The Fairness Challenge in Computer Networks

Robert Denda[1,2], Albert Banchs[3], and Wolfgang Effelsberg[2]

[1] ENDITEL Endesa Ingeniería de Telecomunicaciones S.A.,
Departamento de I+D, Sevilla, Spain,
`rdenda@enditel.es`
[2] Lehrstuhl für Praktische Informatik IV, Universität Mannheim, Germany,
`{denda,effelsberg}@pi4.informatik.uni-mannheim.de`
[3] C&C Research Laboratories, NEC Europe Ltd., Heidelberg, Germany,
`Albert.Banchs@ccrle.nec.de`

Abstract. In this paper, the concept of fairness as a future field of research in computer networks is investigated. We motivate the need of examining fairness issues by providing example future application scenarios where fairness support is needed in order to experience sufficient service quality. We further demonstrate how fairness definitions from political science and in computer networks are related and, finally, contribute with this work to the ongoing research activities by defining the *fairness challenge* with the purpose of helping direct future investigations to the white spots on the map of research in fairness.

1 Introduction

Fairness in computer networks deals with the distribution of network resources among applications; i.e., *fairness* is achieved when network resources are distributed in a *fair* way. Investigating *fairness* in computer network aims at two goals. The first goal is to improve the behaviour of networking architectures by adding the valuable concept of distributing resources fairly, which should be considered both for existing and for new scenarios. We call this concept *macro-fairness*, because it deals with the distribution of the overall network resources.

The second goal is to enable new (fair) applications that are currently not implemented in existing networks for various reasons. We refer to this concept as *micro-fairness*. *Micro-fairness* aims at achieving a fair distribution of the network resources at a much finer granularity and is necessary to provide the needed service quality for certain applications. For example, with *micro-fairness*, two packets leaving a single source at some point in time for two different destinations might be required to reach their destinations at exactly the same moment.

Macro-fairness has been studied to a big extend, whereas *micro-fairness* still lacks a lot of further investigation. In the remaining paragraphs of this introduction, we classify *micro-fairness* in the hierarchy of needs of users in computer networks and motivate it via new (fair) application scenarios that require a certain level of fairness.

Current computer networks fulfill most needs of their users. Email, file transfer, WWW access, IP-telephony, video-conferencing, etc. are widely deployed and

J. Crowcroft, J. Roberts, and M. Smirnov (Eds.): QofIS 2000, LNCS 1922, pp. 208–220, 2000.
© Springer-Verlag Berlin Heidelberg 2000

more or less well supported by most computer networks. Nevertheless, there exist real-time application scenarios that are yet not implemented, examples of which are tele stock trading (we think of intra-day stock trading from home), large-scale distributed real-time games, real-time tele auctions, etc. We believe that the main reasons for these applications not to be deployed are insufficient existing networking mechanisms.

At an abstract level, the needs of users in computer networks can be structured hierarchically as a pyramid (see Figure 1), which can be somewhat likened to Maslow's pyramid of human needs [1].

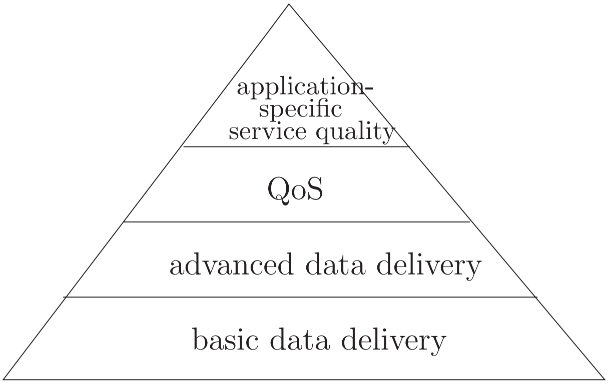

Fig. 1. Pyramid of users' needs in computer networks

The minimum level of users' needs in a computer network is *basic data delivery functionality*, which provides asynchronous off-line data delivery with no explicit requirements on the time or duration of delivery. At a service level, simple email service can be regarded as an example.

The next higher level in the pyramid adds *advanced data delivery functionality*: users want to be provided with synchronous, interactive, and/or two-way on-line data delivery. Examples of such services are file transfer and WWW. Note that at this level, there are still no hard bounds on the time, duration and delay of data delivery.

The third level adds *quality of service (QoS).* Users want to perform real-time multimedia communication involving data streaming for audio/video. Therefore, they need a communication system that offers sufficient quality of service for the traffic. Note that in this context, quality of service means abstract user requirements concerning the data delivery and does not necessarily mean that the communication system needs explicit QoS support: even in well-provisioned best-effort networks users might be content with the quality of service they receive, while the network itself has no explicit mechanisms for providing QoS. Examples of communication that need a certain level of quality of service are telephony, video-conferencing, tele-education, tele-medicine, and many more. At the highest level of the pyramid, the technical feasibility for most applications is already assured through the lower three levels. Nevertheless, there are application scenarios where mechanisms and guarantees are needed that are beyond current technology. Such application scenarios include real-time tele-stock trading, tele-voting, large-scale distributed games and real-time electronic auctions.

For these applications, quality of service provision in a network alone does not suffice: for example, users want not only to be sure that the communication system provides the service with its required QoS guarantees, they also want a guarantee that they have at least the same opportunities as their competitors. *Fairness* is therefore one main aspect of the highest level of the pyramid. In addition to fairness, other needs, such as network availability, security, etc. are also located at the highest level. Note that especially due to the heterogeneity of current networks, the requirements of such applications cannot be fulfilled through QoS guarantees only.

The lower three levels, which are necessary to make network services functional, have been addressed in detail in literature and are still worthwhile a lot of further discussion (to which this paper will not contribute). Unfortunately, high-level concepts to provide the actual service quality for many applications, in particular fairness concepts in computer networks, have not been examined as thoroughly. Especially, it seems that an overall view of fairness concepts is missing. This paper is intended to fill this gap and to shed some light on the importance of fairness concepts for computer networks.

The following parts of the paper are organized as follows. Section 2 borrows models from political science to give both a common sense definition and a formal description of the concept of maximizing *welfare*. The concept of maximized *welfare* in political science corresponds to the commonly used definition of *fairness* in computer networks. In Section 3, existing approaches to *macro-fairness* and new aspects that come with *micro-fairness* are presented.

Section 4 concludes this work with an overview and discussion of open issues and challenges in fairness research for computer networks, thereby defining the *fairness challenge* with various facets that should be investigated within future research.

2 What Is Fairness ?

The concept of fairness has been studied in various scientific areas. Most thorough and theory-based approaches arose from the field of political science and political economics: fair allocations of consumption bundles in an economy have been investigated, and a common sense definition of a fair allocation is given as "an allocation where no person in the economy prefers anyone else's consumption bundle over his own" [2], i.e., "a fair allocation is free of envy" [3]. Even this very general definition indicates the conceptual difficulty of fairness: in order to ensure fairness in a system, all system entities have to be satisfied with their allocated share of the system's goods. Therefore, the distribution mechanism has to take into account the subjective preferences of the system entities.

The famous problem of how to divide a cake fairly into pieces for a number n of hungry and competing cake eaters has been examined in various early studies in the field of econometrics (see, e.g., [4,5] and [6]). The cake division problem illustrates well the difficulty of the fairness concept for simple distribution problems[1]. It should be noted that any algorithm that solves the cake division problem requires active participation of the cake eaters, i.e., they have to signal their preferences to make sure that they are content with the outcome.

In computer networks, the situation is very similar: resources have to be distributed among competing users of the computer network, which can be likened to distributing pieces of cake to competing cake eaters. The practical problem with applying the cake division algorithm to resources in computer networks is that it would require active signaling of the users' preferences upon all changes of resource distribution in the network. For scalability reasons, this approach is clearly not feasible.

In order to avoid this problem of continuous signaling, a concept to express the preferences of a user has been developed: *utility functions* (see [7]). In order to analyze and formalize computer networks, utility functions are defined for networking applications as functions that map a service delivered by the network[2] into the performance of the application for that service. Utility can be considered a measure of how much a user would be willing to pay for the service [8]. In Section 3, we will see that *macro-fairness* is related to *individualistic* utility functions, while *micro-fairness* relates to another type of utility function, which we call *group-constrained* utility.

2.1 Pareto-Efficiency and Welfare

The concepts *pareto-efficiency* and *welfare* in political economics are strongly related to the concept of *fairness* and will be briefly revised to provide a more theoretical definition of *fairness*.

Let us follow [9] to define *pareto-efficiency*:
In general, an allocation ω of resource bundles $(x_1, ..., x_k)$ is *feasible* if the *excess demand* $z(\omega)$ for that allocation is ≤ 0. The *excess demand* is the aggregate vector of demands reduced by the aggregate vector of resources available; thus, an allocation is *feasible*, if the aggregate supply of resources exceeds or equals the aggregate resource requirements of users.
A *utility allocation* u_i represents user i's utility of an allocation ω for a resource

---

[1] An algorithm that solves the cake division problem proposed by Banach and Knaster (see [4]) is very simple: a knife is moved at constant speed over the cake and is poised at each instant, s.t. it could cut a unique slice of the cake. Thus, the potential slice increases monotonely until it becomes the entire cake. The first person to indicate satisfaction with the slice determined by the position of the knife receives that slice (if two persons indicate satisfaction simultaneously, the slice is given to any one of them). Then, the rest of the cake is distributed using the same constructive method.

[2] The service describes all relevant measures, such as delay, throughput, loss rate, etc.

bundle $(x_1, ..., x_k)$. A utility allocation u_i^1 is *dominated* by u_i^2, if u_i^2 is feasible and $u_i^2 > u_i^1$, i.e., if u_i^2 is prefered to u_i^1.
A utility allocation u_i^1 is *pareto-efficient*, if it is *feasible* and *not dominated* by any other feasible utility allocation u_i^2. In more general terms, "a situation is *pareto-efficient*, if there is no way to make any person better off without hurting anybody else" (see [7], Section 16.9).

Pareto-efficiency is clearly a desirable criterion of an allocation. Nevertheless, it is only a weak criterion. The problem is that also an allocation, where one user gets everything can be pareto-efficient, and this allocation is certainly not fair.

Welfare extends the concept of pareto-efficiency in a certain manner: the basic problem of *welfare* (see, e.g., [10]) is to determine, which of the feasible allocations $\omega(x_1, ..., x_k)$ should be selected. For that reason, it is assumed that there exists a general *welfare function* $W(u_1, u_2, ..., u_n)$ that aggregates the individual utility functions u_i of the users. A welfare function is required to be increasing in all of its arguments.

It can be shown that any feasible allocation of maximum welfare must necessarily be pareto-efficient[3]. For that reason, it seems to be very desirable to find an appropriate welfare function and perform the maximization in order to receive maximum welfare while being pareto-efficient. The problem with this approach is the welfare function itself, since it is not clear how to perfectly aggregate individual preferences.

2.2 Examples of Welfare Functions

For different purposes, different examples of welfare functions exist, each corresponding to a different criterion of welfare. For an introductory overview and comparison of different criteria of welfare see [12].

One criterion is the *maximin criterion*, which corresponds to the Rawlsian welfare function $W(u_1, ..., u_n) = min(u_1, ..., u_n)$. The maximin criterion weighs only the utility of the worst-off user.

The *sum of utilities criterion* corresponds to the classical utilitarian welfare function $W(u_1, ..., u_n) = \sum_i u_i$. In contrast to the maximin criterion, this criterion weighs the utility of each user equally.

Both these criteria have certain problems: the maximin criterion does not weigh improvements of those who are not least well off; and the sum of utilities criterion might prefer a situation where some users are very happy and others are very miserable, rather than allowing an allocation where all users are "just happy", i.e., in between extremely happy and very miserable.

These two criteria can be regarded as the limiting cases. In between, there exist a whole range of various compromise welfare functions all aiming at different goals. One example is the *weighted-sum-of-utilities welfare* function $W(u_1, ..., u_n) = \sum_i a_i u_i$, where a_i is a weight assigned to u_i, thereby expressing individual priorities between different users. Another example is the *sum-of-square-roots*

---

[3] For a simple proof, see, e.g., [11].

function $W(u_1, ..., u_n) = \sum_i \sqrt{u_i}$, where users with smaller utilities are given higher relative priorities.

Yet another, but very interesting welfare function is the *sum-of-logs* function $W(u_1, ..., u_n) = \sum_i log(u_i)$, which corresponds to the so-called *Nash criterion*. Note that the *Nash criterion* maximizes the product of additional utilities compared to the status quo. It has been first described by Nash [13] as the solution to the bargaining game in game theory. This maximized welfare function has the property that its outcome is not affected by any linear transformation of a user's utility scales: if a user's utility function is transformed using a positive linear transformation, the solution to maximizing the welfare function yields an allocation which is identical to the allocation before transformation. Therefore, this type of welfare function is independent of changing the scales of the individual utility functions, and inter-user comparisons of utility are not required, which is an interesting property, since the *transferability* of utility remains questionable.

3 Fairness Concepts in Computer Networks

In current computer networks, the term *fairness* corresponds to the concept of maximum welfare as defined in the previous section. The following subsections present the most common fairness definitions using the terminology presented in Section 2, give an overview of existing concepts, mechanisms and open questions in computer networks regarding *macro-fairness*, and introduce the new challenges entailed by *micro-fairness*.

3.1 Examples Fairness Criteria

In the following two paragraphs, we briefly demonstrate how the most common fairness criteria in packet-based communication networks, *maxmin fairness* and *proportional fairness*, can be defined using the concepts of maximum welfare presented in Section 2.

Maxmin Fairness The most popular fairness concept in computer networks, *maxmin-fairness* [14], corresponds to the Rawlsian welfare function $W(u_1,\ldots, u_n) = min(u_1, \ldots, u_n)$ with the individualistic utility functions $u_i(x_1, \ldots, x_n) = x_i,\ \forall i \epsilon \{1, ..., n\}$, i.e., maxmin-fairness yields a solution $x^s = (x^s_1, .., x^s_n)$ for $max(min(x_1, \ldots, x_n))$. A maxmin-fair situation has the property that forall i, x^s_i cannot be increased without simultaneously decreasing x^s_j for some j with $x^s_j \leq x^s_i$ [4].

Proportional Fairness Another interesting fairness criterion is *proportional fairness* (see [16])[5]. A proportional fair allocation is the solution to the welfare

---

[4] For a discussion of *maxmin-fairness*, see, e.g., [15].

[5] Frank Kelly provided quite a substantial amount of work on proportional fairness, which can be found at http://www.statslab.cam.ac.uk/˜ frank .

maximization problem with the welfare function $W(u_1, ..., u_n) = \sum_i log(u_i)$ and individualistic utility functions $u_i(x_1, \cdots, x_n) = x_i$.

It has been demonstrated that additive increase and multiplicative decrease end-to-end congestion control, assuming best effort FIFO queueing with tail dropping inside the network, tends to lead under certain circumstances to *proportional fairness* (see [17,18]). Note, however, that this does not necessarily hold for real Internet scenarios with TCP congestion control (see [19] and [20]).

3.2 Current Fairness Concepts and Mechanisms: Macro-Fairness

Within current computer networks, *macro-fairness* concepts are applied both to medium access control and to data transport. Concerning fair medium access control (MAC), mechanisms have been investigated for shared physical network links such that on average each sender or receiver gets a fair share of the available bandwidth. This issue is of concern both for medium access control in LAN environments, and, more recently, for mobile networks (see, e.g. [21]) and for all-optical networks (see, e.g., [22]). Many related problems to MAC layer fairness, such as for instance fair MAC-layer sharing of a common channel under error conditions, are non-trivial and still require further research.

As for data transport, fairness concepts are relevant for both elastic and real-time traffic [8]. For both types of traffic there exist two approaches to provide fairness: one is providing fairness by defining appropriate cell/packet scheduling and queue management algorithms on networking nodes, whereas the other one is to achieve fairness by end-to-end congestion control mechanisms. When comparing end-to-end fairness mechanisms to queue management fairness mechanisms, it can be noted that the second type results in statistical on-average fairness, whereas queueing and scheduling mechanisms allow for a more precise control for fair rate allocations and have a shorter response time to adjust to new network load situations.

Note that fairness issues that have been addressed in current computer networking research mostly concern the problem of fair bandwidth distribution among competing flows. Fair delay management, fair loss rate distribution, and fair jitter control have hardly been addressed at all levels of abstraction, which, in our opinion, is insufficient for future applications with fairness requirements.

Queue Management Mechanisms for Fairness Providing fairness through queue management and scheduling mechanisms is an approach that has a high impact on the network's architecture, since the fairness algorithms are implemented on switches or routers. But when supported, it can provide the most efficient, flexible and exact mechanism for fairness.

For example, in the ATM TM 4.1 specification [23], various bandwidth related fairness criteria for the ABR service are defined.

For the datagram network case, fairness definitions can be implemented at packet schedulers on routers. For example, there exists a whole range of fair queueing algorithms. For some early examples, see [24] and [25]

The approach of using fair queueing to provide fairness is, for instance, taken by the *user-share differentiation (USD)* scheme [26], which is a proposal for differentiated services [27] that ensures that the bandwidth allocated to traffic from a user is in proportion to the user's share negotiated with the service provider. Implementations of schemes like *USD* use extended versions of fair queueing algorithms like *weighted fair queueing* [25] or variations of it (e.g., *worst-case fair weighted fair queueing* [28], *self-clocked fair queueing* [29], *deficit round robin* [30]).

Although these queueing algorithms lead to a more fair bandwidth distribution among competing and not necessarily all well-behaving flows, they have the disadvantage of operating on a per-flow or per-user basis, the scalability, robustness and feasiblity of which in high-speed networks are still questionable. This is, because fair queueing algorithms have been designed for congestion control and are usually *stateful*, as opposed to *stateless* congestion control algorithms such as *random early detection (RED)* [31] and its variations.

Other DiffServ approaches to achieve fairness without requiring state at the core nodes include [32] and [33,34].

End-to-end Fairness Mechanisms Existing end-to-end fairness mechanisms are usually implemented by end-to-end congestion control schemes.

The problem with end-to-end fairness mechanisms is that these mechanisms normally only work in a cooperative environment, i.e., if all flows competing for network resources are well-behaved. In the Internet, well-behaved means *tcp-friendly*, which is characterized by the property of behaving similar to a TCP flow through not sending at a higher data rate than a similar tcp flow in the same congestion situation[6]. Still, it is very questionable if tcp-friendliness is a valid assumption in the real world: besides TCP traffic, UDP traffic exists in current IP networks and is, for instance, used for real-time flows. The rate control algorithms of UDP-applications in practice are not always tcp-friendly and therefore harm the overall fairness. In addition, there exists the risk of malicious TCP implementations that are on purpose not tcp-friendly in order to increase their individual throughput on the cost of regular TCP flows. There are currently no restrictive control mechanisms implemented that punish those flows that are not tcp-friendly, unless this punishment is done by queue management on network nodes as described above, which implicitly influences the type of fairness. One possible approach to cope with this problem is to identify tcp-unfriendly flows at the routers and punish them with appropriate dropping policies. For an interesting discussion of solutions to the tcp-unfriendliness problem see [32].

Multicast Concerning network level fairness, multicast packet or cell delivery introduces an additional level of complexity. Following [36], we distinguish between *inter-fairness* and *intra-fairness. Inter-fairness* means that multicast flows should exhibit fair behaviour compared to other, unicast flows. *Intra-fairness* relates to fairness inside the multicast scenario, e.g., different multicast sessions

---

[6] For a detailed discussion on tcp-friendly applications and protocols see [35].

among the same group of senders and receivers should exhibit fairness. We follow the authors of [37] in pointing out that is still not entirely clear how inter-fairness among congestion-controlled multicast and TCP traffic should be defined: should a multicast session to n receivers get the same share as one TCP connection or as n TCP connections ?

Another problem with multicast feedback control in general is the *loss path multiplicity (LPM) problem* [38], i.e., a packet can be lost on any of the end-to-end paths in the multicast tree. If the sending rate is controled by loss indications from all receivers, there is the problem that with an increasing number of such paths the sender will further and further reduce its sending rate until it eventually might cease sending. In [38], it has been shown that with such a scheme of controling the sending rate, maxmin fair sharing of bandwidth between unicast and multicast traffic is impossible to achieve due to the LPM problem.

For multicasting real-time traffic as generated by audio/video applications, *layered multicast* (see, e.g., [39,40] or [41]), is a very interesting mechanism to effectively use network resources in a scenario with heterogenous receivers. Still, fairness issues for layered multicast have only been investigated at a very basic level and open a whole new field of future research. In addition, layered multicast adds another whole new level of complexity to the fairness problem if the different multicast layers operate at different sending rates[7].

3.3 Fair Applications: Micro-Fairness

All existing concepts, mechanisms and examples of *macro-fairness* are defined by *individualistic* utility functions (see Section 3.1), meaning functions of type $u_i(x_1, \ldots, x_n) = f(x_i)$, i.e., the utility of a user i only depends on the resource bundles he/she receives (x_i), but not on any other resource bundle x_j, with $j \neq i$.

We believe that especially for highly competitive applications, maximizing welfare with *individualistic* utility functions cannot correctly represent the semantics of the desired fairness.

For that reason, we like to present an example fairness definition for *micro-fairness* using what we call *group-constrained* utility functions, i.e., utility functions of type $u_i(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$, where f depends of at least one resource bundle x_j with $j \neq i$, where $x_1, \ldots, x_n$ are the resource bundles received by the individual communication group members.

The example, which we have called *group-delay constrained utility*, can be applied to real-time trading or real-time games scenarios, i.e., competitive communication scenarios, where the semantics of the application require each participant of a communication group to perceive at most the same average delay than the other users, in order to be able to compete in a long term.

For the definition of this group-delay constrained utility, we take the following individualistic utility function for delay: $u_i(delay_{user\ 1}, \ldots, delay_{user\ n}) =$

---

[7] A very thorough approach to define and examine multi-rate multicast (inter- and intra-) maxmin fairness has been provided in [42].

$f(delay_{user\ i})$. In order to represent this competition, we extend the utility function with group constraints that represent the dependency of a user's utility on the delay received by the other group members: for all users $i \epsilon \{1, \ldots, n\}$ of a communication group

$$u_i(delay_{user\ 1}, \ldots, delay_{user\ n}) = \ldots$$

$$\ldots = \begin{cases} 0 & \exists j \epsilon \{1, \ldots, n\} : avg\ delay_{user\ i} > avg\ delay_{user\ j} \\ f(delay_i) & otherwise \end{cases} \tag{1}$$

where f is the individualistic utility function described above.

The maxmin fair solution using this new type of utility function leads to strict equality concerning the average delay: $\forall i, j : avg\ delay_{user\ i} = avg\ delay_{user\ j}$, i.e., we have strict (and identical) upper and lower bounds for all users of that communication group. The extension we have introduced represents the strong effect of the competition inside the communication group: all users only consider to be fair an exactly equal situation with respect to the received average delay.

Note that in this example, QoS mechanisms for strict delay bounds could be used to achieve the fair resource distribution, once the value of the delay bound is determined according to the fairness definition using this group-delay constrained utility function.

In client-server application scenarios, the abstract parameter on which the utility depends is the parameter *response time*, which encompasses the two-way transmission delay and some processing delay. In that case, a common mechanism for approximating *micro-fairness* is synchronization via transaction control. For instance, in auction scenarios, synchronization mechanisms are necessary for a fair treatment of the participants: all auctioneers want to get at least the same chance to bet during a certain time slot. In such a scenario, a fair mechanism is to collect (and acknowledge) the bets of all participants in a first step, then to evaluate the synchronized bets and to announce the resulting highest bet as input to the next round. Obviously, such auctioning mechanisms are neither very time efficient nor can they provide exact fairness.

We leave it as a remaining challenge for future research in fairness to provide more efficient mechanisms for *micro-fairness*.

4 Conclusion

We hope to have demonstrated that even though specific fairness issues concerning computer networks have already been investigated to some extent, there is a vast amount of interesting and challenging work left to be done. We would like to motivate the reader to participate in further investigation of the wide range of interesting and challenging open topics in the field of fairness in computer networks by directing him/her to the extensions of fairness we believe to be most important for future research: *extension of definition*, *extension to other QoS parameters* and *extension to new applications*.

4.1 Extension of Definition

Currently, fairness is mainly defined for unicast cell or packet delivery. Other types of delivery, such as *multicast*, *broadcast* or *anycast* [43] require an extension of the fairness definition. Examples for multicast include the aspects of *inter-fairness* and *intra-fairness* for best-effort multicast, congestion-controlled multicast, reliable multicast, and layered multi-rate multicast. Also, solutions to the *LPM* problem for different fairness criteria and multicast scenarios with multiple senders and dynamically joining and leaving receivers are at an early stage and worthwhile further investigation. In all of these topics, research has just begun.

4.2 Extension to Other QoS Parameters

The *extension to other QoS parameters* means to not only apply fairness concepts to bandwidth distribution problems, but also consider the fair management and control of *loss rate*, *delay* and *delay jitter*. We believe that especially fair delay management and fair jitter control have to be considered for future applications that require fairness as part of service quality.

4.3 Extension to New Applications: Micro-Fairness

Extension to new applications deals with the aspect of micro-fairness: the current macro-fairness concept in computer networks has to be extended up to the application level, i.e., fair applications should be supported. We believe that such application-semantic fairness is best supported if the communication channel provides the necessary degree of fairness. An integral and comprehensive approach for fairness provisioning, especially based on non-individualistic utility functions, is needed and available as a new field for future research in fairness.

References

1. A. Maslow, *Motivation and Personality*, Harper & Row, New York, 1954. 209
2. D. K. Foley, "Resource allocation in the public sector," *Yale Econ. Essays*, vol. 7, pp. 73–76, 1967. 210
3. Elisha A. Pazner, "Pitfalls in the Theory of Fairness," *Journal of Economic Theory*, vol. 14, pp. 458–466, 1977. 210
4. H. Steinhaus, "Sur la division pragmatique," *Econometrica (supplement)*, vol. 17, pp. 315–319, 1949. 211
5. L. E. Dubins and E. H. Spanier, "How to cut a cake fairly," *American Mathematical Monthly*, pp. 1–17, 1961. 211
6. Harold W. Kuhn, "On Games of Fair Division," in *Essays in Mathematical Economics*, Martin Shubik, Ed. 1967, pp. 29–37, Princeton University Press. 211
7. Hal R. Varian, *Intermediate Microeconomics - A Modern Approach*, W. W. North & Company, New York/London, fifth edition, 1999. 211, 212
8. Scott Shenker, "Fundamental Design Issues for the Future Internet," *IEEE Journal Selected Areas Communication*, vol. 13, pp. 1176–1188, 1995. 211, 214

9. K. J. Arrow and F. H. Hahn, *General Competitive Analysis*, Oliver and Boyd, Edinburgh, 1971. 211
10. Hal R. Varian, “Distributive Justice, Welfare Economics, and the Theory of Fairness,” *Philosophy & Public Affairs*, vol. 4, no. 3, pp. 223–247, 1975. 212
11. Hal R. Varian, *Microeconomic Analysis*, Norton, New York, third edition, 1992. 212
12. Donald Wittman, “A Diagrammatic Exposition of Justice,” *Theory and Decision*, vol. 11, pp. 207–237, 1979. 212
13. J. F. Nash, “The Bargaining Problem,” *Econometrica*, vol. 18, pp. 155–162, 1950. 213
14. J. Jaffe, “Bottleneck flow control,” *IEEE Transactions on Communications*, vol. 7, no. 29, pp. 954–962, July 1980. 213
15. D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1987. 213
16. Frank P. Kelly, “Charging and rate control for elastic traffic,” *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997. 213
17. Frank P. Kelly, A. K. Maulloo, and D. K. H. Tan, “Rate control for communication networks: shadow prices, proportional fairness and stability,” *Journal of the Operational Research Society*, vol. 49, 1998. 214
18. L. Massoulié and J. Roberts, “Fairness and quality of service for elastic traffic,” *CNET-France Tél’ecom*, Feb. 1998. 214
19. Paul Hurley, Jean-Yves Le Boudec, and Patrick Thiran, “A Note on the Fairness of Additive Increase and Multiplicative Decrease,” in *Proceedings of ITC-16*, Edinburgh, June 1999. 214
20. 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 IEEE INFOCOM2000*, March 2000. 214
21. J. R. Moorman and J. W. Lockwood, “Multiclass Priority Fair Queueing for Hybrid Wired/Wireless Quality of Service Support,” in *Proceedings of WoWMoM99*, Seattle, WA, Aug. 1999. 214
22. M. A. Marsan, A. Bianco, E. Leonardi, A. Morabito, and F. Neri, “All-Optical WDM Multi-Rings with Differentiated QoS,” *IEEE Communications Magazine*, pp. 58–66, Feb. 1999. 214
23. ATM Forum Traffic Management Working Group, “ATM Forum Traffic Management Specification Version 4.1,” *ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0121.000.pdf*, March 1999. 214
24. A. Demers, S. Keshav, and S. Shenker, “Analysis and Simulation of a Fair Queueing Algorithm,” in *Proceedings of ACM SIGCOMM’89*, 1989, pp. 3–12. 214
25. A. K. Parekh and R. G. Gallager, “A Generalized Processor Sharing Approach to Flow Control - the Single Node Case,” in *Proceedings of IEEE INFOCOM’92*, May 1992. 214, 215
26. Z. Wang, “User-Share Differentiation - Scalable Service Allocation for the Internet,” *Internet Draft*, Nov. 1997. 215
27. Y. Bernet et. al., “A framework for differentiated services,” Internet Draft, draft-ietf-diffserv-framework-01.txt, November 1998. 215
28. J. C. R. Bennett and H. Zhang, “WF^2Q: Worst-case Fair Weighted Fair Queueing,” in *Proceedings of IEEE INFOCOM’96*, March 1996, pp. 120–128. 215
29. S. Golestani, “A Self-clocked Fair Queueing Scheme for Broadband Applications,” in *Proceedings of IEEE INFOCOM’94*, April 1994, pp. 636–646. 215
30. M. Shreedhar and G. Varghese, “Efficient Fair Queueing using Deficit Round Robin,” in *Proceedings of ACM SIGCOMM’95*, September 1995, pp. 231–243. 215

31. S. Floyd and V. Jacobson, “Random early detection for congestion avoidance,” *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, July 1993. 215
32. Ion Stoica, Scott Shenker, and Hui Zhang, “Core-Stateless Fair Queueing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks,” *ACM SIGCOMM’98*, pp. 118–130, Oct. 1998. 215
33. Albert Banchs and Robert Denda, “SBSD: A Relative Differentiated Services Architecture based on Bandwidth Sares,” Tech. Rep., University of Mannheim, Germany, February 2000. 215
34. Albert Banchs and Robert Denda, “A Scalable Share Differentiation Architecture for Elastic and Real-Time Traffic,” in *Proc. of the Eighth International Workshop on Quality of Service IWQoS 2000*, Pittsburgh, PA USA, June 2000, pp. 42–51. 215
35. J. Mahdavi, “The TCP-friendly web site,” http://www.psc.edu/networking/tcp_friendly.html. 215
36. I. Rhee, N. Balaguru, and G. Rouskas, “MTCP: Scalable TCP-like Congestion Control for Reliable Multicast,” in *Proceedings of IEEE INFOCOM’99*, New York, NY, Mar. 1999. 215
37. H. A. Wang and M. Schwartz, “Achieving Bounded Fairness for Multicast and TCP Traffic in the Internet,” in *Proceedings of ACM SIGCOMM’98*, Vancouver, Canada, Sept. 1998. 216
38. S. Bhattacharyya, D. Towsley, and J. Kurose, “The Loss Path Multiplicity Problem for Multicast Congestion Control,” in *Proceedings of IEEE INFOCOM’99*, New York, NY, Mar. 1999. 216
39. S. McCanne and V. Jacobson, “Receiver-Driven Layered Multicast,” in *Proceedings of ACM SIGCOMM’96*, Oct. 1996. 216
40. L. Vicisano, L. Rizzo, and J. Crowcroft, “TCP-like congestion control for layered multicast data transfer,” in *Proceedings of INFOCOM’98*, Mar. 1998, pp. 996–1003. 216
41. A. Banchs, W. Effelsberg, Ch. Tschudin, and V. Turau, “Active Multicasting of Multimedia Streams,” in *Proceedings of the IEEE Local Computer Networks Conference LCN’98*, October 1998, pp. 150–159. 216
42. D. Rubenstein, Jim Kurose, and Don Towsley, “The Impact of Multicast Layering on Network Fairness,” in *Proceedings of ACM SIGCOMM’99*, Sept. 1999. 216
43. Christian Huitema, *IPv6: The New Internet Protocol*, Prentice Hall, Englewood Cliffs, New Jersey, 1996. 218