RTSP Audio and Video Streaming for QoS in Wireless Mobile Devices

Arun Kumar B. R.[†] **, Lokanatha C. Reddy**[††] **, Prakash S. Hiremath**[†††] **, Naresh.S.S**[††††]

Asst. Prof. Dept. of MCA Sir MVIT, Bangalore & Research Scholar, Dept. of CS, School of Science & Technology, Dravidian University, Kuppam-517425, A. P., India. [†]
Professor, Dept. of CS, School of Science & Technology, Dravidian University, Kuppam-517425, A. P., India[††]
Professor, Dept. of CS, Gulbarga University, Gulbarga-585106, Karnataka, India[†††]
Software Engineer,Silver Software Pvt.Ltd. ,Bangalore-66, Karnataka, India[††††]

Summary

This research work focuses on application layer. By making use of RTSP we can extend control over the delivery of data with real-time properties. RTSP is the robust protocol that can stream multimedia over multicast and nicest in 'one-to-many' applications. RTSP takes advantage of streaming which breaks data into many packets sized according to the available bandwidth between client and server [19] [20]. An usual instance is while enough packets have been received by the client; the user's software can be playing a particular packet, decompressing one packet and downloading another. The user is able to start listening/viewing almost immediately without having to get the entire media file. Both live data feeds and stored clips can be the sources of data. Using RTSP we can reduce the jitter and improve the quality to a significant extent.

Key words: *Delay, Jitter, Multimedia, Quality of Service, Real Time Streaming Protocol, Video.*

1. Introduction

Streaming is the process of transferring data via a channel to its destination with real time characteristics, where it is decoded and consumed via a user/device in real time, i.e., as the data is being delivered on the fly [1], [2], [3], [4], [5], [6]. It differs from non-streaming process because it does not require the entire data to be fully downloaded before it can be seen or used. Streaming is never a property of the data that is being delivered, but is an attribute of the distribution channel. This means that, technically, most media can be streamed up.

1. 1. Necessity of Mobile Streaming [7]:

Mobile handheld devices such as Personal Digital Assistants (PDAs) and Smart phones are increasingly being targeted by service providers to deliver application functionality similar to that found in traditional desktop computing environments. However, these handheld devices at application level can be quite slow and often lack important functionality compared to their desktop counterparts.

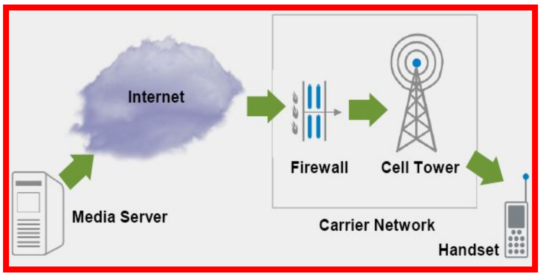

Figure 1: Typical Components of Data Streaming in Mobile Devices [7]

The increasing ubiquity of wireless networks and the decreasing costs of hardware are fueling a proliferation of new classes of mobile wireless handheld devices, including wireless Personal Digital Assistants (PDA) and integrated PDA/cell phone devices [11]. These devices are enabling new forms of mobile computing and communication. Service providers are leveraging these devices to deliver general application functionality similar to what is found in traditional desktop computing environments, including web browsing, email, video, music, financial planning, and personal information management.

The types of protocols currently used for streaming are Real Time Control Protocol (RTCP), Real Time Streaming Protocol (RTSP), Session Description Protocol (SDP) and Real-time Transport Protocol (RTP) [7], [10], [11], [14], [15], [16], [17], [18].

The HTTP browser lets us to browse HTTP directories. To make Simple Player's browser to treat the URL as a directory, include a slash at the end of the URL address [21]. Selecting a media file opens it in Simple Player. To play the simple audio and video files it uses the samples of

Manuscript received January 5, 2008
Manuscript revised January 20, 2008

the corresponding wave file and mpeg file. The audio file is of wav-format and video file is of mpg-format; those are the samples to play and are located in the resource folder. The URL path/site accesses the audio and video files for download and to play. The attachment of files provided from server connects through HTTP location. This requires an internet connection; however, the audio/video file is accessed from an RTSP site. Audio Capture from a default device lets us to capture audio from a radio or other device. The sound is captured and played back and we hear the initial sound followed by the playback of the sound after a brief latency. Entering a URL plays back audio and video files from the Internet. Typing a valid URL at the insertion point and clicking OK initiates to play a WAV, MIDI, or MPEG-1 file. While opening an HTTP directory from which to select a media, we make sure to add a slash to the end of the URL address.

In existing system, the user has to wait for a long time to download and playback the audio and/or video files. The device needs large buffer space for capturing and storage of entire data from the Internet. The downloading of media files needs more amount of channel space and also capturing of streaming data leads to inconvenient. Client device must wait before the complete download occurs.

In this research work RTSP is used which can control multiple data delivery sessions; provide a means for choosing delivery channels such as UDP, multicast UDP and TCP; and provide a means for choosing delivery mechanisms based upon RTP.

2. Real-Time Streaming Protocol [2], [23], [24], [25], [26], [27]:

RTSP takes advantage of streaming which breaks data into many packets, sized according to the bandwidth available between client and server. The idea in RTSP is that it acts as a 'network remote control' for multimedia servers.

RTSP has been designed to be on top of RTP to both control and deliver real-time content. Thus RTSP implementations will be able to take advantage of RTP improvements, such as RTP header compression. Although RTSP can be used with unicast, its use might help to smoothen the change from unicast to IP multicasting with RTP. Real Time Streaming Protocol can also be used with RSVP to set up and manage reserved-bandwidth streaming sessions.

Differences between RTSP and HTTP: The RTSP is intentionally similar in syntax and operation to HTTP/1. 1. However, it differs in a number of important aspects from HTTP [6], [7], [8], [9], [10], [11].

RTSP introduces a number of new methods and has a different protocol identifier. An RTSP server needs to maintain state by default in almost all cases, as opposed to the stateless nature of HTTP. Both an RTSP server and client can issue requests. Data is carried out-of-band by a different protocol [12], [13], [14], [15], [16].

RTSP is defined to use ISO 10646 (UTF-8) rather than ISO 8859-1, consistent with current HTML internationalization efforts. The following operations are supported by RTSP protocol: Retrieval of media from media server, Invitation of a media server to a conference, Addition of media to an existing presentation. RTSP requests may be handled by proxies, tunnels and caches as in HTTP/1.1.

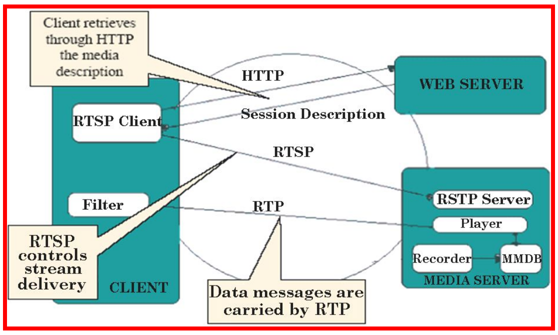

Figure 2: Interoperability Scenario [24]

2. 1 Quality of Services Implications for Streaming:

Streaming implies that a guaranteed rate is ensured to the applications. A very low rate results in low quality of the played video and there is a need for QoS when streaming is deployed. QoS parameters can be negotiated with the Media Server by the user. The negotiation can happen transparently and autonomously for the user. The streaming client can take care of it, based on network connection type, media stream to be downloaded.

2. 2 RTSP messages:

RTSP is text based and uses ISO 10646 character set in UTF-8 encoding. The overhead of text coded protocols is not an issue due to the small size of RTSP messages. RTSP messages can be Requests, Responses.

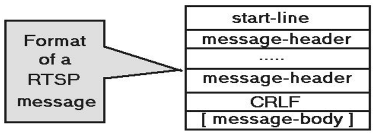

Figure 4: RTSP Message Header

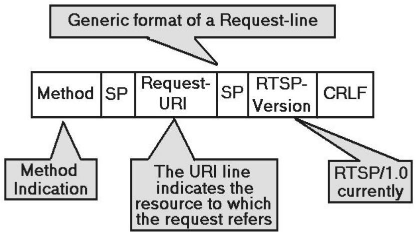

Figure 5: Typical RTSP Message Format

2. 3 Streaming mobility related issues:

Most of the streaming services provided are now meant for desktop (or at least powerful laptops) machines in a wired network. Streaming services provided to mobile terminals must be tailored to fit the requirements of a different kind of network and a different kind of terminal. Streaming media implies a steady bandwidth assured at the recipient side. If the recipient is attached to a wireless network several problems arise. The bandwidth of wireless links is often limited and subject to sudden variations. Terminal mobility adds complexity to the problems (e.g., handovers). Mobile terminals are lightweight, have limited size, power computing and battery duration. In a mobility situation, for the client, a way to cope with bandwidth unsteadiness is to buffer enough packets and play them with a little staggering to prevent situations of loss of connectivity. This is not a final solution. If the periods of non-connectivity last for long then the streaming media delivery halts. In disconnected environments, reliability is most important. Although RTSP ensures application level mobility, TCP is strongly recommended to deal with packet losses, duplicates and retransmissions.

One of the reasons for which streaming is nowadays only performed over wired (with very few exceptions) is the low bandwidth availability that 2G networks offer. GPRS ensures around 40 kbit/sec, in a good situation, which is not enough for streaming. The forthcoming 3G networks appear to be more promising, with their (at least in theory) 384 kbit/sec of guaranteed rate. Mobile devices such as PDAs, Smart Phones, and Internet Appliances have some disadvantages compared with desktop computers or laptop computers connected to the Internet via dial-up networks, such as Low Bandwidth, Connection Stability, Small Display, Limited Input, Limited Main Memory, Limited CPU Speed, Limited Battery Power, Limited Storage capacity etc., [26], [27, [28].

3. Emulator Setting [29], [31], [32], [33], [34]:

The emulator settings are adjusted to more closely resemble a specific device or to test under different resource conditions. Network Proxies: The emulator uses desktop network connection. For example, if the emulator runs a MIDlet that makes an HTTP connection, the emulator attempts to make the HTTP connection using the desktop's network setup. Heap Size: Maximum heap size is set to more closely simulate the conditions on a real device. Here, size is set to be 1024 bytes.

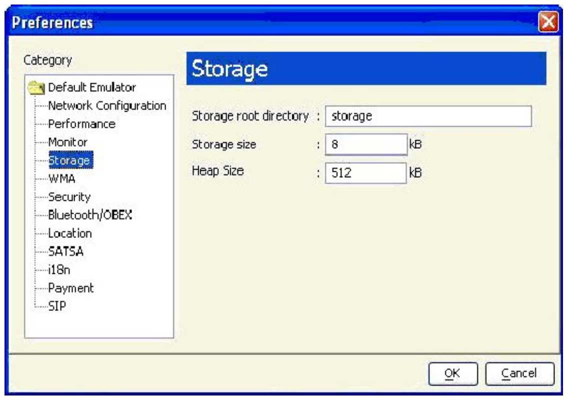

Figure 6: KToolbar Storage Preferences

The emulator has persistent storage, which by default is placed in toolkit\appdb\skin in files with a. db extension. The emulator uses many of the resources of a desktop computer, including its display and network connection. The Sun Java Wireless Toolkit for CLDC enables us to simulate the constrained environment of a real device.

When the application makes any type of network connection, information about the connection is captured and displayed. The Figure 7 shows HTTP requests and responses.

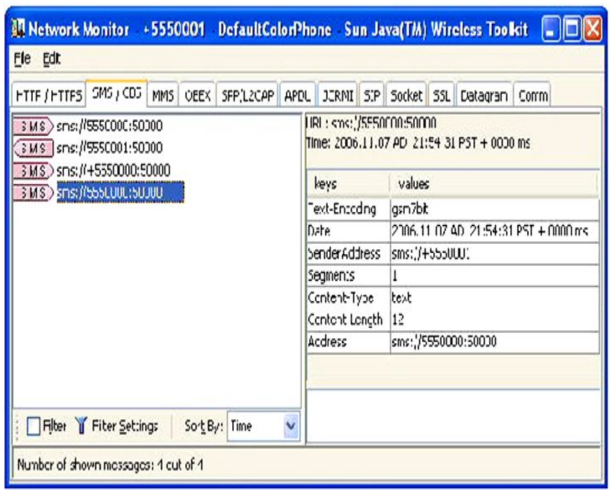

Figure 7: Network Monitor

4. Results and Conclusion:

4. 1 Results:

The primary objective of **streaming here** is the downloading audio/video files from internet and also playing at a time using the buffer to retrieve the data. A Streaming MIDlet created a Simple Player for audio and video media files from remote network servers.

In Figure 8, the values used are based on the simulated network load running for a given mobile device with simulated load from 0 to 1 (0 to 100 percent) capacity against response time (0 to 3 sec) shown with blue line. The response time (0 to 3 sec) is varying as the maximum load is applied; response time is smooth when constant load is put. Load p is applied on the basis of factor $1/(1-p)$.

For application such as audio and video streaming, it does not matter if the packets take 20 msec or 30 msec to be delivered, as long as the transit time is constant. The variation in the packet arrival times is called jitter. Figure 9 shows very high jitter because variation of packet is up to 80 msec which will result in uneven quality of sound and video clips. This happens when streaming technology is not used.

Figure 10 next, shows very controlled jitter or reduced jitter; packet variation may go up to 10 to 12 msec which is with in the control zone. This reduced jitter contributes significantly to quality of the data or sample space.

In Figure 11 next, the output screen shows an incomplete picture or image with flickering resolution: it is a common difficulty with the non-streaming technology (downloading). In Figure12, the output screen shows a good quality image without the flickering problem. A comparison of Figures 11 and 12 shows we can get a better quality image or picture when streaming is used with wireless devices (mobile).

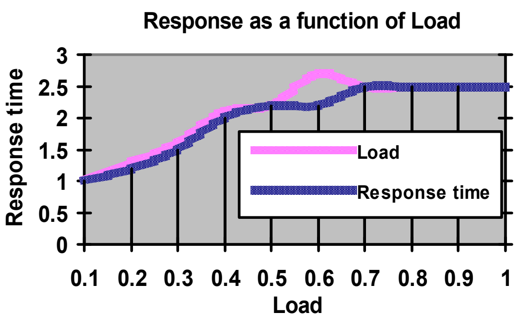

Figure 8: Response time of the devices on load application.

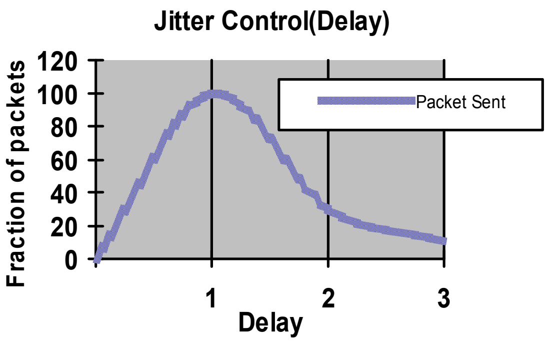

Figure 9: Comparison of packet fractions when streaming technology is not used.

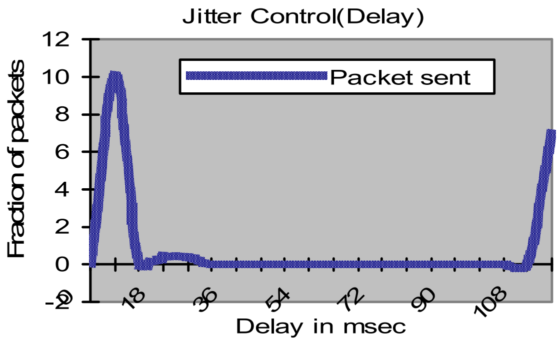

Figure 10: Highly reduced jitter when streaming is used

4. 2 Conclusion:

In this research work wireless mobile handheld devices are considered. We have shown that using RTSP we can reduce the jitter and improve the quality to a significant extent.

Multimedia delivery inherently has strict quality-of-service (QoS) requirement on bandwidth, delay, and delay jitter. The advent of wireless networks further exacerbates the variance of network conditions and brings greater challenges for multimedia delivery. This paper addresses only application layer. To further improve perceived media quality by end users over wireless Internet in mobile devices, QoS supports can be addressed in different layers, including application layer, transport layer, link layer, and so forth.

There are still a lot of issues needed to be further investigated. Efficient work on QoS provisions for multicast media streaming is an area that requires lots of efforts [35]. Mobility also has significant impact on perceived QoS during multimedia streaming. How to maintain an acceptable media quality when handoff happens is another research direction [36].

Figure 11: Output screen showing quality of the video when streaming is not used for accessing the data.

Figure 12: Better output when streaming technology is used for accessing the data.

Enabling media streaming over ad hoc network is more challenging than over traditional wireless networks. In wireless ad hoc networks, dynamic changing topology and interference result in even greater QoS fluctuation. Recently, multi-path media streaming and QoS-aware MAC design are two promising cross-layer approaches to providing QoS support for ad hoc networks [37].

5. References

[1] http://www.bitecomm.co.uk/Prognet_pages/corp_backgrounder.html
[2] http://www.netlab.ohiostate.edu/~jain/cis788-97/ip_multimedia / index.htm
[3] http://www.fit.qut.edu.au/Student/ITB235/papers/Compress/n1991761/assign1b.htm
[4] http://www.mbone.com/lists/
[5] http://www.sone.gov.sg/developers/pnet.html
[6] http://www.cs.ucl.ac.uk/staff/c.perkins/reports/ietf_38/node3.html
[7] http://www.real.com/devzone/library/fireprot/rtsp/index.html/
[8] http://www.isi.edu/div7/rsvp/rsvp-home.html
[9] http://www.cs.columbia.edu/~hgs/rtp/
[10] http://huntleyl.prognet.com/prognet/rt/index.html
[11] http://www.cs.columbia.edu/~hgs/rtsp/
[12] http://ftp://ftp.isi.edu/in-notes/rfc2205.txt
[13] http://www.ietf.org/html.charters/mmusic-charter.html
[14] http://www.ietf.org/
[15] http://www.real.com/devzone/tools/rmsdk/index.html
[16] http://www.cs.columbia.edu/~hgs/rtsp/
[17] ftp://ftp.isi.edu/in-notes/rfc1889.txt
[18] ftp://ftp.isi.edu/in-notes/rfc2326.txt
[19] Stallings, W, *"High Speed Networks, TCP/IP and ATM Design Principles"*, Prentice-Hall Inc.,1998.
[20] http://www.ipmulticast.com/

[21] Berners – Lee, Masinter L and M. Mc-Cahill "Uniform Resource Locator (URL)", RFC1738 DECEMBER 1994.
[22] Schulzrinne H, Cosner S, Fredrick R, V Jacobson "RTT – Transport protocol".
[23] Craig Partridge, *Gigabit Networking,* Addison-Wesley, 1994.
[24] Stephan Thomas, Wiley, IPng and the TCP/IP Protocols, 1996.
[25] Vinay Kumar, NewRiders, *MBONE: Interactive Media on the Internet,* 1996
[26] Kassler, A. Schorr, L. Chen, C. Niedermeier, C. Meyer, M. Helbing, M. Talanda: "Multimedia Communication in Policy based Heterogeneous Wireless Networks", IEEE Vehicular Technology Conference VTC2004-Spring, Milan, Italy, May 2004.
[27] Sevanto J (1999), Multimedia Messaging Services in GPRS and UMTS Wireless Communications and Networking Conference.
[28] Kassler, A. Schorr, L. Chen, C. Niedermeier, C. Meyer, M. Helbing, M. Talanda: "Multimedia Communication in Policy based Heterogeneous Wireless Networks", IEEE Vehicular Technology Conference VTC2004-Spring, Milan, Italy, May 2004.
[29] J2ME Complete Reference - Herbert Schildt, 5th Ed.
[30] Computer Networks by Andrew S Tanenbaum EEE Publication 5th Ed.
[31] James Keogh, "J2ME: The Complete Reference", 2003 Edition, McGraw-Hill/Osborne Publishing, 2005.
[32] Kim Topley, "J2ME in Nutshell", March, 2003 Edition, O'Reilly Publishing, 2005.
[33] Vartan Piroumian, "Wireless J2ME™ Platform Programming", March 2002, Prentice Hall PTR Publishing, 2004.
[34] Pressman, "Software Engineering - A Practitioner's Approach", 5th Ed, Tata McGraw-Hill, 2004.
[35] A. Majumda, D. Sachs, I. Kozintsev, K. Ramchandran and M. Yeung, "Multicast and unicast real-time video streaming over wireless LANs, " *IEEE Trans Circuits Syst. Video Technol.,* vol. 12, no. 6, pp. 524–534, 2002.
[36] Y. Pan, M. Lee, J. Kim, and T. Suda, "An end-to-end multipath smooth handoff scheme for stream media", *IEEE J. Select. Areas Commun.*, vol. 22, no. 4, pp. 653–663, 2004.
[37] S. Mao, S. Lin, S. S. Panwar, Y. Wang, and E. Celebi, "Video transport over ad hoc networks: multistream coding with multipath transport, " *IEEE J. Select. Areas Commun.,* vol. 21, no. 10, pp. 1721–1737, 2003.

Authors

† Arun Kumar. B. R received his MCA Degree from Kuvempu University and M. Phil from M. S University, M. Tech (CS & E) from Dr. MGR University in 1999, 2003 and 2006 respectively. He is working as an Assistant Professor in the Dept. of MCA Sir MVIT, Bangalore, Karnataka, India. He is a Research Scholar in the Dept. of Computer Science at Dravidian University, Kuppam, AP, India and working towards his Ph. D. Degree. His current areas of research are QoS Multicasting in MANET, Network Security, DIP, Cyber Laws, and IPR Laws etc.

†† Lokanatha C. Reddy earned M.Sc. (Maths) from Indian Institute of Technology, New Delhi; M.Tech.(CS) with Honours from Indian Statistical Institute, Kolkata; and Ph.D.(CS) from Sri Krishnadevaraya University, Anantapur. Earlier worked at KSRM College of Engineering, Kadapa (1982-87); Indian Space Research Organization (ISAC) at Bangalore (1987-90). He is the Head of the Computer Centre (on leave) at the Sri Krishnadevaraya University, Anantapur (since 1991); and a Professor of Computer Science and Dean of the School of Science & Technology at the Dravidian University, Kuppam (since 2005). His active research interests include Real-time Computation, Distributed Computation, Device Drivers, Geometric Designs and Shapes, Digital Image Processing, Pattern Recognition and Networks.

††† Prakash. S. Hiremath, M. Sc, Ph. D is currently working as a Professor and Chairman of Dept. of P. G Studies and Research in Computer Science in Gulbarga University, Karnataka, India. His areas of research interests are Wireless Networks, DIP, and Pattern Recognition.

†††† Mr. **Naresh . S. S , MCA,** is currently working as Software Engineer at Silver Software Pvt. Ltd. Bangalore.