AN EMBEDDED NETWORK-ENGINE

Md. Amiruzzaman
College of Electronics and Information Engineering, School of Computer Engineering
Sejong University, Seoul, South Korea
E-mail: m_amiruzzaman@email.com

Abstract: *A new technique of an embedded network-engine which is working on IEEE 802.11 and 802.16e WLAN environment for Video on Demand (VoD) is presented in this paper. The proposed the network-engine with the capability of controlling the transmission between embedded client and server over popular wireless environment has been offered.*

Keyword: *VOD, Network-engine, PDA, Socket*

1. INTRODUCTION

Several video transmission techniques, such as enabled network transmission, storage, compression, large amount of digital information, i.e., audio and video [1], [2], [3] have been developed for the embedded systems including personal device assistant (PDA). Now users want more facilities for wireless devices with multimedia application and want to share information with others. In order to support the transmission techniques, in the last decade the researcher achieved their success for the high speed wireless communication and invented the WiBro (Wireless Broadband) and WiMAX (Worldwide Interoperability for Microwave Access).

We have focused on Video on Demand (VoD) transmission over Wi-Fi and WiBro transmission; this work covers for Server-to-PDA Video on Demand (VoD) transmission [1].

During the research, the memory capacity of PDA was considered, transmission rate, bandwidth, connection procedures and user interaction. The simulation was done for the total process with some real video clips and obtained some simulated performance results which are shown in the result section. During the survey of the related works some work found. Few works have been done by some researchers for the network-engine which is particularly for audio, video streaming [4], games [5] and voice [6]. In this paper, the proposed wireless network-engine with the capability of controlling the input and output device and have analyzed by performance results and described.

The organization of this paper is as follows. In Section 2, literature review is described. In section 3, implementation of our technique is given. In Section 4, comparison with TCP protocol; performance results are shown in section 5, conclusion is made and future work is presented.

2. LITERATURE REVIEW

There are few works found on transferring on RTP protocol for data transfer especially for the resource constraint device-to-server and client-to-server. However, several works have been done on data recovery and feedback control on multimedia stream. Vit Novotny et al. proposed a feedback control technique for multimedia stream data based on SSM (Source Specific Multicast) [7]. Radim Burget et al. proposed the improvements for IPTV (Internet Protocol Television) [8]. H. Schulzrinne et al. describes an end-to-end network transport functions suitable for applications transmitting real-time data [9-10].

Querying live multimedia stream data backup is a challenging problem that is becoming an essential requirement in a growing number of applications [12]. Research has addressed and made good progress in dealing with archived data [13]. Meanwhile, research in stream data has received significant attention for backing up on server [14]. The lack of a design and develop technique has been noticed during my research. This part of thesis presents a technique which is enough capable to handle real-time stream data and make backup on server [11].

3. IMPLEMENTATION

Our network engine for VOD describes the server to output section (VoD data transfer from the server to output device). Figures 1 and 2 is showing the working procedure of our proposed network engine

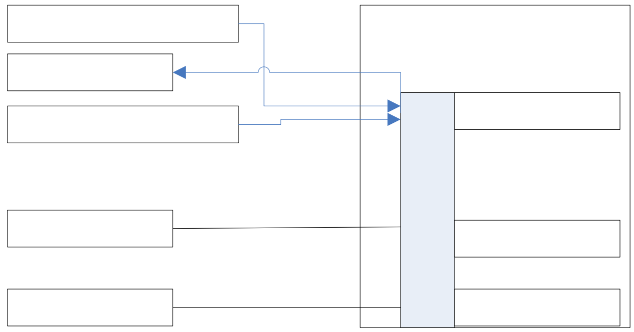

Fig. 1 Working procedure-I

The session control of the proposed network-engine is shown in Figure-1; the sessions are controlled by the help of a session manager. From the above figure it is clear that number of sessions can be run at a time as well as number of applications. The sessions of the connections could be established and resumed. According to the client (e.g., PDA) request the record option could be work, while there is an option to end the session.

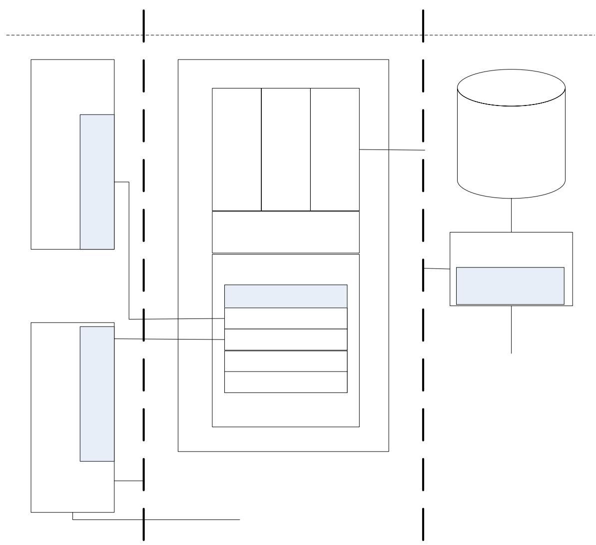

Fig. 2 Working procedure-II

In Figure 2, the simplest structure of the proposed network-engine is shown, where the video files (e.g., VoD) are managed by a simple message server. The VoD server is nothing but based on a message server (if message can transferred between server and client then VoD files also may share). To controlled the message server (i.e., VoD server) by the content consumer (Video file agent), the meta-data server which has all the information about the VoD files (i.e., Video files) is treated as an enterprise data warehouse is connected with the message server via content provider (video file agent). Figures 3 and 4 are representing the implicit structure of the proposed technique.

From Figures 3 and 4 the researcher may get the information that how the proposed network-engine is working by the help of simple socket programming. The proposed network-engine is working a middle-ware, which has two parts: user interface and socket interface. The user interface and the socket interface are working on top of operating system (in case of PDA, Windows Mobile is the OS and in case of server Fedora Core 4 is the OS). From Figures 3 and 4 it is also clear that the network engine is directly dependent on network protocol, which means that in case of protocol change the network-engine might be modified.

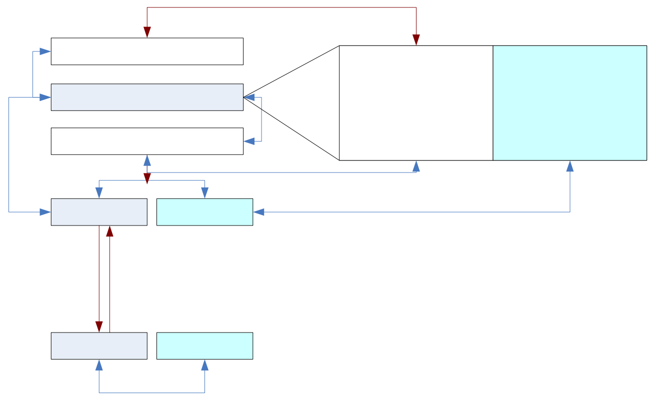

Fig. 3 An overview of VOD network- engine design (I)

In Figure 4, the server part of the network-engine is elaborated. The server has two parts: meta-data and real-data (together called database). The database is directly connected and controlled by the server socket (i.e., network-engine). The database queries are made and managed by the network-engine (e.g., select tables).

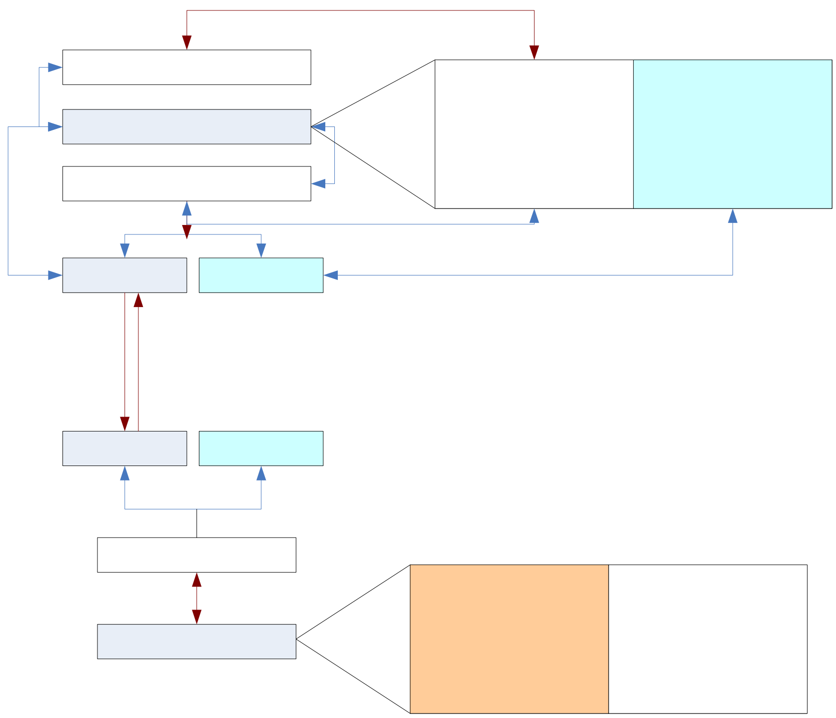

Fig. 4 An overview of VOD network- engine design (II)

3.1 Output section:

In this section I have worked for the data transfer from server to client (e.g., PDA) for downloading, (the saved data from the assigned directory via the database server). On receiving the connection request from the client (PDA) the network-engine establishes a connection between the server and the PDA. After getting the acknowledgement from the server-side socket the client will send request for the VOD list and the network-engine will get connection with the database server as well as with the meta-data. When the user selects VOD names from the list provided by the server; the corresponding VOD data are retrieved from the assigned directory and then are displayed sequentially at the output device. The internal structure of the output section is shown in Figure-5.

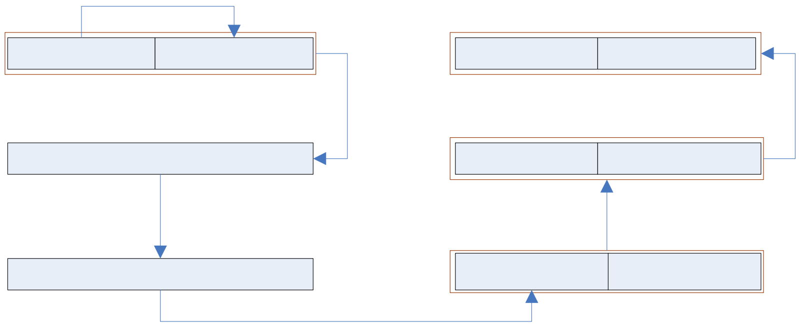

Fig. 5 The internal flow of Server-to-output device

The output system with the user interface is shown in Figure 6.

User Interface

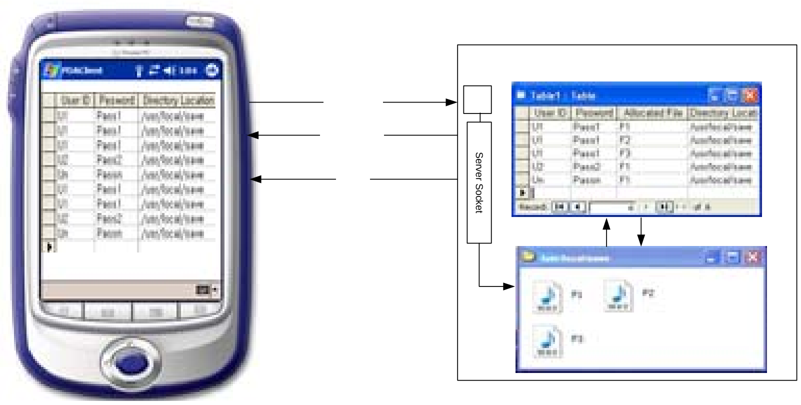

Fig. 6 An overall structure of output section.

Opera

3.2 Indexing Scheme:
An indexing scheme is used to maintaining the proper sequence of received VOD files. This indexing scheme is used to maintain the sequence of meta-data.

The indexing scheme is shown in Figures 7 (a) and 7 (b). I have maintained two indexing schemes: first indexing is for the data transmit in respect of transmitting time and second indexing is for data receive in respect of receiving time.

Netwo

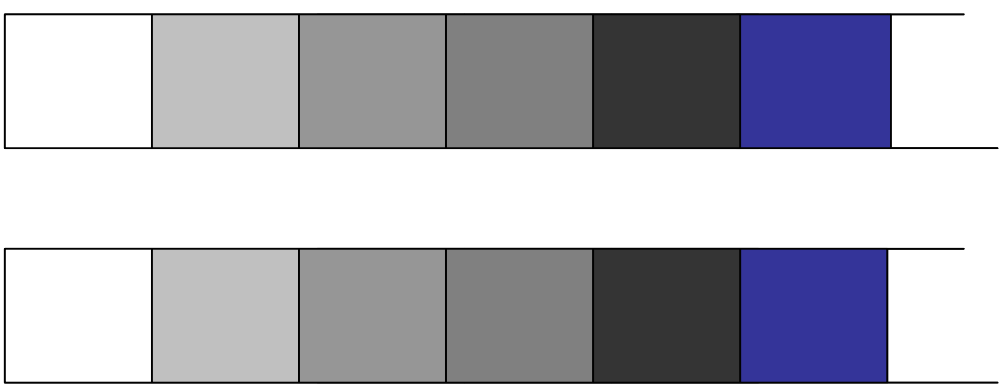

Fig. 7 (a) File transfer with indexing

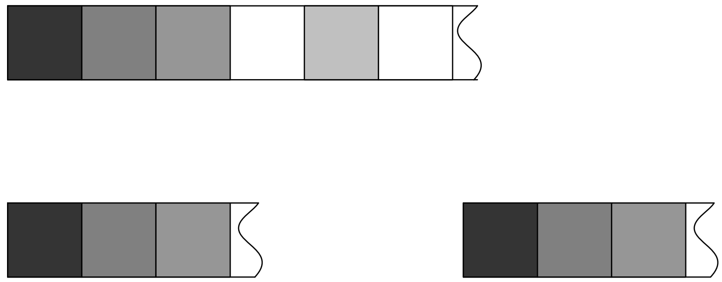

Fig. 7 (b) File transfer with indexing

3.3 Stream Data Backup Server

This technique will work as a real-time video stream data backup server, which is implemented by RTP protocol. Which supports wireless medium (e.g., Wifi and WiBro), the communication implemented and tested on Linux client to Linux server and also Windows Client to Windows server shown in Figure 8. During the communication I also try to maintain a transfer list, which can give us the idea about the transferred file list.

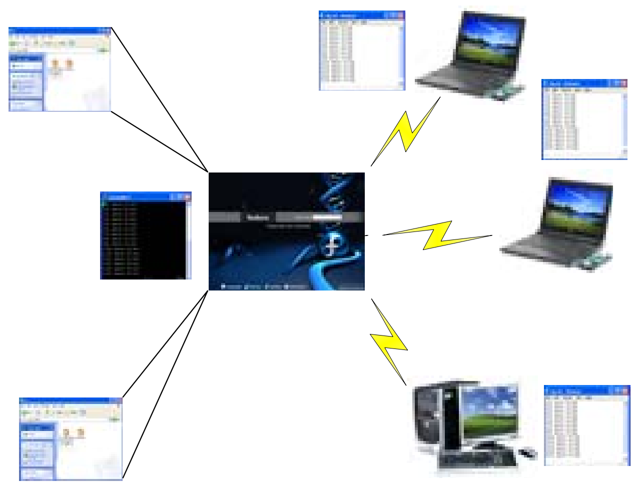

Fig. 8 Proposed network-engine

With the projected explosion of bandwidth availability, the intensive processing tasks and service hosting will move close to consumers on the "intelligent edge" of the network, where a significant portion of the future storage, processing and network management will take place. I address the rationale for this change, the characteristics of the network processor architecture required to address it, and the software development tools needed in order to improve time-to-market without sacrificing stream data backup software performance .

To accomplish my work I use RTP API, with the help of C programming.

4. COMPARISON WITH TCP

The Transmission rate of TCP protocol is:

$$Transmission\_rate = \frac{data\_of\_a\_cycle}{duration\_of\_a\_cycle} \quad (1)$$

Transmission Rate$_{TCP}$ =

$$\left(\frac{S}{R\sqrt{\frac{2bp}{3}} + t(3*\sqrt{\frac{3bp}{8}}\,p)*(1+32*p*p)}\right) \quad (2)$$

Where, *S* is packet size in bytes, *R* duration of cycle in seconds, *t* is transmission timeout, *p* is the average loss rate and *b* is the number of packets acknowledged.

The Transmission rate of RTP protocol is:

$$Transmission\_rate = \frac{maximal\_packet\_size * window\_size}{duration\_of\_a\_cycle} \quad (3)$$

Some applications like video streaming use a simpler protocol called UDP or RTP. Generally, UDP performance numbers will be 15-20 % (percent) higher than TCP performance numbers because there is less protocol overhead associated with UDP [18]

From the above discussion it is understandable that RTP is better than TCP, which supports the proposed method. Discussions also supports that the network-engine which is working for data-backup is beneficial for real-time multimedia stream data transfer.

5. RESULTS

The packet transferring rate between Server PC to PDA have been evaluated, the obtained performance table and graph is shown in Graph 1 and Table 1. During the performance test, several packets with different size has been send to PDA to Server PC as well Server PC to PDA and analyzed the packet lose rate. The whole process was done by the wireless internet connection (i.e., Wifi and WiBro).

Table: Server-PDA data transmission over Bandwidth

Packet size (byte)	Number (Packet)	Received	Lost	Average (ms)
512	50	50	0%	60
1024	50	50	0%	66
2048	50	50	0%	62
4096	50	50	0%	69
8192	50	50	0%	81
16384	50	50	0%	107
32768	50	50	0%	159
65500	50	50	1%	264

Table-1 and Fig. 9 represent the transmission time calculated with respect to packet size.

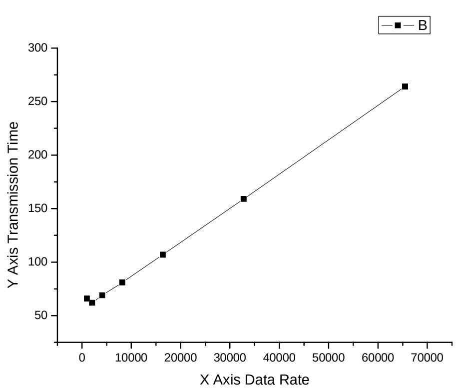

Fig. 9 PDA-server data transmission time

6. CONCLUSION

An embedded network-engine for Video on Demand service is presented in this paper and at the same time a real time stream data backup server is also offered. From the above discussion, it is clear that the proposed network-engine and stream data backup server is good for the wireless communication and especially for the VoD service *[1]*, *[13]*. As a future work I am going to combine my network engine with some techniques for the data recovery during the large-scale of VoD transmission. The proposed work could be helpful for the researcher to improve any kind of data backup technique for embedded network and resource constrained devices.

REFERENCE

[1] Amiruzzaman, Md., Son sung ok, Jaechun No, “An Embedded Network-Engine for Video On Demand Service”, *Proceedings of Korea Computer Congress*, Volume-34, No.-1(A), Seoul 2007, pp.145-148.

[2] Vladimir Nedovic, Chris Nelson, Scott Bowser, and Oge Marques, "Delivery of near real-time soccer match highlights to wireless PDA devices", *Proceedings of the IASTED International Conference on Visualization, Imaging, and Image Processing September 8-10, Benalmádena*, Spain 2003.

[3] Ahmet Ekin and A. Murat Tekalp, “A generic model and sports video processing for summarization and model-based search”, *Handbook of Video Databases* (ed. Borko Furht and Oge Marques), CRC Press, 2003.

[4] Chih-Shoung Huang, et al "Design of video service engine with SCSI for an ADSL-based near Video On Demand system" *IEEE Transactions on Consumer Electronics*, Vol. 43, No. 3, August 1997, , pp.386-393

[5] GNE-Game Networking engine, URL: http://www.gillius.org/gne/

[6] Rakkarsoft, URL: http://www.rakkarsoft.com/

[7] Vit Novotny, Dan Komosny, “Optimization of Large-Scale RTCP Feedback Reporting in Fixed and Mobile Networks”, ICWMC, *Proceedings of the Third International Conference on Wireless and Mobile Communications*, 2007, pp.85

[8] Burget, R., Komosny, D. “Real-time control protocol and its improvements for Internet Protocol Television”, *International Transaction on Computer Science and Engineering*, ISSN 1738-6438, 2006

[9] H. Schulzrinne, S Casner, “RTP: A Transport Protocol for Real-Time Application”, *INTERNET DRAFT*, October 20, 1993

[10] Pierre G. Paulin, Faraydon Karim, Paul Karim. “Network Processors: A Perspective on Market Requirements”, Proceedings *of Processor Architectures and Embedded S/W Tools*. Design, Automation, and Test in Europe pp. 0420

[11] Srinivas R. Chakravarthy and Saligrama R. Agnihothri , “A server backup model with Markovian arrivals and phase type services”, *European Journal of Operational Research*, Volume 184, Issue 2, 16 January 2008, pp. 584-609

[12] Frank Windhofer, Wenqi Wu, Minli Wang, Satyendra K. Singh, Janapriya Saha, Bustanur Rosidi and George Iliakis, "Marked Dependence on Growth State of Backup Pathways of NHEJ", *International Journal of Radiation Oncology, Biology, Physics*, Volume 68, Issue 5, pp. 1462-14701, August 2007

[13] M. Amiruzzaman, "RTP Based Network-Engine for Multimedia Backup Server" *Procedings of Conference KoSbe - 2007*. The Korean Society of Broadcast Engineering, November 3, 2007, Hana Square, Korea University, Seoul, Korea, pp.29-30

[14] Xingwei Wang, Lei Guo and Cunqian Yu "RLDP: A novel risk-level disjoint protection routing algorithm with shared backup resources for survivable backbone optical transport networks", *Computer Communications*, In Press, (Accepted Manuscript), Available online 13 October 2007.

[15] S. W. Ng, Edward Chan, "Evaluation of Equation-Based TCP-Friendly Congestion Control under Lossy Environment", *INTERNET DRAFT*.

[16] Olaf Landsiedel and Gray Minden, "MPEG-4 for Interactive Low-delay Real-time Communication", *Technical Report*, Dec. 2003.

[17] Timothy Sohn, Eiman Zolfaghari, "Experimentation of the Datagram Control Protocol", CS268 Class Project Report, 2002.

[18] Methodology for Testing Wireless LAN Performance with Chariot, *INTERNET DRAFT*, October 2003.