HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Engineering Physics and Mathematics

Vesa Timonen

Simulation Studies on Performance of Balanced Fairness

Master's thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology

Supervisor: Professor Jorma Virtamo
Instructor: Professor Jorma Virtamo

Espoo, 28th October 2003

Author:	Vesa Timonen
Department:	Department of Engineering Physics and Mathematics
Major:	Systems and Operations Research
Minor:	Teletraffic Theory

Title:	Simulation Studies on Performance of Balanced Fairness
Date:	28^{th} October 2003 **Number of pages:** 63
Chair:	S-38 Teletraffic Theory
Supervisor:	Professor Jorma Virtamo
Instructor:	Professor Jorma Virtamo

Most traffic in current data networks is elastic, i.e. the rates of traffic flows adjusts to use all bandwidth available. Concurrent flows compete for the finite resources or capacity of a network and the rate allocated for flows has to be regulated by some control mechanism to avoid congestion and to reduce packet losses in the network.

One essential objective of the bandwidth sharing policies is to assure fairness of the realized rate allocation. Different fairness criteria favor or discriminate sources or traffic classes on different basis. As a mathematical notion fairness can be generalized to an optimization problem. These classical, utility-based fairness criteria are considered in a static network scenario. In a dynamic network scenario using the optimal bandwidth sharing policy adapted from a static scenario can lead to non-optimal results. Also analysis of flow-level characteristics becomes difficult excluding most simple network cases. Utility-based fairness criteria have been proven to be sensitive in the sense that the steady state distribution depends on detailed traffic characters.

Balanced fairness is a new allocation policy that can be considered as the most efficient insensitive allocation. When bandwidth allocation is based on balanced fairness, the distribution of the number of flows in progress and expected throughput depend only on the average traffic load of each flow class. In some cases the exact probability distribution of the number of concurrent flows of different flow classes can be calculated and performance metrics can be evaluated.

In this thesis the utility-based fairness criteria and their generalization to optimization problem is presented. Also the notion of balanced fairness and its main features are described. The effect of allocation policies on flow-level characters is studied via simulations under different network topologies. Three different allocation policies are used in simulations – balanced, max-min and proportional fairness.

In all the cases examined the differences in throughputs provided by different fairness criteria were comparatively small. Generally max-min fairness provided better throughput on the long routes and penalized the shorter ones more than balanced fairness. It was verified that proportional fairness coincides with balanced fairness in homogenous hypercubes, and with max-min fairness in trees. The insensitivity of balanced fairness was verified via sensitivity simulations. Also the sensitivity of max-min fairness and proportional fairness seem to be quite weak. Results attained via simulations follow exactly the analytical results.

Keywords: Fairness criteria, Balanced fairness, Throughput, Simulation

Tekijä: Vesa Timonen

Osasto: Teknillisen fysiikan ja matematiikan osasto
Pääaine: Systeemi- ja operaatiotutkimus
Sivuaine: Teleliikenneteoria

Otsikko: Tasapainotetun reiluuden suorituskyvyn arviointi simuloimalla
Päivämäärä: 28.10.2003 **Sivumäärä:** 63

Professuuri: S-38 Teleliikenneteoria
Työn valvoja: Professori Jorma Virtamo
Työn ohjaaja: Professori Jorma Virtamo

Nykyisten tietoverkkojen liikenne on valtaosin ns. elastista liikennettä. Liikennelähteet säätävät lähetysnopeutensa siten, että kaikki käytössä oleva kapasiteetti tulee hyödynnettyä. Samanaikaiset vuot kilpailevat verkon rajallisista resursseista. Ruuhkautumisen ja pakettihäviöiden estämiseksi lähteiden nopeutta on kontrolloitava jollakin ruuhkanhallintamekanismilla.

Ruuhkanhallintamekanismien yksi keskeinen tarkoitus on taata toteutuvan kaistanjaon reiluus. Eri reiluuskriteerit suosivat tai syrjivät liikennelähteitä tai liikenneluokkia eri perustein. Reiluuden käsite voidaan yleistää optimointiongelmaksi, jossa pyrikimys on löytää kriteerikohtaisen kohdefunktion maksimoiva tai minimoiva kaistanjako. Nämä klassiset utiliteettipohjaiset reiluuskriteerit rakentuvat staattisen verkkomallin varaan. Dynaamisessa verkkoskenaariossa staattisessa mallissa optimaalinen kaistanjako saattaa kuitenkin johtaa epäedulliseen tulokseen. Myös vuotason tunnuslukujen tarkastelu on vaikeaa joitakin yksinkertaisimpia verkkotopologioita lukuunottamatta. Utiliteettipohjaisten reiluuskriteerien tapapainojakauma on riippuvainen liikenteen tunnusluvuista, mikä tekee vuotason tarkastelut vaikeaksi.

Tasapainotettu reiluus on uusi kaistanjakomenetelmä, jota voidaan pitää tehokkaimpana insensitiivinä kaistanjakona. Kaistanjaon noudattessa tasapainotettua reiluutta aktiivisten voiden lukumäärän jakauma sekä läpäisyn odotusarvo riippuvat vain jokaisen vuoluokan keskimääräisestä kuormasta. Joissakin tapauksissa nämä suureet voidaan laskea eksaktisti.

Tässä työssä esitellään keskeisimmät utiliteettipohjaiset reiluuskriteerit sekä niiden yleistys optimointiongelmaksi. Tasapainotetun reiluuden käsite sekä sen ominaisuuksia esitellään. Kaistanjakomenetelmien vaikutusta vuotason suureisiin tutkittiin simuloimalla eri verkkotopologiossa sekä pyrittiin verifioimaan kirjallisuudessa esitettyjä tuloksia. Käytettyjä menetelmiä olivat tasapainotettu, suhteellinen sekä max-min-reiluus.

Kaikissa tarkastelluissa tapauksissa eri kriteerien tuottamat erot läpäisyssä olivat melko vähäiset. Yleisesti ottaen max-min-reiluus suosii pitkiä vuoluokkia tasapainotettua reiluutta enemmän. Simulaatiot tukivat lauseita, joiden mukaan suhteellinen reiluus yhtyy tasapainotettuun reiluuteen homogeenisissa hyperkuutioissa ja vastaavasti max-min-reiluuteen puutopologioissa. Simuloinnit vahvistavat tasapainotetun reiluuden insensitiivisyyden. Myös max-min-reiluuden sekä suhteellisen reiluuden sensitiivisyys oli vähäistä. Simuloidut tulokset vastasivat tarkasti analyyttisia.

Avainsanat: Reiluuskriteerit, tasapainotettu reiluus, läpäisy, simulointi

Acknowledgements

This Master's thesis was written in the Networking Laboratory of Helsinki University of Technology for the FIT project during the year 2003.

First of all, I would like to thank the supervisor and instructor of the thesis, *Professor Jorma Virtamo* for his guidance and patience during the process. *Ph.D. Samuli Aalto* deserves my gratitude for all the help he provided me during the preliminary study that created template for this thesis.

I wish also to thank all the people in the laboratory for creating an enjoyable atmosphere to work. And special credits for indirect contribution go to ... *Mr. Timo Viipuri* for good kicks and punches in the boxing ring, and *Mr. Jari Huttunen* for nice drops and smashes at the badminton court!

Finally, I would like to express my gratitude to my parents and especially to my sister *Elina*, for everything you have provided me thus far.

Espoo, 28th October 2003

Vesa Timonen

Contents

Chapter 1

Introduction

1.1 Background

Most traffic in current data networks is elastic, i.e. the rates of traffic flows adjusts to use all bandwidth available. Concurrent flows compete for the finite resources or capacity of a network and the rate allocated for flows has to be regulated by some control mechanism to avoid congestion and to reduce packet losses in the network. Because of the finite resources, the bandwidth share or, equally, the rate allocation is a compromise that should be fair, which leads to the concept of *fairness*.

The notion of fairness has no unique definition. It may depend on several different session priorities and service requirements, e.g. a session can require a minimum guaranteed rate for sending data or has a maximum on allowed network delay. It is generally accepted that traffic with the same priority should be treated equally. The simplest definition is to allocate the same share to each connection. Different fairness criteria favor or discriminate sources or traffic classes on different basis. The objective can be to use the network capacity as efficiently as possible without considering a single source (the throughput maximization [2, 10, 19, 25]), or on the contrary, the goal can be to ensure as equal sharing of the resources as possible (max-min fairness [2, 10, 19, 25]).

As a mathematical notion fairness can alternatively be thought of as an optimization problem, where the objective is to find a rate allocation that minimizes or maximizes a utility function specific for the used fairness criterion [10, 11, 16]. In this approach e.g. a cost for achieved rate allocation can be easily added to the examination. Further, the optimization approach provides a generalization of the concept of fairness.

The optimal allocation provided by some fairness criteria is considered in a static network scenario where number of flows is fixed. However, in a dynamic network scenario using the optimal bandwidth sharing policy adapted from a static scenario can lead to non-optimal results concerning, e.g. the throughput

of different flow classes. In most cases these allocation policies are sensitive and necessitate that traffic characteristics are known in detail (e.g. distributions of flow sizes and interarrival times). Also analysis of flow-level characteristics becomes difficult excluding most simple network cases [4]. Utility-based fairness criteria have been proven to be sensitive in the sense that the steady state distribution depends on detailed traffic characters, which explains the difficulty of flow-level analysis [6].

Insensitivity is a desirable property of an allocation policy. Knowing the mean values of distributions of traffic characteristics provides enough information to derive the flow-level characteristics in a fixed network scenario. Balanced fairness [6] represents a new allocation policy that can be considered as the most efficient insensitive allocation. When bandwidth allocation is based on balanced fairness, the distribution of number of flows in progress and expected throughput depend only on the average traffic load of each flow class.

With balanced fairness it is sometimes possible to calculate the exact probability distribution of the number of concurrent flows of different flow classes, and further evaluate performance metrics.

1.2 Objectives

Objectives of this thesis can roughly be divided into two parts. Firstly, our aim is to discuss the notion of fairness and the present different fairness criteria. In the same context, the generalized fairness criteria of these utility-based fairness criteria is presented as an optimization problem. Focus is set on a survey of the concept of balanced fairness. We describe the main features of this allocation policy and present some analytical results.

Secondly, we study the effect of allocation policies on flow-level characters via simulations under different network topologies. Three different allocation policies are used in simulations – balanced, max-min and proportional fairness.

Our objective is to examine the throughputs, compare policies used and verify results presented in the literature. Also the sensitivity of max-min and proportional fairness and insensitivity on balanced fairness is investigated.

As a new aspect, we study the flow-specific throughputs in comparison to the throughput that is defined as a quotient of expected flow size and expected flow duration. We also examine the flow durations and statistics of this distribution, including notably the variance of the flow duration.

1.3 Structure of the thesis

The thesis is organized as follows: in Chapter 2 the general network model is presented and main features and simplifications are listed. In Chapter 3 the concept of fairness is discussed in general and the most common static fairness criteria are defined. The optimization problem that generalizes the utility-based fairness criteria is presented. Chapter 4 introduces the concept of balanced fairness. Main assumptions and analytical results are presented.

In Chapter 5 the simulation setups are described and implementation is discussed in brief. In Chapter 6 simulation results are presented and analyzed. Comparison to analytical results is made.

Conclusions are drawn and some further work considerations are presented in Chapter 7.

Chapter 2

Flow level model of network

This Chapter presents the most important simplifications and assumptions made concerning the traffic and its characteristics. Also a mathematical model of general network model, basic restrictions, feasibility and traffic conditions, and user performance are studied. This Chapter is based on the model presented in [4, 5, 6, 7, 8].

2.1 Traffic assumptions

Most of the traffic in current data networks is elastic and is composed of flows transferring digital documents such as files or web pages. This traffic takes place under the control of transmission control protocol (TCP) that controls the sending rates in trying to prevent network congestion.

TCP is carried out on packet level, the control mechanism (i.e. TCP-agent) in the end system measures packet round trip times (RTT) and controls the sending rate of each concurrent flow. The sending rate does not depend only on the resources in use and the amount of flows in progress, but also on a multiplicity of different parameters, e.g. TCP parameters and version and queueing policies in bottleneck links.

Elastic traffic is not time critical at the packet level, i.e. the transfer time of the document does not have a stringent time limit. Also the flow level time-scale is notably long in comparison to packet-level time-scale. Thus, in flow-level considerations the effects of the packet-level time-scale variations are negligible. In this study we focus on flow-level behavior and characteristics, which justifies following simplifications:

- Packet level observations are discarded.
- Assume a fluid model; the flow is transferred through the network as a continuous stream; no storing of data in links or queues.

- Propagation delays discarded; changes in network state are immediate; no delays at transfers.
- When a flow starts, it is immediately received at the destination at the same rate as it is being sent.

2.2 Network model

A network is considered as a set of links $\mathcal{L}$, where each link $l \in \mathcal{L}$ has a finite capacity $C_l > 0$. A random number of flows compete for the access to these links. There are K flow classes, each class k uniquely identified by a route r_k. Let us denote the set of flow classes, or equally, the set of all routes with $\mathcal{R}$. Each route r_k is a non-empty subset of the set $\mathcal{L}$. Let us denote $r \in l$ if route $r \in \mathcal{R}$ traverses through link $l \in \mathcal{L}$, and conversely, $l \in r$ when link l belongs to the set of links used by route r.

A single flow is characterized by the volume of information to be transferred, i.e. the flow size, on the flow's route. The duration of this transfer depends on the flow rate that varies in time, i.e. rate increases when other flows cease and vice versa. Relation between the flow size s and the time t_{end} when flow is transferred on route r is as follows:

$$s = \int_{t_{\text{start}}}^{t_{\text{end}}} c(t)dt,$$

where t_{start} is the arrival time of the flow and $c(t)$ denotes the flow rate at time t, i.e. the capacity allocated to this flow on each link $l \in r$ at time t, $t \in [t_{\text{start}},\ t_{\text{end}}]$. The allocated capacity $c(t)$ is additionally limited by condition $c(t) \leq C_l$ for all $l \in r$.

The state of the network is denoted by $x = (x_1, \ldots, x_K)$, where x_k is the number of active flows of class k. The aggregated capacity ϕ_k is the capacity allocated for all flows of class k. This capacity allocation depends only on the bandwidth sharing policy and the network state x. Within a class k the capacity ϕ_k is shared equally between flows, i.e. each flow of class k is given the capacity of ϕ_k/x_k.

The capacity allocation $\phi = (\phi_1, \ldots, \phi_K)$ is considered feasible when the following condition holds:

$$\sum_{k:l \in r_k} \phi_k(x) \leq C_l \quad \forall l \in \mathcal{L}. \tag{2.1}$$

That is, the aggregated capacity of flow classes traversing through link l may not exceed the link capacity C_l. Let us denote the set of all feasible allocations by $\mathcal{F}$, i.e. $\mathcal{F} \stackrel{\text{def}}{=} \{\phi \mid \phi \text{ is feasible}\}$.

Pareto-efficiency [4, 5, 6, 8] of a capacity allocation is defined as follows:

$$\forall k \text{ s.t. } x_k > 0 \; \exists l \in r_k, \quad \sum_{k':l\in r_{k'}} \phi_{k'}(x) = C_l. \tag{2.2}$$

That is, every active flow traverses through at least one saturated link.

The traffic intensity of class k is denoted by ρ_k, which corresponds to mean volume of information offered by flows of class k per unit of time. The traffic conditions are given by the inequalities:

$$\sum_{k:l\in r_k} \rho_k(x) \leq C_l, \quad l \in \mathcal{L}. \tag{2.3}$$

The mean time of flow transfer can be considered as an essential performance measure for the users. Thus, the class k user performance can be evaluated in terms of throughput, defined as the ratio of the mean flow size $1/\mu_k$ to the mean flow duration s_k in the steady state. Assuming network stability and applying Little's formula, the throughput of an arbitrary class k with arrival rate λ_k of flows is

$$\gamma_k = \frac{1/\mu_k}{E[s_k]} = \frac{\lambda_k/\mu_k}{\lambda_k E[s_k]} = \frac{\rho_k}{E[x_k]}, \tag{2.4}$$

that is, the throughput of flows of class k is the ratio of the traffic intensity to the expected number of flows of class k.

In the simplest case, in which the network reduces to a single link and one traffic class, the network is modelled as a processor sharing queue. When the traffic condition (2.3) is realized, that is, $\rho < C$, where ρ is the traffic intensity of the flow class and C the link capacity, the steady state distribution of the number of flows is geometric with mean ρ/C. Thus, the throughput γ is given by

$$\gamma = C - \rho. \tag{2.5}$$

A link can be considered as a processor sharing queue, where the the flows correspond to the customers and the service rate is proportional to the link capacity. All customers in the queue are served with the same rate. For the M/M/1/PS system, the variance of the flow duration T conditioned on the flow size s, as presented in [18, 22], is

$$\mathrm{Var}(T \mid S = s) \; = \frac{2\lambda s}{\mu^2(1-\frac{\lambda}{\mu})^3} - \frac{2\lambda}{\mu^3(1-\frac{\lambda}{\mu})^4}\left(1 - e^{-(\mu-\lambda)s}\right),$$

and the variance of the flow duration is [18, 22]

$$\mathrm{Var}(T) \; = \frac{(2+\frac{\lambda}{\mu})}{\mu^2(1-\frac{\lambda}{\mu})^2(2-\frac{\lambda}{\mu})} = \frac{1}{\mu^2(1-\rho)^2}\frac{2+\rho}{2-\rho}, \tag{2.6}$$

where λ is the arrival rate of flows and $1/\mu$ is the mean flow size.

Chapter 3

Fairness

In this chapter the notion of fairness is discussed in general. Static fairness criteria and the general concept of utility-based allocation policies are presented. Some results are gathered and some drawbacks are highlighted motivating the aim to find insensitive allocation policies.

3.1 Notion of fairness

The notion of fairness has no unique definition. Fairness can be considered as a criterion, and its functionality is usually measured with some flow-level characteristic like throughput.

The main objective of bandwidth sharing is to use all the available bandwidth without violating the constraints and maintain a certain fairness. The achieved fairness depends closely on the used fairness criterion. Different fairness criteria favor or discriminate single sources or whole traffic classes on different basis.

A straightforward objective for bandwidth allocation is to find a feasible rate allocation that maximizes the total throughput [2, 10, 11, 19, 25]. In a way, it would be the most efficient way to use the network resources. However, the throughput maximization can be considered as an unfair allocation policy. Usually it leads to bandwidth share in which some users are given as much resources as possible and others are neglected. For instance, in a linear network topology (Figure 4.2) allocation maximizing the total throughput allocates capacity for the long route only, if all the other routes have no traffic [29].

The fairness of a single rate allocation can be considered to have different value depending on the viewpoint. Can a single source be ignored if the overall performance of the network gets better? Or should the treatment of the sources be as equal as possible, even if it leads to poor overall performance?

3.2 Max-Min fairness

Max-min fairness [2, 10, 11, 19, 25] is the most common definition for the concept of fairness. Its objective is to maximize the minimum of the given bandwidths, i.e. the rate of any source cannot be increased without decreasing the rate of some other source that already has a smaller rate.

Definition 3.2.1 (Max-min fairness). *A rate allocation ϕ is max-min fair if $\phi \in \mathcal{F}$ and*

$$\forall r \in \mathcal{R}\ \exists\ l \in r \text{ such that } \sum_{k \ni l} \phi_i = C_l \text{ and } \phi_r = \max_{k \ni l} \phi_k. \tag{3.1}$$

Definition 3.2.1 is equivalent with the definition that a rate allocation is max-min fair, if every flow class has a bottleneck link [2, 10]. That is, every flow traverses through a saturated link. Thus, max-min fair allocation realizes (2.2) and is pareto-efficient policy.

The bandwidth allocation fulfilling Definition 3.2.1 can be proved to be unique [19, 24, 25].

In the following two different algorithms for calculating the max-min fair rate allocation are described. The first one, a *global algorithm*, necessitates that the network topology and routes and number of flows are known. In practice, however, maintaining global information is impractical, even impossible. The second algorithm presents a *distributed algorithm* that finds iteratively the max-min fairness without any centralized control mechanism and thus can be implemented in actual networks fairly easily.

Global algorithm

If the network and flow classes are known, there is a simple algorithm for computing the max-min fair allocation [2], the so-called filling algorithm.

The idea is to start with zero rate allocation, i.e. all the flows have no bandwidth allocated. In the first step the rate allocation of all the flows is increased equally until either at least one of the links becomes saturated or at least one of the sources reaches its maximum sending rate. Sending rates of these sources are fixed at the reached level.

At the next step, sending rates of all the non-fixed sources and flows are incremented equally as in previous phase. This step is repeated until either all flows traverse through a saturated link or maximum sending rates are reached.

The algorithm is described in detail in Table 3.1. Following notation is used: A^k is the set of links not saturated at the beginning of step k. P^k denotes the set of flows not passing through any saturated link at the beginning of step

Table 3.1: Global algorithm for max-min fair allocation [2].

Initial conditions: $k = 1$, $F_0^l = 0$, $r_p^0 = 0$, $P^1 = P$, and $A^1 = \mathcal{A}$	
1.	$n_l^k \stackrel{\text{def}}{=}$ number of flows p such that $p \in P^k$ and $p \in l$
2.	$\delta^k \stackrel{\text{def}}{=} \min_{l \in A^k} \left(C_l - F_l^{k-1}\right) / n_l^k$
3.	$r_p^k \stackrel{\text{def}}{=} \begin{cases} r_p^{k-1} + \delta^k & \text{for } p \in P^k, \\ r_p^{k-1} & \text{otherwise} \end{cases}$
4.	$F_l^k \stackrel{\text{def}}{=} \sum_{p \in l} r_p^k$
5.	$A^{k+1} \stackrel{\text{def}}{=} \left\{ l \mid C_l - F_l^k > 0 \right\}$
6.	$P^{k+1} \stackrel{\text{def}}{=} \left\{ p \mid p \notin l \text{ for any link } l \in A^{k+1} \right\}$
7.	$k \stackrel{\text{def}}{=} k + 1$
8.	If $P^k = \emptyset$, then stop; else go to 1.

k. n_l^k is the number of flows using link l and belonging to set P^k. This is the number of flows that share the the unused capacity of link l. δ^k denotes the increment of rate added to sessions in P^k at the k^{th} step.

This algorithm was implemented for the simulations described in Chapter 5.

Distributed algorithm

An asynchronous distributed algorithm for optimal rate calculation across the network is here described following the presentation in [13]. A description in detail can be found in [12].

Congestion management is controlled by control packets that contain two fields: a one bit long underloading bit called *u-bit* and a field called *stamped rate* containing the next sending rate estimate. Each source of flow class sends a control packet through flow class specific route and the destination sends it back. Links change the control packet information if needed, and flow class source gets information how to change its sending rate.

Each link observes traffic traversing through and calculates the capacity available per flow, called the *advertised rate*. A flow class source sets the stamped rate to its current sending rate, clears the u-bit and sends the control packet. The link compares the stamped rate with the advertised rate. If the stamped rate is greater than the advertised rate, the link resets the stamped rate to the value of the advertised rate and sets the u-bit. Otherwise, the control packet information remains intact.

When the control packet reaches its destination, it is sent back to the source. At this moment the stamped rate carries the minimum sending rate estimate for the source. If the u-bit is set, some link has decreased the stamped rate

and is a bottleneck link for this flow class. In this case the source reduces its sending rate to stamped rate of the control packet. Otherwise, the source can increase its sending rate.

All links maintain a list of flows traversing through and their last stamped rates, called *recorded rates*. Link $l \in \mathcal{L}$ divides flows into two separate sets, $\mathcal{S}_l^{\mathcal{R}}$ and $\mathcal{S}_l^{\mathcal{U}}$. The first set contains flows that have stamped rate smaller than or equal to the link's advertised rate. These flows are assumed already to have a restricting link. The latter set contains flows that have stamped rate higher than the link's advertised rate and are thus considered as unrestricted flows.

The advertised rate μ_l of link $l \in \mathcal{L}$ is calculated as follows:

$$\mu_l = \frac{C_l - C_l^{\mathcal{R}}}{n_l - n_l^{\mathcal{R}}},$$

where C_l is the capacity of the link l, n_l is the number of flows traversing through link l, $n_l^{\mathcal{R}}$ is the number of flows in the set $\mathcal{S}_l^{\mathcal{R}}$ and $C_l^{\mathcal{R}} = \sum_{r \in \mathcal{S}_l^{\mathcal{R}}} \phi_r$, i.e. the total capacity allocated for flows not constrained by link l.

3.3 Proportional fairness

Proportional fairness was proposed in [16]. In proportional fairness deviation from the fair allocation causes a negative average proportional change.

Definition 3.3.1 (Proportional fairness). *A rate allocation ϕ is proportionally fair if $\phi \in \mathcal{F}$ and*

$$\sum_{r \in \mathcal{R}} \frac{\phi'_r - \phi_r}{\phi_r} \leq 0 \ \forall \ \phi' \in \mathcal{F}. \tag{3.2}$$

The bandwidth allocation fulfilling the Definition 3.3.1 can be proved to be unique [19, 25].

The proportional fairness penalizes long routes more than max-min fairness with tendency to achieve greater total throughput [19, 25, 29].

3.4 Other fairness criteria

In addition to max-min and proportional fairness, there is a large number of different fairness criteria, e.g. throughput maximization [2, 10, 11, 19, 25], the potential delay minimization [19], α-fairness [20], f_A-fairness [15] and f_A^h-fairness [30].

The objective in potential delay minimization is to minimize the time delay needed to complete transfers [19]. The time delay is thought to be inversely proportional to the sending rate of the source. The provided rate allocation is between max-min and proportional fairness allocation.

Throughput maximization, max-min fairness, proportional fairness and potential delay minimization all have a weighted version, in which a single flow or a flow class is given a weight. This leads to different rate allocations.

f_A-fairness and f_A^h-fairness are fairness criteria that are derived to describe the bandwidth sharing policy provided by TCP [14, 15, 30]. In these criteria, the basis for the criterion has been the congestion control based on additive increase and multiplicative decrease algorithm (AIMD). The rate allocation provided by AIMD for TCP connections is between max-min fair and proportional fair allocations. When assuming equal RTTs, the realized fairness is shown to be f_A-fair. Further, when flow classes are assumed to have different RTTs, the realized allocation is f_A^h-fair.

All these classical fair allocation schemes can be formulated as optimization problems maximizing a given utility function. In the case of max-min fairness and proportional fairness the connection between the criteria and the optimization problem were considered in [16, 17]. The concept of α-fairness [20] provides a generalized criterion that realizes different fairness depending on the value of the parameter α. Thus, the α-fairness is essentially an optimization problem with a specific objective function. The following section deals with this representation of fairness in more detail.

3.5 Utility-based fairness

Utility approach is a more general concept of fairness [10, 11]. Every source has a utility function u_s, where $u_s(\phi_s)$ indicates the value to source s of having the rate ϕ_s. Every link $l \in \mathcal{L}$ has a cost function g_l, where $g_l(f_l)$ indicates the cost to the network of supporting an amount of flow f_l on link l. A *utility fair* allocation is defined as solution to following optimization problem:

$$\begin{aligned} &\text{maximize } H(\phi) = \sum_{r\in\mathcal{R}} u_r(\phi_r) - \sum_{l\in\mathcal{L}} g_l(f_l) \\ &\text{subject to } \sum_{k:l\in r_k} \phi_k(x) \le C_l, \quad l = 1, \ldots, L, \end{aligned} \tag{3.3}$$

where

$$f_l = \sum_{k:l\in r_k} \phi_k(x)$$

and $\mathcal{L}$ is the set of all links and $\mathcal{R}$ is the set of all flow classes or routes.

Table 3.2: Relation of value of α and the realized fairness.

Value of α	Realized fairness
$\alpha \to 0$	Throughput maximization
$\alpha \to 1$	Proportional fairness
$\alpha \to 2$	Potential delay minimization
$\alpha \to \infty$	Max-min fairness

Different fairness criteria can be presented with specific utility and cost functions, and the fair rate allocation is found by the above constrained optimization problem. A common cost function g_l guarantees the feasibility of a rate allocation:

$$g_l(f_l) = \begin{cases} 0 & \text{for } f_l \leq C_l, \\ \infty & \text{for } f_l > C_l. \end{cases}$$

Violating feasibility constraint causes an infinite cost and thus forces the rate allocation into the feasible region.

All essential static fairness criteria can be obtained by a proper formulation of the utility function. The following form of the objective function was introduced in [4], based on the criterion introduced in [20]:

$$u_\alpha^r(\phi_r) = \begin{cases} w_r x_r \log\left(\phi_r / x_r\right) & \text{if } \alpha = 1, \\ w_r x_r \dfrac{\left(\phi_r / x_r\right)^{1-\alpha}}{1-\alpha} & \text{otherwise.} \end{cases} \tag{3.4}$$

Using (3.4) a general utility function is defined as follows:

$$u_{\text{Gen}}(\phi) \stackrel{\text{def}}{=} \sum_{r \in \mathcal{R}} u_\alpha^r(\phi_r). \tag{3.5}$$

The optimization problem (3.3) has a unique global optimum as proved e.g. in [29].

With different values of α and w_r different fairness criteria are achieved [4, 20]. Let us consider the case $w_r \equiv 1$. Now, when $\alpha \to 0$, the solution of the optimization problem (3.3) is the allocation that maximizes the throughput. When $\alpha \to 1$ the solution of (3.3) is proportionally fair [20]. In the case $\alpha \to 2$ the solution of the optimization problem (3.3) minimizes the potential delay. In the case $\alpha \to \infty$ the solution of (3.3) produces max-min fair allocation [20]. Weighted versions of proportional fairness and potential minimization are achieved when $w_r \neq 1$.

3.6 Drawbacks

The optimal allocation provided by some fairness criteria is considered in a static network scenario where number of flows is fixed. However, in a dynamic network scenario using the optimal bandwidth sharing policy adapted from static scenario can lead to non-optimal results concerning, e.g. the throughput of different flow classes. In most cases these allocation policies are sensitive and necessitate that traffic characteristics are known in detail (e.g. distributions of flow sizes and arrival times, session structure). Also the analysis of flow-level characteristics becomes difficult excluding most simple network cases [4]. Utility-based fairness criteria have been proven to be sensitive in the sense that the steady state distribution depends on detailed traffic characters, which explains the difficulty of flow-level analysis [6].

Chapter 4

Balanced fairness

The concept of insensitive bandwidth sharing is discussed, necessary and sufficient conditions for insensitivity are presented. Notion of balanced fairness and its main features are described. Some analytical results are presented. This Chapter is based on the definitions and results presented in [4, 5, 6, 7, 8, 9]

4.1 Background

Classical fairness criteria are determined in a static state of network in which no time scale is considered. These allocation policies give allocation for the current state of network. Even though an allocation is optimal in a static scenario, in a dynamic scenario the situation can be the opposite, e.g. maximizing throughput leads instability in dynamical scale when the number of flows tends to infinity [4]. Utility-based allocations are sensitive (steady state distributions depend on detailed traffic characters) and thus an analysis of the flow-level dynamics is difficult.

Bonald and Proutière have introduced an alternative notion of fairness, *balanced fairness* [6]. When bandwidth allocation is based on balanced fairness, the distribution of number of flows in progress and expected throughput depend only on the average traffic load on each flow class. The performance is insensitive to detailed traffic characteristics such as the distribution of flow sizes and think time durations. The name derives from the set of detailed balance relations satisfied by the instantaneous rates allocated to individual flows, which constitute necessary and sufficient conditions for insensitivity in the underlying stochastic networks [26].

With balanced fairness it is possible to calculate the exact probability distribution of the number of concurrent flows of different flow classes, and further evaluate performance metrics.

4.2 Balance property

Let us define vector e_k as unit vector with value 1 in the k^{th} component and 0 elsewhere, $k = 1, \ldots, K$. The *balance property* is defined as follows:

Definition 4.2.1 (Balance property). *Capacity allocation $\phi = (\phi_1, \ldots, \phi_K)$ is balanced if the following equation holds:*

$$\phi_k(x)\phi_{k'}(x - e_k) = \phi_{k'}(x)\phi_k(x - e_{k'}), \quad \forall k, k' \; \forall x : \; x_k > 0, \; x_{k'} > 0. \tag{4.1}$$

Let $\langle x, x - e_{k_1}, \ldots, x - e_{k_1} - \ldots - e_{k_{n-1}}, 0\rangle$ be a direct path from state x to state 0 (Figure 4.1). This path has length $n \equiv |x|$, that is, the length is equal to the number of flows in state x. The balance property (4.1) implies that the expression

$$\Phi(x) = \frac{1}{\phi_{k_1}(x)\phi_{k_2}(x - e_{k_1}) \cdots \phi_{k_n}(x - e_{k_1} - \ldots - e_{k_{n-1}})} \tag{4.2}$$

is independent of the considered direct path [6]. A positive function $\Phi : \mathbb{Z}_+^K \mapsto \mathbb{R}_+$, referred as balance function, defines the capacities unambiguously:

$$\phi_k(x) = \frac{\Phi(x - e_k)}{\Phi(x)}, \quad \forall x : \; x_k > 0. \tag{4.3}$$

Conversely, if there exists a positive function Φ such that the capacities satisfy (4.3), these capacities are balanced.

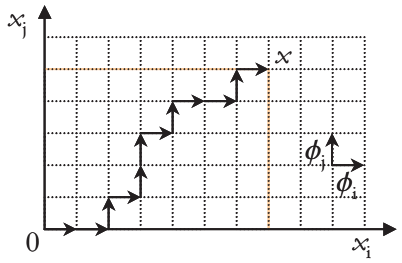


Figure 4.1: Path from state 0 to state x in two-dimensional case.

The balance property (4.1) implies that

$$\frac{\phi_k(x - e_{k'})}{\phi_k(x)} = \frac{\phi_{k'}(x - e_k)}{\phi_{k'}(x)}, \quad \forall x : \; x_k > 0, \tag{4.4}$$

which is an alternative way of writing the balance property. This means that the experienced relative change in allocation of one flow class caused by the removal of a flow of the other class, is equal for all flow class pairs.

In a balanced system the *linear property* holds [7]:

$$\gamma_i = \frac{s}{\mathrm{E}[T|\; S = s]}, \quad \forall s > 0. \tag{4.5}$$

When assuming Poisson flow arrivals and exponential flow size distribution, the *invariant measure* π is shown to be [9]

$$\pi(x_1, \ldots, x_K) = \Phi(x_1, \ldots, x_K)\rho_1^{x_1} \cdots \rho_K^{x_K}. \tag{4.6}$$

The function π corresponds to an invariant measure of the Markov process that describes the evolution of the number of flows of each flow class.

It is shown in [26], that the bandwidth sharing network can be identified with a Whittle network of processor sharing services. Thus, the insensitivity properties of the Whittle networks implies that the flow sizes and think times can have rather general distributions without the requirement of independence, and the invariant measure (4.6) remains valid. Only the Poisson arrivals of the sessions are necessitated.

Sufficient condition for insensitivity

The insensitivity of an allocation is implied by the balance property (4.1). The balance property is the sufficient condition for the insensitivity.

Necessary conditions for insensitivity

Also the converse holds, that is, an allocation is balanced if the invariant measures of the number of flows of each class are insensitive to any traffic character excluding the traffic intensities ρ. The balanced property is implied by each of the following forms of insensitivity [6, 8]:

- *Insensitivity to the flow size distribution*: Assuming Poisson flow arrivals and identically and independently distributed (i.i.d.) flow sizes, the Poisson flow arrival process of any flow class can be changed to any phase-type distribution with the same expected value, and the invariant measures of the process of the number of flows remains unchanged.

- *Insensitivity to the flow arrival process*: Assuming i.i.d. flow sizes, the Poisson flow arrivals process of any flow class can be changed to Poisson session arrivals, and the invariant measures of the process of the number of flows remains unchanged.

- *Time-scale insensitivity*: Assuming Poisson flow arrivals and exponential i.i.d. flow sizes, the flow inter-arrival times and flow sizes of any flow class can be multiplied by the same constant, and the invariant measures of the process of the number of flows remains unchanged.

Any allocation satisfying one of the previous properties is balanced.

4.3 Balanced fairness

There are infinitely many insensitive allocations. In sight of (4.3), each of these allocations is defined by the balance function Φ. Every of these allocations has to be feasible. Thus, (2.1) and (4.3) give

$$\sum_{k:l\in r_k} \Phi(x-e_k) \leq \Phi(x)C_l \quad \forall x \; \forall l \in \mathcal{L},$$

and further,

$$\Phi(x) \geq \frac{1}{C_l} \sum_{k:l\in r_k} \Phi(x-e_k) \quad \forall x \; \forall l \in \mathcal{L}.$$

An unambiguous balanced allocation, denoted as *balanced fairness*, exists satisfying all the capacity constraints (2.1) and at least one with equality at any network state x [5, 6, 8]. The balance function for this allocation is obtained from recursion

$$\Phi(x) = \max_{l\in\mathcal{L}} \left\{ \frac{1}{C_l} \sum_{i:l\in r_i} \Phi(x-e_i) \right\} \tag{4.7}$$

with the initial assumption $\Phi(0) = 1$. Any link realizing the maximum in (4.7) is called *saturated* in state x.

4.4 Throughput calculation

The normalization constant is sum of the invariant measure over all possible states. Let us denote the normalization constant with G. Function G is mapping from set $\mathbb{R}_+^K$ to $\mathbb{R}_+$:

$$\begin{aligned} G(\rho) &= \sum_{x\in\Omega} \pi(x) = \sum_{x_1=0}^{\infty} \cdots \sum_{x_K=0}^{\infty} \pi(x_1,\ldots,x_K) \\ &= \sum_{x_1=0}^{\infty} \cdots \sum_{x_K=0}^{\infty} \Phi(x_1,\ldots,x_K)\rho_1^{x_1}\cdots\rho_K^{x_K}, \end{aligned} \tag{4.8}$$

where traffic load vector is denoted $\rho = (\rho_1,\ldots,\rho_K)$ and $\Omega = \mathbb{Z}_+^K$ is the state space. The function $G(\rho)$ is the generating function of the balance function $\Phi(x)$ and thus contains the same information. That is, function $\Phi(x)$ and the performance measures can be derived from function $G(\rho)$.

Throughput γ_k of class k can now be obtained using the normalization constant. From (2.4) we get

$$\gamma_k = \frac{\rho_k}{E[x_k]} = \frac{G(\rho)}{\frac{\partial}{\partial\rho_k}G(\rho)} = \frac{1}{\frac{\partial}{\partial\rho_k}\log G(\rho)}. \tag{4.9}$$

Calculation of the normalization constant or further, the throughput still remains problematic in most of the network topologies. In [9] an effective way to calculate the normalization constant based on the decomposition of the state space is presented. The normalization constant can also be decomposed to corresponding parts, an further, the normalization constant is sum over these parts. In following, this recursive method is described more in detail.

The state space Ω is decomposed as

$$\Omega = \sum_{k=0}^{K} \sum_{\mathcal{I}_k} \Omega_{\mathcal{I}_k},$$

where sets $\Omega_{\mathcal{I}_k}$ and $\mathcal{I}_k$ are defined as follows: $\mathcal{I}_k \stackrel{\text{def}}{=} \{i_1, \ldots, \imath_k\}$ is a k-tuple of unequal indices such that $1 \leq i_1 < \ldots < i_k \leq N$, $k \leq N$. The set $\Omega_{\mathcal{I}_k} \subset \Omega$ is a k-dimensional set of states such that $\Omega_{\mathcal{I}_k} \stackrel{\text{def}}{=} \{x \mid x_i > 0 \Leftrightarrow i \in \mathcal{I}_k\}$ and $\Omega_{\emptyset} \stackrel{\text{def}}{=} \{(0, \ldots, 0)\}$. Now, the normalization constant $G(\rho)$ can also be decomposed into a sum over partial sums

$$G(\rho) = \sum_{k=0}^{K} \sum_{\mathcal{I}_k} G_{\mathcal{I}_k}(\rho),$$

where $G_{\mathcal{I}_k} \stackrel{\text{def}}{=} \sum_{x \in \mathcal{I}_k} \Phi(x_1, \ldots, x_K) \rho_1^{x_1} \cdots \rho_K^{x_K}$.

For the calculation of the normalization constant $G(\rho)$ the following recursion is derived

$$G_{\mathcal{I}}(\rho) = \frac{\sum_{i \in \mathcal{I}'} \rho_i G_{\mathcal{I} \setminus \{i\}}(\rho)}{C_{\sigma(\mathcal{I})} - \sum_{i \in \mathcal{I}'} \rho_i},$$

where $\mathcal{I}$ is an arbitrary set of flow classes and sets $\sigma(\mathcal{I})$ and $\mathcal{I}'$ are defined as

$$\begin{aligned} \sigma(\mathcal{I}) &\stackrel{\text{def}}{=} \{l \in \mathcal{L} \mid \exists x \in \Omega_{\mathcal{I}} \text{ such that } l \text{ is saturated}\}, \text{ and} \\ \mathcal{I}' &\stackrel{\text{def}}{=} \{i \in \mathcal{I} \mid \sigma(\mathcal{I}) \in r_i\}. \end{aligned}$$

If the set $\sigma(\mathcal{I})$ contains more than one link, any of those can be the basis for the recursion.

For general tree topologies this algorithm is implemented for *Mathematica*-program in Qlib traffic theory function library [23].

4.5 Analytical results

In general, the calculation of the explicit form of the function $\Phi(x)$ is complex and even impossible. Further, the derivation of the normalization constant $G(\rho)$ and the throughputs can turn out to be difficult. In the following sections, some basic network configurations are presented and analytical results in these topologies are gathered from the literature or derived.

4.5.1 Hypercube, grid and line

Line

A n-link linear network, depicted in Figure 4.2, consists of $L = n$ links with capacities C_i and $K = n + 1$ flow classes, where class 0 flows traverse through all the links and class k flows traverse only through link k, $k = 1, \ldots, n$.

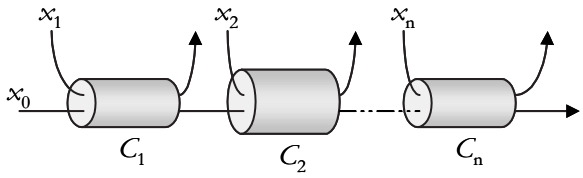


Figure 4.2: Linear network configuration.

Linear network is one the of the most used network topologies in fairness studies. In context of balanced fairness it is considered in [4, 8, 9].

On the contrary to general hypercube, the balance function can be derived for the linear network with unequal link capacities. Let us assume that $\min_{l\in\mathcal{L}} C_l = 1$, i.e. the smallest capacity of the links has value 1. The balance function has now following form [6, 8]:

$$\Phi(x) = \sum_{y:\sum_{l:x_l>0} y_l \le x_0} \prod_{l:x_l>0} \binom{x_l + y_l - 1}{y_l} \frac{1}{C^{x_l+y_l}}.$$

Further, the normalization constant and the throughputs can be derived as shown in Section 4.4. Let us define the minimum link capacity is denoted with C, i.e. $C = \min_{l\in\mathcal{L}} C_l$. The normalization constant is now [9]

$$\begin{cases} G_0(\rho) &= \dfrac{1}{1 - \frac{\rho_0}{C}}, \\ G_i(\rho) &= \dfrac{1 - \frac{\rho_0}{C_i}}{1 - \frac{\rho_0+\rho_i}{C_i}} \cdot G_{i-1}(\rho). \end{cases} \tag{4.10}$$

Applying (4.9) to (4.10) gives the throughput [8, 9]

$$\begin{cases} \gamma_0 &= \left(\dfrac{1}{C - \rho_0} + \displaystyle\sum_{l=1}^{n} \left(\frac{1}{C_l - \rho_l - \rho_0} - \frac{1}{C_l - \rho_0} \right) \right)^{-1}, \\ \gamma_i &= C_i - \rho_i - \rho_0, \end{cases} \tag{4.11}$$

where $i = 1, \ldots, n$.

Proportional fairness has been shown to be insensitive in homogenous linear networks [4].

Grid

An obvious generalization to the linear network configuration is the grid network topology. In the grid network there are two groups of flow classes perpendicular to each other. Flow classes in the same group do not have common links, but each flow class shares one link with every flow class of the other group. Grid network configuration is considered in [4, 5, 6] and is depicted in Figure 4.3.

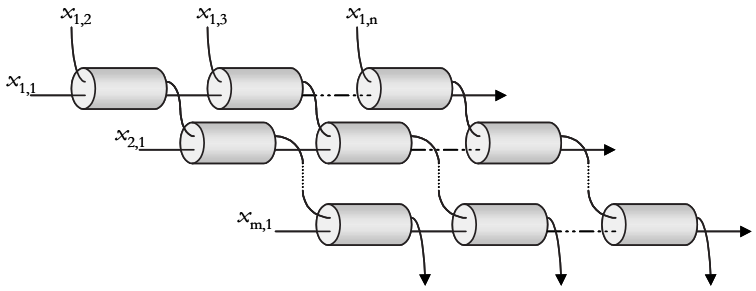


Figure 4.3: Grid network configuration.

For a general grid network the derivation of the balance function or further, the normalization constant and throughput, is difficult. The grid network setup is a special case of a hypercube network.

Proportional fairness has been shown to be insensitive in homogenous grid networks [4].

Hypercube

Hypercube network configuration, considered in [5, 6], is defined as follows:

Definition 4.5.1 (Hypercube). *Hypercube of dimension n is a network of n sets of routes (referred to as directions) such that the set of links is the set of intersections of n routes of different directions.*

By Definition 4.5.1, lines and grids are hypercubes of dimension 2. One-dimensional hypercube reduces to a single link.

Figure 4.4 depicts a three-dimensional hypercube network setup consisting 12 links and 16 flow classes.

In the case of general link capacities, the derivation of analytical formula for the balance function turns out to be difficult. However, assuming unit capacity links, the balance function can be derived.

Let us denote by $\mathcal{D}_1, \ldots, \mathcal{D}_n$ the corresponding directions in a hypercube of dimension n, $n \geq 2$. If the hypercube has unit capacity links, the balance

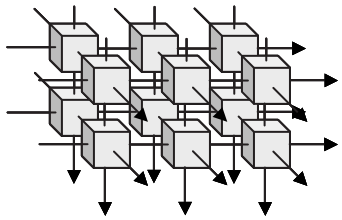


Figure 4.4: Tree-dimensional hypercube configuration.

function is defined as a multinomial coefficient [5, 6]

$$\Phi(x) = \binom{\sum_k x_k}{\sum_{k:r_k \in \mathcal{D}_1} x_k, \ldots, \sum_{k:r_k \in \mathcal{D}_n} x_k}. \tag{4.12}$$

The balanced allocation follows from (4.3):

$$\phi_i(x) = \frac{\frac{(\sum_k x_k)!}{(\sum_{k:r_k \in \mathcal{D}_1} x_k)! \cdots (\sum_{k:r_k \in \mathcal{D}_n} x_k)!}}{\frac{(\sum_k x_k - 1)!}{(\sum_{k:r_k \in \mathcal{D}_1} x_k)! \cdots (\sum_{k:r_k \in \mathcal{D}_i} x_k - 1)! \cdots (\sum_{k:r_k \in \mathcal{D}_n} x_k)!}} = \frac{\sum_{k:r_k \in \mathcal{D}_i} x_k}{\sum_k x_k}.$$

It is proven that proportional fairness coincides with balanced fairness in homogenous hypercubes [6]. This implies the insensitivity of the proportional fairness in homogenous hypercubes. In [6] it is proven that it is the only network topology for which an insensitive utility-based allocation exists.

4.5.2 Hypercycle

Hypercycle configuration, considered in [5, 6], is defined as follows:

Definition 4.5.2 (Hypercycle). *A n-link hypercycle is a network consisting n links and n routes crossing all links except one.*

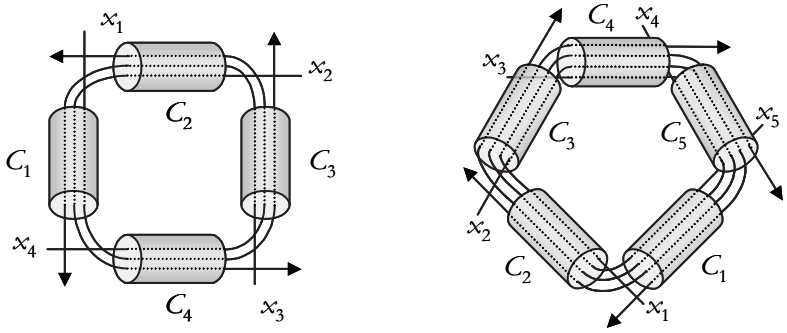
Definition 4.5.2 determines the flow classes unambiguously – a route traversing through $n-1$ of n possible links can be constructed in $\binom{n}{n-1} = n$ ways.

Figure 4.5 depicts two hypercycle network configurations.

The derivation of the explicit balance function for hypercycle topology turns out to be awkward. Even in the simplest three-link case with equal unit capacity links the explicit formula is still to be derived.

Balanced fairness is not Pareto-efficient in hypercycles [5, 6].

In [6] it is proven, that utility-based allocations are not balanced in homogenous hypercycle network topologies.



(a) Configuration with four links.

(b) Configuration with five links.

Figure 4.5: Hypercycle network configurations.

4.5.3 Parking lot

Parking lot network configuration, considered in [9], is illustrated in Figure 4.6. Parking lot network is a special case of tree network topology.

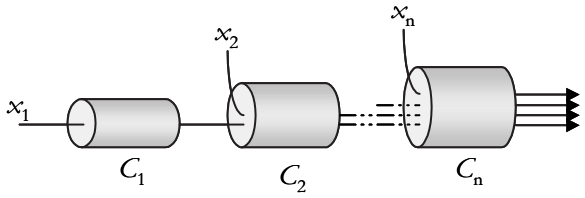


Figure 4.6: Parking lot network configuration.

In n-link parking lot network, there are $L = n$ links with capacities $C_1 \leq \ldots \leq C_n$ and $K = n$ flow classes, where class k flows traverses through the set of links $\{k, \ldots, n\}$, $k = 1, \ldots, n$.

For the parking lot setup the normalization constant is [9]

$$\begin{cases} G_0(\rho) & = \dfrac{1}{1 - \frac{\rho_1}{C_1}}, \\ G_i(\rho) & = \dfrac{1 - \frac{\rho_1 + \cdots + \rho_{i-1}}{C_i}}{1 - \frac{\rho_1 + \cdots + \rho_i}{C_i}} \cdot G_{i-1}(\rho). \end{cases}$$

Denoting the link i load by $R_i = \sum_{j=1}^{i} \rho_j$, we get from the previous

$$G(\rho) = \frac{1}{1 - \frac{R_1}{C_1}} \prod_{i=2}^{n} \frac{1 - \frac{R_{i-1}}{C_i}}{1 - \frac{R_i}{C_i}}. \tag{4.13}$$

Now, applying (4.9) to (4.13) gives the throughput [9]

$$\gamma_i = \left(\frac{1}{C_i - R_i} + \sum_{l=i+1}^{n} \left(\frac{1}{C_l - R_l} - \frac{1}{C_l - R_{l-1}} \right) \right)^{-1}, \tag{4.14}$$

where $i = 1, \ldots, n$.

4.5.4 Trees

A general n-level tree network consists of L links and K flow classes. Let us denote the link of route r_i at level k with l_i^k. Link capacities fulfill the condition $C_i^k \leq C_j^{k-1}$, where C_i^k is the capacity of the link l_i^k. All the routes r_i share the trunk link l^0, i.e. $l_l^0 \equiv l^0$. The following condition assures the tree structure: if $l_i^k = l_j^k$ for flow classes i and j, then $l_i^{k'} = l_j^{k'}$ for all $k' = 0, \ldots, k$.

Tree network configurations are considered in [6, 8, 9], and further, the derivation of the normalization constant and throughputs in [9]. For normalization constant and throughput calculation Qlib traffic theory function library [23] provides useful tools to be used within Mathematica.

The derivation of the normalization constant and the throughput for an arbitrary tree can be carried out as described in Section 4.4. In following we set the focus on *concentration trees*, two-level trees having K flow classes and $K + 1$ links (Figure 4.7).

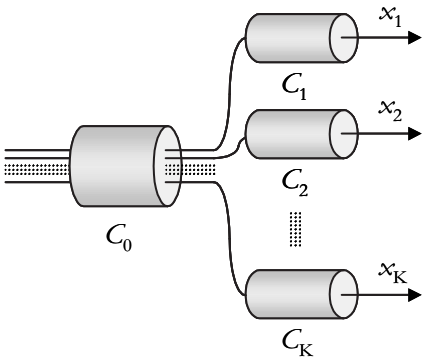


Figure 4.7: Concentration tree network configuration.

Let us assume that the trunk has capacity $C_0 = 1$ and further, that $C_l \leq 1$ for all $l = 1, \ldots, K$ and the equality holds at most for one l. If $\sum_{r:x_r>0} C_r < 1$, we have [6, 8]

$$\Phi(x) = \prod_{r \in \mathcal{R}} \frac{1}{C_r^{x_r}},$$

and otherwise

$$\Phi(x) = \sum_{\substack{z: z_r \le x_r \forall r \\ I(z) \neq \emptyset}} \binom{\sum_r (x_r - z_r) - 1}{\sum_{r \notin I(z)} (x_r - z_r)} \binom{\sum_{r \notin I(z)} (x_r - z_r)}{x_r - z_r, r \notin I(z)} \binom{\sum_{r \in I(z)} (x_r - z_r)}{x_r - z_r, r \in I(y)} \prod_{r \in \mathcal{R}} \frac{1}{C_r^{z_r}},$$

where z is a K-dimensional vector of integers and set $I(z)$ is defined as follows:

$$I(z) \stackrel{\text{def}}{=} \begin{cases} \emptyset, & \text{if } \sum_{r: z_r > 0} C_r > 1, \\ \{r\}, & \text{if } z_r = 0,\ x_r > 0 \text{ and } C_r + \sum_{r': z_{r'} > 0} C_{r'} > 1. \end{cases}$$

For a two-branch tree the balance function reduces to form [6, 8]

$$\Phi(x) = \sum_{z_r \le x_r} \binom{x_1 - z_1 + x_2 - 1}{x_1 - z_1} \frac{1}{C_1^{z_r}} + \binom{x_1 + x_2 - z_2 - 1}{x_2 - z_2} \frac{1}{C_2^{z_2}}.$$

For the two-branch tree, let us assume that $C_1 \le C_0$, $C_2 \le C_0$ and $C_1 + C_2 > C_0$[1]. The normalization constant is [9]

$$G(\rho) = \frac{1}{1 - \frac{\rho_1 + \rho_2}{C_0}} \left(\frac{1 - \frac{\rho_1}{C_0}}{1 - \frac{\rho_1}{C_1}} + \frac{1 - \frac{\rho_2}{C_0}}{1 - \frac{\rho_2}{C_2}} - 1 \right),$$

and applying (4.9) to previous gives

$$\gamma_i = \frac{C_i (C_0 - \rho_1 - \rho_2)(1 - \frac{\rho_i}{C_i})^2 G(\rho)}{C_0 - C_i + C_i (1 - \frac{\rho_i}{C_i})^2 G(\rho)}, \quad i = 1, 2. \tag{4.15}$$

For a concentration tree with more than two branches, the examination of the normalization has to be divided based on how the sums of link capacities over different link subgroups relate to capacity of the trunk link.

For future purposes, let us examine the case of three-branch tree with such link capacities that $C_i \le C_0$ and $C_i + C_j > C_0$, for all $i, j \in \{1, 2, 3\}$, $i \neq j$. The normalization constant has form [9]

$$G(\rho) = \frac{1}{1 - \frac{\rho_1 + \rho_2 + \rho_3}{C_0}} \left(\frac{1 - \frac{\rho_1}{C_0}}{1 - \frac{\rho_1}{C_1}} + \frac{1 - \frac{\rho_2}{C_0}}{1 - \frac{\rho_2}{C_2}} + \frac{1 - \frac{\rho_3}{C_0}}{1 - \frac{\rho_3}{C_3}} - 2 \right),$$

and further, the throughput is

$$\gamma_i = \frac{C_i (C_0 - \rho_1 - \rho_2 - \rho_3)(1 - \frac{\rho_i}{C_i})^2 G(\rho)}{C_0 - C_i + C_i (1 - \frac{\rho_i}{C_i})^2 G(\rho)}, \quad i = 1, 2, 3. \tag{4.16}$$

In [6] it is proven, that utility-based allocations are not balanced in tree network topologies and that utility-based allocations coincides in tree networks.

[1] If $C_1 > C_0$, the link C_1 never bounds the traffic and can be omitted, and the network reduces to a simpler topology. Respectively this holds for the link C_2. If $C_1 + C_2 \le C_0$, the link C_0 never bounds traffic and thus the network reduces to two separate links and flow classes.

Chapter 5

Simulations

In this section simulations are discussed in general. Objectives of the simulations are presented and used network and simulation setups are introduced. The implementation is commented in brief.

5.1 Simulations in general

In simulations three different bandwidth allocation policies were studied — balanced fairness, max-min fairness and proportional fairness. The aim of the simulations was to verify some previous results concerning the throughput and compare the allocation policies in focus. As a new aspect the differences between the results provided by the definition of throughput (2.4) and flow-specific throughput were studied. Also the variance of flow durations was studied.

Sensitivity simulations provide information on the insensitivity of the bandwidth allocation policies in focus. To verify the linear property in balanced systems slow-down factor is measured.

To measure throughputs, flow-specific throughputs, flow durations, and slow-down factors each simulation run contained 1001000 flows traversing through the network. 1000 first flows were ignored as warmup period, and to calculate the deviations the simulation run was divided into 25 batches. In the simulations for the sensitivity inspections each simulation run contained 5001000 flows with 1000 flow warmup period. This increase in flow count observed necessary to reduce the variation in the results.

In simulations, the flows arrived to the network with exponentially distributed interval times, i.e. arrivals constitute a Poisson arrival process. Excluding the sensitivity simulations, the flow size distribution was assumed exponential in all other cases.

Following scenarios were studied via simulations:

- Homogenous traffic
- Heterogenous traffic
- Unimodal, bimodal and uniform flow size distributions
- Constant demand with varying ratio of flow size and arrival rate

In the homogenous case the expected traffic load was the same for all the flow classes. In each simulation the traffic load of all flow classes was increased with the same rate until the demand met the limit set by the traffic condition (2.3).

In the heterogenous case the expected traffic load of one traffic class was varied and all the others were kept constant at value called as the *base load*. The objective was to explore how the the throughputs of different flow classes behave when the traffic of one class differs from the others. As in the case of homogenous traffic, the traffic load of the flow class was increased until the demand met the limit set by the traffic condition (2.3).

In the sensitivity studies two cases were concerned. In the first case, the traffic load of all the traffic classes were kept constant, but the the expected flow size and arrival times of one flow class were varied. This time scale change was carried out so that the ratio of the expected flow size of the varied flow class to the corresponding quantity of the other flow classes was increased from value 0.01 up to value 100.

In the second case, the distribution of the flow size was set to be unimodal, bimodal and uniform instead of exponential distribution. All these three distributions had the same expected value than the used exponential distribution, which was set to be 1. The obtained results concerning the throughput were compared to the throughput obtained with the exponential flow size distribution.

The unimodal distribution is a deterministic distribution, i.e. it gives the expected value with probability 1. The bimodal distribution has two possible values. The first possible value 0.5 has probability 1/3 and the second value 1.25 probability 2/3. In the uniform distribution all the values at range $(0, 2)$ have probability 1/2.

5.1.1 Flow specific throughput

The throughput of the flow class $r \in \mathcal{R}$ is defined as the ratio of the expected flow size and the expected flow duration (2.4):

$$\gamma_r = \frac{\mathrm{E}[S_r]}{\mathrm{E}[T_r]}.$$

However, the throughput can be calculated for a single flow. In this context we denote this single flow throughput as a *flow specific throughput*. Our aim is to find out, how the throughput provided by the common definition differs from the expectation of the flow specific throughputs at corresponding flow class:

$$\gamma_r^{\text{fl}} \stackrel{\text{def}}{=} \mathrm{E}\left[\frac{S_r}{T_r}\right], \tag{5.1}$$

where $r \in \mathcal{R}$.

5.1.2 Slow-down factor

Slow-down factor ς can be defined as a ratio of the expected flow duration and the expected flow size. Let us denote the slow-down factor of a flow class $r \in \mathcal{R}$ as follows:

$$\varsigma_r \stackrel{\text{def}}{=} \mathrm{E}\left[\frac{T_r}{S_r}\right]. \tag{5.2}$$

Linear property (4.5) can be written as

$$\frac{1}{\gamma_r} = \frac{\mathrm{E}[T_r|\ S_r = s]}{s} = \mathrm{E}\left[\frac{T_r}{s}\,\middle|\, S_r = s\right] = \mathrm{E}\left[\frac{T_r}{S_r}\,\middle|\, S_r = s\right], \quad \forall s > 0,$$

from which it follows that

$$\mathrm{E}\left[\frac{T_r}{S_r}\right] = \mathrm{E}\left[\mathrm{E}\left[\frac{T_r}{S_r}\,\middle|\, S_r = s\right]\right] = \frac{1}{\gamma_r},$$

where the first equality follows from the conditioning rule $\mathrm{E}[X] = \mathrm{E}\left[\mathrm{E}[X|Y]\right]$.

That is, in a balanced system the linear property holds and thus the inverse of the throughput is equal to the slow-down factor for all flow classes. It is worth noting that conversely this is not necessarily valid, i.e. if ς_r is equal to $1/\gamma_r$ for all flow classes, it does not imply that the system is balanced.

5.2 Simulation setups

In this section the specific network setups used in the simulations are presented. Most of these network topologies are defined in detail in Section 4.5.

Line

The linear network is illustrated in Figure 4.2.

Two linear networks were used in simulations. The first linear network consists of $L = 2$ links with capacities $C_i \equiv 1$ and three flow classes. The second linear network is composed of $L = 5$ links with capacities $C_i \equiv 1$ and six flow classes.

Grid

The grid network is illustrated in Figure 4.3.

Two-dimensional 2×2 grid network used in simulations consists of four identical links with capacities $C_i \equiv 1$ and four flow classes.

Parking lot

The parking lot network topology is depicted in Figure 4.6.

Three parking lot networks were used in simulations, where the number of links were $L_1 = 3$, $L_2 = 4$ and $L_3 = 5$. In all the cases the link capacities were $C_i = i$, $i = 1, \ldots, L_j$, where j is the case index.

Hypercycle

The hypercycle network topology is shown in Figure 4.5.

Three hypercycle networks were used in simulations, where the number of links were $L_1 = 3$, $L_2 = 4$ and $L_3 = 5$. In all the cases the link capacities were set to $C_i \equiv 1$.

Trees

Simulations were ran for four different tree topologies. Two first tree network setups are illustrated by Figure 4.7. In case of tree 1 the link capacities were $C_0 = 1$ and $C_1 = C_2 = 0.7$. In case of tree 2 the link capacities were $C_0 = 1$ and $C_1 = C_2 = C_3 = 0.5$.

Figures 5.1 and 5.2 depict the tree setups 3 and 4. In these cases the link capacities were $C_1 = 0.5$, $C_2 = 0.7$ and $C_3 = 1$.

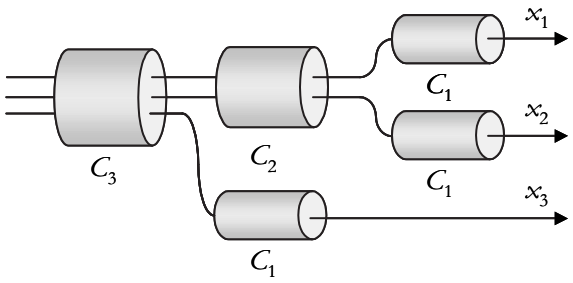


Figure 5.1: Tree topology 3.

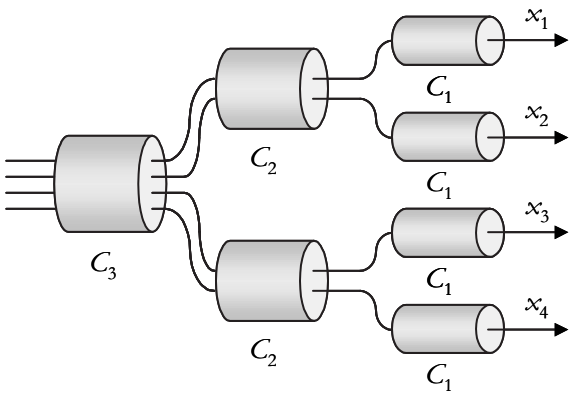


Figure 5.2: Tree topology 4.

5.3 About implementation

The simulations were carried out with a simulation program implemented with C++ programming language [28] and the standard template library (STL) [21]. The simulations were event driven. The event handling and scheduling, the random number generation and part of the statistical analysis where were implemented using the Communication Networks C++ Library (CNCL) [27].

Simulations were run on Sun Microsystems Fire 280R platform with Operating system Solaris 8 and 2 x UltraSPARC-III 750MHz processor.

The simulator and the simulation run is initialized by an input file. The input file contains the information about the length of the simulation run and warmup period, number of batches and the output file name. The network topology is presented as a connection matrix, in which each row profiles a single flow class and column corresponds to a single link in the network. Element 1 in an entry corresponds to case that the flow class traverses through the corresponding link, otherwise the element is set to 0. Also link capacities, the expected flow size and the the flow arrival rate of each flow class is in the input file. The used allocation policy is a command line parameter.

The simulator output consists of the following information: flow class specific demand, mean flow size and variance of the flow size, mean flow duration and the variance of the flow duration, throughput, flow specific throughput and slow-down factor.

Basically the simulator consists of three different parts. The main part reads the input file and initializes the simulator, creates the network topology, links and flow classes. Scheduler generates the flows and takes care of the event handling and scheduling. Allocator calculates the capacity allocation for the flow classes in given state of the network. When the network state changes, that is, a flow is generated or removed from the system, the allocator is called by the scheduler. The statistics are gathered at the same time when a flow is

removed from the system.

For the balanced and max-min fair allocation calculations general allocators were implemented. The allocator for the balanced fair allocation is based on the recursive precalculation of the values of the balance function Φ (4.7) and then using (4.3). The number of precalculated values per flow class has to be fixed in the beginning of the simulation. This defines the size of table allocated for the balance function. If the number of flows in some flow class exceeds the number of precalculated values, the corresponding balance function value and allocation is calculated with (4.7).

Max-min fair allocator was implemented as described in Chapter 3.2 (Table 3.1), based on presentation in [2].

Proportionally fair allocator was implemented by solving the optimization problem (3.3) in line, grid and hypercycle network topologies. Also the proportionally fair allocation for the 3-link parking lot topology was solved to verify the result that utility-fair allocations unite in tree topologies, A general solver was not implemented [1]. The guideline of solving the proportionally fair allocation in the hypercycle network topology is described in Appendix A.

The allocator for the balanced fair allocation turned out to be a bottleneck in the simulation runs. The table of the balance function values, for which the allocator allocates memory, becomes huge even with a quite small number of states per flow class when the number of flow classes increases. That is, if there are K flow classes and the number of precalculated values is n, the size of the allocated table is n^K double-type variables.

Another issue was the calculation of the values of the balance function during the simulation, which turned out to be slow. In practice, when the load of the system increased so much that all the needed balance function values were not precalculated, the constant recursive evaluating of these values stalled the simulation.

For these reasons the simulation setups were restricted to setups with six flow classes at most.

[1] Probably the most efficient algorithm type for this general optimization task would have been gradient projection method [1, 3].

Chapter 6

Results and analysis

In this section the results attained via simulations are gathered and analyzed. Comparison with analytical results is made.

6.1 Throughput

In this section different fairness policies are compared and and examined via throughputs they provide. In each case the throughputs of flow classes are measured in two different traffic scenarios. In the first one the traffic is homogenous, i.e. all the flow classes have equal expected demand of traffic. The demand is increased from zero to the saturation of the bottleneck links.

In the second scenario the expected demand is set to a constant for all the flow classes except for one. For that flow class, the expected demand is increased from zero to the saturation of the bottleneck link.

It following figures *Demand* denotes the demand of flow class, or flow classes, under examination.

In all the cases examined the differences in throughputs provided by different fairness criteria are comparatively small. Generally max-min fairness provides better throughput on the long routes and penalizes the shorter ones more than balanced fairness. It is verified that proportional fairness coincides with balanced fairness in homogenous hypercubes, and with max-min fairness in trees.

6.1.1 Line

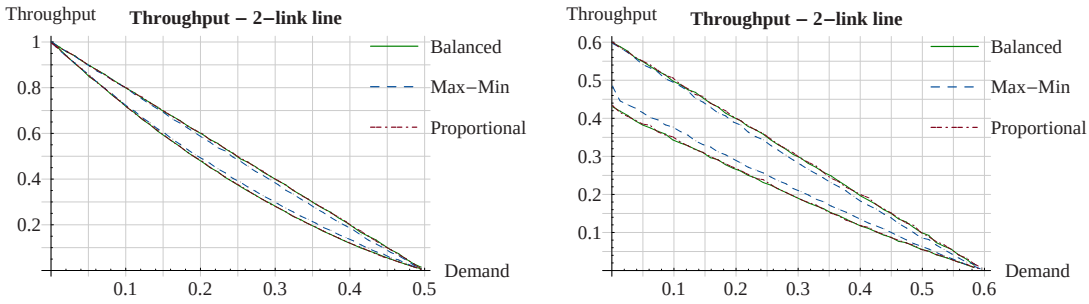
Figure 6.1 presents the linear network configurations used in the simulations.

In Figures 6.2 and 6.3 the upper group of lines refers to the shorter routes and

(a) 2-link line. (b) 5-link line.

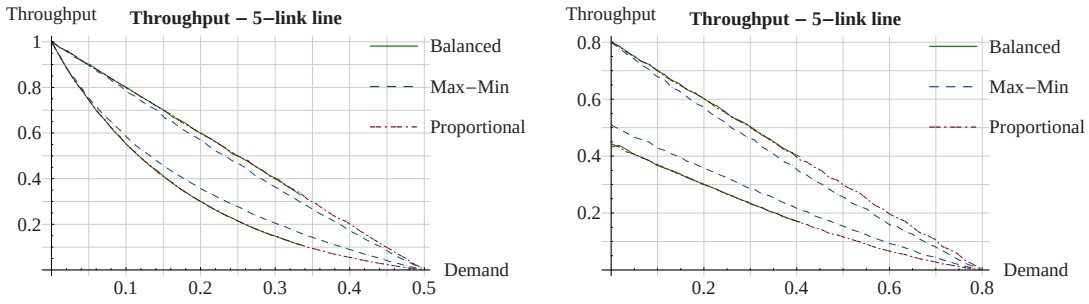
Figure 6.1: Linear network configurations used in simulations.

the lower to the long route. In the case of heterogenous traffic (Figures 6.2(b) and 6.3(b)), *base load* refers to the load of short flow classes which was kept as constant.



(a) Homogenous traffic. (b) Heterogenous traffic, base load 0.40.

Figure 6.2: Throughput in 2-link line.



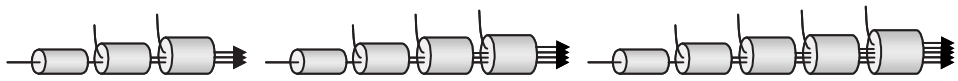
(a) Homogenous traffic. (b) Heterogenous traffic, base load 0.20.

Figure 6.3: Throughput in 5-link line.

Results in both cases are similar. Balanced fairness and proportional fairness provide the same throughput, which stands up for the statement that proportional fairness coincides with balanced fairness in homogenous hypercubes [6]. Max-min fairness provides better throughput on the long route and penalizes the shorter ones. The difference between max-min fairness and balanced fairness as well as proportional fairness is more pronounced in the 5-link network.

6.1.2 Parking lot

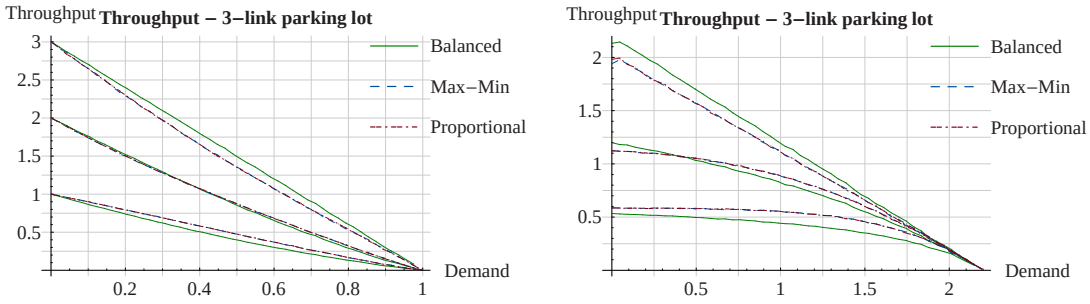
Figure 6.4 represents the parking lot network configurations used in the simulations.



(a) 3-link parking lot. (b) 4-link parking lot. (c) 5-link parking lot.

Figure 6.4: Parking lot network configurations used in simulations.

In Figures 6.5 and 6.6 the uppermost group of lines refers to the shortest route and the lowest group to the longest route. In the case of heterogenous traffic (Figures 6.5(b) and 6.6(b)), base load refers to the constant load 0.40 of all routes excluding the shortest one.



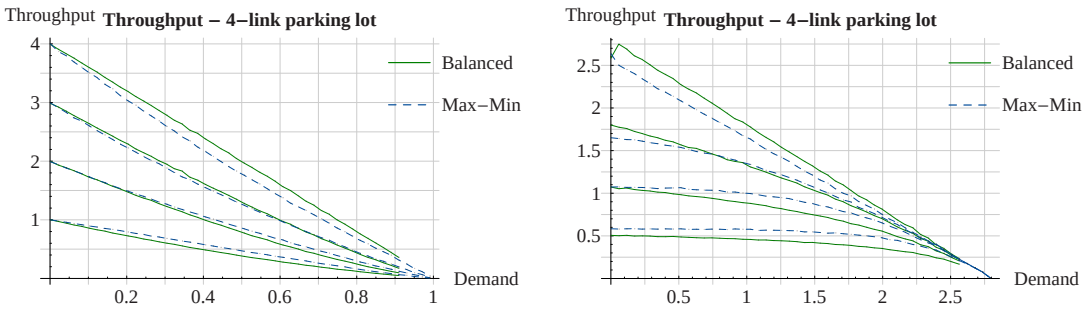
(a) Homogenous traffic. (b) Heterogenous traffic, base load 0.40.

Figure 6.5: Throughput in 3-link parking lot.

In the three link case (Figure 6.5(a)) balanced fairness provides greater throughput for the shortest flow class, and max-min fairness for the long flow class. For the middle flow class the throughput is almost equal — with low load the balanced fairness gives slightly better throughput, and with heavy load vice versa. The throughput provided by proportional fairness and max-min fairness are the same. This stands up for the statement that max-min and proportional fairness, and further, all utility-based allocations coincides in tree networks [6].

In the four link case (Figure 6.6(a)) the results are similar with the three link case. For the flow class traversing through three links balanced fairness provides better throughput with low load, but when the demand reaches the value 0.70, the curves unite. For the two link flow class the difference is more evident; with demand greater than 0.20, the max-min fairness gives greater throughput.

With heterogenous traffic the differences become more obvious (Figures 6.5(b) and 6.6(b)). Balanced fairness favors shorter routes, and with heavy link load



(a) Homogenous traffic.

(b) Heterogenous traffic, base load 0.40.

Figure 6.6: Throughput in 4-link parking lot.

the shortest route is the only one having greater throughput than what max-min fairness provides.

The five link case was excluded, because the results are similar to the presented ones.

6.1.3 Grid

Figure 6.7 presents the grid network configuration used in the simulations.

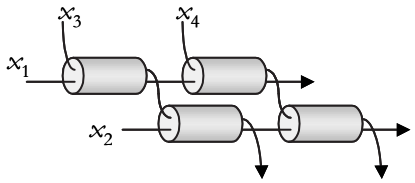
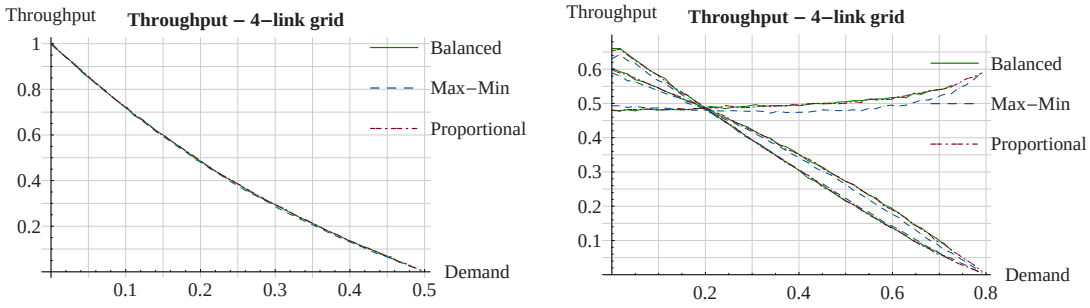


Figure 6.7: 2×2 grid network configuration used in simulations.

Figure 6.8 shows the throughput in a 2×2 grid. In the case of heterogenous traffic, the base load of flow classes 2 to 4 was 0.20 and the load of flow class 1 was varied.

With homogenous traffic all the three allocation policies provide the same throughput. The throughputs do not decrease linearly as a function of demand, but the curves are slightly convex. In Figure 6.8(b) the group of throughputs with the deepest downward trend corresponds to flow class 1. The other group of downward lines corresponds to flow classes 3 and 4, and the third group to flow class 2.

Heterogenous traffic brings out some differences in throughputs provided by different criteria. The load of flow class 1 is varied, and the other classes are set to the base load of 0.20. Balanced fairness and proportional fairness provide



(a) Homogenous traffic.

(b) Heterogenous traffic, base load 0.20.

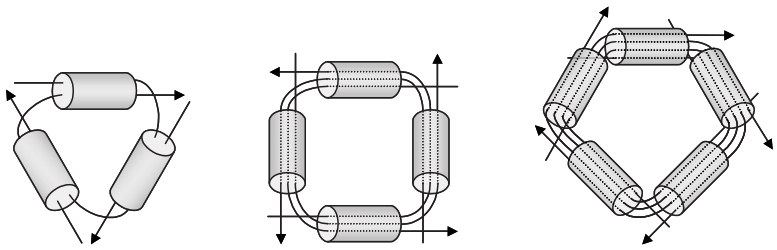
Figure 6.8: Throughput in 2×2 grid.

the same throughput, but in comparison, max-min fairness provides smaller throughput for flow classes 1 and 2 when the demand is below 0.20, and smaller throughput for flow classes 2, 3 and 4 when the demand is above 0.20. For the flow classes 3 and 4 the throughput of max-min fairness becomes smaller and unites with balanced and proportional fairness for flow class 1, when the demand is above 0.20. When demand is 0.20, all the throughputs unite. At this point the traffic is homogenous.

The slight upward trend in the throughput of flow class 2 is explained by the fact that flow classes 1 and 2 do not share any links. When the load of class 1 increases, the bandwidth allocated for classes 3 and 4 decreases. Flow classes 2 and 3, as well as classes 2 and 4 have common link, and thus the allocation for class 2 is increased with all policies.

6.1.4 Hypercycle

Figure 6.9 represents the hypercycle network configurations used in the simulations.



(a) 3-link hypercycle network configuration.

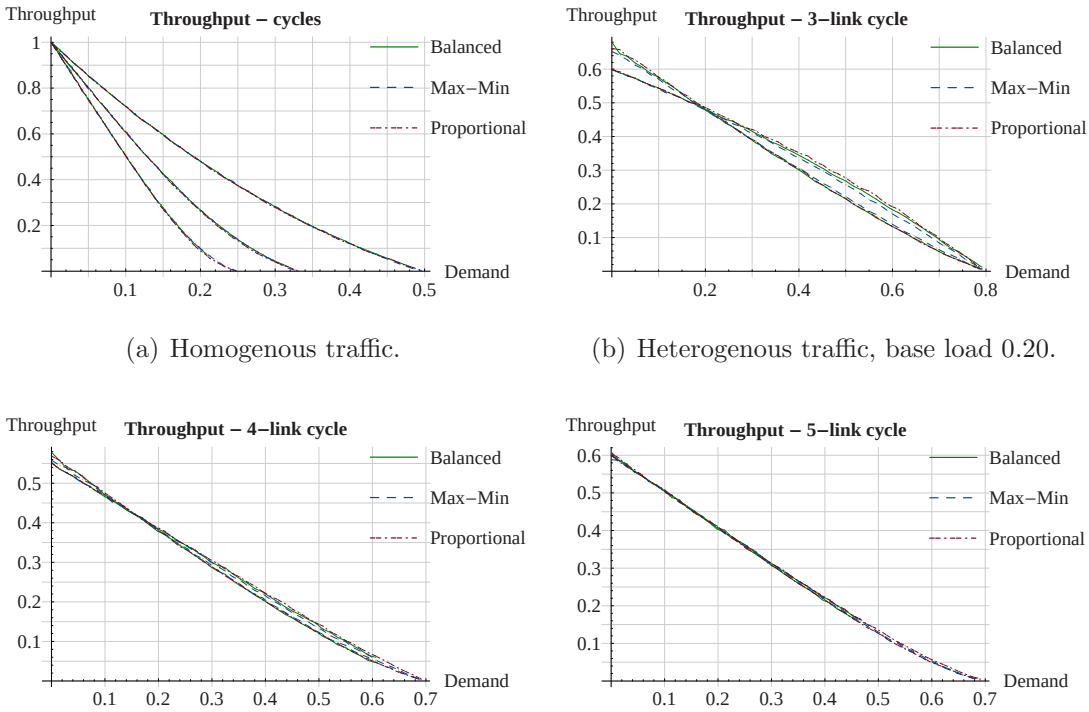
(b) 4-link hypercycle network configuration.

(c) 5-link hypercycle network configuration.

Figure 6.9: Hypercycle network configurations used in simulations.

Figure 6.10 depicts the throughput in hypercycles. In the case of heterogenous

traffic, the load of flow class 1 was varied and loads of all the others was kept constant. In the 3, 4 and 5 link cases the base loads were 0.20, 0.15 and 0.10, respectively.



(a) Homogenous traffic.

(b) Heterogenous traffic, base load 0.20.

(c) Heterogenous traffic, base load 0.15.

(d) Heterogenous traffic, base load 0.10.

Figure 6.10: Throughput in hypercycles.

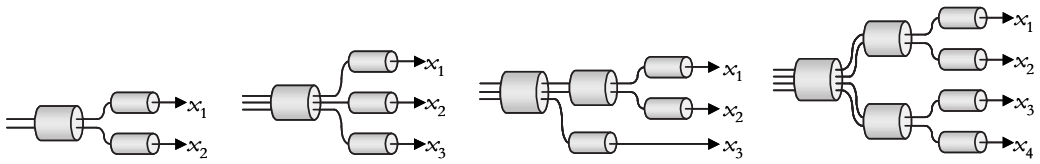
If Figure 6.10(a) throughputs provided by the homogenous traffic in all the tree cases are shown. The steepest group of curves corresponds to the 5 link case and the flattest group to the 3 link case, respectively. All the allocation policies provide the same throughput.

In the three link case (Figure 6.10(b)) balanced fairness and proportional fairness give the same throughput. When the load of the varied flow class 1 is below the base load 0.20, the fixed classes get higher throughput and above value 0.20 vice versa. At point 0.20, the traffic is homogenous and all the throughputs are equal.

Differences between max-min fairness and balanced fairness (as well as with proportional fairness) are minor. With demand below 0.20 throughputs unite for flow class 1, for other classes balanced fairness and proportional fairness provide slightly better throughput. Above value 0.20 balanced fairness and proportional fairness provide greater throughput for class 1, but max-min fairness for the other classes.

6.1.5 Trees

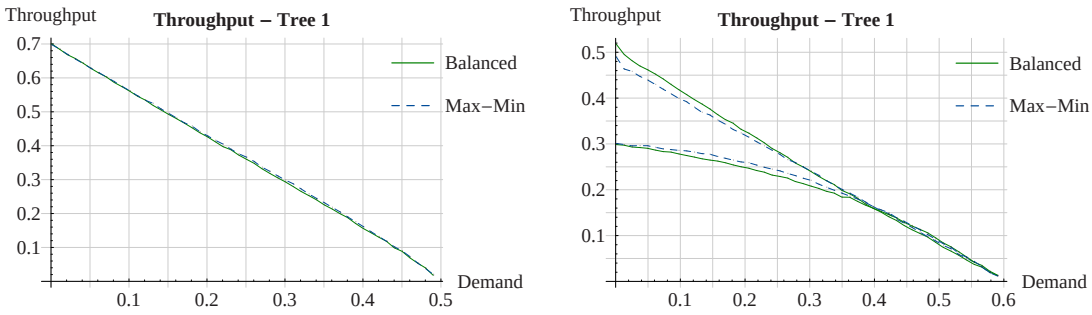
Figure 6.11 illustrates the tree network configurations used in the simulations.



(a) Tree topology 1. (b) Tree topology 2. (c) Tree topology 3. (d) Tree topology 4.

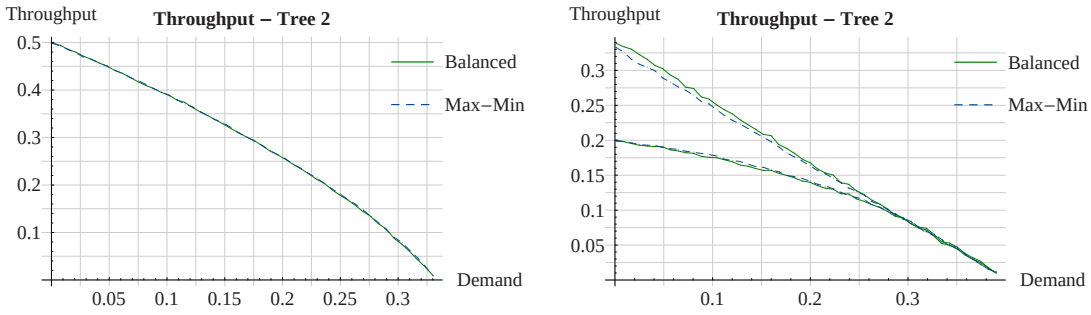
Figure 6.11: Tree topologies used in simulations.

The throughputs in the tree network setups are shown in Figures 6.12, 6.13, 6.14 and 6.15. In the case of heterogenous traffic, the load of flow class 1 was varied in all the cases. The base load was set to the values 0.40, 0.30, 0.30 and 0.20 in the cases of tree 1, 2, 3 and 4, respectively.



(a) Homogenous traffic. (b) Heterogenous traffic, base load 0.40.

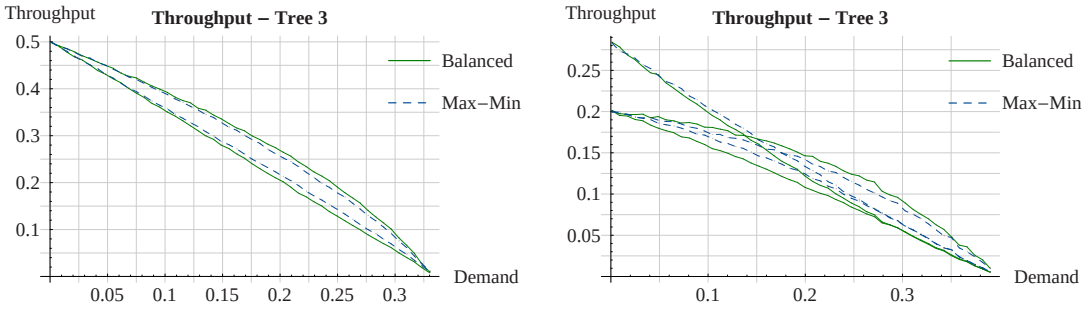
Figure 6.12: Throughput in tree 1.



(a) Homogenous traffic. (b) Heterogenous traffic, base load 0.30.

Figure 6.13: Throughput in tree 2.

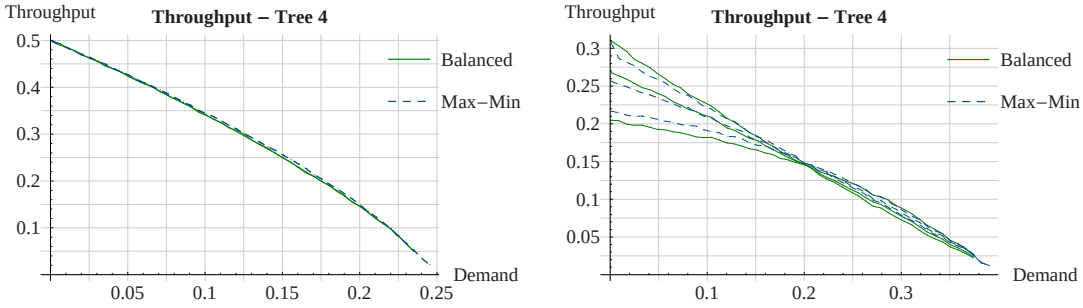
In the symmetric cases, trees 1, 2 and 4, the throughput provided by balanced fairness and max-min fairness seem to be the same. In tree 3 the throughput



(a) Homogenous traffic.

(b) Heterogenous traffic, base load 0.30.

Figure 6.14: Throughput in tree 3.



(a) Homogenous traffic.

(b) Heterogenous traffic, base load 0.20.

Figure 6.15: Throughput in tree 4.

provided by both policies is greater for the shortest flow class 3. Balanced fairness gives smaller throughput for the flow classes 1 and 2 in comparison to max-min fairness. For the flow class 3 the result is reversed.

In case of heterogenous traffic some differences come out also in the symmetric cases. In trees 1 and 2 results are similar (Figures 6.12(b) and 6.13(b)). For the varied flow class the throughput decreases almost linearly for both of the allocation policies. When the load of the varied class meets the base load, the curves cross. Below the base load balanced fairness provides higher throughput for the varied flow class and max-min fairness for the other flow classes, respectively.

Above the base load differences are minimal. Both policies provide slightly better throughput for the varied flow class. Balanced fairness gives slightly greater throughput for the varied class, and max-min fairness for the others.

In the case of tree 3 (Figure 6.14(b)) the throughput of the varied flow class 1 is highest when the load is below 0.15, but above this, flow class 3, which shares only the trunk link with flow classes 1 and 2, gets the best throughput. Balanced fairness provides better throughput for the flow class 3, and max-min fairness for classes 1 and 2.

In the case of tree 4 (Figure 6.15(b)) the varied flow class 1 gets the best throughput with both policies, when the demand is below the base load 0.20. Also the flow class 2 belonging to the same branch with class 1 gets better throughput with demand below the base load. When the load of the class 1 gets the value 0.20, all curves unite. This point corresponds to a homogenous case. With load above the base load, classes 3 and 4 get the best throughput.

Balanced fairness provides higher throughput for classes 1 and 2 and max-min fairness for classes 3 and 4, when the load is below base load. When load increases, the result gets reversed.

6.2 Sensitivity

In this section the sensitivity of allocation policies is studied via throughput measurements. In each case the expected demand is set to be constant for all flow classes. However, for one flow class, the ratio of the expected flow size and the expected arrival rate is varied so that the demand still remains constant, i.e. the time scale of the flow class is varied. For the flow class under observation, the ratio of the expected flow size and the expected flow size of the other flow classes is increased from value 0.01 to value 100. In the following figures this ratio is denoted as *FSize Ratio*.

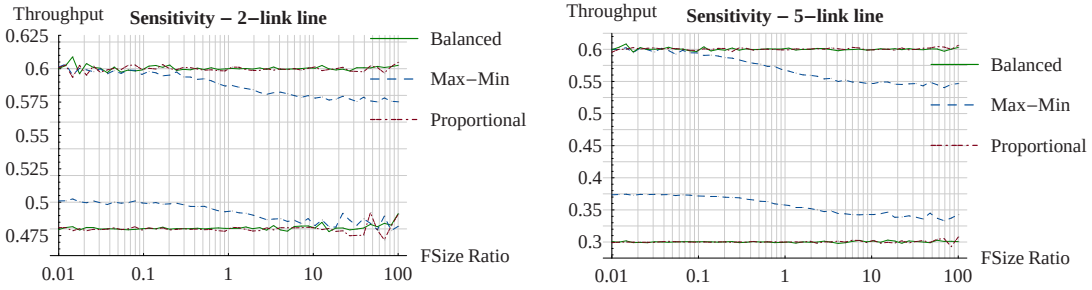
By the theory balanced fairness is insensitive. This is verified by the simulation results. Also the sensitivity of max-min fairness and proportional fairness seem to be quite weak.

6.2.1 Line

Figure 6.16 depicts the throughputs in linear network topologies. The base load of all flow classes is set to 0.20. Flows on the long route is varied, thus FSize Ratio is the expected flow size on the long route in proportion to the expected flow size on the short routes.

In Figures 6.16(a) and 6.16(b) the upper group of curves refers to the short flow classes, and the lower group to the long flow class.

For balanced fairness and proportional fairness the throughput remains constant as a function of flow size ratio. In the case of max-min fairness, the throughput is equal to that of balanced fairness for the short route classes, when the flow size ratio is small. Correspondingly, when the flow size ratio increases, the throughput of the long route class approaches that of the balanced fairness. That is, when a flow class is slowed down, i.e. expected flow size is increased and arrival rate decreased, the throughput provided by max-min fairness approaches the throughput given by balanced fairness.



(a) Sensitivity in 2-line networks. Base load 0.20.

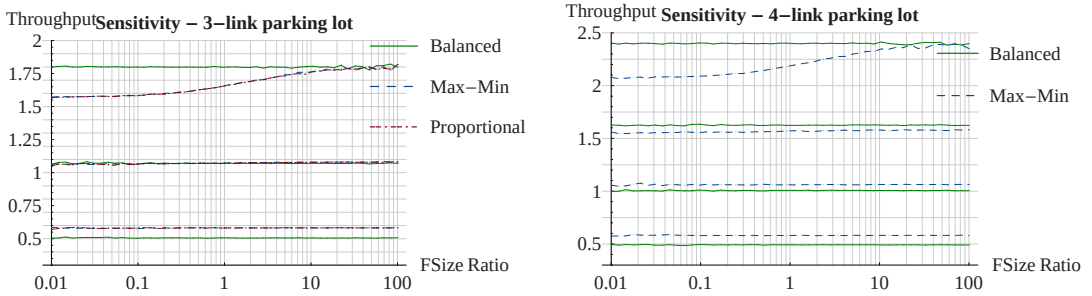
(b) Sensitivity in 5-line networks. Base load 0.20.

Figure 6.16: Sensitivity in line networks.

Max-min fairness is sensitive in linear network topologies, whereas balanced fairness and proportional fairness remain insensitive.

6.2.2 Parking lot

In Figures 6.17(a) and 6.17(b) the throughputs in parking lot network topologies are shown. The base load of all flow classes is set to 0.40. Flows on the shortest route are varied and FSize Ratio is the expected flow size on the shortest route in proportion to the expected flow size of the other classes.



(a) Sensitivity in 3-link parking lot. Base load 0.40.

(b) Sensitivity in 4-link parking lot. Base load 0.40.

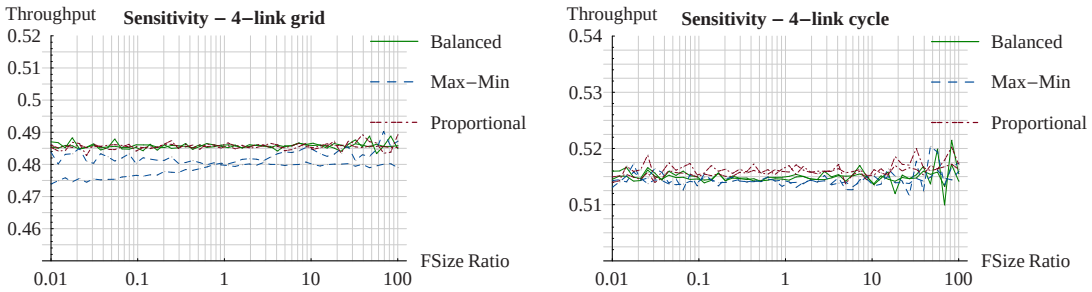
Figure 6.17: Sensitivity in parking lot networks.

The throughput provided by balanced fairness stays constant. Max-min fairness provides constant throughput for flow classes that are not varied. For the shortest class the throughput increases and approaches the throughput given by balanced fairness.

In parking lot network topologies balanced fairness is insensitive and max-min fairness is sensitive.

6.2.3 Grid and hypercycle

Throughputs in 2×2 grid and in four link hypercycle are depicted in Figures 6.18(a) and 6.18(b), respectively. For the grid network the base load of all the flow classes is set to value 0.20. The base loads for hypercycle topologies are 0.20, 0.125 and 0.10 for three, four and five link cases, respectively.



(a) Sensitivity in 2×2 grid. Base load 0.20.

(b) Sensitivity in 4-link hypercycle. Base load 0.125.

Figure 6.18: Sensitivity in grid and hypercycle networks.

In the grid case balanced fairness and proportional fairness provide the same constant throughput. When the flow size ratio is small, the throughput provided by max-min fairness for the non-varied flow classes is equal to the throughput provided by balanced fairness. As the ratio increases, the throughput of the varied class approaches the throughput curve of balanced fairness and the curve of non-varied classes recedes. That is, slowing down a flow class leads to same throughput with balanced fairness.

In grid networks max-min fairness is sensitive, whereas balanced fairness and proportional fairness are insensitive.

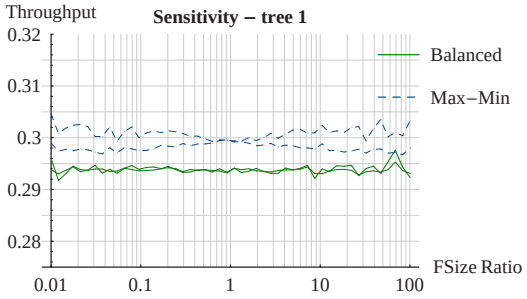
In the hypercycle case all the allocation policies give a constant throughput. Thus, in hypercycle networks balanced fairness, max-min fairness as well as proportional fairness are insensitive.

6.2.4 Trees

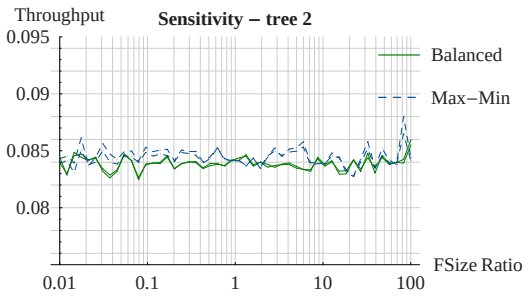
Throughputs in the tree network setups are shown in Figure 6.19. Base loads were set as follows: 0.30 for trees 1, 2 and 3, and 0.20 for tree 4.

In all the cases balanced fairness provides constant throughput.

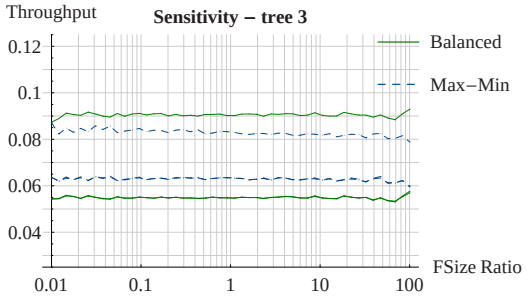
In the case of tree 1, the throughput of the varied flow class given by max-min fairness decreases and approaches the the throughput provided by balanced fairness. For the non-varied class the result is opposite. Again, slowing down a flow class leads to the same throughput with balanced fairness.



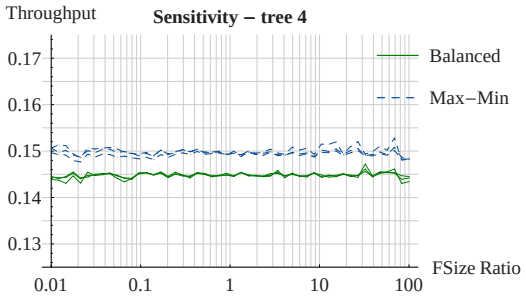
(a) Sensitivity in tree 1. Base load 0.30.



(b) Sensitivity in tree 2. Base load 0.30.



(c) Sensitivity in tree 3. Base load 0.30.



(d) Sensitivity in tree 4. Base load 0.20.

Figure 6.19: Sensitivity in tree networks.

In cases 2 and 4 the max-min fairness provides a constant throughput for all flow classes. In case 3 the throughput of the varied flow class traversing through two links remains constant, but the throughput of the non-varied flow classes slightly degreases when the flow size ratio increases.

As stated in the case of parking lot topology, in tree topologies balanced fairness is insensitive, and max-min fairness is sensitive.

6.3 Flow size distributions

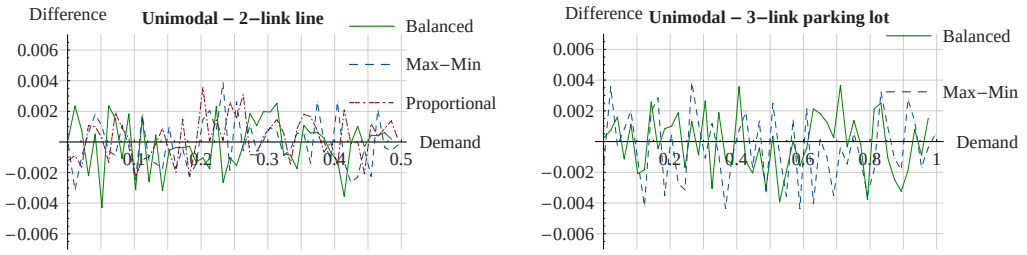
To study the sensitivity with respect to the flow size distribution, three different flow size distributions (in addition to the exponential distribution) were used in the simulations – unimodal, bimodal and uniform flow size distributions.

On the grounds of Section 6.2 the examination is predefined to the linear and parking lot network scenarios in which the time scale change was evident, and thus the sensitivity of the allocation policies seem to be most significant. The used traffic was homogenous.

However, the throughputs provided by these three different flow size distribu-

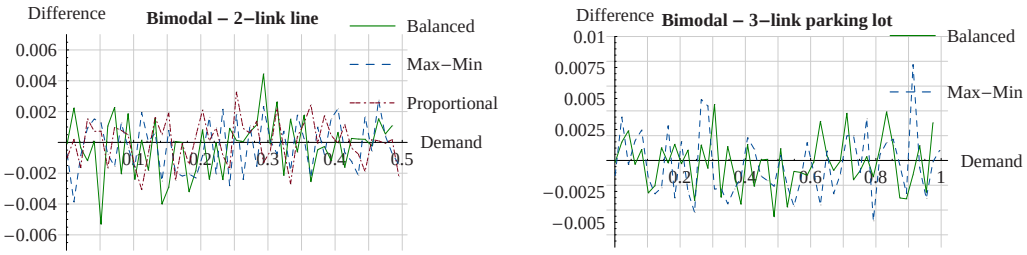
tion seem to be pretty much the same as obtained with the exponential flow size distribution. Thus, in place of throughput the difference to the throughputs obtained in Section 6.1 is examined.

If Figure 6.20 the the difference in the throughputs is depicted in the case of unimodal flow size distribution. Respectively, Figures 6.21 and 6.22 illustrate the difference in cases of bimodal and uniform flow size distributions.



(a) Difference in throughputs in 2-link line. (b) Difference in throughputs in 3-link parking lot.

Figure 6.20: Unimodal flow size distribution.

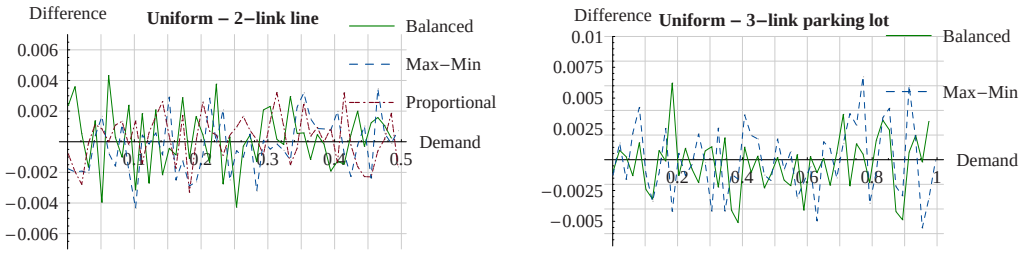


(a) Difference in throughputs in 2-link line. (b) Difference in throughputs in 3-link parking lot.

Figure 6.21: Bimodal flow size distribution.

In the all the cases the difference to the throughput provided by the exponential flow size distribution is negligible and seems to be just numerical noise with approximately same amplitude. This implies that the balanced fairness is insensitive in sense of the flow size distribution, as it was supposed to be. Also the utility-based allocations, max-min fairness and proportional fairness, do not seem to be sensitive with respect to the flow size distribution.

This result implies that the flow size distribution is not significant factor when the sensitivity of utility based allocation policies is considered. These policies are more sensitive with respect to the time-scale changes.

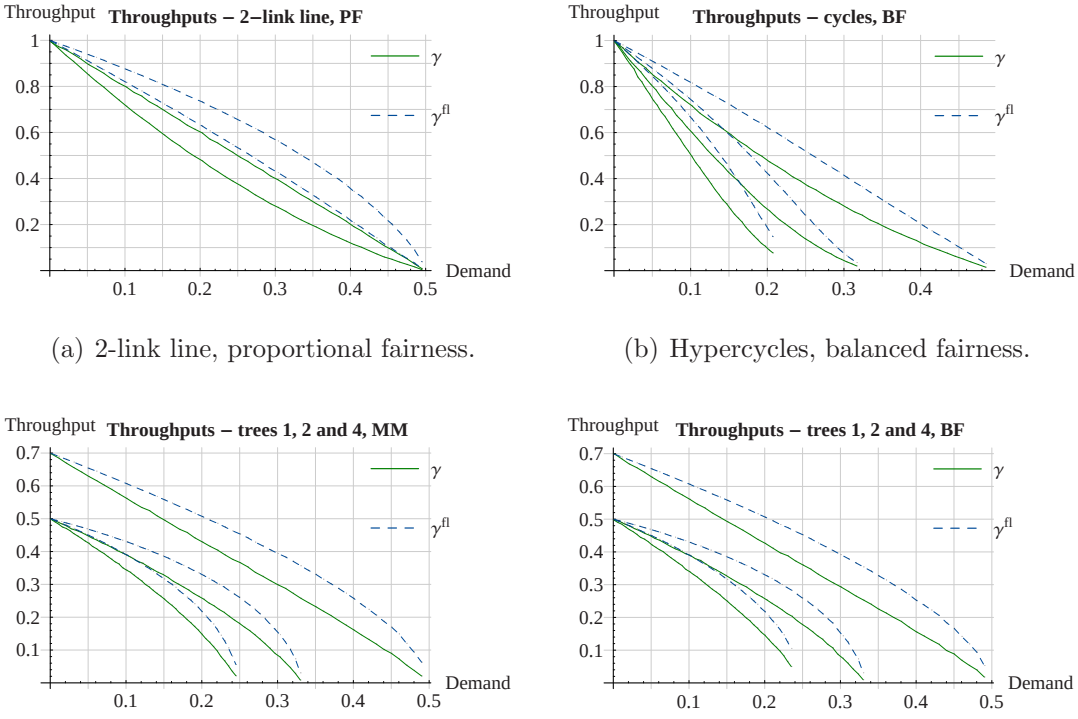


(a) Difference in throughputs in 2-link line. (b) Difference in throughputs in 3-link parking lot.

Figure 6.22: Uniform flow size distribution.

6.4 Flow specific throughput

In simulations the common throughput (2.4) was compared to the flow specific throughput defined by (5.1). Figure 6.23 depicts these throughputs provided by different allocation policies in line, hypercycle and tree network topologies.



(a) 2-link line, proportional fairness.

(b) Hypercycles, balanced fairness.

(c) Trees 1, 2 and 4, max-min fairness

(d) Trees 1, 2 and 4, balanced fairness

Figure 6.23: Throughput and flow specific throughput.

The main difference is readily seen, the flow specific throughput is notably greater than the classic throughput in all the cases and with all the allocation policies.

The explanation to this result is quite straightforward. Calculation of the mean of the flow durations evens the effect of the smallest values. In the sequence of the inverses of the flow durations, the smallest durations produce relatively high inverse values. Thus, the mean of the inverses of the flow durations is higher than the inverse of the mean of the flow durations.

The maximum flow specific throughput is attained when a flow traverses through an empty network and gets the the maximal possible capacity. When the load of the network is below the maximum, statistically some of the flows reach relatively high throughput, even the maximum with a finite probability. In the sequence of the flow specific throughputs small flow durations provide high throughputs, and thus the mean of this sequence remains relatively high in comparison to the ratio of the expected flow size and the expected flow duration.

When the demand is not vanishingly small or near the maximal, these two throughputs approach each others. With a very small load the network is practically empty, and in sight of an arriving flow the network reduces to a single constraining link. Thus, the classical throughput is the link capacity brought down with the load, as stated by (2.5). Correspondingly, in the case of the flow specific throughput, the flows get practically the maximal throughput on average.

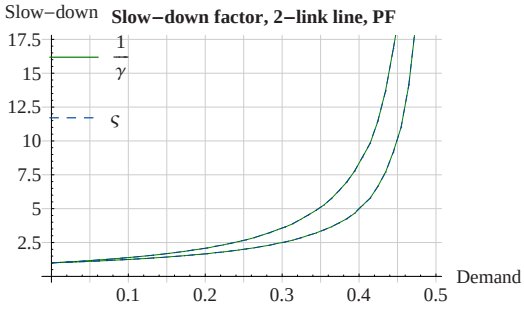
In the case of a heavy load, the flow durations approach infinity on average, and both throughput definitions tend to value zero.

6.5 Slow-down factor

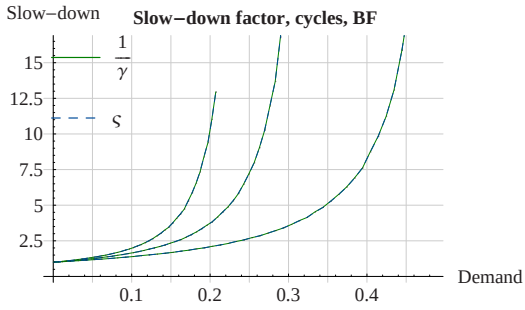
The slow-down factor is defined by (5.2). The slow-down factors and the inverses of throughputs are illustrated in Figure 6.24 concerning line, hypercycle and tree network topologies. These results are in the case of heterogenous traffic.

For the balanced fairness (Figures 6.24(b) and 6.24(d)) the observed quantities unite as was expected. Figure 6.24(a) depicts the same result for a 2-link line in the case of proportional fairness. It it known that proportional fairness coincides with balanced fairness in homogenous hypercubes [5], thus the curves are supposed to coincide. In the case of max-min fairness in trees 1, 2 and 4 (Figure 6.24(c)), the slow-down factors and the inverses of the throughputs unite also.

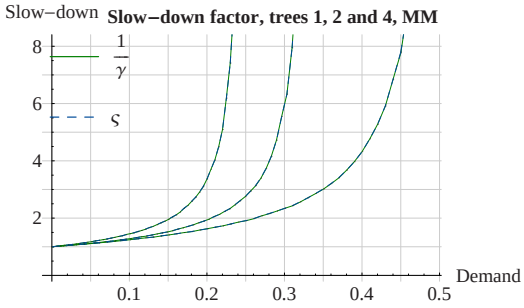
Actually this recurs in all the other cases — the slow-down factor and the inverse of throughput coincide for all the allocation policies with homogenous and heterogenous traffic alike. This verifies the accuracy of the linear property in the case of balanced allocation and also implies that the sensitivity of the max-min and proportionally fair allocations is quite weak in the cases studied.



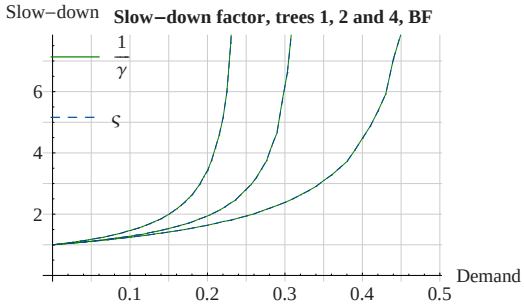
(a) 2-link line, proportional fairness.



(b) Hypercycles, balanced fairness.



(c) Trees 1, 2 and 4, max-min fairness



(d) Trees 1, 2 and 4, balanced fairness

Figure 6.24: Slow-down factor and the inverse of throughput.

6.6 Variance of flow duration

Figure 6.25 depicts the variance of flow duration in a single link network containing one flow class.

This is fundamentally a processor sharing queue and the variance of the flow duration has an explicit formula (2.6). Figure 6.25 shows that the variance of the flow duration provided by simulation follows closely the exact result[1].

Figure 6.26 depicts the flow duration variances in the case of homogenous traffic and network topologies 2-link line, 3-link parking lot, 3-link cycle, 2×2 grid and all the tree topologies. In the cases of 2-link line (Figure 6.26(a)) and 3-link parking lot (Figure 6.26(b)) only the results for the long route flow class are depicted.

In all the cases different allocation policies produce the same variance for the flow duration, no significant differences can be detected. Also the shape of the variance is the same for all network setups, even the values seem to be almost the same.

These results imply that the variance of the flow duration is robust in sense of

[1]This one link simulation was carried out to verify that the simulator provides correct results.

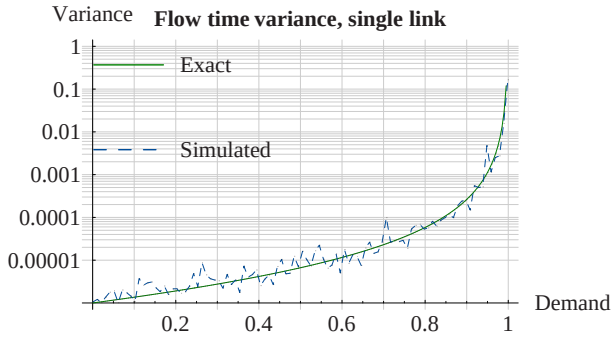


Figure 6.25: Variance of flow duration in a single link.

the allocation policy.

6.7 Wasted bandwidth

Balanced fairness is not Pareto-efficient in hypercycles [6].

Figure 6.27 depicts the proportion of wasted bandwidth in percents on the z-axis in three and four link hypercycles with unit capacity links. The relative waste of bandwidth for allocation ϕ is defined as follows:

$$\text{waste}(\phi) \stackrel{\text{def}}{=} \max_{r \in \mathcal{R}} \left\{ \min_{l \in r} \left\{ \frac{C_l - \sum_{r' \in l} \phi_{r'}}{C_l} \right\} \right\}. \tag{6.1}$$

In Figure 6.27(a) the solid surface describes the case in which the number of flows in flow class 1 is fixed at value $x_1 = 10$. The wire framed surface corresponds to the case $x_1 = 40$. Respectively, in Figure 6.27(b) the solid surface corresponds to the case $x_1 = x_2 = 10$ and wire framed to the case $x_1 = x_2 = 40$.

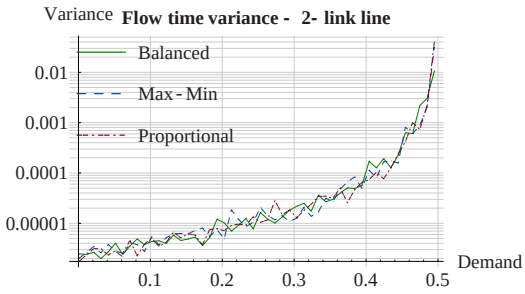
Both Figures 6.27(a) and 6.27(b) show that the maximum of the function (6.1) is reached, when one of the flow classes has one flow and the others have equal number of active flows. In [6] these balanced allocations are presented in three and four link hypercycles with unit capacity links:

$$\phi_1^3(x) = \frac{n+1}{3n+1}, \quad \phi_2^3(x) = \phi_3^3(x) = \frac{1}{2},$$

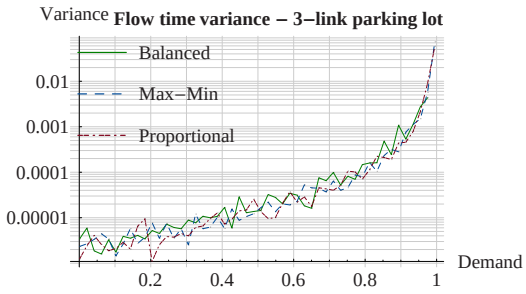
where $x = (1, n, n)$, $n \geq 1$, and

$$\phi_1^4(x) = \frac{2n^2+3n+1}{11n^2+6n+1}, \quad \phi_2^4(x) = \phi_3^4(x) = \phi_3^4(x) = \frac{1}{3},$$

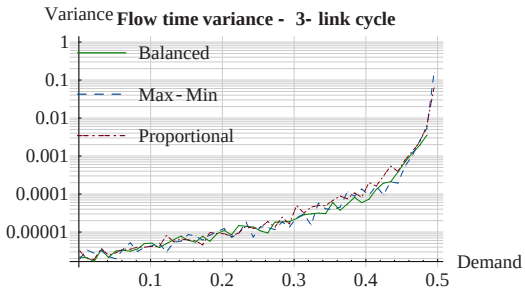
where $x = (1, n, n, n)$, $n \geq 1$. In the previous, ϕ^3 refers to the three link case and ϕ^4 to the four link case. Now, the maximum amount of wasted bandwidth



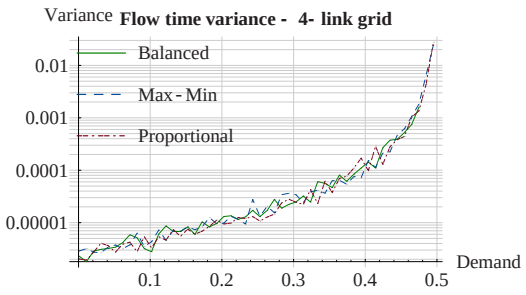
(a) Variance of flow duration in 2-link line.



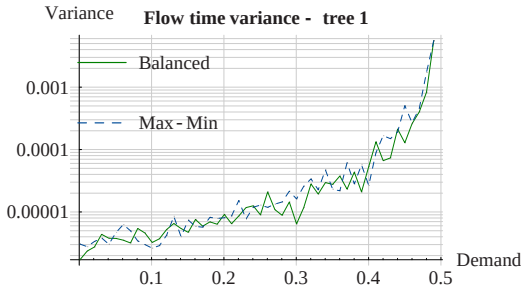
(b) Variance of flow duration in 3-link parking lot.



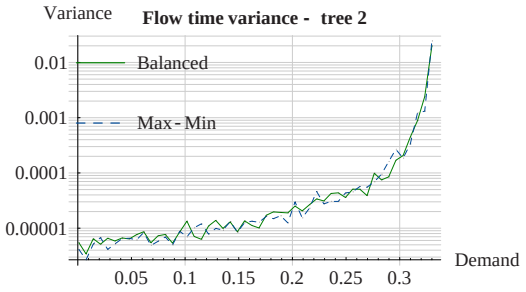
(c) Variance of flow duration in 3-link cycle.



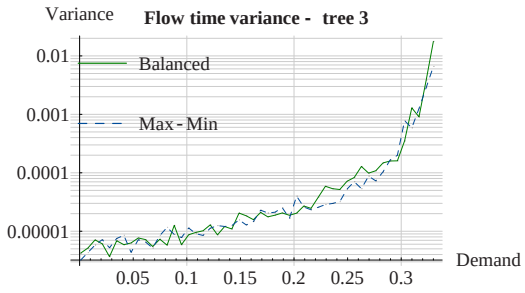
(d) Variance of flow duration in 2×2 grid.



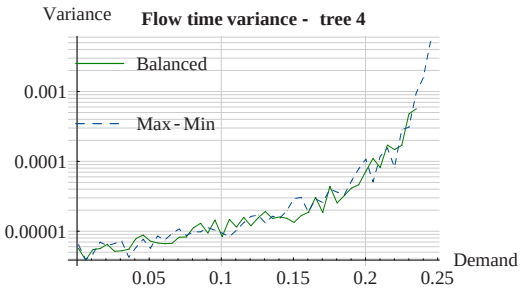
(e) Variance of flow duration in tree 1.



(f) Variance of flow duration in tree 2.

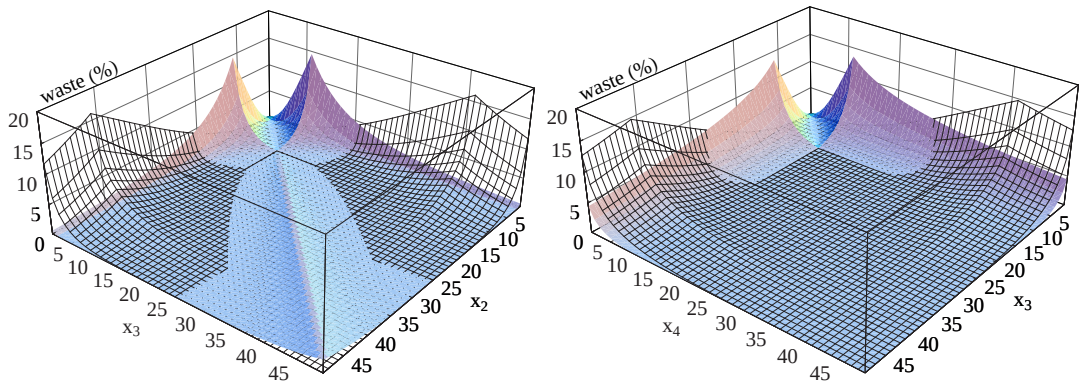


(g) Variance of flow duration in tree 3.



(h) Variance of flow duration in tree 4.

Figure 6.26: Variance of flow duration.



(a) Waste on 3-link hypercycle.

(b) Waste on 4-link hypercycle.

Figure 6.27: Inefficiency of balanced fairness on hypercycles.

is calculated as follows in the three link case:

$$\lim_{n\to\infty} C - (\phi_1 + \phi_2) = \lim_{n\to\infty} 1 - \frac{n+1}{3n+1} - \frac{1}{2} = \lim_{n\to\infty} \frac{n-1}{6n+2} = \lim_{n\to\infty} \frac{1-1/n}{6+2/n} = \frac{1}{6}.$$

Similarly, it can be shown that the maximum waste in the four link case is 5/33.

6.8 Numerical allocation comparison

In symmetric tree topologies and in hypercycles the throughputs provided by the max-min fair and balanced allocation policies seem to be the same for homogenous traffic (Figures 6.8, 6.10, 6.12, 6.13 and 6.15). Thus, it can be asked whether these allocations are the same.

The most reliable way to verify the difference or equality of allocations is to derive analytical expressions for the allocations. However, a comparison can be made numerically by calculating the allocations and measuring the difference with some distance function. In our comparisons, the following measure function is used:

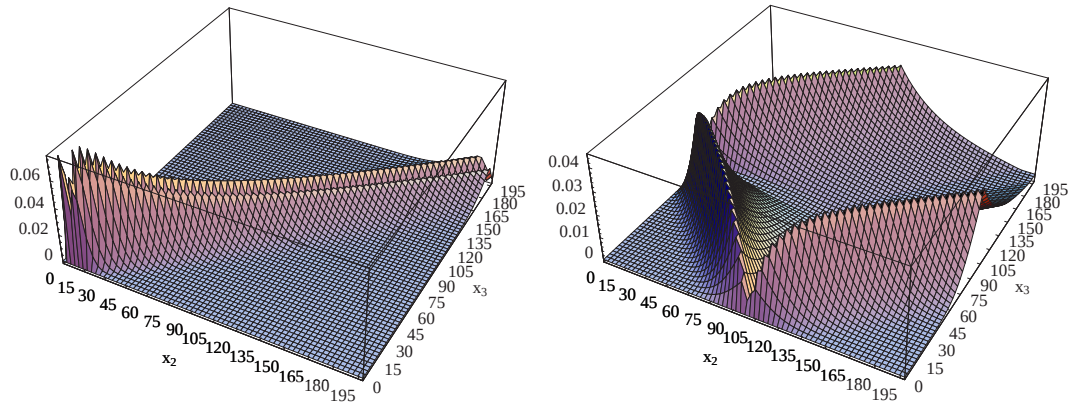
$$\left\|\phi^{\mathrm{BF}} - \phi^{\mathrm{MM}}\right\| = \sum_{r\in\mathcal{R}} \left|\phi_r^{\mathrm{BF}} - \phi_r^{\mathrm{MM}}\right|, \tag{6.2}$$

where ϕ^{BF} and ϕ^{MM} are rate allocation vectors produced by max-min fairness and balanced fairness, respectively.

Allocation comparison in tree 2 network

In Figure 6.28(a) allocation comparison is made for the tree 2 network. Number of flows at flow class 1 fixed to the value $x_1 = 10$. Similarly, Figure 6.28(b)

depicts the case where the number of flows in flow class 1 is fixed to the value $x_1 = 100$.



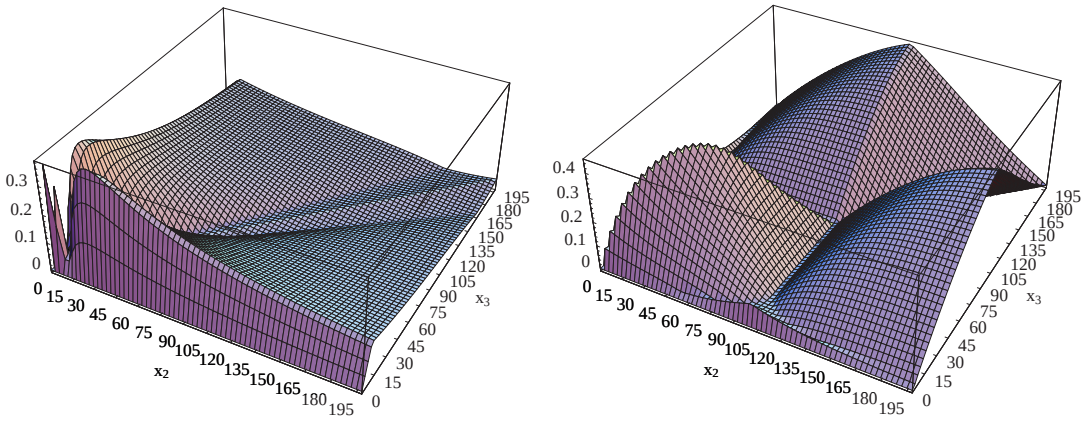
(a) Number of flows at class 1 $x_1 = 10$ (b) Number of flows at class 1 $x_1 = 100$

Figure 6.28: Allocation comparison in tree 2 network.

Figures 6.28(a) and 6.28(b) demonstrate it clearly that the allocations are not identical. The difference is maximized (in sense of the measure (6.2)) when the number of flows in flow class i is x_i, and the other flow classes have $x_j = \frac{1}{2}x_i$ flows, $i, j \in \{1, 2, 3\}$, $j \neq i$.

Allocation comparison in three link hypercycle network

In Figure 6.29(a) allocation comparison is made for the three link hypercycle network. Number of flows in flow class 1 fixed to the value $x_1 = 10$. Similarly, Figure 6.29(b) depicts the case in which the number of flows in flow class 1 is fixed to the value $x_1 = 100$.



(a) Number of flows at class 1 $x_1 = 10$. (b) Number of flows at class 1 $x_1 = 100$.

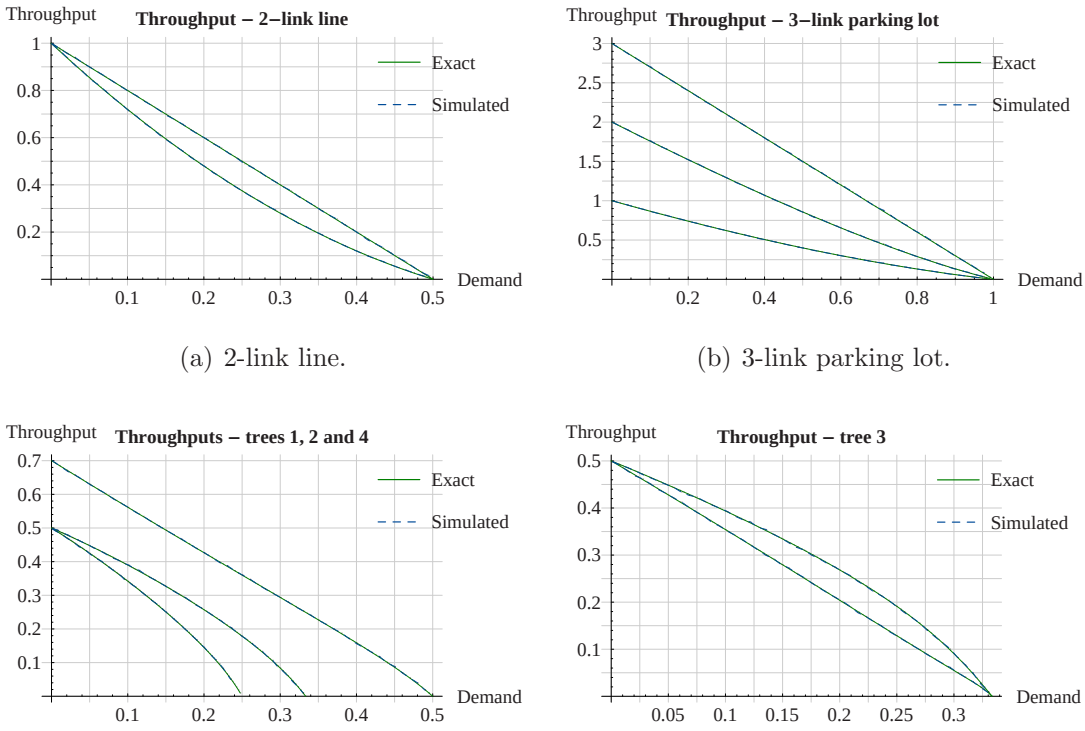
Figure 6.29: Allocation comparison in three link hypercycle.

Figures 6.29(a) and 6.29(b) show clearly, that the allocations are not equal. The difference is maximized when the number of flows at flow class i is x_i, and the other flow classes have $x_j = \frac{1}{2}x_i$ flows, $i, j \in \{1, 2, 3\}$, $j \neq i$.

6.9 Comparison with analytical results

To verify the correctness of the simulated results, the throughputs provided by the balanced fairness were compared to throughputs given by the analytical results presented in Chapter 4.

For the linear network the exact throughput is given by (4.11) with parameter values $C_i \equiv 1$ and $n = 2$ or $n = 5$ for the two and five link cases, respectively. In the case of the parking lot network (4.14) provides the exact throughput with parameter values $C_i = i$, $i = 1, \ldots, n$, and n is the number of links in the network under examination. For trees 1 and 2 the exact throughput is given by (4.15) and (4.16) with parameter values $C_0 = 1$ and $C_1 = C_2 = .7$ for the tree 1 and values $C_0 = 1$ and $C_1 = C_2 = C_3 = .5$ for the tree 2. For trees 3 and 4 the exact throughputs were calculated using Mathematica and the Qlib traffic theory function library [23].



(a) 2-link line.

(b) 3-link parking lot.

(c) Trees 1, 2 and 4.

(d) Tree 3.

Figure 6.30: Comparison between analytical and simulated throughputs.

In Figure 6.30 the exact and simulated throughputs are shown in four different cases. In all these cases the traffic was homogenous.

Figure 6.30(a) depicts the throughputs in a 2-link line and Figure 6.30(b) in the case of a 3-link parking lot network. In Figure 6.30(c) the throughputs of trees 1, 2 and 4 are shown. The uppermost curves correspond to tree 1 and the lowest to tree 4. Figure 6.30(d) presents the throughput of tree 3.

In all the cases presented the simulated throughput follows exactly the analytically produced throughput. This result realizes also in the heterogenous cases. The results give support to the assumption of the validity of the simulator.

Chapter 7

Conclusions

7.1 Summary

Most traffic in current data networks is elastic, i.e. the rates of traffic flows adjust to use all bandwidth available. The main objective of bandwidth sharing is to use all the available bandwidth without violating the constraints and maintain a certain fairness. The achieved fairness depends on the used fairness criterion. Different fairness criteria favor or discriminate single sources or whole traffic classes on different basis.

The traditional approach has been the max-min fairness [2, 10, 19, 25], which tends to allocate the same share to each flow class. Another central fairness criterion is the proportional fairness [16], in which deviation from the fair allocation causes a negative average change. As a mathematical notion fairness can be thought of as an optimization problem (3.3), where the objective is to find a rate allocation that minimizes or maximizes a utility function specific for the used fairness criterion [10, 11, 16]. A general utility function is given by (3.4). With different values of α and w_r different fairness criteria are achieved.

The optimal allocation provided by these utility-based fairness criteria is considered in a static network scenario where number of flows is fixed. In a dynamic network scenario using the optimal bandwidth sharing policy adapted from static scenario can lead to non-optimal results. In most cases the traffic characteristics have to be known in detail (e.g. distributions of flow sizes and arrival times, session structure). Also analysis of flow-level characters becomes difficult excluding most simple network cases. Utility-based fairness criteria have been proven to be sensitive in the sense that the steady state distribution depends on detailed traffic characters, which explains the difficulty of flow-level analysis.

The balance property (4.1) implies the insensitivity of an allocation . Balanced fairness [6] represents a new allocation policy that can be considered as the most efficient insensitive allocation. Insensitivity makes it possible that know-

ing the mean values of distributions of traffic characteristics provides enough information to derive the flow-level characteristics in a fixed network scenario. When bandwidth allocation is based on balanced fairness, the distribution of number of flows in progress and expected throughput depend only on the average traffic load of each flow class.

In this study the effect of allocation policies on flow-level characteristics was studied via simulations under different network topologies. Used topologies were linear, parking lot, grid, hypercycle and tree networks. Three different allocation policies were used in simulations – balanced, max-min and proportional fairness.

In line network topology max-min fairness provided better throughput on the long route and penalized the shorter ones more than balanced fairness. Balanced fairness and proportional fairness provided the same throughput. With homogenous traffic all the three allocation policies provide the same throughput in grid network. In the hypercycle networks all the allocation policies provided the same throughput with homogenous traffic. With the heterogenous traffic differences between max-min fairness and balanced fairness (as well as with proportional fairness) were minor. In the tree network topologies balanced fairness provided greater throughput for the shorter route flow classes, and max-min fairness for the longer route flow classes. It was verified in the three link parking lot network that in tree networks max-min and proportional fairness coincide.

Simulations concerning a time scale change provided information about the sensitivity of the allocation policies. Balanced fairness remained insensitive in all the simulation cases as was supposed. Max-min fairness was sensitive in linear, grid and tree network topologies. Proportional fairness was insensitive in the linear and grid networks, which is verified by the statement that proportional fairness coincides with balanced fairness in homogenous hypercubes. It also seems that in hypercycle networks balanced fairness, max-min fairness as well as proportional fairness are insensitive.

Three different flow size distributions, unimodal, bimodal and uniform distribution, were used in the simulations. The results suggest that both max-min and proportional fairness are rather insensitive with respect to the flow size distribution.

The linear property (4.5) turned out to be valid for all allocation policies in the studied cases. The result implies that the sensitivity of the max-min and proportionally fair allocations is quite weak in the cases studied.

Simulation results of the variance of flow durations showed that different allocation policies produced the same value, no significant differences were seen. Also the shape of the variance as a function of the load was the same for all the network setups. These results imply that the variance of the flow duration is robust with respect to the allocation policy.

Overall, the simulations suggest that the performance of the max-min fairness and the proportional fairness is relatively close to the performance of balanced fairness. Also these utility-based allocations seem to be rather insensitive.

Max-min fairness is used in practical real life applications, thus balanced fairness provides an effective tool to approximate and evaluate the performance of these applications in an analytical way.

7.2 Further work

The concept of balanced fairness is still quite new and provides plenty of different directions for the future research.

The calculation of the values of the balance function via recursion is, in practice, slow and requires excessively memory. One possibility is to try to find out more effective ways to calculate these values.

Also the classical, utility based allocation policies still provide lots of subjects for research. One quite interesting question is the existence of the global and distributed algorithms for calculating the allocations. These exist for the max-min allocation policy, and the allocation is a special solution of the optimization problem (3.3). Thus, could some similar algorithm be derived for the other allocations that are solutions for the same problem with different parameters of the target function (3.5)?

Bibliography

[1] Mokhtar S. Bazaraa, Hanif D. Sherali, and C. M. Shetty. *Nonlinear Programming : theory and algorithms*. John Wiley & Sons, second edition, 1993.

[2] Dimitri Bertsekas and Robert Gallager. *Data Networks*. Prentice Hall, second edition, 1992.

[3] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.

[4] Thomas Bonald and Laurent Massoulié. Impact of fairness on internet performance. In *Proceedings of the SIGMETRICS/Performance*, pages 82–91, 2001.

[5] Thomas Bonald and Alexandre Proutière. Insensitive bandwidth sharing. In *Proceedings of the IEEE Globecom 2002*, pages 82–91, 2002.

[6] Thomas Bonald and Alexandre Proutière. Insensitive bandwidth sharing in data networks. *Queueing Systems*, 44:69 – 100, 2002.

[7] Thomas Bonald and Alexandre Proutière. Insensitivity in processor-sharing networks. *Performance Evaluation*, 49:193 – 209, 2002.

[8] Thomas Bonald and Alexandre Proutière. On performance bounds for balanced fairness. To appear in Performance Evaluation, 2003.

[9] Thomas Bonald, Alexandre Proutière, Jim Roberts, and Jorma Virtamo. Computational aspects of balanced fairness. In *Proceedings of the 18th International Teletraffic Congress (ITC-18)*, pages 801 – 810, September 2003.

[10] Jean-Yves Le Boudec. Selected lecture notes. SSC-D02, Ecole Polytechnique Fédérale de Lausanne, October 1998.

[11] Jean-Yves Le Boudec. Rate adaptation, congestion control and fairness: A tutorial. Ecole Polytechnique Fédérale de Lausanne, December 2000.

[12] Anna Charny. An algorithm for rate allocation in a packet-switching network with feedback. Master's thesis, Massaschusetts Institute of Technology, May 1994.

[13] Anna Charny, David D. Clark, and Raj Jain. Congestion control with explicit rate indication. In *Proceedings of the IEEE International Conference on Communications*, pages 1954–1963, 1995.

[14] Guy Fayolle, Arnaud de La Fortelle, Jean-Marc Lasgouttes, Laurent Massoulié, and Jim Roberts. Best-effort networks: Modeling and performance analysis via large networks asymptotics. In *Proceedings of the INFOCOM'01*, pages 709 – 716, 2001.

[15] Paul Hurley, Jean-Yves Le Boudec, and Patrick Thiran. A note on the fairness of additive increase and multiplicative decrease. In *Proceedings of the 16th International Teletraffic Congress (ITC-16)*, June 1999.

[16] Frank Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33 – 37, January 1997.

[17] Frank Kelly, Aman Maulloo, and David Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49, 1998.

[18] Leonard Kleinrock. *Queueing Systems*, volume 2: Computer Applications. John Wiley & Sons, 1976.

[19] Laurent Massoulié and Jim Roberts. Bandwidth sharing: Objectives and algorithms. In *Proceedings of the INFOCOM'99 (3)*, pages 1395 – 1403, 1999.

[20] Jeonghoon Mo and Jean Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556 – 567, 2000.

[21] David Musser and Atul Saini. *STL tutorial & reference guide: C++ programming with standard template library.* Addison-Wesley, 1996.

[22] Teunis J. Ott. The sojourn time distributions in the M/G/1 queue with processor sharing. *Applied Probability*, 21:360 – 378, 1984.

[23] Qlib traffic theory function library. Networking Laboratory, Helsinki University of Technology. http://www.netlab.hut.fi/qlib/.

[24] Bożidar Radunović and Jean-Yves Le Boudec. A unified framework for max-min and min-max fairness with applications. In *Proceedings of the 40th Annual Allerton Conference on Communication, Control, and Computing*, October 2002.

[25] Jim Roberts and Laurent Massoulié. Bandwidth sharing and admission control for elastic traffic. In *Proceedings of the ITC Specialist Seminar*, October 1998.

[26] Richard Serfozo. *Introduction to Stochastic Networks.* Springer, 1999.

[27] A. Speetzen, M. Junius, M. Steppler, M. Bter, and D. Pesch. CNCL. CNCL Reference Manual, 1998. http://www.comnets.rwth-aachen.de/doc/cncl/index.html.

[28] Bjarne Stroutrup. *The C++ Programming Language*. Addison Wesley, 1997.

[29] Vesa Timonen. Static Fairness Criteria in Telecommunications. Special assignment, Networking Laboratory, Helsinki University of Technology.

[30] Milan Vojnović, Jean-Yves Le Boudec, and Catherine Boutremans. Global fairness of additive-increase and multiplicative-decrease with heterogeneous round-trip times. In *Proceedings of the INFOCOM'00 (3)*, pages 1303 – 1312, 2000.

Appendix A

Karush-Kuhn-Tucker conditions

In this appendix the Karush-Kuhn-Tucker conditions are presented for a general optimization problem as defined in [1]. Results are applied to solve the proportionally fair allocation in the hypercycle network setup.

General optimization problem and optimality conditions

Let us define a general optimization problem P:

$$\min_{\phi} f(\phi) \tag{A.1}$$

subject to

$$g_i(\phi) \leq 0, \quad i = 1, \ldots, m, \tag{A.2}$$
$$h_i(\phi) = 0, \quad i = 1, \ldots, l, \tag{A.3}$$
$$\phi \in \Omega,$$

where Ω is a nonempty open set in $\mathbb{R}^n$, and functions f, g and h are as follows:

$$\begin{aligned} f &: \mathbb{R}^n \mapsto \mathbb{R}, \\ g_i &: \mathbb{R}^n \mapsto \mathbb{R}, \quad i = 1, \ldots, m, \\ h_i &: \mathbb{R}^n \mapsto \mathbb{R}, \quad i = 1, \ldots, l. \end{aligned}$$

Necessary conditions

Let ψ be a feasible solution to problem P (A.1), and let us define set $\mathcal{I} = \{i : g_i(\psi) = 0\}$. Now, let us assume that following requirements hold: Functions f

and g_i, $i \in \mathcal{I}$, are differentiable at point ψ, g_i is continuous at point ψ for all $i \notin I$, and h_i is continuously differentiable at point ψ for all $i = 1, \ldots, l$. Further, functions $\nabla g_i(\psi)$ for $i \in \mathcal{I}$ and $\nabla h_i(\psi)$ $i = 1, \ldots, l$ are linearly independent.

If ψ is a local solution for problem P, there exists unique scalars u_i for $i \in \mathcal{I}$ and v_i for $l = 1, \ldots, l$, such that

$$\nabla f(\psi) + \sum_{i \in \mathcal{I}} u_i \nabla g_i(\psi) + \sum_{i=1}^{l} v_i \nabla h_i(\psi) = 0 \tag{A.4}$$
$$u_i \geq 0 \quad \forall i \in \mathcal{I}.$$

It is said that the Karush-Kuhn-Tucker (KKT) conditions hold at ψ if the *Lagrangian multipliers* u_i and v_i exist fulfilling condition (A.4).

Sufficient conditions

Let ψ be a feasible solution to problem P (A.1), and let us denote $\mathcal{I} = \{i : g_i(\psi) = 0\}$. Let us assume that following requirements hold: The KKT conditions hold at point ψ, and sets $\mathcal{J}$ and $\mathcal{K}$ are defined as follows: $\mathcal{J} = \{i : v_i > 0\}$ and $\mathcal{K} = \{i : v_i < 0\}$

Solution ψ is a global optimum of the problem P, if function f is pseudoconvex at point ψ, function g_i is quasiconvex at point ψ for all $i \in \mathcal{I}$, function h_i is quasiconvex at point ψ for all $i \in \mathcal{J}$ and h_i is quasiconcave at point ψ for all $i \in \mathcal{K}$.

Proportional fairness in hypercycle topology

In hypercycle topology (see Section 4.5.2) a bandwidth allocation ϕ is considered as proportionally fair, if it is the solution of the following optimization problem:

$$\max_{\phi} \sum_{r \in \mathcal{R}} x_r \log \frac{\phi_r}{x_r} \tag{A.5}$$

subject to

$$\sum_{\substack{r \in \mathcal{R} \\ r \neq l}} \phi_r \leq C_l, \quad l = 1, \ldots, L, \tag{A.6}$$
$$\phi_r \geq 0 \quad \forall r \in \mathcal{R}.$$

Let us rewrite (A.5) in the following form:

$$\max_{\phi} \sum_{r \in \mathcal{R}} x_r \log \frac{\phi_r}{x_r} = -\min_{\phi} \left(-\sum_{r \in \mathcal{R}} x_r \log \frac{\phi_r}{x_r} \right).$$

Further, from (A.6) we get

$$\sum_{\substack{r\in\mathcal{R}\\ r\neq l}} \phi_r \leq C_l \Leftrightarrow \sum_{\substack{r\in\mathcal{R}\\ r\neq l}} \phi_r - C_l \leq 0.$$

Thus, using notation of the previous section, we have

$$\begin{aligned} f(\phi) &\stackrel{\text{def}}{=} -\sum_{r\in\mathcal{R}} x_r \log\frac{\phi_r}{x_r}, \\ g_l(\phi) &\stackrel{\text{def}}{=} \sum_{\substack{r\in\mathcal{R}\\ r\neq l}} \phi_r - C_l \leq 0 \quad l = 1,\ldots,L. \end{aligned}$$

Condition $\phi_r \geq 0$ for all $r \in \mathcal{R}$ can be neglected because these constraints clearly are not active; if some $\phi_r \to 0$, the value of the target function $f(\phi) \to +\infty$. Thus, at the optimum condition $\phi_r > 0$ must hold for all $r \in \mathcal{R}$, and thus the corresponding Lagrangian multipliers are zero.

For the KKT conditions the gradients are derived[1]:

$$\begin{aligned} \nabla f(\phi) &= -\left(\frac{x_1}{\phi_1},\ldots,\frac{x_K}{\phi_K}\right), \\ \nabla g_l(\phi) &= (1_{l\neq 1}, 1_{l\neq 2},\ldots,1_{l\neq L}), \quad l = 1,\ldots,L. \end{aligned}$$

Now, the KKT conditions are

$$\begin{aligned} \nabla f(\phi) &+ \sum_{l\in\mathcal{L}} u_l \nabla g_l(\phi) \\ &= -\left(\frac{x_1}{\phi_1},\ldots,\frac{x_K}{\phi_K}\right) + \sum_{l\in\mathcal{L}} u_l\,(1_{l\neq 1}, 1_{l\neq 2},\ldots,1_{l\neq L}) \\ &= -\left(\frac{x_1}{\phi_1},\ldots,\frac{x_K}{\phi_K}\right) + \left(\sum_{l=2}^{K} u_l,\ldots,\sum_{\substack{l=1\\ l\neq i}}^{K} u_l,\ldots,\sum_{l=1}^{K-1} u_l\right) \\ &= \left(\sum_{l=2}^{K} u_l - \frac{x_1}{\phi_1},\ldots,\sum_{\substack{l=1\\ l\neq i}}^{K} u_l - \frac{x_i}{\phi_i},\ldots,\sum_{l=1}^{K-1} u_l - \frac{x_K}{\phi_K}\right) = 0 \end{aligned}$$

and

$$\begin{aligned} u_l g_l(\phi) &= 0, \quad l = 1,\ldots,L, \\ u_l &\geq 0, \quad l = 1,\ldots,L. \end{aligned}$$

[1]Function 1_{cond} is defined as follows: $1_{\text{cond}} \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if cond is true,} \\ 0, & \text{otherwise.} \end{cases}$

The proportionally fair allocation is now solution of non-linear system of equations:

$$\begin{aligned} \sum_{\substack{l=1 \\ l \neq i}}^{K} u_l - \frac{x_i}{\phi_i} = 0, \quad i = 1, \ldots, K, \\ u_l g_l(\phi) = 0, \quad l = 1, \ldots, L. \end{aligned} \tag{A.7}$$

That is, the system consists of $K + L = 2K$ equations and of $2K$ unknown variables ϕ_i and u_l. Feasible solution necessitates that Lagrangian multipliers $u_l \geq 0$.

The function f is convex, and the feasible region defined by inequalities (A.6) is also convex[2]. Thus, the sufficient KKT conditions hold, and the solution of (A.7) provides the global optimum.

Proportional fairness in 4-link hypercycle

In the case of 4-link hypercycle we have $C_l \equiv 1$ and $K = L = 4$. Thus, the system of equations (A.7) gets the following form:

$$\begin{cases} u_2 + u_3 + u_4 - \dfrac{x_1}{\phi_1} = 0, \\ u_1 + u_3 + u_4 - \dfrac{x_2}{\phi_2} = 0, \\ u_1 + u_2 + u_4 - \dfrac{x_3}{\phi_3} = 0, \\ u_1 + u_2 + u_3 - \dfrac{x_4}{\phi_4} = 0, \end{cases} \quad \text{and} \quad \begin{cases} u_1(\phi_2 + \phi_3 + \phi_4 - 1) = 0, \\ u_2(\phi_1 + \phi_3 + \phi_4 - 1) = 0, \\ u_3(\phi_1 + \phi_2 + \phi_4 - 1) = 0, \\ u_4(\phi_1 + \phi_2 + \phi_3 - 1) = 0. \end{cases}$$

Let us denote $\Sigma_x = \sum_{i=1}^{4} x_i$. The solution of the system of equations is

$$\begin{cases} u_i = \Sigma_x - 3x_i, \\ u_j = \Sigma_x - 3x_j, \\ u_k = \Sigma_x - 3x_k, \\ u_l = \Sigma_x - 3x_l, \\ \phi_i = \phi_j = \phi_k = \phi_l = \dfrac{1}{3}, \end{cases} \quad \text{or} \quad \begin{cases} u_i = \dfrac{x_j \Sigma_x}{x_i + x_j}, \\ u_j = \dfrac{x_i \Sigma_x}{x_i + x_j}, \\ u_k = u_l = 0, \\ \phi_i = \phi_j = \dfrac{x_i + x_j}{\Sigma_x}, \\ \phi_k = \dfrac{x_k}{\Sigma_x}, \\ \phi_l = \dfrac{x_l}{\Sigma_x}, \end{cases}$$

[2]For proof, see e.g. [29].

or

$$\begin{cases} u_i = \dfrac{(-x_i + x_j + x_k)\Sigma_x}{x_i + x_j + x_k}, \\ u_j = \dfrac{(x_i - x_j + x_k)\Sigma_x}{x_i + x_j + x_k}, \\ u_k = \dfrac{(x_i + x_j - x_k)\Sigma_x}{x_i + x_j + x_k}, \\ u_l = 0, \\ \phi_i = \phi_j = \phi_k = \dfrac{x_i + x_j + x_k}{2\Sigma_x}, \\ \phi_l = \dfrac{x_l}{\Sigma_x}, \end{cases}$$

where $\{i, j, k, l\}$ is some permutation of tuple $\{1, 2, 3, 4\}$. The optimal allocation ϕ is now function of the network state x and it is defined explicitly by the condition $u_i \geq 0$, $i = 1, \ldots, 4$.