An Overview of Network-Aware Applications for Mobile Multimedia Delivery

Jinwei Cao
University of Arizona
jcao@cmi.arizona.edu

Dongsong Zhang
Univ. of Maryland, Baltimore County
zhangd@umbc.edu

Kevin M. McNeill
University of Arizona
kevin@ece.arizona.edu

Jay F. Nunamaker, Jr.
University of Arizona
jnunamaker@cmi.arizona.edu

Abstract

Network-aware Applications is a promising new concept in which applications are aware of network conditions and thus can adapt to the varying environment to achieve acceptable and predictable performance. This paper reviews the current research on network-aware applications, with a focus on their appliance on mobile multimedia applications. First, different frameworks or architectures of network-aware applications are introduced. Research issues and activities are then discussed in detail from two basic aspects of network-aware applications: network awareness and network adaptation. After the discussion about network-aware applications in general network environments, special problems and requirements of mobile multimedia applications are summarized, and different network-aware application approaches for mobile multimedia delivery are compared with respect to these requirements. Finally, we provide some suggestions to network-aware mobile multimedia application developers and identify current challenges in this area.

1. Introduction

Current computer networks and the Internet are becoming more and more heterogeneous. It is common that applications operate across different types of hosts, wired and wireless networks, with different resource availability [1]. Mobile multimedia application is a typical example of such applications that run in a heterogeneous network. For example, a PDA user can connect to an existing cellular network such as CDMA or a wireless LAN, and retrieve a Web page consisting of images and video from a remote server in a wired network. The interconnectivity of different networks makes pervasive computing an exciting reality, but it also poses many challenges for application developers. If an application is kept transparent to network changes, data will be generated and transmitted at a fixed rate, and there can be only two results. One is that the quality of data is reduced so that even a client with low bandwidth access can receive the data with little delay. The other is that the data is delivered in high quality so that the clients with high bandwidth access will experience satisfactory levels of performance. However, there will always be users who cannot be served at predictable and satisfactory levels of quality [3]. Therefore, in order to serve more users with different network capabilities, applications have to be able to adapt to changes in networks. This requirement is even more critical for mobile multimedia applications, because multimedia contents, especially audio and video, demand for a much higher peak bandwidth [35,36]. If an application does not change the data quality to be delivered according to network changes, a huge amount of multimedia data sent from a wired network will encounter unbearable delay or errors when transmitting in a wireless network with limited bandwidth [2, 38].

Network-aware applications are such applications that can deal with the problems described above. Described by Bolliger in [5], a network-aware application "attempts to adjust its resource demands in response to network performance variations". In most current network-aware applications, changes in network environments refer to changes in the following parameters of network quality [6,7]: *bandwidth,* which is the minimum link capacity among all the links from a source host to a destination host or throughput; *throughput* that measures the number of bytes of data transferred per second

experienced by a particular flow; end-to-end *packet loss* between hosts; *delay or latency,* which is the time taken for a message to be transmitted; and *jitter,* the variation in delay or response time. Awareness of dynamic context, such as the user's location or usage profile, may also be of interest, but it is beyond the scope of this paper.

According to Bolliger's description [5] and other related research [8-10], network-aware applications have two basic aspects: they must have the ability to monitor or get information from network monitors about the current status of the underlying network (**network *awareness***), and be able to adjust their behavior based on the collected information (**network *adaptation***). However, even though a lot of research has been done in these two fields individually, until recently researchers begin studying how to integrate "awareness" and "adaptation" to make applications more robust to network variations. In the following sections of this paper, we first review a few integrated frameworks proposed for network-aware applications, and then summarize relevant research in those two aspects. Because the concept of network-aware applications is particularly useful for mobile multimedia applications, we then discuss in details about the special problems, requirements, and solutions for network-aware mobile multimedia applications. Finally, we conclude with a few suggestions and challenges for network-aware mobile multimedia applications.

2. An overview of general network-aware applications

2.1. Frameworks of network-aware applications

As we mentioned in the introduction section, to our best knowledge, currently there are only a few studies that try to combine awareness and adaptation together, and most of them focus only on conceptual frameworks or architectures without full implementation [5,11,12].

A generic framework for developing network-aware applications is proposed by Bolliger in [5]. The core of this framework is a feedback closed-loop that controls adjustment of an application to network properties as shown in Figure 1. This feedback loop is designed as an adaptation layer sitting between the application layer and the lower layers in a common network model (e.g. TCP/IP). A *monitor and react* phase in this loop obtains information about the network status such as available bandwidth, decides which object needs to be adapted and how to adapt, and determines a QoS goal for the object. A *prepare* phase then applies a transformation strategy to the objects to realize the QoS goal as determined in the monitor and react phase. Finally, the prepared objects are transmitted in a *transmit* phase. In this three-phase feedback loop, the *monitor and react* phase is the key phase. To determine the required quality and the adaptation to be applied, an application/network QoS mapping scheme is also proposed. If the desired quality of the application is significantly different from the possible quality of the network, the sender must find objects to transform. However, the transformation algorithms are heavily application-dependent, and therefore cannot be specified in a general framework.

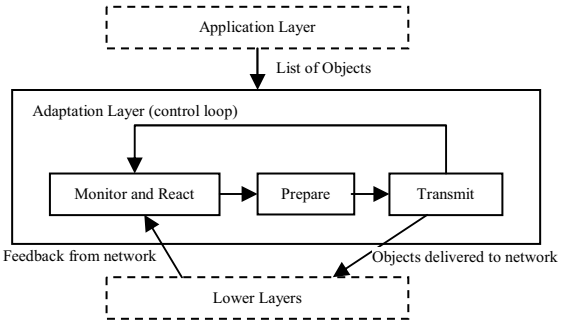

Figure 1. Feedback control loop [5]

Bolliger's framework provides a good abstraction of network-aware applications. It consists of some novel concepts like application/network QoS mapping. However, any application developer that wants to apply this framework still needs to specify the concrete functions used in each phase. This specification process will be domain and application dependent.

Other frameworks or architectures of network-aware applications include Odyssey [11], a platform for mobile data access, and Enable [12], a service to help applications with network tuning. The basic idea of Odyssey is similar to the framework proposed by Bolliger, but the latter is more general and abstract. Odyssey focuses on applications for mobile information access. In Odyssey, a client operating system is responsible for getting information about resource availability, while applications on the client side decide the adaptation policy. This approach requires some modifications to be done with the host's kernel, therefore it is not suitable for many hosts that are already connected to the Internet [13]. The other architecture, Enable service architecture, is restricted to TCP tuning (determining the optimal TCP parameters for a given network path). Therefore, it cannot be applied to many other situations such as multimedia applications in wireless environments.

2.2. Network awareness

To be aware of changes in a network environment, applications have to find a way to monitor the network. Usually network monitoring refers to collecting raw data about network status such as bandwidth and latency, but some systems also have the ability to transform the raw data statistically and present results based on application semantics [14]. The latter is similar to the concept of application/network QoS mapping proposed in [5] and bandwidth modeling in [9]. However, this presentation or mapping scheme is rarely used in current network monitoring systems because of its complexity.

The most common classification of existing network monitoring methods is based on the traffic generated by the method [8]. In active monitoring, network measurements are done by sending additional testing messages, and this will inevitably introduce extra traffic to the network. Passive monitoring techniques, in contrast, rely only on the traffic that applications generate as they communicate with other nodes in the network. The network status information is piggybacked on packets traversing the network, so it will not cause extra traffic in the network.

2.2.1. Active monitoring

Active monitoring can be done by both simple active probing services running from a single host and complex monitoring systems with probes distributed in the network. Examples of the former include standard ping [6] and bprobes [17]. Examples of the latter include NIMI [19], topology-d [20], and agent-based systems [14]. With the cost of extra traffic, active monitoring can have more control in the monitoring process. It can easily measure the characteristics of an entire network path between two hosts such as: packet round trip time (RTT), average packet loss, and available bandwidth, but is hard to get information about a single point in a network [21].

Extra traffic is always a problem for resource-limited networks such as wireless networks. In addition, another problem of active monitoring is that the results obtained by test packets generally do not match those of actual user packets [6]. Therefore, the monitoring results may not be an accurate representation of actual usages. However, by using test packets emulating actual user packets, active monitoring can measure the quality of a network everywhere, even for links with no traffic in. This is useful for connection initialization and server selection.

2.2.2. Passive monitoring

Passive monitoring is commonly used in current network aware applications, such as in [22]. Compared to active monitoring, passive monitoring can perform precise evaluations on a particular point in a network such as traffic/protocol mixes, accurate bit or packet rates, or packet timing/inter-arrival timing. Passive monitoring is more accurate than active monitoring because actual user packets are measured. However, fast processing is necessary to measuring all actual user packets securely [6]. And the reliance on passive observations can easily lead to information out-of-date, since information is only collected when a host contacts a remote site [23].

Examples of passive monitoring include: the SNMP (Simple Network Management Protocol); kernel-implementations of packet capture such as tcpdump/libpcap [25]; special hardware; software systems such as Nprobe [26] and SPAND [23]. A good software system design can reduce some disadvantages of passive monitoring. For example, in SPAND, passive measurements are collected and shared in local domains in order to eliminate redundant information so that the measurements can be kept up-to-date.

2.2.3. Hybrid monitoring

According to the above discussion, it is obvious that both active monitoring and passive monitoring have advantages and disadvantages. These two methods are more complementary other than exclusive of each other. Thus, a hybrid monitoring scheme is expected to be a better solution by combining the advantages of these two methods [27]. A hybrid monitoring system, EXPAND, has been proposed in [27]. It uses active probing only on demand when passive information is unavailable, such as when there are no active connections and a host wants to know the network conditions for setting up a new connection. An experiment of their prototype shows that the hybrid model can reduce the traffic introduced by active probing, while still getting accurate information.

Another network monitoring system that uses the hybrid principle is Remos [28], in which two types of data collection techniques are implemented: SNMP-based and the use of benchmarks. The SNMP-based passive measurement introduces little extra traffic into a network. For networks that do not support SNMP, a collector has to send explicit requests and use user-level benchmarks to measure the bandwidth and latency of the nodes in a pair-wise fashion. This active measurement will result in heavy usage of network resources, but can monitor more different types of network.

2.3. Network adaptation

2.3.1. What to adapt and how?

Many researchers have pointed out that adaptation can be realized by adjusting the quality of data to be delivered and the adaptation policy can be determined based on specific data types [3,11]. Usually, such adaptation is referred to *transcoding* or *transcaling*, because changing the quality of data is often to change its encoding scheme and parameters. In [3], data types are classified into text, image, audio, and video or image sequence, and specific encodings and *distillation axes* (the parameter that can be modified to change the quality of the data) of these data types are listed in a table. As listed in this table, for example, when the available bandwidth decreases, an image application can either transcode the image to be delivered from TIFF format to JPEG, or simply reduce the resolution or color depth of the TIFF image. This transformation strategy can solve the problem of resource limitation from the source. However, it requires applications to have enough knowledge about the low-level encoding and decoding techniques, and it is usually computational intensive.

Besides such data transformation in the application layer, adaptation can also be done at lower layers such as the transportation layer. These lower-layer adaptation techniques include using flow control techniques such as adjusting the TCP buffer size or RTP (Real-time Transport Protocol, a protocol run on top of UDP for multimedia information transport) buffer size [2] to reduce congestion and smooth jitter, and applying FEC (Forward Error Correction) when there are high error rates detected in the network [22,29,30]. There are also some other special approaches available, such as server selection [28], changing unicast to multicast [30], as well as partial caching and joint delivery of contents [31]. Adaptation in lower layers is less application dependent, therefore can be implemented in operating systems instead of in specific applications. It can also respond to network variations more quickly than adaptation in the application layer. However, it is less effective than the application-layer adaptation, since it cannot control the data rate from the source.

2.3.2. Where to adapt?

Bolliger describes three possible places for implementing adaptation in computer networks [3]. We summarize them in the following.

Receiver-initiated adaptation: In the Odyssey system [11] discussed in section 2.1, all the adaptation work is done by the client, which is the receiver of data from a remote data sender. Such a receiver-initiated adaptation strategy has a benefit of scalability, since servers do not need to be changed when there are new applications requesting for adaptation. Also it is more efficient because it understands the context of the object(s) to be adapted, such as the level of importance of an image in a Web page to be delivered. Therefore, receiver-initiated adaptation can make better *response time-quality* tradeoffs for complex applications [3]. However, although it is the receiver to decide when and how to adapt, the real adaptation work, such as transcoding, usually has to be done by the server or a proxy. Since the receiver cannot know the server's computation capability, it is often hard for them to coordinate and execute the adaptation policy.

Proxy-based adaptation: Many systems implement their adaptation functionalities separately in a proxy between the sender and the receiver. A big advantage of this approach is its transparent design. Neither the receiver nor the sender needs to be changed to support network adaptation. Therefore it is scalable and cost effective. One example of this approach is a self-adaptive distributed proxy system proposed in [22].

The cost of this transparency is that a proxy has little knowledge about the receiver and the sender, as well as the context of the object to be adapted. So compared with receiver or sender initiated approach, this approach is not very flexible since it typically can only provide a few static adaptation policies [3]. In addition, this approach must track the network conditions of both the server-to-proxy and proxy-to-server connections to make adaptation decisions. Although this will result in more accurate decisions, the required computation resource and time can affect the efficiency of this approach.

Sender-initiated adaptation: Fewer examples using this approach are found [53]. Although senders as content providers have full control on content quality and have more computational power to make reliable adjustment on content quality, sender-initiated adaptation is seriously lacking of scalability. Whenever there are new types of applications requesting adaptation, the server has to be changed to provide adaptation policies that are appropriate for the applications.

2.3.3. Agility vs. stability

The adaptation methods have to decide a tradeoff between agility and stability. It is desired that an application can react to changes in the environment quickly (*agility*). However, agility is usually achieved by sacrificing stability. When network conditions vary too quickly such as moving from a wired network to a wireless network, if an adaptation is taken too fast, the user of the application will experience sudden and

unpredictable large changes. If the network conditions continue to fluctuate rapidly, the application quality will appear to be very unstable to users since humans are intolerant of frequent, perceptually large changes [11, 32]. A proper tradeoff between agility and stability can be achieved by using a skepticism strategy [11] or using filters [33]. The former works as delaying the corresponding adaptation in quality until it is clear that changes in performance are persistent. The latter works as using filters to analyze network variations so that applications can react to persistent changes but tolerate transient ones. However, adding stability into adaptation is still an on-going research.

3. Network-aware applications for mobile multimedia delivery

Any network applications can be implemented as network-aware applications. However, some monitoring or adaptation techniques may be more suitable for some applications in certain networks than others (e.g. the SNMP-based approaches can only be used in wired networks). So the general approaches cannot provide enough details for application developers to help determine the right network-aware strategy. In this section, we give an in-depth discussion on problems, requirements and solutions for a special type of network-aware applications – network-aware mobile multimedia applications, because as we mentioned in the introduction section, network-awareness is very critical for mobile multimedia applications.

3.1. Problems of mobile multimedia applications

The number of mobile users is increasing every day around the world. Wireless networking technologies, as well as the widespread use of mobile devices such as PDAs, cell phones, and laptops, make pervasive computing a reality. Many applications such as email applications can now successfully run in mobile wireless networks. However, there are still many challenges in moving multimedia applications that deliver integrated contents in different formats – text, images, graphics, animations, voice and video – over a wireless network [34]. It is well-known that multimedia contents, especially audio and video, require a much higher network bandwidth. For example, video is hard to be transmitted in the second generation mobile networks (e.g. GSM) with bandwidth less than 28.8kbps [37], and is more realistic in a wireless LAN with bandwidth of 6 Mbps to 54 Mbps (802.11a) or a 3G mobile access network (e.g. UMTS) with bandwidth up to 2Mbps [2,38]. In addition to limited resource in wireless networks, both wireless and mobility features cause troubles to network quality, such as varying bandwidth, variable bit error rate, possibly asymmetric connectivity, and unexpected quality degrade during handoff [39]. Generally, we can group the problems in mobile multimedia applications based on three major causes of these problems: wireless, mobility, and multimedia. They are summarized in Table 1, drawn from [40-44].

Considering such a heterogeneous and varying network environment, as well as the multimedia applications' reliance on network resources, it is really critical to enable mobile multimedia applications to be network-aware. The problems with wireless, mobility, and multimedia also pose some special requirements for network awareness and adaptation.

3.2. Network awareness for mobile multimedia applications

What to measure: In [39], bandwidth, cost, delay bounds, and security factors are proposed as parameters for making adaptation decisions in mobile network applications. Among these four parameters, bandwidth is the most important measure and is usually monitored in any type of applications. Cost and security factors are rarely mentioned in other literatures about mobile network applications, and it is difficult to measure them too. Delay bound is another important measure, especially for mobile multimedia applications, since streaming media is very sensitive to latency. Besides bandwidth and latency, error rate is also a very important measure for mobile multimedia applications because multimedia compression is very sensitive to errors. As a result, in order to achieve network-awareness for mobile multimedia applications, at least three measures of networks, namely bandwidth, latency and error rates, need to be monitored. To precisely represent these network quality measures in applications, a standard measure of application-level data quality for mobile multimedia applications is also required, as well as a mapping scheme between this application measure and the network measures. For example, in the Odyssey system, "fidelity", which is the degree to which a data item used by a mobile client matches a reference copy, is used as a measure of data quality. It is mapped to the available bandwidth of a network [11].

How to measure: Since network conditions may change very quickly in wireless networks, we need network monitoring methods for mobile multimedia applications to be able to detect changes as fast as possible. Active monitoring, in addition to competing with applications for scarce bandwidth in wireless networks, causes large delay to get results, therefore is not proper for agile network monitoring [27]. On the other hand, passive monitoring is not directly valid in

wireless environments too, because passive monitoring relies on the data load passing a host and is hard to measure other factors such as error rate caused by impairments. It also cannot measure an unopened connection or failed connections, which are very common in wireless environments [27]. As stated in [39], both methods are useful for some situations and having one should not preclude the other. Therefore, although most of the current prototypes of network-aware applications are using passive monitoring [22,45], we believe that a hybrid approach will be the best choice for network monitoring in mobile multimedia applications. An example can be found in [27].

Table 1. Problems of mobile multimedia applications

Cause	Description	Problem
Wireless Connection	Atmospheric conditions, physical obstacles, or impairments at the physical layer, such as multipath fading or inference	• Less bandwidth, higher transmission latency • Time-varying error characteristics (high packet loss and low reliability • Time-varying channel capacity
Mobility	Users' movement can change the distance between the base station and the mobile host, or cause wireless-wireless handoff (radio resource reuse) and wireline-wireless handoff	• Rapid and radical changes in available resource capacity such as bandwidth • Frequent topology changes • Unexpected delays, packet losses, or completely loss of service.
Multimedia	Media compression methods are sensitive to errors	• The greater the compression and the higher error rate, the more severe the visual disruptions. • Error correction to reduce errors will introduce delay in the network
	Streaming, as a technique to provide real time multimedia content delivery, is resource-hungry, connection-oriented, and sensitive to latency in the network	• "Unfair share" of the bandwidth • Vulnerable to breaks in transmission, such as connection loss during handoff

Besides these general requirements on monitoring methods, there are also some special requirements for mobile multimedia applications. For example, streaming media applications require timely delivery of data and use protocols like RTP [46]. Therefore, many TCP-based network monitoring techniques (e.g. [12]) cannot be applied to streaming media applications. The RTP Control Protocol (RTCP) can be used to get feedback including packet loss and jitter information from the receivers of an RTP media stream [47]. However, packet loss or jitter can be found in the feedbacks from a wireless network, caused by either congestion or error in the radio links. If the sender cannot distinguish congestion from error, it may apply inappropriate adaptation methods and the problem will remain. A proxy between wired and wireless networks can be used to solve this problem by splitting an RTP connection [47]. Another solution is to completely measure an end-to-end communication link piecewise [4], i.e., consider the wired and wireless parts separately and decide the end-to-end characteristics by combining the piecewise characteristics. For protocols other than RTP, formulas for calculating the transport error rate and available UDP throughput in 3G wireless networks are given in [48]. An mmdump tool [49] can also be used for monitoring multimedia traffic on the Internet controlled by H.323, RTSP and similar multimedia session control protocols.

Finally, network monitoring methods must be able to detect movements beyond the range of a wireless base station or changes of network types (e.g. from a wireless network to a wired network), for supporting adaptation during handoff. In [42], "heart beats" between a mobile host and an adjacent "anchor" host are used to validate link layer connectivity. It may be beneficial to incorporate this approach into current network monitoring methods for mobile multimedia applications.

3.3. Adaptation for mobile multimedia applications

Several key requirements of adaptation for mobile multimedia applications are described in [22]: automated data format adaptation without user intervention, graceful quality degradation, seamless handoffs across networks during roaming, and high QoS with low jitter, delay, and guaranteed bandwidth. Also applications need to be able to react to any path failures with recovery mechanisms. To fulfill these requirements, a lot of different approaches can be applied and the adaptation for mobile multimedia

applications continues being a hot research topic. We summarize several adaptation approaches as follows:

Multiple encoding: For the same image or the same audio or video stream, servers can store multiple copies with different encoding methods or parameters, and choose an appropriate copy based on network conditions. This is the easiest way of adaptation for multimedia applications. However, the huge storage requirements and the fixed adaptation capability make it the last choice for adaptation in mobile multimedia applications.

Transcoding: It refers to the mapping of a non-scalable stream to another non-scalable stream with a different compression rate (e.g. MPEG to H.263). Transcoding avoids the necessity of storing multiple files corresponding to various bit rates. However, this approach requires decoding and recoding of each media stream. Therefore it is very computation-intensive and requires a deep understanding of different encoding and decoding algorithms.

Layered encoding (transcaling and multicasting): In current research, this is the most popular way to provide adaptation for multimedia applications [29,30,38,42]. This approach usually generates a base-layer and one or more enhancement layers to cover the desired bandwidth range. By applying multicast [30], no separate delivery of streams is needed. Layers can be added or dropped by joining or leaving a multicast group. For example, if congestion is detected, the highest layer will be dropped. Transcaling, on the other hand, is a generalization of transcoding. It derives one or more scalable streams covering different bandwidth ranges from another scalable stream [38]. A proxy-based solution requires minimal changes to the base station and the mobile client, and is the common implementation technique for this layered encoding approach [50].

To avoid users' intervention, applications also need some intelligence to automatically choose the encoding algorithm(s) best suited to their media type and network type [42]. For example, encoding algorithms used in a wireless network should enable both high functionality and low complexity, therefore algorithms for wireless video encoding place a lower emphasis on motion compensation and a higher emphasis on intra-frame coding [42].

This layered encoding approach requires less computation than pure transcoding. However, it still needs intensive knowledge of media compression and requires layered encoding in the first place. For some legacy systems using non-layered encoding, there must be a transcoding step at first.

Rate shaping: An application can also adjust the traffic rate generated by encoders through changing the following media encoder parameters: image resolution, video frame rate, quantization parameter, and movement detection threshold. This approach will not change the data encoding format, but can still adjust the quality by changing the compression rates [42].

In addition to the above application-level adaptation, there are also some adjustments that can be done in the lower layers for mobile multimedia applications. We discuss them as follows.

Smoothing: A streaming media application can smooth the variations in the available bandwidth by adjusting the size of the receiver buffer and the initial buffer delay, which refers to the period which incoming data are pre-buffered before the actual playback starts [46]. A good adjustment of receiver buffer and initial buffer delay can greatly alleviate the problem of jitter and delay.

Error correction: Retransmission is a common way to minimize packet losses caused by radio link errors, but it is not proper for real-time applications involving media streaming [47]. Therefore, to deal with radio link errors, applications have to make their media streams more robust against packet losses by changing the frequency of transmitting FEC packets, or changing the media encoding parameters [47]. In [48], hybrid delay-constrained ARQ and FEC are used for base layer error protection, and UEP (Unequal Error Protection) is used to protect different enhance layers.

It is also possible for mobile multimedia applications to encounter errors from inter-stream interference between streams because of limited link resources. To prevent these types of error, MobiWeb [50] allows only one stream at a time to be adapted until no further adaptation is necessary. An admission control algorithm and a dynamic prioritization scheme are used to select a stream to adapt and guarantee the minimum quality in the stream's performance.

Handoff control: Handling handoffs between different networks or different cells in cellular networks is a special problem of adaptation for mobile network applications. Generally the necessary handoff control techniques include reconfiguration and dynamic control over the topology to instantly set up a virtual topology [51]. Mobile agents can also be used for rerouting a mobile device's flow bundle from an old access point to a new one [40]. In [41], an anticipatory handoff control strategy is used and rerouting is avoided during handoff. By establishing branch connections to the neighborhood of a mobile host in advance, a handoff is completed by allocating resources to an appropriate branch connection and grafting it into the original connection. Handoff latency is therefore shortened by avoiding rerouting.

About the problem of where to apply adaptation, both proxy-based and sender-initiated strategies are fine for mobile multimedia applications. However, the receiver-initiated strategy may not be a good choice, since mobile devices such as PDAs usually have very limited power and computing resources. Utilizing adaptation algorithms in such devices may cause computational delay.

Finally, considering agility and stability, in the mobile wireless networking environments, a more conservative adaptation policy is desired since it can lead to a more stable operation [40,50]. As stated in [32], users prefer lower but stable QoS rather than higher but varied and unpredictable QoS. In wireless networks, variations usually take place very fast and adaptation should smooth the variation to some extent to avoid changing and unpredictable application performance.

3.4. Available frameworks or systems of mobile multimedia applications

Besides Odyssey [11] that we discussed in section 2.1, there are some other frameworks or systems specially proposed for mobile multimedia applications. As listed below, these systems try to incorporate both network awareness and network adaptation into their design, although most of them are still in conceptual or prototyping stage.

A self-adaptive distributed proxy system that provides streaming multimedia service to mobile wireless clients is proposed in [22]. Passive monitoring techniques are used to measure real-time network variations. The adaptation techniques include transcoding, dynamic relocation of transcoders, and automatic insertion of FEC and compression into the data transcoding path. A prototype is developed as a streaming video playback involving a series of transcoding proxies and a mobile client. The distributed proxy design allows different adaptation techniques used in the same system, thus making the system very flexible and scalable. However, the complexity for implementation also increases due to this design.

In [30], an adaptive framework has been developed to provide adaptive video transportation over broad-band wireless networks. This framework consists of three components: scalable video coding, network aware adaptation of end systems, and adaptive QoS support from networks. The focus of their design is the adaptation techniques such as layered encoding, multicasting, and error control for video applications. Bandwidth managers are maintained in base stations in wireless networks for monitoring network bandwidth variations. Although this framework is designed for video applications over wireless IP networks, it can also be extended to other multimedia applications in more heterogeneous network environments.

An architectural framework is proposed in [52] for designing middleware platforms to support network-aware mobile multimedia applications based on an extended CORBA computational model. This architecture is mostly built from a software engineering point of view. Similar effort can also be found in Mobiware (http://comet.columbia.edu/mobiware), which is a mobile middleware toolkit that enables mobile multimedia applications to adapt to time-varying mobile network conditions. The Mobiware toolkit is also software intensive and is built on CORBA and Java distributed object technology. Mobiware provides a set of open programmable interfaces and algorithms for adaptive mobile networking, and it can be implemented on mobile devices, wireless access points, as well as mobile capable switch/routers [40].

4. Conclusions

In this paper, we have highlighted research studies about network-aware applications, especially network-aware mobile multimedia applications. As a result of this review, we show that a feedback loop concept is critical to any network-aware applications. Although this loop may not be the same as described in [5], it must have two critical components. One is for monitoring networks and obtaining information of changes in networks. The other is to select the right object to adapt and select the adaptation methods appropriate to the applications based on the feedback from networks. Although network monitoring can be independent of applications, adaptation approaches are usually highly dependent on applications. For mobile multimedia applications, wireless features, mobility, and multimedia content all pose challenges to both network awareness and network adaptation. We consider the following issues to be important in developing network-aware mobile multimedia applications, not only in existing work described in literature but also for future development.

- How to combine active monitoring and passive monitoring in a hybrid approach to obtain fast and accurate feedback from mixed wired-wireless networks?
- What network quality measures are most important for making adaptation decisions? How to map the application layer quality measures to the lower layer quality measures?
- How to tradeoff the computational complexity of media transcoding or transcaling and other adaptation techniques with the benefit of adaptation

for mobile multimedia applications? A utility function (benefit/cost comparison) may be useful in choosing the right adaptation techniques.

- How to obtain a smooth adaptation in mobile multimedia applications during network handoffs caused by user mobility?
- Finally, how to implement monitoring and adaptation functions in wireless networks?

Although some questions have been addressed in some studies reviewed in this paper, most of them are still in the conceptual or prototyping stage. Further research is expected to answer all of the questions in the future.

5. References

[1] C. Hesselman and H. Eertink, "A Scalable QoS Adaptation Service for Mobile Multimedia Applications," presented at Proceedings of the 6th EUNICE Open European Summerschool (EUNICE2000), Enschede, The Netherlands, 2000.

[2] J. Kim and A. Jamalipour, "Traffic Management and QoS Provisioning in Future Wireless IP Networks," *IEEE Personal Communication*, pp. 46-55, 2001.

[3] J. Bolliger, "A framework for network-aware applications,". ETH Zurich: Institute of Computer Systems, 2000.

[4] L. Cheng and I. Marsic., "Piecewise Network Awareness Service for Wireless/Mobile Pervasive Computing," *Mobile Networks and Applications (MONET)*, vol. 7, pp. 269-278, 2002.

[5] J. Bolliger and T. Gross, "A Framework-Based Approach to the Development of Network-Aware Applications," *IEEE Transactions on Software Engineering*, vol. 24, 1998.

[6] M. Toshiya, N. Kazuo, M. Shouji, and M. Hiroyuki, "QOS MONITORING SYSTEM," Yokogawa, Technical Report No.34, 2002.

[7] T. Hou, Y. Dong, and Z.-L. Zhang, "Network Performance Measurement and Analysis -- Part 1: A Server-Based Measurement Infrastructure," Fujitsu Laboratories of American, Technical Report FLA-NCRTM98-01, July 6th 1998.

[8] W. Caripe, G. Cybenko, K. Moizumi, and R. Gray, "Network awareness and mobile agent systems," *IEEE Communications Magazine*, vol. 36, pp. 44 -49, 1998.

[9] J. Bolliger, T. Gross, and U. Hengarmer, "Bandwidth Modelling for Network-aware Applications," presented at Proceedings of ACM INFOCOM '99, New York, 1999.

[10] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Scalable video coding and transport over broadband wireless networks," *Proceedings of the IEEE*, vol. 89, pp. 6 -20, 2001.

[11] B. Noble, "System support for mobile, adaptive applications," *IEEE Personal Communications Magazine*, vol. 7, pp. 44 -49, 2000.

[12] B. L. Tierney, D. Gunter, J. Lee, M. Stoufer, and J. B. Evans, "Enabling Network-Aware Applications," presented at 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10'01), San Francisco, California, 2001.

[13] A. Al-bar and I. Wakeman, "A survey of adaptive applications in mobile computing," presented at International Conference on Distributed Computing Systems Workshop, 2001.

[14] Y. I. Wijata, D. Niehaus, and V. S. Frost, "A scalable agent-based network measurement infrastructure," *IEEE Communications Magazine*, vol. 38, pp. 174 -183, 2000.

[17] R. L. Carter and M. E. Crovella, "Measuring bottleneck-link speed in packet switched networks," Computer Science Department, Boston University, Technical Report BU-CS-96-006, March 1996.

[19] V. Paxson, A. Adams, and M. Mathis, "Experiences with NIMI," presented at Proceedings of the Passive & Active Measurement Workshop, 2000.

[20] K. Obraczka and G. Gheorghiu, "The performance of a service for network-aware applications," presented at Proceedings of the SIGMETRICS symposium on Parallel and distributed tools, Welches, Oregon, United States, 1998.

[21] J. Curtis and T. McGregor, "Review of Bandwidth Estimation Techniques," presented at New Zealand Computer Science Research Students' Conference, University of Canterbury, New Zealand, 2001.

[22] Z. M. Mao, H. W. So, B. Kang, and R. H. Katz, "Network Support for Mobile Multimedia using a Self-adaptive Distributed Proxy," presented at 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV-2001), 2001.

[23] S. Seshan, M. Stemm, and R. H. Katz., "SPAND: shared Passive Network Performance Discovery," presented at 1st Usenix Symposium on Internet Technologies and Systems (USITS '97), Monterey, CA, 1997.

[25] ""tcpdump/libpcap,"," *http://www.tcpdump.org/*, 2001.

[26] A. Moore, J. Hall, E. Harris, C. Kreibech, and I. Pratt, "Architecture of a Network Monitor," presented at Proceedings of the Fourth Passive and Active Measurement Workshop (PAM 2003), 2003.

[27] B. Landfeldt, P. Sookavatana, and A. Seneviratne, "The Case for a Hybrid Passive/Active Network Monitoring Scheme in the Wireless Internet," presented at ICON 2000, Singapore, 2000.

[28] N. Miller and P. Steenkiste, "Collecting Network Status Information for Network-Aware Applications," presented at Proceedings of IEEE Infocom 2000, 2000.

[29] C.-S. Wu, G.-K. Ma, and B.-S. P. Lin, "Personal mobile multimedia communications in a wireless WAN environment," presented at IEEE First Workshop on Multimedia Signal Processing, 1997.

[30] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Scalable video transport over wireless IP networks," presented at The 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2000), 2000.

[31] S. Jin and A. Bestavros, "Accelerating Internet Streaming Media Delivery using Network-Aware Partial Caches," Computer Science Department, Boston University, Technical Report BUCS-TR-2001-023, October 2001.

[32] A. Bouch and M. A. Sasse, "Network Quality of Service: What do users need?," presented at Proceedings of

the 4th International Distributed Conference (IDC'99), Madrid, Spain, 1999.

[33] M. Kim and B. D. Noble, "SANE: stable agile network estimation," University of Michigan, Department of Electrical Engineering and Computer Science, Ann Arbor, MI, Technical Report CSE-TR-432-00, August 2000.

[34] J. Arreymbi and M. Dastbaz, "Issues in delivering multimedia content to mobile devices," presented at Sixth International Conference on Information Visualisation, 2002.

[35] A. Krikells, "Mobile multimedia considerations," *IEEE Concurrency*, vol. 7, pp. 85 -87, 1999.

[36] A. Krikelis, "Considerations for a new generation of mobile multimedia communication systems," *IEEE Concurrency*, vol. 8, pp. 80 -82, 2000.

[37] M. Sawada, N. Tani, M. Miki, and Y. Maruyama, "Advanced mobile multimedia services and applied network techniques," presented at IEEE 1998 International Conference on Universal Personal Communications (ICUPC '98), 1998.

[38] H. Radha, "TranScaling: a video coding and multicasting framework for wireless IP multimedia services," presented at Proceedings of the 4th ACM international workshop on Wireless mobile multimedia, Rome, Italy, 2001.

[39] C. E. Perkins, "Mobile networking in the Internet," *Mobile Networks and Applications*, vol. 3, pp. 319 - 334, 1999.

[40] O. Angin, A. T. Campbell, M. E. Kounavis, and R. R.-F. Liao, "The mobiware toolkit: programmable support for adaptive mobile networking," *IEEE Personal Communications Magazine*, vol. 5, pp. 32 -43, 1998.

[41] K. Lee, "Adaptive network support for mobile multimedia," presented at Proceedings of the first annual international conference on Mobile computing and networking, Berkeley, California, United States, 1995.

[42] J. Inouye, S. Cen, C. Pu, and J. Walpole, "System support for mobile multimedia applications," presented at Proceedings of the IEEE 7th International Workshop on Network and Operating System Support for Digital Audio and Video, 1997.

[43] R. Alonso, Y.-L. Chang, L. Iftode, and V. S. Mani, "Managing video data in a mobile environment," *ACM SIGMOD Record*, vol. 24, pp. 28 - 33, 1995.

[44] A. Stirling, "Mobile multimedia platforms," presented at 52nd IEEE Vehicular Technology Conference (VTC 2000), 2000.

[45] R. Han, P. Bhagwat, R. LaMaire, T. Mummert, V. Perret, and J. Rubas, "Dynamic Adaptation in an Image Transcoding Proxy for Mobile Web Browsing," *IEEE Personal Communications Magazine*, 1998.

[46] L. Huang, U. Horn, F. Hartung, and M. Kampmann, "Proxy-based TCP-friendly streaming over mobile networks," presented at Proceedings of the 5th ACM international workshop on Wireless mobile multimedia, Atlanta, Georgia, USA, 2002.

[47] T. Yoshimura, T. Ohya, T. Kawahara, and M. Etoh, "Rate and Robustness Control with RTP Monitoring Agent for Mobile Multimedia Streaming," presented at Proceedings of IEEE International Conference on Communications (ICC) 2002, 2002.

[48] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "Network-Adaptive Scalable Video Streaming over 3G Wireless Network," presented at special session on Video Communication over 3G and Beyond, IEEE International Conference on Image Processing(ICIP'01), Greece, 2001.

[49] R. Caceres, N. Duffield, A. Feldmann, J. Friedmann, A. Greenberg, R. Greer, T. Johnson, C. Kalmanek, B. Krishnamurthy, D. Lavelle, P. Mishra, K. K. Ramakrishnan, J. Rexford, F. True, and J. v. d. Merwe, "Measurement and analysis of IP network usage and behavior," *IEEE Communications Magazine*, pp. 144-151, 2000.

[50] M. Margaritidis and G. C. Polyzos, "MobiWeb: Enabling Adaptive Continuous Media Applications over Wireless Links," presented at IEEE International Conference on Third Generation Wireless Communications, Silicon Valley, San Francisco, California, 2000.

[51] A. Alwan, R. Bagrodia, N. Bambos, M. Gerla, L. Kleinrock, J. Short, and J. Villasenor, "Adaptive mobile multimedia networks," *IEEE Personal Communications Magazine*, vol. 3, pp. 34 -51, 1996.

[52] G. S. Blair, G. Coulson, N. Davies, P. Robin, and T. Fitzpatrick, "Adaptive middleware for mobile multimedia applications," presented at Proceedings of the IEEE 7th International Workshop on Network and Operating System Support for Digital Audio and Video, 1997.

[53] R. Rejaie, M. Handley, and D. Estrin, "RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet," presented at Proc. IEEE INFOCOM, New York, NY, 1999.