be repeated at higher-level hierarchies with logical links being aggregated into logical paths and high-level logical links.

There are two basic types of computation in such process. Sequential aggregation is to compute the service curve of a path based on the service curves of the links and parallel aggregation is to compute the service curve for a logical link between two nodes or aggregated paths between nodes. These two aggregations can be done by recursively calling two operations, $\oplus$ and *odot* between two service curves κ_1 and κ_2.

Definition 1: Let $\kappa_1, \kappa_2 : \mathbf{R}^+ \rightarrow \mathbf{R}^+$ be two service curves. The operation $\oplus$ is defined as follows: for any bandwidth $w \in \mathbf{R}^+$, $(\kappa_1 \oplus \kappa_2)(w) = \kappa_1(w) + \kappa_2(w)$.

Theorem 1: Let κ_1 and κ_2 be the service curves of two sequential links (v_1, v_2) and (v_2, v_3), respectively. $\kappa_1 \oplus \kappa_2$ is the service curve of path $p = v_1 \rightarrow v_2 \rightarrow v_3$.

Proof: A request (w, d) can be supported by p if and only if it can be split into two parts, (w, d_1) and (w, d_2) with $d_1 + d_2 = d$, such that they can be supported by (v_1, v_2) and (v_2, v_3), respectively. Namely, d_1 and d_2 should satisfy $d_1 \geq \kappa_1(w)$ and $d_2 \geq \kappa_2(w)$. Hence,

$$\begin{aligned} d_1 + d_2 = d \quad & \Longleftrightarrow \quad d \geq \kappa_1(w) + \kappa_2(w) \\ & \Longleftrightarrow \quad d \geq (\kappa_1 \oplus \kappa_2)(w) \end{aligned}$$

Let κ_p be the service curve of p; a service request (w, d) can be supported by p if and only if $d \geq \kappa_p(w)$. Therefore, $\kappa_p = \kappa_1 \oplus \kappa_2$. ∎

Definition 2: Let $\kappa_1, \kappa_2 : \mathbf{R}^+ \rightarrow \mathbf{R}^+$ be two service curves. The operation $\odot$ is defined as follows: for any bandwidth $w \in \mathbf{R}^+$, $(\kappa_1 \odot \kappa_2)(w) = \min(\kappa_1(w), \kappa_2(w))$

Theorem 2: Suppose (a, b) is a logical link that represents two paths, p_1 and p_2, from a to b. Let κ_1 and κ_2 be the service curves of p_1 and p_2, respectively. $\kappa_1 \odot \kappa_2$ is the service curve of (a, b).

Proof: Let s_{p_1} and s_{p_2} be the supported services of p_1 and p_2, respectively. The union of s_{p_1} and s_{p_2} forms the supported services of (a, b). A request (w, d) can be supported by (a, b) if and only if it can be supported by either p_1 or p_2, namely, $d \geq \kappa_1(w) \vee d \geq \kappa_2(w)$. We further have

$$
\begin{aligned}
d \geq \kappa_1(w) \vee d \geq \kappa_2(w) &\iff d \geq \min(\kappa_1(w), \kappa_2(w)) \\
&\iff d \geq (\kappa_1 \odot \kappa_2)(w)
\end{aligned}
$$

Let $\kappa_{a,b}$ be the service curve of p; a service request (w, d) can be supported by p if and only if $d \geq \kappa_{a,b}(w)$ by (II). Therefore, $\kappa_{a,b} = \kappa_1 \oplus \kappa_2$. ∎

Theorem 3: The set of service curves is closed under the operations $\odot$ and $\oplus$ and has the properties of associativity and commutativity.

Proof: The result of two service curves under $\odot$ and $\oplus$ is still a service curve by Theorems 1 and 2. Therefore, the set of service curve functions is closed under $\odot$ and $\oplus$. It is straightforward to prove that $\odot$ and $\oplus$ are associative and commutative since $+$ and $\min$ are associative and commutative. ∎

Thus far we have completed the basic algebra for service curves. The purpose is to introduce a general approach to aggregate network state information. The service curves can be the service staircases or approximation curves. The network aggregation process can be considered as recursively applying $\odot$ and $\oplus$ to the service curves of the links (paths) in the topology.

III. Approximation Curves

It is necessary to approximate the aggregated domain state in order to reduce the space requirement. While improving information accuracy will always increase the space complexity, our goal is to limit the space complexity for storing the aggregated state between two border nodes to a constant independent of the size of the domain and maximize the information accuracy at the same time. Before we introduce our new approximations, a criteria for comparing the quality of different approximation methods is defined.

We use Figure 2 to illustrate the distortion caused by an approximation curve κ'. Suppose κ' instead of κ is stored in the aggregated network. A service request (w, d) in (*over-estimated*) region II will be accepted based on the stored information but rejected in reality. On the other hand, a request (w, d) in (*under-estimated*) region III can be supported in reality but rejected by the stored state information. Therefore,

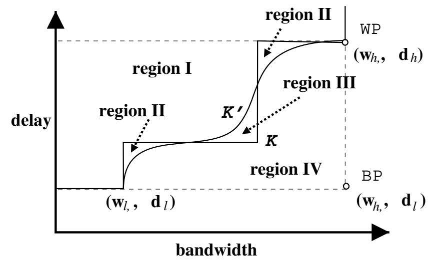

Fig. 2. area distance

It is natural to use the sizes of these regions to judge the quality of approximation by κ'.

Given a service staircase κ and an approximation curve κ', the *positive area distance* (p.a.d.), $\Delta_+(\kappa, \kappa')$, measures the total size of regions of over-estimated services. The *negative area distance* (n.a.d.), $\Delta_-(\kappa, \kappa')$, measures the total size of regions of under-estimated services.

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

where $\kappa(w) - \kappa'(w) = 0$ if $\kappa'(w) = \infty \wedge \kappa(w) = \infty$.

The selection of these two criteria is based on the design philosophy of the system. $\Delta_-(\kappa, \kappa')$ means aggressive as minimizing it would require most supportable services to be accepted, while $\Delta_+(\kappa, \kappa')$ only requires that the accepted requests receive the desired QoS. As an alternative criteria, *area distance* (a.d.), $\Delta(\kappa, \kappa') = \Delta_+(\kappa, \kappa') + \Delta_-(\kappa, \kappa')$ can be considered as a tradeoff between these two. Our goal is to find the appropriate approximation curves so that $\Delta_+(\kappa, \kappa')$, $\Delta_-(\kappa, \kappa')$, and/or $\Delta(\kappa, \kappa')$ are minimized without incurring too much overhead.

A. Polynomial Curve Approximation

Let the first representative of the staircase κ be (w_l, d_l), which is the convex corner with the smallest bandwidth, and the last representative be (w_h, d_h), which is the convex corner with the largest bandwidth, our first approximation method uses a polynomial function

$$
\kappa^{pc}(w) = \begin{cases} d_l & ; w \leq w_l \\ \sum_{i=0}^{n} \alpha_i w^i & ; w_l < w \leq w_h \\ \infty & ; w > w_h \end{cases}
$$

Actually the *line segment* in [4] is a special case of polynomial curves with $n = 1$. With a larger n, better accuracy is achieved but the overhead is increased as well.

The problem of finding coefficients $\alpha_0, \alpha_1, ..., \alpha_n \in \mathbf{R}$ such that $\Delta(\kappa, \kappa^{pc})$ is minimized is difficult. Instead, we use the least square method to find the polynomial that best minimize the least-square error as an approximation. Given m data points (w_1, d_1), ..., (w_m, d_m) on the staircase, the least square method tries to minimize the cost function

$$
\phi(\alpha_0, \alpha_1, ..\alpha_n) = \sum_{i=1}^{m} (d_i - \kappa^{pc}(w_i))^2
$$

By taking the first derivatives we have the following linear equations with $k = 0, 1, ...n$.

$$\frac{\partial \phi}{\partial \alpha_k} = -2 \sum_{i=1}^{m} w_i^k (d_i - \sum_{j=0}^{n} \alpha_j w_i^j) = 0$$

The values of $\alpha_0, \alpha_1, ...\alpha_n$ can be calculated by solving the above linear equations.

The space needed for storing k^{pc} is $n + 5$ float numbers, with $n + 1$ floats for the coefficients and 4 floats for w_l, d_l, w_h, and d_h.

B. Cubic Spline Approximation

In polynomial approach, we assume a global approximation. It can be improved by taking into consideration the local information of the service staircase. Our second approximation method uses a cubic spline κ^{cs}, which approximates the service staircase κ segment by segment with piecewise polynomial functions. We make a brief description of cubic spline in the following. Details about curve fitting by spline functions can be found in [5].

Let $w_l = \lambda_0 < \lambda_1 < ... < \lambda_g = w_h$ be an even partition of the interval $[w_l, w_h]$. Take two additional values with the same spacing on both sides of the interval: λ_{-2} and λ_{-1} smaller than w_l; λ_{g+1} and λ_{g+2} greater than w_h. A series of cubic spline basis functions are defined. For $0 \le j \le g$,

$$B_j(w) = (\lambda_{j+2} - \lambda_{j-2}) \sum_{s=-2}^{2} \frac{(\lambda_{j+s} - w)_+^3}{\prod_{k=-2,k\neq s}^{2} (\lambda_{j+s} - \lambda_{j+k})}$$

where $(x)_+$ equals x if $x \ge 0$, and 0 otherwise.

The cubic spline $\kappa^{cs}(w)$ is a linear combination of $B_j(w)$.

$$\kappa^{cs}(w) = \begin{cases} d_l & ; w \le w_l \\ \sum_{j=0}^{g} \alpha_j B_j(w) & ; w_l < w \le w_h \\ \infty & ; w > w_h \end{cases}$$

where $\alpha_0, \alpha_1, ..., \alpha_g$ are coefficients, which determine the shape of the cubic spline. The larger the value of g, the more the number of coefficients and the better the approximation.

Similar to the previous subsection, we use the least square method to find the coefficients that minimizes the least-square error defined below.

$$\phi(\alpha_0, \alpha_1, ..., \alpha_g) = \sum_{i=0}^{g} (d_i - \kappa^{cs}(w_i))^2$$

where (w_1, d_1), ..., (w_m, d_m) are m data points selected from the service staircase, $m \ge g + 1$. The first derivatives of the $\phi(\alpha_0, \alpha_1, ..\alpha_g)$ give us the following $(g + 1)$ linear equations with $k = 0, 1, ..g$.

$$\frac{\partial \phi}{\partial \alpha_k} = -2 \sum_{i=1}^{m} B_k(w_i)(d_i - \sum_{j=0}^{g} \alpha_j B_j(w_i)) = 0$$

The values of $\alpha_0, \alpha_1, ..\alpha_g$ can be calculated by solving the linear equations.

The space needed for storing k^{cs} is $g + 5$ float numbers, with $g + 1$ floats for the coefficients and 4 floats for w_l, d_l, w_h, and d_h. The minimal value for g is one.

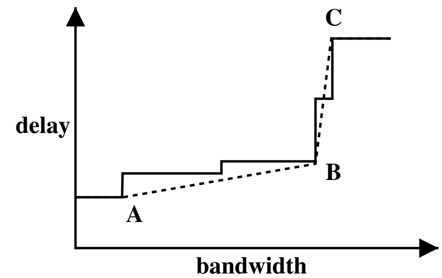

Fig. 3. polyline approximation

C. Polyline Approximation

Although a cubic spline approximates the local trends of a staircase very well, it has a weakness. Like a polynomial curve, a cubic spline incurs heavier distortion near the corner points as it uses a *continuous* curve to approximate the *discrete* steps (Figure 2).

The corner points contain more information than the other points on the staircase because they determine the shape of the staircase. Furthermore, some corner points are more important than others. For example, there are eight corner points in Figure 3. Points A, B, and C are more important than the other five because they control the global shape of the staircase. Hence, we can approximate the staircase by only keeping the most important corner points. In Figure 3, we may keep only A, B, and C, and use a polyline κ^{pl} across them to approximate the service staircase.

Now the problem is to determine which corner points are more important than the others. Suppose we are allowed to choose only k corner points due to a space limitation. We would like to choose such k points that minimize $\Delta(\kappa, \kappa^{pl})$. Suppose there are n corner points in total. The number of different combinations of k points is C_n^k. It takes $O(n)$ time to compute $\Delta(\kappa, \kappa^{pl})$ for each combination. Therefore, for a brute-force algorithm, it takes $O(nC_n^k)$ time to find the best k points. In the following, we describe a heuristic algorithm with a time complexity of $O(nk)$.

Given a set of n corner points on the staircase, $\{A_1, ..., A_n\}$. Initially, the approximation polyline κ^{pl} is a line segment connecting A_1 and A_n, represented as A_1—A_n. Compute the distances from all other corner points to the polyline, and find the corner point A_x with the maximum distance to A_1—A_n. Then insert A_x into the polyline, which becomes A_1—A_x—A_n. Repeat the above process until the polyline contains k corner points.

The space needed for storing k^{pl} is $2k$ float numbers, two for each selected corner point.

IV. Simulation Results

In this section, we study the performance of different approximation approaches by simulations. We compare *polynomial curve* (PC), *cubic spline* (CS), and *polyline* (PL), together with the existing approaches of *line segment* (LS) [4], *Korkmaz-Krunz* (KK) [3], *best point* (BP) and *worst point* (WP). The domain topologies in our simulation follow the

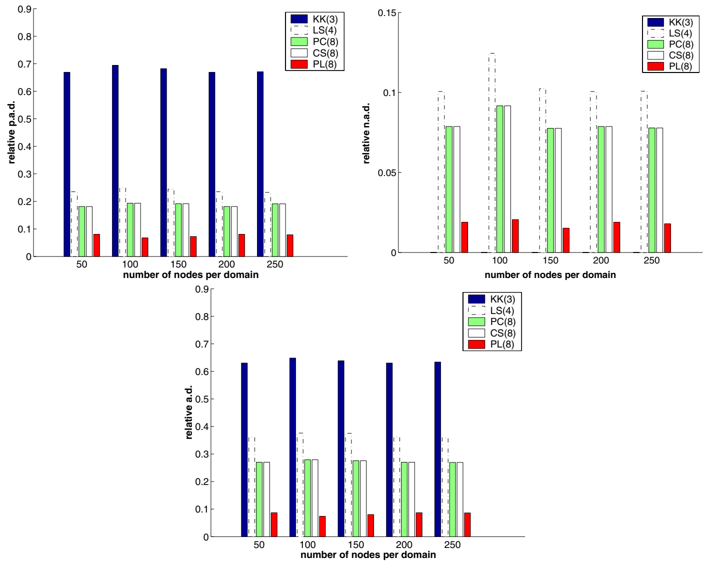

Fig. 4. performance comparison of different domain size

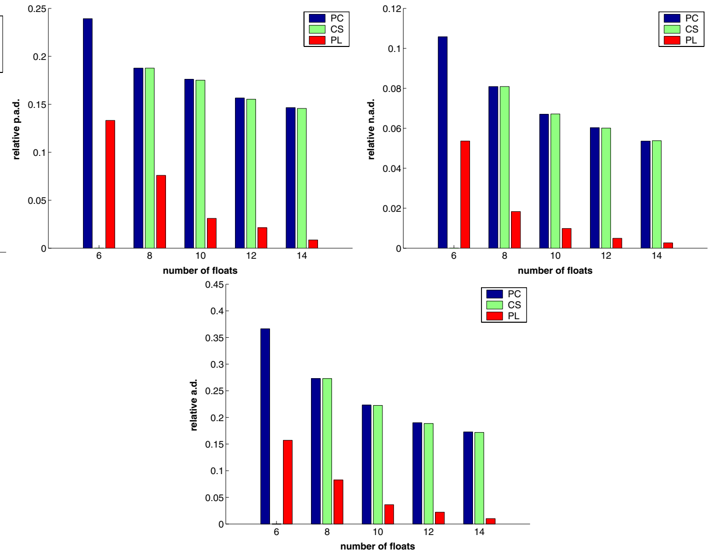

Fig. 5. accuracy improvement with respect to space

Waxman model [6] with the average node degree of 4. 10% of the nodes in the domain are randomly selected as the edge nodes. We assign the delays and bandwidths of the physical links based on the exponential distribution with an average of 100 units. The performance of the different service curves are calculated and presented after taking the average of 100 independently-generated domains. In our simulations, we use the relative area distance compared with the BP or WP to judge these approaches in order to eliminate the influence of the different domains. Since the n.a.d. of the BP is always zero, the relative n.a.d. is computed compared with the WP instead.

The approximation accuracies of different aggregation approaches are presented in Figure 4. While KK and LS have fixed space requirements, PC, CS and PL are adjustable as to the aggregated state space. In our simulations, all these approaches use 8 float numbers for fairness. Our result shows that PC, CS, and PL outperform LS and KK. While the performance of PC and CS is comparable, PL is the best among all these approaches. Its area distance is roughly one fourth while its space requirement is only doubled compared with LS in our simulation. KK has significant positive area distance, close to 70% that of BP. Its negative area distance is always zero given its special design nature.

Figure 5 shows the accuracy/space tradeoffs by PC, CS, and PL. In this simulation, the domain size ranges from 50 to 250 nodes, following a uniform distribution. The horizontal axis is the space (number of floats) used to store the service curve. The results for CS are missing when the number of floats is 6 due to the fact that the minimal space requirement of CS is 7. When the space increases, all these approaches achieve better approximation accuracy. The figures demonstrate that PL has the best rate of improvement when the space is relatively small and it achieves the best accuracy/space tradeoff compared to CS and PC approaches.

V. Conclusion

In this paper, we propose a new way of representing the network state aggregation in delay-bandwidth sensitive networks. The QoS state information can be best described as a service curve supporting $\oplus$ and $\odot$ operations and the aggregation of the state information is considered as a process applying these two operations reccursively. The detailed algebra together with the algorithms of the recursive aggregation of the state information in multi-level hierarchical networks has been discussed. We approximate the service curve by different approaches, that is, the polynomial curve, the cubic spline, and the polyline approach. These new curves achieve much better accuracy in approximating the original state information both globally and locally, depending on the space used to store the curves. Better accuracy is achieved by larger space. The extensive simulation results shows that the new approximation approaches significantly outperform Korkmaz-Krunz algorithm [3] and Line Segment algorithm [4]. Among our new approaches, polyline has the best performance when the required space size increases.

References

[1] W. C. Lee, "Spanning Tree Method for Link State Aggregation in Large Communication Networks," in *IEEE INFOCOM'95*, vol. 1, Boston, MA, USA, Apr. 1995, pp. 297–302.

[2] A. Iwata, H. Suzuki, R. Izmailov, and B. Sengupta, "QoS aggregation algorithms in hierarchical ATM networks," in *ICC'1998*, vol. 1, Atlanta, GA, USA, June 1998, pp. 243–248.

[3] 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, no. 4, pp. 295–325, 2000.

[4] K.-S. Lui, K. Nahrstedt, and S. Chen, "Routing with Topology Aggregation in Delay-Bandwidth Sensitive Networks," in *to appear in IEEE Transactions on Networking*.

[5] P. Dierckx, *Curve and Surface Fitting with Splines*. Oxford, England: Oxford University Press, Inc., 1993.

[6] B. M. Waxman, "Routing of Multipoint Connections," *IEEE Journal of Selected Areas in Communications*, pp. 1617–1622, Dec. 1988.

