

Quality Feedback Flows in Future Networks

Markus Fiedler

Abstract

The current rise in community-based content sharing and growing popularity of personalised multimedia transmission is accompanied by user-side quality demands. Internet, on the other hand, still provides merely best-effort service and thus only implicit feedback on critical network conditions through packet loss and delays. In order to satisfy future quality demands, feedback on quality problems needs to become much more explicit than what is the case today. In order to highlight the potential for improvement, this paper first identifies and analyses today's feedback flows between users, applications, networks and the corresponding service providers. Based on this inventory, a new kind of feedback system employing self-organising overlay technology is introduced and exemplified. This future-proof system, developed within the European series of Networks of Excellence EuroNGI, EuroFGI and EuroNF, provides added feedback value to the above-mentioned stakeholders and furthermore a smooth transition path from current legacy towards future service management solutions.

Keywords

Quality of Experience; Self-organisation; Autonomic Networking; Service Management; Network Management

1. Introduction

The emergence of new wireless and wired technologies, sinking communication cost and increasingly capable clients, servers and peers have boosted the traffic in the networks. Access speeds are rising in spite of flat rates pressed by competition. Peers and their owners offer and hunger for content and computing power to be shared across the Internet. Many communities have arisen; file sharing and online gaming have been booming for many years; and video sharing is becoming increasingly popular. Virtual reality is experienced through “Second Life” (www.secondlife.com). As compared to some years before, the role of the networked computer has changed from a web-and-mail device to a mediator of multimedia experience. Typical communication tasks within communities include direct communication (chat, voice, video, etc.), content sharing (photos, videos, music, movies, etc.) and joint activities (on-line gaming, discussions, etc.).

The availability and performance of such fancy networked applications has also pushed user expectations. A well-working application includes, while a stalling application excludes a user from participating in the community and thus in desired joint experiences. Content shall be consumable in an undisturbed way, i.e. without spatial and temporal artefacts such as blocks and freezes, and interactive services need to be responsive within typical limits given by user patience. Even though it is commonly said that content downloading is elastic, users are not necessarily ready to wait too long for the desired file to arrive. In cases of disrupted interaction, streaming or download, the blame is often put on the network – typically phrased as “Internet is not working” – no matter the real source of the problem.

No matter which application is used, bad real-time behaviour and responsiveness as well as unpredictable outages and delays may make users disappointed; angry; or abandon the application or the service, maybe forever. That final step is called *churn*, which means leaving one operator in favour of another. As opposed to the rather monopolistic mono-technology situation at times when private Internet communications arose, a user nowadays might be able to choose between a couple of ISPs and also between different access technologies. Users consider churn when another operator offers – or at least seems to offer – a better service-versus-price ratio. From an operator’s point of view, churn is a major point of concern, as it entails a loss of a customer and thus of revenue. Thus, user satisfaction – covered by the notion of *Quality of Experience (QoE)* (Nokia, 2005; ITU-T, 2008) et al. – has become a key ingredient in successful service provisioning. Transparency of network operations in terms of (a) invisibility of the network operations in the unproblematic case and (b) information of the users in case of problems is an important lead in quality insurance, the role of which is becoming increasingly important given the recent changes in the communications landscape.

On this background, this paper focuses on quality feedback issues. Its importance and its types are put forth in Section 2. The quality feedback scenario in current Internet and its shortcomings are discussed in detail in Section 3. Section 4 proposes a generic architecture that is able to provide different kinds of feedback to different stakeholders, even in future scenarios, and Section 5 concludes the paper and summarises directions for future research.

2. The Role of Quality Feedback

Most users are agnostic of network operations until problems appear. A typical example is waiting for a web page to be displayed. According to (Nielsen, 1994), a waiting time of more than one second allows for interruption of thoughts, and ten seconds are the limit for keeping interest on the process. (Rajamony, 2001) claims that more than four seconds waiting time make users feel bored; (Bouch, 2000) reports a limit of five seconds for high user rating of web delivery perception. (Zona, 1999) devises a limit of eight seconds of user readiness to accept delays when doing e-business. A very clear user statement is contained in the study (Bouch, 2000): "If it's slow, I won't give my credit card number." The time intervals indicated above are valid for cases in which nothing happens on the screen until the page appears all at once. Displaying a page gradually while downloading it increases the threshold for high user rating to 39 s (Bouch, 2000). Obviously, as long as the user sees something happen, such as a picture appearing out of the blur or line-by-line, (s)he becomes much more patient. Again from (Bouch, 2000): "As long as you see things coming up it's not nearly as bad as just sitting there waiting and again you don't know whether you're stuck". While waiting, unconscious users may retry and worsen a potential overload situation by un-deliberately carrying out Denial of Service attacks on heavily loaded resources through their retrials. Catastrophes such as September 11 usually lead to break-downs of news services and networks. Many information-hungry people (re-)try, but virtually no one is getting any service anymore. Signalling overload problems to users might cause users to be patient, relax and to retain trust into the system: "I think it's great. . . saying we are unusually busy, there may be some delays, you might want to visit later. You've told me now. If I decide to go ahead, that's my choice." (Bouch, 2000).

Explicit feedback comprises messages used to timely notify other stakeholders of problems that might affect their perception or function, and is in general appreciated even by otherwise agnostic users. Explicit feedback helps to avoid states of uncertainty. This is crucial for user perception and in the context of machine-to-machine communication.

Implicit feedback, on the contrary, is provided by absence of expected action, which points at underlying problems that are not signalled themselves. Other involved parties need to observe the corresponding process and react upon any late or non-delivery of expected data. In order to avoid getting stuck when facing implicit feedback, many applications or protocols implement timeouts, upon which e.g. a new request or retransmission is triggered.

From a user's point of view, explicit feedback provides relief from incertitude about the success of an operation. For instance, a user in front of a web browser that has been showing nothing new during the last ten seconds might consider reloading the page. Beyond the boredom limit of four seconds, the user has spent additional six seconds in front of the browser just waiting and wondering what is going to happen until (s)he finally triggers the retrial, hoping for a better experience this time. For instance, an indication of the expected waiting time or just a message of the type "please hold" might help to relieve the situation. Upon reception of such a message, the user might consider taking action before time is wasted. In daily life, there are many similar situations in which timely feedback provides a relief, e.g. notifications about late flights or trains. In the context of machine-to-machine communication, feedback within well-defined timeframes is crucial for the stability of control loops.

So far in the Internet, there has been some kind of “one-size fits all” attitude. In particular, Internet delivers datagrams in a best-effort manner, which means that network-layer communications happens at own risk: Users receive implicit feedback from the network through delayed or non-delivery of packets. Transport level protocols such as the *Transport Control Protocol* (TCP) were designed to ensure delivery and impose flow control, both at the expense of data delay. Application-layer protocols such as the *Real-time Transport Protocol* (RTP) carry out measurements in order to allow for dynamic adaptation to the circumstances in end-to-end transport. Applications, on the other hand, take things in their own hands. A prominent example is the selfish behaviour of Skype that may duplicate the sent amount of data in view of a bottleneck, which improves QoE at the cost of increased load on the network and particularly in the bottleneck (*Hossfeld, 2007*). New communication paradigms such as *seamless communications* for automatic access network switching – cf. for instance (*Fiedler, 2005*) – and *autonomic networking* applying self-organisation principles – cf. for instance (*Binzenhöfer, 2006*) – typically rely upon both types of feedback. Implicit feedback is the standard fall-back solution in case the explicit feedback fails. In seamless communications, for instance, such a situation is given when a mobile unit suddenly runs out of coverage of the used wireless access network. In autonomic networking, failure of one entity may prevent the production of explicit feedback, which means that the partner entities have to catch such a situation through the interpretation of implicit feedback.

Research on quality feedback typically focuses on control loops within network equipment, e.g. (*Nguyen, 2004*), within servers, e.g. (*Wei, 2005*), and as part of the ICT design process (*Evia, 2004*). Some further references will be given in Section 3. So far, we are not aware of any systematic inventory of feedback flows between different stakeholders. (*Gelenbe, 2006*) provides similar underlying ideas on a conceptual level, however without explicit focus on feedback flows. A recently published operator’s vision on future Internet (*Guillemin, 2007*) describes the roles of different stakeholders, but not feedback relationships between them. It however claims better integration and interaction of services and networks and identifies the necessity of corresponding management. Our quality feedback can be seen as a part of this.

A list of wishes for quality feedback flows looks as follows: Clear organisational *responsibilities* need to exist, i.e. which entity is telling what; clear *communication* and *interpretation* of messages are mandatory; clear *timings* need to exist in order not to endanger user patience or the functioning of control loops. Further items include *scalability* such that the feedback solution can grow with the service and networks to be monitored; *modularity* for unproblematic integrations into existing systems; *light weight* in order not to challenge resource-constrained environments; and *backwards-compatibility* to existing network management solutions in order to enable a smooth transition. We will revisit this list in Section 4.

3. Concurrent Quality Feedback Flows

This section provides an inventory and classification of feedback flows in today’s networked systems. Besides of the contacts to other communication partners such as members of the same community, a user typically has relationships to Application Service Providers (ASP) and Network Service Providers (NSP), which includes Internet Service Providers (ISP). Internet Backbone Providers (IBP), on the other hand, are in general hidden to end users.

In the sequel, we assume a cut between application and network between OSI layers 5–7 and above (applications and application-oriented protocols) and OSI layers 4 and below (transport protocols towards physical layer), respectively. The notion “network” may comprise several IP networks belonging to different ISPs/IBPs. Figure 1 illustrates feedback flows that are discussed in the following. The abbreviations close to the arrows match the ones introduced in Sections 3.1 – 3.4, describing potential flows of feedback originated by network, application, user and ASP. In the sequel, we anticipate the reason of the feedback to be network problems. In order not to make the picture too complex, we will neither consider internal feedback loops within network elements, cf. (Nguyen, 2004), nor within instances of applications as part of their respective local control loops, e.g. (Wei, 2005).

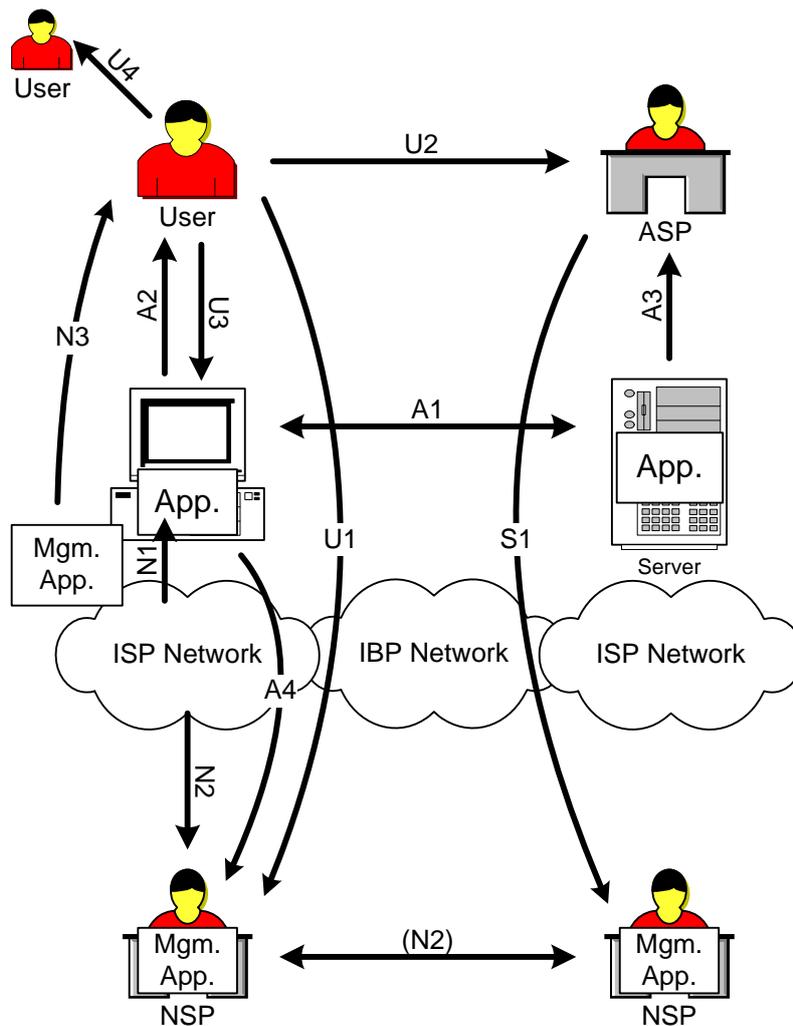


Figure 1. Concurrent quality feedback flows.

3.1 Feedback from the network

N1 – Network towards application: In the IP world, this type of feedback is mostly implicit. Networks make the application suffer from problems as packets get delayed or lost. Transport protocols hand over the data streams towards the application, either “as-is” including losses and delays in case of the User Datagram Protocol (UDP), or they transform losses into delays in case of TCP. In other words, no explicit feedback towards the application is provided. An approach for explicit notification of network bandwidth and backlog is provided in (*Raghuveer, 2004*). Also, the method described in (*Fiedler, 2003*) has been designed to communicate the degree of network transparency towards the application.

N2 – Network towards NSP: This feedback is obtained from explicit monitoring using the Simple Network Management Protocol (SNMP), which is actually a layer-7 protocol. Implicit feedback is used as fall-back solution: ping requests that remain without answer indicate connectivity problems. Monitoring traditionally applies rather long time scales when polling devices. If the aggregate load on a specific link exceeds a – mostly experience- or best-practice-based – threshold quite frequently, that particular link's capacity is upgraded, i.e. bandwidth is thrown onto the problem. Generally, a NSP takes care of the own network in the first hand. Exchange of management information on network level with other ISP or IBP is sparse. So, neither a single user's nor end-to-end perception are captured by standard network management.

N3 – Network towards user: The user feels network problems in an implicit way through the application and is in some cases informed explicitly e.g. through warnings or error messages issues by the operating system (such as “cable disconnected”). Rather rudimentary tools such as ping or traceroute are available in most operating systems, in some cases even a bandwidth monitor. However, the information presented is rather cryptic and needs expert knowledge to be interpreted. Amongst others, (*Fiedler, 2003*) proposed a visual performance indicator that visualizes the footprint of the network on the throughput perceived by a data stream.

3.2 Feedback from the application

A1 – Application towards application: Implicit feed-back is seen from non-optimal interaction of remote processes influenced by the interconnecting networks. As pointed out before, some applications measure the network impact explicitly in order to derive control actions. The application-level Real Time Control Protocol (RTCP) allows for sender and receiver reports, e.g. (*Sivalakrishnan, 2008*). Such reports might be evaluated e.g. by a videoconferencing application in order to adapt itself to the network conditions. This kind of application-level feedback is a particular strength of many Peer-to-Peer (P2P) applications such as Skype (<http://www.skype.com>).

A2 – Application towards user: The typical implicit feed-back is a consequence of N1 and A1. The user gets to feel the applications-perceived problems as far as the application is not able to compensate for problems originating from the network. Some applications provide explicit feedback, e.g. a web-browser after expiry of a time-out based on implicit feedback from the network, or based on the explicit feedback provided in A1. A plug-in for the browser Firefox is the Fasterfox utility (<http://fasterfox.mozdev.org>), displaying a download time counter in the lower-right corner of the

browser window. Icons such as “hourglass” and progress bar or warnings might be displayed, e.g. by the web browser or the video-conference application.

A3 – Application (server side) towards ASP: In order to receive explicit feedback of quality as perceived by the user, ASPs test the applications themselves by running test sessions (calls, TV, etc.) or requests (web sites, etc.). Thus, this type of feedback is the same as A2 despite of the addressee.

A4 – Application towards NSP: Also, the network provider might be interested in explicit feedback in how the application perceives the network. To this end, it might sniff for special packets containing application-level status information, such as RTCP reports. (*Simmons, 2004*) reports the use of such feedback for settings and policies for differentiated services.

3.3 Feedback from the user

U1 – User towards NSP: Typically but eventually, a user provides feedback only in case of problems and abnormalities, and “keeps quiet” if everything behaves as expected. However, user silence does not necessarily need to mean user peace of mind (*Nokia, 2005*). A typical explicit user reaction, however, consists in blaming the closest network provider (ISP) for any kind of trouble with the networked application that is experienced. Especially if the user's connectivity is affected, this reporting has to be done by other means of communication, e.g. by phone. However, in best-effort Internet, the situation can be quite complex. There may be several providers that have to be addressed and that are usually convinced that the problem is not to be found in their part of the network. Given their quite limited possibilities of monitoring (cf. N3), an average user might find it hard to find out about the real nature of a problem. Implicit feedback may consist of churn (*Nokia, 2005*).

U2 – User towards ASP: Some applications, web sites etc. welcome users to leave explicit comments about the performance. As in U1, users might choose to inform providers about problems rather implicitly – by giving up using a service – than by providing explicit feedback.

U3 – User towards application: Explicit feedback can be taken into account both in the design phase (*Hayashi, 2007*) and operational phase (*Bertini, 2004*) of the service. Some applications also allow for explicitly changing settings in order to cope with problems, e.g. by lowering the bit rate of a video conference in order to make the stream more robust to jitter. Again, the implicit way of dealing with the problems is to give up using the service or application, which amongst others has been studied in (*Chen, 2009*) for gaming.

U4 – User towards user: (*Nokia, 2005*) highlights the risk that users tell each other explicitly of problems instead of their NSP or ASP, thus spreading the bad experience to other customers and thus the risk of churn.

3.4 Feedback from the ASP

S1 – ASP towards NSP: Upon perception of quality problems (A2) or user complaints (U2), an ASP might contact the corresponding NSP in order to ensure the quality of the service's network connectivity according to the SLA.

4. Future Self-Organising Quality Feedback

Figure 1 showed a quite involved picture of feedback flows. Given the enormous multitude of applications and the increasing demands of users, and although increasingly many applications fix problems by themselves, this involvedness is not expected to decrease by itself.

The AutoMon project carried out within the European FP6 Network of Excellence EuroNGI developed a concept for a self-organising quality monitoring system capable of providing explicit feedback to all imaginable stakeholders (*Binzenhöfer, 2006*). Similar thoughts of applying feedback and self-organisation, pointing out the need of an “intelligent and sensible dialogue between users, services and the network” are provided by (*Gelenbe, 2006*). The latter proposes specific network entities in the form of intelligent network routers. In contrast, the AutoMon solution is a pure decentralised and scalable end-to-end software solution that applies P2P technology to self-organise the monitoring and feedback generation processes.

A key thought with the AutoMon infrastructure is to decentralise and systemise the feedback generation process in order to be able to better keep track of the individual performance perception. The user is relieved from the bothersome need to report in person, and the operator is relieved from the impracticable task to check each user’s end-to-end relationships by default. The enabler for this new functionality is a software entity called *Distributed Network Agent* (DNA) that is able to passively monitor and actively test communication facilities. The latter happens both locally and remotely through the cooperation with other DNAs. Inter-DNA communication allows for the exchange of test results and third-party observations. Any ASP and NSP can run a DNA itself and thus get first-hand information regarding the end-to-end perceived performance. The overlay spanned by DNAs is self-organising, which means near-to-zero configuration effort for any joining or leaving DNA. A detailed description of the DNA concept is found in (*Binzenhöfer, 2006*).

In the sequel, we will focus on the – exclusively explicit and future-proof – feedback flows enabled by the DNA, illustrated in Figure 2. According to their role in the management-and-feedback overlay, P stands for “peer”.

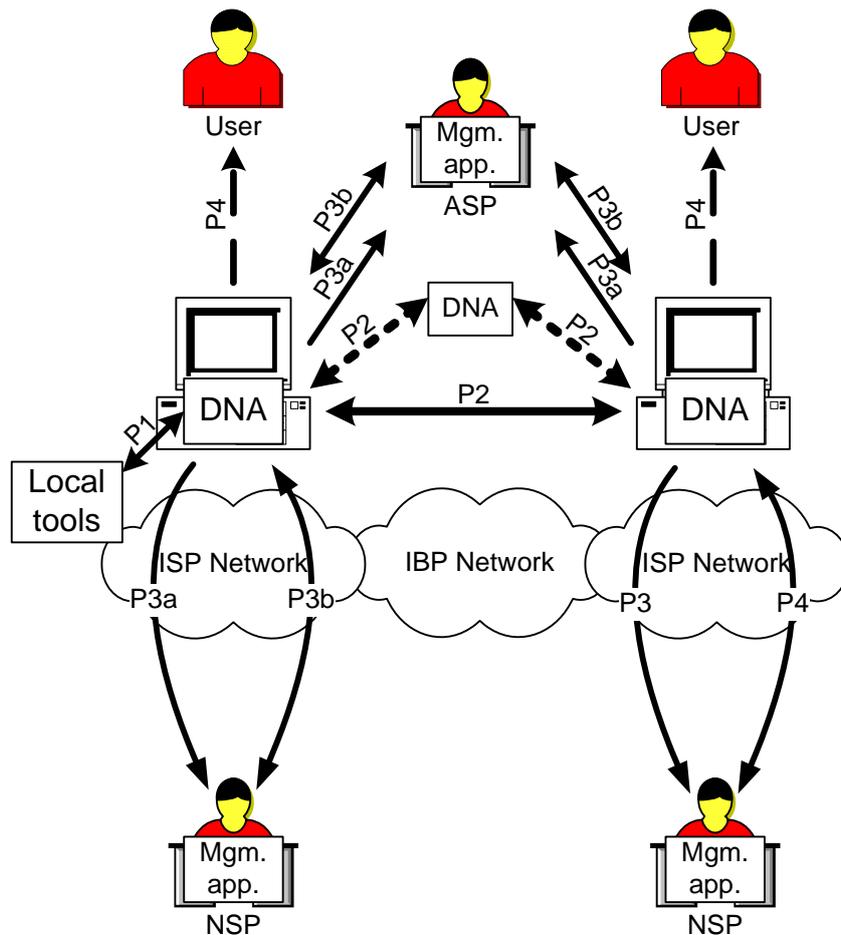


Figure 2. Future explicit quality feedback flows.

The quality feedback flows depicted in Figure 2 can co-exist with the ones shown in Figure 1, and have the potential to successively replace the majority of them.

P1 – DNA towards local tools: The DNA has the ability to carry out local tests using the existing tools provided by the operating system (cf. N3) on behalf of the user. Both the explicit and implicit feedback received from the tools is interpreted and if necessary transformed into explicit feedback towards other DNAs (P2), NSPs (P3), and users (P4). For instance, extraordinary long ping times or extraordinary high ping losses might trigger messages to these stakeholders.

P2 – DNA towards DNA: The DNAs apply inter-DNA communication for the self-management of their overlay, for the exchange of monitoring information – e.g. in order to implement comparative performance assessment methods as described in (Fiedler, 2003) – and for the initiation of remote tests. The availability of monitoring and test results all over the DNA overlay enables completely new ways of problem identification, classification and location. DNAs might gossip problems amongst themselves, and the fact that a DNA disappears due to a connection cut-off will not go unnoticed by the self-management mechanisms of the overlay. Other DNAs will be able to report this problem. As pointed out before, ASP and NSP might participate in the DNA overlay and thus integrate the additional monitoring capabilities into their existing network and service management systems. This way, they might get hold of problems before the user gets aware of them.

P3 – DNA towards NSP: In order to provide QoS feedback to legacy *Network Management Systems* (NMS), a QoS interface following common standards is required. To this aim, the Internet management protocol SNMP (Simple Network Management Protocol) used by virtually all network operators must be considered. In the SNMP organization model, the DNA assumes the role of an agent and provides two types of explicit feedback:

- **P3a: SNMP traps**, which are unsolicited alarm messages sent towards the NMS in case the DNA perceives a problem that has to be reported;
- **P3b: SNMP responses** as answers to SNMP requests issued by the NMS, either as reaction to the trap (P3a) or on a regular, NMS-scheduled basis. Access of the data is specified by a dedicated part of the enterprise-specific MIB (Management Information Base). This part is supposed to contain data that can be used by service and network operators in order to pinpoint problems.

Interfacing with existing NMS helps also to minimize costs for introducing new services and their DNA-based QoS management and allows for a smooth transition.

P4 – DNA towards user: Thinking about user being agnostic, this feedback is provided only if necessary, but then timely and to the point. ASP and NSP can take on user roles and thus observe the performance precisely in the same way as their users. Thus, they can also correlate this feedback with the observations from P2, which provides valuable knowledge about the dissemination of problems from the network towards the user.

Going back to the list of wishes presented at the end of Section 2, the presented self-organising DNA concept can implement the organisational responsibilities, communications, timings and backwards compatibilities as required, while maintaining scalability and modularity as provided by the overlay. This functionality comes at the price of effort to maintain the overlay. The latter is however quite small in volume (*Binzenhöfer, 2006*) and can thus be considered lightweight.

5. Conclusions and Outlook

This paper focused on quality feedback aspects and flows between different users, application and network service providers in case of network problems, which deteriorate user perception and in the long run even the relationship between particular users and operators. An inventory of existing feedback flows, originated by network, application, user and service provider, was presented. This inventory shows a rather confusing, uncoordinated composition of implicit and explicit feedback flows between the different stakeholders. Thinking of ever-growing possibilities, expectations and demands on networked applications from the users' side, a change has become necessary. Well-organised, timely, unambiguous and explicit feedback is needed for future flexible and dynamic service provisioning. The feedback-generating process needs to be scalable and modular in order to be able to grow with and into the systems of devices, services and networks to be supervised. An example for a self-organising feedback infrastructure, developed within a European Network of Excellence, which satisfies those criteria, was presented. The full potential of such a feedback infrastructure is yet to be discovered through large-scale implementations, demonstrations, and

trials with users and providers, respectively.

Acknowledgments

The author would like to thank the AutoMon team (Kurt Tutschku, Andreas Binzenhöfer, Patrik Arlos, Stefan Chevul, Mattias Schmid and Stefan Köhler) for the inspiring and fruitful cooperation, as well as the European FP 6/7 Networks of Excellence EuroNGI, EuroFGI and EuroNF for providing the framework and instruments for joint research on issues for networks of the future.

References

Bertini, M., Cucchiara, R., Del Bimbo, A., Prati, A. (2004). Content-based video adaptation with user's preferences. 2004 IEEE Int. Conf. on Multimedia and Expo (ICME'04), 3: 1695–1698.

Binzenhöfer, A., et al. (2006): DNA, a P2P-based framework for distributed network management. 2nd EuroNGI IA.8.2 Workshop, Lake Como, Italy, July 2005. Springer Verlag, LNCS 3883: Wireless Systems and Network Architectures in Next Generation Internet, 2006.

Bouch, A., Kuchinsky, A., Bhatti, N. (2000): Quality is in the eye of the beholder: Meeting user's requirements for Internet quality of service. Technical Report HPL-2000-4, HP Laboratories Palo Alto.

Chen, K.-T., Huang, P., Lei, C.-L. (2009). Effect of network quality on player departure behavior in online games. IEEE Trans. on Parallel and Distributed Systems, 20 (5): 593–606.

Evia, C. (2004): Quality over quantity: A two-step model for reinforcing user feedback in transnational web-based systems through participatory design interface. IEEE Trans. on Professional Communication, 47 (1): 71–74.

Fiedler, M., Tutschku, K., Carlsson, P., Nilsson, A. (2003): Identification of performance degradation in IP networks using throughput statistics. Proc. 18th International Teletraffic Congress (ITC-18), Berlin, Sept. 2003: 399–407.

Fiedler, M., et al. (2005): Generic communication requirements of ITS-related mobile services as basis for seamless communications. Proc. NGI 2005, Rome, April 2005.

Gelenbe, E. (2006): Users and service in intelligent networks. IEE Proc. on Intelligent Transport Systems, 153 (3): 213–220.

Guillemin, F. (ed) (2007): The future Internet : the operator's vision. EURESCOM Project Report P1657, Nov 2007.

Hayashi, T., Yamagishi, K., Tominaga, T., Takahashi, A. (2007). Multimedia quality integration function for videophone services. IEEE GLOBECOM '07, 2735–2739.

Hossfeld, T., Tran-Gia, P., Fiedler, M. (2007): Quantification of Quality of Experience for edge-based applications. Proc. 20th International Teletraffic Congress (ITC-20), Ottawa, June 2006.

ITU-T (2008), ITU-T Recommendation G.1080: Quality of experience requirements for IPTV services, Genève, 2008/12.

Nguyen, C., Hoang, D., Zhao, I., Lavian, T. (2004). Implementation of a quality of service feedback control loop on programmable routers. Proc. of 12th IEEE Int. Conf. on Networks, 2: 578–582.

Nielsen, A. (1994): Usability Engineering. Morgan Kaufman, San Francisco.

Nokia (2005): Nokia White Paper: Quality of Experience (QoE) of mobile services: Can it be measured and improved?

Raghuvver, A., Kusmierek, E., Du, D. (2004): Network-aware rate adaptation for video streaming. Proc. IEEE INFOCOM 2001, Vol. 2: 959–965.

Rajamony, R., Elnozahy, M. (2001): Measuring client-perceived response times on the www. 3rd USENIX Symposium on Internet Technologies and Systems (USITS), San Francisco, March 2001.

Simmons, B., Lutfiyya, H. (2004). Using application feedback in differentiated services and policies. IEEE/IFIP Network Operations and Management Symposium, 2004 (NOMS 2004): 45–58.

Sivabalakrishnan, M., Majula, D. (2008). Analysis of decision feedback using RTCP for multimedia streaming over 3G. Proc. IEEE INFOCOM 2008: 852–860.

Wei, J., Xu, C.-Z. (2005). Feedback control approaches for Quality of Service Guarantees in web servers. Annual meeting of the North American Fuzzy Information Processing Society, 2005 (NAFIPS 2005): 700–705.

Zona Research Inc. (1999): The economic impacts of unacceptable web-site download speeds.

Author

Markus Fiedler

received his Dr.-Ing. degree in Electrical Engineering with focus on ICT from Universität des Saarlandes, Saarbrücken, Germany, in 1998. Since then, he has been with Blekinge Institute of Technology (BTH), Karlskrona, Sweden, where he received the Docent degree in Telecommunication Systems in 2006. He is holding the position of an Associate Professor in the School of Computing. His main research interests are in Quality of Experience and its matching to network performance parameters, and seamless communications with focus on decision making, yielding “Always Best Connected”. He is coordinating and participating in national and international projects in those domains, for instance the Swedish KKS-funded project QoEMoVi, the European FP7 Network of Excellence EuroNF and the FP7 STREP PERIMETER, respectively.