

Efficient Bandwidth Management Using Embedded Network Processor for Quality Distribution of Multimedia Over Mobile Networks

R.Pandian, V.Ramachandran and ²P.Seethalakshmi,
Department of CSE Anna University Chennai, Tamilnadu India
²SSN College of Engineering, Kalavakkam, India

Abstract: The main objective of this study is to implement an embedded bandwidth manager using network processor to improve the quality distribution of multimedia streams to internet enabled mobile clients. In this paper the bandwidth requirement for each media stream in mobile applications is converted into channel time requirement for that stream. The embedded bandwidth manager allots for each stream a share of channel time depending on its requirement relative to other flows and the stream accordingly regulates its packet transmission. As bandwidth and traffic characteristics of various flows in the network change, the channel time proportion allotted to the individual streams by the bandwidth manager varies dynamically providing minimum bandwidth requirements for each stream. Utilizing the high performance parallel processing features of network processor, the admission control and dynamic bandwidth management functions are implemented on the Intel IXP1200 network processor simulator. The six micro engines with four hardware threads for each are utilized for the modules Ingress classification, flow establishment, flow teardown, flow renegotiation, packet generation and dispatching. As these modules are processed in parallel, the quality distribution of multimedia over mobile networks is enhanced.

Key Words: Bandwidth manager, mobile networks, network processor, channel time

INTRODUCTION

With the advent of broadband wireless communication, information dissemination through storage and distribution of multimedia streams to wireless mobile devices has become a significant component of the Internet traffic. Video transmission over mobile networks is more challenging than the transmission over other conventional networks due to the changes in the network topology that result in a relatively shorter lifetime of the network paths and high transmission bit error rates during fading periods. While several techniques must be extended to introduce mobile multimedia streaming applications quality distribution is one of the key issues in supporting multimedia streaming in mobile networks. Most of the coding and transmission schemes proposed for media transmission over wireless channel employ layered coding with unequal error protection^[1,2]. A strong error protection scheme is needed to achieve an acceptable quality of video. Error protection is useful only if proper channel allocation is carried out and the multimedia stream is transmitted. Research in resource and channel allocation in wireless systems including both fixed bandwidth and variable sized bandwidth has received much attention in the literature^[3-5]. Hence

efficient channel allocation is one way to avoid error and protect video data.

In recent times, much effort has been put into solving the problems of transmitting multimedia streams over wireless networks as the network bandwidth has increased with the processor performance doubled and memory access latencies reduced. The disparity in relative improvements of network bandwidth, processor and memory performance makes the design of network systems that operate at link speeds quite challenging.

To meet simultaneously the demands of flexibility and high performance, an alternative to general-purpose processors is embedded network processors. The emerged network processors are AMCC's np7XXX^[6], Agere's payload plus^[7], IBM's power np^[8], Silicon Access's iFlow^[9], Motorola's C Port^[10] and Intel's IXP^[11] families of network processors. There is a significant interest in network processors that can handle packets arriving at high line speeds, with several commercial products coming on to the market recently^[12]. Network processors employ parallelism to overcome the latency due to memory limitations. Intel IXP 1200 includes six micro-engines, each supporting four hardware contexts which automatically switches to another context when the current context stalls on a memory operation.

The QoS requirements of multimedia applications include maximum throughput and minimum delay for media streams over mobile networks and solving the transmission problems due to unreliable and dynamic nature of wireless networks can be carried out using network processors. This study proposes an embedded bandwidth management scheme using IXP1200 to increase the processing speed for admission control and flow establishment that enhances the performance of multimedia transmission in mobile networks.

BANDWIDTH MANAGEMENT SYSTEM IN MOBILE NETWORKS

The cellular wireless communication network has its coverage area partitioned into cells, where each cell is served by a Base Station (BS). Several BSs are connected to Mobile Switching Centre (MSC), which manages all calls in a large geographical area during call setup and handoff. The wireless Multimedia Management System shown in Fig. 1 allows a BS to communicate with many multimedia servers concurrently whose link is wireless. The system consists of K scalable server nodes and N wireless clients. The system can contain a variety of media servers depending on the needs of the mobile hosts. In this model all mobile hosts communicate to the media servers through the base stations and unlike fixed networks most of the time there is only one route between the base station and a mobile host. As the base station connects to many mobile hosts, special roles such as channel allocation and dynamic bandwidth management are assigned to base station. By using network processor at the base station, processing speed for admission control and flow establishment is increased thereby providing better multimedia service to the clients. The embedded bandwidth manager resides in a base station whose IP address is well known in the wireless network. It is assumed that the network has a set of media flows. Each media flow M is identified by its media flow_id which has source IP address, source port number, destination IP address, destination port number. Each media flow has a minimum bandwidth requirement $B_{\min}(m)$ and maximum bandwidth requirement $B_{\max}(m)$ that are converted as minimum channel time $CT_{\min}(m)$ and maximum channel time $CT_{\max}(m)$ requirement. The new flow in the server sends the Bandwidth Manager a request message containing its flow_id, $B_{\min}(m)$ and $B_{\max}(m)$. From these details the embedded bandwidth manager calculates the channel time $CT_{\text{allot}}(m)$ that could be allocated and the flow is added to the set of flows. This allocated channel time is communicated to the server and the admitted flow calculates its transmission rate. The

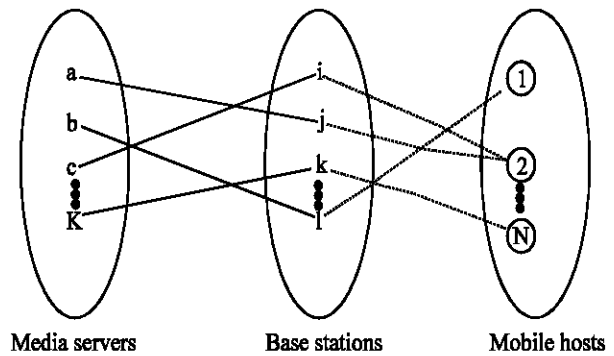


Fig. 1: Multimedia network system with mobile clients

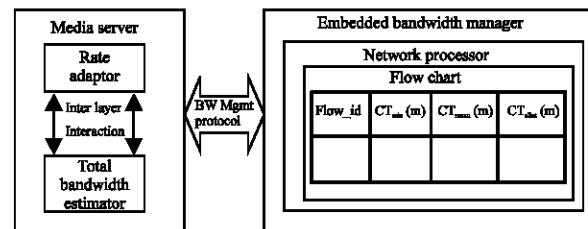


Fig. 2: Embedded bandwidth management system

server transmits using this transmission rate until it stops or a new $CT_{\text{allot}}(m)$ is allocated to it.

Architecture: The architecture of the embedded bandwidth management system consists of three major components as shown in Fig. 2.

- The media server performing rate adaptation and bandwidth estimation
- The embedded bandwidth manager
- The bandwidth management protocol

Media server: The media server transmits the multimedia data to the mobile client after its negotiation with the bandwidth manager in accordance with its bandwidth requirements. The server regulates its bandwidth consumption in accordance with the channel time allotted to it and controls the packet transmission rate. The bandwidth estimator at link layer performs total bandwidth estimation for all media flows sourced at the server by measuring the bandwidth for each media flow. When the total bandwidth perceived by a media flow changes its Rate Adaptor at the application level re-negotiates with the Bandwidth Manager for the channel time as the quality of the media is associated with the bandwidth.

Embedded bandwidth manager: The embedded bandwidth manager performs admission control at the

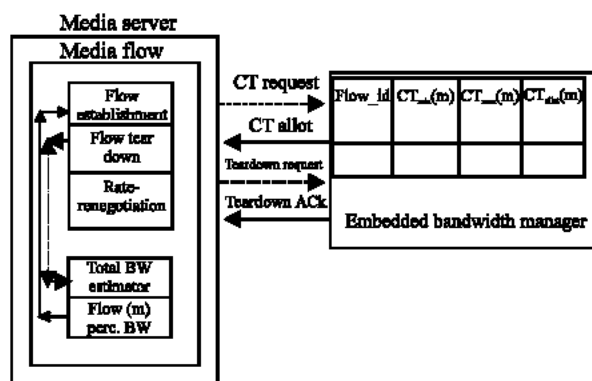


Fig. 3: Interactive bandwidth mangament protocol

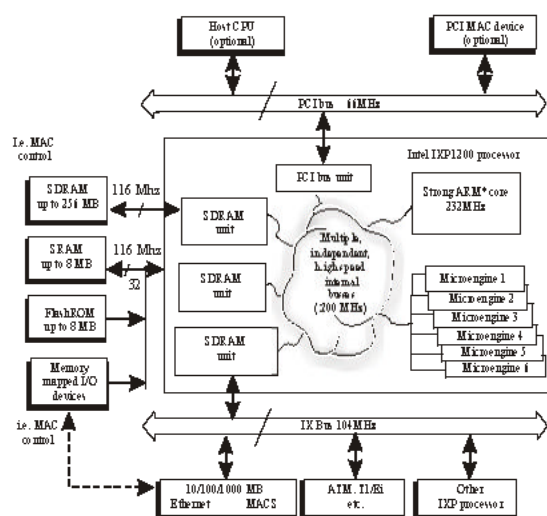


Fig. 4: Intel IXP1200 block diagram

time of media flow establishment and bandwidth redistribution at the time of media flow teardown. The bandwidth manager maintains a flow table with the contents flow-id, maximum and minimum channel requirement and the available bandwidth. Once a media flow is admitted the bandwidth manager allocates at least minimum channel time based on max-min fairness algorithm. The max-min algorithm tries to satisfy maximum channel time requirements for as many flows as possible and the remaining bandwidth is allotted to the incoming flows whose channel time proportion is between their maximum and minimum request. The BM also performs re-negotiation either when some flow detects a change in its perceived bandwidth or when its traffic characteristics change. The BM rejects a flow if it cannot support even its minimum channel time requirement. The channel time remaining as unallotted after admitting all media flows is also allotted fairly for the further incoming media flows.

Interactive bandwidth management protocol: The interactive bandwidth management protocol is used for the interactions between the server and the embedded bandwidth manager. The BM is invoked at the time of media flow establishment, media flow teardown and significant change in media flow's traffic pattern. The notation used in the description of the protocol and the actions that occur when the events happen is shown in Fig. 3.

The media flow interacts with the Bandwidth Estimator periodically to identify the available bandwidth. From the available bandwidth it calculates its minimum and maximum requirements and negotiates with the BW manager.

NETWORK PROCESSOR AS EMBEDDED BANDWIDTH MANAGER

The Intel IXP1200 network processor is used as a bandwidth manager that has six main functional units that are described briefly below as in Fig. 4

- **StrongARM core** - A full 32-bit RISC processor core with integrated caches that can be used for management functions, running routing protocols, exception handling, and other tasks.
- **Six Micro-engines** - These compact, efficient RISC engines are used for any function requiring high-speed packet inspection, data manipulation, and data transfer. These are fully programmable 32-bit engines with a 5-stage execution pipeline and a large (256) register set. Hardware multithreading and context-sensitive registers enable very fast context switching.
- **SDRAM Unit** - A shared, intelligent memory interface that can be accessed by the StrongARM core, the Microengines and devices on the PCI bus. Capable of moving blocks of data between the SDRAM and the Microengines or the IX Bus unit or devices on the PCI Bus.
- **SRAM Unit** - A shared, intelligent memory interface that is accessible by the StrongARM and the Microengines. Capable of moving blocks of data between the SRAM and the Microengines.
- **PCI Bus Interface Unit** - A standard interface that may be used to interface other PCI devices or another host processor.
- **IX Bus Unit** - An intelligent data movement engine controlled by the Microengines that is capable of transferring blocks of data between the IXP1200 and networking devices such as MACs and SARs. While the IX Bus Unit can perform many sophisticated data transfer operations internal to the IXP1200, the (external) IX Bus Interface itself is simple, and it is straightforward to design peripherals for it and other devices.

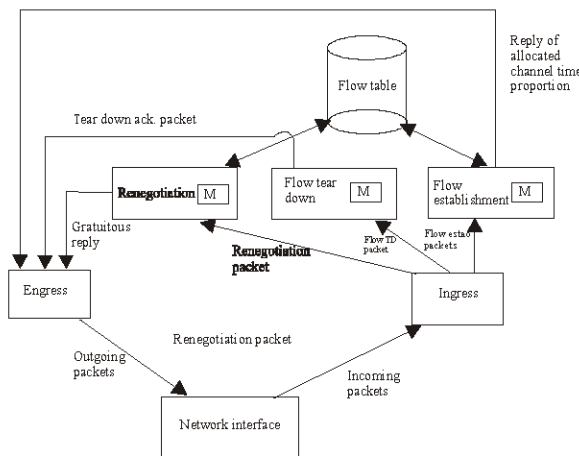


Fig. 5: Architecture of embedded bandwidth manager

The Bandwidth Manager is designed using network processor which can be embedded in any network domain. The various modules in the BM are assigned to each microengine. The modules and data used in the bandwidth manager are shown in Fig. 5. It consists of an Ingress classifier Module for classifying the incoming traffic of different services and to move into further modules. A Flow establishment Module establishes a new flow if the requested flows minimum bandwidth requirements are satisfied and updates the same on the flow table. The Flow Teardown module removes the flow from the existing flows on completion of transmission and updates on the flow table. The Flow Renegotiation Module negotiates new channel time for the existing flows when there is a demand for additional bandwidth. The Egress module sends response and acknowledgement packets for incoming packets to the network. Ingress packet classifier does the Packet Classification & Queuing. Basically it reads packet from the signaling network and classifies whether the incoming packet is a flow establishment packet or a flow teardown packet or a flow Renegotiation packet. Once the packets are classified they are put into its appropriate queue namely flow establishment queue, flow teardown queue and flow renegotiation queue. Egress processing module has packet generator that has an input queue feeding it from one side. This queue contains records, which have enough information as to the type and contents of the packet to be created. The different types of packets created are

- Flow establishment - reply Packets
- Flow Teardown - Acknowledgement Packets.
- Renegotiation - reply Packets.

Once the packets are created they are put into their corresponding queues and a dispatcher then dispatches these queues in a round-robin manner and the bandwidth manager controls the flows in wireless networks.

RESULTS

The admission control and dynamic bandwidth management for video over mobile adhoc network is achieved by the bandwidth manager implemented using Intel IXP1200 Network Processor having six micro engines with four hardware threads for each microengine. The Ingress packets that are received from the network and Egress packets that are to be dispatched over the network are stored in the SDRAM with a considerable buffer size. The flow table containing the unique id of the existing flows is implemented with SRAM. The flow establishment rate, establishment latency and resource utilization are calculated to determine the system performance. The latency distribution of SRAM, SDRAM, FBICSR, Scratchpad RAM and resource utilization of various microengines are analyzed. Most of the accesses take 20 cycles to complete and the minimum latency for accessing the FBI CSR is 9 cycles. The minimum latency for SDRAM accesses is 40 cycles and the maximum is 100 cycles. The minimum latency for accessing SRAM is 16 cycles and maximum is 80 cycles. The utilization of the microengines for flow establishment is 58.6%, flow teardown and ingress classifier module utilizes 57%, packet generation module utilizes 55.2% and dispatcher module utilizes 53.8%. The high level of parallelism and the RISC architecture gave a boost in the performance of the bandwidth manager.

CONCLUSIONS

The admission control and dynamic bandwidth management scheme for mobile network is achieved by simulating the real world traffic with mixed incoming signals (Flow establishment request, flow teardown request, Renegotiation request and erroneous signals) by implementing the Embedded Bandwidth Manager using Intel IXP1200 workbench simulator and the performance of the system was excellent and satisfied. The Intel IXP1200 network processor was found suitable for the implementation of embedded bandwidth manager as it has all necessary on board facilities that are required by the Bandwidth Management system. The high level of parallelism and the RISC architecture gave the system a significant boost in performance.

REFERENCES

1. Nahrstedt, K. Described Quality of Service in networked Multimedia systems and applications CRC press pp: 217-252.
2. Wu, L., R. Sharma and B. Smith, 1997. Thin streams: An Architecture for Multicasting Layered Video in proceedings of 7th International workshop NOSSDAV,97, USA.
3. Boukerche, A., S. Hong and T. Jacob, 2000. A distributed dynamic channel allocation Algorithm for mobile communication systems. In proceedings IEEE Eighth International symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2000), pp: 73-81.
4. Boukerche, A., S. Hong and T. Jacob, 2002. A distributed algorithm for dynamic channel allocation mobile networks and applications, 7: 115-126.
5. Georganas, N.D., R. Steinmetz and T. Nakagawa, 1996. Synchronization issues in multimedia communication IEEE J. Selected Areas in Communication, 14: 1-3.
6. AMCC. AMCC's np7 xxx Series of network processors.
7. Agere. Agere's payload plus family of network processors <http://www.agere.com/enterprise-metro-access/network-processors.html>
8. IBM. Ibm powernp Network processor <http://www.3.ibm.com/chips/Ibm-powernp-Np4>.
9. Silicon Access iflow family of processors <http://www.siliconaccess.com>
10. Motorola. The Motorola.c port family of network processors. <http://e-www.motorola.com>
11. Intel. Intel ixp family of network processors. <http://www.intel.com/index.html>
12. Intel Corporation .IXP 1200 Network Processor. Data Sheet , September 2000.