Maximum Flow Routing with Weighted Max-Min Fairness

Miriam Allalouf and Yuval Shavitt

School of Electrical Engineering
Tel Aviv University

Abstract. Max-min is an established fairness criteria for allocating bandwidth for flows. In this work we look at the combined problem of routing and bandwidth allocation such that the flow allocation for each connection will be maximized and fairness will be maintained. We use the weighted extension of the max-min criteria to allocate bandwidth proportionaly to the flows' demand. Our contribution is an algorithm which, for the first time, solve the combined routing and bandwidth allocation problem for the case where flows are allowed to be splitted along several paths. We use multi commodity flow (MCF) formulation which is solved using linear programming (LP) techniques. These building blocks are used by our algorithm to derive the required optimal routing and allocation.

1 Introduction

Traffic engineering is a paradigm where network operators control the traffic and allocate resources in order to achieve goals such as, maximum flow or minimum delay. One challenge is to allow different flows to share the network, so that the total flow will be maximized while fairness will be preserved. Given a network, and a set of demands between its nodes, a network operator may wish to maximize their profit by routing the traffic so it will maximize the total allocated bandwidth the network can carry, or namely the total assigned net flow. To do this we allow flows to be arbitrarily split inside the network.

One way to maximize the network flow is to formulate it as a maximum multi-commodity flow (MCF) problem which can be solved using linear programming (LP). While the solution will maximize the flow, it will not always do it in a fair manner. Flows that traverse several congested links will be allocated very little bandwidth or non at all, while flows that traverse short hop distances will receive a large allocation of bandwidth.

In an attempt to introduce fairness to the maximum flow problem the concurrent multi-commodity flow problem was suggested. This MCF LP formulation requires that all demands will be equally satisfied and seeks a routing that maximizes network flow in equal portions per demand. However, the end solution under-utilizes the network, sometimes saturating only a small fraction of it.

In this paper we suggest an algorithm, called WMCM (Weighted Max-min fair Concurrent MCF), that finds routing and resource allocation that maximizes the network link utilization, but, at the same time, adheres to the weighted max min fairness criterion [1]. The WMCM algorithm is based on the maximum concurrent multi-commodity flow problem and bridges the gap between the two other solutions described above.

To clarify the difference between the different algorithms consider the example in Figure 1, which depicts a line network with four nodes 1,2,3, and 4 and one unit capacity links (1,2), (2,3), and (3,4). We have a flow from node 1 to node 3, a flow from node 1 to node 2, one flow from node 2 to node 3 and one flow from node 2 to node 4. The maximum MCF problem results in an allocation vector (0,1,1/2,1/2) starving the two-hop 'pair 1' flow to achieve the maximum possible flow of 2 units. The max-min fair [2] vector in this case is (1/3,2/3,1/3,1/3). The WMCM algorithm are with equal weights for all pairs will results with the max-min vector given above. In case pair 1 is given a weight double from the rest of the nodes, the concurrent MCF problem will allocate it double the bandwidth allocated for the flows in its bottleneck link (link(2,3)) and weighted max-min vector is (1/2,1/2,1/4,1/4).

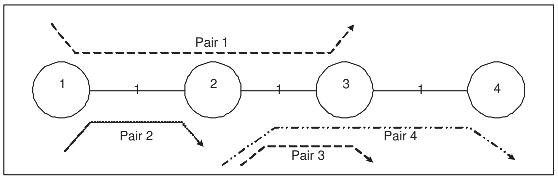

Fig. 1. Example of flows assignment: Maximum Flow:(0,1,1/2,1/2), Maximum Concurrent: (1/3,1/3,1/3,1/3), max-min fair and the WMCM algorithm weighted max-min fair:(1/3,2/3,1/3,1/3), The rate vector for the demand vector of (2,1,1,1) is (1/2,1/2,1/4,1/4)

2 Related Work

The Max-min fairness bandwidth allocation was mostly defined in the context of one fixed path per session, where a session is defined by a pair of terminals. A simple algorithm that finds the max-min fair allocation where routing is given appears in [2].

Chen and Nahrstedt [3] provide max-min fair allocation routing. They present an unweighted heuristic algorithm that selects the best path so the fairness-throughput is maximized upon an addition of a new flow. Thier algorithm assumes the knowledge of the possible paths for each new flow.

Many other distributed algorithms deal with dynamic adjustments of flow rates to maintain max-min fairness when the routes are given [4, 5, 6, 7]. The above algorithms differ by the assumptions on the allowed signaling, and available data. Bartal *et al.* [7] find the total maximum flow allocation in a network for given routes using distributed computations as the input to the global MCF LP problem.

Kelly *et al.* [8] propose proportional fairness concepts and a convergence algorithm. Mo and Warland [9] generalize the proportional fairness and produce end-to-end flow control of TCP streams by changing the transmission window size, but again they deal with flow allocation without routing.

LP in Traffic Engineering. Many works considered the multi-commodity flow allocation[1] as the mathematical formalism suitable for traffic engineering and path design. Ott *et al.* [16] suggest different off-line LP formulations to optimally spread paths so the excess bandwidth will be maximized in an MPLS networks. In order to avoid link "overflow" caused by flows fluctuation, they calculate how to reduce the assigned throughput per link. Mitra and Ramakrishnan [17] present algorithms that are based on the MCF LP problem allocating bandwidth for the various QoS-type services: QoS and Best-Effort traffic. Most of the studies, including the ones mentioned above, chose an MCF formulation that considers the demands but they do not discuss the max-min fairness in conjunction with maximum throughput as the WMCM algorithm does.

3 Algorithm

We consider as input a network topology and directional links capacities, a list of ingress-egress pairs, and per-pair traffic average-rate demand. Traffic between ingress-egress pair may be split arbitrarily among different paths. We model the network as a general directed graph where each arc label represents link capacity. Each (ingress,egress) peering point pair is represented by a different commodity with some demanded rate.

Our goal is to fulfills clients' demands optimally while keeping a fair sharing of the allocated bandwidth, to lay the set of paths to be used between each pair in the network, and to allocate them bandwidth in a maximal way. The fairness criterion is defined by the weighted max-min fairness.

The WMCM algorithm solves, iteratively, the maximum concurrent MCF LP until network saturation is achieved. Each iteration is computed on the residual capacity from the previous iteration with non-saturated flows left. First, we will provide the formal definition of the weighted max-min criterion (subsection 3.1), then we will describe the maximum concurrent flow (subsection 3.2), and finally we will present our WMCM algorithm (subsection 3.3).

3.1 Weighted max-min criterion: definitions

The weighted max-min fairness criterion is an extension [1] of the max-min fairness criterion [2]. While the max-min definition is stated for the case where each flow takes a single path, it can be also applied to the case where a flow may be split among several paths.

Definition 1. *The Commodity Rate Vector, cr, is a vector whose elements are the rates which were assigned to the commodities.*

Definition 2. *A Flow Rate Vector, f_i, is a vector of the rates assigned to a set of paths of commodity i.*

[1] Relevant mathematical and algorithmic background on MCF can be found in [10, 11, 12]. Specific theoretical aspects on the max. concurrent MCF problem and its complexity can be found in [13, 14, 15].

From the above definitions we can write $\sum_{P_{ij} \in P_i} f_{ij} = cr_i$. The weighted max-min fair algorithm finds the optimal commodity rate vector cr^* and a flow rate vector f_i per each commodity rate cr_i^* such that cr^* is the lexicographically largest feasible vector.

Definition 3. *The vector cr is said to be max-min fair if it is feasible and if each of its elements cr_i cannot be increased without decreasing any other element cr_k for which $cr_i \geq cr_k$*

Definition 4. *The vector cr is said to be weighted max-min fair if it is feasible and if for each commodity i, cr_i cannot be increased (maintaining feasibility) without decreasing any other element cr_k for which $cr_i/dem_i \geq cr_k/dem_k$.*

The two definitions above also holds when traffic may be split to several path.

3.2 Tools

The **Maximum Concurrent flow** problem is stated as follows. Let G=(V,A) be a directed graph with nonnegative capacities $c(a), \forall a \in A$. If $a \notin A$ $c(a) = 0$. There are K different commodities: $C_1, \ldots, C_K$ where commodity i is specified by the triplet $C_i = (s_i, t_i, dem_i)$. The pair (s_i, t_i) are the source and the sink of commodity i respectively and dem_i is its rate demand. Each pair is distinct, but vertices may participate in different pairs. The objective is to maximize z so all the $i = 1, \ldots, K$, $z \times dem_i$ units of the respective commodities can be routed simultaneously, subject to flow conservation and link capacity constraints. The objective z is the equal maximal fraction of all demands. There are two ways to formulate this problem: path flow and arc flow formulations.

Maximum concurrent flow: Path Flow formulation:
Linear program PR assigns the maximum commodity flow to P_i, the set of all paths between s_i and t_i that belong to the same commodity i is restricted by the fairness criterion. The assigned net flow per arc a is the sum of the net flows of the paths passing this arc. PR's solution is composed of the assigned net flow $f(P_{ij})$ $\forall P_{ij} \in P_i,\ i = 1, \ldots, K$ and the maximal fairness value z.

LP PR: Path Flow Formulation

$$\text{maximize } z$$
$$\text{subject to}$$
$$\forall a \in A, \sum_{i=1}^{K} \sum_{P \in P_i} f(P) \leq c(a) \text{ /*} a \text{ is an arc on } P\text{*/} \tag{1}$$
$$\forall i = 1, \ldots, K, \sum_{P \in P_i} f(P) \geq z \cdot dem_i \tag{2}$$
$$\forall P \in P_{i=1,\ldots,K} f(P) \geq 0, z \geq 0$$

The size of this linear problem grows with the number of possible paths between any pair of nodes and can be exponentially large when the network is highly connected.

For this reason we will reformulate this problem so it could be solved in a polynomial number of steps:
Maximum concurrent flow: Arc Flow formulation:
The variable $acf_k(i,j)\ \forall k = 1,\ldots,K\ \forall a = (i,j) \in A$ holds the assigned net flow per commodity k over arc (i,j). For each vertex $v \in V - \{s_k, t_k\}$, we require that the total flow of commodity k into vertex v is equal to the total flow of commodity k out of vertex v. $\forall a \in A$, the total flow over all commodities is at most $c(a)$. The total flow rate of commodity i out of vertex s_i represents the maximum possible flow per commodity i and should be greater than or equal to the fraction of the respective demand (Eq. 4).

Problem PR' Arc Formulation

$$\text{maximize } z$$
$$\text{subject to}$$
$$\forall a = (u,v) \in A, \sum_{k=1..K} acf_k(u,v) \leq c(u,v) \tag{3}$$
$$\forall k = 1..K, \forall v \in V - s_k, \sum_{v \in V} acf_i(s_k,v) \geq z * dem_i \tag{4}$$
$$\forall k = 1..K, \forall (u,v,w) \in V - s_k, t_k, acf_k(u,v) - acf_k(v,w) = 0 \tag{5}$$
$$\forall k = 1..K\ \forall u,v \in V, acf_k(u,v), z \geq 0 \tag{6}$$

PR' is a reformulation of PR which leads to a $O(K \cdot m)$ variables and $O(m + Kn)$ constraints where m is the number of arcs, n the number of vertices, and K the number of commodities, that can be solved in a polynomial number of steps.

The solution of PR' problem gives us the same maximal portion of all the demands restricted by links capacities. The LP variables hold the maximal value of the allocated net flow per commodity. To derive the set of routing paths per each commodity, we need to apply, per commodity, the **flow decomposition algorithm** as described in figure 2 and [18, 17]. This algorithm runs in a polynomial number of steps. It decomposes (separately for each commodity) the total net flow per commodity k over all arcs into a set of paths.

3.3 WMCM - Weighted max-min fair concurrent MCF algorithm

As noted before, the solution of the maximum concurrent MCF can lead us to an unsaturated network. Assuming two commodities with disjoint paths and the available bandwidth per commodity 1 is larger than that of commodity 2. Restricted by Eq. 4, namely z, PR' will assign both commodities the same net flow size although commodity 2 has more available bandwidth. We have developed a new algorithm that merges the desire for network saturation and fairness. The formal description of algorithm is given in Figure 3.

The main idea behind the WMCM algorithm is to increase the allocated net flow per arc over the residual graph while keeping the weighted fairness criterion at each iteration. We use the maximum concurrent MCF problem solution to increase it optimally.

```
Flow decomposition algorithm
for each commodity k do
    set P_k = null
    P_k holds the paths for commodity k
    while TRUE
      let p be a  path  between (s_k, t_k).
      if path p was found then
         minacf_k = min_{(u,v)∈p}(acf_k(u, v))
         Add p to set P_k
          f(p) = minacf_k
         ∀(w, x) ∈ p, acf_k(w, x) = acf_k(w, x) − minacf_k
      else go to 1
```

Fig. 2. The flow decomposition algorithm [18].

The algorithm recieves as input the list of commodities, $KCOMM$, the vector of demands, dem, and the graph, G. Each iteration starts (see line 4) with a reduced number of commodities (line 19), recalculated demands (line 17) and a residual graph of the unassigned capacities (lines 10-12). $\Delta acf_k(i,j)$ holds the net flow assigned in the iteration, and z holds the fraction of the demands that was fulfilled. Finally, $acf_k(i,j)$ holds the total accumulated net flow per commodity k over arc (i,j). The flow decomposition algorithm is performed (line 21) once and provides the maximum routing with the weighted max-min fair allocation. The strength of the algorithm is due to the optimality and scalability of the maximum concurrent MCF problem.

The WMCM algorithm provides us with the commodity rate vector cr, and a set of flow rate vectors f_k, the rates of the paths $P_k^j \in P_k, k = 1, \ldots, K$ per commodity k, composing each cr_i.

Theorem 1. *The commodity rate vector cr provided by the WMCM is weighted max-min fair.*

Before proving theorem 1, lets state the following lemma and explanations. The rate vector acf^n is composed of the accumulated net flow $acf_k(s_k, j)$ per commodity k where s_k is the source of commodity k, at the end of each iteration n. K^n is a set of the commodities that participate in iteration n.

Lemma 1. *$\exists x(n), y(n)$ and $u(n)$ such that $\forall i \in K^n$, we have the following:*

- $\Delta acf_i^n = y(n) * dem_i$ *(The increased rate is in proportion to the demand)*
- $acf_i^n = u(n) * dem_i$ *(The accumulated rate is in proportion to the demand)*
- $demRes_i^n = x(n) * dem_i$ *(The residual demand is in proportion to the original demand).*

Proof. We prove by induction on the number of the iterations.

WMCM(KCOMM,dem,G)

1. /* Initialization stage */
2. $\forall(i,j) \in A, k = 1..K, acf_k(i,j) = 0$
3. $G_{RES} = G$ /* G is the original graph */
4. **while** $(KCOMM \neq null)$ **do**
5. /* The calculation of the increased net flow by this iteration*/
6. **Perform** LP PR' on G_{RES} with $KCOMM$ and dem_k
7. **Returns**: z and $\Delta acf_k(i,j)$ $\forall a = (i,j) \in A$
8. /* Variable maintenance for the next iteration */
9. /* G_{RES} calculation: $c(a)$ hold links capacities */
10. $\forall a = (i,j) \in A, k = 1, \ldots, K,$
11. $c(a) = c(a) - acf_k(i,j)$
12. if $c(a) = 0, A = A \setminus \{a\}$ /* prune saturated links */
13. /*Commodities and demands calculations */
14. $lastKCOMM = KCOMM$
15. **for** commodity $k \in lastKCOMM$ **do**
16. $\forall a = (i,j), acf_k(i,j) = acf_k(i,j) + \Delta acf_k(i,j)$
17. $dem_k = dem_k - z \cdot dem_k$
18. **if** G_{RES} has no connectivity for k **then** /* k got maximum possible assignment */
19. $KCOMM = KCOMM \setminus \{k\}$
20. /*end of while*/
21. **Perform** "flow decomposition algorithm" on G and $acf_k(u,v), k = 1..K.$
22. **Returns** per commodity k: set of paths P_k and flows $\forall_{p_i^j \in P_i} f(p_i^j)$

Fig. 3. WMCM weighted max-min fair concurrent allocation algorithm

For n=1, the recalculated demand is $(1 - z_1) * dem_i$ (see lines 17), the rate increase (calculated in lines 7) is $z_1 * dem_i$ and the total rate is $z_1 * dem_i$.

The following three equations are the *induction assumption* for iteration n,

$$\Delta acf_i^n = z_n * \prod_{l=1}^{n-1} (1 - z_l) * dem_i \text{ /*Rate increase */} \tag{7}$$

$$acf_i^n = (\sum_{l=1}^{n} z_l * \prod_{m=1}^{l-1} (1 - z_m)) * dem_i \text{ /*Total rate*/} \tag{8}$$

$$demRes_i^n = \prod_{l=1}^{n} (1 - z_l) * dem_i \text{ /*Residual Demands*/} \tag{9}$$

Proof for iteration $n + 1$*:* $KCOMM^{n+1}$ holds the commodities that participate in iteration $n{+}1$. The rate increase (WMCM line 7) for any commodity $i \in KCOMM^{n+1}$ is:

$$\Delta acf_i^{n+1} = z_{n+1} * demRes_i^n = z_{n+1} * \prod_{l=1..n} (1 - z_l) * dem_i \tag{10}$$

Where the first transition is due to Eq. (9).

The accumulated rate for any commodity $i \in KCOMM^{n+1}$ is:

$$acf_i^{n+1} = acf_i^n + \Delta acf_i^{n+1} = (\sum_{l=1}^{n} z_l * \prod_{m=1}^{l-1}(1-z_m)) * dem_i + z_{n+1} * \prod_{h=1}^{n}(1-z_h) * dem_i$$
$$= (\sum_{l=1..n+1} z_l * \prod_{m=1}^{l-1}(1-z_m)) * dem_i$$

Where the first transition is due to Eqs. 8 and 10.

The residual demands are calculated at the end of iteration $n+1$ as follows:

$$demRes_i^{n+1} = (1-z_{n+1}) * demRes_i^n = \prod_{l=1}^{n+1}(1-z_l) * dem_i \quad (11)$$

Where the first transition is due to Eq. (9).

The lemma was proved by setting $x(n+1) = \prod_{l=1..n+1}(1-z_l)$, setting $y(n+1) = z_{n+1} * \prod_{l=1..n}(1-z_l)$ and setting $u(n+1) = \sum_{l=1..n+1} z_l * \prod_{m=1..l-1}(1-z_m)$ as required.

$\square$

We now return to the proof of Theorem 1

Proof. We prove by induction.

Base step: In the first iteration, where $n = 1$, acf^1 is the solution of PR' where for all commodities i and j, $acf_i^1 = z_1 * dem_i$ and $acf_j^1 = z_1 * dem_j$ implying $acf_i^1/acf_j^1 = dem_i/dem_j$. acf_i^1 can not be increased this iteration due to the PR' restrictions (line 7).

Induction Assumption: The weighted max-min fair order holds for iteration n. acf^n is feasible and if for each commodity i, acf_i^n cannot be increased without decreasing any other acf_j^n for some commodity j for which $acf_i^n/acf_j^n \geq dem_i/dem_j$.

Iteration $n+1$: $KCOMM^{n+1}$ is the set of all the commodities that participate in iteration $n+1$. $KCOMM^{SAT}$ is the set of all commodities that were saturated before, in one of the previous iterations. We distinguish among 3 cases for any commodity i and j:

- Case 1: Both commodities were saturated in the previous iterations, such that $i, j \in KCOMM^{SAT}$. This case holds trivially because of the induction assumption.
- Case 2: Only one of the two commodities was saturated before. W.l.o.g., assume that $i \in KCOMM^{n+1}$ and $j \in KCOMM^{SAT}$. Commodity j cannot increase its flow since it was deleted from the list (see line 19 in WMCM). If it was deleted in the previous iteration, n, then $acf_i^n/acf_j^n = dem_i/dem_j$ holds before starting iteration $n+1$, and thus any increase in commodity i rate (see lines 7 and 16) will implies $acf_i^n/acf_j^n > dem_i/dem_j$. If j was deleted before the previous iteration, n, we know that $acf_i^n/acf_j^n > dem_i/dem_j$ and then any increase in i's rate will keep the relation.

– Case 3: Both commodities participate in iteration $n+1$, thus, $i, j \in KCOMM^{n+1}$. Since both commodities participated in all the previous iterations, they gained rates such that $acf_i^n / acf_j^n = dem_i / dem_j$. As proved in lemma 1, the gain increase in this iteration keeps the same relation between the rates such that $acf_i^{n+1} / acf_j^{n+1} = dem_i / dem_j$.

Finally, $KCOMM^{SAT}$ is reduced in each iteration which ensures termination.

□

4 Concluding remarks

We presented a new traffic engineering algorithm for routing demands in a networks in a way that maximizes the flows and maintain fairness. The weighted max-min fair criterion serves better current needs for quality of service regarding clients demands. The algorithm can be used when clients request are above capacity to minimize the loss of revenue and maximize customer satisfaction. It can also be applied when network is not congested to lower the maximum load on the links.

References

[1] Marbach, P.: Priority service and max-min fairness. IEEE/ACM Transactions on Networking **11** (2003) 733–746

[2] Bertsekas, D., Gallager, R.: Data Networks. 2nd edn. Prentice Hall, Englewood Cliffs, NJ (1992)

[3] Chen, S., Nahrstedt, K.: Maxmin fair routing in connection-oriented networks. In: Proceedings of Euro-Parallel and Distributed Systems Conference (Euro-PDS '98). (1998) 163–168

[4] Afek, Y., Mansour, Y., Ostfeld, Z.: Phantom: a simple and effective flow control scheme. Computer Networks **32** (2000) 277–305

[5] Awerbuch, B., Shavitt, Y.: Converging to approximated max-min flow fairness in logarithmic time. In: INFOCOM (2). (1998) 1350–1357

[6] Bartal, Y., Farach-Colton, M., Yooseph, S., Zhang, L.: Fast, fair, and frugal bandwidth allocation in atm networks. Algorithmica (special issue on Internet Algorithms) **33** (2002) 272–286

[7] Bartal, Y., Byers, J., Raz, D.: Global optimization using local information with applications to flow control. In: the 38th Ann. IEEE Symp. on Foundations of Computer Science (FOCS). (1997)

[8] Kelly, F.P., Maulloo, A.K., Tan, D.K.H.: Rate control for communication networks: Shadow prices, proportional fairness and stability. Operational Research Society **49** (1998) 237–252

[9] Mo, J., Warland, J.: Fair end-to-end window-based congestion control. IEEE/ACM Transactions on Networking **8** (2000) 556–567

[10] Hochbaum, D.S.: Approximation Algorithms for NP-Hard Problems. PWS Publishing Company (1997)

[11] Vazirani, V.V.: Approximation Algorithms. Springer-Verlag (2001)

[12] Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms. MIT Press, Cambridge, Massachusetts (2001)

[13] Shahroki, F., Matula, D.W.: The maximum concurrent flow problem. Journal of the ACM **37** (1990) 318–334

[14] Plotkin, S.A., Shmoys, D.B., Tardos, E.: Fast approximation algorithms for fractional packing and covering problems. Mathematics of Operations Research **20** (1995) 257–301
[15] Cheriyan, J., Karloff, H., Rabani, Y.: Approximating directed multicuts. In: FOCS. (2001) 291–301
[16] Ott, T., Bogovic, T., Carpenter, T., Krishnan, K.R., Shallcross, D.: Algorithm for flow allocation for multi protocol label switching. Research report, Telcordia (2001) Telcordia Technical Memorandum TM-26027.
[17] Mitra, D., Ramakrishnan, K.: A case study of multiservice, multipriority traffic engineering design for data networks. In: IEEE GLOBECOM. (1999) 1077–1083
[18] Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Networks Flows. Prentice-Hall, New Jersey (1993)