Weighted MaxMin fairness Bandwidth Allocation with Maximum Flow Routing

Miriam Allalouf

Joint work with Yuval Shavitt

Tel Aviv University

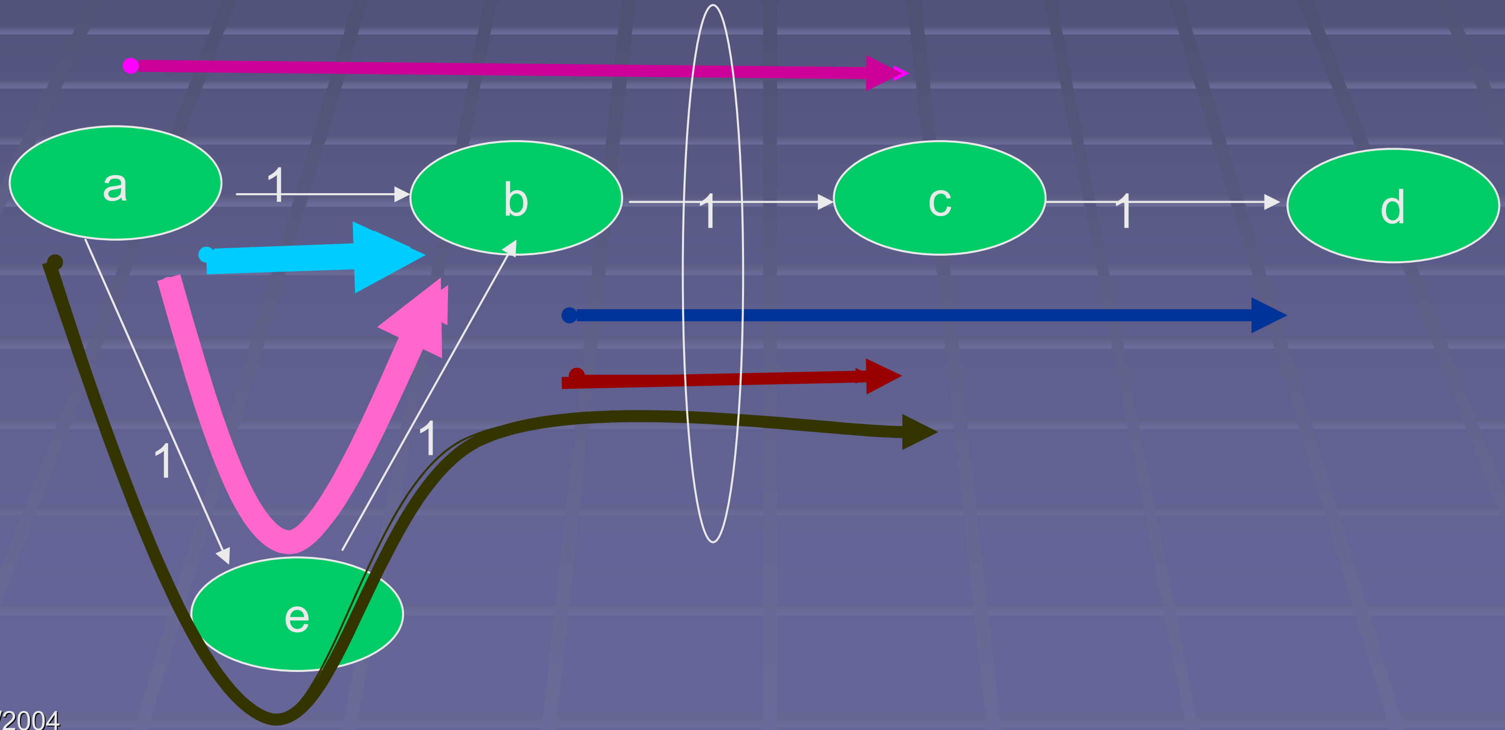
TEL AVIV UNIVERSITY אוניברסיטת תל-אביב

Max Min Fairness

- Fairness:
  - Each consumer gets as much resources as any other
- MaxMin Fairness:
  - Maximize the share of the consumer with the minimum allocation
  - Allow others to get more
  - Repeat this argument repeatedly, namely
  - The consumers that can increase their resource portion without hurting other's will increase it again as any other that can.

Max-Min Fairness: Example

Path rate vector = (1/4,1/4,3/4,3/4,1/4,1/4)



Outline

- Wide Context: Network Design & Traffic Engineering, Bandwidth Allocation
- Weighted Max-Min Fairness Definition
- Related Work
- Model and Maximum Concurrent Multi Commodity Flow Problem
- Weighted Max-min fair Concurrent MCF (WMCM) Algorithm
- Consequent work: WMCM approximation centralized and distributed algorithms

Traffic Engineering Goals

- Traffic Engineering Goals
  - Maximum flow or bandwidth allocation
  - Minimum end-to-end delay.
  - Route assignment
  - Traffic control
  - Service quality differentiation
  - Balanced or fair bandwidth allocation between different aggregates of flows
- Hybrid goal: maximal bandwidth allocation while keeping best possible fairness.

Traffic Engineering Input

The input to the TE algorithm is a list of customers with their associated demands.

Customers :

- Peering points: ingress-egress pairs
- E.g., represent ATM VPs, or MPLS tunnels.
  - aggregates of connections (e.g., TCP) that belong to university campus, business client, client ISP.

Demands:

- Maximum required rate traffic demands, one per pair.

Multi path traffic between ingress-egress pair comprising each demand.

Network Model

- INPUT:
  - Directed graph G=(V,A)
  - Each arc label represents link capacity.
  - Commodity per customer identified by (ingress, egress, demand) triplet.
- Goal
  - Allocate each commodity maximum bandwidth
  - Fulfill clients' demands
  - Keep a fair sharing of the allocated bandwidth
  - Lay the set of paths to be used between each pair in the network,
  - The fairness criterion is defined by the weighted max-min fairness.

Weighted Max-Min Fairness: Definitions

- Commodity Rate Vector, *cr* : its elements are the rates assigned to the commodities.
- Path Rate Vector, f_i : its elements are the rates assigned to a set of paths of commodity *i*.
- The commodity rate is the sum of the rates of all its paths.

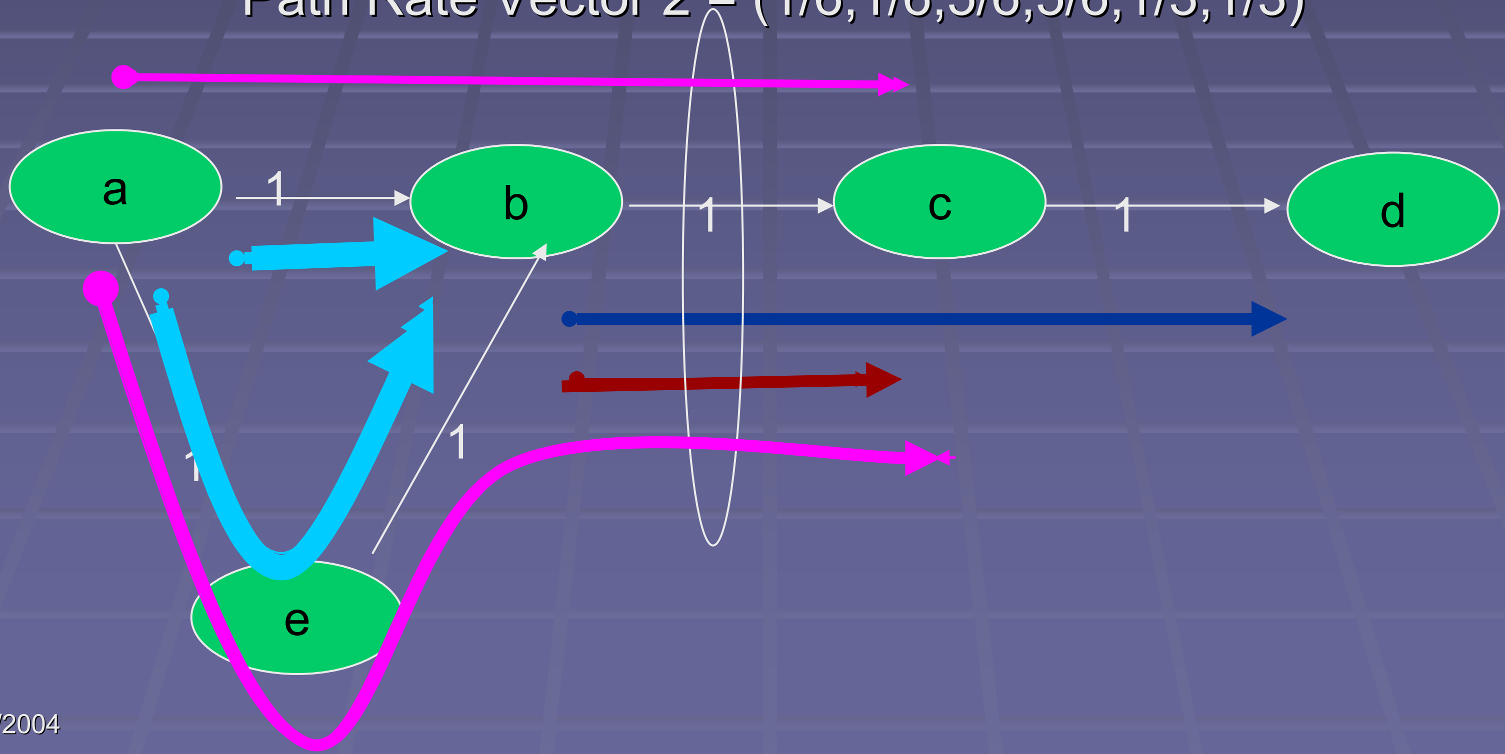
Weighted Max-Min Fair: Example

Commodities s-d (a,b),(a,c),(b,c),(b,d) dem: (1,1,1,1)

Commodity Rate Vector = (5/3,1/3,1/3,1/3)

Path Rate Vector 1 = (1/3,0,2/3,1,1/3,1/3)

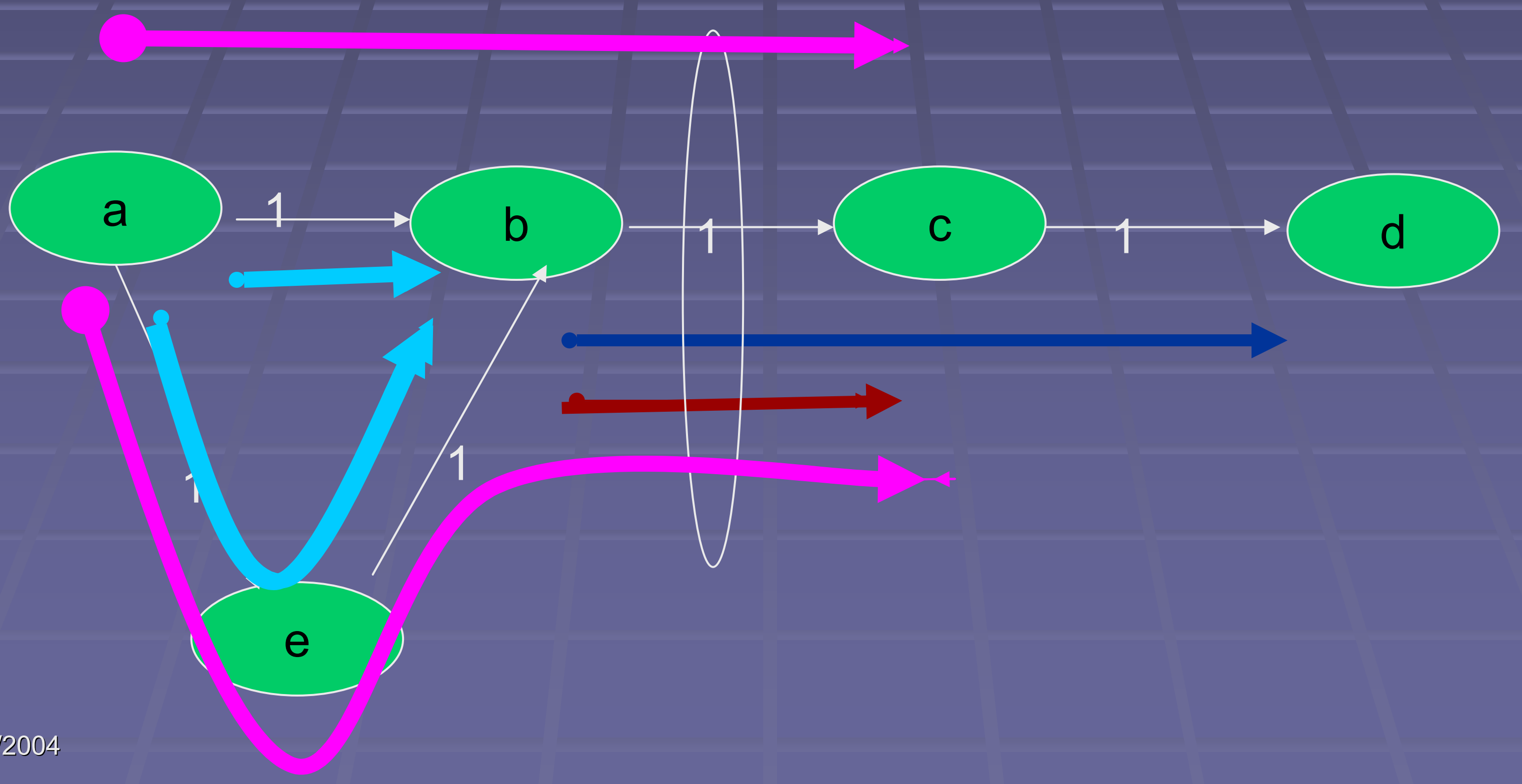
Path Rate Vector 2 = (1/6,1/6,5/6,5/6,1/3,1/3)



Weighted MaxMin Fair: other demands Example

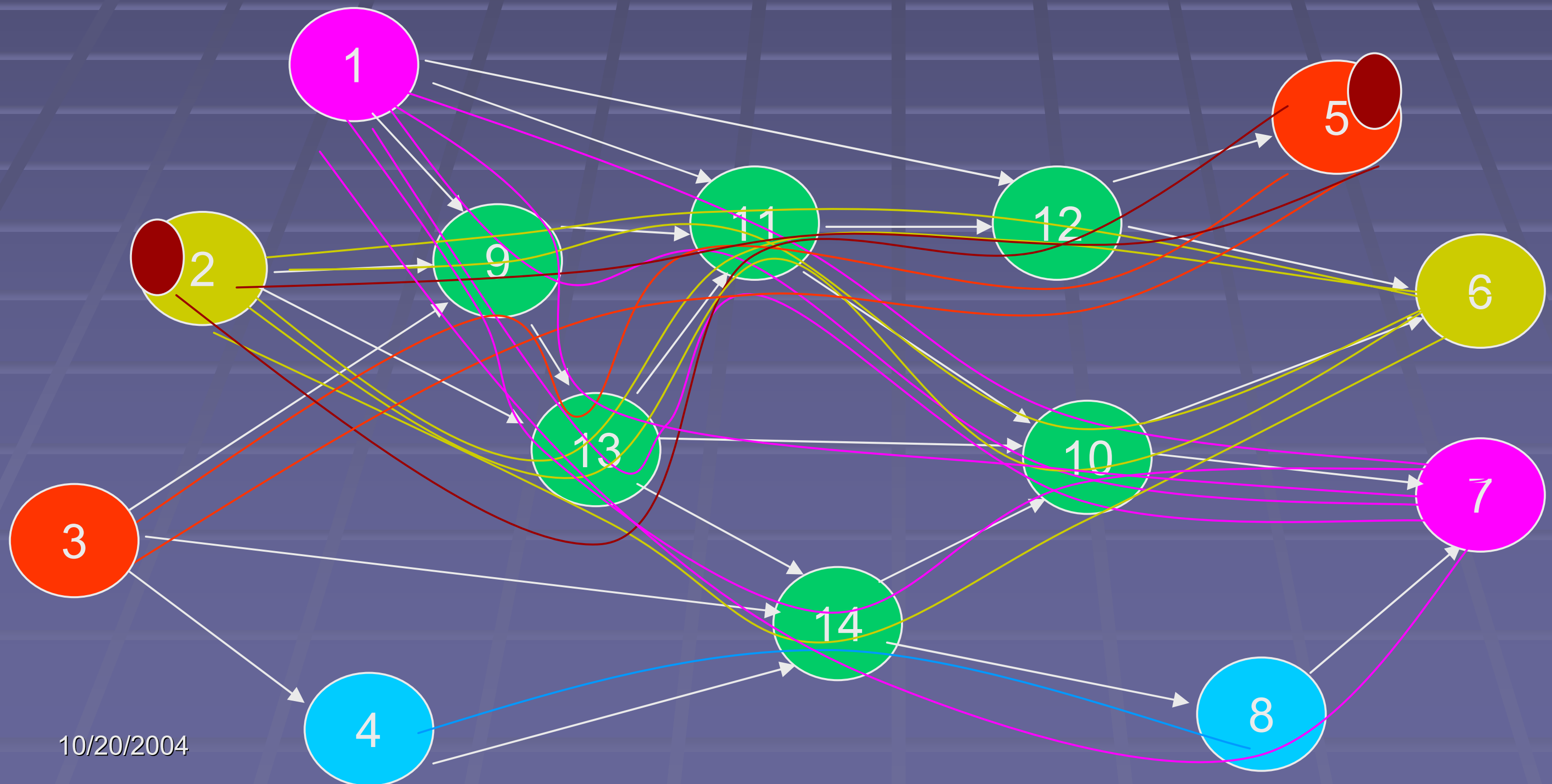
Commodities s-d (a,b),(a,c),(b,c),(b,d) dem: (1,2,1,1)

Commodity Rate Vector = (1/2,3/2,1/4,1/4)



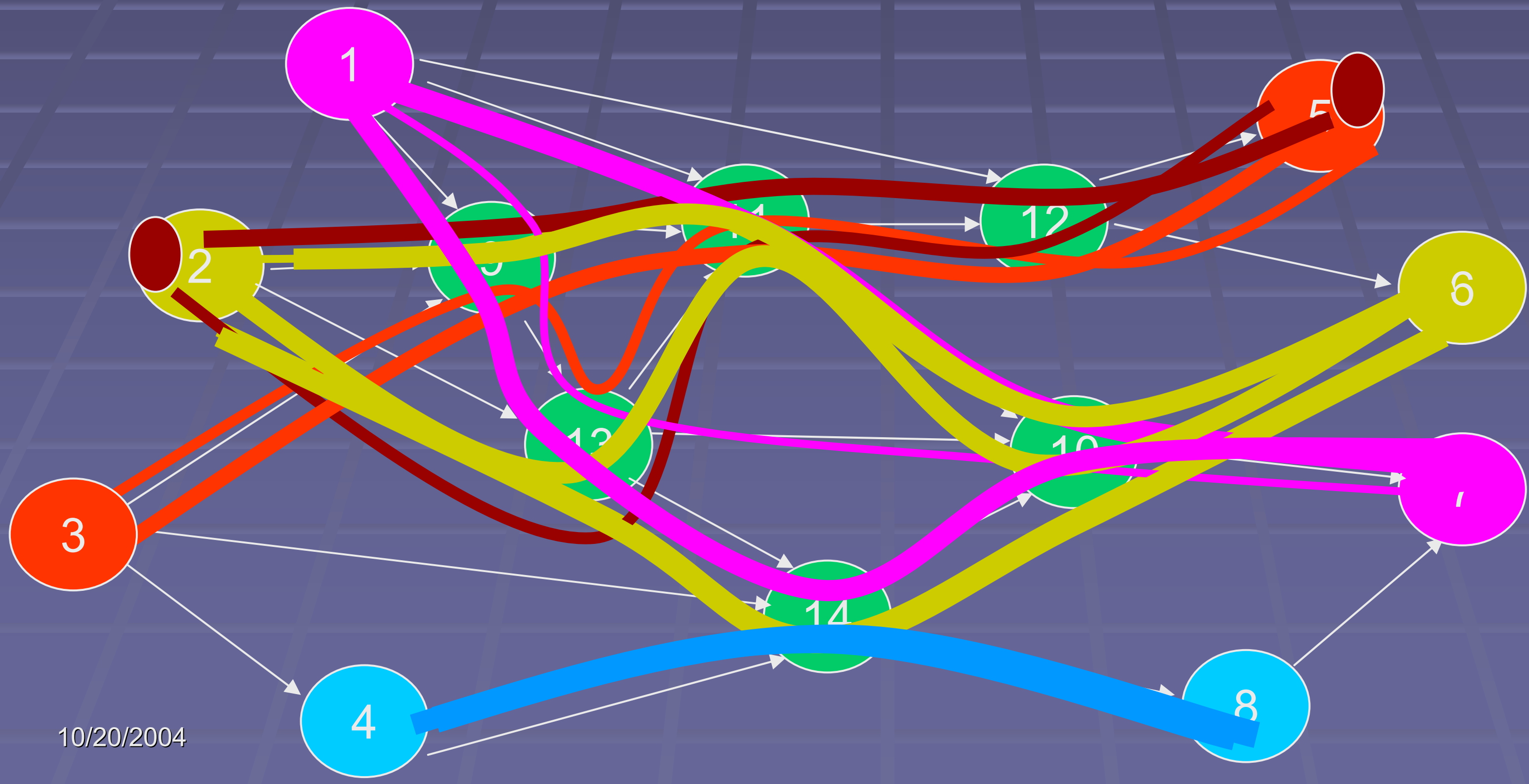
Weighted MaxMin Fair: More complex Example of 15 nodes

5 Commodities s-d (1,7),(2,6),(3,5),(4,8) and (2,6) all with dem: (1,1,1,1,1)



Weighted MaxMin Fair: More complex Example of 15 nodes

Commodity Rate Vector = ((3/4,3/4,1/2,1,1/2)



Weighted Max-Min Fairness: Definitions

- A commodity rate vector *cr* is max-min fair if
  - it is feasible
  - it is maximal
  - each of its elements cr_i cannot be increased without decreasing any other element cr_k for which $cr_i \geq cr_k$
- A commodity rate vector *cr* is weighted max-min fair if
  - it is feasible
  - it is maximal
  - if each of its elements cr_i cannot be increased without decreasing any other element cr_k for which $cr_i /dem_i \geq cr_k/dem_k$
- The two definitions above also hold when traffic may be split to several paths.
- A maximal (weighted) max-min vector may not be unique.

Max-Min Fair
single path per customer

- Jaffe's algorithm finds the max-min fair allocation: (see "Data Networks", D. Bertsekas and R. Gallager,1992)
  - Routing is fixed and given.
- Chen and Nahrstedt [CN98] provide max-min fair allocation routing: selects the best path so the fairness-throughput is maximized upon an addition of a new flow
  - assumes the knowledge of all the possible paths for each new flow.
  - Incremental

Max-Min Fair single path distributed flow control

- Other distributed flow control algorithms deal with dynamic adjustments of flow rates to maintain max-min fairness where routing is given:
  - Awerbuch, B., Shavitt, Y.: Converging to approximated max-min flow fairness in logarithmic time. INFOCOM 1998 .
  - Afek, Y., Mansour, Y., Ostfeld, Z.: Phantom: a simple and effective flow control scheme, Computer Networks, 2000.
  - Bartal, Y., Farach-Colton, M., Yooseph, S., Zhang, L.: Fast, fair, and frugal bandwidth allocation in ATM networks. Algorithmica (special issue on Internet Algorithms), 2002.

Other Fairness Criteria

- Different attitude: The proportional fairness concepts and a convergence algorithm. But again flow allocation without routing:
  - Kelly, F.P., Maulloo, A.K., Tan, D.K.H.: Rate control for communication networks: Shadow prices, proportional fairness and stability. Operational Research Society 1998).
  - Mo, J., Warland, J.: Fair end-to-end window-based congestion control. IEEE/ACM Transactions on Networking 2000

Intermediate

- ✓ Wide Context: Network Design & Traffic Engineering, Bandwidth Allocation
- ✓ Weighted Max-Min Fairness Definition
- ✓ Related Work
- ❑ Model and Maximum Concurrent Multi Commodity Flow Problem
- ❑ Weighted Max-min fair Concurrent MCF (WMCM) Algorithm
- ❑ Consequent work: WMCM Approximation Centralized and Distributed Algorithm

Multi-Commodity Flow (MCF) Problems

- The MCF formulation and solutions are used for traffic engineering problems.
- Linear program formulation: only way to solve the MCF with polynomial number of steps
- Two variants:
  - Maximum Multi-Commodity Flow Problem
  - Maximum Concurrent Multi-Commodity Flow problems

Maximum Multi-Commodity Flow (MCF) Problem

The objective:

maximize the sum of the routed flows (commodities),

subject to:

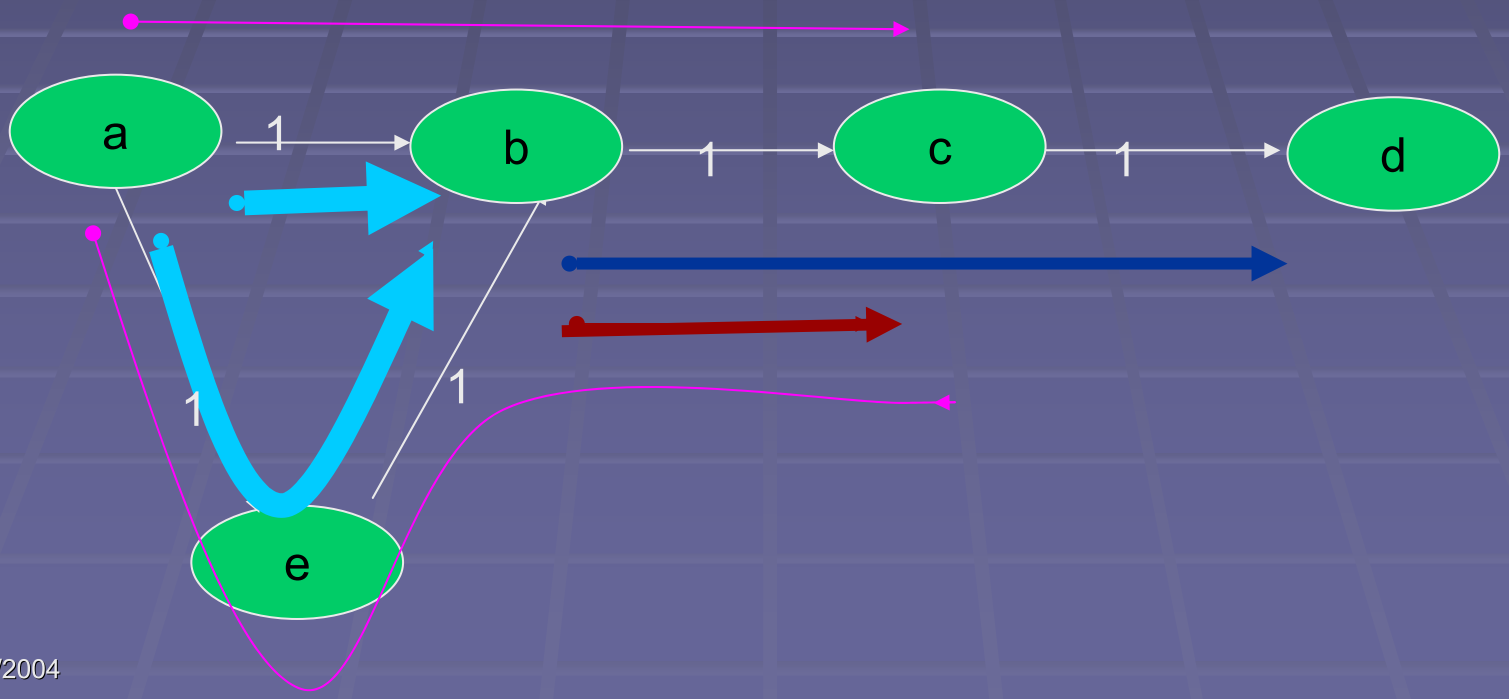
edge capacity constrains.

- MCF Model:
  - Graph G=(V,A), Capacities of A.
  - Multi Commodities represented by (s_i,t_i) – source – destination nodes

Maximum Multi-Commodity Flow (MCF) : Example

Commodities source-destination: (a,b),(a,c),(b,c),(b,d)
  Commodity rate vector = (0,2,1/2,1/2)

Path rate vector = (0,0,1,1,1/2,1/2)



Caveat with MCF

- Not fair:
  - Some flows may get more than others
  - Some flows will get no bandwidth at all

Maximum Concurrent Multi-Commodity Flow (MCF) Problem

The objective:

maximize z, the equal portion of the respective commodity demand which is allocated.

subject to:

edge capacity constrain.

- Model:
  - Graph G=(V,A), arc capacities.
  - Multi Commodities represented by (s_i,t_i,dem_i) – source–destination nodes and commodity demand average rate.

Primal: Maximum concurrent MCF problem – PR'

maximize z

subject to

$$\forall a \in A, \sum_{i=1}^{K} \sum_{P \in P_i} f(P) \leq c(a)$$

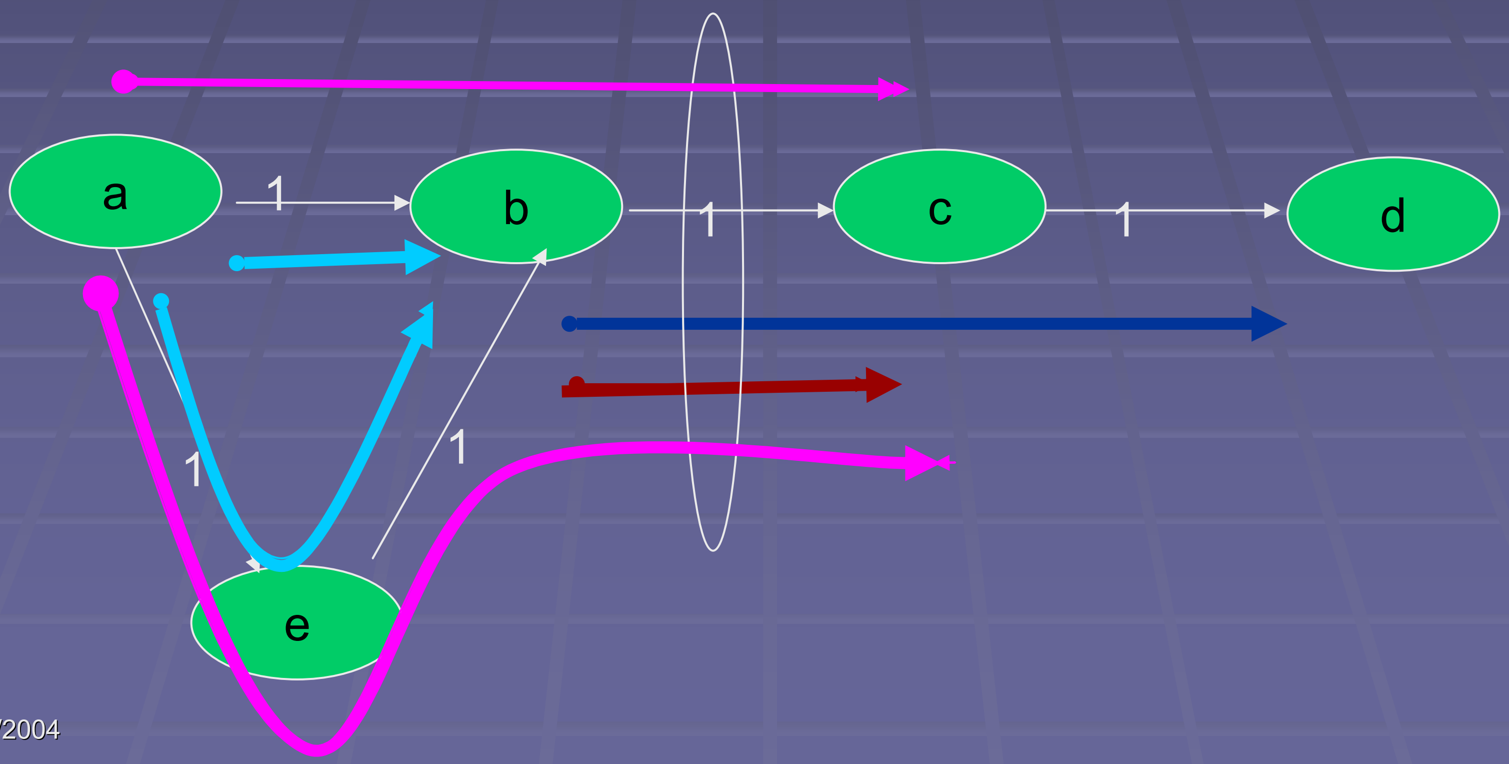
$$\forall i, \sum_{P \in P_i} f(P) \geq z \cdot dem_i$$

$$\forall P \in P_{i=1...K} f(P) \geq 0, z \geq 0$$

Maximum Concurrent MCF Problem : Example

Commodities source-destination: (a,b),(a,c),(b,c),(b,d)
dem: (1,1,1,1)

Commodity Rate Vector = (1/3,1/3,1/3,1/3)



Problems with Concurrent MCF

- It's the communist way:
  - Everyone gets an equal share which may be tiny
  - The network resources are not fully utilized
  - Most of the links may not be saturated

Weighted Max-min fair Concurrent MCF (WMCM) Algorithm

- The WMCM algorithm finds the weighted maxmin rate vector, *cr*, and a set of paths rate vectors, f_k, the rates of the paths P_{kj} in P_k, $k=1..K$ per commodity *k*, composing each cr_i.
- Finds routing
- Allocates bandwidth that
  - Maximizes the network link utilization,
  - Keeps weighted max min fairness criterion.
- The WMCM algorithm is based on the maximum concurrent multi-commodity flow problem.

Weighted Max-min fair Concurrent MCF (WMCM) Algorithm - details

- INPUT:
  - Directed graph G with links capacities, List of commodities and demands.
- WMCM Body:
  - Iterate over the residual graph
    - Solve the maximum concurrent MCF problem on the residual graph.
    - Remove saturated commodities.
    - Recalculated residual demands for the non-sat. commodities
    - Recalculate the residual graph of the unassigned capacities
  - The flow decomposition algorithm is performed once and provides the maximum routing with the weighted max-min fair allocation.

Weighted Max-min fair Concurrent MCF (WMCM) Algorithm - Analysis

- Keeps the weighted fairness criterion at each iteration.
- Each iteration starts with a reduced number of commodities
- Running time is $K*T_{concMCF}$ where $T_{concMCF}$ is the running time of solving maximum concurrent MCF LP.

The WMCM Correctness

Lemma:

At the end of iteration *n*, if two commodities *i* and *j* are unsaturated (can still grow) → $cr_i^n / dem_i = cr_j^n / dem_j$

Theorem:

The commodity rate vector *cr* provided by the WMCM is weighted maxmin fair → (Show that commodity *i* did not increase its bandwidth on the account of another commodity *j*).

Theorem Proof:

Proof is based on the MCMCF solution at each iteration.

Consider two commodities *i* and *j, let* m_i *and* m_j *be the last iteration i and j* were allocated. *W.l.o.g* $m_i <= m_j$

- Case 1: $m_i = m_j$ → *i* and *j* grew together and fairly.
- Case 2: $m_i < m_j$ → Z_{mi} is the MCMCF solution in iteration m_i. It guarantees that commodity *i* got the maximal bandwidth allocation portion till iteration m_i+1. At start of iteration m_i+1: $cr_i / dem_i = cr_{mi} / dem_j$. The connectivity test shows that commodity *i* is saturated but *j* can grow.

WMCM correctness - Lemma

- Vector acf_n - accumulated net flow $acf_k(s_k, j)$ per commodity k. where s_k is the source of commodity k, at the end of iteration n.
- K_n is a set of the commodities in iteration n.

$\exists$ x(n), y(n) *and* u(n) *such that* $\forall i \in K_n$, :

- $\Delta acf_n^i = y(n) \cdot dem^i$ *(The increased rate is in proportion to the demand)*
- $acf_n^i = u(n) \cdot dem^i$ *(The accumulated rate is in proportion to the demand)*
- $demRes_n^i = x(n) \cdot dem^i$ *(The residual demand is in proportion to the original demand).*

WMCM – Theorem Proof by induction

- *Base step*: n=1 (1st iteration) with acf_1, z_1. For commodities i and j
  - $acf_1 i = z_1 \cdot dem^i$ $acf_1 j = z_1 \cdot dem^j$ →
  - $acf_1 i / acf_1 j = dem_i / dem_j$.
- *Induction assumption*: acf_n is feasible and if for each commodity i, $acf_n i$ cannot be increased without decreasing any other $acf_n j$ for some commodity j
  - $acf_n i / acf_n j \geq demi/demj$.

WMCM – Theorem Proof by induction

- Iteration n + 1: KCOMMn+1 is the set of all the commodities in iteration n+1. KCOMMSAT is the set of all commodities that were saturated before, in one of the previous iterations. We distinguish among three cases for any commodity i and j:

1. Case 1: Both commodities were saturated in the previous iterations, such that i, j ∈ KCOMMSAT . Holds trivially because of the induction assumption.
2. Case 2: Only one of the two commodities was saturated before. Assume that i ∈ KCOMMn+1 and j ∈ KCOMMSAT . Commodity j cannot increase its flow since it was deleted from the list. If it was deleted in the previous iteration, n, then acfn i /acfnj = demi/demj holds before starting iteration n + 1, and thus any increase in commodity i rate will imply acfni /acfnj > demi/demj . If j was deleted before the previous iteration,n, we know that acfni /acfnj > demi/demj and then any increase in i's rate will keep the relation.
3. Case 3: Both commodities participate in iteration n+1, thus, i, j ∈ KCOMMn+1.Since both commodities participated in all the previous iterations, they gained rates such that acfni /acfnj = demi/demj . As proved in lemma 1, the gain increase in this iteration keeps the same relation between the rates such that acfn+1i /acfn+1j =demi/demj .

- Finally, KCOMMSAT is reduced in each iteration which ensures termination.

Consequent work (submitted)

- Linear program runtime can be large, though polynomial.
- WMCMApprox : an ε-approximation algorithm that is based on WMCM ideas, solves the problem faster and more practical
- Two types of algorithms were developed: centralized and distributed.

WMCMApprox Approximation Algorithm

- An FPTAS approximation algorithm.
- WMCMApprox algorithm extends primal-dual techniques to achieve a solution to the weighted max-min fair problem.
- Two types of algorithms were developed: centralized and distributed.

WMCMApprox Algorithm

The WMCMApprox algorithm computes a $(1-\varepsilon)^{-3}$ -approximation to the max-min fair flow in time $O(\varepsilon^{-2} Km^2)$ where *m* is the number of edges and *K* is the number of the commodities.

The algorithm was implemented using MATLAB

Summary

- ✓We extended the MaxMin fairness criterion to the case of multi-path routing.
- ✓We presented a polynomial algorithm for the weighted MaxMin fairness problems
  - ✓Demands are taken as weights
  - ✓The algorithm finds both routing and bandwidth allocation
- ✓Approximation

Please help us map the Internet

Download the DIMES agent from:

www.netdimes.org