Context-Aware, QoE-Driven Adaptation of Multimedia Services

Karthik Srinivasan[1], Poorva Agrawal[1], Rajat Arya[1], Nadeem Akhtar[2], Deepak Pengoria[2], and Timothy A. Gonsalves[3]

[1] Infosys Limited, Bangalore, India
{s_karthik,poorva_agrawal,rajat_arya}@infosys.com
[2] Centre of Excellence in Wireless Technology, Chennai, India
{nadeem,deepak}@cewit.org.in
[3] Indian Institute of Technology Mandi, Himachal, India
tag@iitmandi.ac.in

Abstract. Delivery of multimedia services over heterogeneous wireless networks is a challenging proposition because of the diverse characteristics of the underlying wireless technologies. The problem is compounded further by the availability of a wide range of end user devices such as desktops, notebooks, tablets and smartphones. To provide a consistent and uniform Quality of Experience to the end user, we propose a context-aware service adaptation framework in this paper. The basic idea is to monitor the user and network context and leverage this information for adapting services to match the device and network characteristics.

Keywords: Multimedia services, service delivery, service adaptation, QoE, context-awareness, heterogeneous network.

1 Introduction

The fast-changing telecommunications market is characterized by increasing heterogeneity of access technologies and devices. On the access side, we see a plethora of wireless technologies like 3G, 4G and WiFi etc. Despite increasing convergence between these networks, there exist differences in terms of network architecture, QoS mechanisms, supported data rates and so on. On the end-user side, there is great diversity in the hardware and software capabilities of these devices. The challenge is to provide access to multimedia services to heterogeneous devices over heterogeneous access networks, with consistent and uniform Quality of Experience (QoE) to the end user, while utilising the available network resources efficiently.

QoE is generally considered as a subjective measure of a customer's perception of the performance of a network and the services it offers (web browsing, phone call, TV broadcast etc.). While QoS refers to the performance in terms of metrics such as packet loss, delay and jitter etc, QoE relates to the overall user experience while accessing and using the provided services. For instance, a user with a low-bandwidth connection may be satisfied by a low/moderate quality video stream

C. Borcea et al. (Eds.): MobilWare 2012, LNICST 65, pp. 236–249, 2013.

whereas someone who is paying a premium for a high-bandwidth connection will be satisfied only if the video is of a high quality. In other words, QoE is influenced greatly by the context.

1.1 Background

We first present a set of use cases, using video as an example, to show the need for context aware adaptation to improve QoE.

- Case 1: Consider a user with a subscription for 3G data connection, with volume-based charging. Assume that the user has multiple 3G-capable devices with different screen sizes and resolutions. When the user accesses YouTube videos over a low-resolution device, to push a high-resolution video wastes valuable billable bandwidth.
- Case 2: Consider a user with a device that has a high-res display and that is capable of using both 3G and WiFi. Typically, while streaming a video, the user will choose the access technology that provides a higher data rate. However, if the residual battery power is low, the user may switch to a lower-bandwidth connection and opt for a low-res version of the video.
- Case 3: Consider an example where a user has a 3G subscription that costs more while roaming. To reduce the cost of streaming a video, the user may choose a low-res stream even though sufficient bandwidth is available and the device is capable of displaying high-res videos.

In all the three examples mentioned above, the choice of video quality was determined by multiple factors such as cost, screen resolution, available bandwidth and battery power. For the end user, the acceptable QoE also varies in consonance with these factors. Bearing this in mind, we propose a framework for service adaptation that takes into account several types of context, to provide the best possible QoE to the user.

1.2 Related Work

There are two aspects to service adaptation: i) feedback and trigger(s) for adaptation and ii) the actual modification of one or more characteristics of the multimedia stream. To achieve consistent QoE for video services, adaptive streaming techniques are used such that variations in resource availability can be matched by appropriate adaptation of the video stream. Such adaptation is realized using techniques such as caching [1] [2] [3], rate-adaptation, and buffer tuning. Rate adaptation techniques mainly employ three methods: 1) transcoding; 2) scalable encoding and 3) stream-switching. Transcoding [4] uses on-the-fly encoding of the raw content for varying the bit-rate of the stream to match the resource constraints. Scalable encoding relies upon scalable codecs for manipulating the streaming rate without re-encoding [5]. Stream-switching requires that multiple copies of the content is encoded at different rates and stored at the source [6]. The streaming server chooses the appropriately encoded stream depending on network quality and/or user feedback. There are many commercial implementations of adaptive streaming such as Adobe Dynamic HTTP Streaming, Apple HTTP Adaptive Streaming, and Microsoft Smooth Streaming.

Research in rate-adaptation has focused on quality feedback, determining resource availability and the adaptation logic. The adaptation approaches described in [7] [8] [9] focus primarily on matching the rate to available bandwidth. In [10], context-aware content adaptation for mobile devices is proposed. Context here refers to profile repositories, user preferences and real-time information such as network speed, connection protocol etc. The emphasis here is on one-time adaptation of content. In [11], the authors present an approach for content adaptation procedure for web-based mobile services by utilizing device capability databases and generic page transformation. Here, the focus is on web browsers and only device characteristics are considered. [12] takes a more holistic view of context and proposes a multi-dimensional adaptation framework. However, the approach is somewhat static since the decision on adaptation is taken at the beginning. Context-aware adaptation for ubiquitous web access is described in [13]. This uses a middleware for context-awareness along with an intermediary based architecture for content adaptation. The focus here is on device-centric adaptation of Web content. [14] proposes a context aware and resources aware dynamic service adaptation approach for a pervasive computing system device. The adaptation is applied to computing services in this case.

As the preceding discussion indicates, significant research efforts have been undertaken on service adaptation. However, the proposed methods are restrictive in terms of the type of context used, the type of service for adaptation that is supported and the type of adaptation. Furthermore, the notion of QoE is rarely mentioned in the context of service adaptation.

1.3 Our Contribution

We propose an elaborate context-aware framework for dynamic adaptation of services (particularly multimedia services). In particular, we focus on three aspects: a) multi-dimensional contexts b) model driven adaptation and c) operation under constraints to optimize end user QoE.

The rest of this paper is organized as follows. In Section 2, we present a model for analyzing services and evolving guidelines for context aware adaptations. Section 3 describes our framework to enable effective implementation of the model and Section 4 applies the same for multimedia services. In Section 5, we present details of a Proof-of-Concept implementation of the proposed framework. We conclude in Section 6, with pointers to future work.

2 A Model for Analyzing Context Aware Service Adaptation

Our key objective is to deliver optimal QoE to users under varying runtime conditions. As discussed in section 1, this can be achieved by dynamic adaptation of the service based on the current context. Towards this, we propose a model driven approach to analyze a service and evolve comprehensive guidelines for service adaptation under different constraints. We first present the key components of the model and then describe the approach for analyzing the service using the same.

The model comprises of two key components entities and functions. Entities describe the environment associated with the delivery of a service, along with the constraints under which service needs to be delivered. Functions define how the different entities can be combined for effective adaptation. Entities associated with a service are as follows:

a. *Domains:* Represent the participants/actors involved in the delivery of a service.
b. *Contexts:* Collection of information that together provides a comprehensive view of a domain. Information can be static and/or dynamic (varying during the course of service delivery).
c. *Actions:* Activities that can be executed by a domain that leads to appropriate adaptation of the service.

Figure 1 depicts the relation between the above defined entities as well as a sample representation for multimedia content delivery.

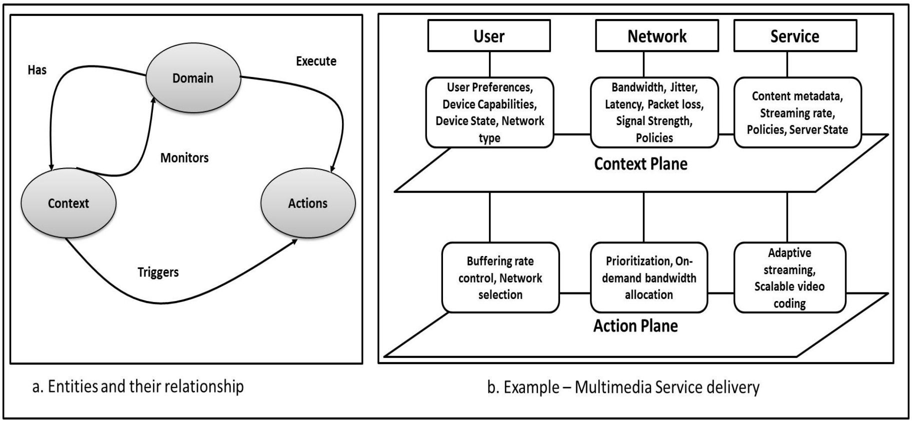

Fig. 1. Entities: Relationships and Example

Functions can be classified into those that provide information and those that process the same for effecting adaptation decisions.

a. Monitoring functions are responsible for continuously gathering information associated with the different contexts. A single function can monitor one or more contexts and provide specialized features like aggregation and selective notification.

b. Decision functions are responsible for detecting when a service needs to be adapted and recommending the appropriate action(s). Decision functions embed within themselves, the mapping between contexts and their impact on QoE as well as the possible sets of actions that can help in delivering the best possible end-user experience. They can also act as constraint enforcers, limiting the set of adaptations that are possible in a given instance based on current context.

We propose a 2-step approach, to arrive at the guidelines for adaptation. This is depicted in Figure 2 and the activities to be performed at each step are described below.

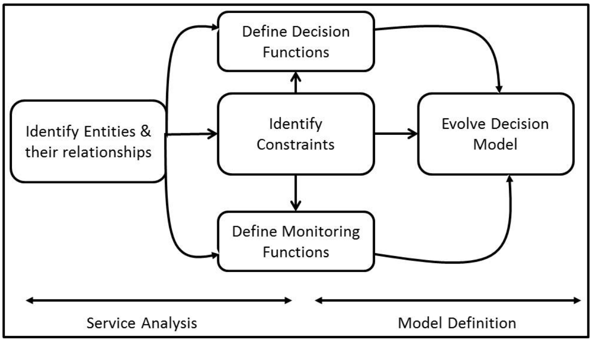

Fig. 2. Service analysis and model definition

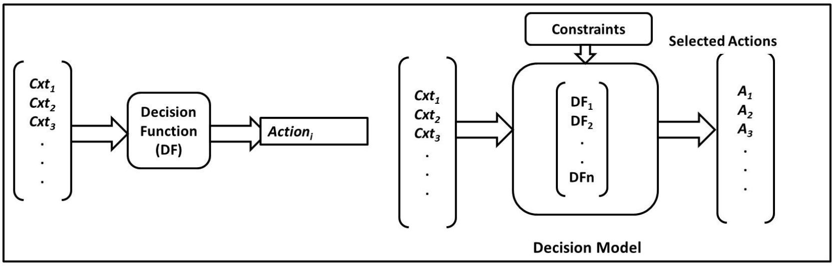

Fig. 3. Decision functions

a. *Service Analysis:* This involves identification of relevant entities and their relationships. These include domains participating in the service delivery, their associated contexts and actions. Also, for each action, the contexts whose variations can trigger the execution of the action are identified.
b. *Model Definition:* We propose representation of adaptation guidelines as decision models. A decision model incorporates within itself, a multitude of decision functions and intelligence about the triggers for execution of different decision function sequences. Figure 3 depicts the operation of a decision

model. Triggers to the model are typically changes in context. The model specifies the decision function sequences to be executed for a given trigger. Decision models offer the flexibility of dynamic addition/removal of contexts from the model based on runtime constraints. The exact decision model depends on the monitoring and decision functions available for a given set of constraints.

3 An Implementation Framework for Context Aware Service Adaptation

In Section 2, we described a model for analyzing services and arriving at guidelines for service adaptation. Real world implementation of such a model would need to take into account different types of constraints including those related to contexts to be monitored (what and how) and the adaptation actions to be performed (what and who). Constraints can also include restrictions around domains accessible at runtime.As an example, we list below a few constraints or opportunities related to delivering multimedia content across different networks and devices.

- Service provider not having information about the state of mobile device and the last mile network through which the device is accessing the content.
- End users should be able to specify their preferences for the network(s) to be used while viewing specific type(s) of multimedia content.
- Where the end user device and the service provider infrastructure can collaborate, it should be possible to exploit the same to improve QoE.

From the above, it is clear that an effective adaptation solution should provide a high degree of flexibility when it comes to executing the different aspects of the model. We propose a framework with the following properties to address the requirements and constraints discussed so far.

- The framework consists of a set of loosely coupled components, each of them highly configurable and capable of independent operation.
- Interactions between components are event driven.
- End-to-end service adaptation is achieved by deploying instances of the framework across one or more participating domains.
- Each instance of the framework can be configured to a high level of granularity.

Figure 4 depicts our service adaptation framework. The different framework components are described below.

a. *Control Unit:* This is the brain of the framework. It is responsible for setting up and managing a framework instance. Key functions include, activation/deactivation of components, configuration of different components tuned to a specific implementation, monitoring the state of different components and enabling run-time control/re-configuration of individual components. It has direct access to all components in the framework via services published by each of the components.

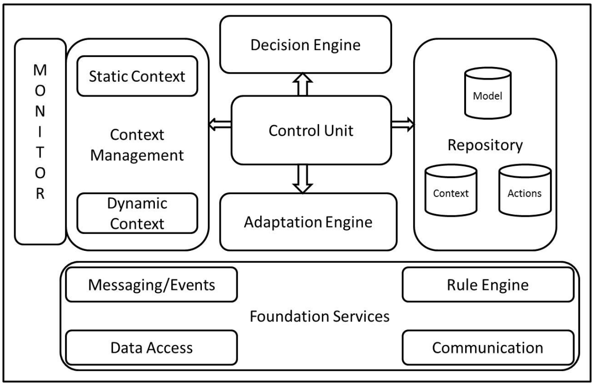

Fig. 4. Service Adaptation Framework

b. *Foundation Services:* The framework includes a set of common services that is required for effective functioning of the different components. These include a messaging service to enable exchange of information across components, a rule engine to support execution of decision functions, a uniform data access service to retrieve persistent information related to different components and a communication service to support information exchange between different framework instances running across domains.

c. *Context Manager & Monitor:* Contexts can be static (values are fixed) or dynamic (values can change at run-time). The context manager provides services for accessing static context and generates events notifying changes in the values of dynamic contexts. It supports switching on/off monitoring of specific contexts, selection of monitoring function for a specific context as well as controlling how monitored context needs to be reported. Reporting options include support for selective (based on pre-set conditions) reporting of context values.

d. *Decision Engine:* This component is responsible for loading and executing decision models. Decision models and their associated functions are represented as rule sets that can be dynamically executed based on received context change triggers.When an adaptation trigger is detected, this component will generate an event specifying the list of actions to be executed. Actions can include any of the following.
   - Adaptations that can be performed in the current domain.
   - Adaptations that need to be performed in a different domain.
   - Information that needs to be sent to another domain (where the decision model is being executed in a distributed fashion).

e. *Adaptation Engine:* The execution of actions selected by the decision engine is managed by this component. Implementation of actions will typically be platform specific. The adaptation engine interfaces with platform specific components to execute the selected action(s). For actions that are not associated with the current domain, it will send details of the adaptation trigger and recommended action(s) to the target domain (where it can be executed) using the communication services.

f. *Repository:* All persistent information associated with the service is managed through the repository. These include information pertaining to contexts, actions as well as the decision model.

Figure 5 depicts the flow of information between different components during initialization and processing of a change in context values. For clarity we have separated the monitor component from that of context manager.

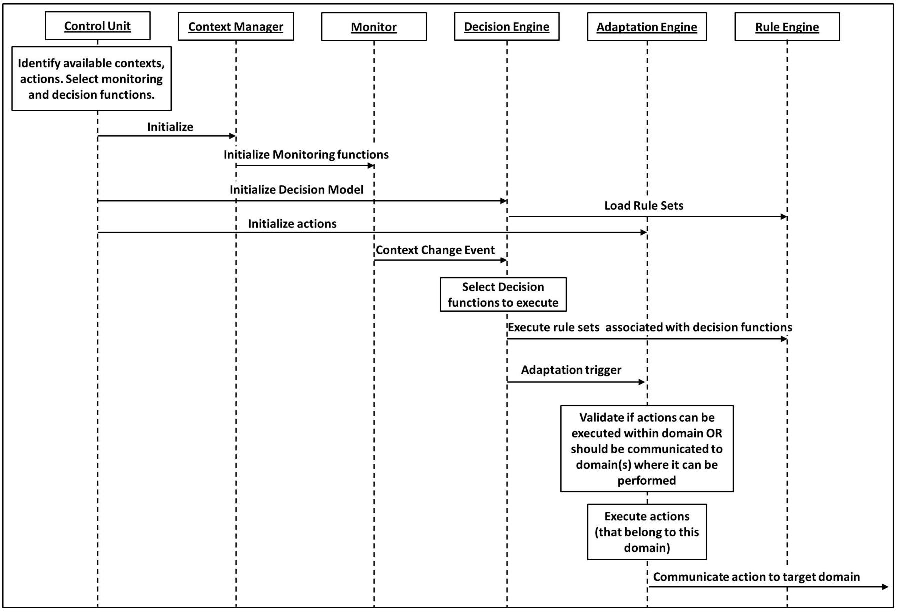

Fig. 5. Framework Message flow

4 Delivering Context Aware Multimedia Services

We now proceed to show how the model described in section 2 and the framework described in section 3 can be used to deliver multimedia streaming services over heterogeneous devices and networks with optimal QoE. We apply the 2-step approach presented in Figure 2 to arrive at the decision model that can be used for service adaptation.

a. *Service Analysis:* Figure 1 identifies the participating domains (User, Network and Service) as well as an indicative list of associated contexts and actions for multimedia delivery services. An analysis of the different contexts show that some of them (e.g. network state), can be monitored with different levels of granularity at all participating domains. Also, it is possible to exchange information about static context (e.g. device capabilities, content metadata) between the different domains.

b. *Model definition:* Here, we first identify the monitoring and decisions functions and then evolve the decision model based on the same. The decision model can be further refined based on constraints related to specific deployment scenarios.

The monitoring functions associated with the service include those related to device (CPU, Memory, Battery levels), network (Current connection type, Signal strength, session state (Jitter, Latency, Packet Loss), Available bandwidth) and server (CPU, Memory, Energy utilization, Server load (number of simultaneous streaming sessions)).

Decision functions are derived using a 3-step process:

a. Identifying the cause-effect relationship between the different contexts and the estimated as well as acceptable QoE.
b. Identifying the set of actions that can lead to best possible QoE in a given situation.
c. Using the above to identify trigger points for adaptation and the recommended actions.

The above process is iterative, beginning with individual contexts and their relationship to the QoE. Subsequently multiple related contexts are combined and their impact on QoE evaluated. For example, both network congestion and weak signal strength manifests as playout delays in the case of HTTP streaming. However the adaptation actions to be triggered in each of the cases are different. The decision function evaluates both signal strength and packet latency to decide on the required adaptation. Signal strength variations are handled by controlling the buffering rate at the user side. Congestion can be eased by switching to a lower bit rate video.

As discussed in section 3, in the real-world, context aware adaptation has to happen under different kinds of constraints. For example, in the case of multimedia services, based on how the service is delivered, the service provider may be able to control one or more of the participating domains. Constraints are handled in the following manner.

- Deriving the decision model that can operate within the stated constraints by adding/removing contexts, decision functions and associated actions.
- Leveraging the flexibility offered by the framework defined in section 3 to effectively implement the above across participating domains. This is depicted in Figure 6.

Figure 6a depicts how the framework components are distributed across domains when the adaptation decisions can be made only in the client.

Figure 6b describes the scenario where the user and service domains collaborate to deliver optimal QoE. Here both the monitoring and decision functions are distributed across the user and service domains. Also, the server can now use additional context information (e.g. policies, SLAs, server state information) to control how and where adaptations need to be done. In both the scenarios, the device can receive information about the content being streamed as application data. Information such as supported bit rates helps the decision model to limit the set of adaptations that can be performed.

Component	Features	Deployment Domain
Context Manager	User, Device, Content, Network	Client
Monitor	Device State Monitor, Network Connection Monitor, Network State Monitor	Client
Decision Engine	Decision model based on the selected context and available actions	Client
Adaptation Engine	Vary Buffering Rate, Select Network	Client
	Vary streaming rate	Server

a. Client side adaptation

Component	Features	Deployment Domain
Context Manager	User, Device, Network (Connection type, signal strength)	Client
	Content, Network (jitter, latency, available bandwidth), Policy	Server
Monitor	Device State Monitor, Network Connection Monitor	Client
	Network State Monitor	Server
Decision Engine	Decision model split across client and server to enable localized decision making (e.g. control buffering rate based on signal strength)	Client and Server
Adaptation Engine	Vary Buffering Rate, Select Network	Client
	Vary streaming rate	Server

b. Client and Server side adaptation

Fig. 6. Sample Framework configurations

5 Proof of Concept Implementation

We now describe a proof of concept (PoC) application developed using the framework described in Section 3. The PoC demonstrates a sample multimedia service operating in a constrained environment. It enables users to search and access video content over web using mobile devices with the following constraints.

- The service should support mobile devices running android version 2.2 or higher.
- Video content is hosted on a third party hosting infrastructure that supports APIs to dynamically switch the video bit rate of the served content through APIs.
- Metadata about the content is available in a locally hosted web server.
- The service should be accessible over a range of devices (with different resolutions) and access networks (WiFi/3G/4G).

The need to deliver content over web restricts the possible adaptation to that of varying the video bit rate using APIs provided by the video hosting infrastructure. Also, Android OS restricts network state monitoring to only detection of network type change and transmit/receive statistics.

Applying the steps described in section 2, we evolved a decision model subject to constraints specified above. The associated decision functions can be categorized as those triggered during startup and continuously at runtime.

a. Startup functions determine the initial video quality (and hence the corresponding bit-rate) using the static context associated with device (e.g. form factor, resolution) and current network type (WiFi/3G/4G).
b. Run-time functions are selected based on corresponding context change triggers. These include decision functions to handle changes in network type and available bandwidth, variations in traffic arrival patterns and video playout state changes (e.g. Seek, Pause, Resume).

The model is implemented using the framework presented in section 3. Given the above described constraints, the framework is largely implemented on the client side. However, the actual adaptation function is distributed across client and server, with the client invoking the requests to change video bit rate and the server performing the actual switch. A logical view of the implemented system is presented in Figure 7.

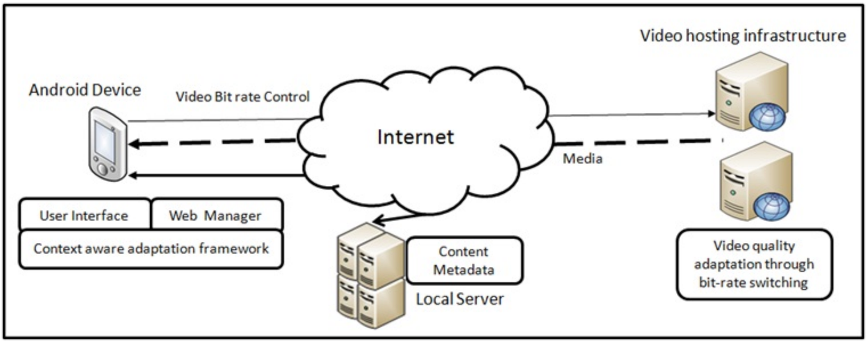

Fig. 7. System Description

The local server holds information about the content (metadata) that is being served. The PoC has three major client side components:

a. User Interface interacts with the local server to enable end-users to search and select video to view.
b. Web Manager interacts with the video hosting infrastructure to fetch and render the video. It also sends video bit rate change requests to the server based on triggers from the adaptation engine.
c. Framework components responsible for monitoring context and triggering appropriate adaptations.
    - *Monitoring:* Static context information is fetched at the beginning of the session. These include device capabilities as well as content metadata. We register receivers with the Android operating system to detect network change events (e.g. switching from 3G to WiFi). Available bandwidth is estimated by continuously monitoring application specific network traffic statistics. Arrival traffic patterns are deduced based on the variance in traffic statistics.

- *Decision engine:* Decision model is represented as rule sets and executed using a rule engine.
- *Adaptation engine:* Triggers appropriate events in the web manager which in turn triggers video bit rate changes by the server.

The User Interface and Web Manager components are implemented as native Android activities. The framework is implemented as a service invoked when the user selects a video to view.

The effectiveness of adaptation was verified using the following tests. The tests were started after the user selects the content to view.

a. Changing the network from WiFi to 3G and vice-versa.
b. Continuously varying the available bandwidth (e.g. throttle bandwidth to 256kbps, increase the same to 512kbps and then to 1Mbps and so on).
c. Use devices with different form factors and resolutions (e.g. HTC Desire S (480x800) and HTC Explorer (320x480)).

For each test, we observed the variations in QoE when compared to a system without adaptation. This is measured as a function of variation in number of times the player moved between buffering and playing. Larger the number, lower the QoE as user experience is impacted by frequent buffering. The expected adaptation triggers were verified with actual adaptations.

Figure 8 and Figure 9 depict the QoE as a function of variation in buffering in 2 different scenarios. Figure 8 depicts the scenario where bandwidth is varied as per the following sequence (Normal 300kbps 512 kbps 300kbps 512kbps). Bandwidth was maintained at each level in the sequence for a period of 5 minutes. Figure 9 depicts the performance under consistently low bandwidth (256kbps). As we can see, in both cases, introducing adaptation smoothens out the playout leading to an improved QoE.

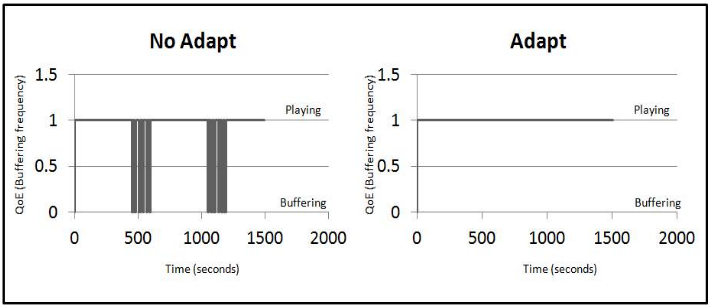

Fig. 8. Buffering under varying bandwidth

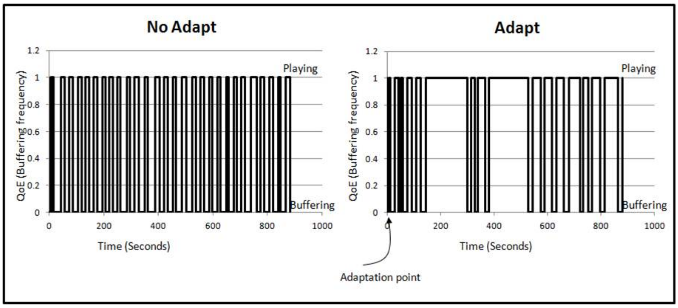

Fig. 9. Buffering under low bandwidth

6 Conclusions

This paper presents an approach to analyze services and evolve guidelines for adaptation to optimize end user QoE. We also described a generic framework to enable effective implementation of the adaptation model. As a specific case, we presented a context-aware service adaptation framework for multimedia service delivery over heterogeneous wireless networks and devices. The objective of such adaptation is to provide a consistent QoE to the end user while making the most efficient use of resources in heterogeneous networking environment characterized by wide diversity in access network characteristics and device capabilities.

To demonstrate the framework, we have implemented a PoC which shows how the key characteristics (bitrate, resolution etc) of a video stream can be modified on the fly in response to change in network bandwidth and/or device handover. This is realized with the help of a context monitor which provides feedback to the adaptation engine located at the server side.

This work is currently being applied in a pilot project for the National Program on Technology Enhanced Learning (NPTEL) [15], an initiative of the Indian government to promote e-learning using videos of lectures by faculty at different IITs (Indian Institute of Technology). Work is also underway to enhance the model to enable multi-modal communications and information delivery in disaster management systems.

Acknowledgments. This research is funded by the Indo-UK Advanced Technology Centre (IU-ATC), a research initiative funded jointly by the Department of Science and Technology (DST) in India and the Engineering and Physical Sciences Research Council (EPSRC) in UK. We would also like to acknowledge Ms. Naga Jyothi for her efforts in implementing parts of the PoC.

References

1. Goebbels, S.: Smart Caching for Supporting Video Streaming in Heterogeneous Wireless Networks. In: Proceedings of the 11th International Symposium on Wireless Personal Multimedia Communications (2008)

2. Gomaa, H., Messier, G., Davies, R., Williamson, C.: Media Caching Support for Mobile Transit Clients. In: Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (2009)
3. Lee, S.J., Ma, W.-Y., Shen, B.: An interactive video delivery and caching system using video summarization. Elsevier Computer Communication Journal (2002)
4. Xin, J., Lin, C.-W., Sun, M.-T.: Digital Video Transcoding. Proceedings of the IEEE 93(1) (January 2005)
5. Schwarz, H., Marpe, D., Wiegand, T.: Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. IEEE Transactions on Circuits and Systems for Video Technology 17(9) (September 2007)
6. Zambelli, A.: IIS smooth streaming technical overview. Microsoft Corporation (2009)
7. De Cicco, L., Mascolo, S., Palmisano, V.: Feedback Control for Adaptive Live Video Streaming. In: Proceedings of the Second Annual ACM Conference on Multimedia Systems (2011)
8. Tappayuthpijarn, K., Liebl, G., Stockhammer, T., Steinbach, E.: Adaptive video streaming over a mobile network with TCP-friendly rate control. In: Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly (2009)
9. Balk, A., Gerla, M., Maggiorini, D., Sanadidi, M.Y.: Adaptive video streaming: pre-encoded MPEG-4 with bandwidth scaling. Computer Networks 44(4) (2004)
10. Lemlouma, T., Layada, N.: Context-Aware Adaptation for Mobile Devices. In: Proc. IEEE Int. Conf. on Mobile Data Management (2004)
11. Schmohl, R., Baumgarten, U., Kothner, L.: Content Adaptation for Heterogeneous Mobile Devices using web-based Mobile Services. In: Proceedings of MoMM (2007)
12. Attou, A., Moessner, K.: Context-Aware Service Adaptation Management. In: Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (2007)
13. Malandrino, D., Mazzoni, F., Riboni, D., Bettini, C., Colajanni, M., Scarano, V.: MIMOSA: context-aware adaptation for ubiquitous web access. Journal of Personal and Ubiquitous Computing 14(4), 301–320
14. Miraoui, M., Tadj, C., Fattahi, J., Ben Amar, C.: Dynamic Context-Aware and Limited Resources-Aware Service Adaptation for Pervasive Computing. Advances in Software Engineering (2011)
15. National Programme on Technology Enhanced Learning, `http://www.nptel.iitm.ac.in`