연속적인 미디어를 위한 네트워크 적응형 전송 및 네트워킹 지원 설계 이슈들

정회원 김 종 원*

Design Issues in Network Adaptive Delivery and its Networking Support for Continuous Media

Jong-Won Kim* *Regular Member*

ABSTRACT

Delivering rich and continuous media contents robustly over a wide range of network conditions of the wired/wireless Internet is a highly challenging task. To address this challenges, the continuous media applications at the edge of network has become more and more adaptive while the best-effort Internet is slowly progressing towards improved networking services. That is, the role of network adaptive media delivery, which dynamically links the quality demand of application contents to the underlying networking services, has become more crucial. In this paper, we will first review the required network adaptation functionalities seen from the application side: congestion control / rate control, error control, and synchronization / adaptive playout. Then, we start the coverage of networking support issues that helps the realization of network adaptive media streaming - from network support and protocol support toward consolidated support via middleware. Finally, we propose a dynamic network adaptation framework that efficiently leverages its awareness of both media application (including contents) and underlying networking support.

요 약

다양한 미디어 콘텐츠를 최선형 서비스에 머무르고 있는 유무선 인터넷의 네트워크 상태에 무관하게 전달하기 위해서는 네트워크 가장자리에서 동작하는 미디어 응용프로그램들이 보다 적응화되어야 한다. 즉 응용프로그램에서 전달하고자 하는 미디어 콘텐츠가 요구하는 네트워크 품질에 대한 요구와 기반 네트워크 서비스를 적응적으로 연결하는 것이 매우 중요하다. 본 논문에서는 먼저 응용프로그램에게 요구되는 혼잡제어 및 전송율 제어, 오류 제어, 그리고 동기화 및 적응형 재생 등과 같은 네트워크 적응화 기능들에 대해 의논한다. 이어서 네트워크 적응적 미디어 스트리밍을 실현하는 기반이 되는 요소들을 물리적인 네트워크, 프로토콜 지원에서 미들웨어에 걸치는 총체적인 지원의 구도하에서 설명한다. 최종적으로는 상기한 미디어 응용프로그램과 네트워크 지원 기반을 동시에 이해하면서 실현되는 동적인 네트워크 적응화의 구조를 제안한다.

Key Words: Network adaptive media streaming, network adaptation, and networking support.

* 광주과학기술원 정보통신공학과 Networked Media 연구실 (jongwon@kjist.ac.kr)
논문번호 : 030035-0122, 접수일자 : 2003년 1월 22일
※ 본 연구는 정보통신부 정보통신연구진흥원에서 지원하고 있는 정보통신기초연구지원사업과 광주과학기술원 기관고유사업의 지원에 의해 수행되었습니다.

1. Introduction

With the enormous growth of the Internet, more and more applications are distributing media contents to worldwide users. The average size of media contents transferred over the Internet are exponentially increasing everyday. It can be partly supported by the up-coming broadband network infrastructure. However, to provide *broadband, rich, and continuous media* contents reliably and scalably, efficient utilization and proper manage-ment of limited networking resources become increasingly important. Realization of high-quality media delivery over the IP Internet faces lots of challenges. Media applications in general have very strict requirements on the networking service, thus making he current best-effort Internet model less than sufficient. They require harmonization of stable networks and systems, feasible signaling/transport protocols, and network-adaptive applications to achieve acceptable-quality media distribution. From the network side, upcoming QoS (quality of service) networks can alleviates several complications of current best-effort Internet. That is, networks are slowing evolving toward improved QoS services to guarantee loss, delay, and bandwidth. Enhanced systems are also emerging to date to better support reliable and scalable media delivery. From the protocol side, we need suit of protocols to provide inter-operable/monitored transport channel over IP network. For example, application-layer protocol pair, RTP/RTCP (Realtime Transport Protocol / RTP Control Protocol), helps the real-time transport of media[1]. Thus, successful media delivery over the Internet requires the coordinated networking support from the networks, the protocols, the systems, and the applications.

The key for the successful delivery of broadband, rich media, however, is still with the applications at the edge. Recent years the media applications have become more and more adaptive to address the limitation of best-effort Internet [1-3]. Media applications at the server and client systems are required to response to the dynamic fluctuation of underlying networks. Either in a proactive or in a reactive manner, they are c ontrolling sending rate in response to congestion control, applying different error controls, and adjusting end-to-end latency for synchronization. With this network adaptation, they are satisfying the requirement of both application and involved systems in face of diverse network fluctuations and heterogeneities.

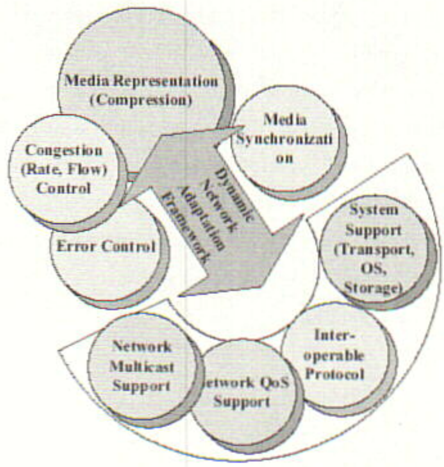


Figure 1: Involved issues for network-adaptive media delivery and their relationship.

For the network-adaptive delivery of media, we can categorize involved issues and visualize their relationship as shown in Fig. 1. That is, the required network adaptation needs networking supports from network, protocol, and system. Networking support for media delivery includes network issues such as multicast and QoS. Inter-operable protocols, especially application-layer protocols, are also very important pieces. Systems are also part of

[1]Unless constrained by the firewall (where the use of TCP is enforced), media delivery applications adopt transport options based on UDP (usually together with RTP).

media delivery chain and they handle the transport with operating system and storage. Security support for media delivery and content identification / discovery support are required, too. If possible, more unified support under the name of middleware is preferred way to advance in the future. With the networking support - infrastructure - in place, an effective media delivery application should select appropriate solution components and integrate them seamlessly. They should harmonize solution components to achieve the end-to-end performance while maintaining feasibility, scalability, and others.

As discussed above, the role of network adaptation is to link the quality demand of media application to the underlying networking services. Successful network adaptation needs to leverage the awareness of both media contents and underlying networking support. For this, well-established prioritization (or layering in coarse adaptation case) can play an important role for the efficient network adaptation. That is, via the prioritized media contents, coordination between different priority packets and networking service levels can be established efficiently. We propose to adopt layered relative priority index (layered-RPI) in order to accommodate diverse needs for prioritization in different granularity. For example, packetized media stream can be prioritized in session (or flow), layer (or frame), and packet levels. With the layered-RPI, network-adaptive media delivery is proposed. The proposed framework includes following components: 1) relative prioritization of media contents at the sender (i.e., based on the so called distortion and corruption model), 2) network adaptation tools to match the fluctuation of the underlying networks, and 3) forward/backward interaction mechanisms assisting the dynamic network adaptation.

In Section 2, we review the required network adaptation functionalities seen from the application side. Then, networking support issues are covered in Section 3 ranging from network support, protocol support, and consolidated support via middleware. A dynamic network adaptation framework is discussed in Section 4, where two kinds of deployment cases are reviewed.

2. Required Network Adaptation Functionalities

Delivering rich media over the Internet is nevertheless a challenging problem since the best-effort Internet does not guarantee the sufficient QoS (i.e., in terms of available bandwidth, delay limit, and loss percentage). That drives the needs for adaptive applications at the edges of network that response to the dynamic fluctuation of underlying networks. While keeping the friendliness with the majority TCP traffics by adopting the end-to-end congestion control, they are adjusting the sending rate, applying different error controls, and maintaining the synchronized, timely rendering of media contents. In the followings, we discuss the required network adaptation functionalities in the order of TCP-friendly congestion control and rate control, proactive and reactive error controls, and synchronization with adaptive playout.

2.1 TCP-Friendly Congestion Control / Rate Control

Bulky losses and delay variations are mainly caused by the insufficient available bandwidth over the path of network traffics. At the onset of network congestion, end system is required to reduce bandwidth usage (by releasing packets at slower pace) for its traffic to the level that is thought as an available bandwidth between the corresponding systems over the traffic route. Thus, the deployment of end-to-end congestion control is indispensable to help the Internet from catastrophic failure. Naturally, the dominant TCP traffics are congestion controlled and it has contributed to the robustness of the current Internet for more than two decades. Thus, for non-TCP traffics that lacks standardized congestion control and thus generally overwhelm TCP traffics, we need to deploy congestion control mechanism that is friendly to the TCP traffics, i.e., *TCP-friendly congestion control* [4].

The candidate solutions are being actively sought recently and they are well surveyed in [5]. The representative idea is to explicitly estimate the corresponding bandwidth of TCP's steady-state throughput which would be measured over the same route as

$$T = \min\left[\frac{W\cdot s}{R},\ \frac{s}{R\sqrt{\frac{2bp}{3}} + t_{RTO}\cdot\min\left[1,3\sqrt{\frac{3bp}{8}}\right]p(1+32p^2)}\right] \quad (1)$$

This Eq. (1) from [4] gives TCP's throughput T in steady-state as a function of the number of acknowledged packets b, the maximum size of the congestion window W, round-trip time R, retransmission timeout value t_{RTO}, packet size s, and steady-state loss probability p. Based on the estimation of TCP-compatible throughput, the sender adjusts its transmission rate to the estimation following the TCP's AIMD (additive increase multiplicative decrease) behavior to achieve TCP-friendliness. However, the saw-tooth (caused by AIMD) behavior of the resulting transmission rate hurts the media application that usually demands consistent pumping of media streams. Thus, as done in TFRC [6] and SFRAM [7], the end-to-end congestion control should try to be media-friendly as well as TCP-friendly. In addition, the congestion manager (CM) of IETF (Internet Engineering Task Force) addresses standardizing the congestion control for non-TCP traffics and unifying it with that of TCP traffics [8]. The CM is an end-system module that enables an ensemble of multiple concurrent streams from a sender destined to the same receiver and sharing the same congestion properties to perform proper congestion avoidance and control.

Typically congestion control dictates the available bandwidth and the sender responses by reducing the sending rate. If the required rate reduction is temporary (i.e., short-term fluctuation), the flow may be smoothed by the network de-jittering buffers. However, longer and larger variations should be met by controlling the sending rate (i.e., by adapting the quality of media delivery). That is, a sender should scale the quality of transmission (known as quality adaptation) depending on prevailing network conditions [1, 9]. In doing so, the main challenge is to minimize the variations in quality while obeying the congestion controlled rate-limit.

The required rate control can be realized in a number of ways. In case of on-line streaming with real-time encoder, the encoder's rate controller can respond to the update of the available bandwidth as studied in [3]. However, for the streaming of pre-encoded media contents, the situation is different. One option, called simulcast, encodes the stream at various target rates and switches between the previously encoded layers as the available bandwidth changes. Alternatively one may think of rate shaping(or filtering) that matches the rate of a pre-compressed stream to the target rate constraint. In a more systematic approach, a layered scalable coding scheme is used to tackle the bandwidth fluctuation problem. In the layered coding (hierarchical encoding), the stream is encoded at a base layer and one or more enhancement layers, which can be combined to render the stream at higher quality. As the available bandwidth varies, the number of enhancement layers is adjusted by the sender. However, even switching between layers of stream is not a trivial task.

2.2 Error Controls: Proactive and Reactive

Packet loss is one of the key factors affecting the perceptual quality of media delivery. Although media applications are somewhat resilient to packet loss (i.e., it does not need perfect reliability), excessive packet losses can result in severe impairment to the playback continuity. Extensive efforts to recover and conceal the network losses can be classified into several categories. There are retransmission-based ARQ (automatic repeat request) [10], packet-level FEC (forward error correction) [11], hybrid FEC/ARQ [12], and error concealment [13].

As a reactive error control, retransmission-based ARQ has been believed inappropriate for real-time applications since it requires at least one round-trip time to repair a lost packet. Without proper arrangement, it is not easy to use for multicast environment due to feedback implosion. However, despite of latency drawback and multicast scalability problem, delay-constrained ARQ is the most attractive candidate because of its bandwidth efficiency and low processing complexity. Packet-level FEC is an alternative solution to packet loss (or erasure) and can be grouped into a proactive error control. FEC is appealing to delay-stringent applications and thus has been considered appropriate for real-time multicast environment. However, the redundancy overhead, the processing complexity, and the reduced inefficiency for burst loss are weak points. Thus, diverse forms of hybrid FEC/ARQ are being adopted to enhance the reliability and repair efficiency [12].

On the contrary, error concealment is a receiver-based, application-level scheme to mitigate the impact of a packet loss. That is, the major role of error concealment is not the actual recovery of the lost packets but the reconstruction of the missing information with adjacent packets of the lost. If perfect reliability is not required as in the case of continuous media and network losses are rather isolated, error concealment becomes very useful. Lost audio packets, for example, can be concealed by insertion-based, interpolation-based, and regeneration-based schemes [14].

2.3 Synchronization and Adaptive Playout

The playout of delivered media needs to be synchronized according to the associated timing information (e.g., timestamp per each packet). The goal of media synchronization is to reconstruct the timing relation of media contents at the clients. The noticeable disruption (due to buffer underflow or system overload) and/or discrepancy (due to playout timing skews between different media objects) in the playout significantly degrades the playback quality. There are three types of synchronization such as *intra-media*, *inter-media*, and *inter-client* synchronization [15 18]. The intra-client media synchronization (including both intra/inter-media synchronizations) has been considered as an important issue. To keep the synchronization, media applications have to deal with the network delay jitter and loss. They also need to keep the smooth playout regardless of the exceptional events at the system. In addition, we can expect to see different clients are playing different portion of media. This situation is happening since each client is connected to the sender through paths of different bandwidth, loss, and delay. The capability difference of client system is another reason. Thus, we need to address the playback synchronization issue in both intra- and inter-client aspects while properly controlling the buffers at the clients.

Intra-media synchronization imposes the temporal constraint on packets of a single stream [18]. To control the skew, the timing discrepancy, dynamic schemes like discarding/skipping, shortening/extension of duration, and virtual time contraction/expansion have been developed [17]. Note that allowed skew between different types of media is tightly coupled with human perception. Extending it to multiple media streams, inter-media synchronization handles lip synchronization for audio and video [19 - 21]. Human perception is more sensitive to audio discontinuity and thus the audio is generally chosen as the reference. The inter-media synchronization can be classified into feedback-based and timestamp-based approaches. While simpler feedback-based scheme is suitable for light-weight sessions, the timestamp-based approach can synchronize tightly-coupled session at the cost of complexity.

To maintain the playout synchronized despite of the network fluctuations and system limitations, an *adaptive playout* is required [22]. The main role of the adaptive playout control is to reduce the discontinuity incurred by packet over-/underflows and momentary CPU overload. An effective playout adaptation allows us to avoid excessive packet droppings at the application, conceal the network fluctuations and thus minimize the degradation of perceptual media quality. With the audio time-scale modification [23], we can significantly enhance the adaptation capability of the client at the cost of increased computation. To perform the adaptive playout with the time-scale modification, we need to know the allowed ratio within which a player can manipulate the playback speed without being detected by the user s perception. Even though the allowed playout variation differs based on the type of audio (including the silence), we assume that playout variation up to 50% is usually unnoticeable. Note that the playback speed should be changed with caution to keep the playout consistent and smooth.

3 Networking Support for Network Adaptive Media Delivery

3.1 Inter-Operable Delivery with Protocol Support

Protocols designed and standardized for media delivery provide such services as network addressing, transport, and session control. Fig. 2 illustrates relevant protocols, based on the Internet multimedia conferencing architecture of IETF MMUSIC (multiparty multimedia session control) working group [24]. The protocols can be classified into three categories: network-layer protocol such as Internet protocol (IP), transport-layer protocols such as UDP and TCP, and application-layer protocols. Real time streaming protocol (RTSP), real-time transport protocol / RTP control protocol (RTP/RTCP), session description protocol (SDP), session announcement protocol (SAP), and session initiation protocol (SIP) are application layer protocols to be discussed here.

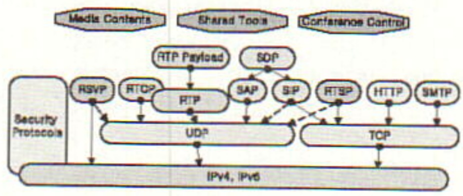


Figure 2: Inter-operable media delivery related protocols.

RTP is the representative (standardized) protocol for real-time media transport in the Internet [25]. RTP is carried on top of UDP. Although it handles real-time delivery based on application level framing (ALF) concept [26], it in itself does not enforce anything to guarantee service quality but acts as a helper. RTP is typically working in conjunction with RTCP to monitor and feedback session status and quality. The pair covers sequencing and loss detection, timing recovery, synchronization, simple session control, and QoS reporting. The profile of RTP/RTCP has been evolving for audio and video transmission as in [27]. It provides a standardized fixed header followed by a payload. The 12-byte RTP header contains payload type identifying the specific content type carried in the payload, sequence number being incremented for each packet, timestamp describing the sampling instant, synchronization source for identifying original RTP source, contributing source, and others. The monitoring of a RTP session is carried in RTCP packets such as receiver report (RR) and sender report (SR). For example, RR contains latest sequence number received, number of lost packets, estimated packet inter-arrival, and relevant timestamps. Exchanged RTCP information can be used to assist congestion control, session management, and synchronization. To support multicast, RTCP messages are limited to 5% of RTP bandwidth and 5 seconds of frequency. Note however that a new rule for RTCP feedback is being drafted to remove this restriction [28].

RTSP (Real Time Streaming Protocol) is a client-server multimedia presentation control protocol, designed to address the needs for efficient delivery of streamed multimedia over IP networks [29]. It leverages existing web infrastructure and works well both for large audiences as well as single-viewer media-on-demand. RTSP is designed to work with time-based media, such as streaming audio and video, as well as any application where application-controlled, time-based delivery is essential. It has mechanisms for time-based seeks into media clips, with compatibility with many timestamp formats, such as SMPTE timecodes. In addition, RTSP is designed to control multicast delivery of streams, and is ideally suited to full multicast solutions, as well as providing a framework for multicast-unicast hybrid solutions for heterogeneous networks like the Internet.

SDP (session description protocol) is actually a media header format (or textual syntax), intended for describing multimedia sessions for the purposes of session announcement (e.g., SAP), session invitation (e.g., SIP), and other forms of multimedia session initiation (e.g., with RTSP, e-mail using MIME extensions, and HTTP) [30]. SDP is designed to convey sufficient information to enable participating into the session. SAP (session announcement protocol) provides session description in SDP via multicast, which also serves as a crude address allocation protocol [31]. A SAP server periodically multicasts an announcement packet to a well-known multicast address and port, which receivers can listen on to build a complete directory of sessions. The SIP (session initiation protocol) is an application-layer control (signaling) protocol for creating, modifying and terminating sessions with one or more participants [32]. SIP invitations used to create sessions and carry session descriptions which allow participants to agree on a set of compatible media types. It transparently supports name mapping and redirection services, which supports personal mobility [27] - users can maintain a single externally visible identifier regardless of their network location. SIP supports five facets of establishing and terminating multimedia communications: user location, user availability, user capabilities, session setup, and session management. SIP works as a component that can be used with other IETF protocols (e.g., RTSP, RTP, SDP, etc) to build a complete multimedia architecture.

3.2 Multicast Support for Multi-point Media Distribution

As a viable option to save the precious bandwidth, multicast provides immense merits in designing media applications. Especially for continuous media (e.g., audio and video), it allow us to support a large set of clients simultaneously. The native IP multicast (called as *any source multicast*, ASM) directly inherits the open properties of the Internet and thus it has no restrictions for creating groups, sending or receiving multicast traffics to/from the groups. To become a session member a host reports group membership to the nearest query router managing group membership through internet group management protocol (IGMP) for IPv4. Routers exchange signaling messages according to routing protocols to set up a multicast spanning tree connecting all session members. Based on the way the multicast spanning tree is built [33], existing multicast routing protocols are further divided into flood-and-prune (dense-mode) and explicit-join styles. The former includes distance vector multicast routing protocol (DVMRP), multicast open shortest path first (MOSPF), and protocol independent multicast dense mode (PIM-DM) while the latter is represented by PIM sparse mode (PIM-SM) and core-based tree (CBT) protocols. The popular choice for multicast deployment is the PIM-SM that supports adaptive transition between source-based and shared tree. In addition, multiprotocol extensions to border gateway protocol (MBGP) / multicast source discovery protocol (MSDP) are need to advertise reverse paths towards sources and disseminate session information across the network domains.

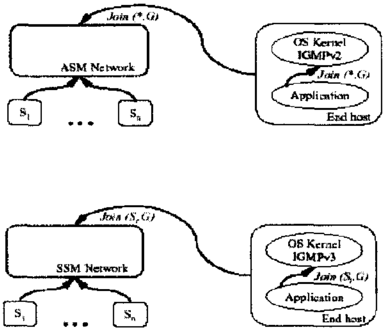


Figure 3: Multicast service models: Any source multicast (ASM) and source specific multicast (SSM).

However, ASM multicast model is facing problems including lack of access control, multicast address allocation and collision, deployment complexity, and others. Due to this manageability and scalability problems of ASM multicast, the ubiquitous deployment of IP multicast on the Internet is expected only after several years later [34]. Fortunately, new multicast service models, named as *source-specific multicast* (SSM) [35], *application-layer multicast* (ALM) [36], are appearing as a promising solution for multicast media distribution. The SSM simplifies the protocol complexity of multicast routing at the network and reduces the deployment cost at the expense of providing a limited service model, i.e., suitable only for one-to-many multicast services. A key difference between the ASM and the SSM is illustrated in Fig. 3 [37]. The SSM model requires a SSM-aware application, which has the source filtering functionality at end-hosts for joining or leaving a multicast group (or channel). It uses a specific source address and a SSM destination address (i.e., a multicast group address) to subscribe the multicast channel. The channel subscription in the SSM is performed using IGMPv3. Also, there are tunneling techniques proposed recently such as UMTP (UDP multicast tunneling protocol) [38] and AMT (automatic multicast tunneling protocol) [39]. Like this, in ALM, mixture of schemes in either network or application layer is utilized to build overlay multicast. These more feasible multicast models, together with the enhanced support for media delivery from the underlying networks and the ever-increasing computing power in the end-systems, has the potential to make multicast applications wide spread.

Scalable and reliable multicast need to solve so called packet implosion and exposure problems [40]. Extensive reliable multicast proposed so far includes scalable reliable multicast (SRM), reliable multicast transport protocol (RMTP), log-based receiver-reliable multicast (LBRM), active error recovery/nominee congestion avoidance (AER/NCA), and lots of others [41, 42]. The receiver-driven layered multicast (RLM) and its hybrid FEC/pseudo-ARQ extension is also important [43, 44]. Recently, by the IETF RMT (reliable multicast transport) work group, building blocks are being proposed based on the NACK (negative acknowledgement), tree-structured ACK (TRACK), advanced layered coding and FEC (ALC/FEC), and generic router assist (GRA) schemes [45].

3.3 QoS-network Support for Quality Media Delivery

The best-effort Internet is gradually moving towards providing a different level of assurance in terms of network QoS parameters within its resource capacity. Two representative approaches in the IETF are *integrated services* (IntServ) with the resource reservation protocol (RSVP) [46] and *differentiated services* (Differv or DS) [47]. In the beginning, the QoS problem of Internet has been approached via resource allocation of IntServ/RSVP, in which each flow attempts to reserve the resource so that the packet loss rate and the delay are bounded. Although it can provide guarantees, the associated admission control scheme is so complicated that it is still difficult and premature to deploy. The emerging DiffServ scheme in IP-QoS methods enables to provide service differentiation in a simple and scalable manner. In the DiffServ model, resources are allocated differently for various aggregated traffic flows based on a set of bits (i.e., DS byte defined in IP header). Consequently, the DiffServ approach allows different QoS grades to different classes of aggregated traffic flows.

DiffServ working group in IETF have defined two services: a premium service (PS), which expects the virtual leased line service to support low loss and delay/jitter, and an assured service (AS), which provides better than best-effort but without guarantee. Per-hop behaviors (PHBs) specify the required forwarding behaviors for the packets according to the DS levels. The expedited forwarding (EF) PHB [48] for DiffServ PS specifies a forwarding behavior in which packets see very small losses and queuing delays. The EF PHB, based on the priority queuing, better suits latency stringent applications at the cost of higher price. The assured forwarding (AF) PHB [49] for DiffServ AS specifies preferential dropping of best-effort and/or out-of-profile packets when congestion occurs. By limiting the amount of AF flows and by managing the best-effort traffic appropriately, network nodes can ensure a lower loss to AF marked packets. As a result, the DiffServ provides DS levels of different losses and delays. Thus, the DiffServ architecture, especially the relative DiffServ [50], as a deployment is an attractive approach that does not require admission control, resource reservations, or signaling. It can provide higher classes receive better services(e.g., lower delay/jitter and lower loss rate) than lower classes.

However, depending on the application, reservation on resource with more specific guarantee may be preferred. A resource manager (a.k.a., bandwidth broker [51]) can be employed to complement IntServ/RSVP with DiffServ in the pursuit of end-to-end QoS [52]. Like this, there is a trade-off between easy deployment/management of the QoS network and strict provisioning of QoS service. Recently, in a desire to sort out a simplified solution for QoS signaling, IETF NSIS (next step in signaling) working group starts to develop the requirement, architecture, and protocols for signaling QoS. In summary, we believe simpler network architecture with more complex end-system is more preferable and is wellmatched with the Internet end-to-end design concept. Adequate and scalable QoS support in a network is still required for QoS-sensitive applications even though networked media applications becomes more network-aware and network-adaptive.

3.4 Toward Consolidated Support via Middleware

So far, we reviewed networking support from protocol, multicast network, and QoS network. However, lots of issues are not covered yet. Systems, which handle the transport with operating system and storage, constitute important portion of media delivery chain. System support issues are heavily related to the design problem of efficient and reliable storage and retrieval of media stream on/from disk arrays. Also, lots of media applications require scalable, navigable identification. For example, user need a user-friendly, portable, global identifier to initiate a conference session and authentication (as part of security support) to verify corresponding party. Thus, we need a glue, *middleware*, that constitutes a layer of software between the network and the applications. This software provides services such as identification, authentication, authorization, directories, and security. Unfortunately, in current Internet, applications usually have to provide these services themselves, which leads to competing and incom-patible standards and implementations. By promoting standardization and inter-operability, middleware will make advanced network applications much easier to use. Thus, as a set of core software components that permit scaling of applications and networks tools that take the complexity out of application integration, middleware constitutes a second layer of the IT infrastructure. With middleware support in place, an effective media delivery application should select proper solution components. They can integrate solution components to achieve the end-to-end performance while maintaining feasibility and addressing scalability, security, and so on.

With this goal, several initiatives are being made to build middleware. In large scale, there is Globus toolkit Grid middleware activity that attempts to help Grid computing by integrating software applications and tools so that they will work together easily and seamlessly over the Internet [53]. Grid computing, which facilitates sharing of online resources and applications across multiple sites, requires software tools that provide standards for security, resource and data management, communication, job scheduling, and other functions [54]. Globus toolkit provides a bag of services that can be used either independently or together. It includes resource allocation manager (GRAM), security infrastructure (GSI), directory service (MDS), global storage (GASS), and monitoring (HBM). Also, complementing the Grid middleware effort, Internet2's middleware work is focusing on the the deployment of interoperable core middleware services on issues like identification, authentication, authorization, directories, and security.

4. Dynamic Network Adaptation Framework

To promote coordinated interaction between the network-adaptive applications and the networking supports, we are proposing the following network adaptation framework.

4.1 Overview of Network Adaptation Framework

The proposed dynamic network adaptation framework is illustrated in Fig. 4 focusing on the delivery of rich media contents. The media (especially video) contents are first pre-processed and layer-encoded. Following the target (albeit assumed) constraints on the bandwidth and buffer, constant quality rate control manipulates the rate composition among the base and enhancement layers in the layered encoding. At the same time they are analyzed with R-D (rate-distortion) and corruption model [55]. Then the encoded media stream with the associated R-D/corruption model parameters is passed to the network adaptation/prioritized packetization module to wait for the delivery. In this module, the encoded stream is first tailored (or transcoded) in the source rate/error-resilience sense to match the given estimated available bandwidth/loss/delay of the underlying network. Then it is packetized with priority (i.e., the layered-RPI) before going through the network adaptation at the sender. Based only on the priority, they are adapted to the network condition in rate/loss/delay sense. That is, the packets are selectively discarded and protected with differentiation. Note that various types of feedbacks are available to guide the required network adaptation.

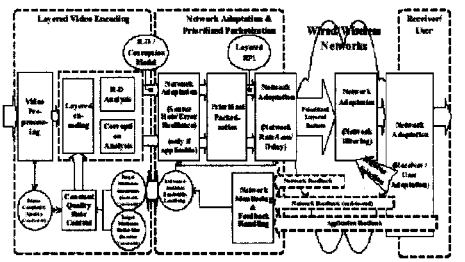


Figure 4: Packetized media delivery with the dynamic network adaptation.

Guided by both end-to-end network feedback from end-to-end congestion control and network feedback from router indicating congestion, the sender could initiate the network adaptation. Application level feedback can also notify about the playout status at the corresponding party and may request the speedup of slowdown of transmission for synchronized playback. Once sent to the network, it may go through network-initiated adaptation, which is also called as network filtering (e.g., with schemes such as priority-based packet dropping and receiver-based layer selection). Finally, the delivered packets are adaptively processed at the receiver to match the receiver capability and user preference.

The proposed framework basically assumes the existence of a network or other equivalent options that support prioritized variable-rate delivery of media stream and the associated end-to-end performance and cost (e.g., pricing) models. Since a media codec has several options to trade the compression efficiency for flexible delay manipulation, error resilience, and network friendliness, the coordination (i.e., network adaptation) framework has to provide an simplified interaction process between the media application and the target network (or networking services). Note that the interaction is taking place at multiple junctions as the media stream is delivered from the sending application, via the underlying network, and to the receiving application. Note that the main purpose of introducing the layered-RPI is to abstract and isolate the coding details from the network adaptation task. By assigning layered-RPI to each media stream in an appropriate manner, the proposed framework can accommodate the demand of each stream to achieve the best end-to-end performance in adapting to the fluctuating networks. Given the prioritization of media stream, the proposed network adaptation can be controlled in both feedforward and feedback sense. They need to accommodate the fluctuation of the given network in addition to the inherent variability of media stream and receiver/user heterogeneity. Thus, the adaptation should focus on how to dynamically react to sudden changes in the application and network.

4.2 Source Prioritization and Layered-RPI

For the media applications, the layered-RPI assignment should reflect the influence of each stream (or down to packet) to the end-to-end quality. Packets will be marked by the content-aware applications in the granularity of session, flow, layer, and/or packet. Among three key parameters for QoS (rate, error, and delay), it is important to associate priority for loss and delay, respectively. Note that the rate (or bandwidth) is linked with the layering itself and the extreme case of this is MPEG-4 FGS (fine-grained scalability) layering in video case.

Most of existing prioritization schemes are in coarse granularities of session, flow, and layer. The per-flow prioritization is promoted under the name of user-based allocation within access networks [56]. Also, lots of prioritization for the UEP is best matched with layer-based differentiation as done in [57] with object-based scalability. For delay, the session-based granularity to account for the delay effect of the source seems a first choice. Since the application context (e.g., interactive vs MOD-style) plays a crucial role in delay prioritization, RDI (relative delay index) is kept constant for whole session (e.g., session (A) and (B) in Fig. 5).

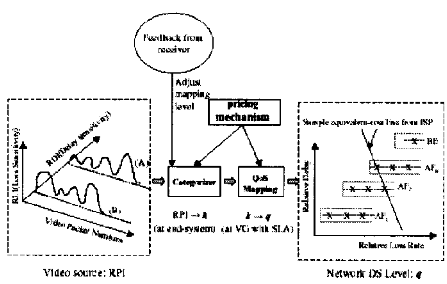


Figure 5: Source prioritization using RPI and network adaptation for loss and delay.

However, the prioritization can be differentiated down to each packet to enable a fine-grained differentiation. Packet-based prioritization may be adopted to accurately account for the impact of each packet to the end-to-end quality. Especially the loss impact, quantified by RLI (relative loss index), is dependent on the employed media coding scheme. In case of standard-based video such as ISO/IEC MPEG-4 and ITU-T H.263. At the packet-level, there are dependency relations such as semantic and prediction dependency. The semantic packet-level dependency exists between a packet that includes header parameters and other dependent packets that needs them for decoding. Also, linked by spatial or temporal prediction, the corruption caused by a packet loss can affect the decoding of following packets. This is called the prediction packet-level dependency and we have developed a corruption model to quantify this dependency (i.e., loss impact of a packet) [55]. With the proposed corruption model, the loss impact of each macroblock is explained by taking into account the error concealment, the temporal dependency, and the loop filtering effect. The corruption of macroblocks in a packet is then merged to explain the RLI. For more detailed discussion of this topic, we refer to [55].

4.3 Desired Behavior of Network Adaptation

Basically, well-implemented network adaptation can bring benefit to both media applications at end and networks by providing better service match at the willingness to pay more complexity. For the network-adaptive applications, the dynamic network adaptation should consider interests of both. That is, the application should get benefit from its network adaptation capability while the network enjoys the benefit of different charging and maximizes the end-user satisfaction. Under a given cost constraint, an efficient network adaptation is trying to match the layered-RPI to the underlying network service level. In this process, the adaptation granularity has to be manipulated intelligibly. The effectiveness is dependent on the accurate association of layered-RPI to each packet, which is one of key investigation issues.

In general, the issue of solving all these network adaptation for rate/loss/delay at the same time is too challenging to be solved simultaneously. Thus, depending on the situation, one may focus on the error and delay control separating the rate control issue. For example, one can simply enforce maximum to the allowed transmitting rate by token bucket (TB) policing in either per-flow or aggregate-flow sense. Then, for QoS 2-tuple *f*delay, loss*g*, which is actually the major concern of the proposed layered-RPI, appropriate network adaptation is requested in different degrees by user applications, anticipating different levels of guarantee (or assurance) according to the price paid. That is, each application will demand its loss rate/delay preference by marking the layered-RPI, which is further divided into loss and delay part, respectively. In the proposed network adaptation, each media stream can demand different loss and delay treatment as shown in the left side of Fig. 5. Underlying network will meet these demands with its service provisioning capability. It may provide several differentiated delay and loss levels as shown in the right side of Fig. 5 like the DiffServ network [47]. Or similarly we can envision the effect of FEC/ARQ error controls for the above differentiation.

4.4 Dynamic Network Adaptation: Examples

Now we introduce two kinds of deployment cases for the proposed framework. The first one focuses on the packet-level unequal error protection (UEP) and handles the mapping of prioritized packet to the available QoS service of the DiffServ network [58]. The other example is showcasing the case of synchronized multicast streaming that exploits the network adaptive interaction between multiple clients [59].

Network Adaptation: QoS-mapping for Media over DiffServ Network: This approach under the relative DiffServ paradigm focuses on the QoS mapping between network-aware streaming media applications and DS levels. We designated the term video gateway (VG) for the traffic conditioning entity and the VG (co-located with boundary router) is responsible for realizing the

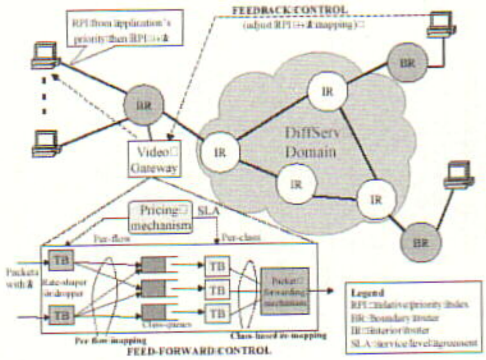


Figure 6: Feedback and feedforward control fo media over DiffServ.

specialized traffic managements for attached applications. The QoS mapping covers both layered-RPI prioritization and feedforward/feedback QoS mapping. Each packet is categorized into layer k by RPI at end-systems, without knowing about other applications. An assigned RDI limits the range of $k \rightarrow q$ mapping level to meet a certain statistical delay range. From given loss-rates, and unit costs of DS levels, we can extract an effective mapping set $(k \rightarrow q)$ under total cost constraint in layer granularity [60]. The traffic conditioning is performed via TB-based re-marking by degrading $k \rightarrow q$ mapping level when a flow or a class traffic volume exceed allowed bandwidth level in the feedforwad control. Feedback-based network adaptation enable the fine-tuned control on top of coarse feedforward mapping. Receiver sends a report of delay/packet loss to sender whenever necessary and can ask the QoS mapping level of $RLI \rightarrow k$ and RDI when he/she can not satisfy the received quality. Also client (or receiver) asks the QoS mapping level to be decreased when he/she thinks current received quality is over-provisioned in order to reduce the charging bill under the scenario that video server charges differently. This feedback mechanism enables the whole network adaptation to be adjusted dynamically and to stablize the end-to-end QoS within an acceptable range. Please refer to [60] for the performance results.

Network Adaptation: Synchronized Multicast Media Streaming: To reduce the playback discontinuity and mitigate the heterogeneity, we can establish a *synchronized multicast streaming framework*, where the synchronized playout of all clients is adaptively managed. Note that synchronization issue becomes more important as the media streaming goes broadband and high-quality. In the proposed scenario, each client is required to keep synchronization despite of the exceptional system events as well as the network fluctuations. To assist each client for this challenge, we propose to adaptively control the playback speed of playout. By extending the audio/speech adaptive playout with time-scale modification in [23, 61] to audio/video, the playback speed of client can be varied. That is, based on the combined buffer (i.e., network and application buffers) occupancy level, the player at the client is adaptively expanding or contracting the playout within the range that it does not hurt the viewers perceptually.

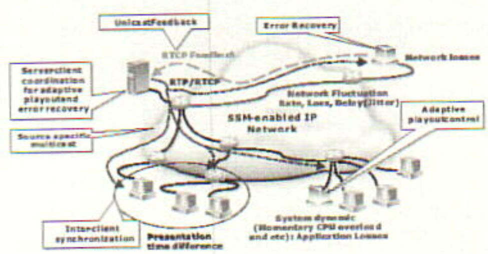


Figure 7: Synchronized multicast media streaming scenario.

Thus, we can proactively conduct the adaptive playback not only to reduce the playback discontinuity but also to guarantee high-quality playback with flexible error controls. In summary, the proposed solution consists of 1) local playback adaptation (guided by a playback factor) based on the combined buffer occupancy with the error control, 2) unicast RTCP feedback on the presentation time as well as the channel status, 3) inter-client synchronization with the aid from the server, and 4) cumulative NACK-based error recovery with the assistance of adaptive playout control. More specifically, each client locally controls the playback speed to prevent buffer overflow/underflow (subsequently to prevent playback discontinuity) and to assist the delay-constrained retransmission attempt if allowed. This local adaptation is then reviewed at the server by aggregating client feedbacks and the server will issue target presentation time to coordinate and synchronize all the clients. Cumulative NACK-based error recovery is also assisted by the adaptive playback to secure enough time for request and reply of retransmission. Furthermore, to reduce unnecessary feedbacks, whether to request retransmission or not is selectively determined. Results show that the proposed framework can reduce the playback discontinuity without degrading the media quality while enhancing the inter-client synchronization by mitigating the client heterogeneity. At the same time, it can assist the retransmission-based error recovery with the adaptive playout and reduce bandwidth through the selective retransmission and cumulative feedbacks. Please refer to [59] for details.

5. Conclusion

In this paper, we have reviewed challenges in delivering broadband, rich media contents reliably and scalably over a variety of underlying networks. For the interim period until fully established networking services are operational, we view that the media applications at the edge of network has to be more and more adaptive. After discussing the required network adaptation functionalities and the variety of networking support issues, we propose a dynamic network adaptation framework that efficiently leverages its awareness of both media application (including contents) and underlying networking support. We believe the framework can give insight to the desired interaction between the network-adaptive media applications and the media-aware networking services of the future.

REFERENCES

[1] D. Wu, Y. Hou, W. Zhu, Y. Zhang, and J. Peha, "Streaming video over the Internet: Approaches and directions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 1, Feb. 2001.

[2] R. Rejaie, M. Handley, and D. Estrin, "Quality adaptation for unicast audio and video," in *Proc. ACM SIGCOMM*, Sept. 1999.

[3] J. Kim et al., "TCP-friendly internet video streaming employing variable frame-rate encoding and interpolation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 7, Oct. 2000.

[4] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Trans. on Networking*, vol. 7, no. 4, pp. 458-472, Aug 1999.

[5] J. Widmer, R. Denda, and M. Mauve, "A survey on TCP-friendly congestion control," *IEEE Network Magazine*, vol. 13, no. 3, pp. 28 37, May 2001.

[6] S. Floyd et al., "Equation-based congestion control for unicast applications," in *Proc. ACM SIGCOMM*, Aug 2000, pp. 43-56.

[7] Y. Kim, J. Kim, and C.-C. J. Kuo, "Smooth and fast rate adaptation mechanism (SFRAM) for TCP-friendly Internet video," in *Proc. Packet Video Workshop*, Sardinia, Italy, May 2000.

[8] H. Balakrishnan and S. Seshan, "The congestion manager," RFC 3124, IETF, June 2001.

[9] R. Rejaie, M. Handley, and D. Estrin, "Layered quality adaptation for internet video streaming," *IEEE Journal on Selected Areas of Communications*, vol.18, 2000.

[10] C. Papadopoulos and G. M. Parulkar, "Retransmission-based error control for continuous media application," in *Proc. of*

the 6th International Workshop for Network and Operating Systems Support for Digital Audio and Video, Apr. 1996.

[11] E. Biersack, "Performance evaluation of forward error correction in ATM networks," in *Proc. ACM SIGCOMM*, Aug. 1992, pp. 248-258.

[12] J. Nonnenmacher, E. Biersack, and D. Towsley, "Parity-based loss recovery for reliable muticast transmission," *IEEE/ACM Trans. on Networking*, vol. 6, no. 4, pp. 349-361, Aug. 1998.

[13] M. Wada, "Selective recovery of video packet loss using error concealment," *IEEE Journal on Selected Areas in Communication*, vol. 7, pp. 807-814, June 1989.

[14] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet-loss recovery techniques for streaming audio," *IEEE Network Magazine*, vol. 12, pp. 40-48, Sept. 1998.

[15] G. Blakowski and R. Steinmetz, "A media synchronization survey: Reference model, specification, and case studies," *IEEE Journal on Selected Areas in Communication*, vol. 14, no. 1, pp. 5-35, Jan 1996.

[16] Y. Ishibashi, S. Tasaka, and H. Ogawa, "A comparison of media synchronization quality among reactive control schemes," in *Proc. IEEE INFOCOM*, Apr. 2001, vol. 1, pp. 77-84.

[17] Y. Ishibashi and S. Tasaka, "A comparative survey of synchronization algorithms for continuous media in network environments," in *Proc. IEEE LCN 2000*, Nov. 2000, pp. 337-348.

[18] P. Rangan, S. Kumar, and S. Rajan, "Continuity and synchronization," *IEEE Journal on Selected Areas in Communication*, vol. 14, no. 1, pp. 52-60, 1996.

[19] M. Yuang, B. Lo, Y. Chen, and P. Tien, "A synchronization paradigm with QoS guarantees for multimedia communications," in *Proc. IEEE GLOBECOM*, 1999, vol. 1a, pp. 214-220.

[20] Y. Ishibashi, S. Tasaka, and Y. Tachibana, "Adaptive causality and media synchronization control for networked multimedia applications," in *Proc. IEEE ICC*, 2001, vol. 3, pp. 952-958.

[21] A. Corte, A. Lombardo, S. Palazzo, and G. Schembra, "A feedback approach for jitter and skew enforcement in multimedia retrieval services," in *Proc. IEEE GLOBECOM*, 1995, pp. 790-794.

[22] B. Moon, J. Kurose, and D. Towsley, "Packet audio playout delay adjustment: performance bounds and algorithms," *ACM/Springer Multimedia Systems*, vol. 5, no. 1, pp. 17 28, Jan. 1998.

[23] F. Liu, J. Kim, and C.-C. J. Kuo, "Adaptive delay concealment for internet voice applications with packet-based time-scale modification," in *Proc. IEEE ICASSP*, May 2001.

[24] M. Handley, J. Crowcroft, C. Bormann, and J. Ott, "The internet multimedia conferencing architecture," Internet draft, IETF, 1997.

[25] A. Schulzrinne, S. Casner, Frederderick, and V. Jacobson, "RTP: A transport protocol for real-time applications," RFC 1889, IETF, Jan. 1996.

[26] D. Clark and D. Tennenhouse, "Architecture considerations for a new generation of protocols," in *Proc. ACM SIGCOMM*, Sept. 1990, pp. 201-208.

[27] H. Schulzrinne, "RTP profile for audio and video conferences with minimal control," RFC 1890, IETF, Jan 1998.

[28] J. Ott et al., "Extended RTP profile for RTCP-based feedback," Internet Draft draft-ietf-avt-rtcpfeedback-02.txt, IETF, July 2001.

[29] H. Schulzrinne, A. Rao, and R. Lanphier, "Real time streaming protocol (RTSP)," RFC 2326, IETF, Apr. 1998.

[30] M. Handley and V. Jacobson, "SDP: Session description protocol," RFC 2327, IETF, Apr. 1998.

[31] M. Handley, C. Perkins, and E. Whelan, "Session announcement protocol," RFC

2974, IETF, Oct. 2000.

[32] J. Rosenberg et al., "SIP: Session initiation protocol," RFC 3261, IETF, June 2002.

[33] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefe, "Deployment issues for the IP multicast service and architecture," *IEEE Network Magazine*, vol. 14, pp. 88-98, Jan. 2000.

[34] K. C. Almeroth, "The evolution of multicast: from the MBone to interdomain multicast to Internet2 deployment," *IEEE Network Magazine*, vol. 14, pp. 10-20, Jan. 2000.

[35] H. Holbrook and B. Cain, "Source-specific multicast for IP," Internet Draft draft-ietf-ssm-arch-00.txt, IETF, Nov. 2001.

[36] S. G. Rao Y. Chu and H. Zhang, "A case for end system multicast," in *Proc ACM SIGMETRICS*, June 2000, pp. 1-12.

[37] K. C. Almeroth, S. Bhattacharyya, and C. Diot, "Challenges of integrating ASM and SSM IP multicast protocol architectures," in *Proc. Tyrrhenian International Workshop on Digital Communication (IWDC)*, 2001, pp. 343-360.

[38] R. Finlayson, "The UDP multicast tunneling protocol," Internet Draft draft-finlayson-umtp-06.txt, IETF, Mar. 2001.

[39] D. Thaler et al., "IPv4 automatic multicast without explicit tunnels (AMT)," Internet Draft draftietf-mboned-auto-mulitcast-00.txt, IETF, Feb. 2001.

[40] C. Papadopoulos, G.Parulkar, and G. Varghese, "An error control scheme for large-scale multicast applications," in *Proc. IEEE INFOCOM*, Mar. 1998, pp. 1188-1196.

[41] K. Obraczka, "Multicast transport protocols: A survey and taxonomy," *EEE Communications Magazine*, vol. 36, no. 1, pp. 94-102, Jan. 1998.

[42] B. N. Levine and J. J. Garcia-Luna-Aceves, "Comparison of reliable multicast protocols," *Multimedia Systems*, vol. 6, no. 5, pp. 334-348, 1998.

[43] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proc. ACM SIGCOMM*, Aug. 1996, pp. 117-130.

[44] P. A. Chou, A. E. Mohr, A. Wang, and S. Mehrotra, "Error control for receiver-driven layered multicast of audio and video," *IEEE Trans. on Multimedia*, vol. 3, pp. 108-122, Mar. 2001.

[45] M. Handley, S. Floyd, B. Whetten, R. Kermode, L. Vicisano, and M. Luby, "The reliable multicast design space for bulk data transfer," RFC 2887, IETF, Aug. 2000.

[46] R. Braden et al., "Resource reservation protocol (RSVP) version 1 functional specification," RFC 2205, IETF, Sept. 1997.

[47] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," RFC 2475, IETF, Dec. 1998.

[48] B. Davie et al., "An expedited forwarding PHB (per-hop behavior)," RFC 3246, IETF, Mar. 2002.

[49] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured forwarding PHB group," RFC 2597, IETF, June 1999.

[50] C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional differentiated services: Delay differentiation and packet scheduling," in *Proc. ACM SIGCOMM*, Boston, MA, September 1999.

[51] Zhi-Li Zhang et al., "Decoupling QoS control from core routers: A novel bandwidth broker architecture for scalable support of guaranteed services," in *Proc. ACM SIGCOMM*, Aug. 2000, pp. 71-83.

[52] Y. Bernet et al., "A framework for integrated services operation over Diffserv networks," RFC 2998, IETF, Nov. 2000.

[53] I. Foster and C. Kesselman, "Globus: A metacomputing infrastructure toolkit," *International J. Supercomputer Applications*, vol. 11, no. 2, pp. 115-128, 1997.

[54] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," *International J. Supercomputer Applications*, vol. 15, no. 3, 2001.

[55] J.-G. Kim, J. Kim, J. Shin, and C.-C. J. Kuo, "Coordinated packet-level protection with a corruption model for robust video transmission," in *Proc. SPIE Visual Communication Image Processing (VCIP)*, Jan. 2001.

[56] A. Banchs and R. Denda, "A scalable share differentiation architecture for elastive and real-time traffic," in *Proc. IEEE/IFIP IWQoS*, June 2000.

[57] Y. T. Hou, D. Wu, B. Li, T. Hamada, I. Ahmad, and H. J. Chao, "A differentiated services architecture for multimedia streaming in next generation Internet," *Computer Networks*, vol. 32, no. 2, pp. 185-209, Feb. 2000.

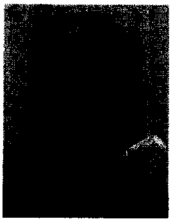
[58] J. Shin, J.-G. Kim, J. Kim, and C.-C. J. Kuo, "Dynamic QoS mapping framework for relative service differentiation-aware video streaming," *European Transaction on Telecommunication*, vol. 12, no. 3, pp. 217-230, May/June 2001.

[59] J. Jo and J. Kim, "Synchronized one-to-many media streaming with adaptive playout control," in *Proc. SPIE ITCOM*, July 2002.

[60] J. Shin, J. Kim, and C.-C. J. Kuo, "Quality-of-service mapping mechanism for packet video in differentiated services network," *IEEE Trans. on Multimedia*, vol. 3, pp. 219-231, June 2001.

[61] E. Steinbach, N. F¨arber, and B. Girod, "Adaptive playout for low-latency video streaming," in *Proc. International Conference on Image Processing (ICIP)*, Oct. 2001, pp. 962-965.

김 종 원 (Jong-Won Kim) 정회원



1987, 1989, 1994년 2월: 서울대학교 제어계측공학과 학사, 석사, 박사
2001년 9월~현재: 광주과학기술원 정보통신공학과 부교수
2002년 12월~현재: ANF Application Tech. Area 의장, GFK Access Grid WG 의장
1998년 12월~2001년 7월: 미국 Univ. of Southern California, EE-Systems Department 연구조교수
1994년 3월~1999년 7월: 공주대학교 전자공학과 조교수

<주관심분야> Networked Media Systems and Protocols focusing "Reliable and Flexible Delivery for Integrated Media over Wired/Wireless Networks (네트워크미디어: http://netmedia.kjist.ac.kr)