Fair Allocation of Utilities in Multirate Multicast Networks

Saswati Sarkar[†]
Department of Electrical Engineering
University of Pennsylvania
email: swati@ee.upenn.edu
and
Leandros Tassiulas
Dept. of Electrical and Computer Engineering and Institute for Systems Research
University of Maryland, College Park, MD, USA
email: leandros@isr.umd.edu

Abstract

We study fairness in a multicast network. We assume that different receivers of the same session can receive information at different rates. We study fair allocation of utilities, where utility of a bandwidth is an arbitrary function of the bandwidth. The utility function is not strictly increasing, nor continuous in general. We discuss fairness issues in this general context. Fair allocation of utilities can be modelled as a nonlinear optimization problem. However, nonlinear optimization techniques do not terminate in finite number of iterations in general. We present an algorithm for computing a fair utility allocation. Using specific fairness properties, we show that this algorithm attains global convergence and yields a fair allocation in polynomial number of iterations.

1 Introduction

We will study resource allocation and congestion control in real time multicast networks in this paper. Multicast is communication between a single source and a group of destinations. The source sends one copy of a message and this copy is replicated only at the branching points of the multicast route. More traditional forms of communication like unicast with a single source and a single destination, and broadcast with a single source communicating to all other nodes in the network are special cases of multicast. Real time multicast applications are increasing all the time. Examples are collaborative applications like audio or video teleconferencing, video-on-demand services, distance learning etc. So far, multicasting has been used primarily in the Internet, but future ATM networks are likely to deploy multicasting in a large scale, particularly in applications like broadcast video, video-conferencing, multiparty telephony and workgroup applications[4]. Resource allocation in multicast networks is more complex than that in unicast and broadcast networks. This calls for a separate study of these problems in the multicast scenario.

[*]Parts of this paper have been presented at Allerton' 1999
[†]Corresponding Author.

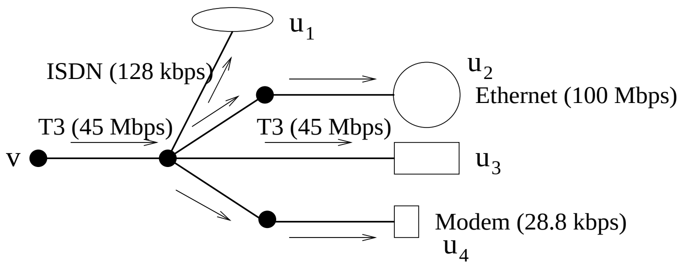

Figure 1: A sample network indicating network heterogeneity

We concentrate on real time applications like audio and video in this paper. Many real time applications like teleconferencing, audio and video broadcasting require communication within a group, and hence multicast is the inherent mode of delivery in these applications. Congestion control is critically important for real time applications, because many of these consume significant bandwidth, and tax the network resources severely. Also, the users need to adapt to available bandwidth of the network, which fluctuates from time to time. For this purpose, real time sources use a hierarchical or a layered coding scheme. This layered transmission scheme have been used for both video[21] and audio[3] transmissions over the internet and has potentials for use in ATM networks as well[7]. In this approach, a signal is encoded into a number of layers that can be incrementally combined to provide progressive refinement. The receivers adapt to congestion by adding and dropping layers. As each layer is added, there is an improvement of quality of the received signal and additional bandwidth is needed to transport the combined stream. When a layer is dropped, there is graceful degradation in reception quality. Every receiver would like to receive as many layers as possible. However, networks have resource limitations, and delivering a large number of layers would cause acute congestion. Resource allocation strategies become important in this situation.

Layered encoding can be used to counter network heterogeneity. A multicast application may have many destinations, and end systems can have widely varying bandwidth connectivities. End systems range from fast ethernets (100 Mbps) to slow modems (28.8 kbps). The paths to different destinations may have different bandwidth capacities, e.g., one may consist of multi-megabit links, such as, $T3$ (45 Mbps) and another may have a 128 kbps ISDN line (Refer to the network shown in figure 1 for an example). A single rate of transmission per session is likely to either overwhelm the slow receivers or starve the fast ones, in absence of additional provisions. This is not an issue in unicast networks because there is only one receiver per application.

Layered encoding allows multirate transmission. The receivers of the same session are allowed to receive information at different service rates. The source transmits at a rate matching the fastest of its receivers. At every link, the transmission rate of a session is equal to that of the fastest session receiver downstream of the link. So the transmission rate of the same session may be different in different links depending on the configuration of the receivers downstream. Thus, unlike unicast networks, there is no concept of session rates as such. One needs to consider receiver

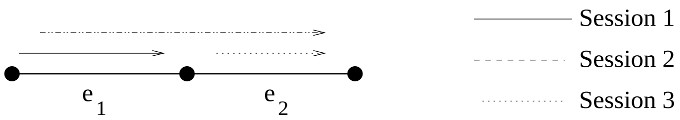

Figure 2: Session 1 traverses both links, and sessions 2 and 3 traverse one link each. Both links have one unit of bandwidth each.

rates separately.

As we mentioned before, it may not be possible to accommodate the service demands of the receivers without overloading the network resources. Resource sharing objectives will then determine the individual allocations. Fair allocation of service rates has several advantages. Networks often provide class based service. The users belonging to the same class have the same service agreements and pay the same revenues. So, it is expected that the network would treat the members of the same class similarly, i.e., the network should be fair within the same class. Besides, fair allocation of service does not penalize any user excessively, and naturally allocates a certain minimum quality of service to all users.

There are several possible notions of fairness. A simplistic notion is to serve all users at the same rate. However, some users may receive information through poor quality links, while some others may receive information through high quality links. Serving all users at the same rate would tie down all users to the low quality service in this case. So this notion of fairness is not useful.We adopt the notion of maxmin fairness presented in [2]. Maxmin fairness is a good notion of fairness and as [14] points out, maxmin fairness satisfies many intuitive fairness properties in a multirate multicast network. A bandwidth allocation is maxmin fair if it is not possible to respect the resource constraints and increase the bandwidth of a receiver without decreasing that of any other receiver which has equal or lower rate. Recently, [8] has advocated a new notion of fairness, namely proportional fairness. A feasible rate vector $\vec{r}^1$ is proportionally fair, if for any other feasible rate vector $\vec{r}^2$, the aggregate of proportional changes is non-positive, i.e., $\sum_{i=1}^{M} \frac{r_2^i - r_1^i}{r_1^i} \leq 0$. Consider the network in figure 2 as a distinguishing example. Maxmin fair allocation allocates 0.5 units to all sessions. Proportionally fair allocation allocates 1/3 unit of bandwidth to the long session and 2/3 units of bandwidth to the short sessions each. So it is apparent that proportionally fair allocation discriminates against long sessions. The argument in favor of this discrimination is that the long sessions consume more resources than the short ones, and hence should not be treated in the same manner as the short sessions. This argument raises the question that if we need to discriminate between sessions, then how should the amount of discrimination be determined. In our opinion, this should depend on several issues like the bandwidth requirements, revenues earned etc. However, it is clear that we need to broaden our definition of fairness. We would focus on maxmin fair allocation in multirate multicast networks in this paper, but at the same time we would generalize our definition of fairness by using the notion of utilities.

As discussed above, it may not be desirable to assign equal importance to all users. This is because different users have different bandwidth requirements and fetch different revenues. Network should treat them accordingly. For example, a video source requires more bandwidth than an audio source, and these two should not be allocated equal resources. Also, certain users may be willing to pay higher revenues, and then it is fair that they get a better quality service. It is important to generalize the notions of fairness so that one can differentiate among users within the framework of fairness.The concept of utilities is useful in this context. Utility of a receiver is a function connecting the bandwidth granted to the receiver and the "value" associated with the bandwidth. The term "value" is an abstract notion. It simply indicates a measure of the satisfaction of the user or even the amount paid by the user. A possible fairness objective can be to allocate bandwidths so that the utilities are distributed fairly. We would consider fair allocation of utilities in the multirate, multicast scenario. Users need to announce their utility functions. Network would charge the users in accordance with their utilities, and at the same time allocate the resources so that the corresponding utility allocation is fair. Consider a simple one link example to clarify this concept. A user who needs a high bandwidth, and is willing to pay high revenues will declare a slowly increasing utility function, whereas a user who is satisfied with a low bandwidth, and does not want to pay high revenue would declare a rapidly increasing utility function. The fair allocation of utilities in this case is to equalize the utilities of the two users, and the former would receive a higher bandwidth than the latter at this fair operating point.

Utility fairness provides a unified framework for diverse fairness objectives. We study a concrete example to argue this point. We have studied fair allocations of discrete bandwidth layers in [18] under the assumption that hierarchical signal structure is the same for different sessions. More precisely, layers for different sessions have the same bandwidth. Under this assumption, fairness of layer allocation is equivalent to the fairness of rate allocation, and the two need not be investigated separately. This does not hold any longer when different sessions have different layer bandwidths. Consider a specific example. Consider a single link network. Two sessions traverse the link. Source of the first session transmits unit bandwidth layers. Every layer of the second session consumes 2 units of bandwidth. Let the capacity of the link be 9 units. A bandwidth allocation allocating 5 units of bandwidth to the first session and 4 units to the second one is intuitively fair (discrete bandwidth layers do not allow allocation of 4.5 units each). However, this allocation gives session 1, 5 layers and session 2, 2 layers. Intuitively, $(3, 3)$ is a "fairer" layer allocation. The corresponding bandwidth allocation, $(3, 6)$ does not seem fair. So, fair allocation of layers and bandwidths are different objectives. The fairness objective depends on the specific requirements of particular networks, and can be decided on a case by case basis. However, these different fairness objectives fall under the same framework, if we consider the fair allocation of utilities, where utility of a bandwidth can be the bandwidth itself, or the number of layers obtained from the bandwidth. The former would lead to the rate fairness problem, and the latter to the fair layer allocation problem. Likewise, utility can be any other function of bandwidth as well.

We review related work in this area. Well known network protocols for layered transmission, RLM (Receiver-driven Layered Multicast)[13] and LVMR (Layered Video Multicast with Retransmissions)[10] do not handle fairness among sessions very well, when there are multiple sessions competing for bandwidth[11]. An alternative layer allocation scheme has been proposed in [11]. There is empirical evidence that this scheme improves fairness among sessions for networks with multiple video sessions sharing only one link. But, there is no experimental or analytical evidence that the scheme works well for more complex networks, with sessions sharing several links with each other. Besides [11] does not compute the actual rates or the number of layers allocated to the receivers in an arbitrary network, under some well defined notion of fairness. An algorithm for computation of maxmin fair rates has been proposed in [17]. This paper assumes that the feasible set is continuous. However, depending on the coding scheme, for layered transmission, feasible set is often discrete. This happens when layer bandwidths are fixed and can not be modulated. Recently, [18] addressed the fair allocation of discrete bandwidth layers. But, it considers the special case in which bandwidth consumed by every layer is the same irrespective of the particular layer or the source. The algorithm presented in [18] does not yield a maximally fair layer allocation, in a more general case i.e., if different layers consume different bandwidths, or if the network can allocate any rate in a particular bandwidth range and only a discrete set of bandwidths in another range. The situation becomes significantly different in this general case. The general case is relevant in the current day networking scenario. For example, every additional layer induces an exponential increase in the total bit rate in the codec of [12]. A study of fair allocation of utilities will capture this general scenario as well, and will unify the frameworks of [17] and [18]. Recently, researchers have studied the parallel problem of utility maximization in unicast networks. Low *et. al.* has developed a distributed algorithm for computing the bandwidths which maximize the sum of the utilities of all receivers[9]. Utility maximization remains an open problem in multirate, multicast networks. We would investigate a complementary problem in this chapter, that of fair allocation of utilities in multirate, multicast networks. Fair allocation of utilities have not been investigated in the multicast case. Cao *et. al.* has presented an algorithm for maxmin fair allocation of utilities in the unicast case[5]. This is the only work in utility fairness, so far. A basic assumption for this algorithm is that utility functions are strictly increasing. We would furnish an example to show that this algorithm is ambiguous and does not provide a fair allocation when utility functions are not strictly increasing, even in the unicast case. We will also discuss that in many cases of practical interest utility functions do not increase strictly. Fairness issues differ significantly, when utility functions do not increase strictly.

An utility allocation is *maxmin fair* if it is not possible to increase the utility allocated to a receiver without decreasing that of another receiver receiving equal or lesser utility. As it turns out, when utility functions do not increase strictly, maxmin fair allocation of utilities may not exist. A lexicographically optimal utility vector is one which maximizes its minimum component in a feasible set, subject to this maximization, it maximizes the second minimum, etc*. In general, lexicographically optimal utility allocation exists even when the utility function is not strictly increasing. However, computation of lexicographically optimal utility allocation is an NP-hard problem for non-strictly increasing utility functions. Lexicographically optimal utility allocation

*Maxmin fairness and lexicographic optimality will be defined formally later.

is identical to the maxmin fair utility allocation, when the latter exists and is thus polynomial complexity computable for strictly increasing utility functions, though. We need to revert to a weaker notion of fairness, maximal fairness. A utility allocation is maximally fair if no other utility allocation is "fairer" in some sense. That is, if a utility allocation is maximally fair, then to increase the utility received by a receiver s, we must lower that of another receiver j to a value less than the new utility of s. If a utility allocation is maximally fair, then any other utility allocation will be "unfair" or "less fair" to some receiver. Maximal fairness is a weaker notion of fairness as compared to maxmin fairness and lexicographic optimality. But, maximal fairness has various desirable fairness properties,e.g., it coincides with maxmin fairness and lexicographic optimality when maxmin fair utility allocation exists. We discuss other desirable fairness properties of maximally fair allocations later. Maximal fairness is probably the best option for non-strictly increasing utility functions, in view of the nonexistence of maxmin fair utility allocation and computational complexity of lexicographically optimal utility allocation. We will present a polynomial complexity algorithm for computing a maximally fair utility allocation in this paper. This algorithm yields a maxmin fair utility vector, if it exists. The algorithm does not use network specific properties, and is hence a fairly general purpose one. We also have provisions for minimum utility requirements and maximum utility constraints in our model.

This paper is organized as follows. We discuss our assumptions on the utility functions in section 2. We present our network model in section 3. We also introduce our fairness notion in this section. We present an algorithm for computing fair allocation of utilities in section 4. We prove that the algorithm yields a fair allocation of utilities in polynomial number of iterations in section 5. We conclude this paper in section 6.

2 Allocation of Utilities

We first discuss some possible utility functions and our assumptions on the utility functions. Utility of a bandwidth can be a measure of the reception quality a receiver obtains at that bandwidth. This measure can be objective as well as subjective. Objective measure can be quantities like information theoretic rate distortion, or the signal to noise ratio, where noise consists of the quantization noise. Obviously, these measures depend on the coding used, and hence it is important to allow different sessions to have different utility function, because different sessions may use different coding schemes. Other measures can be the number of layers obtained from a particular bandwidth. In fact, utility functions can be used to represent the hierarchical structure during bandwidth allocation. As an example scenario, we discuss this in details next. Subjective measures can be perceptual quality.

There can be various possibilities in hierarchical coding. The hierarchical structure may be completely flexible, i.e., given the desired rates of various receivers, the source generates as many layers as distinct receiver rates and tunes the layer bandwidths so as to match the receiver rates. The hierarchical structure may be predetermined, i.e., layer bandwidths can not be tuned to match

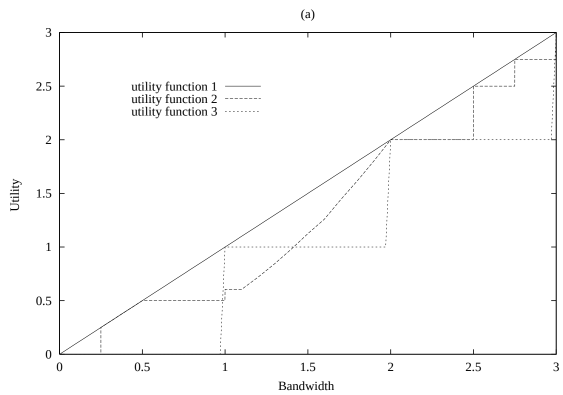

Figure 3: This figure shows three utility functions. For the first and the third, utility of bandwidth x is the actual bandwidth which can be allocated when x units of bandwidth are available. Utility function 1 models the case when hierarchical structure is completely flexible and any bandwidth can be allocated, if it is available. The utility function is linear and the utility set (set of allowable utilities) is the nonnegative real line. Utility function 3 models the case when hierarchical structure is completely predetermined. Only a discrete set of rates can be allocated and the utility function is stair-case. Utility set is the set of nonnegative integers. Utility function 2 models the case when hierarchical structure is flexible in certain ranges of bandwidth and predetermined in other ranges. Here, the utility is the qualitative gain from the bandwidth. In the flexible region, the function increases strictly, linearly or sublinearly. In the predetermined region, it is a stair-case function. Note that the step lengths are unequal, indicating unequal layer bandwidths for the same source. The utility set consists of closed intervals and some discrete points.

all desired receiver rates and receivers can not partially subscribe to layers. Also, it is possible that layer bandwidths can be fine tuned in certain ranges, while the granularity is coarse in other ranges. Hierarchical structure of different sources may be different, i.e., signal structure of some sessions may be flexible, while the signal structure may be predetermined for some others. Bandwidths consumed by layers may be different for different sources. Bandwidths consumed by different layers of the same source may also be different. We can model all these scenarios by choosing the utility functions appropriately. For example, the utility function should increase strictly, either linearly or sublinearly, for bandwidth ranges in which the hierarchical structure is flexible. The utility function should be stair case in the bandwidth range in which the hierarchical structure is predetermined. The step lengths may be chosen equal or unequal. The former represents the case in which all layers of the same source have equal bandwidth. The latter represents the case of nonuniform bandwidth layers. Refer to figure 3 for an example (we will introduce the concept of utility sets next). Thus, in general, the utility functions will neither be continuous nor strictly increasing. Fairness under this assumption is vastly different from the case in which the utility functions are strictly increasing.

We define our assumptions on the utility functions now. These assumptions are mild, and a large number of utility functions satisfy these assumptions. Formally, "utility" is a function of the bandwidth. We assume that the utility function for receivers of the same session is the same. "Utility set" of session i, A_i is the range of the utility function of session i. Receivers of a session i can possibly receive a utility if it belongs to the utility set of the session. For example, if utility of a bandwidth is the number of layers assigned out of the bandwidth, then the utility set is the set of nonnegative integers. If utility of a bandwidth is the bandwidth itself, then the utility set is the nonnegative real line. Refer to figure 3 for some example utility functions. We assume that the utility function satisfies the following technical assumptions for every i.

1. $A_i \cap (x, \infty) \neq \phi$, $\forall x \in R$.

2. Given any $y \geq 0$, $\max(A_i \cap [0, y])$ exists.

3. If $x \notin A_i$, then $\min(A_i \cap (x, \infty))$ exists.

4. Given any utility x, there always exists a minimum bandwidth required for session i to attain utility x, i.e., the set $\{y : u_i(y) = x\}$ has a minimum, where $u_i(y)$ is the session i utility corresponding to bandwidth y. The minimum bandwidth for utility x is denoted $\Upsilon_i(x)$.

5. The bandwidth function Υ_i is right continuous and strictly increasing ($\Upsilon_i(x)$ is the minimum bandwidth required to attain the session i utility x).

Figure 3 shows some sample utility sets.

The first assumption indicates that the utility set is not upper bounded. If necessary, we can impose maximum utility constraints separately. The other assumptions are more technical. Assumption (2) indicates that if there exists elements in a utility set arbitrarily close to but less

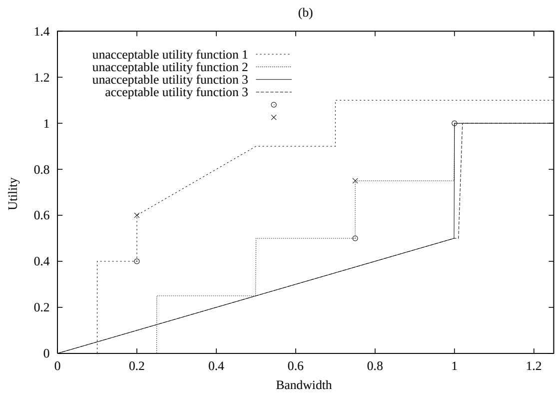

Figure 4: This figure shows three unacceptable utility functions. The first utility function (unacceptable utility function 1) is not acceptable with data point 'o.' The utility set violates property (3). However the utility function becomes acceptable if the data point is replaced by the one marked, 'x'. Similarly, the second utility function (unacceptable utility function 2) is not acceptable with data point 'o', (property (4) is violated). Again the utility function becomes acceptable if the data point is replaced by the one marked, 'x'. The utility function for the third one (unacceptable utility function 3), with data point 'o' does not satisfy property (2). However, a small modification indicated by "acceptable utility function" satisfies property (2).

than a positive real number z, then z is an allowable utility. Assumption (3) indicates that, if a real number x is not an allowable utility then the set of utilities greater than x has a minimum. In the last assumption (assumption (5)), continuity is intuitive because if the valid utilities in the utility set are very close, then the minimum required bandwidths should also be very close. Strictly increasing property of the minimum bandwidth is also intuitive. However, the utility function may not increase strictly and the same utility can correspond to a range of bandwidth. Only, the minimum bandwidth required to attain any particular utility needs to increase strictly. This implies that the utility functions must be non-decreasing. Hence, we do not make this assumption explicitly.

The last three assumptions exclude some functions from being valid utility functions. Refer to Figure 4 for examples of unacceptable utility functions. Any utility function satisfying properties (1) to (5) is acceptable. The assumptions are mild and most of the utility functions encountered in the networking context satisfy these assumptions. Also the utility functions of receivers of the same session must be the same. This assumption is reasonable, because utility structure, (e.g., hierarchical structure) often depends on source and there is only one source per session.

3 Network Model

We describe our network model in this section. We consider an arbitrary topology network with N multicast sessions. A multicast session is identified by the pair (v, U), where v is the source node of the session and U is the group of intended destination nodes. We assume that the traffic from node v is transported across a predefined multicast tree to nodes in U. The tree can be established during connection establishment phase if the network is connection oriented or can be established by some well known multicast routing protocol like DVMRP[6], CBT[1], etc. in an internet type network.

We call every source destination pair of a session a virtual session. For example, if a session n has source v_n and destination set U_n, where $U_n = \{u_{n1}, \ldots, u_{nt}\}$, then this session would correspond to t virtual sessions, $(v_n, u_{n1}), \ldots, (v_n, u_{nt})$. To ensure fairness in a multirate network, we need to consider fair utility allocation for the virtual sessions separately, instead of those for the overall sessions.

Let r_{ij} denote the utility allocated to the jth virtual session of the ith session. Let there be M virtual sessions in the entire network. A utility allocation vector $\vec{r}$ is an $M-$dimensional vector, with components r_{ij} (i.e., a utility allocation vector gives the utility allocations for all virtual sessions in the network). For simplicity, we will use only one set of indices for the virtual sessions, i.e., $(r_1, \ldots, r_M)$ is a utility allocation vector. An $M-$ dimensional vector $\vec{r} = (r_1, \ldots, r_M)$ is a feasible utility allocation if

1. the utility allocated to every receiver belongs to the utility set, i.e., $r_s \in A_{\chi(s)}, \ \forall \ s \in \{1, \ldots, M\}$, where $\chi(s)$ is the session of virtual session s.

2. $\mu_i \leq r_i \leq p_i \ \forall i$, where μ_i and p_i are respectively the minimum and maximum utiliities of virtual session i, $p_i \geq \mu_i \geq 0$,

3. Total bandwidth consumed in every link is less than or equal to the capacity of the link. Bandwidth consumed in a link is the sum of the bandwidths consumed by different sessions traversing the link. Bandwidth consumed by a session in a link is the bandwidth consumed by the *session link utility*, where session link utility is the maximum utility allocated to virtual sessions of the same session traversing the link. Let $n(l)$ denote the set of sessions traversing link l, $m(k,l)$ denote the set of virtual sessions of session k passing through link l, C_l denote the capacity of link l, λ_{il} denote the session link utility of session i in link l and $\Upsilon_i(x)$ denote the minimum bandwidth required by session i to attain utility x . Then $\lambda_{il} = \max_{j \in m(i,l)} r_j$ and

$$\sum_{i \in n(l)} \Upsilon_i(\max_{j \in m(i,l)} r_j) \leq C_l \text{ or equivalently } \sum_{i \in n(l)} \Upsilon_i(\lambda_{il}) \leq C_l \qquad \text{(capacity condition).}$$

   In a nutshell, this condition says that an essential condition for the feasibility of a utility allocation is that the corresponding bandwidth allocation satisfies the capacity constraints.

Example 3.1: Consider the network shown in figure 5. For session 1 the utility of bandwidth x is 0 for $x < 2$ and $\lfloor x - 1 \rfloor$ otherwise. This models the scenario where the utility of a bandwidth is the number of layers obtained from the bandwidth, and the first layer consumes 2 units, whereas

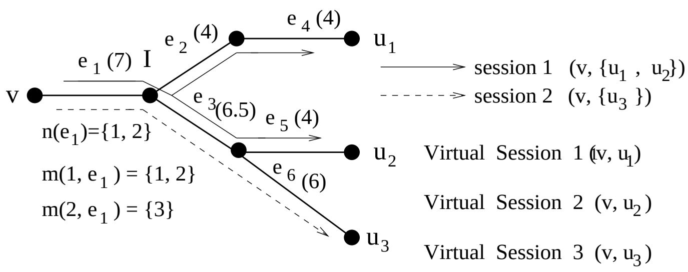

Figure 5: The numbers in brackets, () denote the capacities of the respective links.

others consume 1 unit each. So bandwidth consumed by session link utility x is $x+1$ for $x > 0$ and 0 otherwise. For session 2 utility of bandwidth x is x itself. This models the scenario in which the hierarchical structure is flexible. Here r_i is the utility allocated to virtual session i. The capacity constraint for link e_1 is $\lambda_{1e_1} + 1 + \lambda_{2e_2} \leq 7$, where $\lambda_{1e_1} = \max(r_1, r_2)$ and $\lambda_{2e_2} = r_2$.

We define the fairness notions for the utility allocations. A feasible utility allocation is maxmin fair if it is not possible to maintain feasibility and increase the utility of a virtual session without decreasing that of any other virtual session which has equal or lower utility. More formally, a feasible utility allocation vector $\vec{r}^1$ is maxmin fair if it satisfies the following property with respect to any other feasible utility allocation vector $\vec{r}^2$: if there exists i such that the ith component of $\vec{r}^2$ is strictly greater than that of $\vec{r}^1$ ($r_i^2 > r_i^1$), then there exists j such that the jth component of $\vec{r}^1$, r_j^1 is less than or equal to the ith component of $\vec{r}^1$, r_i^1 ($r_j^1 \leq r_i^1$) and the jth component of $\vec{r}^2$ (r_j^2) is strictly less than the jth component of $\vec{r}^1$ ($r_j^2 < r_j^1$). A maxmin fair utility allocation may not exist in general for non strictly increasing utility functions. To see this, consider the case in which utility of a bandwidth is the number of layers obtained from the bandwidth, and every layer of every session has the same bandwidth. Maxmin fair allocation of utilities reduces to the maxmin fair allocation of uniform bandwidth layers of [18]. It was shown in [18] that maxmin fair layer allocation does not always exist. Refer to Figure 6 for an example.

We will consider the notion of lexicographic optimality. Given an $M-$ dimensional vector $\vec{V}$, define its lexicographically ordered version $\hat{V}$ as follows: $\hat{v}_j = v_k$ for some k and $\hat{v}_1 \leq \hat{v}_2 \leq \ldots \hat{v}_M$. In other words, components of $\hat{V}$ are an ordered version of those of $\vec{V}$. A vector $\vec{V}^1$ is lexicographically greater than a layer allocation vector $\vec{v}^2$ if there exists i such that $\hat{v}_i^1 > \hat{v}_i^2$ and $\hat{v}_j^1 = \hat{v}_j^2$ if $j < i$. Vectors $\vec{v}^1$ and $\vec{v}^2$ are lexicographically equal if $\hat{v}^1 = \hat{v}^2$. Vector $\vec{v}^1$ is lexicographically less than $\vec{v}^2$ if $\vec{v}^2$ is lexicographically greater than $\vec{v}^1$. A feasible vector $\vec{v}$ is lexicographically optimal if every feasible layer allocation vector is lexicographically less than or equal to $\vec{v}$. Informally, a vector is lexicographically optimal, if its smallest component is the largest amongst the smallest components of all feasible vectors, subject to this it has the largest second smallest component and so on. For example $(1, 0, 3, 3)$ is a lexicographically optimal layer allocation in Figure 6. So, this is the lexicographically optimal utility allocation as well, if the utility of a bandwidth is the number of layers allocated.

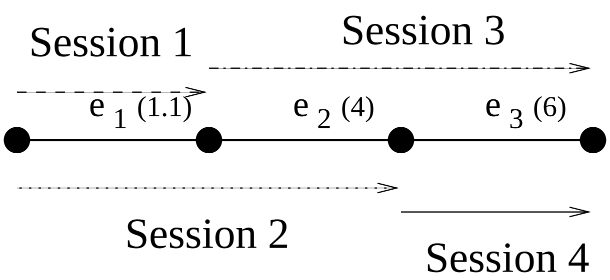

Figure 6: This figure shows an example network which does not have a maxmin fair layer allocation, but has a lexicographically optimal layer allocation. The sources emit unit bandwidth layers. Hence, there are only a finite number of layer allocations. None of these vectors are maxmin fair. For each of these vectors, it is possible to maintain feasiblity and increase the number of layers of one session, possibly hurting another session, but the other session has higher number of layers. For example, consider the allocation $(1, 0, 3, 3)$. It is possible to increase the number of layers of session 2, by hurting session 1. However session 1 has higher number of layers than session 2 in this allocation.

Incidentally, lexicographic optimality and maxmin fairness are different concepts, though several researchers have used these interchangeably. Figure 6 illustrates the difference. Maxmin fair utility allocation does not exist in the network shown there, but lexicographically optimal allocation exists. We have shown in [16] that if a vector is maxmin fair in a feasible set, then it is lexicographically optimal. The important difference is that a lexicographically optimal allocation always exists in any closed, bounded set, but a maxmin fair allocation may not exist in such a set. A lexicographically optimal utility allocation vector always exists under our assumptions on the utility set. This is because clearly the feasible set is bounded. Also, the feasible set is closed from properties (2) and (3). However, its computation is NP-hard. This follows from the fact that lexicographic optimal allocation of layers is a special case of lexicographic optimal allocation of utilities, and the former has been proved to be NP-hard in [18]. Refer to [19] for proof.

So we revert to our weaker definition for fairness, maximal fairness. We first define the concept of *relative fairness* of utility vectors. This concept has been introduced in [15]. A utility allocation vector $\vec{r}^1$ is *fairer* than another utility allocation vector $\vec{r}^2$, if for every virtual session i which has greater utility under $\vec{r}^2$ than under $\vec{r}^1$, there is some other virtual session j whose utility was already no more than that of i under $\vec{r}^1$, and has been decreased further by $\vec{r}^2$. A more formal definition of relative fairness follows.
A utility allocation vector $\vec{r}^1$ is *fairer* than another utility allocation vector $\vec{r}^2$ if

- $\vec{r}^1 \neq \vec{r}^2$ and
- if there exists an i such that $r_i^1 < r_i^2$, then there exists a j such that $r_j^1 \leq r_i^1$ and $r_j^2 < r_j^1$ (r_k^l is the kth component of $\vec{r}^l$, $l = 1, 2$.)

A utility allocation vector $\vec{r}$ is maximally fair if it is feasible and if no other feasible utility allocation vector is fairer than $\vec{r}$.

As we show later, maximally fair utility allocation exists and can be computed in polynomial complexity. But, first we investigate, whether maximal fairness is a good notion of fairness. Note that from the definition of maximal fairness, it is clear that if a utility allocation is maximally fair, then we can increase the utility of a receiver s, only by decreasing that of another receiver to a value below that of the new utility value of receiver s. Thus we can increase the utility of s only by being unfair to some other receiver. Also, if maxmin fair utility allocation exists, then it is the only maximally fair utility allocation. This follows because by definition of relative fairness, the maxmin fair utility allocation (if one exists) is fairer than any other feasible utility allocation. We have proved this in [19]. So any algorithm for computation of a maximally fair utility allocation will yield a maxmin fair utility allocation, if one exists. Besides, any lexicographically optimal vector is maximally fair in any feasible set(refer to [16] for proof). So lexicographically optimal utility allocation is one of the maximally fair ones. Thus maximal fairness is a good notion of fairness.

We would like to point out that it is not obvious that a maximally fair utility allocation can be computed in polynomial complexity. Consider the simple case of strictly concave utility functions. Maxmin fair allocation exists in this case, and hence as observed before any algorithm for computing a maximally fair allocation will compute a maxmin fair allocation. Note that maxmin fair utility allocation can be computed by a series of concave optimizations with nonlinear feasible sets. The first step would be to maximize the utility of any one receiver subject to the capacity constraints and an added constraint that utilities of all receivers in the network must be equal. Let u_1 be the output of this optimization. Next, the computation needs to find out which receivers can have a utility increase without hurting other receivers. Let S be the set of these receivers. In the next stage, again utilities of any one of the receivers in S should be maximized under the capacity constraints and the additional constraints that the utility of no receiver falls below u_1, and the utilities of all receivers in S are equal. The process continues. Note that each of these steps is a nonlinear optimization, and standard nonlinear optimization theory does not guarantee optimum computation in a finite number of steps. The situation becomes more complicated when the utility functions are more general, as we are considering here (note that in our case utility functions are not even continuous and can have discrete jumps). However, we will exploit the specific properties of these optimizations and the nature of the feasible sets to develop a polynomial complexity algorithm.

We have earlier presented polynomial complexity algorithms for computing the maximally fair utilities for the special cases that $u_i(x) = x$ and $u_i(x) = b\lfloor \frac{x}{b} \rfloor$, for every session i and some $b > 0$[17][18]. However, utility functions had well tractable structure in both cases. This structure does not exist in the general case we treat here, and as we discussed earlier in many cases of practical interest utility functions do not exhibit any of these structures. So it would be interesting from a mathematical as well as a practical standpoint to study this more general case. In fact, as expected complexity increases in this more general case as compared to the special cases mentioned above. While the algorithm for the case $u_i(x) = x$ completes in M iterations (M is the total number of virtual sessions) and the algorithm for the case $u_i(x) = b\lfloor \frac{x}{b} \rfloor$, completes in $|\mathcal{L}|M$ iterations ($\mathcal{L}$ is the set of links), the algorithm for this general case may require up to $|\mathcal{L}|M^2$ iterations. There exists necessary and sufficient conditions for testing whether an allocation is maximally fair in both the special cases, but these are not sufficient in the general case. We will present a necessary and sufficient condition in this general case later in this paper.

We first describe the algorithm of [5] for computing maxmin fair utility allocation in unicast networks and show that it is ambiguous and does not even yield a maximally fair allocation when the utility functions do not increase strictly, even for unicast networks. This is the only work in fair allocation of utilities. The algorithm of [5] computes a fair share of bandwidth for every session in every link on the path of the session. The fair share is computed so that the utilities of all sessions traversing the link are equalized, and the link capacity is fully utilized. Every session gets a bandwidth equal to the minimum fair share on its path. Next, sessions which attain the minimum utility are considered bottlenecked. The bandwidths of the bottlenecked sessions are not changed any further. Now, the available capacity of a link is the difference between the link capacity and the sum of the bandwidths of the bottlenecked sessions. The next iteration computes the fair shares similarly, except that it uses available capacity instead of the full link capacity, and considers nonbottlenecked sessions only. Now we show that this algorithm does not obtain fair allocations, if utility functions are not strictly increasing.

Example 3.2: Consider a network with a single link. Two sessions traverse the link. The link has capacity 1 unit. Let the utility of a bandwidth be the floor of the bandwidth. Thus if x units of bandwidth is assigned to a session, the utility of the session is $\lfloor x \rfloor$ (For the hierarchical transmission case, this utility is equal to the number of layers obtained from the bandwidth if the sources emit unit bandwidth layers). The capacity of the link will be utilized and both the sessions will get equal utility only if they are assigned α and $1-\alpha$ units respectively, where $0 < \alpha < 1$. If $\alpha \leq 0.5$, session 1 is saturated. If $\alpha \geq 0.5$, session 2 is saturated. The algorithm does not specify the value of α in this case. For $\alpha = 0.5$, both sessions saturate and the algorithm terminates with a layer allocation $(0, 0)$. Wlog $\alpha < 0.5$. Session 1 is saturated with 0 utility. The available capacity is $1-\alpha$ units. It has to be assigned to session 2, which still gets $\lfloor 1-\alpha \rfloor = 0$ utility. Session 2 now saturates. So for all possible executions of this algorithm, the utility allocation is $(0, 0)$. This clearly is not a maximally fair utility allocation. So this approach does not obtain maximally fair utility allocation always. In fact, the approach is ambiguous as it does not specify α in this case.

It remains to be seen whether the algorithm in [5] can be modified to attain maximally fair utilities for non strictly increasing utility functions. No modification will attain maxmin fair utility allocation in all cases, since maxmin fair utility allocation does not exist always.

We conclude this section with a few observations.

There exists a necessary and sufficient condition for a utility allocation to be maximally fair. A utility allocation is maximally fair iff every virtual session has a "generalized bottleneck link" on its path. We will introduce the notion of a "generalized bottleneck" link in page 38. As we mentioned before, there are similar concepts of bottleneck links in the special cases studied in [17] and [18]. However, the bottleneck condition is more complicated in this general case.

This observation will illustrate the scope of fair allocation of utilities. In general, all bandwidth allocations which satisfy the capacity constraints in the links are not interesting. The interesting ones are those in which no component can be increased without decreasing another component† (similar to the notion of pareto optimality). We denote these "boundary allocations." For unicast and unirate multicast networks, given any boundary allocation there exists a choice of linear utility functions such that the boundary allocation corresponds to the maxmin fair utility allocation corresponding to the chosen utility functions. The linear utility function can be obtained as follows. If $(\gamma_1, \ldots, \gamma_M)$ is the bandwidth allocation, then the utility function for session i is $u_i(x) = \frac{\max_j \gamma_j}{\gamma_i} x$. We have tacitly assumed that $\gamma_i \neq 0$, $\forall i$, and note that under this assumption, these utility functions satisfy the assumptions in Section 2. If $\gamma_i = 0$ for some i, then the utility functions can be $u_i(x) = x + \max_j \gamma_j - \gamma_i$. Again these satisfy our assumptions in Section 2. For multirate, multicast the same observation holds if we allow different utility functions for different receivers for the same session. We believe the algorithm we present here, and the framework in general can be extended to allow this possibility. Incidentally, [9] showed a parallel result which states that for certain network topologies, given any boundary allocation, there exists utility functions, such that the bandwidth allocation corresponds to total utility maximization. It follows that the output of fair allocation of utilities can be obtained in their framework of utility maximization, but that is not the best approach for utility fairness. This is because the algorithms for utility maximization in [9] do not terminate in finite number of iterations, and construction of the new utility functions for maximization given the utility functions for fairness, is likely to be more complicated than directly computing the fair allocation.

Maximum utility constraints can be incorporated by adding artificial links between receivers with maximum utility constraints and the rest of the network. Capacity of such an artificial link is equal to the bandwidth consumed by the maximum utility of the respective receiver. So, henceforth we shall ignore the maximum utility constraints.

4 Algorithm for Computation of Maximally fair Utility Allocation

Now we present an algorithm for computing a maximally fair utility allocation. First, we introduce certain terminologies. Notations like $\chi(s)$, $n(l)$ and $m(i, l)$, are defined in page 10 respectively.

Ψ_i is a real valued function defined as $\Psi_i(x) = \inf_{y \in A_i \cap (x, \infty)} y$, $\forall\, i$. Loosely speaking, $\Psi_i(x)$ is the smallest session i utility greater than x. If the utility set of session i, is a set of integers, then $\Psi_i(x) = \lceil x \rceil$ if x is not an integer and $\Psi_i(x) = x + 1$, otherwise. This function is well defined by property (1) of the utility set.

$\Phi_i(x)$ is a real valued function defined as $\Phi_i(x) = \max_{y \in A_i \cap [0, x]} y$. Thus, $\Phi_i(x)$ is the largest session

†The component decreased may have a value greater than, equal to, or less than the component increased.

i utility not exceeding x. If the utility set of session i, is a set of integers, then $\Phi_i(x) = \lfloor x \rfloor$. This function is well defined by property (2) of the utility set.

The minimum session link utility, μ_{il} is the maximum of the minimum utilities of session i virtual sessions traversing l, i.e., $\mu_{il} = \max_{s \in m(i,l)} \mu_s$.

L_s is the set of links traversed by virtual session s.

A virtual session s is saturated w.r.t. a utility vector $\vec{r}$ if it traverses a link l in which

$$\begin{aligned} r_s &= \lambda_{\chi(s)l} \\ \text{If } r_s = \Psi_{\chi(s)}(r_s),\ \sum_{i \in n(l)} \Upsilon_i\left(\lambda_{il}\right) &= C_l \\ \text{If } r_s < \Psi_{\chi(s)}(r_s),\ \sum_{i \in n(l)} \Upsilon_i\left(\lambda_{il}\right) &> C_l - \left(\Upsilon_{\chi(s)}\left(\Psi_{\chi(s)}(r_s)\right) - \Upsilon_{\chi(s)}\left(r_s\right)\right) \end{aligned}$$

A session is *saturated* on a link l if all the virtual sessions of the session traversing the link l are saturated.

$\eta_l(k)$ denotes the link control parameter of link l at the end of the kth iteration. Link control parameter is the iterate which would be used in computation of the maximally fair utilities. An utility equal to $\eta_l(k)$ would have been allocated to the unsaturated virtual sessions traversing the link, if there were no bandwidth constraints on other links, and feasible utility allocations need only satisfy the capacity constraints.

$\eta_{il}(k)$ denotes the session link parameter of session i traversing link l. It is the utility assigned to session i, if there were no bandwidth constraints for any of its virtual sessions on other links and feasible utility allocations need only satisfy the capacity and the minimum utility constraints (i.e., utility set is the nonnegative real line). Note that $\eta_{il}(k)$ differs from $\eta_l(k)$ because of the minimum rate requirements of session i in link l.

$\omega_s(k)$ is the utility that is assigned to virtual session s at the end of the kth iteration if it is restricted to receive no more than any of its session link control parameters, $\eta_{\chi(s)l}(k)$, on its path. Here, it is the largest valid utility in $A_\chi(s)$ not exceeding its minimum session link control parameter on its path. Note that $\omega_s(k)$ differs from individual session link control parameters $\eta_{il}(k)$ on account of bandwidth constraints in other links and the nature of the individual utility sets of the sessions.

$\Omega_{il}(k)$ is the utility allocated to session i in link l, under the utility vector $\vec{\omega}(k)$. $\Omega_{il}(k) = \max_{j \in m(i,l)} \omega_j(k)$.

$r_s(k)$ is the utility allocated to virtual session s at the end of the kth iteration. The value of $r_s(k)$ can either be equal to or greater than $\omega_s(k)$. $\vec{r}(k)$ denotes the utility vector at the end of the kth iteration, with components, $r_s(k)$.

$\lambda_{il}(k)$ is the utility allocated to the session i in link l at the end of the kth iteration. It is actually the maximum of the utilities allocated to the virtual sessions in $m(i,l)$ at the end of the kth iteration.

$\Lambda(k)$ is the set of virtual sessions which are saturated w.r.t. utility allocation $\vec{\omega}(k)$.

$S(k)$ denotes the set of unsaturated virtual sessions at the end of the kth iteration.

$\Xi_l(k)$ denotes the set of unsaturated sessions traversing link l at the end of the kth iteration.

$F_l(k)$ denotes the total bandwidth consumed by the saturated sessions traversing link l at the end of the kth iteration.

The algorithm follows.

1. $k = 0$, $\eta_l(0) = 0$, $F_l(0) = 0$, $\Xi_l(0) = n(l)$ $\forall$ link l, $S(0) = \{1, \ldots, M\}$, $r_j(0) = \mu_j$, $\forall j \in S(0)$, $\lambda_{il}(0) = \max_{j \in m(i,l)} r_j(0)$.

2. $k \to k+1$

3. For every link l in the network compute the link control parameter. For this, first compute a variable $\alpha_l(k)$. If $\Xi_l(k-1) \neq \phi$, then $\alpha_l(k)$ is computed as follows:

$$\alpha_l(k) = \sup_{\theta}\{\theta : F_l(k-1) + \sum_{i \in \Xi_l(k-1)} \Upsilon_i \left(\max\left(\Phi_i(\theta), \lambda_{il}(k-1)\right)\right) \leq C_l\}$$

   (It can be shown that the above set is nonempty for all k.)
   If $F_l(k-1) + \sum_{i \in \Xi_l(k-1)} \Upsilon_i \left(\max\left(\Phi_i(\alpha_l(k)), \lambda_{il}(k-1)\right)\right) \leq C_l$, then link control parameter for a link l is $\eta_l(k) = \alpha_l(k)$, else link control parameter, $\eta_l(k)$ is chosen s.t. $\eta_l(k) < \alpha_l(k)$, and $\eta_l(k)$ satisfies the following conditions:

   (a) $\eta_l(k) \geq \max\left(\eta_l(k-1), \min_{i \in \Xi_l(k-1)} \lambda_{il}(k-1)\right)$ and

   (b) if session $i \in \Xi_l(k-1)$ and $\lim_{\theta \uparrow \alpha_l(k)} \max\left(\Phi_i(\theta), \lambda_{il}(k-1)\right) <$ $\max\left(\Phi_i\left(\alpha_l(k)\right), \lambda_{il}(k-1)\right)$, then $\eta_l(k) \geq \lambda_{il}(k-1)$ and $(\eta_l(k), \alpha_l(k)) \cap A_i = \phi$,

   For all unsaturated sessions i passing through link l, $(i \in \Xi_l(k-1))$, $\eta_{il}(k) = \max(\eta_l(k), \lambda_{il}(k-1))$.

4. Compute $\omega_s(k)$ for all virtual sessions $s \in S$, where $\omega_s(k) =$ $\Phi_{\chi(s)}(\min_{l \in L_s} \eta_{\chi(s)l}(k))$, if $s \in S(k-1)$, else $\omega_s(k) = r_s(k-1)$.

5. For every link l in the network, compute $\Omega_{il}(k)$ for every session i in $n(l)$, $\Omega_{il}(k) = \max_{s \in m(i,l)} \omega_s(k)$.

6. Compute the set of virtual sessions saturated during the kth iteration under utility vector $\vec{\omega}(k)$, $\Lambda(k)$. (Refer to page 16 for saturation conditions)

7. If $\Lambda(k) \neq \phi$, compute the utilities allocated to the virtual sessions after the kth iteration, via, $r_s(k) = \omega_s(k)$, $\forall s$ and go to step (9).

8. If possible, find a virtual session $s \in S(k-1)$ such that,

$$\begin{aligned} \omega_s(k) &= \Phi_{\chi(s)}\left(\min_{l \in L_s} \eta_l(k)\right), \\ \Psi_s\left(\omega_s(k)\right) &> \omega_s(k), \end{aligned}$$

for some $l \in L_s$ s.t. $\Phi_{\chi(s)}\left(\min_{l_1 \in L_s} \eta_{l_1}(k)\right) = \Phi_{\chi(s)}\left(\eta_l(k)\right)$

$$\begin{aligned} \omega_s(k) &= \Omega_{\chi(s)l}(k) \text{ and} \\ \mathrm{Ran}(\Phi_{\chi(s)}) \cap (\eta_l(k), \alpha_l(k)) &= \phi, \end{aligned}$$

and for all $l \in L_s$ s.t. $\Phi_{\chi(s)}\left(\min_{l_1 \in L_s} \eta_{l_1}(k)\right) = \Phi_{\chi(s)}\left(\eta_l(k)\right)$

$$\Omega_{il}(k) = \max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k-1)\right) \;\; \forall\, i \in \Xi_l(k-1)$$

If no such s is found in $S(k-1)$, again $r_j(k) = \omega_j(k)$ for all virtual sessions, j, otherwise compute $r_j(k)$ for all virtual sessions j as

$$r_j(k) = \begin{cases} \omega_j(k) & j \neq s \\ \Psi_{\chi(j)}\left(\omega_j(k)\right) & \text{otherwise.} \end{cases}$$

9. For every link l in the network compute the session link utility in link l, for every session in $n(l)$, $\lambda_{il}(k) = \max_{s \in m(i,l)} r_s(k)$.

10. Compute the set of virtual sessions unsaturated after the kth iteration, $S(k)$. (Refer to page 16 for saturation conditions)

11. If $S(k) = \phi$, i.e., all virtual sessions are saturated, the algorithm terminates, else go to the next step.

12. For every link l, compute the set of unsaturated sessions passing through link l at the end of the kth iteration, $\Xi_l(k)$.

13. For every link l, for which $\Xi_l(k) \neq \phi$, compute the bandwidth consumed by the saturated sessions passing through link l, $F_l(k) = \sum_{i \in \Xi_l(k)} \Upsilon_i\left(\lambda_{il}(k)\right)$.

14. Go to step (2).

At every iteration k, the algorithm computes a "fair utility share" for every session i, in every link l, the session link parameter, $\eta_{il}(k)$. Since a virtual session can receive utilities in utility set A_i only, utility $\Phi_i\left(\eta_{il}(k)\right)$ is offered to all virtual sessions of session i traversing the link. A virtual session is allocated the minimum utility offered on its path. If no virtual session saturates in the current iteration, then the residual bandwidth may be used to increment the utility of some virtual session traversing this link and satisfying the properties mentioned in step (8). The utility of such a virtual session is increased to the next higher value. This is done because otherwise, the algorithm can continue forever. This is because in the next iteration, the same link control parameter will be computed and the process repeats again and again. A session is saturated if all its virtual sessions are saturated. The bandwidth consumed by the saturated sessions, if any, are computed. This bandwidth is subtracted from the link capacity, and the link control parameters are recomputed at the beginning of every iteration as per step (3) and the process continues. The algorithm determines which virtual session and sessions are saturated. A new iteration starts if still there are unsaturated virtual sessions. Upon termination, we have a maximally fair utility allocation. A natural question to ask is that whether this algorithm terminates at all. This is because, as it appears, utility of a virtual session may increase one step at an iteration, and the utility functions may have arbitrarily large number of step jumps in any bandwidth range. However, we would show that the algorithm terminates in a finite number of iterations, and the number of step jumps do not affect the number of iterations required for termination.

As it turns out, the utility function $u(x) = b\lfloor \frac{x}{b} \rfloor$ treated in [18] is a special case of the more general utility functions we study here. We point out the additional complications. Note that the computation of the link control parameters differs from [18]. To accommodate the general nature of the problem, we needed an additional quantity $\alpha_l(k)$ to compute the link control parameter $\eta_l(k)$ of link l in this case. There is no unique choice of the link control parameter, and among the possible choices, we need to choose a value which is at least as large as that of the previous iteration. There is a unique choice for the link control parameter in the algorithm of [18], and this choice ensures that the link control parameter does not decrease with subsequent iterations. The saturation conditions differ in the two algorithms. Also, selection of the virtual session for utility incrementation differs in the two algorithms. All these distinctions also apply to another special case $u_i(x) = x$, studied in [17]. In addition, every receiver just receives the minimum session link control parameter on its path in [17], and the utility incrementation step is not required there. The utility incrementation step is crucial for convergence here.

Theorem 1 (Finite-Termination Theorem) *The algorithm terminates in at most $M + |\mathcal{L}|M^2$ number of iterations, where $\mathcal{L}$ is the set of links and M is the number of virtual sessions.*

Theorem 2 (Generalized Maximal Fairness) *The output utility allocation vector is*

1. *maximally fair,*

2. *maxmin fair, if a maxmin fair utility allocation exists.*

We prove these in the next section. We would like to point out that the termination result holds independent of the nature of the utility functions. The utility functions can have step jumps, and the size of the step jumps are allowed to be arbitrarily small. The number of iterations required in this case is still upper bounded by $M + |\mathcal{L}|M^2$. This is somewhat surprising as it appears from the algorithm that the utility of a virtual session may increase by a step at a time. However, it is not necessary that the utility of a virtual session increases one step at a time, but can increase by several steps depending on the utility functions of the session and other sessions sharing the link with this virtual session, and the link capacities on the path of the virtual session. Hence, the number of iterations is finite. We prove this more formally later.

Every step of this algorithm has a complexity of $O(\kappa|\mathcal{L}|M)$, where κ depends on the utility functions. If we assume κ to be a constant, then the complexity of every step is $O(|\mathcal{L}|M)$. The algorithm must terminate in $M + |\mathcal{L}|M^2$ iterations. Thus the overall complexity of this algorithm is $O(|\mathcal{L}|^2 M^3)$. The constants will depend on the computation of the limits for the utility functions. We conclude this section with certain salient features of this algorithm.

This algorithm is amenable to distributed implementation. The criteria for determination of utility of a virtual session uses information along the path of the virtual session mainly. The only place where the algorithm uses global information is that $r_s(k) = \Psi_{\chi(s)}(\omega_s(k))$, for at most one virtual session, s. This feature of the algorithm is not crucial to the proof of maximal fairness of the output and is a matter of convenience. This increase in utility can be carried out for multiple virtual sessions, subject to feasibility and as long as they satisfy the criteria of step (8) and the algorithm will still output a maximally fair utility allocation.

This algorithm terminates in fewer number of iterations in some special cases. For example, if the utility function is $u_i(x) = x$ for every session i, then the algorithm needs at most M iterations. We have treated this special case in [17] and the algorithm presented there is a special case of this one. If the utility functions are such that the ratio between the maximum and minimum step jumps are bounded by a constant c, then the algorithm needs at most $c|\mathcal{L}|M$ iterations. Again the case we treated in [18] ($u_i(x) = b\lfloor \frac{x}{b} \rfloor$ for every session i) falls under this category and the algorithm presented there is a special case of this one. We would like to point out that we have not made any assumptions on the step jump sizes of the utility functions here.

5 Proof of Theorems 1 and 2

We prove that the algorithm presented in Section 4 outputs a maximally fair utility allocation in a finite number of iterations. We will prove Theorems 1 and 2 in this section. At first, we present some properties of the intermediate utility allocations of this algorithm. We will use these in proving both the theorems. The second subsection will prove Theorem 1 and the last will prove Theorem 2.

5.1 General Assumptions and Properties of the Algorithm

We first present some properties of the Φ function. Recalling the definition, $\Phi_i(x)$ is a real valued function defined as $\Phi_i(x) = \max_{y \in A_i \cap [0,x]} y$, for session i. Intuitively, $\Phi_i(x)$ is the largest session i

utility not exceeding x. If the utility set of session i, is a set of integers, then $\Phi_i(x) = \lfloor x \rfloor$. Function Φ_i satisfies the following properties.

1. Φ_i is a nondecreasing function for all i.

2. $\Phi_i(x) \leq x$.

3. $\Phi_i(\Phi_i(y)) = \Phi_i(y)$, for all y.

4. $\text{Ran}(\Phi_i) \cap (\Phi_i(x), x] = \phi$.

5. $\Phi_i(x)$ is right continuous for all x.

(1), (2) and (3) follow from the definition. (4) follows from the observation that $\text{Ran}(\Phi_i) = A_i$ and $A_i \cap (\Phi_i(x), x] = \phi$ by definition. Now we argue (5). Let $\Phi_i(x) = x$. Thus $\forall \delta > 0$, $x = \Phi_i(x) \leq \Phi_i(x+\delta) \leq x + \delta$. It follows that $|\Phi_i(x+\delta) - \Phi_i(x)| \leq \delta$, $\forall \delta > 0$. Let $\Phi_i(x) < x$. Thus $x \notin A_i$. Thus by property (3) of A_i there exists $y > x$ such that $A_i \cap (x, y) = \phi$. Also $(\Phi_i(x), x] \cap A_i = \phi$. Thus $(\Phi_i(x), y) \cap A_i = \phi$. Thus $\Phi_i(z) = \Phi_i(x)$, for all $z \in (x, y)$. Hence right continuity follows.

Next, we present certain properties of the Ψ_i function. Again, Ψ_i is a real valued function defined as $\Psi_i(x) = \inf_{y \in A_i \cap (x,\infty)} y$, $\forall\ i$. Loosely speaking, $\Psi_i(x)$ is the smallest session i utility greater than x. If the utility set of session i, is a set of integers, then $\Psi_i(x) = \lceil x \rceil$ if x is not an integer and $\Psi_i(x) = x + 1$, otherwise. Function $\Psi_i(x)$ satisfies the following properties:

1. $\Psi_i(x) \geq x \quad \forall i$ (follows from definition.)

2. Ψ_i is a nondecreasing function for all i.

3. $\Phi_i\left(\Psi_i(x)\right) = \Psi_i(x)$. This can be argued as follows. Given any $\delta > 0$, $[\Psi_i(x), \Psi_i(x) + \delta) \cap \text{Ran}\left(\Phi_i\right) \neq \phi$. Thus $\Phi_i\left(\Psi_i(x) + \delta\right) \geq \Psi_i(x)\ \forall\ \delta > 0$. If $\Phi_i\left(\Psi_i(x)\right) < \Psi_i(x)$, then by the right continuity of Φ_i, there exists a $\epsilon > 0$ such that $\Phi_i\left(\Psi_i(x) + \epsilon\right) \in \left[\Phi_i\left(\Psi_i(x)\right), \Psi_i(x)\right)$, but by the previous observation, $\Phi_i\left(\Psi_i(x) + \epsilon\right) \geq \Psi_i(x)$, for all $\epsilon > 0$. This is a contradiction.

We will use these properties in our proofs.

We assume that every link lies on the path of at least one session, i.e., $n(l) \neq \phi$, $\forall l \in \mathcal{L}$. If a network has a link which does not lie on the path of any session, it can be removed, without affecting the feasible set of rate vectors. We also assume that the set of feasible utility vectors is nonempty and the minimum utility of virtual session s, μ_s satisfies the following properties:

1. $\mu_s \geq 0$ and

2. $\mu_s \in A_{\chi(s)}$, for all virtual sessions s. This means that μ_s is in the range of the function, $\Phi_{\chi(s)}$, for all virtual sessions s.

First, we need to show that it is possible to execute this algorithm. More precisely, we need to show that the set

$\{\theta : F_l(k-1) + \sum_{i \in \Xi_l(k-1)} \Upsilon_i(\max(\Phi_i(\theta), \lambda_{il}(k-1))) \leq C_l\}$ is nonempty in every iteration k. We will actually prove this in Lemma 5. So technically Lemmas 1 to 4 state properties under the assumption that the above set is nonempty for all iterations $1, \ldots k$. We do not explicitly state this assumption in these Lemmas though. Under this assumption,

$$\sum_{i \in n(l) \setminus \Xi_l(k-1)} \Upsilon_i(\lambda_{il}(k-1)) + \sum_{i \in \Xi_l(k-1)} \Upsilon_i(\max(\Phi_i(\eta_l(k)), \lambda_{il}(k-1))) \leq C_l \tag{1}$$

The last inequality follows from the definitions of $\alpha_l(k)$ and $\eta_l(k)$.

The following lemma establishes an intuitive property, i.e., the utilities allocated to the virtual sessions at the end of every iteration belong to the respective utility sets.

Lemma 1 *For all virtual sessions s and iterations k, $\Phi_{\chi(s)}(r_s(k)) = r_s(k)$, $\Phi_{\chi(s)}(\omega_s(k)) = \omega_s(k)$. For all sessions i, link l and iterations k, $\Phi_i(\lambda_{il}(k)) = \lambda_{il}(k)$, $\Phi_i(\Omega_{il}(k)) = \Omega_{il}(k)$. It follows that $r_s(k), \omega_s(k) \in A_{\chi(s)}$, and $\Omega_{il}(k), \lambda_{il}(k) \in A_i$.*

Proof of Lemma 1: We show that the lemma holds for $k = 0$. $r_s(0) = \omega_s(0) = \mu_s$. By assumption, μ_s is in the range of $\Phi_{\chi(s)}$ function. By property (3), $\Phi_{\chi(s)}(\mu_s) = \mu_s$. Thus $\Phi_i(\lambda_{il}(0)) = \Phi_i(\mu_{il}) = \mu_{il} = \lambda_{il}(0)$. $\Omega_{il}(0) = \lambda_{il}(0)$. Thus the result holds for $\Omega_{il}(0)$ as well. Let the lemma hold for iteration k. Thus, $\omega_s(k+1) = r_s(k)$, if $s \notin S(k)$. $\omega_s(k+1) = \Phi_{\chi(s)}(\min_{l \in L_s} \eta_{ls}(k+1))$, if $s \in S(k)$. From induction hypothesis and property (3) of the $\Phi_{\chi(s)}$ function, $\Phi_{\chi(s)}(\omega_s(k+1)) = \omega_s(k+1)$. Note that $r_s(k+1) = \omega_s(k+1)$ or $r_s(k+1) = \Psi_{\chi(s)}(\omega_s(k+1))$. In either case, $r_s(k+1)$ is in the range of the $\Phi_{\chi(s)}$ function.

$$\begin{aligned}
\lambda_{il}(k+1) &= \max_{j \in m(i,l)} r_j(k+1) \\
\Phi_i(\lambda_{il}(k+1)) &= \Phi_i\left(\max_{j \in m(i,l)} r_j(k+1)\right) \\
&= \max_{j \in m(i,l)} \Phi_i(r_j(k+1)) \text{ (since } \Phi_i \text{ is a nondecreasing function)} \\
&= \max_{j \in m(i,l)} r_j(k+1) \text{ (since } i = \chi(j) \text{ and} \\
&\qquad\qquad \Phi_{\chi(j)}(r_j(k+1)) = r_j(k+1)) \\
&= \lambda_{il}(k+1)
\end{aligned}$$

It can be proved similarly that $\Phi_i(\Omega_{il}(k+1)) = \Omega_{il}(k+1)$. The result holds for iteration $k+1$. The result follows by induction. □

Observe that for any link l_1 on the path of virtual session j,

$$\begin{aligned}
\Phi_{\chi(j)}\left(\eta_{\chi(j)l_1}(k)\right) &= \max\left(\Phi_{\chi(j)}(\eta_{l_1}(k)), \Phi_{\chi(j)}\left(\lambda_{\chi(j)l_1}(k-1)\right)\right) \\
&\qquad \text{(since } \Phi_{\chi(j)} \text{ is a nondecreasing function)} \\
&= \max\left(\Phi_{\chi(j)}(\eta_{l_1}(k)), \lambda_{\chi(j)l_1}(k-1)\right)
\end{aligned} \tag{2}$$

The last equality follows from Lemma 1. We will use (2) several times. The following lemmas present upperbounds for $w_j(k)$ and $\Omega_{il}(k)$, for all virtual sessions j, sessions i and links l. These bounds will be used several times as well.

Lemma 2 *For all virtual sessions j and iterations k,*

$$\omega_j(k) \leq \max\left(\Phi_{\chi(j)}\left(\eta_l(k)\right), \lambda_{\chi(j)l}(k-1)\right) \text{ if } l \in L_j \tag{3}$$

For all sessions i, links l on the path of i, and iterations k,

$$\Omega_{il}(k) \leq \max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k-1)\right), \tag{4}$$

Proof of Lemma 2: If virtual session $j \notin S(k-1)$, and $l \in L_j$, $\omega_j(k) = r_j(k-1) \leq \max(\Phi_{\chi(j)}\left(\eta_l(k)\right), \lambda_{\chi(j)l}(k-1))$. If $j \in S(k-1)$, and $l \in L_j$

$$\begin{aligned}
\omega_j(k) &= \Phi_{\chi(j)}\left(\min_{l_1 \in L_j} \eta_{\chi(j)l}(k)\right) \\
&= \min_{l_1 \in L_j} \Phi_{\chi(j)}\left(\eta_{\chi(j)l_1}(k)\right) \\
&\qquad \text{(since } \Phi_{\chi(j)} \text{ is a nondecreasing function)} \\
&= \min_{l_1 \in L_j}\left(\max\left(\Phi_{\chi(j)}\left(\eta_{l_1}(k)\right), \lambda_{\chi(j)l_1}(k-1)\right)\right) \text{ (from (2))} \\
&\leq \max\left(\Phi_{\chi(j)}\left(\eta_l(k)\right), \lambda_{\chi(j)l}(k-1)\right)
\end{aligned}$$

Thus (3) follows. Now, (4) follows from (3). □

We will present some relations which we shall use repeatedly.

$$\text{Note that } \Omega_{il}(k) = \lambda_{il}(k-1) \text{ if } i \in n(l) \setminus \Xi_l(k-1) \tag{5}$$

$$\begin{aligned}
C_l &\geq F_l(k-1) + \sum_{i \in \Xi_l(k-1)} \Upsilon_i\left(\max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k-1)\right)\right) \text{ (from (1))} \\
&= \sum_{i \in n(l) \setminus \Xi_l(k-1)} \Upsilon_i\left(\lambda_{il}(k-1)\right) + \sum_{i \in \Xi_l(k-1)} \Upsilon_i\left(\max\left(\Phi_i\left(\eta_l(k)\right)\right), \lambda_{il}(k-1)\right) \\
&\geq \sum_{i \in n(l) \setminus \Xi_l(k-1)} \Upsilon_i\left(\Omega_{il}(k)\right) + \sum_{i \in \Xi_l(k-1)} \Upsilon_i\left(\Omega_{il}(k)\right) \\
&\qquad \text{(from (4) of Lemma 2 and (5) and since } \Upsilon_i \text{ is nondecreasing)} \\
&= \sum_{i \in n(l)} \Upsilon_i\left(\Omega_{il}(k)\right)
\end{aligned} \tag{6}$$

The following lemma proves that if no new virtual session saturates in step (6) of the algorithm in iteration k, then the utility of any unsaturated virtual session j can be increased to the next level from $\omega_j(k)$, if all other virtual sessions have utilities assigned by the utility vector $\vec{\omega}(k)$. The proof follows from the saturation condition for a virtual session. We will use this result to prove several other properties.

Lemma 3 *If $k \geq 1$, $\Lambda(k) = \phi$, then for all virtual sessions $j \in S(k-1)$,*

$$\Upsilon_{\chi(j)}\left(\max(\Omega_{\chi(j)l}(k), \Psi_{\chi(j)}(\omega_j(k)))\right) + \sum_{i \in n(l), i \neq \chi(j)} \Upsilon_i(\Omega_{il}(k)) \leq C_l, \ \forall\, l \in L_j$$

Proof of Lemma 3: Consider any virtual session $j \in S(k-1)$ and a link $l \in L_j$. Since $j \notin \Lambda(k)$, There are three possibilities:

1. $\omega_j(k) < \Omega_{\chi(j)l}(k)$,
2. $\Psi_{\chi(j)}(\omega_j(k)) > \omega_j(k)$ and $\sum_{i \in n(l)} \Upsilon_i(\Omega_{il}(k)) \leq C_l - \left(\Upsilon_{\chi(j)}\left(\Psi_{\chi(j)}(\omega_j(k))\right) - \Upsilon_{\chi(j)}(\omega_j(k))\right)$
3. $\Psi_{\chi(j)}(\omega_j(k)) = \omega_j(k)$ and $\sum_{i \in n(l)} \Upsilon_i(\Omega_{il}(k)) \neq C_l$.

In the first case, $\Psi_{\chi(j)}(\omega_j(k)) \leq \Omega_{\chi(j)l}(k)$, since $\Omega_{\chi(j)l}(k)$ is in the range of $\Phi_{\chi(j)}$ (Lemma 1). Thus $\left(\max(\Omega_{\chi(j)l}(k), \Psi_{\chi(j)}(\omega_j(k))\right) = \Omega_{\chi(j)l}(k)$. Thus,
$\Upsilon_{\chi(j)}\left(\max(\Omega_{\chi(j)l}(k), \Psi_{\chi(j)}(\omega_j(k))\right) + \sum_{i \in n(l), i \neq \chi(j)} \Upsilon_i(\Omega_{il}(k)) = \sum_{i \in n(l)} \Upsilon_i(\Omega_{il}(k))$.

This is because if $j \notin S(k-1)$, and virtual session j traverses through link l, then $\omega_j(k) = r_j(k-1)$ and $j \notin S(k-1)$ for all $j \in m(i,l)$, if $i \in n(l) \backslash \Xi_l(k-1)$. From (4) of Lemma 2, (5), and since Υ_i is increasing, $\sum_{i \in n(l)} \Upsilon_i(\Omega_{il}(k)) \leq \sum_{i \in n(l) \backslash \Xi_l(k-1)} \Upsilon_i(\lambda_{il}(k-1)) + \sum_{i \in \backslash \Xi_l(k-1)} \Upsilon_i(\max(\Phi_i(\eta_l(k)), \lambda_{il}(k-1))) \leq C_l$. The last inequality follows from (1). Hence, the lemma holds in the first case.

Let the first case not hold. Thus $\omega_j(k) = \Omega_{\chi(j)l}(k)$. Now, $\Psi_{\chi(j)l}(\omega_j(k)) \geq \omega_j(k)$. Thus $\max\left(\Omega_{\chi(j)l}(k), \Psi_{\chi(j)}(\omega_j(k))\right) = \Psi_{\chi(j)l}(\omega_j(k))$. In the second case,

$$\begin{aligned}
& \Upsilon_{\chi(j)}\left(\max\left(\Omega_{\chi(j)l}(k), \Psi_{\chi(j)}(\omega_j(k))\right)\right) + \sum_{i \in n(l), i \neq \chi(j)} \Upsilon_i(\Omega_{il}(k)) \\
= \ & \Upsilon_{\chi(j)}\left(\Psi_{\chi(j)}(\omega_j(k))\right) - \Upsilon_{\chi(j)}\left(\Omega_{\chi(j)l}(k)\right) + \sum_{i \in n(l)} \Upsilon_i(\Omega_{il}(k)) \\
= \ & \Upsilon_{\chi(j)}\left(\Psi_{\chi(j)}(\omega_j(k))\right) - \Upsilon_{\chi(j)}(\omega_j(k)) + \sum_{i \in n(l)} \Upsilon_i(\Omega_{il}(k)) \\
\leq \ & C_l \text{ (from assumption)}
\end{aligned}$$

In the third case, $\max(\Omega_{\chi(j)l}(k), \Psi_{\chi(j)}(\omega_j(k))) = \max(\Omega_{\chi(j)l}(k), \omega_j(k))$, since $\Psi_{\chi(j)}(\omega_j(k)) = \omega_j(k)$. Thus $\max(\Omega_{\chi(j)l}(k), \omega_j(k)) = \Omega_{\chi(j)l}(k)$.

$$\Upsilon_{\chi(j)}\left(\max(\Omega_{\chi(j)l}(k), \Psi_{\chi(j)}(\omega_j(k)))\right) + \sum_{i \in n(l), i \neq \chi(j)} \Upsilon_i(\Omega_{il}(k)) = \sum_{i \in n(l)} \Upsilon_i(\Omega_{il}(k)) \tag{7}$$

From (7) and (6) in the third case

$$\Upsilon_i\left(\max(\Omega_{\chi(j)l}(k), \Psi_{\chi(j)}(\omega_j(k)))\right) + \sum_{i \in n(l), i \neq \chi(j)} \Upsilon_i(\Omega_{il}(k)) \leq C_l$$

Thus the lemma holds in all three cases. □

The following lemma shows that the utilities assigned by the maximal fairness algorithm can not decrease in subsequent iterations. The result follows from the steps of the algorithm. We will use this result in proving that the utility allocations at the end of every iteration are feasible(Lemma 10) and a termination result (Lemma 8).

Lemma 4 *For all iterations $k \geq 0$, and virtual sessions s, $r_s(k+1) \geq r_s(k)$ and $\omega_s(k+1) \geq r_s(k)$. For all sessions i and links l, $\lambda_{il}(k+1) \geq \lambda_{il}(k)$ if $k \geq 0$.*

Proof of Lemma 4: If $s \notin S(k)$, $\omega_s(k+1) = r_s(k)$. Let $s \in S(k)$. Consider any link $l \in L_s$.

$$\begin{aligned} \eta_{\chi(s)l}(k+1) &= \max(\eta_l(k+1), \lambda_{\chi(s),l}(k)) \\ &\geq r_s(k) \; \forall \; l \in L_s \qquad (8) \\ &\quad (\text{since } \lambda_{il}(k) = \max_{j \in m(i,l)} r_j(k) \text{ and } s \in m(\chi(s), l), \text{ if } l \in L_s) \\ \omega_s(k+1) &= \Phi_{\chi(s)}\left(\min_{l \in L_s} \eta_{\chi(s)l}(k+1)\right) \\ &\geq \Phi_{\chi(s)}\left(r_s(k)\right) \; (\text{from (8) and since } \Phi_i \text{ is nondecreasing }) \\ &= r_s(k) \; (\text{by Lemma 1}) \end{aligned}$$

$$\text{Thus } \omega_s(k+1) \geq r_s(k) \; \forall \; s$$

Thus $r_s(k+1) \geq \omega_s(k+1) \geq r_s(k)$.

Since $\lambda_{il}(k+1) = \max_{j \in m(i,l)} r_j(k+1)$ and $r_j(k+1) \geq r_j(k)$ $\forall j$, $\lambda_{il}(k+1) \geq \lambda_{il}(k)$. □

Now we show that it is possible to execute this algorithm. We actually show that the set $\{\theta : F_l(k-1) + \sum_{i \in \Xi_l(k-1)} \Upsilon_i\left(\max\left(\Phi_i(\theta), \lambda_{il}(k-1)\right)\right) \leq C_l\}$ is nonempty in every iteration k. The proof is by induction. We first show that this lemma holds for iteration 1. Then we assume that the lemma holds for iterations $1, \ldots, k$ (and hence $\eta_l(1), \ldots \eta_l(k)$ exists), and show that the lemma holds for iteration $k+1$. Now, we use Lemma 3 in the induction step for iteration k. This is justified because under the induction hypothesis, the set $\{\theta : F_l(k-1) +$ $\sum_{i \in \Xi_l(k-1)} \Upsilon_i\left(\max\left(\Phi_i(\theta), \lambda_{il}(k-1)\right)\right) \leq C_l\}$ is nonempty for iteration k. This lemma also shows that the link control parameter does not decrease in subsequent iterations and satisfies all the desirable properties of step (3) of the algorithm. We will use this result to prove a bound on the utilities (Lemma 9), a termination condition (Lemma 7) and a lemma proving a bottleneck condition (Lemma 11).

Lemma 5 *If $k \geq 1$ and the algorithm has not terminated in $k-1$ iterations, then*

$$\exists \; \theta, \; s.t. \; F_l(k-1) + \sum_{i \in \Xi_l(k-1)} \Upsilon_i\left(\max\left(\Phi_i(\theta), \lambda_{il}(k-1)\right)\right) \leq C_l \qquad (9)$$

$$F_l(k-1) + \sum_{i \in \Xi_l(k-1)} \Upsilon_i\left(\max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k-1)\right)\right) \quad \leq \quad C_l \qquad (10)$$

$$\eta_l(k) \geq \max\left(\eta_l(k-1), \min_{i \in \Xi_l(k-1)} \lambda_{il}(k-1)\right) \qquad (11)$$

If $\eta_l(k) < \alpha_l(k)$ *and session* $i \in \Xi_l(k-1)$ *and*
$\lim_{\theta\uparrow\alpha_l(k)} \max\left(\Phi_i(\theta), \lambda_{il}(k-1)\right) < \max\left(\Phi_i\left(\alpha_l(k)\right), \lambda_{il}(k-1)\right)$

$$\left.\begin{array}{lcl} \eta_l(k) & \geq & \lambda_{il}(k-1) \\ (\eta_l(k), \alpha_l(k)) \cap Ran(\Phi_i) & = & \phi \end{array}\right\} \tag{12}$$

Proof of Lemma 5: Let $k=1$. The algorithm can not terminate in 0 iterations. We know that $S(0) \neq \phi$, $F_l(0) = 0$, $\eta_l(0) = 0$, $\lambda_{il}(0) = \mu_{il}$, and $\Xi_l(0) = n(l) \neq \phi$. Since the feasible set of utility vectors is nonempty, $\sum_{i\in n(l)} \Upsilon_i\left(\lambda_{il}(0)\right) \leq C_l$. For $\theta = \min_{j\in n(l)} \lambda_{jl}(0)$, $\max(\Phi_i(\theta), \lambda_{il}(0)) = \lambda_{il}(0)$, since $\Phi_i(\theta) \leq \theta$ by property (2) and thus $\sum_{i\in n(l)} \Upsilon_i\left(\max(\Phi_i(\theta), \lambda_{il}(0))\right) = \sum_{i\in n(l)} \Upsilon_i\left(\lambda_{il}(0)\right)$. Thus there exists θ, such that
$\sum_{i\in n(l)} \Upsilon_i\left(\max(\Phi_i(\theta), \lambda_{il}(0))\right) \leq C_l$. This proves (9) for $k=1$. If $\eta_l(1) = \alpha_l(1)$, then $\sum_{i\in n(l)} \Upsilon_i\left(\max\left(\Phi_i\left(\eta_l(1)\right), \lambda_{il}(0)\right)\right) \leq C_l$. Let $\eta_l(1) < \alpha_l(1)$. If $\sum_{i\in n(l)} \Upsilon_i\left(\max\left(\Phi_i\left(\eta_l(1)\right), \lambda_{il}(0)\right)\right)$
$> C_l$, then since Υ_i and Φ_i are nondecreasing, $\sum_{i\in n(l)} \Upsilon_i\left(\max\left(\Phi_i(\theta), \lambda_{il}(0)\right)\right) \leq C_l$, only if $\eta_l(1) \geq \theta$. Thus $\alpha_l(1) > \eta_l(1)$ is not the desired supremum. This proves (10) by contradiction. Now we show that $\eta_l(1) \geq \max\left(\eta_l(0), \min_{j\in\Xi_l(0)} \lambda_{jl}(0)\right)$. Since $\eta_l(0) = 0$ and $\lambda_{il}(0) \geq 0$ for all i, $\max\left(\eta_l(0), \min_{j\in\Xi_l(0)} \lambda_{jl}(0)\right) = \min_{j\in\Xi_l(0)} \lambda_{jl}(0) = \min_{j\in n(l)} \lambda_{jl}(0)$. The last equality follows since $\Xi_l(0) = n(l)$. Since $\theta = \min_{j\in n(l)} \lambda_{jl}(0)$, satisfies the inequality $\sum_{i\in n(l)} \Upsilon_i\left(\max(\Phi_i(\theta), \lambda_{il}(0))\right) \leq C_l$, $\alpha_l(1) \geq \min_{j\in n(l)} \lambda_{jl}(0)$. If $\eta_l(1) = \alpha_l(1)$, then $\eta_l(1) \geq \min_{j\in n(l)} \lambda_{jl}(0)$. If $\eta_l(1) < \alpha_l(1)$, then $\sum_{i\in n(l)} \Upsilon_i\left(\max(\Phi_i\left(\alpha_l(1)\right), \lambda_{il}(0))\right) > C_l$. Since Φ_i and Υ_i are nondecreasing functions, this means $\alpha_l(1) > \min_{j\in n(l)} \lambda_{jl}(0)$. Thus it is possible to choose $\eta_l(1)$ s.t. $\eta_l(1) \in [\min_{j\in n(l)} \lambda_{jl}(0), \alpha_l(1))$. Thus $\eta_l(1) \geq \max(\eta_l(0), \min_{j\in\Xi_l(0)} \lambda_{jl}(0))$. Thus (11) holds for $k=1$. Now let $\eta_l(1) < \alpha_l(1)$ and $i \in n(l)$ and $\lim_{\theta\uparrow\alpha_l(1)} \max\left(\Phi_i(\theta), \lambda_{il}(0)\right) < \max\left(\Phi_i\left(\alpha_l(1)\right), \lambda_{il}(0)\right)$. Let $\alpha_l(1) \leq \lambda_{il}(0)$. For all $\theta < \alpha_l(1)$, $\Phi_i(\theta) \leq \theta < \alpha_l(1) \leq \lambda_{il}(0)$. Also $\Phi_i\left(\alpha_l(1)\right) \leq \alpha_l(1) \leq \lambda_{il}(0)$. Thus $\lim_{\theta\uparrow\alpha_l(1)} \max\left(\Phi_i(\theta), \lambda_{il}(0)\right) = \lambda_{il}(0)$ and $\max\left(\Phi_i\left(\alpha_l(1)\right), \lambda_{il}(0)\right) = \lambda_{il}(0)$. This contradicts the assumption that
$\lim_{\theta\uparrow\alpha_l(1)} \max\left(\Phi_i(\theta), \lambda_{il}(0)\right) < \max\left(\Phi_i\left(\alpha_l(1)\right), \lambda_{il}(0)\right)$. So $\alpha_l(1) > \lambda_{il}(0)$. It follows that $\eta_l(1)$ can be always chosen such that $\eta_l(1) \in [\lambda_{il}(0), \alpha_l(1))$. First part of (12) follows. Also we have proved that $\alpha_l(1) > \lambda_{il}(0)$. If $\Phi_i\left(\alpha_l(1)\right) < \alpha_l(1)$, then any choice of $\eta_l(1) \in \left(\Phi_i\left(\alpha_l(1)\right), \alpha_l(1)\right)$ satisfies the second part of (12) by property (4). So let $\Phi_i\left(\alpha_l(1)\right) = \alpha_l(1)$. Thus $\Phi_i\left(\alpha_l(1)\right) > \lambda_{il}(0)$. Thus $\max\left(\Phi_i\left(\alpha_l(1)\right), \lambda_{il}(0)\right) = \Phi_i\left(\alpha_l(1)\right) = \alpha_l(1)$. Thus $\lim_{\theta\uparrow\alpha_l(1)} \max\left(\Phi_i(\theta), \lambda_{il}(0)\right) < \alpha_l(1)$. For any $\theta \in (\lambda_{il}(0), \alpha_l(1))$ $(\alpha_l(1) > \lambda_{il}(0))$, $\Phi_i(\theta) \geq \lambda_{il}(0)$, by property (4) and since $\lambda_{il}(0)$ is in the range of the Φ_i function by assumption. It follows that $\lim_{\theta\uparrow\alpha_l(1)} \max\left(\Phi_i(\theta), \lambda_{il}(0)\right) = \lim_{\theta\uparrow\alpha_l(1)} \Phi_i(\theta)$. (Both limits exist since both are bounded and nondecreasing.) It follows that $\lim_{\theta\uparrow\alpha_l(1)} \Phi_i(\theta) < \alpha_l(1)$. Let $\beta < \alpha_l(1)$ and $\beta \in \text{Ran}(\Phi_i)$. By property (3), $\Phi_i(\beta) = \beta$. Thus $\lim_{\theta\uparrow\alpha_l(1)} \Phi_i(\theta) \geq \beta$, since Φ_i is nondecreasing. It follows that $\beta \notin (\lim_{\theta\uparrow\alpha_l(1)} \Phi_i(\theta), \alpha_l(1))$. Thus $\text{Ran}(\Phi_i) \cap (\lim_{\theta\uparrow\alpha_l(1)} \Phi_i(\theta), \alpha_l(1)) = \phi$. Thus any choice of $\eta_l(1) \in (\lim_{\theta\uparrow\alpha_l(1)} \Phi_i(\theta),\ \alpha_l(1))$ satisfies the second part of (12). Thus if $\eta_l(1) < \alpha_l(1)$, then the lemma holds for iteration $k=1$ for any choice of $\eta_l(1) \in (z(1), \alpha)$, where $z(k) = \max\left(\min_{i\in\Xi_l(k-1)\setminus\mathcal{B}(k)} \lambda_{il}(0), \max_{i\in\mathcal{B}(k)} \lambda_{il}(0), \max_{i\in\mathcal{B}_1(k)} \Phi_i\left(\alpha_l(1)\right),\right.$
$\left.\max_{i\in\mathcal{B}_2(k)} \lim_{\theta\uparrow\alpha_l(1)} \Phi_i(\theta)\right)$, $\mathcal{B}(k) = \{i \in \Xi_l(k-1), \lim_{\theta\uparrow\alpha_l(k)} \max\left(\Phi_i(\theta)\ ,\ \lambda_{il}(k-1)\right) < \max\left(\Phi_i\left(\alpha_l(k)\right), \lambda_{il}(k-1)\right)\}$, $\mathcal{B}_1(k) = \{i \in \mathcal{B}(k), \Phi_i\left(\alpha_l(k)\right) < \alpha_l(k)\}$, $\mathcal{B}_2(k) = \mathcal{B}(k) \setminus \mathcal{B}_1(k)$. If $\eta_l(1) = \alpha_l(1)$, then also the lemma

holds for iteration $k = 1$.

Let the lemma hold for iterations $1, \ldots, k$ and $S(k) \neq \phi$. Thus there exists $\eta_l(p)$, $p = 1, \ldots, k$. We will show that the lemma holds for the $k+1$th iteration. We will show that

$$F_l(k) + \sum_{i \in \Xi_l(k)} \Upsilon_i \left(\max \left(\Phi_i \left(\max \left(\eta_l(k), \min_{j \in \Xi_l(k)} \lambda_{jl}(k) \right) \right), \lambda_{il}(k) \right) \right) \leq C_l$$

If $j \notin S(k-1)$, $r_j(k) = \omega_j(k) = r_j(k-1)$. If $l \in L_j$, $r_j(k-1) \leq \lambda_{\chi(j),l}(k-1) \leq \max(\Phi_{\chi(j)}(\eta_l(k)), \lambda_{\chi(j),l}(k-1))$.

$$r_j(k) \leq \max \left(\Phi_{\chi(j)}(\eta_l(k)), \lambda_{\chi(j),l}(k-1) \right) \text{ if } j \notin S(k-1), l \in L_j \tag{13}$$

$r_j(k) \geq \omega_j(k)$ for all virtual sessions j. Consider a link l, s.t. $\Xi_l(k-1) \neq \phi$.

1. Let $r_j(k) = \omega_j(k)$ for all virtual sessions j traversing through link l. If $j \in S(k-1)$, and $l \in L_j$,

$$\begin{aligned} r_j(k) &\leq \max \left(\Phi_{\chi(j)}(\eta_l(k)), \lambda_{\chi(j),l}(k-1) \right) \qquad (14) \\ &\qquad \text{(from (3) of Lemma 2)} \\ \lambda_{il}(k) &\leq \max \left(\Phi_i(\eta_l(k)), \lambda_{il}(k-1) \right), \ \forall\, i \in n(l) \qquad (15) \\ &\qquad \text{(from (13) and (14))} \end{aligned}$$

From (15),

$$\max \left(\Phi_i(\eta_l(k)), \lambda_{il}(k) \right) \leq \max \left(\Phi_i(\eta_l(k)), \lambda_{il}(k-1) \right) \ \forall\, i \in n(l) \tag{16}$$

$$\begin{aligned} F_l(k) &= F_l(k-1) + \sum_{i \in \Xi_l(k-1) \setminus \Xi_l(k)} \Upsilon_i(\lambda_{il}(k)) \\ &\leq F_l(k-1) + \\ &\quad \sum_{i \in \Xi_l(k-1) \setminus \Xi_l(k)} \Upsilon_i \left(\max \left(\Phi_i(\eta_l(k)), \lambda_{il}(k-1) \right) \right) \qquad (17) \\ &\quad \text{(from (15) and since} \Upsilon_i \text{ is nondecreasing)} \end{aligned}$$

$$\begin{aligned} & F_l(k) + \sum_{i \in \Xi_l(k)} \Upsilon_i \left(\max \left(\Phi_i \left(\max \left(\eta_l(k), \min_{j \in \Xi_l(k)} \lambda_{jl}(k) \right) \right), \lambda_{il}(k) \right) \right) \\ = \ & F_l(k) + \sum_{i \in \Xi_l(k)} \Upsilon_i \left(\max \left(\Phi_i(\eta_l(k)), \Phi_i \left(\min_{j \in \Xi_l(k)} \lambda_{jl}(k) \right) \right), \lambda_{il}(k) \right) \\ & \text{(since } \Phi_i \text{ is nondecreasing, } \Phi_i(\max(x,y)) = \max(\Phi_i(x), \Phi_i(y)) \text{)} \\ = \ & F_l(k) + \sum_{i \in \Xi_l(k)} \Upsilon_i \left(\max \left(\Phi_i(\eta_l(k)), \lambda_{il}(k) \right) \right) \\ & \text{(from property (2) and since } \min_{j \in \Xi_l(k)} \lambda_{jl}(k) \leq \lambda_{il}(k) \text{ since } i \in \Xi_l(k) \text{)} \\ \leq \ & F_l(k-1) + \sum_{i \in \Xi_l(k-1) \setminus \Xi_l(k)} \Upsilon_i \left(\max \left(\Phi_i(\eta_l(k)), \lambda_{il}(k-1) \right) \right) + \end{aligned}$$

$$
\begin{aligned}
& \sum_{i \in \Xi_l(k)} \Upsilon_i \left(\max \left(\Phi_i \left(\eta_l(k) \right), \lambda_{il}(k-1) \right) \right) \\
& \text{(from (16), (17) and since } \Upsilon_i \text{ is nondecreasing)} \\
= \; & F_l(k-1) + \sum_{i \in \Xi_l(k-1)} \Upsilon_i \left(\max \left(\Phi_i \left(\eta_l(k) \right), \lambda_{il}(k-1) \right) \right) \\
& \text{(since } \Xi_l(k) \subseteq \Xi_l(k-1) \text{ as } S(k) \subseteq S(k-1)) \\
\leq \; & C_l \text{ (by induction hypothesis)} \qquad (18)
\end{aligned}
$$

2. Now let there exist a virtual session s traversing through link l such that $r_s(k) > \omega_s(k)$, and $\Phi_{\chi(s)} \left(\min_{l_1 \in L_s} \eta_{l_1}(k) \right) < \Phi_{\chi(s)} \left(\eta_l(k) \right)$. Since $r_s(k) > \omega_s(k)$, $r_s(k) = \Psi_{\chi(s)} \left(\omega_s(k) \right)$ and $s \in S(k-1)$. Since $r_s(k) = \Psi_{\chi(s)} \left(\omega_s(k) \right)$, $s \in S(k-1)$ and hence, $\omega_s(k) = \Phi_{\chi(s)} \left(\min_{l_1 \in L_s} \eta_{l_1}(k) \right)$. Thus

$$
\begin{aligned}
\omega_s(k) &= \Phi_{\chi(s)} \left(\min_{l_1 \in L_s} \eta_{l_1}(k) \right) \\
&< \Phi_{\chi(s)} \left(\eta_l(k) \right) \text{ by hypothesis} \\
\text{Thus } \Psi_{\chi(s)} \left(\omega_s(k) \right) &\leq \Phi_{\chi(s)} \left(\eta_l(k) \right) \text{ (by definition of } \Psi_i \text{ functions)} \\
r_s(k) &= \Psi_{\chi(s)} \left(\omega_s(k) \right) \\
&\leq \Phi_{\chi(s)} \left(\eta_l(k) \right) \\
&\leq \max \left(\Phi_{\chi(s)} \left(\eta_l(k) \right), \lambda_{\chi(s)l}(k-1) \right) \qquad (19)
\end{aligned}
$$

Since there exists at most one virtual session j such that $r_j(k) > \omega_j(k)$, $r_j(k) = \omega_j(k)$, $j \neq s$. Consider a virtual session j traversing through link l. Let $j \in S(k-1)$ and $j \neq s$.

$$
\begin{aligned}
\omega_j(k) &\leq \max(\Phi_{\chi(j)} \left(\eta_l(k) \right), \lambda_{\chi(j)l}(k-1)) \text{ (from (3) of Lemma 2)} \\
\text{Thus } r_j(k) &\leq \max \left(\Phi_{\chi(j)} \left(\eta_l(k) \right), \lambda_{\chi(j)l}(k-1) \right) \; \forall \, j \in S(k-1) \setminus \{s\} \qquad (20) \\
\lambda_{il}(k) &\leq \max \left(\Phi_i \left(\eta_l(k) \right), \lambda_{il}(k-1) \right) \; \forall \, i \in n(l) \qquad (21) \\
& \text{(from (13), (19) and (20))}
\end{aligned}
$$

Using (21) it can be proved that

$$
F_l(k) + \sum_{i \in \Xi_l(k)} \max \left(\Phi_i \left(\max \left(\eta_l(k), \min_{j \in \Xi_l(k)} \lambda_{jl}(k) \right) \right), \lambda_{il}(k) \right) \leq C_l \qquad (22)
$$

The proof for (22) is exactly similar to the one for (18) using (15).

3. Now let there exist a virtual session s traversing link l such that $r_s(k) > \omega_s(k)$ and $\Phi_{\chi(s)} \left(\min_{l_1 \in L_s} \eta_{l_1}(k) \right) = \Phi_{\chi(s)} \left(\eta_l(k) \right)$. As before, $r_s(k) = \Psi_{\chi(s)} \left(\omega_s(k) \right)$. First let $r_s(k) \leq \max \left(\Phi_{\chi(s)} \left(\eta_l(k) \right), \lambda_{\chi(s)l}(k-1) \right)$

$$
\begin{aligned}
r_j(k) &= \omega_j(k), j \neq s \\
&\leq \max \left(\Phi_{\chi(s)} \left(\eta_l(k) \right), \lambda_{\chi(j)l}(k-1) \right) \text{ if } j \in S(k-1) \qquad (23) \\
& \text{(from (3) of Lemma 2)} \\
\lambda_{il}(k) &\leq \max \left(\Phi_{\chi(s)} \left(\eta_l(k) \right), \lambda_{il}(k-1) \right) \; \forall i \in n(l) \qquad (24) \\
& \text{(from (13), assumption and (23))}
\end{aligned}
$$

Now using (24) it can be proved that

$$F_l(k) + \sum_{i \in \Xi_l(k)} \max\left(\Phi_i\left(\max\left(\eta_l(k), \min_{j \in \Xi_l(k)} \lambda_{jl}(k)\right)\right), \lambda_{il}(k)\right) \leq C_l \tag{25}$$

The proof is exactly similar to the one for (18) using (15).

$$\text{Now let } r_s(k) > \max\left(\Phi_{\chi(s)}\left(\eta_l(k)\right), \lambda_{\chi(s)l}(k-1)\right) \tag{26}$$

Observe that $\forall i \in \Xi_l(k-1)$,

$$\begin{aligned}
\lambda_{il}(k) &\geq \Omega_{il}(k) \\
&\geq \Phi_i\left(\eta_l(k)\right) \text{ (from the algorithm)}
\end{aligned}$$

The last step follows from the incrementation condition of the algorithm since $r_s(k) > \omega_s(k)$, $l \in l_s$ and $\Phi\chi(s)\left(\min_{l_1 \in L_s} \eta_{l_1}(k)\right) = \Phi\chi(s)\left(\eta_l(k)\right)$.

$$\begin{aligned}
\text{Thus } \max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k)\right) &= \lambda_{il}(k) \ \forall\ i \in n(l) \\
&= \Omega_{il}(k) \text{ if } i \neq \chi(s),
\end{aligned} \tag{27}$$

The last step follows since $r_j(k) = \omega_j(k)\ \forall j \in m(i,l)$ as $i \neq \chi(s)$. Now $r_s(k) > \omega_s(k)$. Thus $s \in S(k-1)$.

$$\begin{aligned}
\omega_s(k) &\leq \max\left(\Phi_{\chi(s)}\left(\eta_l(k)\right), \lambda_{\chi(s)l}(k-1)\right) \\
&\qquad \text{(from (3) of Lemma 2 since } l \in L_s) \\
r_s(k) &\leq \Psi_{\chi(s)}\left(\max\left(\Phi_{\chi(s)}\left(\eta_l(k)\right), \lambda_{\chi(s)l}(k-1)\right)\right) \\
&\qquad \text{(since } \Psi_{\chi(s)}(k) \text{ is nondecreasing)}
\end{aligned} \tag{28}$$

From (26), (28) and since $r_s(k)$ is in the range of the function $\Phi_{\chi(s)}$ and from the definition of the function $\Psi_{\chi(s)}$,

$$r_s(k) = \Psi_{\chi(s)}\left(\max\left(\Phi_{\chi(s)}\left(\eta_l(k)\right), \lambda_{\chi(s)l}(k-1)\right)\right) \tag{29}$$

$$\max\left(\Omega_{\chi(s)l}(k), r_s(k)\right) = \Psi_{\chi(s)}\left(\max\left(\Phi_{\chi(s)}\left(\eta_l(k)\right), \lambda_{\chi(s)l}(k-1)\right)\right) \tag{30}$$
$$\text{(from (4) of Lemma 2 and (29))}$$

$$\begin{aligned}
r_j(k) &= \omega_j(k), j \neq s \\
r_s(k) &= \Psi_{\chi(s)}\left(\omega_s(k)\right) \\
&\geq \omega_s(k)
\end{aligned} \tag{31}$$

$$\begin{aligned}
\text{Thus } \lambda_{\chi(s)l}(k) &= \max\left(\Omega_{\chi(s)l}(k), r_s(k)\right) \\
&= \Psi_{\chi(s)}\left(\max\left(\Phi_{\chi(s)}\left(\eta_l(k)\right), \lambda_{\chi(s)l}(k-1)\right)\right)
\end{aligned} \tag{32}$$

The last inequality follows from (30). Now, from (27), (32) and since $\chi(s) \in n(l)$,

$$\begin{aligned}\max\left(\Phi_{\chi(s)}\left(\eta_l(k)\right), \lambda_{\chi(s)l}(k)\right) &= \Psi_{\chi(s)}\left(\max(\Phi_{\chi(s)}\left(\eta_l(k)\right), \lambda_{\chi(s)l}(k-1)\right) \\ &= \max\left(\Omega_{\chi(s)l}(k), r_s(k)\right) \qquad (33)\end{aligned}$$

The last inequality follows from (30). Thus if $r_s(k) > \max\left(\Phi_{\chi(s)}\left(\eta_l(k)\right),\right.$ $\left.\lambda_{\chi(s)l}(k-1)\right)$,

$$\begin{aligned}
& F_l(k) + \sum_{i \in \Xi_l(k)} \Upsilon_i\left(\max\left(\Phi_i\left(\max\left(\eta_l(k), \min_{j \in \Xi_l(k)} \lambda_{jl}(k)\right)\right), \lambda_{il}(k)\right)\right) \\
= \ & F_l(k) + \sum_{i \in \Xi_l(k)} \Upsilon_i\left(\max\left(\Phi_i\left(\eta_l(k)\right), \Phi_i\left(\min_{j \in \Xi_l(k)} \lambda_{jl}(k)\right), \lambda_{il}(k)\right)\right) \\
& \quad \text{(since } \Phi_i \text{ is nondecreasing, } \Phi_i(\max(x,y)) = \max(\Phi_i(x), \Phi_i(y)) \\
= \ & F_l(k) + \sum_{i \in \Xi_l(k)} \Upsilon_i\left(\max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k)\right)\right) \\
& \quad \text{(from property (2) and since } \min_{j \in \Xi_l(k)} \lambda_{jl}(k) \leq \lambda_{il}(k) \text{ as } i \in \Xi_l(k)) \\
= \ & F_l(k-1) + \sum_{i \in \Xi_l(k-1) \setminus \Xi_l(k)} \Upsilon_i\left(\lambda_{il}(k)\right) + \\
& \sum_{i \in \Xi_l(k)} \Upsilon_i\left(\max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k)\right)\right) \qquad \text{(since } \Xi_l(k) \subseteq \Xi_l(k-1)) \\
\leq \ & F_l(k-1) + \sum_{i \in \Xi_l(k-1) \setminus \Xi_l(k)} \Upsilon_i\left(\max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k)\right)\right) \\
& + \sum_{i \in \Xi_l(k)} \Upsilon_i\left(\max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k)\right)\right) \qquad \text{(since } \Upsilon_i \text{ is nondecreasing)} \\
= \ & F_l(k-1) + \sum_{i \in \Xi_l(k-1)} \Upsilon_i\left(\max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k)\right)\right) \\
& \quad \text{(since } \Xi_l(k-1) \supseteq \Xi_l(k)) \\
= \ & F_l(k-1) + \sum_{i \in \Xi_l(k-1), i \neq \chi(s)} \Upsilon_i\left(\Omega_{il}(k)\right) + \Upsilon_{\chi(s)}\left(\max(\Omega_{\chi(s)l}(k), r_s(k))\right) \\
& \quad \text{(from (27), (33) and since } \chi(s) \in \Xi_l(k-1)) \\
= \ & \sum_{i \in n(l) \setminus \Xi_l(k-1)} \Upsilon_i\left(\Omega_{il}(k)\right) + \sum_{i \in \Xi_l(k-1), i \neq \chi(s)} \Upsilon_i\left(\Omega_{il}(k)\right) \\
& + \Upsilon_{\chi(s)}\left(\max\left(\Omega_{\chi(s)l}(k), r_s(k)\right)\right) \\
& \quad \text{(since } \Omega_{il}(k) = \lambda_{il}(k-1) \forall i \in n(l) \setminus \Xi_l(k-1)) \\
= \ & \sum_{i \in n(l), i \neq \chi(s)} \Upsilon_i\left(\Omega_{il}(k)\right) + \Upsilon_{\chi(s)}\left(\max\left(\Omega_{\chi(s)l}(k), \Psi_{\chi(s)}\left(w_s(k)\right)\right)\right) \\
\leq \ & C_l \qquad (34)
\end{aligned}$$

Note that (34) follows from Lemma 3, since $r_s(k) > w_s(k)$ means $s \in S(k-1)$ and $\Lambda(k) = \phi$.

Since virtual session s traverses link l and $\Phi_{\chi(s)}$ is nondecreasing, $\Phi_{\chi(s)}\left(\min_{l_1 \in L_s} \eta_{l_1}(k)\right) \leq \Phi_{\chi(s)}\left(\eta_l(k)\right)$. Thus cases (2) and (3) are the only possibilities, when $r_s(k) > \omega_s(k)$ for some virtual session s traversing through link l. Case (1) covers the other possibility, i.e., when $r_j(k) = \omega_j(k)$ for all virtual sessions traversing the link l. From (18), (22), (25) and (34), $\theta = \max\left(\eta_l(k), \min_{j \in \Xi_l(k)} \lambda_{jl}(k)\right)$ satisfies $F_l(k) + \sum_{i \in \Xi_l(k)} \max\left(\Phi_i(\theta), \lambda_{il}(k)\right) \leq C_l$. Thus (9) holds for $k+1$. Relations (10), (11) and (12) can be argued in a similar manner as for iteration 1. Hence the result follows from induction. □

5.2 Proof of Generalized Finite-Termination Theorem

Now we will prove that the algorithm terminates in a finite number of iterations (Theorem 1). We prove it using the lemmas in the previous subsection and the following additional lemmas.

The following lemma proves a technical condition. We will use this lemma to show an important termination result (Lemma 7).

Lemma 6 *If $\Xi_l(k-1) \neq \phi$ and $F_l(k-1) + \sum_{i \in \Xi_l(k-1)} \Upsilon_i\left(\max\left(\Phi_i(\theta), \lambda_{il}(k-1)\right)\right) < C_l$, then $\alpha_l(k) > \theta$.*

Proof of Lemma 6: Since Φ_i and Υ_i functions are right continuous, given any $\delta > 0$, there exists an $\epsilon_i > 0$, such that $\Upsilon_i\left(\max\left(\Phi_i(\theta+\epsilon), \lambda_{il}(k-1)\right)\right) \leq \Upsilon_i\left(\max\left(\Phi_i(\theta), \lambda_{il}(k-1)\right)\right) + \delta$. Let $\delta = \frac{C_l - \left(F_l(k-1) + \sum_{i \in \Xi_l(k-1)} \Upsilon_i(\max(\Phi_i(\theta), \lambda_{il}(k-1)))\right)}{|\Xi_l(k-1)|}$, and $\epsilon = \min_{i \in \Xi_l(k-1)} \epsilon_i$. Note that $\epsilon > 0$, as $|\Xi_l(k-1)|$ is finite. Note that $F_l(k-1) + \sum_{i \in \Xi_l(k-1)} \Upsilon_i\left(\max\left(\Phi_i(\theta+\epsilon), \lambda_{il}(k-1)\right)\right) \leq C_l$. By definition, $\alpha_l(k) \geq \theta + \epsilon$. Since $\epsilon > 0$, $\alpha_l(k) > \theta$. □

The following lemma shows that if no virtual session saturates in step (6) of the algorithm, then at least one virtual session meets the condition for utility incrementation in step (8). The proof looks at the link $l_{\min}$ which attains the minimum link control parameter, amongst all those which carry at least one unsaturated virtual session. It chooses an unsaturated session $i_{\min}$ traversing link $l_{\min}$ by certain criteria, and argues that all the unsaturated virtual sessions of $i_{\min}$ satisfy the required properties. The result follows.

Lemma 7 *If $\Lambda(k) = \phi$, there always exists a virtual session $s \in S(k-1)$ such that*

$$\begin{aligned} \omega_s(k) &= \Phi_{\chi(s)}\left(\min_{l \in L_s} \eta_l(k)\right), \\ \Psi_{\chi(s)}\left(\omega_s(k)\right) &> \omega_s(k), \end{aligned}$$

For some $l \in L_s$ s.t. $\Phi_{\chi(s)}\left(\min_{l_1 \in L_s} \eta_{l_1}(k)\right) = \Phi_{\chi(s)}\left(\eta_l(k)\right)$

$$\begin{aligned} \omega_s(k) &= \Omega_{\chi(s)l}(k) \text{ and} \\ Ran(\Phi_{\chi(s)}) \cap \left(\eta_l(k), \alpha_l(k)\right) &= \phi, \end{aligned}$$

and for all $l \in L_s$ s.t. $\Phi_{\chi(s)}\left(\min_{l_1 \in L_s} \eta_{l_1}(k)\right) = \Phi_{\chi(s)}\left(\eta_l(k)\right)$

$$\Omega_{il}(k) = \max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k-1)\right) \ \forall\, i \in \Xi_l(k-1)$$

Proof of Lemma 7: Let $\Lambda(k) = \phi$. Let $l_{\min} = \arg\min_{l \in \cup_{j \in S(k-1)} L_j} \eta_l(k)$. Note that $l_{\min}$ is well defined as $S(k-1) \neq \phi$. If more than one link attain the minimum, choose any one of them as $l_{\min}$. If $\alpha_{l_{\min}}(k) = \eta_{l_{\min}}(k)$, then $i_{\min} = \arg\min_{i \in \Xi_{l_{\min}}(k-1)} \lambda_{il_{\min}}(k-1)^{\ddagger}$. If $\alpha_{l_{\min}}(k) > \eta_{l_{\min}}(k)$, then $F_{l_{\min}}(k-1) + \sum_{i \in \Xi_{l_{\min}}(k-1)} \Upsilon_i \left(\max\left(\Phi_i\left(\alpha_{l_{\min}}(k)\right), \lambda_{il_{\min}}(k-1)\right)\right) > C_{l_{\min}}$, whereas from the definition of $\alpha_l(k)$, and the nondecreasing property of Φ_i and Υ_i, $F_{l_{\min}}(k-1)+\sum_{i \in \Xi_{l_{\min}}(k-1)} \Upsilon_i \left(\max\left(\Phi_i(\theta), \lambda_{il_{\min}}(k-1)\right)\right) \leq C_{l_{\min}}$, for any $\theta < \alpha_{l_{\min}}(k)$. This means that $\lim_{\theta \uparrow \alpha_{l_{\min}}(k)} \max\left(\Phi_i(\theta), \lambda_{il_{\min}}(k-1)\right) < \max\left(\Phi_i\left(\alpha_{l_{\min}}(k)\right), \lambda_{il_{\min}}(k-1)\right)$ for some $i \in \Xi_{l_{\min}}(k-1)$. In this case $i_{\min}$ is one such i.

We first show that $\mathrm{Ran}(\Phi_{i_{\min}}) \cap (\eta_{l_{\min}}(k), \alpha_{l_{\min}}(k)) = \phi$. If $\alpha_{l_{\min}}(k) = \eta_{l_{\min}}(k)$, $(\eta_{l_{\min}}(k), \alpha_{l_{\min}}(k)) = \phi$. Thus $\mathrm{Ran}(\Phi_{i_{\min}}) \cap (\eta_{l_{\min}}(k), \alpha_{l_{\min}}(k)) = \phi$. Otherwise, by Lemma 5 and the choice of $i_{\min}$, $\mathrm{Ran}(\Phi_{i_{\min}}) \cap (\eta_{l_{\min}}(k), \alpha_{l_{\min}}(k)) = \phi$. In either case,

$$\mathrm{Ran}(\Phi_{i_{\min}}) \cap (\eta_{l_{\min}}(k), \alpha_{l_{\min}}(k)) = \phi \tag{35}$$

Also in both cases ($\alpha_{l_{\min}}(k) > \eta_{l_{\min}}(k)$, or $\alpha_{l_{\min}}(k) = \eta_{l_{\min}}(k)$), by Lemma 5 and the definition of $i_{\min}$, $\eta_{l_{\min}}(k) \geq \lambda_{i_{\min}l_{\min}}(k-1)$. Consider a $s \in m(i_{\min}, l_{\min}) \cap S(k-1)$ ($m(i_{\min}, l_{\min}) \cap S(k-1) \neq \phi$ as $i_{\min} \in \Xi_{l_{\min}}(k-1)$.

$$\begin{aligned}
\eta_{i_{\min}l_{\min}}(k) &= \max(\eta_{l_{\min}}(k), \lambda_{i_{\min}l_{\min}}(k-1)) \\
&= \eta_{l_{\min}}(k) \text{ (since } \eta_{l_{\min}}(k) \geq \lambda_{i_{\min}l_{\min}}(k-1)) && (36) \\
&\leq \eta_l(k)\ \forall\ l \in L_s, \text{ (since } s \in m(i_{\min}, l_{\min}) \cap S(k-1), \\
&\qquad \text{and from the definition of } l_{\min}) \\
&\leq \eta_{i_{\min}l}(k)\ \forall\ l \in L_s && (37) \\
\text{Thus } \min_{l \in L_s} \eta_{i_{\min}l}(k) &= \eta_{i_{\min}l_{\min}}(k) \text{ (from } l_{\min} \in L_s \text{ and (37))} \\
\text{Thus } \omega_s(k) &= \Phi_{i_{\min}}\left(\eta_{i_{\min}l_{\min}}(k)\right) \text{ (since } s \in m(i_{\min}, l_{\min}) \cap S(k-1), \\
&\qquad \omega_s(k) = \Phi_{\chi(s)}\left(\min_{l \in L_s} \eta_{i_{\min}l}(k)\right)) && (38) \\
&= \Phi_{i_{\min}}\left(\eta_{l_{\min}}(k)\right) \text{ (from (36))} && (39) \\
&= \Phi_{i_{\min}}\left(\min_{l \in L_s} \eta_l(k)\right) \text{ (since by the definition of } l_{\min}, \\
&\qquad \eta_{l_{\min}}(k) \leq \eta_l(k) \forall l \in L_s \text{ and } l_{\min} \in L_s && (40)
\end{aligned}$$

Note that $\omega_s(k) = \Phi_{i_{\min}}\left(\eta_{i_{\min}l_{\min}}(k)\right) = \max\left(\Phi_{i_{\min}}\left(\eta_{i_{\min}}\right), \lambda_{i_{\min}l_{\min}}(k)\right) \geq \Omega_{i_{\min}l_{\min}}(k)$. The first equality follows from (38), the second equality follows from (2) and the last inequality follows from (4) of Lemma 2. Since $s \in m(i_{\min}, l_{\min})$, it follows that $\omega_s(k) \leq \Omega_{i_{\min}l_{\min}}(k)$.

$$\text{Thus } \omega_s(k) = \Omega_{i_{\min}l_{\min}}(k) \tag{41}$$

As argued before, $\Phi_{i_{\min}}\left(\min_{l \in L_s} \eta_l(k)\right) = \Phi_{i_{\min}}\left(\eta_{l_{\min}}(k)\right)$. Consider any link $l \in L_s$ such that $\Phi_{i_{\min}}\left(\eta_l(k)\right) = \Phi_{i_{\min}}\left(\eta_{l_{\min}}(k)\right)$. Consider a session $i \in \Xi_l(k-1)$.

‡$i_{\min}$ is well defined as $\Xi_{l_{\min}}(k-1) \neq \phi$. If more than one session attain the minimum, choose any one of them as $i_{\min}$.

1. Let $\max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k-1)\right) = \lambda_{il}(k-1)$.

$$\begin{aligned} \Omega_{il}(k) &\geq \lambda_{il}(k-1) \text{ (Lemma 4)} \\ &= \max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k-1)\right) \\ \text{Thus } \Omega_{il}(k) &= \max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k-1)\right) \end{aligned} \tag{42}$$

   The last step follows from from (4) of Lemma 2)

2. Now let $\max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k-1)\right) = \Phi_i\left(\eta_l(k)\right)$. Consider any $j \in m(i,l) \cap S(k-1)$ ($m(i,l) \cap S(k-1) \neq \phi$ as $i \in \Xi_l(k-1)$).

$$\begin{aligned} \Phi_i\left(\eta_{il}(k)\right) &= \max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k-1)\right) \text{ (from (2))} \\ &= \Phi_i\left(\eta_l(k)\right) \\ &= \Phi_i\left(\eta_{l_{\min}}(k)\right) \text{ (from the choice of } l\text{)} \\ &\leq \Phi_i\left(\eta_{l_1}(k)\right) \ \forall\, l_1 \in L_j, \text{ (since } j \in m(i, l_{\min}) \cap S(k-1), \\ &\qquad \text{from the definition of } l_{\min} \text{ and } \Phi_i \text{ is nondecreasing)} \\ &\leq \Phi_i\left(\eta_{il_1}(k)\right) \ \forall\, l_1 \in L_j \text{ (since } \eta_{l_1}(k) \leq \eta_{il_1}(k) \text{ and} \\ &\qquad \Phi_i \text{ is nondecreasing)} \end{aligned} \tag{43}$$

$$\text{Thus } \min_{l_1 \in L_j} \Phi_i\left(\eta_{il_1}(k)\right) = \Phi_i\left(\eta_{il}(k)\right) \text{ (from } l \in L_j \text{ and (43))} \tag{44}$$

$$\begin{aligned} \omega_j(k) &= \Phi_i\left(\min_{l_1 \in L_j} \eta_{il_1}(k)\right) \text{ (since } j \in S(k-1)) \\ &= \min_{l_1 \in L_j} \Phi_i\left(\eta_{il_1}(k)\right) \text{ (since } \Phi_i \text{ is nondecreasing)} \\ &= \Phi_i\left(\eta_{il}(k)\right) \text{ (from (44))} \\ &= \max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k-1)\right) \forall j \ \in m(i,l) \cap S(k-1) \\ &\qquad \text{(from (2))} \end{aligned} \tag{45}$$

   Thus $\Omega_{il}(k) = \max_{j \in m(i, l_{\min})} \omega_j(k) \geq \max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k-1)\right)$, since $m(i,l) \cap S(k-1) \neq \phi$. From (4) of Lemma 2,

$$\Omega_{il}(k) = \max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k-1)\right) \tag{46}$$

$$\begin{aligned} \text{Thus } \Omega_{il}(k) &= \max\left(\Phi_i\left(\eta_l(k)\right), \lambda_{il}(k-1)\right) \forall i \in \Xi_l(k-1), \forall l \in L_s \text{ if} \\ &\qquad \Phi_i\left(\eta_l(k)\right) = \Phi_i\left(\min_{l_1 \in L_s} \eta_{l_1}(k)\right) \end{aligned} \tag{47}$$

This follows from (42) and (46).

Now we show that $\Psi_{i_{\min}}(\omega_s(k)) > \omega_s(k)$. At first let $\eta_{l_{\min}}(k) < \alpha_{l_{min}}(k)$. By (39) and property (4) of the $\Phi_{i_{\min}}$ function, $\mathrm{Ran}(\Phi_{i_{\min}}) \cap (\omega_s(k), \eta_{l_{\min}}(k)] = \phi$. By (35), $\mathrm{Ran}(\Phi_{i_{\min}}) \cap (\eta_{l_{\min}}(k), \alpha_{l_{\min}}(k)) = \phi$.

$$\text{Thus } \mathrm{Ran}(\Phi_{i_{\min}}) \cap (\omega_s(k), \alpha_{l_{\min}}(k)) = \phi. \tag{48}$$

It follows that $\Psi_{i_{\min}}(\omega_s(k)) \geq \alpha_{l_{\min}}(k) > \eta_{l_{\min}}(k) \geq \Phi_{i_{\min}}(\eta_{l_{\min}}(k)) = \omega_s(k)$. The last equality follows from (39). Thus $\Psi_{i_{\min}}(\omega_s(k)) > \omega_s(k)$. Now let $\eta_{l_{\min}}(k) = \alpha_{l_{\min}}(k)$. Let $\Psi_{i_{\min}}(\omega_s(k)) \leq \omega_s(k)$. Clearly $\Psi_{i_{\min}}(\omega_s(k)) \geq \omega_s(k)$. It follows that $\Psi_{i_{\min}}(\omega_s(k)) = \omega_s(k)$. Note that $s \notin \Lambda(k)$ as $\Lambda(k) = \phi$, and $s \in S(k-1)$. Thus s violates a saturation condition of page 16 w.r.t. $\vec{\omega}(k)$. Now, $\omega_s(k) = \Omega_{i_{\min} l_{\min}}(k)$ and $\Psi_{i_{\min}}(\omega_s(k)) = \omega_s(k)$. Thus, it must hold that $\sum_{i \in n(l_{\min})} \Upsilon_i(\Omega_{i l_{\min}}(k)) \neq C_{l_{\min}}$. From (6),

$$\begin{aligned}
C_{l_{\min}} &> \sum_{i \in n(l_{\min})} \Upsilon_i(\Omega_{i l_{\min}}(k)) \\
&= F_{l_{\min}}(k-1) + \sum_{i \in \Xi_{l_{\min}}(k-1)} \Upsilon_i(\Omega_{i l_{\min}}(k)) \\
&\qquad (\text{if } i \in n(l_{\min}) \setminus \Xi_{l_{\min}}(k-1), \Omega_{i l_{\min}}(k) = \lambda_{i l_{\min}}(k-1)) \\
&= F_{l_{\min}}(k-1) + \sum_{i \in \Xi_{l_{\min}}(k-1)} \Upsilon_i(\max(\Phi_i(\eta_{l_{\min}}(k)), \lambda_{i l_{\min}}(k-1))) \qquad (49)
\end{aligned}$$

The last equality follows from (47). By definition of $l_{\min}$, $\Xi_{l_{\min}}(k-1) \neq \phi$. Thus from (49) and Lemma 6, $\alpha_{l_{\min}}(k) > \eta_{l_{\min}}(k)$. However, this contradicts the assumption that $\alpha_{l_{\min}}(k) = \eta_{l_{\min}}(k)$. Thus

$$\Psi_{i_{\min}}(\omega_s(k)) > \omega_s(k) \tag{50}$$

The lemma follows from (35), (40), (41), (47) and (50) and the observation that $i_{\min} = \chi(s)$. □

The following lemma shows that the number of iterations in which there is a utility incrementation in step (8) is upperbounded by $|\mathcal{L}|M^2$. Since Lemma 7 indicates that there is always a utility incrementation if no virtual session saturates in step (6), this means that the number of iterations in which no virtual session saturates in step (6) is at most $|\mathcal{L}|M^2$. The proof uses Lemmas 7 and 4. Interestingly, this result is independent of the nature of the utility functions.

Lemma 8 *Let $T = \{k : \Lambda(k) = \phi\}$. The cardinality of T is at most $|\mathcal{L}|M^2$ where $\mathcal{L}$ is the set of links and M is the number of virtual sessions.*

Proof of Lemma 8: From Lemma 7, for every iteration $k \in T$ there exists at least one virtual session s and link l pair s.t. $l \in L_s$, $s \in S(k-1)$ which satisfies the following properties.

$$\begin{aligned}
\Phi_{\chi(s)}\left(\min_{l_1 \in L_s} \eta_{l_1}(k)\right) &= \Phi_{\chi(s)}(\eta_l(k)) && (51) \\
\omega_s(k) &= \Phi_{\chi(s)}(\eta_l(k)) && (52) \\
\omega_s(k) &= \Omega_{\chi(s)l}(k) && (53) \\
\forall i \in \Xi_l(k-1), \qquad \Omega_{il}(k) &= \max(\Phi_i(\eta_l(k)), \lambda_{il}(k-1)) && (54)
\end{aligned}$$

$$\Psi_s(\omega_s(k)) > \omega_s(k) \quad (55)$$

$$\mathrm{Ran}(\Phi_{\chi(s)}) \cap (\eta_l(k), \alpha_l(k)) = \phi \quad (56)$$

Now $r_s(k) = \Psi_{\chi(s)}(\omega_s(k)) > \omega_s(k)$ for one such virtual session s. We call such a virtual session $s(k)$ and the link $l(k)$. We will show that the pair (j, l_1) can occur for at most $|n(l_1)|$ times in the sequence $\{(s(k), l(k))\}$. There can be at most $|\mathcal{L}|M$ different virtual session, link pairs. Thus $|T| \leq |\mathcal{L}|M \max_{l\in\mathcal{L}} |n(l)| \leq |\mathcal{L}|MN \leq |\mathcal{L}|M^2$, where N is the total number of sessions.

Now we show that the pair (j, l_1) can occur at most $|n(l_1)|$ times in the sequence $\{(s(k), l(k))\}$. Let $(s(k), l(k)) = (s(t), l(t)) = (s, l)$, $k < t$. Since $r_s(k) = \Psi_{\chi(s)}(\omega_s(k))$, it follows from (55) and (52) that $r_s(k) > \omega_s(k) = \Phi_{\chi(s)}(\eta_l(k))$. Also $r_s(k) \leq \omega_s(t) = \Phi_{\chi(s)}(\eta_l(t))$. The first inequality follows from Lemma 4, since $t > k$. The second follows from (52). It follows that $\Phi_{\chi(s)}(\eta_l(t)) > \Phi_{\chi(s)}(\eta_l(k))$. Since $\Phi_{\chi(s)}$ is a nondecreasing function, it follows that $\eta_l(t) > \eta_l(k)$. Since $\Xi_l(t-1) \subseteq \Xi_l(k-1)$, either $\Xi_l(t-1) = \Xi_l(k-1)$ or $\Xi_l(t-1) \subset \Xi_l(k-1)$. Let $\Xi_l(t-1) = \Xi_l(k-1)$. It follows that $F_l(t-1) = F_l(k-1)$.

$$\begin{aligned}
C_l &\geq F_l(t-1) + \sum_{i\in\Xi_l(t-1)} \Upsilon_i\left(\max\left(\Phi_i(\eta_l(t)), \lambda_{il}(t-1)\right)\right) \\
&\qquad \text{(from lemma 5)} \\
&\geq F_l(k-1) + \sum_{i\in\Xi_l(t-1)} \Upsilon_i\left(\max\left(\Phi_i(\eta_l(t)), \lambda_{il}(k-1)\right)\right) \\
&\qquad \text{(since } \lambda_{il}(k-1) \leq \lambda_{il}(t-1) \text{ by Lemma 4)} \\
&= F_l(k-1) + \sum_{i\in\Xi_l(k-1)} \Upsilon_i\left(\max\left(\Phi_i(\eta_l(t)), \lambda_{il}(k-1)\right)\right) \\
&\qquad \text{(since } \Xi_l(k-1) = \Xi_l(t-1)\text{)}
\end{aligned} \quad (57)$$

Thus from definition, $\alpha_l(k) \geq \eta_l(t)$. If $\alpha_l(k) = \eta_l(t)$, then from (57), $F_l(k-1) + \sum_{i\in\Xi_l(k-1)} \Upsilon_i\left(\max\left(\Phi_i(\alpha_l(k)), \lambda_{il}(k-1)\right)\right) \leq C_l$. This means that $\eta_l(k) = \alpha_l(k)$. Thus $\eta_l(k) = \alpha_l(k) = \eta_l(t)$. But, we have shown that $\eta_l(t) > \eta_l(k)$. It follows that $\alpha_l(k) > \eta_l(t)$. Let, if possible, $\Phi_{\chi(s)}(\eta_l(t)) \leq \eta_l(k)$. Since $\Phi_{\chi(s)}(\eta_l(t)) > \Phi_{\chi(s)}(\eta_l(k))$, $\Phi_{\chi(s)}(\eta_l(t)) \in (\Phi_{\chi(s)}(\eta_l(k)), \eta_l(k)]$. This contradicts property (4) of the $\Phi_{\chi(s)}$ function. So $\Phi_{\chi(s)}(\eta_l(t)) > \eta_l(k)$. Since $\Phi_{\chi(s)}(\eta_l(t)) \leq \eta_l(t)$ by property (2) of $\Phi_{\chi(s)}$, $\Phi_{\chi(s)}(\eta_l(t)) \in (\eta_l(k), \eta_l(t)]$. Since $\eta_l(t) < \alpha_l(k)$, $\Phi_{\chi(s)}(\eta_l(t)) \in (\eta_l(k), \alpha_l(k))$. This contradicts (56). This is a contradiction. Thus $\Xi_l(t-1) \neq \Xi_l(k-1)$. Thus $\Xi_l(t-1) \subset \Xi_l(k-1)$. If $\Xi_l(t-1) = \phi$, then $s \notin S(t-1)$ as $l \in L_s$ and $s(t) \neq s$. Thus (s, l) can occur at most $|n(l)|$ times in the sequence. □

Now we prove Theorem 1. The proof uses Lemmas 7 and 8.

Proof of Theorem 1 (Generalized Finite Termination Theorem): Clearly at least one of the following holds for every iteration $k \geq 1$.

1. $\Lambda(k) = \phi$
2. $S(k) \subset S(k-1)$ (proper subset)

Clearly $S(k) \subseteq S(k-1)$ for all $k \geq 1$. So (2) can hold for atmost M iterations, where $S(0) = \{1, \ldots, M\}$. Lemma 8 shows that (1) can hold for at most $|\mathcal{L}|M^2$ iterations. Thus the algorithm must terminate in $M + |\mathcal{L}|M^2$ iterations. □

5.3 Proof of the Generalized Maximal-Fairness Theorem

Now we shall prove that the output of the algorithm is maximally fair (Theorem 2). We will use lemmas in the first subsection (Lemmas 1 to 5) and some additional lemmas stated below. The important steps in this proof are to show that the ouput of every iteration is a feasible utility vector (Lemma 10) and whenever a virtual session saturates, it has a generalized bottleneck link in every subsequent iteration (Lemma 11). We have not introduced the notion of a generalized bottleneck link yet. We will do so while stating lemma 11. Actually, a utility allocation is maximally fair if and only if every virtual session has a generalized bottleneck link. However, we only prove the sufficiency result in this paper. We need only the sufficiency condition for proving Theorem 2. The maximal fairness of the output follows from the generalized-bottleneck lemma (Lemma 12).

We introduce another function below. Let $\psi_i(x)$ be the minimum utility not less than x. More formally, $\psi_i(x)$ is a function defined as follows:

$$\psi_i(x) = \inf_{y \in R} \{y : x \leq y = \Phi_i(z), \text{ for some } z \in R\}$$

For example, if all layers have bandwidth of b units, $\Phi_i(x) = b\lfloor \frac{x}{b} \rfloor$ and $\psi_i(x) = b\lceil \frac{x}{b} \rceil$. Since Φ_i is a nondecreasing function, ψ_i is a nondecreasing function for all x.

The next lemma bounds the utilities allocated to the virtual sessions at the end of every iteration by a function of the link control parameters on their paths. This function ψ is the equivalent of the ceiling function for the uniform bandwidth layer problem. Note that in this case the utility allocated to a virtual session can exceed the link control parameter of any link on its path. However, the result is that it can not exceed the ψ function of a link control parameter, for any link on its path, unless its utility is less than or equal to its minimum session link utility on the link. We will use this lemma in proving an important property of the intermediate rate allocations presented in Lemma 11.

Lemma 9 *If $l \in L_s$ and $r_s(k) > \mu_{\chi(s)l}$, then $r_s(k) \leq \psi_{\chi(s)}(\eta_l(k))$ for all virtual sessions s and all iterations $k \geq 0$.*

Proof of Lemma 9: By initialization, $r_s(0) = \mu_s \leq \mu_{\chi(s)l}$ $\forall l \in L_s$ and all virtual sessions s. Thus the lemma holds by vacuity for $k = 0$. Let the lemma hold for iterations $0, 1, \ldots, k$, $k \geq 0$. We will prove that the lemma holds for the $k+1$th iteration. Let $l \in L_s$. Let $r_s(k+1) > \mu_{\chi(s)l}$. If $s \notin S(k)$, $r_s(k+1) = \omega_s(k+1) = r_s(k)$. Thus $r_s(k) > \mu_{\chi(s)l}$. By induction hypothesis, $r_s(k) \leq \psi_{\chi(s)}(\eta_l(k))$. Thus $r_s(k+1) = r_s(k) \leq \psi_{\chi(s)}(\eta_l(k))$. By Lemma 5, $\eta_l(k) \leq \eta_l(k+1)$. Since $\psi_{\chi(s)}$ is a nondecreasing function, it follows that $\psi_{\chi(s)}(\eta_l(k)) \leq \psi_{\chi(s)}(\eta_l(k+1))$. Thus $r_s(k+1) \leq \psi_{\chi(s)}(\eta_l(k+1))$.

Now let $s \in S(k)$. Let $r_s(k+1) = \omega_s(k+1)$. We need to show that $\omega_s(k+1) \leq \psi_{\chi(s)}(\eta_l(k+1))$. Since $s \in S(k)$, $\omega_s(k+1) \leq \max\left(\Phi_{\chi(s)}(\eta_l(k+1)), \lambda_{\chi(s)l}(k)\right)$, from (3) of Lemma 2.

1. Let $\omega_s(k+1) \leq \lambda_{\chi(s)l}(k)$. Note that $\lambda_{\chi(s)l}(k) \geq \omega_s(k+1) = r_s(k+1) > \mu_{\chi(s)l}$. Since $\lambda_{\chi(s)l}(k) > \mu_{\chi(s)l}$, there exists $j \in m(\chi(s), l)$, such that $r_j(k) = \lambda_{\chi(s)l}(k) > \mu_{\chi(s)l}$. By induction hypothesis, $r_j(k) \leq \psi_{\chi(s)}(\eta_l(k))$. As argued before, using Lemma 5, $\psi_{\chi(s)}(\eta_l(k)) \leq \psi_{\chi(s)}(\eta_l(k+1))$. Thus $\omega_s(k+1) \leq \lambda_{\chi(s)l}(k) = r_j(k) \leq \psi_{\chi(s)}(\eta_l(k+1))$.

2. Let $\omega_s(k+1) \leq \Phi_{\chi(s)}(\eta_l(k+1))$. Now, $\Phi_{\chi(s)}(\eta_l(k+1)) \leq \psi_{\chi(s)}(\eta_l(k+1))$. Hence $\omega_s(k+1) \leq \psi_{\chi(s)}(\eta_l(k+1))$.

It follows that $r_s(k+1) \leq \psi_{\chi(s)}(\eta_l(k+1))$. Now let $r_s(k+1) > \omega_s(k+1)$. (For $s \in S(k)$, the only possibilities are $r_s(k+1) = \omega_s(k+1)$ and $r_s(k+1) > \omega_s(k+1)$.) Thus $\omega_s(k+1) = \Phi_{\chi(s)}(\min_{l_1 \in L_s} \eta_{l_1}(k+1))$. It follows that

$$
\begin{aligned}
\omega_s(k) &\leq \Phi_{\chi(s)}(\eta_l(k+1)) \ \text{(since } l \in L_s \text{ and } \Phi_{\chi(s)} \text{ is a} \\
&\qquad\qquad \text{nondecreasing function)} \qquad (58)\\
r_s(k) &= \Psi_{\chi(s)}(\omega_s(k)) \\
&\leq \Psi_{\chi(s)}\left(\Phi_{\chi(s)}(\eta_l(k+1))\right) \ \text{(from (58) and since } \Psi_{\chi(s)} \\
&\qquad\qquad \text{is nondecreasing)} \\
&\leq \Psi_{\chi(s)}(\eta_l(k+1)) \ \text{(since } \Phi_{\chi(s)}(\eta_l(k+1)) \leq \eta_l(k+1) \\
&\qquad\qquad \text{and } \Psi_{\chi(s)} \text{ is nondecreasing)}
\end{aligned}
$$

Thus the result holds for the $k+1$th iteration. The lemma follows by induction. □

The following lemma shows that the utility allocation at the end of every iteration is feasible. This is argued as follows. Virtual session utilities belong to the respective utility sets (Lemma 1). Utility of a virtual session at the 0th iteration is its minimum utility. Utility of a virtual session does not decrease in subsequent iterations (Lemma 4). Thus minimum utility constraints are satisfied. We show that the capacity constraints hold using Lemma 3 and (6). The formal proof follows. We will use this feasibility result in the proof of Theorem 2.

Lemma 10 *The rate allocation $\vec{r}(k)$ at the end of the kth iteration is feasible, $k \geq 0$.*

Proof of Lemma 10: We prove by induction. Note that $r_s(0) = \mu_s$, for all virtual sessions s. Thus $\vec{r}(0)$ satisfies the minimum utility requirements. Also, $\mu_j \in A_{\chi(j)}$ by assumption for all virtual sessions j. Note that $\lambda_{il}(0) = \mu_{il}$, for all sessions $i \in n(l)$ and all links l. Since the set of feasible rate allocation vectors is nonempty, $\sum_{i \in n(l)} \Upsilon_i(\mu_{il}) \leq C_l$, for all link l. Thus $\vec{r}(0)$ is a feasible utility vector.

Let $\vec{r}(k)$ be feasible. Let the algorithm not terminate in k iterations, i.e., $S(k) \neq \phi$. We will prove that $\vec{r}(k+1)$ is feasible. For all virtual sessions j, $r_j(k+1) \in A_{\chi(j)}$ by Lemma 1. For all virtual sessions s, $r_s(k+1) \geq r_s(k) \geq \mu_s$. The first inequality follows from Lemma 4 and the last from the feasibility of $\vec{r}(k)$. Thus $\vec{r}(k+1)$ satisfies the minimum utility requirements. If $r_j(k+1) = \omega_j(k+1)$, for all virtual sessions j, traversing through link l ($\chi(j) \in n(l)$), then $\lambda_{il}(k+1) = \Omega_{il}(k+1)$, $\forall$

$i \in n(l)$. Thus $\vec{r}(k+1)$ satisfies the capacity condition from (6). If $r_j(k+1) \neq \omega_j(k+1)$, for one or more virtual sessions j traversing through link l, then $r_s(k+1) = \Psi_{\chi(s)}\left(\omega_s(k+1)\right)$, for some virtual session s traversing through link l and $r_j(k+1) = \omega_j(k+1)$, $j \neq s$. Thus $\lambda_{il}(k+1) = \Omega_{il}(k+1)$, $i \neq \chi(s)$ and $\lambda_{\chi(s)l}(k+1) = \max\left(\Psi_{\chi(s)}\left(\omega_s(k+1)\right), \Omega_{\chi(s)l}(k+1)\right)$.

$$\begin{aligned}\sum_{i \in n(l)} \Upsilon_i\left(\lambda_{il}(k+1)\right) &= \sum_{i \in n(l), i \neq \chi(s)} \Upsilon_i\left(\Omega_{il}(k+1)\right) \\ &\quad + \Upsilon_{\chi(s)}\left(\max\left(\Psi_{\chi(s)}\left(\omega_s(k+1)\right), \Omega_{\chi(s)l}(k+1)\right)\right) \\ &\leq C_l\end{aligned}$$

The last inequality follows from Lemma 3 since $r_s(k+1) = \Psi_{\chi(s)}\left(\omega_s(k+1)\right)$, $s \in S(k)$, and hence $\Lambda(k+1) = \phi$. Thus $\vec{r}(k+1)$ satisfies the capacity condition in this case as well. □

We first introduce the notion of a *generalized bottleneck link*. Next, we prove Lemma 11 which states that if a virtual session saturates in an iteration k, then the virtual session has a generalized bottleneck link w.r.t. the utility vectors at the end of all the subsequent iterations. We will use this result in proving Theorem 2. The proof uses lemmas 1, 4, 5, 9 and 10.

A link l is a generalized bottleneck link for a virtual session s w.r.t. a utility allocation $\vec{r}$ if

1. $r_s = \lambda_{\chi(s)l}$,

2. If $r_s < \Psi_{\chi(s)}(r_s)$, then

$$\begin{aligned}& C_l - \sum_{i \in n(l)} \Upsilon_i\left(\lambda_{il}\right) \\ + & \sum_{\substack{i \in n(l), \\ \lambda_{il} > \max\left(\mu_{il}, \psi_i\left(\Psi_{\chi(s)}(r_s)\right)\right)}} \left(\Upsilon_i\left(\lambda_{il}\right) - \Upsilon_i\left(\max\left(\mu_{il}, \psi_i\left(\Psi_{\chi(s)}(r_s)\right)\right)\right)\right) \\ < \ & \Upsilon_{\chi(s)}\left(\Psi_{\chi(s)}(r_s)\right) - \Upsilon_{\chi(s)}(r_s)\end{aligned}$$

   We follow the convention that $\sum_{z \in A} z = 0$, if set $A = \phi$.

3. If $r_s = \Psi_{\chi(s)}(r_s)$, then $\sum_{i \in n(l)} \Upsilon_i\left(\lambda_{il}\right) = C_l$, and for any session i traversing the link ($i \in n(l)$), if $\lambda_{il} > \mu_{il}$ then $\lambda_{il} \leq \psi_i(r_s)$.

Utility allocation $\vec{r}$ is maximally fair iff every virtual session has a generalized bottleneck link, w.r.t. $\vec{r}$. We will prove the sufficiency of this condition in Lemma 12. However, first we prove Lemma 11.

Lemma 11 *If $s \notin S(k)$, $k \geq 0$, then virtual session s has a generalized bottleneck link w.r.t. utility vector, $r(\vec{k})$. Thus there exists a link $l \in L_s$ such that*

1. $r_s(k) = \lambda_{\chi(s)l}(k)$,

2. *If* $r_s(k) < \Psi_{\chi(s)}\left(r_s(k)\right)$, *then*

$$\begin{aligned} & C_l - \sum_{i \in n(l)} \Upsilon_i\left(\lambda_{il}(k)\right) \\ + & \sum_{\substack{i \in n(l), \\ \lambda_{il}(k) > \max\left(\mu_{il}, \psi_i\left(\Psi_{\chi(s)}\left(r_s(k)\right)\right)\right)}} \left(\Upsilon_i\left(\lambda_{il}(k)\right) - \Upsilon_i\left(\max\left(\mu_{il}, \psi_i\left(\Psi_{\chi(s)}\left(r_s(k)\right)\right)\right)\right)\right) \\ < \; & \Upsilon_{\chi(s)}\left(\Psi_{\chi(s)}\left(r_s(k)\right)\right) - \Upsilon_{\chi(s)}\left(r_s(k)\right) \end{aligned}$$

We follow the convention that $\sum_{z \in A} z = 0$, *if set* $A = \phi$.

3. *If* $r_s(k) = \Psi_{\chi(s)}\left(r_s(k)\right)$, *then* $\sum_{i \in n(l)} \Upsilon_i\left(\lambda_{il}(k)\right) = C_l$, *and for any session* i *traversing the link* ($i \in n(l)$), *if* $\lambda_{il}(k) > \mu_{il}$ *then* $\lambda_{il}(k) \leq \psi_i\left(r_s(k)\right)$.

Proof of Lemma 11: Since $S(0) = \{1, \ldots, M\}$, $s \in S(0)$ for all virtual sessions s. Thus the lemma holds by vacuity for $k = 0$. Consider $k > 0$. Let $s \notin S(k)$. There exists t such that $s \in S(t) \setminus S(t+1)$, $t + 1 \leq k$. This implies that $r_s(k+1) = \omega_s(k+1) = r_s(k) = \ldots = r_s(t+1)$.

$$\text{Let } r_s(k) = \Psi_{\chi(s)}\left(r_s(k)\right) \tag{59}$$

Now, (59) implies that

$$r_s(t+1) = \Psi_{\chi(s)}\left(r_s(t+1)\right). \tag{60}$$

Thus since $s \in S(t) \setminus S(t+1)$, there exists a link $l \in L_s$ s.t.

$$r_s(t+1) = \lambda_{\chi(s)l}(t+1) \tag{61}$$

$$\sum_{i \in n(l)} \Upsilon_i\left(\lambda_{il}(t+1)\right) = C_l \tag{62}$$

$$\begin{aligned} \sum_{i \in n(l)} \Upsilon_i\left(\lambda_{il}(k)\right) & \geq \sum_{i \in n(l)} \Upsilon_i\left(\lambda_{il}(t+1)\right) \text{ (from Lemma 4 and since} \\ & \qquad \Upsilon_i \text{ is nondecreasing and since } t+1 \leq k) \\ & = C_l \text{ (from (62))} \end{aligned}$$

$$\text{Thus } \sum_{i \in n(l)} \Upsilon_i\left(\lambda_{il}(k)\right) = C_l \text{ (from feasibility of } \vec{r}(k) \text{ by Lemma 10)} \tag{63}$$

Observe that (63) proves the first part of part (3) of the Lemma. From Lemma 4, $\lambda_{il}(t+1) \leq \lambda_{il}(k)$, since $t + 1 \leq k$. We will show that $\lambda_{il}(t+1) = \lambda_{il}(k)$ $\forall i \in n(l)$. Let there exist $u \in n(l)$ such that $\lambda_{ul}(t+1) < \lambda_{ul}(k)$. Then

$$\begin{aligned} C_l & = \Upsilon_u\left(\lambda_{ul}(t+1)\right) + \sum_{i \in n(l), i \neq u} \Upsilon_u\left(\lambda_{il}(t+1)\right) \text{ (from (62))} \\ & < \Upsilon_u\left(\lambda_{ul}(k)\right) + \sum_{i \in n(l), i \neq u} \Upsilon_i\left(\lambda_{il}(k)\right) \text{ (since } \Upsilon_i \text{ is strictly increasing)} \\ & = \sum_{i \in n(l)} \Upsilon_i\left(\lambda_{il}(k)\right) \end{aligned}$$

Thus $\vec{r}(k)$ is infeasible. This contradicts Lemma 10.

$$\text{Thus } \lambda_{il}(k) = \lambda_{il}(t+1), \forall\ i \in n(l). \tag{64}$$

Since $t+1 \leq k$, $r_s(t+1) \leq r_s(k)$ (Lemma 4). From (61) and (64), $r_s(k) \geq \lambda_{\chi(s)l}(k)$. Clearly from definition $r_s(k) \leq \lambda_{\chi(s)l}(k)$.

$$\text{Thus } r_s(k) = \lambda_{\chi(s)l}(k) \tag{65}$$

Now, (65) proves part (1) of the lemma. Now we shall prove the second part of part (3) of the lemma. First we show that $r_s(t+1) \geq \Phi_{\chi(s)}\left(\eta_l(t+1)\right)$.

1.

$$\text{Let } \lambda_{il}(t+1) \leq \max\left(\Phi_i\left(\eta_l(t+1)\right), \lambda_{il}(t)\right), \forall i \in n(l) \tag{66}$$

$$\begin{aligned}
C_l &= \sum_{i \in n(l) \setminus \{\chi(s)\}} \Upsilon_i\left(\lambda_{il}(t+1)\right) + \Upsilon_{\chi(s)}\left(\lambda_{\chi(s)l}(t+1)\right) \text{ (from (62))} \\
&= \sum_{i \in n(l) \setminus \Xi_l(t)} \Upsilon_i\left(\lambda_{il}(t+1)\right) + \sum_{i \in \Xi_l(t), i \neq \chi(s)} \Upsilon_i\left(\lambda_{il}(t+1)\right) + \\
&\quad \Upsilon_{\chi(s)}\left(\lambda_{\chi(s)l}(t+1)\right) \\
&\quad \text{(since } s \in S(t), \chi(s) \in \Xi_l(t)\text{)} \\
&\leq F_l(t) + \sum_{i \in \Xi_l(t), i \neq \chi(s)} \Upsilon_i\left(\max\left(\Phi_i\left(\eta_l(t+1)\right), \lambda_{il}(t)\right)\right) + \\
&\quad \Upsilon_{\chi(s)}\left(\lambda_{\chi(s)l}(t+1)\right) \text{ (from (66) and since } \Upsilon_i \text{ is nondecreasing)} \\
C_l &\geq F_l(t) + \sum_{i \in \Xi_l(t), i \neq \chi(s)} \Upsilon_i\left(\max\left(\Phi_i\left(\eta_l(t+1)\right), \lambda_{il}(t)\right)\right) + \\
&\quad \Upsilon_{\chi(s)}\left(\max\left(\Phi_{\chi(s)}\left(\eta_l(t+1)\right), \lambda_{\chi(s)l}(t)\right)\right) \text{ (from Lemma 5)}
\end{aligned}$$

$$\begin{aligned}
\text{Thus } \lambda_{\chi(s)l}(t+1) &\geq \max\left(\Phi_{\chi(s)}\left(\eta_l(t+1)\right), \lambda_{\chi(s)l}(t)\right) \\
&\quad \text{(since } \Upsilon_{\chi(s)} \text{ is nondecreasing)} \\
&\geq \Phi_{\chi(s)}\left(\eta_l(t+1)\right) \\
\text{Thus } r_s(t+1) &\geq \Phi_{\chi(s)}\left(\eta_l(t+1)\right) \text{ (from (61))}
\end{aligned} \tag{67}$$

2. Let there exist $i \in n(l)$ such that $\lambda_{il}(t+1) > \max\left(\Phi_i\left(\eta_l(t+1)\right), \lambda_{il}(t)\right)$. Thus $r_j(t+1) > \max\left(\Phi_i\left(\eta_l(t+1)\right), \lambda_{il}(t)\right)$ for at least one $j \in m(i,l)$. From Lemma 2, $\omega_u(t+1) \leq \max\left(\Phi_{\chi(u)}\left(\eta_l(t+1)\right), \lambda_{il}(t)\right)$, if $u \in S(t) \cap m(i,l)$. Also, $\omega_u(t+1) = r_u(t) \leq \max\left(\Phi_u\left(\eta_l(t+1)\right), \lambda_{il}(t)\right)$, if $u \in m(i,l) \setminus S(t)$. It follows that $\omega_u(t+1) \leq \max\left(\Phi_u\left(\eta_l(t+1)\right), \lambda_{il}(t)\right)$, $\forall u \in m(i,l)$. Thus $r_j(t+1) > \omega_j(t+1)$. It follows that $r_j(t+1) = \Psi_{\chi(j)}\left(\omega_j(t+1)\right)$. This means that $j \in S(t)$ and hence $\omega_j(t+1) = \Phi_{\chi(j)}\left(\min_{l_1 \in L_j} \eta_{l_1}(t+1)\right)$. Thus $r_j(t+1) = \Psi_{\chi(j)}\left(\Phi_{\chi(j)}\left(\min_{l_1 \in L_j} \eta_{l_1}(t+1)\right)\right)$. Since $r_j(t+1) > \Phi_{\chi(j)}\left(\eta_l(t+1)\right)$, $\Phi_{\chi(j)}\left(\eta_l(t+1)\right) < \Psi_{\chi(j)}\left(\Phi_{\chi(j)}\left(\min_{l_1 \in L_j} \eta_{l_1}(t+1)\right)\right)$.

Since $l \in L_j$, and $\Phi_{\chi(j)}$ is nondecreasing, $\Phi_{\chi(j)}\left(\min_{l_1 \in L_j} \eta_{l_1}(t+1)\right) \leq \Phi_{\chi(j)}\left(\eta_l(t+1)\right) < \Psi_{\chi(j)}\left(\Phi_{\chi(j)}\left(\min_{l_1 \in L_j} \eta_{l_1}(t+1)\right)\right)$. Thus $\Phi_{\chi(j)}\left(\eta_l(t+1)\right) \in \left[\Phi_{\chi(j)}\left(\min_{l_1 \in L_j} \eta_{l_1}(t+1)\right), \Psi_{\chi(j)}\left(\Phi_{\chi(j)}\left(\min_{l_1 \in L_j} \eta_{l_1}(t+1)\right)\right)\right)$. From the definition of $\Psi_{\chi(j)}$ function, $\Phi_{\chi(j)}\left(\eta_l(t+1)\right) \notin \left(\Phi_{\chi(j)}\left(\min_{l_1 \in L_j} \eta_{l_1}(t+1)\right), \Psi_{\chi(j)}\left(\Phi_{\chi(j)}\left(\min_{l_1 \in L_j} \eta_{l_1}(t+1)\right)\right)\right)$.

$$\text{Thus } \Phi_{\chi(j)}\left(\eta_l(t+1)\right) = \Phi_{\chi(j)}\left(\min_{l_1 \in L_j} \eta_{l_1}(t+1)\right) \tag{68}$$

Since s traverses l and $s \in S(t)$, $\chi(s) \in \Xi_l(t)$. Also, $l \in L_j$. Thus from (68), the fact that $r_j(t) > \omega_j(t)$, and the incrementation condition of the algorithm in step (8), $\Omega_{\chi(s)l}(t+1) \geq \Phi_{\chi(s)}\left(\eta_l(t+1)\right)$. Since $\lambda_{\chi(s)l}(t+1) \geq \Omega_{\chi(s)l}(t+1)$,

$$\begin{aligned} \lambda_{\chi(s)l}(t+1) &\geq \Phi_{\chi(s)}\left(\eta_l(t+1)\right) \\ r_s(t+1) &\geq \Phi_{\chi(s)}\left(\eta_l(t+1)\right) \text{ (from (61))} \end{aligned} \tag{69}$$

In both the above cases,

$$r_s(t+1 \geq \Phi_{\chi(s)}\left(\eta_l(t+1)\right)) \text{ (from (67) and (69))} \tag{70}$$

$$\text{Thus } r_s(t+1) \geq \eta_l(t+1) \tag{71}$$

We argue (71) as follows. Let $r_s(t+1) < \eta_l(t+1)$. From (60), $r_s(t+1) = \Psi_{\chi(s)}\left(r_s(t+1)\right)$. Hence, $\text{Ran}\left(\Phi_{\chi(s)}\right) \cap \left(r_s(t+1), r_s(t+1)+\delta\right) \neq \phi$, $\forall \delta > 0$. Thus $\text{Ran}\left(\Phi_{\chi(s)}\right) \cap \left(r_s(t+1), \eta_l(t+1)\right) \neq \phi$. Since $r_s(t+1) \geq \Phi_{\chi(s)}\left(\eta_l(t+1)\right)$ from (70), $\text{Ran}\left(\Phi_{\chi(s)}\right) \cap \left(\Phi_{\chi(s)}\left(\eta_l(t+1)\right), \eta_l(t+1)\right) \neq \phi$, but this contradicts property (4) of the $\Phi_{\chi(s)}$ function. So $r_s(t+1) \geq \eta_l(t+1)$.

$$\begin{aligned} \psi_i\left(r_s(t+1)\right) &\geq \psi_i\left(\eta_l(t+1)\right) \text{ (from (71) and since } \psi_i \text{ is} \\ &\qquad \text{nondecreasing)} \\ &\geq r_j(t+1) \text{ if } j \in m(i,l) \text{ and } r_j(t+1) > \mu_{\chi(j)l} \\ &\qquad \text{(by Lemma 9)} \end{aligned}$$

$$\text{Thus } \psi_i\left(r_s(t+1)\right) \geq \lambda_{il}(t+1) \text{ if } i \in n(l) \text{ and } \lambda_{il}(t+1) > \mu_{il} \tag{72}$$

$$\text{Thus } \psi_i\left(r_s(k)\right) \geq \lambda_{il}(k) \text{ if } i \in n(l), \lambda_{il}(k) > \mu_{il} \tag{73}$$

Note that (73) follows from (72) since $r_s(t+1) = r_s(k)$ as observed before and $\lambda_{il}(k) = \lambda_{il}(t+1)$, from (64). Thus the second part of part (3) of the lemma holds in this case. So, the lemma holds in this case.

Now let $r_s(k) < \Psi_{\chi(s)}\left(r_s(k)\right)$. Since $r_s(k) = r_s(t+1)$ as observed before, $r_s(t+1) < \Psi_{\chi(s)}\left(r_s(t+1)\right)$. Since $s \in S(t) \setminus S(t+1)$, there exists a link $l \in L_s$ s.t.

$$r_s(t+1) = \lambda_{\chi(s)l}(t+1) \tag{74}$$

and

$$\sum_{i \in n(l)} \Upsilon_i \left(\lambda_{il}(t+1)\right) \quad > \quad C_l - \left(\Upsilon_{\chi(s)}\left(\Psi_{\chi(s)}\left(r_s(t+1)\right)\right) - \Upsilon_{\chi(s)}\left(r_s(t+1)\right)\right) \tag{75}$$

$$\text{Hence, } \Upsilon_{\chi(s)}\left(\Psi_{\chi(s)}\left(\lambda_{\chi(s)l}(t+1)\right)\right) + \sum_{i \in n(l), i \neq \chi(s)} \Upsilon_i \left(\lambda_{il}(t+1)\right) \quad > \quad C_l \tag{76}$$

$$\text{(using (74))}$$

From (76), feasibility of $r(\vec{k})$ (Lemma 10) and since $\lambda_{il}(t+1) \leq \lambda_{il}(k)$ as $t+1 \leq k$ (Lemma 4), $\lambda_{\chi(s)l}(k) < \Psi_{\chi(s)}\left(\lambda_{\chi(s)l}(t+1)\right)$. Also $\lambda_{\chi(s)l}(t+1) \leq \lambda_{\chi(s)l}(k)$. From the definition of the $\Psi_{\chi(s)}$ function, and since $\lambda_{\chi(s)l}(k)$ is in the range of $\Phi_{\chi(s)}$ function (Lemma 1), $\lambda_{\chi(s)l}(k) = \lambda_{\chi(s)l}(t+1)$. As $r_s(t+1) = r_s(k)$, part (1) of the lemma follows from (74). To prove part (2) of the lemma, we first prove that $\psi_i\left(\Psi_{\chi(s)}\left(r_s(t+1)\right)\right) \geq \lambda_{il}(t+1)$, if $i \in n(l)$ and $\lambda_{il}(t+1) > \mu_{il}$. For this we prove below that $\Psi_{\chi(s)}\left(r_s(t+1)\right) \geq \eta_l(t+1)$. Similar to the proof of (70) we will consider two cases separately.

1.
$$\text{Let } \lambda_{il}(t+1) \leq \max\left(\Phi_i\left(\eta_l(t+1)\right), \lambda_{il}(t)\right), \forall i \in n(l) \tag{77}$$

If $i \in n(l) \setminus \Xi_l(t)$, $\lambda_{il}(t+1) = \lambda_{il}(t)$ and
$F_l(t) = \sum_{i \in n(l) \setminus \Xi_l(t)} \Upsilon_i\left(\lambda_{il}(t)\right) = \sum_{i \in n(l) \setminus \Xi_l(t)} \Upsilon_i\left(\lambda_{il}(t+1)\right)$.

$$\begin{aligned}
C_l &< \sum_{i \in n(l) \setminus \{\chi(s)\}} \Upsilon_i\left(\lambda_{il}(t+1)\right) + \Upsilon_{\chi(s)}\left(\Psi_{\chi(s)}\left(\lambda_{\chi(s)l}(t+1)\right)\right) \\
&\qquad \text{(from (76))} \\
&= \sum_{i \in n(l) \setminus \Xi_l(t)} \Upsilon_i\left(\lambda_{il}(t+1)\right) + \sum_{i \in \Xi_l(t), i \neq \chi(s)} \Upsilon_i\left(\lambda_{il}(t+1)\right) + \\
&\quad \Upsilon_{\chi(s)}\left(\Psi_{\chi(s)}\left(\lambda_{\chi(s)l}(t+1)\right)\right) \ \text{(since } s \in S(t), \chi(s) \in \Xi_l(t)\text{)} \\
&\leq F_l(t) + \sum_{i \in \Xi_l(t), i \neq \chi(s)} \Upsilon_i\left(\max\left(\Phi_i\left(\eta_l(t+1)\right), \lambda_{il}(t)\right)\right) + \\
&\quad \Upsilon_{\chi(s)}\left(\Psi_{\chi(s)}\left(\lambda_{\chi(s)l}(t+1)\right)\right) \ \text{(from (77))} \\
C_l &\geq F_l(t) + \sum_{i \in \Xi_l(t), i \neq \chi(s)} \Upsilon_i\left(\max\left(\Phi_i\left(\eta_l(t+1)\right), \lambda_{il}(t)\right)\right) + \\
&\quad \Upsilon_{\chi(s)}\left(\max\left(\Phi_{\chi(s)}\left(\eta_l(t+1)\right), \lambda_{\chi(s)l}(t)\right)\right) \text{(from Lemma 5)}
\end{aligned}$$

$$\begin{aligned}
\text{Thus } \Psi_{\chi(s)}\left(\lambda_{\chi(s)l}(t+1)\right) &> \max\left(\Phi_{\chi(s)}\left(\eta_l(t+1)\right), \lambda_{\chi(s)l}(t)\right) \\
&\qquad \text{(since } \Upsilon_{\chi(s)} \text{ is strictly increasing)} \\
&\geq \Phi_{\chi(s)}\left(\eta_l(t+1)\right) \\
\text{Thus } \Psi_{\chi(s)}\left(r_s(t+1)\right) &> \Phi_{\chi(s)}\left(\eta_l(t+1)\right) \ \text{(from (74))} && (78) \\
\text{Thus } \Psi_{\chi(s)}\left(r_s(t+1)\right) &\geq \eta_l(t+1) && (79)
\end{aligned}$$

Note that (79) can be argued similar to (71). Let $\Psi_{\chi(s)}\left(r_s(t+1)\right) < \eta_l(t+1)$. From the definition of Ψ_i function there exists a y, $y \in \left[\Psi_{\chi(s)}\left(r_s(t+1)\right), \Psi_{\chi(s)}\left(r_s(t+1)\right)+\delta\right)$, given any $\delta > 0$, such that $y = \Phi_{\chi(s)}(z)$ for some z. It follows that there exists one such y in the interval $\left[\Psi_{\chi(s)}\left(r_s(t+1)\right), \eta_l(t+1)\right)$. From (78), $y > \Phi_{\chi(s)}\left(\eta_l(t+1)\right)$. It follows that $y \in \left(\Phi_{\chi(s)}\left(\eta_l(t+1)\right), \eta_l(t+1)\right)$ which contradicts property (4) of $\Phi_{\chi(s)}$ function.

2. Let there exist $i \in n(l)$ such that $\lambda_{il}(t+1) > \max\left(\Phi_i\left(\eta_l(t+1)\right), \lambda_{il}(t)\right)$. Using the same arguement as that for the same assumption in the previous case, i.e., when $r_s(k) = \Psi_{\chi(s)}\left(r_s(k)\right)$, it can be shown that $r_s(t+1) \geq \Phi_{\chi(s)}\left(\eta_l(t+1)\right)$. Now we argue that $\Psi_{\chi(s)}\left(r_s(t+1)\right) \geq \eta_l(t+1)$. Note that $r_s(t+1)$ is in the range of the $\Phi_{\chi(s)}$ function (Lemma 1). If $r_s(t+1) < \eta_l(t+1)$, then $r_s(t+1) \in \left[\Phi_{\chi(s)}\left(\eta_l(t+1)\right), \eta_l(t+1)\right)$ since $r_s(t+1) \geq \Phi_{\chi(s)}\left(\eta_l(t+1)\right)$. We know that it is not possible to have $x \in \left(\Phi_{\chi(s)}\left(\eta_l(t+1)\right), \eta_l(t+1)\right]$ if x is in the range of the $\Phi_{\chi(s)}$ function. Thus $r_s(t+1) = \Phi_{\chi(s)}\left(\eta_l(t+1)\right)$. There does not exist any x in the range of $\Phi_{\chi(s)}$ such that $x \in (r_s(t+1), \eta_l(t+1)]$ as $r_s(t+1) = \Phi_{\chi(s)}\left(\eta_l(t+1)\right)$. Now, $\Psi_{\chi(s)}\left(r_s(t+1)\right) \geq \eta_l(t+1)$ follows from the definition of the $\Psi_{\chi(s)}$ function. If $r_s(t+1) \geq \eta_l(t+1)$, then $\Psi_{\chi(s)}\left(r_s(t+1)\right) \geq \eta_l(t+1)$ from the definition of the $\Psi_{\chi(s)}$ function.

It follows from the above two cases that

$$\begin{aligned} \Psi_{\chi(s)}\left(r_s(t+1)\right) &\geq \eta_l(t+1) && (80)\\ \text{Thus } \psi_i\left(\Psi_{\chi(s)}\left(r_s(t+1)\right)\right) &\geq \lambda_{il}(t+1) \text{ if } i \in n(l) \text{ and } \lambda_{il}(t+1) > \mu_{il} && (81) \end{aligned}$$

Note that (81) can be proved using (80). The proof is similar to that of (72) using (71). It follows from (81) that

$$\begin{aligned} \text{If } \mu_{il} \geq \psi_i\left(\Psi_{\chi(s)}\left(r_s(t+1)\right)\right), \text{ then } \lambda_{il}(t+1) &= \mu_{il} && (82)\\ \text{If } \mu_{il} < \psi_i\left(\Psi_{\chi(s)}\left(r_s(t+1)\right)\right), \text{ then } \lambda_{il}(t+1) &\leq \psi_i\left(\Psi_{\chi(s)}\left(r_s(t+1)\right)\right) && (83) \end{aligned}$$

We will prove part (2) of the lemma using (82) and (83).

$$\begin{aligned} &\sum_{\substack{i\in n(l),\\ \lambda_{il}(k)>\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}(r_s(k))\right)\right)}} \left(\Upsilon_i\left(\lambda_{il}(k)\right) - \Upsilon_i\left(\max\left(\mu_{il}, \psi_i\left(\Psi_{\chi(s)}\left(r_s(k)\right)\right)\right)\right)\right)\\ =& \sum_{\substack{i\in n(l),\\ \lambda_{il}(k)>\mu_{il},\\ \psi_i\left(\Psi_{\chi(s)}(r_s(k))\right)\leq\mu_{il}}} \left(\Upsilon_i\left(\lambda_{il}(k)\right) - \Upsilon_i\left(\mu_{il}\right)\right)\\ &+ \sum_{\substack{i\in n(l),\\ \lambda_{il}(k)>\psi_i\left(\Psi_{\chi(s)}(r_s(k))\right),\\ \psi_i\left(\Psi_{\chi(s)}(r_s(k))\right)>\mu_{il}}} \left(\Upsilon_i\left(\lambda_{il}(k)\right) - \Upsilon_i\left(\psi_i\left(\Psi_{\chi(s)}\left(r_s(k)\right)\right)\right)\right)\\ \leq& \sum_{\substack{i\in n(l),\\ \lambda_{il}(k)>\mu_{il},\\ \psi_i\left(\Psi_{\chi(s)}(r_s(k))\right)\leq\mu_{il}}} \left(\Upsilon_i\left(\lambda_{il}(k)\right) - \Upsilon_i\left(\lambda_{il}(t+1)\right)\right) \end{aligned}$$

$$
\begin{aligned}
&+ \sum_{\substack{i \in n(l), \\ \lambda_{il}(k) > \psi_i\left(\Psi_{\chi(s)}(r_s(k))\right), \\ \psi_i\left(\Psi_{\chi(s)}(r_s(k))\right) > \mu_{il}}} \left(\Upsilon_i\left(\lambda_{il}(k)\right) - \Upsilon_i\left(\lambda_{il}(t+1)\right)\right) \\
&\qquad \text{(from (82) and (83) and since } \Upsilon_i \text{ is nondecreasing)} \\
= & \sum_{\substack{i \in n(l), \\ \lambda_{il}(k) > \max\left(\mu_{il}, \psi_i\left(\Psi_{\chi(s)}(r_s(k))\right)\right)}} \left(\Upsilon_i\left(\lambda_{il}(k)\right) - \Upsilon_i\left(\lambda_{il}(t+1)\right)\right) \\
\leq & \sum_{i \in n(l)} \left(\Upsilon_i\left(\lambda_{il}(k)\right) - \Upsilon_i\left(\lambda_{il}(t+1)\right)\right) \text{ (since } \Upsilon_i \text{ is increasing } \forall\ i, \\
&\qquad \text{and } \lambda_{il}(k) \geq \lambda_{il}(t+1) \text{ as } k \geq t+1) \qquad (84)
\end{aligned}
$$

From (84)

$$
\begin{aligned}
& C_l - \sum_{i \in n(l)} \Upsilon_i\left(\lambda_{il}(k)\right) \\
& + \sum_{\substack{i \in n(l), \\ \lambda_{il}(k) > \max\left(\mu_{il}, \psi_i\left(\Psi_{\chi(s)}(r_s(k))\right)\right)}} \left(\Upsilon_i\left(\lambda_{il}(k)\right) - \Upsilon_i\left(\max\left(\mu_{il}, \psi_i\left(\Psi_{\chi(s)}\left(r_s(k)\right)\right)\right)\right)\right) \\
\leq\ & C_l - \sum_{i \in n(l)} \Upsilon_i\left(\lambda_{il}(k)\right) + \sum_{i \in n(l)} \left(\Upsilon_i\left(\lambda_{il}(k)\right) - \Upsilon_i\left(\lambda_{il}(t+1)\right)\right) \\
=\ & C_l - \sum_{i \in n(l)} \Upsilon_i\left(\lambda_{il}(t+1)\right) \\
<\ & \Upsilon_{\chi(s)}\left(\Psi_{\chi(s)}\left(r_s(t+1)\right)\right) - \Upsilon_{\chi(s)}\left(r_s(t+1)\right) \text{ (by (75))} \\
=\ & \Upsilon_{\chi(s)}\left(\Psi_{\chi(s)}\left(r_s(k)\right)\right) - \Upsilon_{\chi(s)}\left(r_s(k)\right) \qquad (85)
\end{aligned}
$$

Thus part (2) of the lemma holds in this case. □

The following lemma shows that if every virtual session has a generalized bottleneck link in a utility allocation, then the utility allocation is maximally fair. As we mentioned earlier, the converse is also true. We prove only the forward part here. The reverse part follows from the observation that if no link on the path of a virtual session s is a generalized bottleneck link, then we can increment the utility of the virtual session s, without decreasing the utility of any other virtual session to a value below the new utility of virtual session s. We will use this lemma in proving Theorem 2.

Lemma 12 *A feasible utility allocation vector is maximally fair if every virtual session s has a generalized bottleneck link under the utility vector.*

Proof of Lemma 12: Let $\vec{r}^1$ be a feasible rate vector. Let λ_{il}^1 denote the corresponding session link bandwidths. Let every virtual session have a generalized bottleneck link under $\vec{r}^1$. Let there exist a feasible rate allocation vector, $\vec{r}^2$ fairer than $\vec{r}^1$. Define the set τ of virtual sessions as follows. Let $\tau = \{j : r_j^1 \neq r_j^2\}$. Since $\vec{r}^2$ is fairer than $\vec{r}^1$, there exists a virtual session s such that $r_s^2 = \min_{j\in\tau} r_j^2$ and $r_s^2 > r_s^1$. This follows from [15]. Let $r_s^1 < \Psi_{\chi(s)}\left(r_s^1\right)$. From assumption, there exists a link $l \in L_s$, which satisfies the following properties w.r.t. virtual session s.

$$r_s^1 = \lambda^1_{\chi(s)l} \tag{86}$$

$$\begin{aligned} & C_l - \sum_{i\in n(l)} \Upsilon_i\left(\lambda^1_{il}\right) \\ & + \sum_{\substack{i\in n(l),\\ \lambda^1_{il}>\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right)\right)}} \left(\Upsilon_i\left(\lambda^1_{il}\right) - \Upsilon_i\left(\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right)\right)\right)\right) \\ < \; & \Upsilon_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right) - \Upsilon_i\left(r_s^1\right) \end{aligned} \tag{87}$$

We follow the convention that $\sum_{z\in A} z$ is 0 if set $A = \phi$. Let λ^2_{il} be the session link bandwidths corresponding to $\vec{r}^2$.

$$\begin{aligned} & \left(\Upsilon_{\chi(s)}\left(r_s^2\right) - \Upsilon_{\chi(s)}\left(r_s^1\right)\right) - \left(\sum_{\substack{i\in n(l),\\ \lambda^2_{il}<\lambda^1_{il}}} \left(\Upsilon_i\left(\lambda^1_{il}\right) - \Upsilon_i\left(\lambda^2_{il}\right)\right)\right) \\ \leq \; & \left(\Upsilon_{\chi(s)}\left(\lambda^2_{\chi(s)l}\right) - \Upsilon_{\chi(s)}\left(\lambda^1_{\chi(s)l}\right)\right) - \left(\sum_{\substack{i\in n(l),\\ \lambda^2_{il}<\lambda^1_{il}}} \left(\Upsilon_i\left(\lambda^1_{il}\right) - \Upsilon_i\left(\lambda^2_{il}\right)\right)\right) \\ & \text{(since } \lambda^2_{\chi(s)l} \geq r_s^2 \text{ and } \lambda^1_{\chi(s)l} = r_s^1 \text{ by (86))} \\ \leq \; & \left(\sum_{\substack{i\in n(l),\\ \lambda^2_{il}\geq\lambda^1_{il}}} \left(\Upsilon_i\left(\lambda^2_{il}\right) - \Upsilon_i\left(\lambda^1_{il}\right)\right)\right) - \left(\sum_{\substack{i\in n(l),\\ \lambda^2_{il}<\lambda^1_{il}}} \left(\Upsilon_i\left(\lambda^1_{il}\right) - \Upsilon_i\left(\lambda^2_{il}\right)\right)\right) \\ & \text{(since } \lambda^2_{\chi(s)l} \geq r_s^2 > r_s^1 = \lambda^1_{\chi(s)l}\text{)} \\ = \; & \left(\sum_{i\in n(l)} \Upsilon_i\left(\lambda^2_{il}\right)\right) - \left(\sum_{i\in n(l)} \Upsilon_i\left(\lambda^1_{il}\right)\right) \\ \leq \; & C_l - \left(\sum_{i\in n(l)} \Upsilon_i\left(\lambda^1_{il}\right)\right) \text{ (from feasibility of } \vec{r}^2\text{)} \\ < \; & \left(\Upsilon_{\chi(s)}\left(\Psi_{\chi(s)}\left(r_s^1\right)\right) - \Upsilon_{\chi(s)}\left(r_s^1\right)\right) \\ & - \sum_{\substack{i\in n(l),\\ \lambda^1_{il}>\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right)\right)}} \left(\Upsilon_i\left(\lambda^1_{il}\right) - \Upsilon_i\left(\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right)\right)\right)\right) \end{aligned} \tag{88}$$

The last inequality follows from (87). From (88),

$$\Upsilon_{\chi(s)}\left(r_s^2\right) - \Upsilon_{\chi(s)}\left(\Psi_{\chi(s)}\left(r_s^1\right)\right) - \left(\sum_{\substack{i\in n(l),\\ \lambda_{il}^2<\lambda_{il}^1}}\left(\Upsilon_i\left(\lambda_{il}^1\right) - \Upsilon_i\left(\lambda_{il}^2\right)\right)\right)$$
$$< \; - \sum_{\substack{i\in n(l),\\ \lambda_{il}^1>\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}(r_s^1)\right)\right)}}\left(\Upsilon_i\left(\lambda_{il}^1\right) - \Upsilon_i\left(\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right)\right)\right)\right) \tag{89}$$

Now, r_s^2 is in the range of $\Phi_{\chi(s)}$ from the feasibility of $\vec{r}^2$. Thus since $r_s^2 > r_s^1$, $r_s^2 \geq \Psi_{\chi(s)}\left(r_s^1\right)$.

$$\text{Thus}\left(\sum_{\substack{i\in n(l),\\ \lambda_{il}^2<\lambda_{il}^1}}\left(\Upsilon_i\left(\lambda_{il}^1\right) - \Upsilon_i\left(\lambda_{il}^2\right)\right)\right)$$
$$> \sum_{\substack{i\in n(l),\\ \lambda_{il}^1>\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}(r_s^1)\right)\right)}}\left(\Upsilon_i\left(\lambda_{il}^1\right) - \Upsilon_i\left(\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right)\right)\right)\right)$$
$$= \sum_{\substack{i\in n(l),\\ \lambda_{il}^1\geq\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}(r_s^1)\right)\right)}}\left(\Upsilon_i\left(\lambda_{il}^1\right) - \Upsilon_i\left(\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right)\right)\right)\right) \tag{90}$$

Consider a i s.t. $\lambda_{il}^2 < \lambda_{il}^1$. There exists a virtual session $j \in m(i,l)$ such that $r_j^1 = \lambda_{il}^1$. $r_j^1 > \lambda_{il}^2 \geq r_j^2$. Thus $j \in \tau$. It follows that $r_j^2 \geq r_s^2$. As argued before, $r_s^2 \geq \Psi_{\chi(s)}\left(r_s^1\right)$. Thus $r_j^2 \geq \Psi_{\chi(s)}\left(r_s^1\right)$. Since r_j^2 is in the range of the function $\Phi_{\chi(j)}$, from the feasibility of $\vec{r}^2$, it follows from the definition of the $\psi_{\chi(j)}$ function that $r_j^2 = \psi_{\chi(j)}\left(r_j^2\right)$. Hence since $\psi_{\chi(j)}$ is a nondecreasing function, $r_j^2 \geq \psi_{\chi(j)}\left(\Psi_{\chi(s)}\left(r_s^1\right)\right)$. Thus $\lambda_{il}^2 \geq \psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right)$ since $j \in m(i,l)$. From feasibility of $\vec{r}^2$, $\lambda_{il}^2 \geq \mu_{il}$. Thus $\lambda_{il}^2 \geq \max\left(\psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right), \mu_{il}\right)$. So, if $\lambda_{il}^2 < \lambda_{il}^1$, $\lambda_{il}^2 \geq \max\left(\psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right), \mu_{il}\right)$ and hence $\lambda_{il}^2 < \lambda_{il}^1$, implies that $\lambda_{il}^1 > \max\left(\psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right), \mu_{il}\right)$. It follows that

$$\sum_{\substack{i\in n(l),\\ \lambda_{il}^1\geq\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}(r_s^1)\right)\right)}}\left(\Upsilon_i\left(\lambda_{il}^1\right) - \Upsilon_i\left(\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right)\right)\right)\right)$$
$$\geq \sum_{\substack{i\in n(l),\\ \lambda_{il}^2<\lambda_{il}^1}}\left(\Upsilon_i\left(\lambda_{il}^1\right) - \Upsilon_i\left(\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right)\right)\right)\right) \tag{91}$$

From (90) and (91)

$$\sum_{\substack{i\in n(l),\\ \lambda_{il}^2<\lambda_{il}^1}}\left(\Upsilon_i\left(\lambda_{il}^1\right) - \Upsilon_i\left(\lambda_{il}^2\right)\right) \;>\; \sum_{\substack{i\in n(l),\\ \lambda_{il}^2<\lambda_{il}^1}}\left(\Upsilon_i\left(\lambda_{il}^1\right) - \Upsilon_i\left(\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right)\right)\right)\right)$$

Thus there exists at least one i such that $\Upsilon_i\left(\lambda_{il}^1\right) - \Upsilon_i\left(\lambda_{il}^2\right) > \Upsilon_i\left(\lambda_{il}^1\right) -$
$\Upsilon_i\left(\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right)\right)\right)$ and $\lambda_{il}^2 < \lambda_{il}^1$. For this i, $\Upsilon_i\left(\lambda_{il}^2\right) < \Upsilon_i\left(\max\left(\mu_{il},\psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right)\right)\right)$.

This means that $\lambda_{il}^2 < \max\left(\mu_{il}, \psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right)\right)$. Since $\lambda_{il}^2 \geq \mu_{il}$, from the feasibility of $\vec{r}^2$, $\lambda_{il}^2 < \psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right)$. Since $\Psi_{\chi(s)}\left(r_s^1\right) \leq r_s^2$, and ψ_i is a nondecreasing function, $\psi_i\left(\Psi_{\chi(s)}\left(r_s^1\right)\right) \leq \psi_i\left(r_s^2\right)$, and thus $\lambda_{il}^2 < \psi_i\left(r_s^2\right)$. Since λ_{il}^2 is in the range of the Φ_i function, from the definition of the ψ_i function, $\lambda_{il}^2 < r_s^2$. Since $j \in m(i,l)$, $r_j^2 \leq \lambda_{il}^2 < r_s^2$. We had argued that $j \in \tau$. Thus $r_j^2 < r_s^2$, contradicts the minimality of r_s^2, $s \in \tau$. Thus there can not exist a feasible vector $\vec{r}^2$, fairer than $\vec{r}^1$.

Now let $r_s^1 = \Psi_{\chi(s)}\left(r_s^1\right)$. From assumption, there exists a link $l \in L_s$, which satisfies the following properties w.r.t. virtual session s.

$$
\begin{aligned}
r_s^1 &= \lambda_{\chi(s)l}^1 && (92)\\
\sum_{i \in n(l)} \Upsilon_i\left(\lambda_{il}^1\right) &= C_l && (93)\\
\lambda_{il}^1 &\leq \psi_i\left(r_s^1\right) \text{ if } i \in n(l) \text{ and } \lambda_{il}^1 > \mu_{il} && (94)
\end{aligned}
$$

$\lambda_{\chi(s)l}^2 \geq r_s^2 > r_s^1$.

$$
\begin{aligned}
& \text{Thus } \Upsilon_{\chi(s)}\left(\lambda_{\chi(s)l}^2\right) + \sum_{i \in n(l), i \neq \chi(s)} \Upsilon_i\left(\lambda_{il}^1\right) \\
> \; & \Upsilon_{\chi(s)}\left(r_s^1\right) + \sum_{i \in n(l), i \neq \chi(s)} \Upsilon_i\left(\lambda_{il}^1\right) \text{ (since } \Upsilon_{\chi(s)} \text{ is strictly increasing)} \\
= \; & \Upsilon_{\chi(s)}\left(\lambda_{\chi(s)l}^1\right) + \sum_{i \in n(l), i \neq \chi(s)} \Upsilon_i\left(\lambda_{il}^1\right) \text{ (from (92))} \\
= \; & C_l \text{ (from (93))}
\end{aligned}
$$

So, from the feasibility of $\vec{r}^2$, $\lambda_{il}^2 < \lambda_{il}^1$, for some $i \in n(l) \setminus \{\chi(s)\}$. From feasibility of $\vec{r}^2$, $\mu_{il} \leq \lambda_{il}^2$. It follows that $\lambda_{il}^1 > \mu_{il}$. From (94), $\lambda_{il}^1 \leq \psi_i\left(r_s^1\right)$. There exists a virtual session $j \in m(i,l)$ such that $r_j^1 = \lambda_{il}^1$. It follows that $r_j^1 \leq \psi_i\left(r_s^1\right)$. Since $r_s^2 > r_s^1$, and ψ_j is a nondecreasing function, $\lambda_{il}^1 \leq \psi_i\left(r_s^2\right)$. Thus $\lambda_{il}^2 < \psi_i\left(r_s^2\right)$. Clearly $r_j^2 \leq \lambda_{il}^2$. It follows that $r_j^2 < \psi_i\left(r_s^2\right)$. From the feasibility of rate vector $\vec{r}$ and since virtual session $j \in m(i,l)$, r_j^2 is in the range of Φ_i and hence if $r_j^2 \geq r_s^2$, then $r_j^2 \geq \psi_j\left(r_s^2\right)$. It follows that $r_j^2 < r_s^2$. Also $r_j^2 \leq \lambda_{il}^2 < \lambda_{il}^1 = r_j^1$. Thus $r_j^2 \neq r_j^1$. Thus $j \in \tau$. Now, $r_j^2 < r_s^2$ contradicts the minimality of r_s^2, for $s \in \tau$. Thus there does not exist a feasible rate vector fairer than $\vec{r}^1$ in this case. Hence $\vec{r}^1$ is a maximally fair layer allocation vector. □

Now we prove Theorem 2. The proof follows from Lemmas 10 and 11.

Proof of Theorem 2: Let the generalized maximal fairness algorithm terminate in k iterations. Note that $\vec{r}(k)$ is feasible by Lemma 10. Since $S(k) = \phi$, there does not exist a virtual session s in $S(k)$. Thus from Lemma 11 for every virtual session s there exists a link which satisfies the properties of a generalized bottleneck link w.r.t. virtual session s for rate vector $\vec{r}(k)$. Thus $\vec{r}(k)$ is a maximally fair layer allocation vector by Lemma 12. Thus (1) follows.

If a maxmin fair rate allocation exists, then it is fairer than any other rate allocation vector, by definition of relative and maxmin fairness. Thus it is the only maximally fair layer allocation vector in the feasible set, by definition of maximal fairness. Since the output rate allocation vector, $\vec{r}(k)$ is a maximally fair under all circumstances, (2) follows. □

6 Conclusion and Discussion

This paper develops a framework for studying diverse fairness objectives in multirate, multicast scenario. The key contribution is to present an algorithm for computing the fair utility allocations for very general utility functions. The algorithm terminates in polynomial number of iterations. We mention some interesting topics for related future research. We pointed out that computation of lexicographically optimal utility allocation is NP-hard. Hence, we focussed on a weaker notion of fairness, maximal fairness which has intuitively appealing fairness properties. Another direction is to develop approximation algorithms and heuristics for computing lexicographically optimal utility allocation. It may also be possible to develop a lower complexity algorithm for computing maximally fair utility allocation. Another promising direction is to explore the connection between the choice of utility functions and pricing mechanisms. We hope that this study would initiate further research in these areas.

References

[1] T. Ballardie, P. Francis, and J. Crowcroft. Core based trees: an architecture for scalable inter-domain multicast routing, *Proceedings of ACM SIGCOMM,* Ithaca, NY, Sept. 1993, pp. 85-95

[2] D. Bertsekas and R. Gallager, *Data Networks*, Englewood Cliffs, NJ:Prentice-Hall, 1987.

[3] T. Bially, B. Gold, and S. Seneff. A technique for Adaptive Voice Flow Control in Integrated Packet Networks, *IEEE Transactions on Communications,* Vol. 28, No. 3, March 1980, pp. $325-333$

[4] F. Chiussi, Y. Xia and V. Kumar. Performance of Shared-Memory Switches under Multicast Bursty Traffic *IEEE Journal on Selected Areas In Communications*, Vol 15, No. 3, 1997.

[5] Z. Cao and E. Zegura. Utility Max-Min: An Application-Oriented Bandwidth Allocation Scheme *IEEE INFOCOM'*99

[6] S. Deering and D. Cheriton. Multicast routing in datagram internetworks and extended LANs, *ACM Transactions on Computer Systems*, vol 8, no. 2, pp. 54-60, Aug. 1994.

[7] F. Kishino, K. Manabe, Y. Hayashi and H. Yasuda. Variable Bit-Rate Coding of Video Signals for ATM Networks, *IEEE Journal on Selected Areas In Communications*, Vol. 7, No. 5, June 1989.

[8] F. Kelly, A. Maulloo, and D. Tan. "Rate control for communication networks: Shadow prices, proportional fairness and stability" *Journal of Operations Research society,* 49(3):$237-252$, March 1998.

[9] S. Low and D. Lapsley. Optimization Flow Control, I: Basic Algorithm and Convergence, *IEEE/ACM Transactions on Networking,* 7(6), Dec. 1999.

[10] X. Li, S. Paul and M. H. Ammar. Layered Video Multicast with Retransmission (LVMR): Evaluation of Hierarchical rate control, *Proceedings of IEEE Infocom' 98*, San Francisco, CA, March 1998

[11] X. Li, S. Paul and M. H. Ammar. Multi-session ate Control for Layered Video Multicast *Technical Report* GT-CC-$98-21$, College of Computing, Georgia Institute of Technology, 1998

[12] S. McCanne, Scalable Compression and Transmission of Internet Multicast Video. *PhD thesis*, Univ. of California, Berkeley, December 1996.

[13] S. McCanne, V. Jacobson and M. Vetterli. Receiver-Driven Layered Multicast, *Proceedings of ACM SIGCOMM '96*, Stanford, CA, September 1996

[14] D. Rubenstein, J. Kurose and D. Towsley. The Impact of Multicast Layering on Network Fairness *Proceedings of ACM SIGCOMM '*99, *Cambridge, MA, September,* 1999

[15] S. Sarkar and K. N. Sivarajan: Fairness in Cellular Mobile Networks *Proceedings of the 34th Annual Allerton Conference on Communication, Control and Computing,*1996, *pp* $457 - 469$

[16] S. Sarkar: Fairness and Congestion Control in multirate multicast networks. *Phd Thesis*, University of Maryland, College Park, July 2000

[17] S. Sarkar and L. Tassiulas: Distributed Algorithms for Computation of Fair Rates in Multirate Multicast Trees *Proceedings of IEEE INFOCOM'* 2000, Tel Aviv, Israel, March 2000

[18] S. Sarkar and L. Tassiulas: Fair Allocation of Discrete Bandwidth Layers in Multicast Networks *Proceedings of IEEE INFOCOM'* 2000, Tel Aviv, Israel, March 2000

[19] S. Sarkar and L. Tassiulas: Fair Allocation of Discrete Bandwidth Layers in Multicast Networks *Technical Report* TR $99 - 43$, Institute for Systems Research and University of Maryland, 1999.

[20] S. Sarkar and L. Tassiulas: Fair Allocation of Utilities in Multicast Networks *Technical Report* Electrical Engineering Department, University of Pennsylvania, 2001, Available at http://www.seas.upenn.edu/~swati/publication.htm

[21] T. Turletti and J. C. Bolot Issues with multicast video distribution in heterogeneous packet networks, *Packet Video Workshop*, '94, Portland.