A GENERAL THEORY OF CONSTRAINED MAX-MIN RATE ALLOCATION FOR MULTICAST NETWORKS

Jordi Ros and Wei K Tsai
Department of Electrical and Computer Engineering
University of California, Irvine, CA 92697
{jros, wtsai}@ece.uci.edu

Abstract

This paper presents a general theory of network max-min rate assignment as a lexicographic optimization. The model includes multicast and lower bound constraints. The model for multicast allows the sender to send at the maximum rate allowed by the network and the receivers. Equivalent optimality conditions, especially those which can be coded into practical algorithms, are derived. A reference parallel algorithm is also derived. The theoretical results clearly show the important role of the advertised rates in automatable optimality conditions. The theory also shows that, once the single-link problem is solved, the multi-link (network) problem can be simply solved by recursively applying the algorithm for single-link problem.

I. INTRODUCTION

Max-min fair allocation has been very popular in rate-based flow control for data networks. While there exist other fair allocation policies, the max-min policy is widely adopted. The main reason is that it deals with the bottlenecks explicitly and it gives all sessions competing for the same bottlenecked resource a fair (equal) share. Max-min allocation will not produce the maximum throughput for all users, but it is an optimal policy in the sense of maximizing the rates of the bottlenecked users in the order of fair share rates. In the networking literature, the max-min allocation policy has been largely viewed as a fairness criterion. This has caused the community to under-appreciate the optimization aspects. It is the belief of the authors of this paper that, this under-appreciation may have led to improperly design of max-min protocols.

For example, most max-min protocols do not even converge to the max-min solution due to an over-looked condition called pseudo-saturation defined by us [Tsa99a, IT99]. Further, most protocols do not properly handle the lower bound constraints (such as the minimum cell rate or MCR in ATM networks). Some opts for the trivial case of modifying the max-min criterion into the so-called MCR-plus-fair-share (MFR) condition. The MFR condition reduces the more difficult max-min problem into a much simpler problem with trivial lower bounds (MCR=0 for all connections). While others choose to tackle the harder problem with non-trivial (non-zero) minimum rate constraints, they come up with complicated protocols (see, for example the GMM protocols [Hou98, Kal97, LHR99, HR99]). In our earlier work [Tsa99a, Tsa99b], we show that, by choosing a modified max-min condition, sorting in the GMM protocols can be avoided altogether. In all the existing works on GMM (or max-min with minimum rate constraints), the complexity of the switch code is at least $O(n)$ where n is the number of sessions passing through the switch. With the theory derived in this paper, the switch code can be reduced to $O(1)$ in complexity.

Another major issue to be addressed in max-min flow control is multicast rate allocation. To deal with the problem of bandwidth variation and heterogeneity in multicast streaming applications, layered encoding is most often recommended. A new feature of layered encoding is that there exist multiple rates in the same multicast group. More specifically, a branch's rate is no longer limited by the minimum down-stream branch rate, but rather the maximum down stream branch rate. Thus it become attractive to extend the max-min policy to this multi-rate environment of layered encoded multicast, see [RKT99]. While [RKT99] provides an excellent motivation for multirate max-min allocation for multicast, no automatable optimality condition was derived. An automatable optimality condition is one that can be coded into a realistic computer program. For example, a non-automatable condition is one that requires an exhaustive search over all possible rate allocation solutions, which is not

[1] This research is supported by the National Science Foundation, award ANI-9979469, under the CISE ANIR program.
This work is also supported in part by the University of California, Irvine, and the Generalitat de Catalunya through a Balsells Fellowship to Jordi Ros.

practical.

The time has come to study the max-min problem from the perspective of optimization theory. Most works on max-min protocols arise from the need to design protocols that satisfy max-min fairness. However, the fact that the max-min condition is really an optimality, i.e., necessary and sufficient condition for an underlying optimization problem has been largely ignored. It is the purpose of this paper to pursue this direction and develop a general theory for the network max-min rate assignment problem as an optimization problem. This paper builds on our previous works [Tsa98, Tsa99a, Tsa99b] where the focus has been on max-min protocol design and performance analysis. In contrast, the focus of this paper is analysis of the max-min problem in an optimization theoretic framework. We derive a comprehensive set of optimality conditions for the general max-min problem. The max-min problem is formulated as a max-min lexicographic optimization problem with constraints. A number of new optimality conditions are derived in this paper. Furthermore, all of the previously known max-min conditions are either simply derived or are deduced with little efforts from the general optimality conditions derived in this paper.

In addition, we also derive a reference parallel algorithm for the max-min problem. With the reference algorithm, once the single-link algorithm has been correctly designed, the multi-link (network) problem can be simply solved by recursively applying single-link algorithm. The parallel algorithm can be used as a reference to design an *optimal* max-min rate allocation protocol. The correctness and convergence of the reference algorithm are also proved.

This paper is organized as follows. Section II introduces the CMM (constrained Multicast Max-min) problem and derives the major optimality conditions. III introduces the reference parallel algorithm for constructing compatible distributed protocols. A simulation example of an execution of the algorithm is presented in section IV. Finally, a brief conclusion is provided in Section V.

II. CMMM THEORY

In what follows, we shall adopt the ATM terminology. A VC is a virtual connection. Each virtual connection can either be a multicast or unicast fixed path virtual circuit.

Definition 1. $VVC_i^{\,j}$. We say that $VVC_i^{\,j}$ is a virtual VC if it consists of a set of consecutive links in VC_i for which the flow of the data going through them satisfies: (1) it begins at either a source or a branching point, (2) it ends at either a destination or a branching point and (3) it doesn't cross any branching point. In the notation used, $j \in I_i$ denotes a VVC *j* in VC_i, where I_i is a set of indexes that characterize each VVC in VC_i.

The name virtual VC comes from the fact that branching points can be interpreted now as a set of multiple virtual sources, one for each branch, transmitting each one at different rates. Each of this virtual sources implicitly defines a new flow of data within the complete VC. Hence the name of *virtual* VC.

It is important to note that within a VVC, the rate is constant. In general, note that a whole multicast VC doesn't satisfy this property. This is the main reason for which the concept of VVC is needed when trying to extend classical unicast max-min theory to the multicast case. Indeed, note that the equivalent unicast element to the VVC is the complete VC.

Definition 2. *Rate* $r_i^{\,j}$. We define $r_i^{\,j}$ as the rate at which data through $VVC_i^{\,j}$ is transmitted.

In multicast, QoS (Quality of Service) parameters such as minimal rate constraint have to be designed in a per-destination basis. Because of this, it is necessary to define the concept of *leaf.*

Definition 3. *Leaf.* A VVC is said to be a leaf if it is incident to a destination.

Definition 4. *Downstream* $DS(VVC_i^{\,j})$ *and Upstream* $US(VVC_i^{\,j})$. We define the downstream set of $VVC_i^{\,j}$ as the set of VVCs that belong to the downstream of $VVC_i^{\,j}$, without including $VVC_i^{\,j}$. We define the upstream set of $VVC_i^{\,j}$ as the set of VVCs that belong to the upstream of $VVC_i^{\,j}$, including the same $VVC_i^{\,j}$.

Fig. 1 shows an example to illustrate the previous definition.

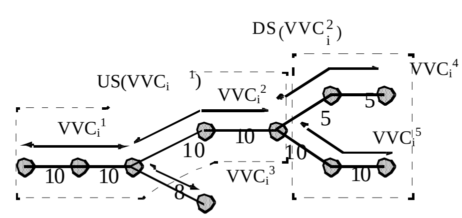

Figure 1. Definitions in a multicast VC_i

Note that from the same figure we have $r_i^1 = r_i^2 = r_i^5 = 10, r_i^3 = 8, r_i^4 = 5$.

Definition 5. *Feasibility.* Let V_u be the set of VVCs crossing link *u*. A set of rates $\mathbf{r} = \{r_i^{\,j}\}$ is feasible if all of the following are true,

1. $F_u = \sum_{VVC_i^j \in V_u} r_i^j \le C_u \quad \forall$ link u (2.1)

2. $r_i^j = \max\{r_i^k | VVC_i^k \in DS(VVC_i^j)\} \; \forall \; VVC_i^j$ (2.2)

3. $m_i^j \le r_i^j \; \forall$ leaf VVC_i^j, (2.3)

where m_i^j is the minimal rate constraint of the destination adjacent to the leaf VVC_i^j.

The *Constrained Multicast Max-Min* (CMMM) Problem can be stated as follows,

$$\text{maximize } \mathbf{x(r)} \text{ subject to } \mathbf{r} \in \mathbf{X} \tag{2.4}$$

where $\mathbf{x}(\mathbf{r}) = \{x_i(\mathbf{r})\}$ is a vector of objectives defined inductively: $x_1 = \min\{r_i^j | \forall i, j\}$, and for $k \ge 2$, $x_k = \min\{r_i^j | \forall i, j \text{ and } r_i^j > x_{k-1}\}$ where $\boldsymbol{X}$ is the feasible constraint set described by definition 5.

The maximization specified in (2.4) is of the lexicographic order: x_k must be maximized before x_{k+1}, for all $k \in L(\mathbf{r})$, where $L(\mathbf{r})$ is the number of distinct rate values in the rate vector **r**.

Definition 6. *Increasing Permutation.* The *increasing permutation* $\tilde{\mathbf{x}}$ of any vector **x** is defined to be a re-arranged (permuted) vector from **x** such that the components of $\tilde{\mathbf{x}}$ are arranged in an increasing order: $\tilde{x}_k \le \tilde{x}_{k+1}$, for all *k*.

Definition 7. *Lexicographic Ordering.* Given two vectors (**x** and **y**) **x** is said to be lexicographically greater than or equal to ***y*** if $x_i < y_i$ for some *i*, then there exists a $k < i$ such that $x_k > y_k$.

By using the previous definition, the following property is straightforward.

Property 1. *The Lexicographic Optimality Condition*. A feasible rate vector ***r*** is a solution to the CMMM problem iff for every feasible rate vector ***y***, $\tilde{\mathbf{r}}$ is lexicographically greater than or equal to $\tilde{\mathbf{y}}$.

The previous property can be re-stated to obtain the *Constrained Multicast Max-Min Fairness*,

Definition 8. *Constrained Multicast Max-Min Fairness (CMMM).* A vector of rates $\mathbf{r} = \{r_i^j\}$ is said to be *constrained multicast max-min (CMMM)* if it is feasible and for each leaf VVC_i^j, r_i^j cannot be increased while maintaining feasibility without decreasing $r_{i'}^{j'}$ for some other leaf $VVC_{i'}^{j'}$ for which $r_{i'}^{j'} \le r_i^j$.

Essentially, the new definition reflects the idea that fairness has to be applied at the destination sites. In other words, the important point here is not to know how much a source can transmit (transmission rate at the source) but to know how much a destination can receive (throughput at the destination). Note that in the unicast case, both parameters happen to be the same, since the flow goes through a unique path (VC) with a constant rate.

In order to develop a generalized multicast theory, it is also necessary to extend the definition of minimal rate constraint at a leaf VVC to any arbitrary VVC.

Definition 9. *Extended Minimal Rate Constraint.* Given an arbitrary VVC_i^j, we define its minimal rate constraint as,

$$m_i^{*j} = \begin{cases} m_i^j, \text{ if } VVC_i^j \text{ is a leaf} \\ \max\{m_i^{*j'} \,|\, VVC_i^{j'} \in DS(VVC_i^j)\}, \text{ otherwise} \end{cases} \tag{2.5}$$

Note that the extension of the minimal rate constraint to any arbitrary VVC includes also the case in which it is a leaf. Note also that the previous expression is actually implicitly defined by the minimal rate constraint at the leafs. In other words, in order to satisfy condition 3 in definition 5 it is necessary that,

$$m_i^{*j} \le r_i^j \tag{2.6}$$

for any VVC_i^j.

Definition 10. *Advertised Rate.* Let Vm_u denote the set of VVCs $\{VVC_i^j\}$ crossing link *u* with the condition that $m_i^{*j} = r_i^j$ at link *u*. The quantity R_u defined below is called the *advertised rate* for link *u*,

$$R_u \equiv \begin{cases} \max\{r_i^j | VVC_i^j \in V_u,\; r_i^j > m_i^{*j}\} & \text{if } Vm_u \ne V_u,\; F_u = C_u \\ \infty & \text{if } F_u < C_u \\ 0 & \text{if } Vm_u = V_u,\; F_u = C_u \end{cases} \tag{2.7}$$

Definition 11. *Constant Path* p_i^j. We define the *constant path* of a leaf VVC_i^j as:

$$p_i^j = \{VVC_i^k \,|\, VVC_i^k \in US(VVC_i^j), r_i^k = r_i^j\} \tag{2.8}$$

In other words, the constant path of a leaf is the set of its upstream VVCs such that transmit the same rate as itself's.

Definition 12. *Bottleneck Link Constrained (BLC).* A leaf VVC_i^j is said to be *bottleneck link constrained (BLC)* if there exists at least one link $u \in VVC_i^k$ for some $VVC_i^k \in P_i^j$ such that:

1. $F_u = C_u$,
2. $r_i^j = r_i^k > m_i^{*k}$ and
3. $r_i^j = R_u$.

Definition 13. *Minimal Rate Constrained (MRC).* A leaf $VVC_i^{\,j}$ is said to be *minimal rate constrained (MRC)* if there exists at least one link $u \in VVC_i^{\,k}$ for some $VVC_i^{\,k} \in P_i^{\,j}$ such that:

1. $F_u = C_u$,
2. $r_i^{\,j} = r_i^{\,k} = m_i^{*k}$ and
3. $r_i^{\,j} \geq R_u$.

Definition 14. *Constrained and Unconstrained leafs.* A leaf $VVC_i^{\,j}$ crossing link *u* is said to be a *constrained leaf* at link *u* if,

1. $VVC_i^{\,j}$ is BLC at link $v \neq u$, or
2. $VVC_i^{\,j}$ is MRC at some link.

Conversely, a leaf $VVC_i^{\,j}$ crossing link *u* is said to be an *unconstrained* leaf at link *u* if it is not constrained at this link.

The previous definition arises from the fact that $VVC_i^{\,j}$ *is* unconstrained *from link u point of view* because $VVC_i^{\,j}$ is not constrained by some other means to set its rate lower or higher than its own advertised rate.

Definition 15. *Saturation.* A link *u* is said to be *saturated* if the total flow is equal to its capacity, this is $F_u = C_u$.

Definition 16. *Pseudo-saturated.* A link *u* is said to be *pseudo-saturated* if every VVC crossing this link is constrained at it and its capacity is not fully utilized, this is $F_u < C_u$.

Theorem 1. *Bottleneck Optimality Condition.* A Feasible rate vector $\mathbf{r} = \{r_i^{\,j}\}$ is CMMM if and only if every leaf $VVC_i^{\,j}$ is either BLC or MRC.

Proof: (only if part) Assume that **r** is CMMM and suppose there exists a leaf $VVC_i^{\,j}$ that is neither BLC nor MRC. Consider also every link *u* in $P_i^{\,j}$ and let $VVC_i^{\,k}$ cross this link *u*, this is $VVC_i^{\,k} \in V_u$. Note that $r_i^{\,k} = r_i^{\,j}$, since both VVCs belong to the same constant path. There are two cases to consider: (1) link *u* is saturated or (2) link *u* is not saturated.

Consider case (1) where link *u* is saturated. There are two other cases: (1.1) $r_i^{\,j} = m_i^{*k}$ or (1.2) $r_i^{\,j} > m_i^{*k}$. Assuming case (1.1), since $r_i^{\,j}$ is not MRC, we have $r_i^{\,j} < R_u$. Consider two more cases: (1.1.1) $Vm_u \neq V_u$ or (1.1.2) $Vm_u = V_u$. If $Vm_u \neq V_u$ (1.1.1), then because of saturation assumption and definition 10, there must exist a $VVC_{i'}^{\,k'}$ crossing link *u* such that $r_{i'}^{\,k'} > m_{i'}^{*k'}$ and $r_{i'}^{\,k'} = R_u$. Since $r_i^{\,j} < R_u$, we have $r_{i'}^{\,k'} > r_i^{\,j}$. Now let $VVC_{i'}^{\,j'} \in DS(VVC_{i'}^{\,k'})$ be the leaf that satisfies $r_{i'}^{\,j'} = r_{i'}^{\,k'}$ (note that from condition 2 in definition 5 such a leaf must exist). We conclude that we can increase $r_i^{\,j}$ while maintaining feasibility at link *u* by decreasing some other leaf's rate $r_{i'}^{\,j'}$, where $r_{i'}^{\,j'} > r_i^{\,j}$. Assume now case (1.1.2), where $Vm_u = V_u$. Then, from saturation condition and definition 10 we have that $R_u = 0$. Since $r_i^{\,j} < R_u$ we arrive at a contradiction. Consider case (1.2). Then we can ensure that $r_i^{\,j} < R_u$. This comes from the fact that from definition 10, $r_i^{\,j} \leq R_u$, and also since $VVC_i^{\,j}$ is not BLC we have that $r_i^{\,j} \neq R_u$. Hence, from definition 10, there must exist a $VVC_{i'}^{\,k'}$ crossing link *u* such that $r_{i'}^{\,k'} > m_{i'}^{*k'}$ and $r_{i'}^{\,k'} = R_u$. Since $r_i^{\,j} < R_u$, we have $r_{i'}^{\,k'} > r_i^{\,j}$. Now let $VVC_{i'}^{\,j'} \in DS(VVC_{i'}^{\,k'})$ be the leaf that satisfies $r_{i'}^{\,j'} = r_{i'}^{\,k'}$ (note that again such leaf must exist). Hence, as in case (1.1.1), we conclude that we can increase $r_i^{\,j}$ while maintaining feasibility at link *u* by decreasing some other leaf's rate $r_{i'}^{\,j'}$, where $r_{i'}^{\,j'} > r_i^{\,j}$.

Finally, assume case (2) where link *u* is not saturated. Then $VVC_i^{\,j}$ can be increased at this link while maintaining feasibility without dropping any other rate.

From all the previous cases, we can conclude that at every link *u* in $P_i^{\,j}$, it is possible to increase $r_i^{\,j}$ by a non-zero amount without decreasing the rate of some $r_{i'}^{\,j'}$ for which $r_{i'}^{\,j'} \leq VVC_i^{\,j}$ while maintaining feasibility at the link. This contradicts the CMMM definition.

(if part) Assume every leaf is either BLC or MRC. Consider the two cases: (1) $VVC_i^{\,j}$ is BLC at a link *u* or (2) $VVC_i^{\,j}$ is MRC at a link *u*.

Assume case (1). Then, in order to increase $r_i^{\,j}$ without violating the capacity constraint, one must decrease some other $r_{i'}^{\,k'}$ such that $VVC_{i'}^{\,k'}$ crosses link *u* and $r_{i'}^{\,k'} > m_{i'}^{*k'}$. Because $VVC_i^{\,j}$ is BLC at link *u*, we have that $r_i^{\,j} = R_u \geq r_{i'}^{\,k'}$. Now let $VVC_{i'}^{\,j'} \in DS(VVC_{i'}^{\,k'})$ be the leaf that satisfies $r_{i'}^{\,j'} = r_{i'}^{\,k'}$. We conclude that we cannot increase $r_i^{\,j}$

while maintaining feasibility at link *u* without decreasing a leaf's rate $r_{i'}^{j'}$, where $r_{i'}^{j'} \le r_i^j$.

Assume case (2) where VVC_i^j is MRC at a link *u*. There are two sub-cases: (2.1) link *u* is saturated or (2.2) link *u* is not saturated. Suppose that link *u* is saturated (2.1). If we increase the rate r_i^j we must drop some other rate in order to satisfy the feasibility condition. We cannot drop any VVC whose rate is at its minimum. Thus, the only rate we can drop must be from any $VVC_{i'}^{k'}$ crossing link *u* such that $r_{i'}^{k'} > m_{i'}^{*k'}$. But by the MRC and the advertised rate definition we have

$$r_i^j \ge R_u = \max\{r_{i'}^{j'} \mid VVC_{i'}^{j'} \in V_u, r_{i'}^{j'} > m_{i'}^{*k'}\}.$$

This means that $r_{i'}^{k'} \le r_i^j$. Now let $VVC_{i'}^{j'} \in DS(VVC_{i'}^{k'})$ be the leaf that satisfies $r_{i'}^{j'} = r_{i'}^{k'}$. We conclude that we cannot increase r_i^j while maintaining feasibility at link *u* without decreasing a leaf's rate $r_{i'}^{j'}$, where $r_{i'}^{j'} \le r_i^j$. Finally, assume case (2.2) where link *u* is not saturated. Then from definition 10 we have that $R_u = \infty$. From the MRC assumption, we have $r_i^j = m_i^{*k} \ge R_u$, where VVC_i^k crosses link *u*. Now note that this last contradicts feasibility condition and, hence, is not possible. ***Q.E.D.***

Theorem 1 gives us the first automatable optimality condition. This condition can be coded into a proper switch code and a corresponding source code, with RM (Resource Management) cells carrying out the feedback signaling. The next theorem will provide us a mathematical formula for both the source and the switch codes.

Theorem 2. *Projection Optimality Condition.* A rate vector **r** is CMMM if and only if for every leaf VVC_i^j its rate satisfies the following condition,

$$r_i^j = \max\{\min\{R_{u'} | u' \in P_i^j\}, \max\{m_i^{*k'} | VVC_i^{k'} \in P_i^j\}\} \quad (2.9)$$

We call the above expression the *projection optimality condition.*

Proof: (only if part) Assume **r** is CMMM and let VVC_i^j be an arbitrary VVC. Then there are two cases to consider: (1) VVC_i^j is MRC at a link *u* or (2) VVC_i^j is BLC at a link *u*. Note that these are non-exclusive cases in the sense that the VVC can be MRC at a link and at the same time it can be BLC at some other link. Assume case (1). Then, we have $r_i^j = m_i^{*k} = \max\{m_i^{*k'} \mid VVC_i^{k'} \in P_i^j\}$, otherwise condition 2 in the feasibility definition would not hold. Also, since $r_i^j \ge R_u$ then we conclude that (9) is true. Consider case (2). There are two more cases: (2.1) $R_u = \min\{R_{u'} \mid u' \in P_i^j\}$, (2.2) $R_u \ne \min\{R_{u'} | u' \in P_i^j\}$. Assume case (2.1), then (2.9) is true since condition 2 in the feasibility definition implies that $r_i^j \ge \max\{m_i^{*k'} | VVC_i^{k'} \in P_i^j\}$ and BLC condition implies $r_i^j = R_u$. Assume now case (2.2), then exits a link u' such that $R_{u'} < R_u$ and $u' \in P_i^j$. There are two more sub-cases: (2.2.1) $r_i^{k'} = m_i^{*k'}$ or (2.2.2) $r_i^{k'} > m_i^{*k'}$, where $VVC_i^{k'} \in V_{u'}$, this is $VVC_i^{k'}$ crosses link u'. If $r_i^{k'} = m_i^{*k'}$ (2.2.1), then because of BLC condition we have $r_i^j = R_u > R_{u'}$. We also have $F_{u'} = C_{u'}$, otherwise $R_{u'} = \infty$ and feasibility condition would not hold. Then, we conclude that VVC_i^j is MRC at link u' and this case is reduced to that of case (1). Assume case (2.2.2), where $r_i^{k'} > m_i^{*k'}$. Then from the advertised rate definition, we have $R_{u'} \ge r_i^{k'}$. Since $r_i^{k'} = r_i^j = R_u$, we must conclude $R_{u'} \ge R_u$, which is a contradiction.

(if part) Assume that (2.9) is true and let *u* be a link such that $R_u = \min\{R_{u'} | u' \in P_i^j\}$. There are two cases:

(1) $R_u > \max\{m_i^{*k'} \mid VVC_i^{k'} \in P_i^j\}$ and

(2) $R_u \le \max\{m_i^{*k'} \mid VVC_i^{k'} \in P_i^j\}$.

Assume case (1). Then from (2.9) we have that $r_i^j = R_u$ and $r_i^j > m_i^{*k}$, where $VVC_i^k \in V_u$. We also have that $F_u = C_u$, otherwise $R_u = \infty$ and feasibility condition would not hold. Now this means that VVC_i^j is BLC at link *u*. Assume case (2), then from (2.9) we have that $r_i^j \ge R_u$. There are two more cases: (2.1) $r_i^j = m_i^{*k}$ or (2.2) $r_i^j > m_i^{*k}$, with $VVC_i^{k'} \in V_u$. If case (2.1) holds, then VVC_i^j is MRC at link *u*. On the other hand, if case (2.2) holds, then from the advertised rate definition we have $R_u \ge r_i^j$. Since we also had that $r_i^j \ge R_u$, we conclude that $r_i^j = R_u$ and, hence, VVC_i^j is BLC at link *u*. ***Q.E.D.***

This second theorem tells us that the source code is very simple, all the hard work is done in the switch code, i.e., the advertised rates at the switches. If one uses the optimality conditions derived in [Hou98, Kal97, LHR99, HR99], the switch has to work very hard (by looping through every single rate associated with each link, thus O(n) computation).

III. THE CPG ALGORITHM

In order to define an algorithm to compute the max-min solution, we need to define the concepts of fair share and remaining fair share.

Definition 17. *Fair Share.* The ratio C_u / N_u, where N_u is the number of VVCs crossing link *u,* is called the *fair share* at link *u*; intuitively, it is the fair share for all the VVCs crossing link *u* while assuming none of them is constrained.

Definition 18. *Remaining Fair Share.* The ratio $(C_u - Fc_u)/(N_u - Nc_u)$ is called the *remaining fair share* at link *u*, where Fc_u is the sum of rates for the constrained VVCs at link *u* and Nc_u is the number of constrained VVCs at link *u*, if $N_u - Nc_u > 0$.

A. The Single-link Algorithm

Theorem 3. *Convergence of the Single-link Projection Algorithm*. The single-link projection algorithm presented in fig. 2 converges to the constrained max-min solution in a finite number of steps.

Single_Link(*u:* singlelink network)

1. Set all VVCs as unconstrained;
2. For any unconstrained $VVC_i^{\,j}$ set $r_i^{\,j} = \dfrac{C_u - Fc_u}{N_u - Nc_u}$;
3. Compute R_u;
4. If $r_i^{\,j} \neq \max\{R_u, m_i^{*j}\}$ for any $VVC_i^{\,j}$ then:
   4.1. Set $r_i^{\,j} = \max\{R_u, m_i^{\,j}\}$ for each $VVC_i^{\,j}$;
   4.2. If $N_u \neq Nc_u$ go to 2;

Figure 2. Single_Link Projection Algorithm

Proof: Note that if the algorithm happens to finish, then the solution is optimal since it satisfies the projection condition. So we only need to show convergence. It is also worth noticing that because *N* is a single-link network, from definition 14 we have that the set of constrained VVCs can only include MRC constrained VVCs.

To prove convergence, consider the first iteration at step (1). Since initially all VVCs are unconstrained, all of them will be assigned the same fair share rate. Now at step (2) there are two cases: (1) $m_i^{*j} \le r_i^{\,j}$ for any $VVC_i^{\,j}$ and (2) $m_i^{*j} > r_i^{\,j}$ for some $VVC_i^{\,j}$. Consider case (1). Then, from the advertised rate definition, $r_i^{\,j} = R_u$ for all VVCs and the algorithm will stop. Consider now case (2). Since the rates for those VVCs such that $m_i^{*j} > r_i^{\,j}$ will be increased to m_i^{*j}, at the next iteration the remaining fare share at step 2 will be smaller. This means that those VVCs that become minimal rate constrained will remain under that condition at the next iteration. Now this shows that at every iteration the number of MRC VVCs does not decrease. We have seen that if it remains the same, then the algorithm finishes. Otherwise, the number of MRC VVCs will keep on incrementing. Now since there is a finite number of VVCs, we can ensure that the algorithm will finish in a finite number of steps. ***Q.E.D.***

Corollary 1. *Single-Link Algorithm Cost.* The worst case cost of the Single-Link Projection Algorithm is $O(N_u)$, with N_u the number of VVCs in link u.

Proof. From proof in theorem 3, in the worst case at each iteration only one VVC becomes MRC. So it will take at most N_u iterations to obtain the constrained max-min solution. ***Q.E.D.***

CPG_ALgorithm (*N:* Network)

0. *L = 1;*
1. For each link *u* in *N*, obtain R_u and $r_i^{\,j}$, for each $VVC_i^{\,j} \in V_u$, by executing Single_Link(Link *j*);
2. For each link *u* such that $R_u = \min\{R_{u'} \mid u' \in US(u) \bigcup DS(u)\}$ in *N* do the following:
   3.1. Remove (from the network) link *u* and any VVC crossing this link as well as their downstream VVCs;
   3.2. Add link *u* as a child node in the CPG graph to any level *L-1* link which had a shared VVC with any link in $US(u) \cup DS(u)$;
   3.3. For each link still in the network, reduce its capacity by the bandwidth assigned to the flows crossing it and deleted in step 3.1. (if any);
3. If the set of VVC leafs in *N* is not empty, do *L=L+1* and goto 1.

Figure 3. CPG Algorithm

B. The Multi-link Algorithm

We now present an algorithm that computes the CMMM rates for the multi-link case: the *Constrained Precedence Graph (CPG) Algorithm*. The constrained precedence graph is similar to the computation precedence graph commonly used in the computation theory except that the precedence relationships are defined in terms of convergence sequence of link advertised rates to their optimal advertised rates.

Before presenting the algorithm, we need to extend the definitions of downstream and upstream so that they can be applied to links. In this case, the downstream of a link u is defined as the set of links that are crossed by at least one VVC that belongs to the set,

$$\bigcup_{VVC_i^j \in V_u} DS(VVC_i^j) \qquad (2.10)$$

Similarly, the upstream of a link u is defined as the set of links that are crossed by at least one VVC that belongs to the set,

$$\bigcup_{VVC_i^j \in V_u} US(VVC_i^j) \qquad (2.11)$$

Theorem 4. *Convergence of the CPG Algorithm.* The rates computed by the CPG algorithm (fig. 3) converge to the CMMM solution in a finite number of steps.

Proof. We require two conditions for the proof: (1) *Finiteness*, the CPG finishes in a finite number of steps; (2) *Correctness*, assuming the CPG finishes then the output rates are CMMM.

To prove finiteness we need to make sure that the following two conditions hold: (1) the set of capacities obtained after removing links and VVCs in step 3.1. are non-negative; (2) at each iteration at least one link is removed so that eventually the network is empty. The first condition is a requirement for the algorithm in order to avoid an incoherent state. To prove it, we need to look at how links at step 2 are chosen. Note that at a particular level *L*, we only chose those links that have minimal advertised rate among the links in both their upstream and downstream sub-trees. Because of the advertised rate definition, a VVC that does not exceed the capacity requirements in the link with minimal advertised rate along its path will not exceed the capacity requirements of any other link that it crosses. Hence, feasibility condition is preserved at each iteration and a new iteration can be safely executed every time step 4 calls back for step 1. Th second condition is trivial if we notice that the set of links defined at step 2 is always non-empty.

To prove correctness, consider an arbitrary leave VVC_i^j and its rate assigned by the CPG algorithm r_i^j. Let also u be the link that caused VVC_i^j to be removed and let VVC_i^k be such that $VVC_i^k \in V_u$. Clearly, link $u \in P_i^j$ and $r_i^j = r_i^k$. Also, $F_u = C_u$ since the output rates from the single-link algorithm always saturate the input link. From the same single-link algorithm, r_i^k can only take two values: (1) $r_i^k = R_u > m_i^{*k}$ or (2) $r_i^k = m_i^{*k} \geq R_u$. But from definitions 12 and 13 cases (1) and (2) means the leaf is BLC and MRC, respectively. Now applying theorem 1 we conclude that the output rates from the CPG algorithm are CMMM. ***Q.E.D.***

IV. SIMULATIONS

Consider the network configuration in figure 5. In this figure, VCs and VVCs are labeled with their name followed by the minimal rate constrain of their associated destination, if any. Links are also labeled with their available capacity. In this section we will apply our theoretical results in order to make a max-min analysis of this network example. We will first compute the max-min rates by using the CPG algorithm obtaining also the CPG graph. Correctness of the rate assignment will be checked by using the bottleneck and projection optimality conditions.

A. CPG Algorithm Execution

Table 1 shows the results of executing the CPG algorithm to our network sample. Each row represents one iteration. Values for the advertised rate at each link and the actions to perform are presented for each iteration. Though the number of bottleneck links is equal to 5, note that it only takes three iterations for the algorithm to converge. This is because step 3 in the CPG algorithm can be executed in parallel for more than one link.

Note that from table 1 links 1 and 4 don't have a shadowed cell since they are never removed from the network when executing the CPG algorithm. Links that are never removed correspond to pseudo-saturated links. For these links, the crossing VVCs are all constrained somewhere else and their capacity is not fully utilized.

From the same table, we can deduce the level at which each bottleneck link belongs: links *L7* and *L8* belong to the first level; links *L2* and *L3* belong to the second level and link *L5* belongs to the third level. In order to know how to join this links to form the CPG graph, we use the rule specified in step 3.2. The resulting CPG graph is shown in figure 4.

The adjacencies in the CPG graph give the precedence ordering. From the graph, we interpret that link 8 has to converge first so that links 2 and 3 can compute the max-min rate for their constrained VVCs. Also, link 5 won't be able to compute their max-min rate assignment before link 2 computes its own.

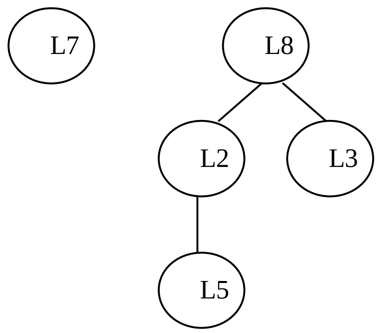

Figure 4. CPG Graph

B. Optimality Conditions

Theorem 1 gives an optimality condition in terms of the bottleneck links in the network. In that theorem we proved that a multicast rate assignment is max-min if and only if any leaf in the network is either BLC or MRC. Table 2 gives the results when checking the optimality condition for each VVC. Note that any VVC leaf is either BLC or MRC at least at one link, which confirms the optimality of the solution.

It could also be proven that the projection optimality condition presented in theorem 2 also holds. We leave the checking of this condition to the reader.

V. Conclusion

This paper derives the first constrained multicast max-min theory for multi-rate multicast connections up to the knowledge of the authors. The theory provides automatable optimality conditions that can be used by protocol designers to check the optimality of the solution. The theory turns out to be mathematically challenging; if one is not careful, some pathological counter-examples can arise. The hard work here is in the design of the advertised rate. Without the proper design of the advertised rate, the corresponding switch code will require $O(n)$ operations.

The linear objectives at each stage can also be generalized into weighted linear case and a specialized class of nonlinear objectives. This extension will be reported in a forthcoming paper.

References

[Aru96]A. Arulambalam, X. Chen, and N. Ansari, "Allocating Fair Rates for Available Bit Rate Service in ATM Networks", IEEE Communications Magazine, November 1996, pp.92-100.

[Hou98]Y. Hou, H. Tzeng, and S. Panwar, "A Generalized Max-Min Rate Allocation Policy and Its Distributed Implementation Using the ABR Flow Control Mechanism", Proc. IEEE Infocom '98, April 1998.

[HR99] Y.H. Ho and T.K. Rad, “A.B. Explicit rate allocation algorithm of generalised max-min fairness for ATM ABR services.” Electronics Letters, vol.35, (no.7), IEE, 1 April 1999. p.530-1. 5

[IT99] M. Iyer and W.K. Tsai, “Constraint Precedence in Max-Min Fair Rate Allocation,” submitted to IEEE trans. on Networking, October 1999.

[Kal97] L. Kalampoukas, 'Congestion Management in High Speed Networks,' Ph.D. Dissertation, University of California at Santa Cruz, August 1997.

[LHR99] Long, Y.H.; Ho, T.K.; Rad, A.B.; Lam, S.P.S. A study of the generalised max-min fair rate allocation for ABR control in ATM. Computer Communications, vol.22, (no.13), Elsevier, 25 Aug. 1999. p.1247-59.

[MY96] Mayer, A., Ofek, Y., and Yung, M., “Approximating max-min fair rates via distributed local scheduling with partial information.” IEEE INFOCOM '96.

[RKT99] D. Rubenstein, J. Kurose, D. Towsley, “The Impact of Multicast Layering on Network Fairness,” ACM SIGCOMM ‘99.

[TS97] Hong-Yi Tzeng and Kai-Yeung Siu, “On max-min fair congestion control for multicast ABR service in ATM.” IEEE Journal on Selected Areas in Communications, vol.15, (no.3), IEEE, April 1997. p.545-56.

[Tsa98]W. Tsai, Y. Kim, and L. Hu, "ASAP: A Non-Per-VC Accounting Max-Min Protocol for ABR Flow Control with Optimal Convergence Speed", IEEE SICON '98, Singapore, July 1998, pp.139-153. http:// www.eng.uci.edu/~netrol.

[Tsa99a]W.K. Tsai and Y. Kim, "Re-Examining Max-min Protocols: A Fundamental Study on Convergence, Complexity, Variations, and Performance," IEEE INFOCOM, 1999.

[Tsa99b]W.K. Tsai and Y. Kim, "Minimum Rate Guarantee without Per-Flow Information," IEEE ICNP ‘99.

[TW96] Tsang, D.H.K. and Wales Kin, Fai Wong, “A new rate-based switch algorithm for ABR traffic to achieve max-min fairness with analytical approximation and delay adjustment.” IEEE INFOCOM '96.

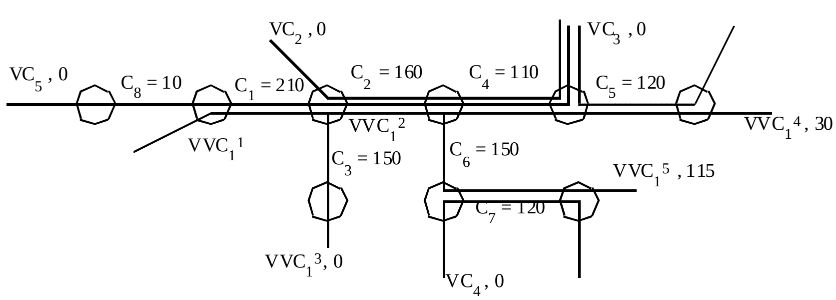

Figure 5. Network Configuration

Table 1. CPG Algorithm execution

	AR(L1)	AR(L2)	AR(L3)	AR(L4)	AR(L5)	AR(L6)	AR(L7)	AR(L8)	Actions
It 1	95	15	150	36.6	60	150	5	10	Remove L7, L8, VVC_1^5 , VC_4 , VC_5 , set $r_1^5 = 115$, $r_4 = 5$, $r_5 = 10$ and $C_1 = 200$, $C_2 = 150$, $C_4 = 100$
It 2	200	35	150	50	60				Remove L2, L3, VVC_1^3 , VVC_1^2 , VC_2 , set $r_1^3 = 150$, $r_1^2 = 115$, $r_2 = 35$ and $C_4 = 75$
It3	200	100		75	60				Remove L5, VC_3 , VVC_1^4 , set $r_3 = 60$, $r_1^4 = 60$.

Table 2. Bottleneck Optimality Condition check

	VVC_1^1	VVC_1^2	VVC_1^3	VVC_1^4	VVC_1^5	VC_2	VC_3	VC_4	VC_5
Rate	150	115	150	60	115	35	60	5	10
Constrained	not a leaf	not a leaf	BLC at L3	BLC at L3	MRC at L7 and L2	BLC at L2	BLC at L5	BLC at L7	BLC at L8