A Service Curve Model With Loss*

Sami Ayyorgun
Research & Development in Advanced Network Technology
Los Alamos National Laboratory
P.O. Box 1663, MS D451
Los Alamos, New Mexico 87545
sami@lanl.gov

Rene L. Cruz
Dept. of Electrical & Computer Engineering
University of California, San Diego
9500 Gilman Drive
La Jolla, CA 92093-0407
cruz@ece.ucsd.edu

June 2003

Abstract—Real-time multimedia applications such as Internet phone and video conferencing are very sensitive to packet delay, jitter, and loss. The performance of protocols for these applications, such as RTP, are also affected by the severity of these phenomenons. A few traffic and service models focusing on the delay and/or jitter aspects of multimedia applications have been previously proposed in the literature. These models, for the most part, do not allow for packet loss, while advantages could be taken from the tolerance of these applications to loss up to a certain degree.

We propose a new service model based on *service curves*, which has a loss aspect. In this model, instead of forcing all the packets to meet their deadlines assigned via a service curve, we allow some packets to be dropped. Specifically, the new model is based on guaranteeing at least a certain fraction of the all packets to meet their deadlines assigned via a service curve. We show that the proposed model has attractive properties such as being *composable*. We also find a condition to employ for an efficient *connection admission control* at a multiplexer delivering services according to the new model. Specifically, we find a necessary and sufficient condition, via a graph-theoretic approach, for the delivery of the services according to the new model, at a multiplexer. Finally, we propose a scheduling algorithm to deliver the services as specified by the new service model, at a multiplexer.

1 Introduction

Most real-time multimedia applications such as Internet phone and video conferencing are very sensitive to packet delay, jitter, and loss. The performance of protocols for these applications, such as RTP, are also affected by the severity of these undesired phenomenons. A few service and traffic models focusing on the delay and/or jitter aspect of multimedia applications have been previously proposed in the literature, e.g., [1, 2, 3, 4, 5]. For the most part, these models do not allow for packet loss, while on the other hand advantages could be taken from the tolerance of these applications to loss up to a certain degree. For example for Internet phone, it has been reported that packet losses up to 20 percent could be tolerated, depending on how the voice is encoded and transmitted [6].

One of the service models mentioned above is called the *service curve model*, and is introduced in [1, 2]. Adopting a discrete-time formulation, a network element is said to deliver a service curve S to an input flow R if the corresponding output flow G satisfies the following inequality for all n

$$G(n) \geqslant \min_{k \leqslant n}\{R(k) + S(n-k)\}$$

where S is a non-decreasing function defined from the integers to the non-negative integers, and takes on the value zero for non-positive values (i.e., $S(n) = 0$ for all $n \leqslant 0$).

Service curves in practice provide a mechanism to systematically assign a deadline to each incoming packet at a network element. A network element delivering a service curve to an incoming flow makes sure that *all* the packets meet their deadlines (i.e., served by no later than their deadlines)

*Los Alamos National Laboratory, Technical Report LA-UR-03-3939, June 2003.

assigned via the service curve. In this original service curve model, packet losses have not been considered.

In this study, we propose a new service model based on service curves, which has a loss aspect. In this model, instead of forcing all the packets to meet their deadlines assigned via a service curve, we allow some packets to be dropped. Specifically, the model is based on guaranteeing at least a certain fraction of the all packets to meet their deadlines assigned via a service curve.

We show that this new model has attractive properties such as being *composable*. The composability of a service model is an important property to have, since it facilitates a systematic treatment of performance guarantees in communication networks. A service model is said to be *composable* if the service delivered by a tandem of two network elements could be represented in terms of the services delivered by each of the network elements, where all services are to be described according to a service model at hand.

To further understand this new service model, as well as to help understand its utilization, and to facilitate any possible deployment of services according to this model, we solve a multiplexing problem. In solving this problem, we find a condition to employ for an efficient *connection admission control* at a multiplexer delivering services according to this model. Specifically, we find a necessary and sufficient condition, via a graph-theoretic approach, for the delivery of the services according to this model, at a multiplexer.

The rest of the paper is organized as follows: Section 2 provides a background and convention. Section 3 examines the new service model which is introduced in section 3.2. Section 3.2 also poses a multiplexing problem whose solution is given in section 3.4. Section 3.5 shows that the new service model is *composable*. Section 4 gives remarks about the new service model, and presents a scheduling algorithm to deliver the services specified by the new service model, at a multiplexer. Finally, section 5 provides a conclusion.

2 Background and Convention

We adopt a discrete-time formulation for the simplicity of exposition. Time is slotted into fixed length intervals, and marked by the integers. The unit of transmission for communication is referred to as a *packet*, in this study. A *flow* is a non-decreasing function defined from the integers to the non-negative integers. Flows in this study are assumed to be bounded (i.e., if R denotes a flow, the value $R(n)$ at infinity could be arbitrarily large but is finite; that is, $R(\infty) < \infty$). The value $R(n)$ of a flow R at time n denotes the total number of packets that has arrived by time n (inclusive) for a *connection*. A *network element* is an input-output device that accepts packets at its input, processes them, and delivers them at its output. A network element is said to be *passive* if it does not generate any packet internally. Network elements are assumed to be passive in this study, for the simplicity of exposition. Packets are assumed to be able to instantaneously arrive and depart at a network element, i.e., a whole packet could arrive instantaneously at time k, and later depart at time n where $n \geqslant k$. Note that a packet could depart in the same interval in which it has arrived; this is sometimes referred to as *cut-through* operation. The capacity $c(n)$ of a network element at time n is the total number of packets that it could deliver (serve) at time n. The function c is called the *instantaneous capacity rate* (or, just the *rate*) of the network element. Its cumulative version $C(n)$, where $C(n) = \sum_{k \leqslant n} c(k)$, is called the *cumulative capacity function* (or, the *capacity flow*) of the network element. Finally, all functions are assumed to be defined from the integers to the integers, unless otherwise noted from here on.

For convenience, the end of proofs are marked by '$\blacksquare$', where the mark is flushed to the right margin.

For mathematical clarity, we provide the following definitions here, which have been previously introduced in the literature.

Definition 1 *Let f and g be any two functions. The* min-+ convolution *of f and g, denoted*[1] *as $f \triangledown g$, is defined as*

$$(f \triangledown g)(n) = \min_{k \leqslant n}\{f(k) + g(n-k)\} .$$

The convolution $f \triangledown g$ is read as "f min-convolved with g", or as "the min-plus convolution of f with g".

Definition 2 *A network element is said to deliver a* service curve *S to an input flow R if the corresponding output flow G satisfies*

$$G(n) \geqslant (R \triangledown S)(n) \quad \textit{for all } n,$$

where S is a non-decreasing function defined from the integers to the non-negative integers, and takes on the value zero for non-positive values (i.e., $S(n) = 0$ for all $n \leqslant 0$).

Definition 3 *A flow R is said to conform to an* arrival curve *A, and denoted as $R \sim A$, if it satisfies*

$$R(n) - R(k) \leqslant A(n-k) \quad \textit{for all } k \leqslant n,$$

where A is a non-decreasing function defined from the integers to the non-negative integers, and takes on the value zero for non-positive values (i.e., $A(n) = 0$ for all $n \leqslant 0$).

The above restriction on flow R conforming to an arrival curve A, put by definition 3, could also be equivalently represented as '$R(n) = (R \triangledown A)(n)$ for all n', which could easily be seen as follows;

$$\begin{aligned} R(n) - R(k) &\leqslant A(n-k) && \text{for all } k \leqslant n \\ R(n) &\leqslant R(k) + A(n-k) && \text{for all } k \leqslant n \\ R(n) &\leqslant \min_{k \leqslant n}\{R(k) + A(n-k)\} \\ &= (R \triangledown A)(n) \end{aligned}$$

now since $A(0) = 0$, $(R \triangledown A)(n)$ is also less than or equal to $R(n)$ (since the set over which the minimum is taken above has also the term $R(n)$ corresponding to $k = n$), thus we get

$$R(n) = (R \triangledown A)(n) .$$

Arrival curves have an interesting property. In order to state this property, we would like to give the following definition.

Definition 4 *A function f is said to be* sub-additive *if it satisfies*

$$f(n) \leqslant f(k) + f(n-k) \quad \textit{for all } k \leqslant n.$$

[1] One reason to choose this notation over some others, for example '$*$', is that there are a companion and related other operators to this operator, which are all employed in some other work (e.g., [7]). We believe that this choice of notation provides a better choice of notations for these other operators in a fitting manner.

The above restriction on a sub-additive function f could also be equivalently represented as '$f(n) \leqslant (f \triangledown f)(n)$ for all n'.

One could assume without loss of generality that an arrival is sub-additive, as it could be seen below; it holds for all n that

$$\begin{aligned}
R(n+l) - R(l) &= R(n+l) - R(n+k) + R(n+k) - R(l) && \text{for all } k \leqslant l \\
&\leqslant A(l-k) + A(n+k-l) && \text{for all } k \leqslant l \\
&= A(l-k) + A(n-(l-k)) && \text{for all } k \leqslant l \\
R(n+l) - R(l) &\leqslant \min_{k \leqslant l} \{A(l-k) + A(n-(l-k))\} \\
&= \min_{u \geqslant 0} \{A(u) + A(n-u)\}
\end{aligned}$$

now, since $A(s) = 0$ for all $s \leqslant 0$, and since A is non-decreasing, we also get

$$\begin{aligned}
&= \min_{u \leqslant n} \{A(u) + A(n-u)\} \\
&= (A \triangledown A)(n)\,.
\end{aligned}$$

Hence, if $A(n)$ is greater than $(A \triangledown A)(n)$ for this particular value of n, one could always replace the value of A at n by $(A \triangledown A)(n)$, and obtain another arrival curve for R. Continuing in this fashion, we could assume without loss of generality that an arrival is sub-additive, which we do in this study.

The *backlog* $Q(n)$ at a network element is the total number of packets that reside in the network element at time n, i.e., if R and G denote the aggregates of the flows at the input and at the output of a network element, respectively, then $Q(n) = R(n) - G(n)$.

The *virtual delay* $d(n)$ at time n for an input flow R at a network element is defined as

$$d(n) = \min\{t : t \geqslant 0,\ G(n+t) \geqslant R(n)\}$$

where G is the corresponding output flow. The virtual delay $d(n)$ is basically the delay experienced by the packets arriving at time n, through the network element, if the packets are to be served in the order in which they have arrived.

It is not difficult to show that the backlog and virtual delays at a network element delivering a service curve to an input flow conforming to a bounded arrival curve are also bounded. The backlog is upper-bounded by the summation of the maximum vertical distance between the arrival curve and service curve a flow, over all flows. The virtual delay at any time for a flow is upper-bounded by the maximum horizontal distance between its arrival curve and the service curve. These results could easily be derived, e.g., refer to [2].

3 Service Curves With Loss

In practice, service curves could be viewed as mechanisms to assign deadlines to the packets of an input flow at a network element. A network element delivering a service curve to an input flow delivers the packets of the flow in such a way that *all* the packets meet their deadlines assigned via the service curve in the original definition. This model does not allow for packet losses. However, it would be more realistic if some of the packets were allowed not to meet their deadlines, and as a result be effectively dropped (or, lost).

With this idea in mind, we introduce a new service model based on service curves, which has a loss aspect. However, before we introduce that model, it would help if we examine the original service curve model more closely, which is done in the following section.

3.1 Implications of The Original Service Curve Definition

Consider a network element with an input flow R and the corresponding output flow G. Suppose that the network element delivers a service curve S to R. This only means that the following inequality holds for all n

$$G(n) \geqslant (R \triangledown S)(n)\,,$$

by definition 2.

To examine what the delivery of service curve S means more closely, let us mark the packets of flow R: We mark all the packets arriving at time n arbitrarily, but distinctly, by the integers in $(R(n-1), R(n)]$. Whenever we refer to a marking of all the packets of a flow, we mean the marking carried out in this fashion, unless otherwise noted from here on.

Next, we define the following terms for all k in $[1, R(\infty)]$;

$$\begin{aligned} r_k &= \min\{t : R(t) \geqslant k\} \\ g_k &= \min\{t : G(t) \geqslant k\} \\ D_k &= \min\{t : (R \triangledown S)(t) \geqslant k\}\,. \end{aligned}$$

The term r_k is the time at which the k-th packet arrives at the network element. Similarly, the term g_k is the time at which the k-th departing packet—not necessarily the k-th arriving packet at the input—departs from the network element, or equivalently it is the time by which at least k-packets have departed from the network element. An interpretation for D_k is provided by the following lemma.

Lemma 1 *The condition in definition 2 is equivalent to the statement '$g_k \leqslant D_k$ for all k'.*

This lemma could be read as "the delivery of service curve S to flow R is equivalent to saying that at least k packets depart by no later than D_k, for all k". The proof is given below.

Proof: First, let us show that the condition in definition 2 implies the statement '$g_k \leqslant D_k$ for all k'. Suppose for some k that we have $g_k > D_k$. Then, we also have

$$G(D_k) < k \leqslant (R \triangledown S)(D_k)$$

which clearly constitutes a contradiction to the condition in definition 2 evaluated at time D_k. The first inequality follows by the definition of g_k and the assumption '$g_k > D_k$ for some k', while the second inequality follows by the definition of D_k. Thus, condition in definition 2 should imply the statement '$g_k \leqslant D_k$ for all k'.

Second, we show that the statement '$g_k \leqslant D_k$ for all k' also implies the the condition in definition 2. Suppose for some n, we have $G(n) < (R \triangledown S)(n)$. Let us denote the value $(R \triangledown S)(n)$ by a. Then, we have

$$g_a > n \geqslant D_a$$

which again clearly constitutes a contradiction to the statement '$g_k \leqslant D_k$ for all k' for k equals to a. The first inequality follows by the assumption $G(n) < a$ and by the definition of g_k, while the second inequality follows by the definition of D_k. Thus, the statement '$g_k \leqslant D_k$ for all k' should also imply the the condition in definition 2, which completes the proof. ■

Hence, by lemma 1, we can interpret D_k as the deadline for the k-th *departing* packet. Or equivalently, it is the time by no later than which at least k packets should depart from the network element, for the delivery of the service curve S.

Note once again that packets need not necessarily depart the network element in the order in which they have arrived. However, we know that the k th departing packet should depart the network element by no later than D_k, and certainly it could not depart before the arrival of the k th arriving packet. Thus, the k-th departing packet could depart somewhere in the interval $[r_k, D_k]$. More accurately, it could depart anywhere in the interval $[\max\{r_k, g_{k-1}\}, D_k]$, where $g_0 = r_1$. Note that the k-th departing packet, for any $k > 1$, could well be the very first arriving packet.

However, due to the difficulty of adopting to assign deadlines to the departing packets in carrying out performance analyses, it is widely practiced by researchers instead that deadlines are assigned the incoming packets. This, however, is at the cost of having only a *sufficient* condition for the delivery of service curves in problems of various scenarios. Specifically, this is indicated by the following lemma.

Lemma 2 *Let D_k be the deadline for the k-th arriving packet, i.e., the packet could depart anywhere in the interval $[r_k, D_k]$. If all the arriving packets meet their deadlines (i.e., delivered by no later than their deadlines), then the service curve S is delivered to R.*

Proof: The proof follows immediately by lemma 1.

The lemma 2 does *not* provide an *if-and-only-if* statement (i.e., it is *not* an equivalence); it is only a sufficiency.

3.2 The New Service Model and A Multiplexing Problem

We propose a more realistic service model that not all packets would need to meet their deadlines assigned via a service curve. Specifically, we give definition 5 for which the following explanations are given: Let R be a flow. We mark all the packets of flow R as we have done before at the beginning of section 3.1 (i.e., we mark all the packets arriving at time n arbitrarily, but distinctly, by the integers in $(R(n-1), R(n)]$). The deadline D_k of the k-th arriving packet is also defined as before; i.e., for all k in $[1, R(\infty)]$, we have

$$D_k = \min\{t : (R \triangledown S)(t) \geqslant k\}\,. \tag{1}$$

Definition 5 (The New Service Model) *A service curve S with loss parameter $1-\alpha$ is said to be delivered to an input flow R by a network element,* if and only if *at least an α fraction of all the packets meet their deadlines assigned via the service curve S, where deadlines are assigned as in (1).*

In this model, the packets that are not to be delivered within their deadlines are not delivered at all, and are considered to be *dropped* (or, *lost*).

To further understand this new service model, as well as to help understand its utilization, and to facilitate any possible deployment of services according to this model, we consider solving the following multiplexing problem.

Problem 1 *A set of flows indexed by $I = \{1, 2, \ldots, |I|\}$ is incident onto a network element with capacity rate $c(n)$, where each flow i ($i \in I$) conforms to an arrival curve A_i, and requests a service curve S_i with loss parameter $1 - \alpha_i$ as explained in definition 5. Find a necessary and sufficient condition on the capacity of the network element for the delivery of these service requests.*

For the simplicity of exposition, we assume, $c(n) \geqslant 1$ for all n.

Clearly, a coordination among packet transmissions are needed at the network element in order for these request to be delivered, if these requests are indeed feasible. Specifically, a *scheduling* algorithm is needed to deliver these requests. A *scheduling* is a mapping of all the packets to be delivered, into the integers where the number of packets mapped into n does not exceed the capacity $c(n)$ of the network element at time n.

Scheduling problems could often be represented as a *matching* problem in graph theory. We could also formulate this problem as a *matching* problem, by first representing any scenario being put in place by packet and capacity arrivals at the network element as a *labeled graph* for any realization of the above problem. The problem is then realized by placing appropriate constraints on the labeled graph. Such a *labeled graph* for any realization of the problem is constructed as follows:

> Represent the k-th arriving packet in each flow i as a node with label k, and denote all the nodes labeled for flow i by V_i. Also represent the each capacity of serving a single packet at time n as a node with label n—hence, there are $c(n)$ many nodes with label n—denote this set of nodes by W. The set of edges denoted by E is obtained by joining the every node with label k in V_i to the nodes with labels in $[r_k, D_k]$ in W. After all the edges have been obtained, delete all the isolated nodes in W. Any such labeled graph is viewed as a bipartite graph $G = (M \cup W, E)$, where $M = \bigcup_{i \in I} V_i$.

Note that a matching in the above construction corresponds to a scheduling in the packet problem.

The graph theory problem that we are interested in is then obtained for any realization of the original problem by putting a constraint on the least number of nodes to be *matched* in each V_i; that is, at least an α_i fraction of all the nodes in V_i needs to be matched, for each $i \in I$, for the service requested by flow i to be delivered.

We can solve this graph theory problem by the solution of a more general matching problem on bipartite graphs, provided in the following section.

3.3 A Matching Problem

We give a brief terminology needed for mathematical clarity; first for *sets*, and then for *graphs*.

We denote the difference of set A from set B by $A - B$, i.e., $A - B$ includes the elements in A but not in B. A *partition* of a set A is a set $\{A_1, A_2, \ldots\}$ of non-empty and disjoint sets whose union is A, where each set A_i is called a *part* of the set A. The *cardinality* of a set A is denoted by $|A|$, i.e., $|A|$ denotes the total number of elements in A.

Let V be a finite set, and E be a set of its 2-element subsets. The ordered pair (V, E) is called a *simple* and *undirected* ***graph***, and usually denoted as $G = (V, E)$. The elements of V are called the *nodes*, and those of E are called the *edges*. The two nodes making up an edge are called its *end-nodes*, and are said to be *joined* by the edge. A graph is *simple* if it has no *loop* (i.e., the end-nodes of no edge are the same) and no *multi-edge* (i.e., the end-nodes of no two edges are the same), and *undirected* if none of its edges is an ordered pair—as these could also be easily inferred from the definition of graphs given here. A graph is called ***bipartite*** if its nodes could be partitioned into two sets M and W such that all the edges have one end-node in M and the other in W, in which case the *bipartite graph* is usually denoted as $G = (M \cup W, E)$. The sets M and W are called the *parts* of the bipartite graph G. A ***matching*** in a graph is a set of edges where no two edges have a common end-node. A matching is said to be *maximum* if no other matching in the graph has a larger cardinality. A node is said to be *matched* in a matching if it is an end-node of an edge in the matching, and it is said to be *unmatched* otherwise. Finally, an edge is said to be *included* in a matching if it is an element of the matching, and *excluded* otherwise.

The graph theory problem whose solution we will employ in solving problem 1 is given below—this problem is not set specifically to solve the problem 1.

Problem 2 *A bipartite graph $G = (M \cup W, E)$, a partition $\{V_j : j \in J\}$ of its nodes where each part V_j is a subset of either M or W, and a mapping f of the partition into the positive integers* *are given. Find the cardinality of a maximum matching such that the number of nodes matched in any part of the partition does not exceed the integer to which it is being mapped.*

A solution to this problem is given by the following theorem proven in [8, 9]. An algorithm to find a such maximum matching in G is also suggested in the proof.

Theorem 1 *Let μ denote a such maximum matching in G as described problem 2. The cardinality of μ is then equal to* $\min_{A \subseteq W} \{\sigma(M - \Pi(A)) + \sigma(A)\}$.

The following definitions are needed to be able to read the theorem; the first two are general considering a graph $G = (V, E)$, the last one is specific to this problem:

1. The ***image*** $\Gamma(A)$ of a set of nodes A is the set of all the nodes that are joined to a node in A.[2]

2. The ***principal*** $\Pi(A)$ of a set of nodes A is the set of all the nodes whose image is a subset of A.

3. A function σ, called the ***permission***, is defined on the power set of $M \cup W$ as follows;

$$\sigma(A) = \sum_{j \in J} \min\{f(V_j), |A \cap V_j|\}.$$

   The name of the function σ follows since $\sigma(A)$ is clearly the maximum number of nodes in A, that is allowed to be matched in G.

3.4 A Solution To Problem 1

The new service model that we have introduced in section 3.2 requires only that at least a certain fraction of the packets meet their deadlines assigned via a service curve.

In problem 1, it is required that at least an α_i fraction of all the packets meet their deadlines. It is clear that these request are feasible *if and only* if there exists a scheduling delivering the minimum number of packets specified by these requests exactly. Specifically, these requests are feasible *if and only* if there exists a matching in the corresponding labeled graph (constructed after problem 1 in section 3.2) with exactly $\lceil \alpha_i |V_i| \rceil$ many nodes being matched in each V_i for all $i \in I$.[3]

In the following section, we will discover a *necessary and sufficient* condition for the existence of a such matching (scheduling) as just described above.

3.4.1 Discovering A Necessary And Sufficient Condition

We can discover the condition for the existence of a such matching (scheduling) as described above by dressing the following instance on problem 2: Construct a labeled graph $G = (M \cup W, E)$, where $M = \bigcup_{i \in I} V_i$ as described after problem 1 in section 3.2, representing a scenario being put in place by packet and capacity arrivals in problem 1. For this labeled graph G, let partition in problem 2 be $\{V_i : i \in I\} \cup \{W\}$, and the mapping f be

$$f(V_i) = \lceil \alpha_i |V_i| \rceil \quad \text{for all } i \in I.$$
$$f(W) = |W| \ .$$

[2] The *image* is also referred to as *neighborhood*, in graph theory.

[3] The notation $\lceil x \rceil$ denotes the least integer greater than or equal to x, and is read as "ceiling of x". Similarly, $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x, and read as "floor of x".

Clearly, we could assume without any loss of generality that $\alpha_i \leqslant 1$, for all $i \in I$.

We can find the cardinality of the maximum matching satisfying the constraints (i.e., no more than $f(V_i)$ many nodes being matched in each V_i) for this instance of the problem 2, by theorem 1 as well. Now, if we set the cardinality of a such maximum matching to be equal to $\sum_{i \in I} f(V_i)$, that would give us a *necessary and sufficient* condition for the existence of a matching with exactly $f(V_i)$ many nodes being matched in each V_i, by the *pigeonhole principle.*

First note that the quantity $\sum_{i \in I} f(V_i)$ is equal to the permission of M (that is, to $\sigma(M)$), as could be seen below;

$$\begin{aligned}
\sigma(M) &= \sum_{j \in J} \min\{f(V_j), |M \cap V_j|\} \\
&= \left(\sum_{i \in I} \min\{f(V_i), |M \cap V_i|\}\right) + \min\{f(W), |M \cap W|\} \\
&= \left(\sum_{i \in I} \min\{f(V_i), |V_i|\}\right) + \min\{f(W), |\emptyset|\} \\
&= \sum_{i \in I} \min\{\lceil \alpha_i |V_i| \rceil, |V_i|\} \\
&= \sum_{i \in I} \lceil \alpha_i |V_i| \rceil \\
&= \sum_{i \in I} f(V_i).
\end{aligned}$$

Thus, setting the cardinality of a such maximum matching in theorem 1 equal to $\sum_{i \in I} f(V_i)$, means setting it equal to $\sigma(M)$.

Second, note that the term $\sigma(M)$ is always an element of the set of which the minimum is taken in theorem 1, which corresponds to setting $A = \emptyset$. In other words, the cardinality of a such maximum matching is always less than or equal to $\sigma(M)$. Thus, the value of the minimum could indeed be $\sigma(M)$ if none of the other terms in the minimum is smaller than $\sigma(M)$. Let us write this condition down, and manipulate it until we obtain a necessary and sufficient condition that we would like to stop: All of the following statements are equivalent to each other for all $A \subseteq W$,

$$
\begin{aligned}
\sigma(M) &\leqslant \sigma(M - \Pi(A)) + \sigma(A) \\
\sigma(M) - \sigma(M - \Pi(A)) &\leqslant \sigma(A) \\
\sigma(M) - \sigma(M - \Pi(A)) &\leqslant |A| \\
\sum_{i \in I} f(V_i) - \sum_{i \in I} \min\{f(V_i), |(M - \Pi(A)) \cap V_i|\} &\leqslant |A| \\
\sum_{i \in I} f(V_i) - \sum_{i \in I} \min\left\{f(V_i), |V_i - \Pi(A)|\right\} &\leqslant |A| \\
\sum_{i \in I} f(V_i) - \sum_{i \in I} \min\left\{f(V_i), |V_i| - |\Pi_i(A)|\right\} &\leqslant |A| \quad \text{where } \ \Pi_i(A) \triangleq \Pi(A) \cap V_i \\
\sum_{i \in I} \max\left\{0, f(V_i) - |V_i| + |\Pi_i(A)|\right\} &\leqslant |A| \\
\sum_{i \in I} \max\left\{0, |\Pi_i(A)| - \lfloor (1 - \alpha_i)\,|V_i| \rfloor\right\} &\leqslant |A| \\
\max\{m_S(A) : S \subseteq I\} &\leqslant |A| \quad \text{where } \ m_S(A) \triangleq \sum_{i \in S} |\Pi_i(A)| - \lfloor (1 - \alpha_i)\,|V_i| \rfloor \\
m_S(A) &\leqslant |A| \quad \text{for all } S \subseteq I.
\end{aligned}
$$

Note that for $S = \emptyset$, the term $m_S(A)$ is equal to 0 for any A, by convention.

Hence, we obtain that the following condition

$$m_S(A) \leqslant |A| \qquad \text{for all } A \subseteq W, \text{ and for all } S \subseteq I \tag{2}$$

is both necessary and sufficient for the existence of a matching (scheduling) that exactly (at least) α_i fraction of all the nodes (packets) in V_i (flow i), for all $i \in I$, are matched (delivered) in any labeled graph as constructed in section 3.2.

3.4.2 The Condition In Terms Of Service Curves, Arrival Curves, and Capacity

In this section, we will find a necessary and sufficient condition in terms of the service curves S_i's and the arrival curves A_i's in problem 1 for the delivery of the services requested by the flows, according to the new service model. We will do this by distilling condition (2).

First of all, note that if we delete all the nodes with label k in a set A in condition (2) that A does not include all the nodes with label k in W, then the left-hand-side of the inequality (that is, $|\Pi_S(A)|$) does not change its value. This holds since the image of a node includes *all* the nodes with label k, if it includes one of them. Hence, by this observation, note that by $|A|$ we are actually effectively representing the total capacity of the network element over a union of *intervals* of time.

To be able to express what we have discussed above compactly, we introduce the following definition: A set of nodes in W is said to be a ***block*** if it includes *all* the nodes in W with labels less than or equal to m and greater than or equal to k, if it includes a node with label k and a node with label m where $k \leqslant m$. Let us denote a block generically by B (suggesting 'Block') unless otherwise noted from here on.

Thus, we have effectively shown the truth of the following lemma.

Lemma 3 *The condition (2) is equivalent to*

$$m_S(A) \leqslant |A| \qquad \textit{for all } A \subseteq W,\ A \textit{ is a union of blocks, and for all } S \subseteq I. \tag{3}$$

Proof: It is clear that (2) $\Rightarrow$ (3), since all the sets A in (3) are also in (2). It is also clear that (3) $\Rightarrow$ (2), since including subsets to a set A in (3), where subsets are not blocks, will only increase the right-hand-side of the inequality, while the left-hand-side remains the same, as we have explained previously. ■

Next, we claim and prove the following lemma.

Lemma 4 *The condition (3) is equivalent to*

$$\sum_{i\in I} \lceil \alpha_i \left|\Pi_i(A)\right| \rceil \leqslant |A| \qquad \textit{for all } A\subseteq W, \textit{ A is a union of blocks.} \tag{4}$$

Proof: The condition (3) implies the condition (4), since condition (3) holds for *all* $R_i \sim A_i$, $i \in I$. Specifically, note that for any A as specified in lemma 3, $m_S(A)$ which we know is equal to

$$m_S(A) = \sum_{i\in S} \left|\Pi_i(A)\right| - \lfloor (1-\alpha_i)\left|V_i\right| \rfloor ,$$

will attain its maximum for some R_i's, where $R_i \sim A_i$, such that no packet has arrived other than the principals of A (i.e., $\Pi_i(A)$) in flow i for any $i \in S$. This holds since $|\Pi_i(A)|$ does not change its value if packets other than other than the principals of A would also arrive for any such flow, while on the other hand the value of $|V_i|$ increases in the summand in $m_S(A)$. Thus, in that case, the value of $m_S(A)$ would decrease. Hence, the maximum of $m_S(A)$ is indeed attained for some flows where no packet has arrived other than the principals of A as specified in condition (3).

For flows where no packet has arrived other than the principals of A, we could replace the summand in $m_S(A)$ by $\lceil \alpha_i \left|\Pi_i(A)\right| \rceil$, since in that case $|V_i| = |\Pi_i(A)|$. Now, since we see that the sum yielding $m_S(A)$ is non-decreasing by the cardinality of S, we could also replace the subscript '$i \in S$' of the sum by '$i \in I$'. This completes one direction of the proof; i.e., (3) $\Rightarrow$ (4).

The other direction, (4) $\Rightarrow$ (3), clearly follows by the explanations given for the first direction, by backtracking them. ■

Since the image of each node (packet) in M is contiguous, we also have the following lemma.

Lemma 5 *The condition (4) is equivalent to*

$$\sum_{i\in I} \lceil \alpha_i \left|\Pi_i(B)\right| \rceil \leqslant |B| \qquad \textit{for all } B\subseteq W, \textit{ B is a block.} \tag{5}$$

Proof: It is clear that (4) $\Rightarrow$ (5), since all the sets B in (5) are also sets in (4).

The other direction, (5) $\Rightarrow$ (4), also holds since the image of each node in M is contiguous. Specifically, note that any A which is not a block, in condition (4), could be represented as the union of two other sets A_1 and A_2, which are also as specified in condition (4), that at least one node w in W exists with a label greater than all the labels of the nodes in one of the sets (A_1 or A_2) and less than all the labels of the nodes in the other set. Then, note that we have for any $i \in I$

$$\Pi_i(A) = \Pi_i(A_1) \cup \Pi_i(A_2) \quad \text{and} \quad \Pi_i(A_1) \cap \Pi_i(A_2) = \emptyset ,$$

since the image of each node in M is contiguous. More specifically, for any flow i, $\Pi_i(A)$ can not include a node whose image has nodes in both A_1 *and* A_2, since in that case its image would also include the node w which is not in A. Therefore, such a node can not be a principal of A. Continuing in this fashion (i.e., representing A_1 and A_2 similarly, until all the parts of a set to be represented become blocks), we could represent A as

$$A = \bigcup_{j\in J} B_j \qquad \text{where each } B_j \text{ is a block}$$

where $J = \{1, 2, \ldots, |J|\}$, $|J| > 1$, and there is at least one node in W between any two consecutive blocks B_j of A (that is, if we assume without loss of generality that all the labels of the nodes in B_i are less than that of B_j whenever $i < j$, there is at least one node w_j in W between B_j and B_{j+1}, for all $j < |J|$, that the label of w_j is greater than the labels of the nodes in B_j and less than the labels of the nodes in B_{j+1}).

Thus, we have

$$\begin{aligned}
\Pi_i(A) &= \bigcup_{j \in J} \Pi_i(B_j) && \Pi_i(B_k) \cap \Pi_i(B_l) = \emptyset \text{ for all } k \neq l, \text{ and for all } i \in I \\
|\Pi_i(A)| &= \sum_{j \in J} |\Pi_i(B_j)| && \text{for all } i \in I \\
\alpha_i\,|\Pi_i(A)| &= \sum_{j \in J} \alpha_i\,|\Pi_i(B_j)| && \text{for all } i \in I \\
\lceil \alpha_i\,|\Pi_i(A)| \rceil &\leqslant \sum_{j \in J} \lceil \alpha_i\,|\Pi_i(B_j)| \rceil && \text{for all } i \in I \ \ (\text{since, } \lceil a + b \rceil \leqslant \lceil a \rceil + \lceil b \rceil \text{ for any } a \text{ and } b)^4 \\
\sum_{i \in I} \lceil \alpha_i\,|\Pi_i(A)| \rceil &\leqslant \sum_{i \in I} \sum_{j \in J} \lceil \alpha_i\,|\Pi_i(B_j)| \rceil \\
&= \sum_{j \in J} \sum_{i \in I} \lceil \alpha_i\,|\Pi_i(B_j)| \rceil \\
&\leqslant \sum_{j \in J} |B_j| && \text{by condition (5)} \\
&= |A|
\end{aligned}$$

which completes the proof. ■

In light of lemma 5, it suffices to find the maximum of the summand in (5) over all $R_i \sim A_i$, in order to find a necessary and sufficient condition in terms of the service curves S_i's and the arrival curves A_i's in problem 1 for the delivery of the services requested by the flows according to our new service model. This is provided by the next lemma. Before we give that lemma, note that maximizing the summand in (5) is the same as maximizing $|\Pi_i(B)|$ over all $R_i \sim A_i$.

Lemma 6 *For any block B in condition (5), whose nodes' labels are all greater than or equal to $k+1$ and less than or equal to $k+n$, for some k and $n \geqslant 1$, we have for any $i \in I$*

$$\max_{R_i \sim A_i} \big\{\, |\Pi_i(B)| \,\big\} = (A_i \triangledown S_i)(n)\,.$$

Proof: First of all, note that for any $i \in I$, the flow $R_i(u) = A_i(u - k)$ also conforms to arrival curve A_i, since surely shifting does not change the burstiness of a flow. Whereby also note that the maximum in the lemma is also greater than or equal to $|\Pi_i(B)|$ for this particular flow. Notice that for this flow, $\Pi_i(B)$ actually includes the first $(R_i \triangledown S_i)(n + k)$ many packets—which is ensured by

[4] Refer to Appendix A for a simple derivation, if need be.

the capacity arrival at time $n+k+1$. Thus, we have

$$
\begin{aligned}
\max_{R_i \sim A_i} \{\, |\Pi_i(B)| \,\} \geqslant |\Pi_i(B)| \\
&= (R_i \triangledown S_i)(n+k) \\
&= \min_{u \leqslant n+k} \{R_i(u) + S_i(n+k-u)\} \\
&= \min_{u \leqslant n+k} \{A_i(u-k) + S_i(n+k-u))\} \\
&= \min_{u-k \leqslant n} \{A_i(u-k) + S_i(n-(u-k))\} \\
&= \min_{v \leqslant n} \{A_i(v) + S_i(n-v))\} \\
&= (A_i \triangledown S_i)(n)\,. \qquad (*)
\end{aligned}
$$

For the other direction, note that for any $i \in I$ and R_i, we have

$$|\Pi_i(B)| = \max\{0, (R_i \triangledown S_i)(n+k) - R_i(k)\}$$

since at most the first $(R_i \triangledown S_i)(n+k)$ many packets could be among the principals of B, while the first $R_i(k)$ many packets could not, as they have arrived before time $k+1$ hence their image have nodes not in B—these are again ensured by the capacity arrivals at time k and $n+k+1$. Thus, we have

$$
\begin{aligned}
|\Pi_i(B)| &= \max\{0, (R_i \triangledown S_i)(n+k) - R_i(k)\} \\
&= \max\{0, \min_{u \leqslant n+k}\{R_i(u) + S_i(n+k-u)\} - R_i(k)\} \\
&\leqslant \max\{0, \min_{k \leqslant u \leqslant n+k}\{R_i(u) + S_i(n+k-u)\} - R_i(k)\} \\
&= \max\{0, \min_{k \leqslant u \leqslant n+k}\{R_i(u) - R_i(k) + S_i(n+k-u)\}\} \\
&\leqslant \max\{0, \min_{k \leqslant u \leqslant n+k}\{A_i(u-k) + S_i(n+k-u)\}\} \\
&= \max\{0, \min_{0 \leqslant u-k \leqslant n}\{A_i(u-k) + S_i(n-(u-k)\}\} \\
&= \max\{0, \min_{0 \leqslant v \leqslant n}\{A_i(v) + S_i(n-v)\}\} \\
&= \min_{0 \leqslant v \leqslant n}\{A_i(v) + S_i(n-v)\}
\end{aligned}
$$

since $A_i(v) = 0$ for all $v \leqslant 0$, and since S_i is non-decreasing, we also have

$$= (A_i \triangledown S_i)(n)\,.$$

Since $|\Pi_i(B)|$ is less than or equal to $(A_i \triangledown S_i)(n)$ which is independent from the arrival pattern of R_i other than the fact that it conforms to A_i, so is its maximum over all $R_i \sim A_i$. Together with the first inequality tagged $(*)$ above, this completes the proof. ■

With lemma 6, we could now give another condition equivalent to condition 5 in lemma 5, which is also the condition in the following theorem. This condition is necessary and sufficient for the existence of a scheduling for the delivery of service requests according to the new model.

Theorem 2 *The following condition is necessary and sufficient for the existence of a scheduling for the delivery of the service requests as stated in problem 1;*

$$\sum_{i \in I} \lceil \alpha_i\,(A_i \triangledown S_i)(n) \rceil \leqslant C(n+k) - C(k) \qquad \text{for all } n \geqslant 0, \text{ and } k.$$

Proof: The proof follows by the arguments and condition (2) in section 3.4.1, and all the preceding lemmas in this section. ■

3.5 Composability of The New Service Model

The composability of a service model is an important property to have, since it facilitates a systematic treatment of performance guarantees in communication networks. A service model is said to be *composable* if the service delivered by a tandem of two network elements could be represented in terms of the services delivered by each of the network elements, where all services are to be described according to a service model at hand.

In this section, we will show that the new service model provided by definition 5 is composable. Specifically, we will prove the following theorem stating the composability of the new model.

Theorem 3 *Let two network elements,* 1 *and* 2*, be in tandem. Let* R_i *and* R_{i+1} *be the input and output flows of the network element* i*, respectively for* i *equals to both* 1 *and* 2*. Furthermore, let network element* i *deliver a service curve* S_i *with loss parameter* $1-\alpha_i$ *to flow* R_i*, again for* i *equals to both* 1 *and* 2*. Then, the service curve* $S_1 \triangledown S_2$ *with loss parameter* $1-\alpha_1\alpha_2$ *is also delivered to flow* R_1 *by the tandem of network elements (i.e., with respect to flow* R_3*).*

In order to prove theorem 3, we would like to provide the following definitions and the lemmas.

We define a *loss operator* L which operates on flows. The operator L is defined for a set of (packet) intervals indexed by a sequence $\mathbf{a} = \langle a_1, a_2, \ldots, a_{2r-1}, a_{2r}\rangle$, where $r \geqslant 0$ and a_i's are non-negative integers, furthermore $a_1 < a_2 < \cdots < a_{2r}$. Let $\mathbf{a}$ denote a such sequence from here on, and let r be called its *rank*. The operator L operating on a flow R cuts the packets of R such for the interval $(a_{2i-1}, a_{2i}]$, all of the $a_{2i} - a_{2i-1}$ many packets that would arrive after the first a_{2i-1} many packets in R are eliminated (cut) from the flow R, for all $1 \leqslant i \leqslant r$. In other words, if we mark the packets of flow R (as we have done before at the beginning of section 3.1), the operator L operating on a flow R eliminates the packets of R with marks falling in $(a_{2i-1}, a_{2i}]$ for all $1 \leqslant i \leqslant r$. Note that the loss operator L defined for the null sequence $\mathbf{a}$ with zero rank (i.e., the sequence corresponding to r being equal to 0), stands for no loss. The new flow LR obtained by operating L on R, is the flow obtained after the cuts as just explained. The value of the flow LR at time n is denoted by L$R(n)$, and could be calculated as follows

$$\mathrm{L}R(n) = R(n) - \sum_{i:\, a_{2i-1} < R(n)} \left(\min\{a_{2i}, R(n)\} - a_{2i-1}\right) \qquad \text{for all } n.$$

Finally, we call the value $\sum_{i=1}^{r}(a_{2i} - a_{2i-1})$ as the *size* of $\mathbf{a}$, and denote it by $|\mathbf{a}|$.

We also define a set $\mathcal{L}(c)$ of loss operators, with parameter c which stands for the maximum amount of total number of *c*uts (or, losses). The set $\mathcal{L}(c)$ is defined for any non-negative real number c, as

$$\mathcal{L}(c) \triangleq \{\mathrm{L} : \mathrm{L} \text{ defined for an } \mathbf{a},\ |\mathbf{a}| \leqslant c\}\,.$$

We first give a lemma showing that any loss operator L preserves the assignment of deadlines via a service curve. More precisely, this is stated in the following lemma for which the following explanations are given: Let R be a flow, S be a service curve, and L be a loss operator defined for $\mathbf{a} = \langle a_1, a_2, \ldots, a_{2r-1}, a_{2r}\rangle$. Mark all the packets of R and LR (as described earlier at the beginning of section 3.1). Let D_k be the deadline of the k-th packet of R, as defined by equation (1). Similarly, let $\mathcal{D}_k$ be the deadline of the k-th packet of LR, which is defined below for all k in $[1, \mathrm{L}R(\infty)]$

$$\mathcal{D}_k \triangleq \min\{t : \mathrm{L}(R \triangledown S)(t) \geqslant k\}\,. \tag{6}$$

Note that a packet with mark k in R, where k is not in $(a_{2i-1}, a_{2i}]$ for any $i \leqslant r$, is the packet with mark k' in LR, where

$$k' = k - \sum_{i:\, a_{2i} < k} (a_{2i} - a_{2i-1})\,. \tag{7}$$

With these explanations, we claim the following lemma.

Lemma 7 *Given a flow R, a service curve S, and a loss operator* L *corresponding to* $\mathbf{a} = \langle a_1, a_2, \ldots, a_{2r-1}, a_{2r}\rangle$*, there holds*

$$\mathcal{D}_{k'} = D_k$$

for all k in $[1, R(\infty)]$, but k is not in $(a_{2i-1}, a_{2i}]$ for any $i \leqslant r$, where all the explanations are as given in the preceding paragraph.

Proof: Considering the flow R, it holds by the definition of deadline D_k given by equation (1) that for any packet k, we have

$$(R \triangledown S)(D_k) \geqslant k\,.$$

Since we know that any packet of flow R with mark k not in $(a_{2i-1}, a_{2i}]$ for any $i \leqslant r$, is the packet with mark k' given by the conversion (7), in flow LR, we have

$$\begin{aligned} (R \triangledown S)(D_k) &\geqslant k \\ (R \triangledown S)(D_k) - \sum_{i:\, a_{2i} < k} (a_{2i} - a_{2i-1}) &\geqslant k - \sum_{i:\, a_{2i} < k} (a_{2i} - a_{2i-1}) \\ \mathrm{L}(R \triangledown S)(D_k) &\geqslant k'\,. \end{aligned}$$

Thus, $\mathcal{D}_{k'} \leqslant D_k$.

For the other direction, consider a packet with mark k' in LR, and the packet in flow R corresponding to packet k' in flow LR, whose mark is to be uniquely determined by (7). Then, we similarly have

$$\begin{aligned} \mathrm{L}(R \triangledown S)(\mathcal{D}_{k'}) &\geqslant k' \\ \mathrm{L}(R \triangledown S)(\mathcal{D}_{k'}) + \sum_{i:\, a_{2i} < k} (a_{2i} - a_{2i-1}) &\geqslant k' + \sum_{i:\, a_{2i} < k} (a_{2i} - a_{2i-1}) \\ (R \triangledown S)(\mathcal{D}_{k'}) &\geqslant k\,. \end{aligned}$$

Thus, we have $D_k \leqslant \mathcal{D}_{k'}$, which completes the proof. ∎

Next, we give a lemma showing that the new service model provided by definition 5 could also be represented equivalently in terms of the loss operator defined in this section.

Lemma 8 *A service curve S with loss parameter $1-\alpha$ is delivered to an input flow R by a network element, as described by definition 5, if and only if*

$$\mathrm{L}R(n) \geqslant G(n) \geqslant \mathrm{L}(R \triangledown S)(n) \quad \textit{for all } n, \textit{ and for any } \mathrm{L} \textit{ in } \mathcal{L}\left((1-\alpha)R(\infty)\right) \tag{8}$$

where G is the corresponding output flow, and the packets of flow LR *are served by their deadlines as assigned via equation (6).*

Proof: The proof follows almost immediately by definition 5 and the definitions of a loss operator L and the set $\mathcal{L}\,(c)$.

Let us first show the direction that definition 5 implies condition (8). Since a service curve S with loss parameter $1-\alpha$ is delivered to the input flow R, that means at most $\lfloor (1-\alpha)R(\infty) \rfloor$ many packets are dropped, while the rest of the packets are delivered within their deadlines. Notice that for any loss pattern that would happen according to this criteria on R, a loss operator L exists in $\mathcal{L}\left((1-\alpha)R(\infty)\right)$, which equivalently represents the loss pattern.

Also, notice that the output flow G obviously could not be larger than $\mathrm{L}R(n)$, i.e., than the output flow when all the packets to be served are delivered as soon as they have arrived. Similarly, it could not be less than the flow where all the packets that are to be served are delivered exactly at their deadlines. Note that this later output flow is equal to $\mathrm{L}(R \triangledown S)(n)$, since the deadlines assigned via the service curve S is preserved by the loss operator L, as shown by lemma 7. This completes the proof for the first direction.

The proof for the other direction follows similarly. Any loss operator L in $\mathcal{L}\left((1-\alpha)R(\infty)\right)$ corresponds to a loss pattern that at most $\lfloor (1-\alpha)R(\infty) \rfloor$ many packets are lost in flow R. By lemma 7, all of the remaining packets in flow $\mathrm{L}R$ have the same deadlines as to be assigned via the service curve S on R. By condition (8), all of the remaining packets meet their deadlines, thus the service curve S with loss parameter $1-\alpha$ is delivered to flow R, which completes the proof. ■

The next lemma given below provides an equivalent representation of the relation $F(n) \leqslant G(n)$ for all n, for any two flows F and G. The following notations are provided for the lemma;

$$\begin{aligned} f_k &\triangleq \min\{t : F(t) \geqslant k\} \qquad \text{for all } k \in [1, F(\infty)] \\ g_k &\triangleq \min\{t : G(t) \geqslant k\} \qquad \text{for all } k \in [1, G(\infty)]. \end{aligned} \tag{9}$$

Lemma 9 *Let F and G be any two flows. There holds;*

$$F(n) \leqslant G(n) \quad \textit{for all } n \qquad \Leftrightarrow \qquad f_k \geqslant g_k \quad \textit{for all } k \in [1, F(\infty)].$$

Proof: Let us first show the direction that the left-hand-side implies the right-hand-side. Suppose for some $k \in [1, F(\infty)]$ that we have $f_k < g_k$. Then, we also have

$$G(f_k) < k \leqslant F(f_k)$$

which clearly constitutes a contradiction to the condition '$F(n) \leqslant G(n)$ for all n' at time f_k. The first inequality follows by the definition of g_k and the assumption '$f_k < g_k$ for some k', while the second inequality follows by the definition of f_k. Thus, the left-hand-side of the claimed equivalence implies the right-hand-side.

For the other direction, suppose for some n that $F(n) > G(n)$. Let us denote the value $F(n)$ by a. Then, we have

$$f_a \leqslant n < g_a$$

which clearly constitutes a contradiction to the condition '$f_k \geqslant g_k$ for all $k \in [1, F(\infty)]$' for k equals to a. The first inequality follows by the definition of f_k, while the second inequality follows by the definition of g_k and the assumption '$F(n) > G(n)$ for some n'. Thus, the right-hand-side of the claimed equivalence also implies the left-hand-side, which completes the proof. ■

The next two lemmas given below provide two of the properties of the loss operator L.

Lemma 10 *Let F and G be any two flows that $F(n) \leqslant G(n)$ for all n. There holds for any given loss operator* L *that*

$$\mathrm{L}F(n) \leqslant \mathrm{L}G(n) \quad \textit{for all } n.$$

Proof: Mark the packets of flow F and G (as we have done before at the beginning of section 3.1). Note by lemma 9 that since $F(n) \leqslant G(n)$ for all n, we have $f_k \geqslant g_k$ for all $k \in [1, F(\infty)]$, where f_k and g_k are as defined in (9).

Let L be a loss operator defined for $\mathbf{a} = \langle a_1, a_2, \ldots, a_{2r-1}, a_{2r}\rangle$. The operator L eliminates the packets of flows F and G with marks falling in $(a_{2i-1}, a_{2i}]$ for all $1 \leqslant i \leqslant r$. Obviously, the arrival times of the remaining packets in $\mathrm{L}F$ and $\mathrm{L}G$ do not change by the loss operator L; only their marks change. Thus, if we let k' be the mark of a packet in $\mathrm{L}F$, where k' can be calculated by equation 7, we would still have

$$f_{k'} \geqslant g_{k'} \quad \text{for all } k' \in [1, \mathrm{L}F(\infty)].$$

Thus, it also follows by lemma 9 that $\mathrm{L}F(n) \leqslant \mathrm{L}G(n)$ for all n. ■

Lemma 11 *Let F and G be any two flows. There holds for any given loss operator* L *that*

$$(\mathrm{L}F \triangledown G)(n) \geqslant \mathrm{L}(F \triangledown G)(n) \quad \textit{for all } n.$$

Proof: Let L be a loss operator defined for $\mathbf{a} = \langle a_1, a_2, \ldots, a_{2r-1}, a_{2r}\rangle$.

Let us first note that both sides of the above inequality in the lemma are of the following forms for any given n;

$$\begin{aligned}(\mathrm{L}F \triangledown G)(n) &= F(k) - L_1 + G(n-k) \quad \text{for some } k \leqslant n \\ &= F(k) + G(n-k) - L_1 \\ \text{where} \quad L_1 &= \sum_{i:\, a_{2i-1} < F(k)} (\min\{a_{2i}, F(k)\} - a_{2i-1})\end{aligned}$$

and

$$\begin{aligned}\mathrm{L}(F \triangledown G)(n) &= F(l) + G(n-l) - L_2 \quad \text{for some } l \leqslant n \\ &= \min_{l \leqslant n}\{F(l) + G(n-l)\} - L_2 \\ \text{where} \quad L_2 &= \sum_{i:\, a_{2i-1} < (F \triangledown G)(n)} (\min\{a_{2i}, (F \triangledown G)(n)\} - a_{2i-1}).\end{aligned}$$

Secondly, note that the min-+ convolution $(\mathrm{L}F \triangledown G)(n)$ is calculated in general over a less number of terms than that of $\mathrm{L}(F \triangledown G)(n)$. More precisely, note that the min-+ convolution $(\mathrm{L}F \triangledown G)(n)$ can be calculated as follows: Mark the packets of flows F and $\mathrm{L}F$ (as we have done before at the beginning of section 3.1). Denote the mark of a packet of flow $\mathrm{L}F$ by k', where $k' \in [1, \mathrm{L}F(\infty)]$. We have

$$(\mathrm{L}F \triangledown G)(n) = \min_{t \in T_n}\{F(t) + G(n-t) - L_t\} \tag{10}$$

where

$$\begin{aligned}T_n &= \{n\} \cup \big\{f_{k'} - 1 : f_{k'} \leqslant n,\ k' \in [1, \mathrm{L}F(\infty)]\big\} \\ f_{k'} &= \min\{t : \mathrm{L}F(t) \geqslant k'\} \quad \text{for } k' \in [1, \mathrm{L}F(\infty)] \\ L_t &= \sum_{i:\, a_{2i-1} < F(t)} (\min\{a_{2i}, F(t)\} - a_{2i-1}).\end{aligned}$$

The equality in (10) holds since both F and G are non-decreasing functions.

Note that by (10) we also have

$$\begin{aligned}(\mathrm{L}F \triangledown G)(n) &= F(t^*) + G(n-t^*) - L_{t^*} \qquad \text{for some } t^* \in T_n \\ &\geqslant \min_{t \in T_n^*}\{F(t) + G(n-t)\} - L_{t^*}\end{aligned} \tag{$*$}$$

where

$$T_n^* \triangleq \{t : F(t) + G(n-t) \geqslant F(t^*),\ t \in T_n\}\,.$$

Note that T_n^* is not empty, since we have

$$T_n^* \supseteq \{t : t \geqslant t^*,\ t \in T_n\} \neq \emptyset\,.$$

Let us denote the function obtained by the minimum in line tagged $(*)$ above, by $H(n)$; i.e.,

$$H(n) \triangleq \min_{t \in T_n^*}\{F(t) + G(n-t)\} \quad \text{for all } n.$$

Now, note that since

$$T_n^* \subseteq \{i : 1 \leqslant i \leqslant n\} \quad \text{for any } n,$$

we have

$$H(n) \geqslant (F \triangledown G)(n) \quad \text{for all } n.$$

Hence, by lemma 10 we also have

$$\mathrm{L}H(n) \geqslant \mathrm{L}(F \triangledown G)(n) \quad \text{for all } n.$$

Finally, note that we have for all n

$$\begin{aligned}(\mathrm{L}F \triangledown G)(n) &\geqslant H(n) - L_{t^*} \\ &= F(t) + G(n-t) - L_{t^*} \quad \text{for some } t \in T_n^*\end{aligned}$$

now, since $F(t^*) \leqslant F(t) + G(n-t)$, we get

$$\begin{aligned}&\geqslant F(t) + G(n-t) - \sum_{i:\, a_{2i-1} < F(t)+G(n-t)} \left(\min\{a_{2i}, F(t)+G(n-t)\} - a_{2i-1}\right) \\ &= \mathrm{L}H(n) \\ &\geqslant \mathrm{L}(F \triangledown G)(n)\end{aligned}$$

which completes the proof. ∎

Note that the inequality given by lemma 11 is tight (i.e., there exists flows F and G that the inequality becomes an equality; for example, $F(n) = G(n) = \max\{0, \rho \cdot n\}$).

Finally, we note two very well-known properties of the min-+ convolution. The first one is on the monotonicity of the min-+ convolution, and is given below.

Lemma 12 *Let f, g, and h be any there functions that $f(n) \leqslant h(n)$ for all n. There holds*

$$(f \triangledown g)(n) \leqslant (h \triangledown g)(n) \quad \textit{for all } n.$$

Proof: The proof follows immediately from the definition of min-+ convolution, and is given below for the sake of completeness; it holds for all n that

$$\begin{aligned}(f \triangledown g)(n) &= \min_{k \leqslant n}\{f(k) + g(n-k)\} \\ &\leqslant \min_{k \leqslant n}\{h(k) + g(n-k)\} \\ &= (h \triangledown g)(n)\,.\end{aligned}$$

■

The second one is on the associativity of the min-+ convolution, and is given below.

Lemma 13 *Let f, g, and h be any there functions. There holds*

$$((f \triangledown g) \triangledown h)(n) \geqslant (f \triangledown (g \triangledown h))(n) \quad \text{for all } n.$$

Proof: The proof follows immediately from the definition of min-+ convolution, and is given below for the sake of completeness; it holds for all n that

$$\begin{aligned}((f \triangledown g) \triangledown h)(n) &= \min_{k \leqslant n}\{(f \triangledown g)(k) + h(n-k)\} \\ &= \min_{k \leqslant n}\Big\{\min_{l \leqslant k}\{f(l) + g(k-l)\} + h(n-k)\Big\} \\ &= \min_{k \leqslant n}\Big\{\min_{l \leqslant k}\{f(l) + g(k-l) + h(n-k)\}\Big\} \\ &= \min_{\substack{k \leqslant n \\ l \leqslant k}}\Big\{f(l) + g(k-l) + h(n-k)\Big\}\end{aligned}$$

notice that for a fixed l, $f(l)$ does not change its value, hence the minimum above for that value of l will occur for the minimum of $g(k-l) + h(n-k)$ over all k's; that is, we have

$$\begin{aligned}&= \min_{l \leqslant n}\Big\{f(l) + \min_{l \leqslant k \leqslant n}\{g(k-l) + h(n-k)\}\Big\} \\ &= \min_{l \leqslant n}\Big\{f(l) + \min_{0 \leqslant k-l \leqslant n-l}\{g(k-l) + h(n-k)\}\Big\} \\ &= \min_{l \leqslant n}\Big\{f(l) + \min_{0 \leqslant u \leqslant n-l}\{g(u) + h(n-l-u)\}\Big\} \\ &\geqslant \min_{l \leqslant n}\Big\{f(l) + \min_{u \leqslant n-l}\{g(u) + h(n-l-u)\}\Big\} \\ &= \min_{l \leqslant n}\{f(l) + (g \triangledown h)(n-l)\} \\ &= (f \triangledown (g \triangledown h))(n)\,.\end{aligned}$$

■

We would like to note that the relation in lemma 13 is in general not an equality due to the way the min-+ convolution is defined in this study (more specifically, due to the subscript of the minimum in definition 1 as being '$k \leqslant n$').

We can now prove theorem 3 whose proof follows next.

Proof of Theorem 3:

Following up from the explanations given in the body of theorem 3, let L_i be a loss operator in $\mathcal{L}_i\big((1-\alpha_i)R_i(\infty)\big)$, for i equals to both 1 and 2, which represent the losses that would happen through network element i, in light of lemma 8.

First, note the following inequality holds

$$\lceil \alpha_2 \lceil \alpha_1 R_1(\infty) \rceil \rceil \geqslant \lceil \alpha_1 \alpha_2 R_1(\infty) \rceil$$

for any $\alpha_i \leqslant 1$ (for i equals to both 1 and 2) and $R_1(\infty)$. The quantity $\lceil \alpha_1 R_1(\infty) \rceil$ represents the least number of packets of R_1 that would meet their deadlines through network element 1. Whereby, the quantity $\lceil \alpha_2 \lceil \alpha_1 R_1(\infty) \rceil \rceil$ also represents the least number of packets of R_2 that would meet their deadlines through network element 2, and hence that of R_1 through the tandem of network elements. Also note that the difference between the two sides of the above inequality could at most be 1, and the inequality is tight (i.e., there exists values α_1 and $R_1(\infty)$ that the inequality becomes an equality).

Second, notice that any composite loss operator $\mathrm{L}_2\mathrm{L}_1$ corresponds to a loss operator L in $\mathcal{L}\left((1-\alpha_1\alpha_2)R_1(\infty)\right)$. This could be noted by the remarks in the preceding paragraph, or could also be noted as follows: Let $\mathbf{a}_i$ be the corresponding $\mathbf{a}$ for the loss operator L_i, for i equals to both 1 and 2. Then, we have

$$\begin{aligned} |\mathbf{a}_1| + |\mathbf{a}_2| &\leqslant (1-\alpha_1)R_1(\infty) + (1-\alpha_2)\,\alpha_1 R_1(\infty) \\ &= (1-\alpha_1\alpha_2)R_1(\infty)\,. \end{aligned}$$

Similarly, for any loss operator L in $\mathcal{L}\left((1-\alpha_1\alpha_2)R_1(\infty)\right)$, there exist two loss operators L_1 and L_2 (as noted at the beginning of the proof) to represent the packet losses that would happen through network element 1 and 2, respectively.

Now, it follows in light of lemma 8 and the above notes that we have

$$\begin{aligned} R_3(n) &\leqslant \mathrm{L}_2 R_2(n) && \text{by lemma 8} \\ &\leqslant \mathrm{L}_2(\mathrm{L}_1 R_1(n)) && \text{by lemmas 8 and 10} \\ &= \mathrm{L}R_1(n) && \text{by the remarks in the preceding paragraph.} \end{aligned} \qquad (*1)$$

Similarly, we also have

$$\begin{aligned} R_3(n) &\geqslant \mathrm{L}_2(R_2 \triangledown S_2)(n) && \text{by lemma 8} \\ &\geqslant \mathrm{L}_2(\mathrm{L}_1(R_1 \triangledown S_1) \triangledown S_2)(n) && \text{by lemmas 8, 12, and 10} \\ &\geqslant \mathrm{L}_2\big(\mathrm{L}_1((R_1 \triangledown S_1) \triangledown S_2)\big)(n) && \text{by lemmas 11 and 10} \\ &= \mathrm{L}_2\big(\mathrm{L}_1(R_1 \triangledown (S_1 \triangledown S_2))\big)(n) && \text{by lemma 13} \\ &= \mathrm{L}(R_1 \triangledown (S_1 \triangledown S_2))(n) && \text{by the remarks given in the third paragraph.} \end{aligned} \qquad (*2)$$

Notice that the relation in the line above where we have invoked lemma 13 is an equality due to the fact that any service curve takes on the value 0 for non-positive values of its argument by definition. This could be followed more clearly in the proof of lemma 13 where changes according to this note would take effect.

Now, combining the lines tagged $(*1)$ and $(*2)$, and also working the derivations for each one of them backwards, we get

$$\mathrm{L}R_1(n) \geqslant R_3(n) \geqslant \mathrm{L}(R_1 \triangledown (S_1 \triangledown S_2)) \quad \text{for all n, and for any L in } \mathcal{L}\left((1-\alpha_1\alpha_2)R_1(\infty)\right). \qquad (11)$$

Finally, in light of lemma 8, if we were to assign deadlines to the packets of $\mathrm{L}R_1(n)$ via $\mathrm{L}(R_1 \triangledown (S_1 \triangledown S_2))$, i.e., for any $k \in [1, \mathrm{L}R_1(\infty)]$ the deadline $\mathcal{D}_k$ of packet k is given by

$$\mathcal{D}_k = \min\{t : \mathrm{L}(R_1 \triangledown (S_1 \triangledown S_2))(t) \geqslant k\},$$

then the equation 11 implies that the service curve $S_1 \triangledown S_2$ with loss parameter $(1-\alpha_1\alpha_2)$ has been delivered to flow R_1 by the tandem of network elements, which completes the proof. ■

By a repeated application of theorem 3, we would also obtain the composability of the new service model for any number of network elements in tandem.

4 Remarks About The New Service Model and A Scheduling Algorithm

In this section, we give some remarks about the new service model, and present a scheduling algorithm in section 4.1, to deliver the services as specified by the new service model, at a multiplexer. Both for the remarks and the algorithm, the insight is provided by theorem 2 which we heavily utilize in this section.

First follows the remarks.

Note that for $\alpha_i = 1$, for all $i \in I$, (i.e., when there is no loss), the condition in theorem 2 gives the sufficient condition for the delivery of service requests according to the original service curve definition. This is again a sufficient condition (for the delivery of services according to the original service curve definition), only because one would adopt assigning deadlines to the incoming packets in getting this result, as it could noted in light of lemma 2. Results to this end are already available in the literature, but for specific scheduling algorithms (e.g., SCED (*Service-curve Based Earliest-Deadline-First*) [10]) and mostly for constant capacity rate servers. In this sense, the result obtained here for this special case (i.e., , $\alpha_i = 1$ for all $i \in I$) is more general.

As a second remark, it might be thought that the service provided by the new service model (given by definition 5) is the same as delivering a service curve S_i to a trimmed flow R'_i for each flow i, where $R'_i(n)$ is equal to $\lceil \alpha_i R_i(n) \rceil$ for all n. However, as it could be seen from theorem 2 and the remarks given in the previous paragraph that delivering what is being suggested to flows at a network element places a more stringent condition on the capacity of the network element, than what is really needed. This is noted more precisely in the following two paragraphs.

First of all, let us note that any trimmed flow R'_i conforms to arrival curve $\lceil \alpha_i\, A_i(n) \rceil$, since the untrimmed flow R_i conforms to arrival curve A_i, which is shown below; it holds for all k and $n \geqslant 0$ that

$$\begin{aligned} R'_i(n+k) - R'_i(k) &= \lceil \alpha_i R_i(n+k) \rceil - \lceil \alpha_i R_i(k) \rceil \\ &\leqslant \lceil \alpha_i R_i(n+k) - \alpha_i R_i(k) \rceil \qquad \text{since, } \lceil a-b \rceil \geqslant \lceil a \rceil - \lceil b \rceil \text{ for any } a \text{ and } b^5 \\ &= \lceil \alpha_i (R_i(n+k) - R_i(k)) \rceil \\ &\leqslant \lceil \alpha_i\, A_i(n) \rceil \qquad \text{since } R_i \sim A_i\,. \end{aligned}$$

Note that the above inequality is tight over all $R_i \sim A_i$, i.e., there exist values $R_i(n+k)$ and $R_i(k)$ that the inequality becomes an equality. Hence, it fits if we denote the arrival curve of the trimmed flow R'_i by $A'_i(n)$ which is defined to be equal to $\lceil \alpha_i\, A_i(n) \rceil$ for all n.

Thus, it suffices if the following condition

$$\sum_{i \in I} (A'_i \triangledown S_i)(n) \leqslant C(n+k) - C(k) \qquad \text{for all } n \geqslant 0, \text{ and } k$$

holds on the capacity of the network element, for the delivery of service curve S_i to R'_i, as it could be seen by the first remark given in the third paragraph in this section. However, it is not difficult to note that

$$\lceil \alpha_i\, (A_i \triangledown S_i)(n) \rceil \leqslant (A'_i \triangledown S_i)(n) \quad \text{for all } n,$$

which concludes the remark that the suggested service actually places a more stringent condition on the capacity of the network element as pointed out earlier, and is not the same as the service provided by definition 5. The difference between the two sides in the above inequality actually

[5] Refer to Appendix A for a simple derivation, if need be.

translates more dramatically on the capacity of the network element, since the condition on the capacity of the network element includes a sum over all the flows in I, and holds for all time shifts k.

Secondly, note that the suggested service model in the fourth paragraph does not necessarily preserve the deadlines of the remaining packets in the trimmed flow R'_i, where deadlines are assigned via the service curve S_i, for all $i \in I$. This could clearly be seen by the following inequality

$$(R_i \triangledown S_i)(n) \geqslant (R'_i \triangledown S_i)(n)$$

which holds for all n, by the monotonicity property of the min-+ convolution shown by lemma 12. The new deadlines assigned to the packets of the trimmed flow R'_i are, in fact, in general larger than that of the untrimmed flow R_i. Hence, this note further shows that the suggested service model in the fourth paragraph is not actually the same as the service that we have proposed by definition 5.

One could in fact make a better suggestion than what is being suggested by the second remark in the fourth paragraph. Namely, instead of delivering S_i to the trimmed flow R'_i, one could think of delivering a trimmed service curve S'_i the the trimmed flow R'_i, where $S'_i(n)$ is defined to be equal to $\lceil \alpha_i \, S_i(n) \rceil$ for all n, and for all $i \in I$.

However again, as it could be seen from theorem 2 and the first remark given in the third paragraph that delivering this suggested service to flows at a network element also places a more stringent condition on the capacity of the network element, than what is really needed.

This again could be noted more precisely as follows. We have shown earlier that each trimmed flow R'_i conforms to arrival curve A'_i, where A'_i is as defined before (i.e., $A'_i(n) = \lceil \alpha_i \, A_i(n) \rceil$ for all n, and for all $i \in I$). Hence, as it could be seen from the first remark that the following condition

$$\sum_{i \in I} (A'_i \triangledown S'_i)(n) \leqslant C(n+k) - C(k) \qquad \text{for all } n \geqslant 0, \text{ and } k$$

suffices to hold on the capacity of the network element for the delivery of service curve S'_i to R'_i. Again, however, it is easy to note that

$$\lceil \alpha_i \, (A_i \triangledown S_i)(n) \rceil \leqslant (A'_i \triangledown S'_i)(n) \quad \text{for all } n,$$

which concludes that this suggested service also places a more stringent condition on the capacity of the network element as pointed earlier, and is also not the same as the service that we have proposed by definition 5. It is not difficult to note that the difference between the two sides in the above inequality is at most 1, as could be followed in Appendix B. Thus, the condition placed by this service model on the capacity of the network element is larger than what is really needed by at most $|I|$, in any interval $(k, n+k]$ of time.

Similarly note that this suggested service model does not also necessarily preserve the deadlines of the remaining packets in the trimmed flow R'_i, where deadlines are assigned via the service curve S'_i, for all $i \in I$. This could be seen by the following inequality

$$(R_i \triangledown S_i)(n) \geqslant (R'_i \triangledown S'_i)(n)$$

which holds for all n (which again follows by the monotonicity property of the min-+ convolution shown by lemma 12). The new deadlines assigned to the packets of the trimmed flow R'_i are also in general larger than that of the untrimmed flow R_i. Hence, this note further shows that the second suggested service model as well is not the same as the service that we have proposed by definition 5.

Ironically, although both of the suggested models assign deadlines to packets, which are in general larger than that of the proposed model by definition 5, the condition placed by them on the capacity of the network element is more stringent than the condition we have in theorem 2.

4.1 A Scheduling Algorithm to Deliver Services According to The New Model

In this section, we propose a scheduling algorithm to deliver the services as specified by the new service model provided by definition 5, at a multiplexer. We show that the proposed algorithm delivers the services as specified by the new model, by utilizing theorem 2.

We call the algorithm that we propose as L-SCED (*Lossy* SCED, or spelled out more specifically as *Lossy Service-curve-Based-Earliest-Deadline-First*), whose pseudo-algorithm is given below.

Algorithm L-SCED

Input: A set $I = \{1, 2, \ldots, |I|\}$, where $|I| > 0$, indexing flows.
A flow R_i and an arrival curve A_i for each $i \in I$, where R_i conforms to A_i.
A service curve S_i and a loss parameter $1 - \alpha_i$, for each $i \in I$.
A capacity rate $c(n)$ of a network element onto which flows indexed by I are incident.

Output: A scheduling algorithm to deliver service curve S_i with loss parameter $1 - \alpha_i$ to flow R_i, for all $i \in I$, where flows are incident onto a network element with capacity rate $c(n)$.

Body:

1. For each $i \in I$, do
   (a) mark all the packets arriving at time n in flow R_i arbitrarily, but distinctly, by the integers in $(R(n-1), R(n)]$,
   (b) compute the deadline $D_{i,k}$ of packet k in flow R_i as

$$D_{i,k} = \min\{t : (R_i \triangledown S_i)(t) \geqslant k\}$$

   for all k in $[1, R_i(\infty)]$, and assign deadline $D_{i,k}$ to packet k,
   (c) eliminate $\lfloor (1-\alpha_i) R_i(n) \rfloor - \lfloor (1-\alpha_i) R_i(n-1) \rfloor$ many packets at any time n, arbitrarily from flow R_i.
2. At any time n, schedule as many packets as possible, not exceeding $c(n)$, with earliest (i.e., smallest $D_{i,k}$) deadlines among all the packets (i.e., for all i and k) which remain in the queue, for transmission at time n.

Before we move on to prove that L-SCED does indeed deliver the services as specified in its Output, if the capacity of the network element satisfies a certain condition, we would like to give the following remarks about the algorithm first:

- The deadlines of the packets are assigned in accordance with the service model provided by definition 5, and they do not change after the elimination done in step 1-(c) in the Body of the algorithm.
- Exactly $\lfloor (1-\alpha_i) R_i(n) \rfloor$ many packets are eliminated (dropped) by any time n (inclusive), which could easily be seen by the telescoping sum given below;

$$\begin{aligned}\text{total \# of eliminations by time } n &= \sum_{k \leqslant n} \Big(\lfloor (1-\alpha_i) R_i(k) \rfloor - \lfloor (1-\alpha_i) R_i(k-1) \rfloor \Big) \\ &= \lfloor (1-\alpha_i) R_i(n) \rfloor,\end{aligned}$$

  hence, exactly $\lceil \alpha_i R(\infty) \rceil$ many packets are scheduled for transmission in step 2 in the Body of the algorithm.

- Thus, if all the packets in step 2 in the Body of the algorithm meet their deadlines (i.e., served by no later than their deadlines), then service curve S_i with loss parameter $1-\alpha_i$ is delivered to flow R_i, for all $i \in I$.

Theorem 4 *The algorithm* L-SCED *delivers the services as specified in its* Output, *for its any given* Input, *if and only if the capacity of a network element satisfies the condition given in theorem 2, namely*

$$\sum_{i\in I} \lceil \alpha_i \, (A_i \triangledown S_i)(n) \rceil \leqslant C(n+k) - C(k) \qquad \text{for all } n \geqslant 0, \text{ and } k.$$

Proof: The proof follows in two steps.

Step 1: For any given Input of L-SCED, compute the deadlines of each packet as specified in the Body of the algorithm, and trim each flow R_i by eliminating packets also as specified in the Body of the algorithm. Denote the trimmed flow for R_i by R_i'. Note by the remarks given right before theorem 4 that

$$R_i'(n) = \lceil \alpha_i \, R_i(n) \rceil \quad \text{for all } n.$$

Construct a labeled graph as we have described before in section 3.2, but this time for the trimmed flows. For any such labeled graph, a complete matching where all the nodes in each V_i is matched, exists *if and only if* the following condition holds;

$$\sum_{i\in I} |\Pi_i(B)| \leqslant |B| \qquad \text{for all } B \subseteq W,\ B \text{ is a block.} \tag{12}$$

This could be seen easily by lemma 5 where we would set all α_i's in the lemma to 1.

Next, we show the following claim.

Claim: For any block B in condition (12), whose nodes' labels are all greater than or equal to $k+1$ and less than or equal to $k+n$, for some k and $n \geqslant 1$, we have for any $i \in I$

$$\max_{R_i'} \{ \, |\Pi_i(B)| \, \} = \lceil \alpha_i \, (A_i \triangledown S_i)(n) \rceil \, .$$

Proof of Claim: The proof of claim is similar to that of lemma 6. First of all, again note that for any $i \in I$, the flow $R_i(u) = A_i(u-k)$ also conforms to arrival curve A_i, since surely shifting does not change the burstiness of a flow. Consider the trimmed flow R_i' for this particular R_i. Note that the maximum in claim is greater than or equal to $|\Pi_i(B)|$ for this particular trimmed flow R_i'. Notice by lemma 7 that for this trimmed flow, $\Pi_i(B)$ actually includes the first $\lceil \alpha_i \, (R_i \triangledown S_i)(n+k) \rceil$ many packets—which is ensured by the capacity arrival at time $n+k+1$. This could be seen more clearly if we consider *the* specific loss operator L such that

$$\mathrm{L}R_i(n) = R_i'(n) \quad \text{for all } n.$$

In other words, this specific loss operator L is such that it eliminates as many packets as specified in step 1-(c) in the Body of L-SCED, at any time n from flow R_i. Since L will also act the same way on $R_i \triangledown S_i$ in getting the unchanged deadlines of the trimmed flow R_i', the principals of B happen to be as we have specified earlier in this paragraph. Thus, we have

$$\begin{aligned} \max_{R_i'} \{ \, |\Pi_i(B)| \, \} &\geqslant |\Pi_i(B)| \\ &= \lceil \alpha_i \, (R_i \triangledown S_i)(n+k) \rceil \\ &= \lceil \alpha_i \, (A_i \triangledown S_i)(n) \rceil \end{aligned} \tag{$*$}$$

where the last equality follows from the corresponding derivation in the proof of lemma 6.

For the other direction, note that for any trimmed flow R'_i, we have

$$|\Pi_i(B)| = \max\left\{0, \lceil \alpha_i\,(R_i \triangledown S_i)(n+k)\rceil - \lceil \alpha_i\, R_i(k)\rceil\right\}$$

since again at most the first $\lceil \alpha_i\,(R_i \triangledown S_i)(n+k)\rceil$ many packets could be among the principals of B, while the first $\lceil \alpha_i\, R_i(k)\rceil$ many packets could not, as they have arrived before time $k+1$ hence their image have nodes not in B—these are again ensured by the capacity arrivals at time k and $n+k+1$. Thus, we have

$$\begin{aligned} |\Pi_i(B)| &= \max\left\{0, \lceil \alpha_i\,(R_i \triangledown S_i)(n+k)\rceil - \lceil \alpha_i\, R_i(k)\rceil\right\} \\ &\leqslant \max\left\{0, \left\lceil \alpha_i\,(R_i \triangledown S_i)(n+k) - \alpha_i\, R_i(k)\right\rceil\right\} \qquad \text{since, } \lceil a-b\rceil \geqslant \lceil a\rceil - \lceil b\rceil \text{ for any } a \text{ and } b \\ &= \max\left\{0, \left\lceil \alpha_i\left((R_i \triangledown S_i)(n+k) - R_i(k)\right)\right\rceil\right\} \\ &\leqslant \lceil \alpha_i\,(A_i \triangledown S_i)(n)\rceil \end{aligned}$$

where the last inequality again follows from the corresponding derivation in the proof of lemma 6.

Since $|\Pi_i(B)|$ is less than or equal to $\lceil \alpha_i\,(A_i \triangledown S_i)(n)\rceil$ which is independent from the arrival pattern of R_i (and hence that of R'_i) other than the fact that it conforms to A_i, so is its maximum over all R'_i. Together with the first inequality tagged $(*)$ above, this completes the proof of the claim.

Thus, continuing from condition (12), we see that the following condition

$$\sum_{i\in I} \lceil \alpha_i\,(A_i \triangledown S_i)(n)\rceil \leqslant C(n+k) - C(k) \qquad \text{for all } n \geqslant 0, \text{ and } k,$$

is necessary and sufficient for the existence of a complete matching for all the nodes in M, with the loss pattern as specified in L-SCED. Note that this is the same condition as specified in the body of theorem 4 (and, hence theorem 2).

Step 2: For any given complete matching μ where all the nodes in M are matched in any labeled graph constructed in Step 1, we can do the following. Let two nodes in M with labels m_1 and m_2, be matched to nodes in W with labels w_1 and w_2, respectively. Suppose $w_2 < w_1$, while the deadline D_2 of node (packet) m_2 is larger than the deadline D_1 of node (packet) m_1. In other words, suppose the packet corresponding to node m_2 is scheduled for transmission earlier than the packet corresponding to node m_1 (that is, $w_2 < w_1$), while the deadline of packet m_2 is larger than that of packet m_1 (that is, $D_2 > D_1$). Notice that in this case we could always come up with another matching μ' by only switching the nodes to which the nodes with labels m_1 and m_2 are matched; that is,

$$\mu' = \Big(\mu - \big\{\{m_1, w_1\}, \{m_2, w_2\}\big\}\Big) \cup \big\{\{m_1, w_2\}, \{m_2, w_1\}\big\}.$$

Also note that, by μ', any the node in M is still matched to a node in W within its image, since μ is a matching, and since

$$\begin{aligned} &\text{for node } m_1, \quad w_2 < w_1 \leqslant D_1 \\ &\text{for node } m_2, \quad w_1 \leqslant D_1 < D_2\,. \end{aligned}$$

Continuing in this fashion, we would come up with a complete matching μ' where all the nodes in M are matched, for any given μ that, stating in terms of packets, no packet is scheduled for transmission earlier than another packet with a smaller deadline. Notice that μ' corresponds to a scheduling as specified in step 2 in the Body of L-SCED. Together with Step 1, this completes the proof of theorem 4. ■

Finally, we would like give two more remarks about L-SCED:

1. A more involved coordination among packet transmissions turns out to be not needed to deliver the services as specified in the Output of L-SCED, since the capacity of a network element is lower-bounded very conservatively by the condition in theorem 4 (and hence, theorem 2).

2. L-SCED drops exactly $\lfloor(1-\alpha_i R_i(\infty)\rfloor$ many packets from each flow i ($i \in I$), and does so somewhat uniformly.

5 Conclusion

The original service curve model introduced in [1, 2] does not consider packet losses. In this study, we have proposed a new service model based on service curves, which has a loss aspect. In the new model, packets of a flow are assigned deadlines via a service curve as in the original definition, however, only at least a certain fraction of all the packets are required to be transmitted within their deadlines.

We have shown that the proposed is *composable*. The composability of a service model is an important property to have, since it facilitates a systematic treatment of performance guarantees in communication networks. A service model is said to be *composable* if the service delivered by a tandem of two network elements could also be represented in terms of the services delivered by each of the network elements, where all services are described according to a service model at hand.

We have also solved a multiplexing problem for the delivery of services according to the new model. Specifically, we have found a necessary and sufficient condition on the capacity of a network element for the delivery of services according to the new model to a set of flows. This condition could be employed for an efficient *connection admission control* at a multiplexer delivering services according to the new model. We have taken a graph-theoretic approach to find this condition.

With the original service curve model, we have indicated a similar condition, where all α_i's are set to 1, to employ for connection admission control at a multiplexer. If this condition is not satisfied for a set of flows at a multiplexer, then either some of the flows or all the flows in the set are to be turned down for service, in order for the rest of flows in the set to receive service according to the original model. However, with the new service model, a renegotiation could instead be performed by offering a loss parameter $1-\alpha_i$ to each flow i in the set, so that the necessary and sufficient condition in theorem 2 could be satisfied for the delivery of services according to the new model.

Also note that besides the flexibility provided by a such renegotiation mentioned above, the new service model enables multiplexing potentially more streams at a multiplexer, and hence a more efficient operation.

Finally, we have proposed a scheduling algorithm called L-SCED to deliver the services as specified by the new service model, at a multiplexer.

Future work includes carrying out this study into a probabilistic setting in order to have statistical gains. One could also consider a similar loss model where packet space of a flow would be partitioned into some set of packet blocks where the service to be delivered for each set of block would be just as we have examined in this study, for a more controlled distribution of losses over the entire flow. One could examine such a model in almost the same way as we have presented in this study. However, one might want to propose a such model only if the distribution of packet losses provided by L-SCED is not sufficient, since losses in L-SCED happen to be so somewhat uniformly.

A Two Simple Inequalities For Ceiling Function

First Inequality

We first show that

$$\lceil a+b \rceil \leqslant \lceil a \rceil + \lceil b \rceil \quad \text{for any real } a \text{ and } b. \tag{13}$$

Let us represent each number a and b as the sum of their integer and fractional parts, i.e.,

$$a = \lfloor a \rfloor + \{a\} \qquad \text{and} \qquad b = \lfloor b \rfloor + \{b\}$$

where the notation $\{x\}$ denotes the fractional part of x, i.e., $\{x\} = x - \lfloor x \rfloor$.[6]

Note that for any real number x and an integer n, the following equality clearly holds by the definition of $\lceil \cdot \rceil$

$$\lceil n+x \rceil = n + \lceil x \rceil \tag{14}$$

which could easily be seen as follows;

$$\begin{gathered} \lceil x \rceil - 1 < x \leqslant \lceil x \rceil \\ n + \lceil x \rceil - 1 < n + x \leqslant n + \lceil x \rceil \,. \end{gathered}$$

By (14), also note that

$$\lceil x \rceil = \lfloor x \rfloor + \lceil \{x\} \rceil \,.$$

Thus, for the left-hand-side of the claimed inequality, we have

$$\lceil a+b \rceil = \lfloor a \rfloor + \lfloor b \rfloor + \lceil \{a\} + \{b\} \rceil \,,$$

and for the right-hand-side, we have

$$\lceil a \rceil + \lceil b \rceil = \lfloor a \rfloor + \lfloor b \rfloor + \lceil \{a\} \rceil + \lceil \{b\} \rceil \,.$$

For $\{a\} > 0$ and $\{b\} > 0$, we have

$$\lceil \{a\} \rceil + \lceil \{b\} \rceil = 2 \,,$$

whereas

$$\lceil \{a\} + \{b\} \rceil \leqslant 2$$

since the fractional part of any number is strictly less than 1.

When either one of the fractional parts is equal to zero, the claimed inequality clearly becomes an equality, and holds as it could be seen by (14).

Finally, note by the above proof that the difference between the two sides of inequality (13) is at most 1; i.e.,

$$\lceil a \rceil + \lceil b \rceil - \lceil a+b \rceil \leqslant 1 \quad \text{for any real } a \text{ and } b.$$

[6]The notation $\{x\}$ is adopted from [11].

Second Inequality

Second, we show that

$$\lceil a-b \rceil \geqslant \lceil a \rceil - \lceil b \rceil \quad \text{for any real } a \text{ and } b.$$

One quick proof follows directly from the first inequality (13) that we have shown before in the appendix, as follows;

$$\begin{aligned} \lceil a \rceil &= \lceil b + (a-b) \rceil \\ &\leqslant \lceil b \rceil + \lceil a-b \rceil \quad \text{by the first inequality (13)} \end{aligned}$$

hence,

$$\lceil a \rceil - \lceil b \rceil \leqslant \lceil a-b \rceil .$$

A direct proof without utilizing the inequality (13) could also be given, which follows next. Again, let us represent each number a and b as the sum of their integer and fractional parts, as we have done before in showing the first inequality (13). Note again by equality (14) that the left-hand-side of the claimed inequality could be represented as

$$\lceil a-b \rceil = \lfloor a \rfloor - \lfloor b \rfloor + \lceil \{a\} - \{b\} \rceil ,$$

and the right-hand-side could be represented as

$$\lceil a \rceil - \lceil b \rceil = \lfloor a \rfloor - \lfloor b \rfloor + \lceil \{a\} \rceil - \lceil \{b\} \rceil .$$

For $\{a\} \leqslant \{b\}$, we have

$$\lceil \{a\} - \{b\} \rceil = 0$$

since the fractional part of any number is strictly less than 1, and hence

$$-1 < \{a\} - \{b\} \leqslant 0.$$

Whereas, for the right-hand-side we have

$$\lceil \{a\} \rceil - \lceil \{b\} \rceil \leqslant 0$$

since,

$$\begin{aligned} \{a\} &\leqslant \{b\} \\ \lceil \{a\} \rceil &\leqslant \lceil \{b\} \rceil \\ \lceil \{a\} \rceil - \lceil \{b\} \rceil &\leqslant 0 . \end{aligned}$$

For $\{a\} > \{b\}$, we have

$$\lceil \{a\} - \{b\} \rceil = 1$$

again since the fractional part of any number is strictly less than 1, and hence

$$0 < \{a\} - \{b\} < 1.$$

Whereas, for the right-hand-side we have

$$\lceil\{a\}\rceil - \lceil\{b\}\rceil \leqslant 1$$

since,

$$\begin{aligned} \{b\} &< \{a\} \\ \lceil\{b\}\rceil &\leqslant \lceil\{a\}\rceil \\ 0 \leqslant \lceil\{a\}\rceil &- \lceil\{b\}\rceil \leqslant 1 \end{aligned}$$

again by the fact that the fractional part of any number is strictly less than 1.

Finally, again note by the proof that the difference between the two sides of the second inequality is also at most 1; i.e.,

$$\lceil a-b\rceil - \lceil a\rceil + \lceil b\rceil \leqslant 1 \quad \text{for any real } a \text{ and } b.$$

B An Inequality Involving Minimum and Ceiling

Let f and g be any two functions, and let a be any non-negative real number. We also define the following functions

$$\begin{aligned} f'(n) &\triangleq \lceil a \cdot f(n)\rceil \quad \text{for all } n, \\ g'(n) &\triangleq \lceil a \cdot g(n)\rceil \quad \text{for all } n. \end{aligned}$$

We first show that

$$\lceil a\,(f \triangledown g)(n)\rceil \leqslant (f' \triangledown g')(n) \quad \text{for all } n \tag{15}$$

whose proof follows next; it holds for all n that

$$\begin{aligned} \lceil a\,(f \triangledown g)(n)\rceil &= \left\lceil a \cdot \min_{k\leqslant n}\{f(k)+g(n-k)\}\right\rceil \\ &= \min_{k\leqslant n}\left\{\left\lceil a\cdot(f(k)+g(n-k))\right\rceil\right\} \\ &= \min_{k\leqslant n}\left\{\left\lceil a\cdot f(k)+a\cdot g(n-k)\right\rceil\right\} \\ &\leqslant \min_{k\leqslant n}\left\{\lceil a\cdot f(k)\rceil + \lceil a\cdot g(n-k)\rceil\right\} \quad \text{by inequality (13)} \\ &= (f' \triangledown g')(n)\,. \end{aligned}$$

Secondly, we show that the difference between the two side of inequality (15) is at most 1: The minimum in $(f \triangledown g)(n)$ is realized for a k which is less than or equal to n, i.e.,

$$\begin{aligned} (f \triangledown g)(n) &= \min_{u\leqslant n}\{f(u)+g(n-u)\} \\ &= f(k)+g(n-k) \quad \text{for some } k \leqslant n. \end{aligned}$$

Hence,

$$\begin{aligned} \lceil a\,(f \triangledown g)(n)\rceil &= \left\lceil a\cdot(f(k)+g(n-k))\right\rceil \\ &= \left\lceil a\cdot f(k)+a\cdot g(n-k)\right\rceil\,. \end{aligned} \tag{16}$$

Now, since the set over which the minimum is taken in $(f' \triangledown g')(n)$ includes the term

$$\lceil a \cdot f(k) \rceil + \lceil a \cdot g(n-k) \rceil$$

whose difference with the quantity in (16) is at most 1 as it could be seen by the note at the end of the proof of inequality (13), the difference of the minimum $(f' \triangledown g')(n)$ from the quantity in (16) is also at most 1.

References

[1] A. K. Parekh, R. G. Gallagher. *A generalized processor sharing approach to flow control in integrated services networks: the single-node case*, IEEE/ACM Transaction on Networking, vol. 1, pp. 344–357, 1993.

[2] R. L. Cruz. *Quality of Service Guarantees in Virtual Circuit Switched Networks*, IEEE Journal of Selected Areas in Communication, 13(6): 1048–1056, 1995.

[3] D. Ferrari, D. Verma. *A Scheme for Real-Time Channel Establishment in Wide-Area Networks*, IEEE Journal on Selected Areas in Communications, vol. 8, pp. 368–379, April 1990.

[4] H. Zhang, D. Ferrari. *Rate-Controlled Static Priority Queueing*, Proc. IEEE INFOCOM, San Francisco, CA, September 1993.

[5] Z. Wang, J. Crowcroft. *Analysis of Burstiness and Jitter in Real-Time Communications*, The Proceedings of SIGCOMM, pp. 13–19, 1993.

[6] J. Kurose, K. Ross. *Computer Networking: A Top-Down Approach Featuring The Internet*, Addison-Wesley Longman, 2nd Edition, 2003.

[7] S. Ayyorgun, W.-C. Feng. *A Probabilistic Definition of Burstiness Characterization: A Systematic Approach*, Technical Report, Los Alamos National Laboratory, LA-UR-03-3668.

[8] S. Ayyorgun, R. L. Cruz, *A Bigraph Matching Theorem.* Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing, pp. 124–126, Sept. 22–24 1999.

[9] S. Ayyorgun. *Feasibility of Serving Packet Streams With Delay and Loss Requirements.* Ph.D. Dissertation, Department of Electrical and Computer Engineering, University of California, San Diego, 2001.

[10] H. Sariowan. *A Service-curve Approach to Performance Guarantees in Integrated-service Networks.* Ph.D. Dissertation, Department of Electrical and Computer Engineering, University of California, San Diego, 1996.

[11] R. L. Graham, D. E. Knuth, O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*, Addison-Wesley, 2nd ed., 1994.