

Intelligent Synthesis Mechanism for Deriving Streaming Priorities of Multimedia Content

Gheorghita Ghinea, *Member, IEEE*, George D. Magoulas, *Member, IEEE*, and Christos Siamitros

Abstract—We address the problem of integrating user preferences with network Quality of Service parameters for the streaming of media content, and suggest protocol stack configurations that satisfy user and technical requirements to the best available degree. Our approach is able to handle inconsistencies between user and networking considerations, formulating the problem of construction of tailor-made protocols as a prioritization problem, solvable using fuzzy programming.

Index Terms—Communication protocols, fuzzy prioritization methods, multimedia communications, quality of service, subjective multimedia quality.

I. INTRODUCTION

STREAMING of multimedia content needs to incorporate capabilities for responding to changes originating from the user/applications or the system/network. To effectively respond to these changes, future networked multimedia systems will require fast renegotiation protocols and adaptive mechanisms, capable of actively modifying the configuration of a communication system so that at any one time the mechanisms used to transfer data are the most appropriate for the connection over which communication is taking place.

Indeed, integrating user-level expectations with parameters characterizing underlying network performance is a problem seldom studied in multimedia streaming, for it attempts to bridge the gap existing between user perceptions of multimedia quality, on the one hand, and the *Quality of Service* (QoS) with which multimedia is transmitted over the network, on the other. Work in this respect has focused on the effects that different video frame rates have on human satisfaction with the multimedia presentation [1], [20], [21] on the perceptual impact of errors [17], delay [2] and jitter [9], or, alternatively, on the development of metrics for assessing subjective multimedia quality based on models of the human visual system [16], [19].

However, only rarely is such research carried forward in the development of adaptive streaming applications. Accordingly, the QUASAR project [10] exploits human perceptual tolerance to media losses and frame dropping, as do [8] and [11], respectively. While [16] builds upon the previously formulated

distortion metric, [18] uses identified subjective acceptance of occasional media errors to develop a streaming mechanism in which such errors are distributed without negative perceptual impact. However, such approaches generally fall short on two counts: firstly, the perceptual tests on which they are based ignore multimedia's *infotainment* duality (namely that all multimedia applications are located somewhere on the information-entertainment continuum); secondly, many of them assume that users of distributed multimedia technology have considerable technical skills which, given the proliferation of the Web, represents the exception rather than the norm today.

In our work, we address both issues. Thus, our approach to user-perceived QoS encompasses not only the traditional view of a user's satisfaction with the presentation quality of a multimedia application, but also his/her ability to understand, analyze and synthesize their informational content—Quality of Perception (QoP). Accordingly we evaluated QoP through a series of empirical tests, whose results [5] indicated that QoS must also be specified in terms of perception, understanding and absorption of content if multimedia presentations are to be truly effective from both a user as well as technical perspective. Based on these results, we propose a scheme for intelligent multimedia streaming in which users do not necessarily need to specify their preferences. We formulate the problem in terms of Multicriteria Decision Making [3], [22], and propose a solution in terms of fuzzy programming [22] to resolve inconsistencies between user and network considerations and derive streaming priorities of multimedia content geared toward ensuring an optimum user experience.

II. INTELLIGENT SYNTHESIS OF COMMUNICATION MECHANISMS

Our approach uses the framework of the *Dynamically Reconfigurable Stacks Project* (DRoPS), which provides an infrastructure for the implementation and operation of multiple adaptable protocols [4]. DRoPS-based communication protocols are composed of fundamental mechanisms, called microprotocols, which perform arbitrary protocol processing operations. The complexity of processing performed by a microprotocol is not defined by DRoPS and may range from a simple protocol function, such as a checksum, to a complex layer of a protocol stack, such as TCP. In addition, protocol mechanisms encapsulated within a microprotocol may be implemented in hardware or software. If appropriate hardware is available, the microprotocol merely acts as a wrapper, calling the relevant hardware function.

Microprotocols are encapsulated in loadable modules, allowing code to be dynamically loaded into a running operating

Manuscript received June 11, 2003; revised September 27, 2003. This work was supported in part by the EPSRC, U.K. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Benoit Macq.

G. Ghinea and C. Siamitros are with the School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, Middlesex UB8 3PH, U.K. (e-mail: george.ghinea@brunel.ac.uk; christos.siamitros@brunel.ac.uk).

G. D. Magoulas is with the School of Computer Science and Information Systems, Birkbeck College, University of London, London WC1E 7HX, U.K. (e-mail: g.magoulas@dc.s.bbk.ac.uk).

Digital Object Identifier 10.1109/TMM.2005.858391

TABLE I
ADAPTABLE FUNCTIONALITY IN DRoPS

Protocol mechanism	Implementations
Sequence control	none (<i>micro1</i>) complete (<i>micro2</i>)
Flow control	none (<i>micro3</i>) window based (<i>micro4</i>)
Acknowledgement scheme	IRQ (<i>micro5</i>) PM-ARQ (<i>micro6</i>)
Checksums	none (<i>micro7</i>) block checking (<i>micro8</i>) full CRC (<i>micro9</i>)

system and executed without recompiling a new kernel. Each such microprotocol can be implemented via a number of adaptable functions; in particular, microprotocols may also represent the absence of a particular function, such as *micro1*, as shown in Table I.

A. The Proposed Solution: Multicriteria Decision Making

There are practically no universal elementary measurable properties in terms of which user perception can be defined, nor is there a measurable relationship relating measurable properties of multimedia presentations to a user's perception. Nevertheless, our evaluation of QoP revealed that multimedia perceptual quality varies with the number of media flows, the type of medium and application, as well as the context-dependent relative importance of each medium [5]. Thus, each multimedia application can be characterized by the relative importance of the Audio (*A*), Video (*V*), and Textual (*T*) components as conveyors of information, as well as the Dynamism (*D*) of the presentation. On the other hand, five commonly considered network level QoS parameters have been incorporated in our model: Bit Error (*BER*), Segment Loss (*SL*), Segment Order (*SO*), Delay (*DEL*), and Jitter (*JIT*).

However, as our experiments highlighted [5], the end-user may not clearly specify a desired parameter value and may prefer the use of linguistic phrases to describe their priorities, e.g., “*Video is slightly more/less important than Audio*”. Although information about such questions is vital in making correct design decisions, it is very difficult to quantify the fuzziness in users' perception and in the meaning of their words. Our approach uses a formulation of the problem in terms of multicriteria decision making (MDM) [3], [13], [22]:

Let a set of *K* alternative microprotocols {*micro1*, *micro2*, *micro3*, ..., *microK*} and a set of *P* decision criteria {*BER*, *SO*, *SL*, *DEL*, *JIT*, *A*, *V*, *T*, *D*} that relate to user and network considerations be given. Then the problem is to rank the microprotocols in terms of their total preferences when all the decision criteria are considered simultaneously, and ultimately find a protocol stack configuration that satisfies all user and technical requirements to the best available degree.

There are two steps in solving this problem.

- 1) Attach numerical measures to the relative importance of the criteria and to the impacts of the alternatives on these criteria. This consists of two subprocedures: i) determine the relative importance of the criteria and ii) determine the relative standing of each alternative with respect to each criterion.

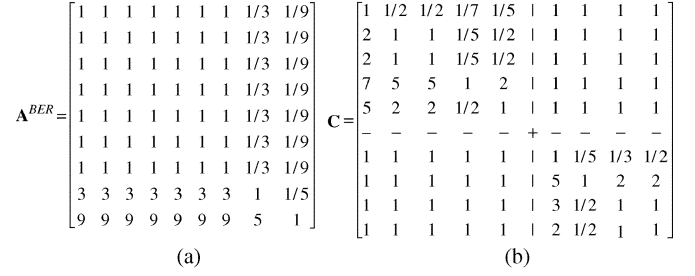


Fig. 1. (a) Matrix describing the relative importance of microprotocols with respect to BER. (b) Matrix describing the relative importance of QoS and perceptual parameters.

- 2) Process the numerical values to determine a ranking priority of each alternative.

Step 1 is explained in this subsection, whilst Step 2 will be explained in the next. Step 1 involves the definition of a number of reciprocal judgement matrices. A judgement matrix A^p is an $(K \times K)$ matrix in which element a_{ij} indicates the preference of alternative A_i over alternative A_j in terms of the decision criterion C_p (for $i, j = 1, 2, 3, \dots, K$; $p = 1, 2, \dots, P$). It is also assumed that the decision maker has determined a $(P \times P)$ matrix C which details the pairwise relative importance between the criteria themselves. The rationale for determining the preferences and the relative importance has its origins in psychology, where a number of experiments have shown that individuals cannot simultaneously compare more than seven objects (± 2) [15]. Usually, pairwise comparisons are quantified by using a scale of nine grades, e.g. $a_{ij} \in \{1, 2, 3, \dots, 8, 9\}$, and the relation $a_{ji} = 1/a_{ij}$ also holds [12]. In our user experiments [5], all pairwise relations were encoded and communicated to the users using the following conventions: 1 \equiv “equally important”; 2 \equiv “slightly more important”; 1/2 \equiv “slightly less important”; 3 \equiv “weakly more important”; 1/3 \equiv “weakly less important”; 4 \equiv “moderately more important”; 1/4 \equiv “moderately less important”; 5 \equiv “strongly more important”; 1/5 \equiv “strongly less important”; 6 \equiv “essentially important”; 1/6 \equiv “essentially less important”; 7 \equiv “demonstrably important”; 1/7 \equiv “demonstrably less important”; 8 \equiv “highly important”; 1/8 \equiv “highly less important”; 9 \equiv “absolutely important”; 1/9 \equiv “absolutely less important”. Although we found users appreciated this equivalence between linguistic and numerical terms, numerical representation shall be used henceforth for simplicity.

Accordingly, in our case we have a total of ten matrices. An example of a judgement matrix A^p , where $p = BER$ is shown in Fig. 1(a). Rows and columns of A^{BER} correspond to the various microprotocols, while each entry a_{ij} ($i, j = 1, 2, 3, \dots, 8, 9$) of the matrix represents a designer-defined numerical judgement giving the relative suitability of the nine microprotocols to fulfil a specified requirement, e.g., a weighting of $a_{ij} = 1$ for $i, j = 1, 2$ can be interpreted as “*microprotocol 1 is equally important as microprotocol 2 with respect to BER*”. Analogously, the strongest of these methods, the full CRC (*micro9*), has the highest weight (a value of $a_{9j} = 9$; $j = 1, 2, \dots, 7$) in comparison with all the others, while a relatively weak block checking algorithm (*micro8*) is considered to be *weakly more important* ($a_{8j} = 3$,

$j = 1, 2, \dots, 7)$ than microprotocols from other functionality classes.

The matrix **C** of each criterion with respect to all the other criteria, shown in Fig. 1(b), is the only one whose values may fluctuate as a result of changes in the operating environment, as well as a consequence of changes in *user preferences* and *perceptions*. Fig. 1(b) provides an instance of this matrix used in our model; the respective criteria (corresponding to rows and columns of the matrix) are, in order, *BER*, *SO*, *SL*, *DEL*, *IT*, *A*, *V*, *T*, and *D*. **C** integrates user-centric and technical requirements and could conceptually be split-up into four submatrices.

- A 5×5 matrix, in the upper left part of matrix **C**, giving the relative importance of the *BER*, *SO*, *SL*, *DEL*, and *JIT* criteria with respect to one another. This matrix is allowed to change dynamically during the transmission of a multimedia clip. For example, an intelligent controller may, as a result of a delay-intolerant audio application being subjected to high network delays, change the numerical judgements of the submatrix to reflect a more radical bias in favor of the delay component.
- A 4×4 matrix, located in the bottom right of matrix **C**. Here user input can reflect personal choices of the relative importance of the video, audio and textual components in the context of the application, as well as a relative characterization of the dynamism of the multimedia clip. A priori values in this case could also reflect the result of user-consultations, as indeed is our case [5].
- A 5×4 and a 4×5 matrix which reflect designer choices of the relative importance of the five QoS parameters considered on *A*, *V*, *T*, and *D*. The elements of these matrices reflect the results of our previous work and define the balance between the relative importance of user-centric/subjective and QoS parameters [5].

B. Deriving Streaming Priorities With the Fuzzy Programming Method

Streaming priorities are derived as a result of solving the decision making problem defined in the previous subsection based on the data available in the various judgement matrices. This derivation of priorities includes: i) determining *weighted priorities* w_i ($i = 1, \dots, 9$) for the criteria and (ii) determining *relative weights* $w_{j,i}$, which denote how preferable is microprotocol j with respect to criterion i . Weighted priorities and relative weights are then synthesized to yield the *overall ranking priority* of each microprotocol, and thus determine their inclusion/exclusion in the protocol stack.

In a real-world context, it is not uncommon for there to be inconsistencies between technical and perceptual information represented in the judgement matrices. This leads to matrix entries that are just estimations of ideal judgements. To alleviate this situation, the fuzzy programming method (FPM), capable of solving even highly inconsistent matrices, can be used [12]–[14]. This formulates the prioritization problem as a maximin fuzzy programming problem, which finds a crisp priority vector, maximizing the overall decision-maker's satisfaction with the final solution. Along this line, FPM enables judgements to be expressed either as crisp, intervals or fuzzy numbers. In order to deal with the uncertainties in the estimation of the judgements and avoid computational overheads

$$\mathbf{A} = \begin{bmatrix} 1 & [l_{12}, u_{12}] & [l_{13}, u_{13}] & \dots & [l_{1K}, u_{1K}] \\ [l_{21}, u_{21}] & 1 & [l_{23}, u_{23}] & \dots & [l_{2K}, u_{2K}] \\ [l_{31}, u_{31}] & [l_{32}, u_{32}] & 1 & \dots & [l_{3K}, u_{3K}] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ [l_{K1}, u_{K1}] & [l_{K2}, u_{K2}] & [l_{K3}, u_{K3}] & \dots & 1 \end{bmatrix} \quad (\text{a})$$

$$\mathbf{R} = \begin{bmatrix} 1 & -u_{12} & 0 & 0 & \dots & 0 \\ -1 & l_{12} & 0 & 0 & \dots & 0 \\ 1 & 0 & -u_{13} & 0 & \dots & 0 \\ -1 & 0 & l_{13} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & 0 & \dots & -u_{1p} \\ -1 & 0 & 0 & 0 & \dots & l_{1p} \\ 0 & 1 & -u_{23} & 0 & \dots & 0 \\ 0 & -1 & l_{23} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -u_{p-1,p} \\ 0 & 0 & 0 & \dots & -1 & l_{p-1,p} \end{bmatrix} \quad (\text{b})$$

Fig. 2. (a) General form of an interval comparison matrix. (b) General form of a constraints matrix.

we describe the information with interval judgments. This can be done by introducing tolerance parameters d_{ij} that will consider each judgment as an interval with lower and upper bounds $a_{ij} = (l_{ij}, u_{ij})$ [13]. Thus, a comparison matrix can be reformulated [Fig. 2(a)].

The idea behind the FPM is a geometrical representation of the prioritization process as an intersection of hyperlines. The FPM determines the values of the priorities, corresponding to the common intersection point of all hyperlines by seeking priority vectors that satisfy the corresponding inequalities

$$l_{ij} \leq \frac{w_i}{w_j} \leq u_{ij}, \quad i = 1, 2, \dots, K-1, \quad j = 2, 3, \dots, K, \quad j > i. \quad (1)$$

In case of inconsistent matrices, the hyperlines have no common intersection point, i.e., the intersection set is empty. Thus, the FPM represents the hyperlines as fuzzy lines and finds the solution of the approximate priority assessment problem, as an intersection point of these fuzzy lines with values for the priorities that satisfy all judgements “as well as possible”. Previous work gives evidence that FPM is able to produce better results than other methods when the degree of inconsistency is high [14]; this is a valuable property in our case. By following [13], the problem can be formulated as the standard linear programming problem of (2), where the objective is to maximize λ , a measure of intersection region of the fuzzy lines, subject to a set of $m = p(p-1)$ constraints, given in matrix form, $R \in \mathbb{R}^{m \times p}$, in Fig. 2(b)

$$\begin{aligned} \max \quad & \lambda \text{ subject to } \lambda d_k + R_k w \leq d_k, \\ & k = 1, \dots, m, \quad 1 \geq \lambda \geq 0, \\ \text{and} \quad & \sum_{i=1}^p w_i = 1, \quad w_i > 0, \quad i = 1, \dots, p, \end{aligned} \quad (2)$$

TABLE II
RELATIVE SCORES OF THE ALTERNATIVES COMPARISON MATRICES WITH RESPECT TO EACH CRITERION

	micro1	micro2	micro 3	micro 4	micro 5	micro 6	micro 7	micro 8	micro 9
BER	0.0574	0.0574	0.0574	0.0574	0.0574	0.0574	0.0574	0.1541	0.4441
SO	0.1082	0.2754	0.0787	0.0787	0.1049	0.1180	0.0787	0.0787	0.0787
SL	0.0881	0.0881	0.0881	0.0881	0.1872	0.2753	0.0617	0.0617	0.0617
DEL	0.2280	0.0638	0.2280	0.0638	0.0517	0.0456	0.2280	0.0456	0.0456
JIT	0.1614	0.1958	0.0771	0.3335	0.0474	0.0316	0.0743	0.0474	0.0316
V	0.0842	0.2254	0.0842	0.1883	0.0979	0.0818	0.0462	0.0694	0.1226
A	0.2175	0.0658	0.2175	0.0658	0.0590	0.0568	0.2529	0.0354	0.0293
T	0.1761	0.0833	0.1761	0.0833	0.1250	0.0833	0.1761	0.0625	0.0341
D	0.0843	0.0826	0.0826	0.3710	0.0759	0.0759	0.0759	0.0759	0.0759

In (2), p is the number of elements compared; $w = (w_1, w_2, \dots, w_p)^T$ is the vector of priority weights; k indicates the k th row of matrix R ; and the values of the tolerance parameters d_k represent the admissible interval of approximate satisfaction of the crisp inequalities $R_k w < 0$. For the practical implementation of the FPM, it is reasonable for all the d_k to be set equal [14]. The optimal solution to the problem is a vector (w^*, λ^*) , whose first component maximizes the degree of membership of the fuzzy feasible region set, whilst the second gives the value of the maximum degree of satisfaction. The method is explained in [13].

After deriving the underlying weights from the comparison matrices through the FPM technique, the weighted priority, w_i , and the relative weight, $w_{j,i}$, are synthesized using weighted sum aggregation in order to find the preference of microprotocol j with respect to all criteria/requirements simultaneously. Preference is denoted by W_j and determines the overall ranking priority, or weight, of microprotocol j (obviously, the microprotocol with the maximum overall value W_j will be chosen)

$$W_j = \sum_{i=1}^P w_i \cdot w_{j,i}. \quad (3)$$

III. EXPERIMENTS AND RESULTS

We use four scenarios illustrating the ability of our approach to select appropriate microprotocols and construct a suitably-tailored protocol stack depending on the prevailing network environment and user perceptual preferences. All the nine alternative matrices were evaluated and solved by applying the FPM. The relative scores thus obtained are presented in Table II, where for example, the first four microprotocols considered have an equal importance with respect to managing SL. However, the most important microprotocol for segment loss is micro6, which has the highest relative score.

A. Scenario 1: General Case

In this example, no assumptions are made about the underlying network conditions, or about the multimedia content to be transported over the network. As such, the *a priori* judgement values which arise from technical considerations, as well as user judgements resulting out of our experiments [5] are used, and are given in Fig. 1(b). As can be observed from the table,

TABLE III
PRIORITY WEIGHTS, w_i , WITH RESPECT TO THE CRITERIA ($i = 1, \dots, 9$)

	General case	Visually-impaired/ High delay
BER	0.0858	0.0426
SO	0.0951	0.0481
SL	0.0951	0.0481
DEL	0.1573	0.0862
JIT	0.1105	0.0585
V	0.0936	0.0554
A	0.1495	0.5142
T	0.1066	0.0735
D	0.1066	0.0735

delay and audio are the most important criteria from a technical and user point of view, respectively. This is because distributed multimedia applications have an essential real-time characteristic, which makes delay the primordial factor from the technical point of view. On the other hand, our work on perceptual aspects of multimedia has confirmed previous experiments in highlighting that the most important medium in a multimedia presentation, from a user perspective, is the audio component.

Furthermore, from a technical angle, the SL criterion has the same importance as the SO criterion, and thus in Fig. 1(b), these two criteria are shown to be “*equally important*”. Similarly, for the user, the Text (T) criterion is as important as Dynamism (D). Moreover, as can be observed, QoS and perceptual parameters are considered to be equal to unity, which reflects a balance between perceptual and QoS considerations in this initial scenario.

By applying the FPM, the weights w_i are derived (see Table III). Finally, by synthesizing the relative scores (see Table II) and the priority vector, using (3), the overall priorities (scores) of the alternatives are obtained (see Fig. 3).

Micro1 is suggested as the best alternative, as it has a high relative score for both delay (the most important parameter from a technical/QoS point of view) as well as audio (the most important parameter from a user point of view). Although micro7 is even better suited to manage these two parameters, if one considers the overall set of parameters, it is micro1 that achieves at least equal or higher relative scores than micro7 for eight out of the nine parameters/criteria (see Table II).

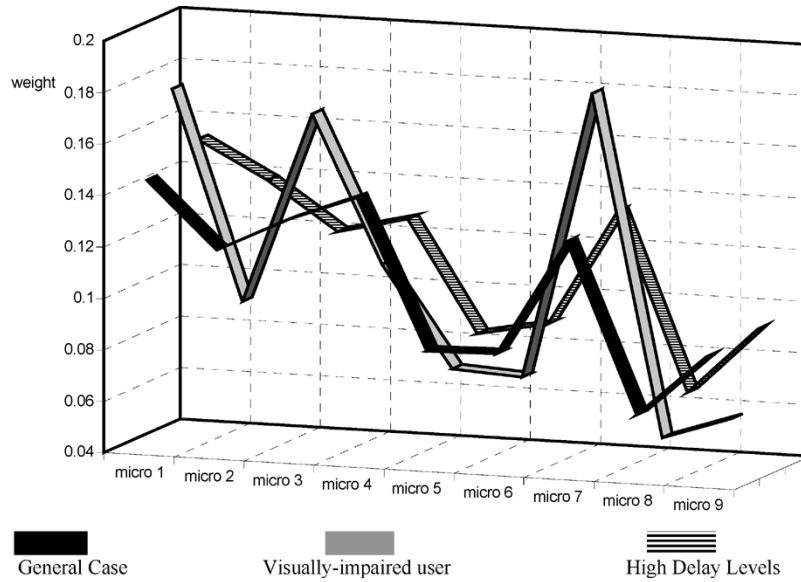


Fig. 3. Microprotocol priority weights for three cases.

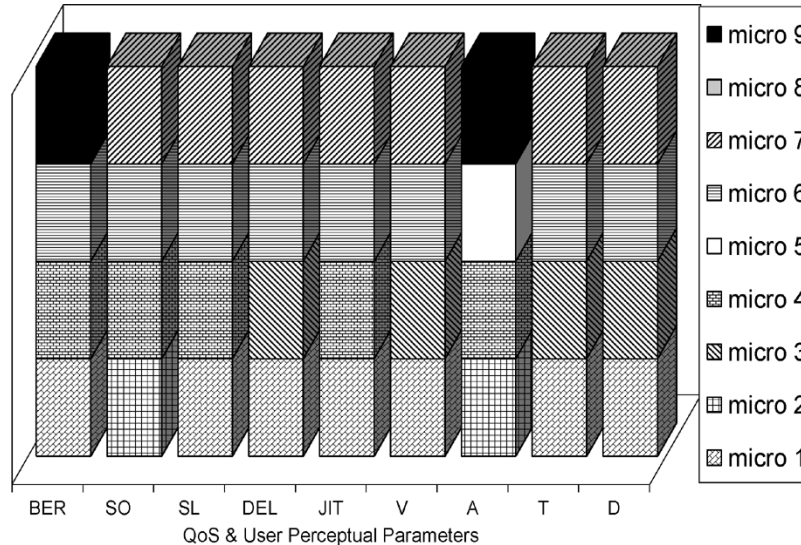


Fig. 4. Resulting DRoPs protocol stack when QoS and perceptual parameters are, in turn, “demonstrably important”.

B. Scenario 2: Accessibility Requirements and Network Delays

In this scenario, we treat cases whereby the QoP and QoS parameters considered in our model are, in turn, “demonstrably important” with respect to all other parameters. For instance, it seems logical for a visually impaired user to specify that the video and textual components of the multimedia presentation are not of paramount importance, whereas audio becomes “demonstrably important”. Similar considerations apply to a situation where high delays are encountered on a network. Thus, the audio- and delay-related entries respectively, of the criteria with respect to the criteria matrix reflect these requirements by taking values in the set $\{7, 1/7\}$. In these cases there is no impact on the evaluation of the relative scores (see Table II), which depends on the nine constant alternatives with respect to each criterion judgment matrices. However, the two situations do affect the evaluation of the priority weights w_i . The new priority weights vector are shown in Table III.

Finally, the overall priority values are calculated by applying (3), and are shown in Fig. 3. Thus, in the case of audio being “demonstrably important”, the most important microprotocols with respect to audio are micro7, micro1, micro3 (in this order), with overall priority values 0.1843, 0.1766, and 0.1701, respectively, which, indeed, are the ones that preferentially target the audio.

In Fig. 4, we show the resulting protocol stack constructed using the DRoPS framework, when each of the perceptual and QoS parameters becomes, in turn, “demonstrably important” with respect to all the others. Thus, in the case where SL is “demonstrably important” then, as Fig. 4 depicts, the DRoPS protocol stack is made up of micro1, micro4, micro6 and micro7. Whilst the choice of micro6 is to be expected, as it is the only microprotocol in the DRoPS framework explicitly able to handle losses, the choice of micro4 highlights the importance of flow control for segment losses, which would prevent, for instance, buffer overflows. The choice of micro1 and micro7

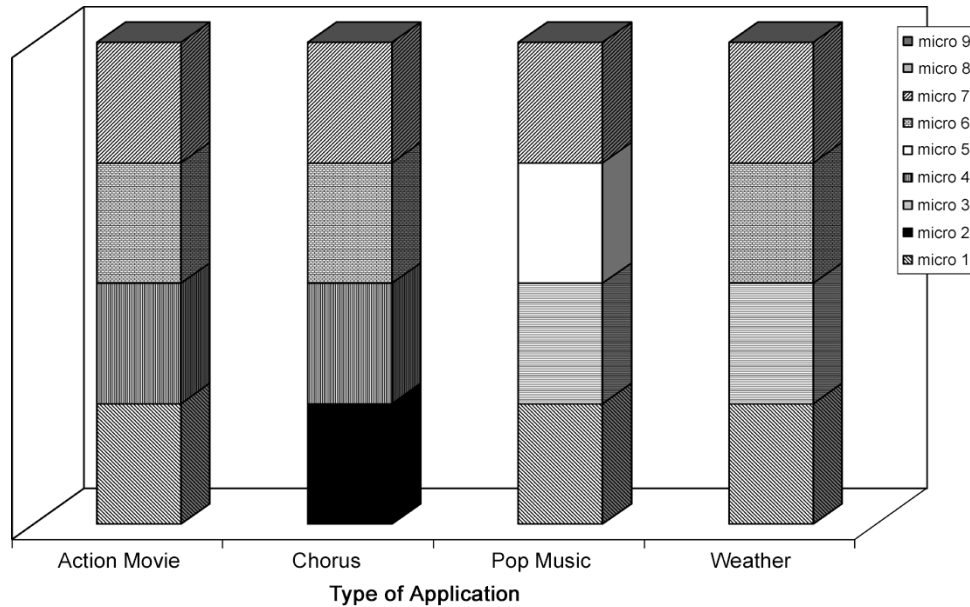


Fig. 5. Content-dependent DRoPS protocol stack.

reflects the streamlined functionality of the protocol stack, as these microprotocols, by not acting on sequence control and bit errors, respectively, reduce computational overhead.

Similar observations apply when perceptual parameters are “demonstrably important”. Accordingly, with the exception of audio, all media components of multimedia presentations are tolerant to bit errors. Thus, the case when audio is considered “demonstrably important” is the only one in which the resulting protocol stack includes in its configuration micro9, the most suited microprotocol to handle bit errors. The fact that most distributed multimedia applications have real-time constraints as well as being tolerant to bit errors, is reflected in the choice of the “no-frills” micro7 in all other cases, for this type of functionality. The delay-intolerant nature of distributed multimedia applications is also reflected in the choices of micro1 and micro3 in the suggested protocol stacks when video and text are “demonstrably important”. The choice of micro6 for these two scenarios reflects, however, the importance of not losing segments of information, particularly in the case of compressed media, where any information loss would propagate through subsequent media units.

C. Scenario 3: Content-Dependent Adaptation

In this example, users were presented with a series of 4 MPEG video clips (Action Movie, Chorus, Pop Music and Weather Forecast), representing a variety of subject matter, and were asked to rate these clips on a scale of 1–7 according to the four user/perceptual criteria listed above. The average results thus obtained for each clip were categorised into “Low”, “Medium” and “High”. Afterwards, this scale was mapped to the set {1, 3, 9}. Thus, in the case of the Chorus clip if the user judged the importance of the audio (A) component of a clip as High(= 9) and the video (V) component as Medium(= 3), then, in the pairwise comparison matrix, the preference of audio against video will be defined as the ratio of $9/3 = 3$, i.e. weakly more important.

Referring to Fig. 1(b), the only part that suffers modifications is the lower-right user/perceptual submatrix. Of course, during clip transmission, the user, can, if desired, change these *a priori* characterizations to ones that suit his/her taste.

After obtaining the priority vectors and the overall priority values of each microprotocol, the DRoPS protocol stack is constructed by choosing, for each of the four main functionalities provided by the DRoPS framework, the most appropriate microprotocol. Thus, our method results in application-specific multimedia communication protocol stacks and Fig. 5 depicts the resulting stack after applying our approach to the four different multimedia video clips.

IV. CONCLUSIONS

This paper has proposed a fuzzy prioritization approach to protocol management. Our approach has distinct advantages over previous work: users themselves do not necessarily need to specify preferences; it is more comprehensive than preceding attempts which have usually focused on the viewing enjoyment of multimedia clips, ignoring the infotainment duality of multimedia; by employing fuzzy priority assessment, it is able to handle inconsistencies and uncertain judgements.

We recognize, however, that issues such as scalability have yet to be addressed—whilst our proposed method is flexible enough to be used with existing schemes for controlling quality of service parameters, further investigations are needed in order to provide large-scale personalized multimedia streaming. The question of building and testing the efficiency of the proposed scheme in practice is also top of our priorities and preliminary results in this respect have already been reported [7] in the context of a medical application.

ACKNOWLEDGMENT

G. D. Magoulas wished to acknowledge support from the EPSRC, U.K.

REFERENCES

- [1] R. T. Aptecker, J. A. Fisher, V. S. Kisimov, and H. Neishlos, "Video acceptability and frame rate," *IEEE Multimedia*, vol. 2, no. 3, pp. 32–40, Autumn/Fall 1995.
- [2] A. Bouch, A. Kuchinsky, and N. Bhatti, "Quality is in the eye of the beholder," in *Proc. CHI 2000*, The Hague, Holland, 2000, pp. 297–304.
- [3] S. J. Chen and C. L. Hwang, "Fuzzy multiple attribute decision making: methods and applications," in *Lecture Notes in Economics and Mathematical Systems*. Berlin, Germany: Springer-Verlag, 1992, p. 375.
- [4] R. S. Fish and R. J. Loader, "A kernel based adaptable protocol architecture," in *Proc. 24th Euromicro Conf.*, Västerås, Sweden, 1998, pp. 1029–1036.
- [5] G. Ghinea and J. P. Thomas, "Quality of perception: User quality of service in multimedia presentations," *IEEE Trans. Multimedia*, to be published.
- [6] G. Ghinea, J. P. Thomas, and R. S. Fish, "Quality of perception to quality of service mapping using a dynamically reconfigurable communication system," in *Proc. Globecom '99*, Rio, Brazil, pp. 2061–2065.
- [7] G. Ghinea, G. D. Magoulas, and A. O. Frank, "Intelligent protocol adaptation in an enhanced medical e-collaboration environment," *Int. J. Artif. Intell. Tools*, vol. 13, no. 1, pp. 199–218, 2004.
- [8] C. K. Hess and R. H. Campbell, "Media streaming protocol: an adaptive protocol for the delivery of audio and video over the internet," in *Proc. IEEE ICMCS'99*, Florence, Italy, 1999, pp. 903–907.
- [9] K. Hikichi, H. Morino, S. Matsumoto, Y. Yasuda, I. Arimoto, M. Ijume, and K. Sezaki, "Architecture of haptics communication system for adaptation to network environments," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Tokyo, Japan, 2001, pp. 744–747.
- [10] C. Krasic, J. Walpole, and W. Feng, "Quality-adaptive media streaming by priority drop," in *Proc. NOSSDAV'03*, Monterey, CA, 2003, pp. 307–310.
- [11] A. Mahanti, D. L. Eager, M. K. Vernon, and D. J. Sundaram-Stukel, "Scalable on-demand media streaming with packet loss recovery," *IEEE/ACM Trans. Netw.*, vol. 11, no. 2, pp. 195–209, Apr. 2003.
- [12] L. Mikhailov, "Deriving priorities from fuzzy pairwise comparison judgements," *Fuzzy Sets and Syst.*, vol. 134, no. 3, pp. 365–385, 2003.
- [13] —, "A fuzzy programming method for deriving priorities in the analytic hierarchy process," *J. Oper. Res. Soc.*, vol. 51, pp. 341–349, 2000.
- [14] L. Mikhailov and M. G. Singh, "Comparison analysis of methods for deriving priorities in the analytic hierarchy process," in *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, Tokyo, Japan, 1999, pp. 1037–1042.
- [15] G. A. Miller, "The magical number seven plus or minus two: some limits on our capacity for processing information," *Psycholog. Rev.*, vol. 63, pp. 81–97, 1956.
- [16] Y. Sermadevi, M. A. Masry, and S. S. Hemami, "MINMAX rate control with a perceived distortion metric," in *Proc. SPIE—Visual Communications and Image Processing*, San Jose, CA, 2004.
- [17] S. Varadarajan, H. Q. Ngo, and J. Srivastava, "An adaptive, perception-driven error spreading scheme in continuous media streaming," in *Proc. ICDCS 2000*, Taipei, Taiwan, 2000, pp. 475–483.
- [18] —, "Error spreading: A perception-driven approach to handling error in continuous media streaming," *IEEE/ACM Trans. Netw.*, vol. 10, no. 1, pp. 139–152, Feb. 2002.
- [19] O. Verscheure, P. Frossard, and M. Hamdi, "User-oriented QoS analysis in MPEG-2 video delivery," *Real-Time Imaging*, vol. 5, pp. 305–314, 1999.
- [20] T. Yamazaki, "Subjective video assessment for adaptive quality of service control," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Tokyo, Japan, 2001, pp. 517–520.
- [21] G. Yadavalli, M. A. Masry, and S. S. Hemami, "Frame rate preferences in low bit rate video," in *Proc. IEEE ICIP 2003*, Barcelona, Spain, 2003, pp. 441–444.
- [22] H.-J. Zimmermann, *Fuzzy Set Theory and Its Applications*, 2nd ed. Boston, MA: Kluwer, 1991.



Gheorghita Ghinea (M'02) received the B.Sc. and B.Sc. (Hons.) degrees in computer science and mathematics in 1993 and 1994, respectively, and the M.Sc. degree in computer science in 1996, all from the University of the Witwatersrand, Johannesburg, South Africa. He received the Ph.D. degree in computer science from the University of Reading, Reading, U.K., in 2000.

He is a Lecturer in the School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, U.K. His research interests span per-

petual aspects of multimedia, quality of service, and multimedia resource allocation, as well as computer networking and security issues.



George D. Magoulas (M'02) received the B.Eng., M.Eng., and Ph.D. degrees from the University of Patras, Patras, Greece, in electrical and computer engineering.

He is a Reader in Computer Science in the School of Computer Science and Information Systems, Birkbeck College, University of London, London, U.K. He previously held Lecturer and Senior Lecturer posts at Brunel University, Uxbridge, U.K. His research is focused on learning and evolution algorithms with applications to intelligent adaptive

systems.

Dr. Magoulas is a member of User Modeling Inc., the Technical Chamber of Greece, and the Hellenic Artificial Intelligence Society.

Christos Siamitros received the M.Sc. degree in computation from UMIST, Manchester, U.K., and the Ph.D. degree from the School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, U.K., in 2004.

His research interests are in the areas of modeling for discrete optimization problems and in fuzzy multicriteria decision making.