Terminal–Side QoE Estimations for Cross–Layer Network Control

Martín Varela, Jukka–pekka Laulajainen

Terminal–side QoE Estimations for Cross–Layer Network Control

Towards a user–centric approach to network management

Martín Varela[1] and Jukka–Pekka Laulajainen[1] *

VTT Technical Research Centre of Finland
{martin.varela | jukka--pekka.laulajainen}@vtt.fi

Abstract. While the layered OSI model for networks provides a very clean conceptual framework for network design, certain applications are not completely indifferent to what happens at the lower layers. This is particularly true for multimedia and other real–time applications, which usually have stringent constraints on network performance in order to deliver an acceptable Quality of Experience (QoE) to the user.
Several cross–layer approaches for optimizing application quality have been proposed in the literature. In this paper we show how by combining network QoS measurements and application–level knowledge, significant improvements can be made to current management solutions.

1 Introduction

Ubiquitous Internet access is no longer a nice idea, but a fact. Everyday devices such as phones, tablets and laptops keep users connected practically everywhere, over a variety of network technologies, and with relatively low costs. As a consequence, networked services, such as social networks, Internet telephony, video sharing, etc. have become an important part of users' activities on the Internet, every day, at all times, and anywhere.

As users become used to having these services available, and depend on them, the issue of how to ensure appropriate service levels, and a good user experience becomes very important. This is especially the case for media services, which not only require the availability of network access, but also have constraints on network performance.

The Quality of Experience (QoE) for media services depends on many factors, but generally it will be mainly defined by both the application's requirements (e.g. video, telephony, music streaming, all have different requirements vis–à–vis the network) and characteristics, and the actual performance of the underlying network. Other factors, such as equipment characteristics, along with user expectations and environmentalfactors also play an important role in defining the QoE; however, those are usually fixed and it is not possible to do much in order to improve on them.

* This work was partially supported by the CELTIC Easy Wireless 2 and CELTIC IPNQSIS projects, Tekes and VTT Technical Research Centre of Finland.

In this work we focus on media services and their quality from the users' point of view. In particular, we explore how QoE estimations on the terminal side can be used to perform cross–layer control actions at the network level, in order to better cope with performance issues commonly encountered in IP networks.

The rest of the paper is organized as follows. Section 2 describes how accurate QoE estimations can be obtained single–sidedly on the terminal, in real–time. In Section 3 we describe possible applications of the QoE measurements for network control purposes, and in Section 4 we provide an example of cross–layer QoE–based control. Finally, we conclude the paper and present future research ideas in Section 5.

2 Measuring QoE

Measuring the quality as perceived by the users – or QoE – is difficult since there is a very strong subjective component implicit in its definition. There is a large and long standing body of work dealing with methods for defining and measuring the quality of media and media services [3, 27, 5], ranging from qualitative approaches, to quantitative ones, and yet other, psychometrics–based ones [9, 13]. The most commonly ones, however, consist of basically asking a panel of (usually *naïve*) test subjects to give ratings to media samples or live services under a very controlled environment (cf. ITU-T P.800 [20] for speech quality, ITU-R BT.500 [6] for video quality, ITU-T P.920 [23] for interactive services, etc.). These testing methods produce, in a way, a "true value" for quality, as they consider the subjective experience of a large enough panel of users[1]. They are, however, very expensive and cumbersome to implement, and hence other approaches to assessing quality have been developed.

In contrast to the aforementioned *subjective* methods, they are usually referred to *objective* or *instrumental* methods, and they do not rely on human subjects. These instrumental methods can be classified according to whether they are signal–based or parametric, or on whether they use a full–reference signal for comparison, a reduced reference, or no reference at all. For example, for speech applications, the most widely used instrumental metric is currently PESQ [21] (currently being superseded by P.OLQA [22]), which is a full–reference model. Full–reference models produce some of the most accurate assessments[2], but are by definition ill-suited for being used in any control mechanisms, since reference signals are not available in real–time for normal applications.

For control purposes, accurate, no–reference, single–sided models are needed in order to obtain QoE estimations in real–time. A metric such as Pseudo–Subjective Quality Assessment (PSQA) [17] lends itself well to this task. PSQA is a parametric methodology for estimating QoE, usually in the form of Mean

[1] Note that there is, within the QoE research community, a large push to move away from these types of tests towards more sophisticated ones. In practice, however, most subjective testing nowadays is carried out in this way.

[2] In terms of correlation with subjective assessment.

Opinion Score (MOS) values. It has been used successfully for VoIP quality assessment, both listening [25] and conversational [7], as well as video [15, 8].

2.1 About PSQA

PSQA works by mapping, via a statistical estimator, network– and application–level parameters known[3] to affect perceptual quality to actual subjective scores. Some examples of such are the encoding used, the error protection and concealment, the loss rate, delay and jitter in the network, etc.

For each parameter π a range of possible values is identified, according to the expected usage context. This range lies between a minimal value $\pi_{\min}$, and a maximal one, $\pi_{\max}$. Within this range, a set of representative values is chosen.

Let $\mathcal{P} = \{\pi_1, \cdots, \pi_P\}$ be the set of selected parameters and $\{p_{i1}, \cdots, p_{iH_i}\}$ the set of possible values chosen for parameter π_i, with $p_{i1} < p_{i2} < \cdots < p_{iH_i}$. A *configuration* of the set of quality-affecting parameters, is any set of possible values for each one, i.e. a configuration $\boldsymbol{\gamma}$ is a vector $\boldsymbol{\gamma} = (\gamma_1, \cdots, \gamma_p)$, where $\gamma_i \in \{p_{i1}, \cdots, p_{iH_i}\}$

For even a moderate number of parameters, the total number of possible configurations (that is, $\prod_{i=1}^{P} H_i$) is usually very large, and subjective testing is not feasible over the whole parameter space. Therefore, a small subset of this space is selected to be used as (part of) the input data for the chosen statistical estimator used in the learning phase. This is usually a 3–layer feed–forward Random Neural Network (RNN) [10], although it could, in principle, be any other suitable machine learning tool.

In order to generate a media database composed of samples corresponding to different configurations of the selected parameters, a simulation environment or a testbed must be implemented[4]. This is used to transmit media samples (or perform live tests, as needed) from the source to the destination and to control the underlying network. Every configuration in the defined input data must be mapped into the system composed of the network, the source and the receiver. The destination stores the transmitted media sample and associates it with the corresponding configuration.

Thus, for each media sample σ, and set of S configurations, a set $\mathcal{T} = \{\sigma_1, \cdots, \sigma_S\}$ of degraded samples is constructed. These samples have encountered varied transmission conditions. That is, the clip σ_s corresponds to the result of sending σ through the source-network system where the selected P parameters have the values of configuration γ_s.

Once the material (and/or testbed) is ready, a subjective assessment campaign is conducted. The MOS values obtained from the testing campaign are then mapped to each sample (and thus to the corresponding configuration); the value corresponding to sequence σ_s is denoted here by ψ_s.

Finally, a suitable RNN is trained and validated. The trained RNN can be seen as a function f having P real variables, such that

[3] Or at least strongly suspected.

[4] In the case of interactive assessment, a suitable testbed must be implemented.

(i) for any sample $\sigma_s \in \mathcal{T}$, $f(v_{1s}, \cdots, v_{P,s}) \approx \psi_s$,
(ii) and such that for *any other* configuration with values $(v_1, \cdots, v_P)$, $f(v_1, \cdots, v_P)$ is close to the MOS that would be associated in a subjective evaluation with a media sample for which the selected parameters had those specific values $v_1, \cdots, v_P$.

As with any statistical learning tool, the performance of the resulting estimator is evaluated on validation data, which is not used during training. The most relevant metric for this performance is the correlation of the MOS estimations produced by the PSQA implementation and subjective scores. PSQA can achieve very high correlations for both voice (listening and conversational) and video applications.

Once the RNN is trained, calculating the MOS estimations is computationally trivial, which makes PSQA a good choice for use in resource–constrained devices, where signal–based estimations, such as the one described in ITU-T recommendation P.563 [19] might be too costly in both processing time and energy consumption.

3 Possible Applications

When talking about network management and control mechanisms, it is commonly understood that these are issues dealt with by network operators, within the network itself. In fact, access control mechanisms, traffic differentiation mechanisms, resource provisioning, etc. are mostly implemented on the network side, and the terminal (and hence the user or the application) has little chance of having any impact on them.

There are, however, some things that can be best done on the terminal side, especially when it comes to end–to-end QoE, since at least one service end–point resides on the terminal itself. One such application is mobility management. Be it based on Mobile IP [11], IMS Service Continuity [1] (or VCC in older networks), Media independent handover (MIH) [2], or Host Identity Protocol (HIP) [16], the terminal is usually in the best position to determine when a handover is needed.

Another application is simultaneous multi–streaming of SVC streams. In this context, a multi–access terminal could choose to have different layers of the scalable video stream sent over different networks, depending on the available networks and their QoS.

Packet marking for traffic differentiation, as proposed in [18] for intra–stream marking, in [26] for LAN–level priority management for VoIP, or in [14] for QoS management in wireless home networks, is also a good candidate application for performing control on the terminal end.

Moving beyond the terminal itself, it is possible to exploit the QoE measurements when making network–side control decisions, if suitable signaling from the terminal to the network is implemented. Similarly, terminal side QoE estimations could be used to adapt the operation of the application in use. For this purpose, signaling QoE information between the terminal and applications should be provided.

4 Case Study: Improving Mobility Management via QoE Estimations

In this section we present a case study on an application of QoE estimations to network–level control mechanisms. We have developed a prototype smart mobility management tool which significantly enhances the performance of Mobile IP (MIP) systems, by using PSQA–based QoE estimations for VoIP to signal when a handover is needed.

4.1 Mobile IP in a Nutshell

Mobile IP [11] allows users to attach to networks in different administrative domains while still being reachable through an address in their own network (called *home network*). Incoming traffic towards the home address of the mobile node is forwarded to a care–of address, to which the node is currently bound.

As the node roams through different networks, it creates mobility bindings, and these are used by a so–called *home agent* to reach the mobile node at any given time. If the mobile node has multi–homing capabilities, it is possible for it to seamlessly roam through different network technologies (for example, from Ethernet to WiFi to 3G).

Typically, in this multi-homing scenario, MIP solutions implement a priority–based policy mechanism for choosing an access network. The priorities are usually based on expectations of cost and performance of the different access networks, but do not usually consider actual performance of the network (beyond being able to reach the home agent), nor application–level performance. This leads to situations in which a mobile node will remain attached to a network with sub–par performance, even though other networks are available and might provide better quality levels.

4.2 Prototype Implementation and Experimental Setup

The proposed solution prototype improves on this situation by monitoring the QoE of the application being used (in this case VoIP, but it can trivially be extended to video or other applications where PSQA or a similar mechanism might be used), and making handover decisions when the quality is degraded below a configurable threshold.

Several strategies are available to make the handover decisions, depending on the desired "dynamism" of the solution. The most stable strategy considers a sliding weighted average of the MOS estimations over a configurable time window. The weighted average can be replaced with a simple, non-weighted average if a more "reactive" behaviour is desired. It is also possible to consider only the last estimation before deciding whether to perform a handover, although this strategy is likely to result in unwarranted handovers due to minor network impairments.

The prototype also allows for roaming back into the previous network after a configurable period, which is useful in the case of temporary impairments.

The current implementation of the prototype uses VTT's QoSMeT tool [24] to perform passive measurements of the network performance, and feed the obtained values to a PSQA implementation and the decision logic. The decision logic then interacts with Birdstep's[5] MIP solution to perform the actual handovers. In this way, the MIP solution is augmented with a user–centric view of network performance, namely, forcing a handover when the application quality (QoE) drops below acceptable levels.

The prototype has been tested in a variety of scenarios and it has yielded very promising results. In this paper, a simple, yet representative scenario for network congestion is presented. A mobile node with Ethernet and WiFi connections, is attached, via its Ethernet connection, to a Linux router running NetEm [12]. NetEm is then used to emulate the effects of network congestion on the link by introducing packet losses. We compare the behaviour of the plain MIP solution, and that of the QoE–based smart mobility management proposed.

4.3 Performance Improvements

Figure 1 shows the degradation of listening quality for a VoIP stream using GSM and media–dependent FEC (as described in [4]). Since the impairment is not too severe, the mobility manager does not lose contact with the home agent, and no handover is performed, despite the call quality dropping below usable levels.

The improvements made by augmenting the handover decision–making with QoE estimations is visible in Figures 2 and 3. In these figures, the prototype was configured to consider a weighted average of the MOS estimations over a $10s$ sliding window. In Figure 2 the handover is triggered about $6s$ after the congestion period starts, and in another $3s$ the quality has once again reached acceptable levels.

In the case of Figure 3, the option to roam back to the previous network after a period of about $35s$ was enabled, and a second handover can be observed at about the $84s$ mark. In the same figure, some losses in the WiFi connection can also be observed (at $72s$), but as the impairment was very small and of short duration, no action was taken.

5 Conclusions and Future Work

In this paper we have discussed possible applications of QoE estimations on the terminal side to achieve some degree of user–centric network control. While there are limits to the type of network–level control that can be implemented by the terminal itself, several interesting possibilities for improving the QoE of

[5] Formerly Secgo Mobile IP. `http://www.birdstep.com/Products/`

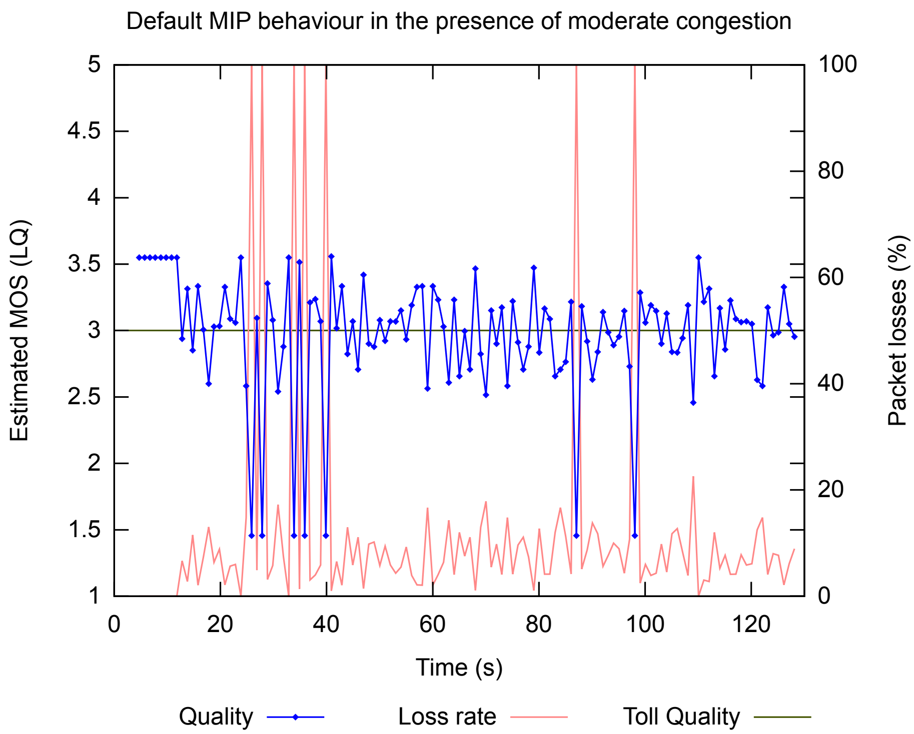

Fig. 1. When the QoE is not considered for handover decisions, the listening quality can significantly degrade in the presence of temporary network congestion.

media services can be explored. We substantiated our claims by means of a case study in which a Mobile IP–based system was significantly improved by taking a cross–layer approach and considering application–level QoE estimations when deciding whether to perform a handover.

Further work on this area will include other applications of control on the terminal side, as discussed in Section 3, as well as network–side control mechanisms (such as access control, traffic marking and shaping, etc.). This last topic is one of the main targets of the ongoing CELTIC IPNQSIS project.

References

1. 3GPP TSG SA WG2. 3GPP TS 23.237: IP Multimedia Subsystem (IMS) Service Continuity; Stage 2, 2008.
2. IEEE 802.21. Std 802.21-2008 – IEEE Standard for Local and Metropolitan Area Networks – Part 21: Media Independent Handover Services, January 2009.
3. J. Allnatt. Subjective rating and apparent magnitude. *International Journal of Man-Machine Studies*, 7(6):801–816, November 1975.

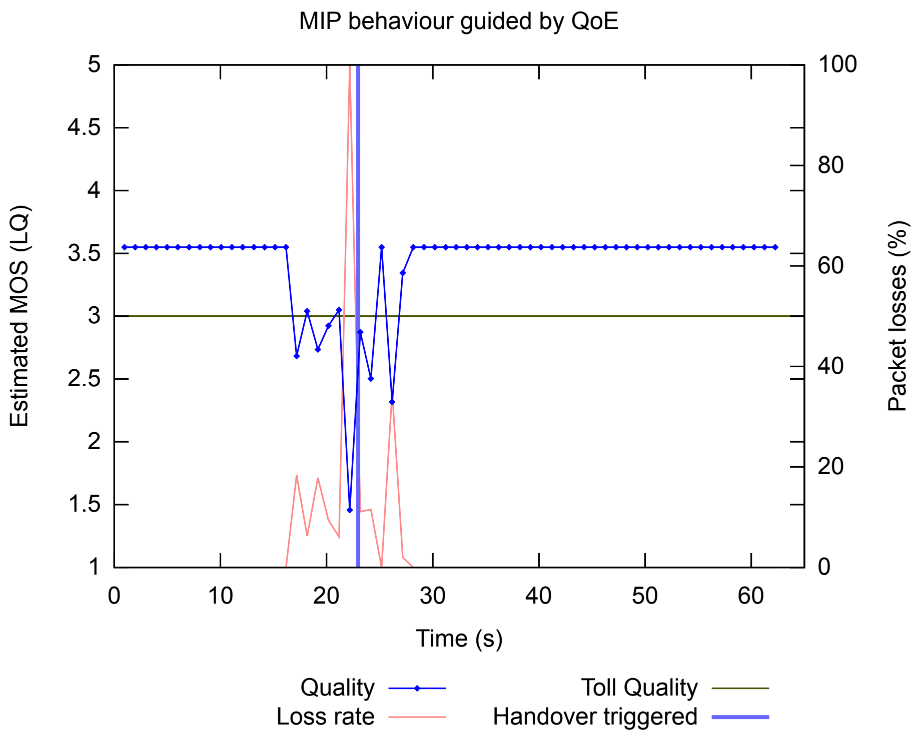

Fig. 2. Taking QoE into account when making the handover decision results in a fast recovery from a degradation in listening quality.

4. Jean Bolot and Andrés Vega Garcia. The case for FEC-based error control for packet audio in the internet. In *ACM Multimedia Systems*, 1996.
5. A. Bouch, M. A. Sasse, and H. DeMeer. Of packets and people: A user-centered approach to quality of service. *Proceedings of IWQoS2000*, page 189197, 2000.
6. ITU-R Recommendation BT.500-10. Methodology for the subjective assessment of the quality of television pictures, March 2000.
7. Ana Couto da Silva, Martín Varela, Edmundo de Souza e Silva, Rosa Leão, and Gerardo Rubino. Quality assessment of interactive real time voice applications. *Computer Networks*, 52:11791192, April 2008.
8. Ana Couto da Silva, Pablo Rodriguez-Bocca, and Gerardo Rubino. Optimal Quality-of-Experience design for a P2P Multi-Source video streaming. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 22 –26, May 2008.
9. Virginie Durin and Laetitia Gros. Measuring speech quality impact on tasks performance. In *INTERSPEECH*, pages 2074–2077. ISCA, 2008.
10. Erol Gelenbe and Jean–Michel Fourneau. Random neural networks with multiple classes of signals. *Neural Computation*, 11(3):953–963, 1999.
11. IETF Network Working Group. IP Mobility Support for IPv4 (RFC 3344), August 2002.

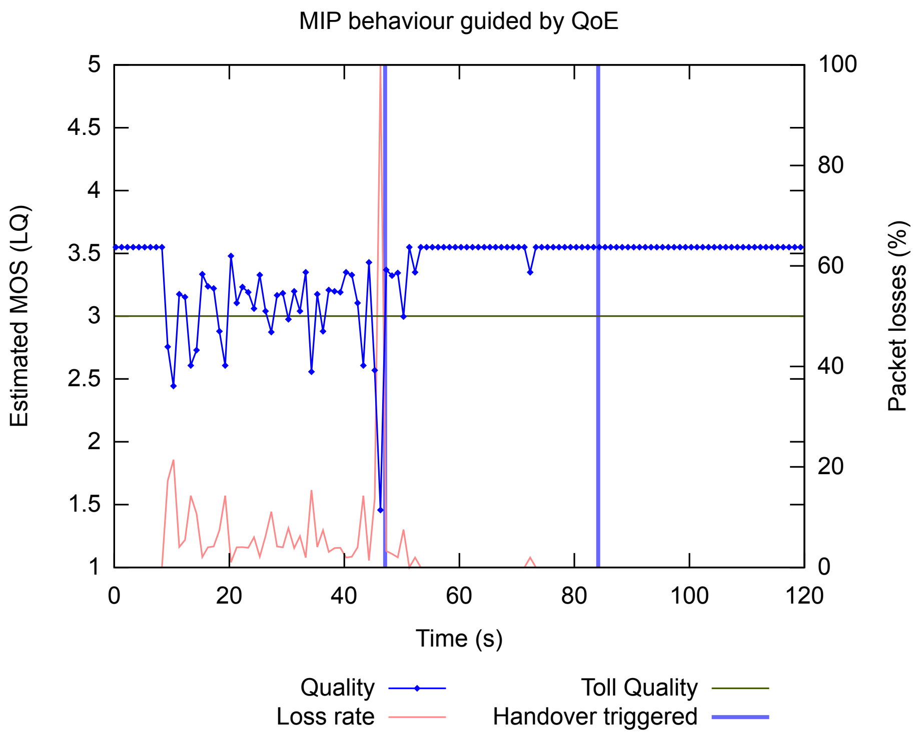

Fig. 3. It is possible to automatically roam back into the previous network after a period of time. This is useful in cases where occasional short congestion periods are common.

12. Stephen Hemminger. NetEm website. `http://www.linuxfoundation.org/collaborate/workgroups/networking/netem`.
13. Satu Jumisko-Pyykkö and Miska M. Hannuksela. Does context matter in quality evaluation of mobile television? In G. Henri ter Hofte, Ingrid Mulder, and Boris E. R. de Ruyter, editors, *Mobile HCI*, ACM International Conference Proceeding Series, pages 63–72. ACM, 2008.
14. Jukka-Pekka Laulajainen. Implementing qos support in a wireless home network. In *WCNC*, pages 3273–3278. IEEE, 2008.
15. Samir Mohamed and Gerardo Rubino. A study of realtime packet video quality using random neural networks. *IEEE Transactions On Circuits and Systems for Video Technology*, 12(12):1071 –1083, December 2002.
16. R. Moskowitz, P. Nikander, Jokela, and T. Henderson. RFC 5201 - Host Identity Protocol, April 2008.
17. Martín Varela. *Pseudo-Subjective Quality Assessment of Multimedia Streams and its Applications in Control*. PhD thesis, INRIA/IRISA, univ. Rennes I, Rennes, France, November 2005.
18. Julio Orozco and David Ros. DiffServ-Aware streaming of h.264 video. In *in Proceedings of the 14th International Packet Video Workshop (PV2004*, 2004.

19. ITU-T Recommendation P.563. Single ended method for objective speech quality assessment in Narrow-Band telephony applications, 2004.
20. ITU-T Recommendation P.800. Methods for subjective determination of transmission quality, 1996.
21. ITU-T Recommendation P.862. Perceptual evaluation of speech quality (PESQ), an objective method for End-To-End speech quality assessment of narrowband telephone networks and speech codecs, 2001.
22. ITU-T Recommendation P.863. Perceptual objective listening quality assessment, 2011.
23. ITU-T Recommendation P.920. Interactive test methods for audiovisual communications, 2000.
24. Jarmo Prokkola, Pekka H. J. Perälä, Mikko Hanski, and Esa Piri. 3G/HSPA Performance in Live Networks from the End User Perspective. In *ICC*, pages 1–6, 2009.
25. Gerardo Rubino, Martín Varela, and Samir Mohamed. Performance evaluation of real-time speech through a packet network: a random neural networks-based approach. *Performance Evaluation*, 57(2):141–162, May 2004.
26. Gerardo Rubino, Martín Varela, and Jean-Marie Bonnin. Controlling multimedia qos in the future home network using the psqa metric. *Computer Journal*, 49(2):137–155, 2006.
27. A. Watson and M. A Sasse. Evaluating audio and video quality in Low-Cost multimedia conferencing systems. *ACM Interacting with Computers Journal*, 8(3):255–275, 1996.