Research Article

Journal of Atoms and Molecules
An International Online Journal
ISSN – 2277 – 1247

COLLABORATIVE APPLICATIONS OF OVERLAY NETWORKS: A PERSPECTIVE.

Namachivayam A*
Department of Computer Science and Engineering, College of Marine Science and Technology, Massawa, Eritrea.

Received on: 14-01-2012 **Revised on: 07-02-2012** **Accepted on: 27-02-2012**

Abstract:

The main objective of this research is to minimize delay in collaborative applications by setting tight bounds. Examples of collaborative applications include video-conferencing, distributed database replication and online games. The efficiency of this work is analyzed and shown to be superior to its counterparts in the execution time. To provide an efficient heuristic to obtain a multicast sub network on an overlay network, given a source and a set of destinations that is within a specified maximum delay and a specified maximum variation in the delays from a source to the destinations.

Key Words:

Multicast routing, Delay and delay variation, Overlay networks

* Corresponding author

Namachivayam.A,
Email: namachi_siva@yahoo.co.in
Tel: + 00291 – 7226136

Introduction:

An approach to ensure minimum delay variation is to buffer the messages at different nodes in the overlay network. Buffering at the source node requires the source node to keep additional information for each destination. The source node will buffer a message for a different amount of time for each destination and transmit the message multiple times over the network; clearly, this is a waste of network bandwidth[2]. Buffering at the source node defeats the purpose of multicasting,

which is one of conserving network bandwidth. Buffering at intermediate nodes requires some nodes to be identified as core nodes in the network. Messages are buffered at these core nodes before they are sent to the destinations. Buffering at destination nodes requires each destination node to buffer the messages before they pass the messages to the application process. In this approach, the source node informs the destination nodes when they can process the received packets[3].

Methodology:

Hardware specification

Processor	:	Any Processor above 500 Mhz
Ram	:	128Mb
Hard Disk	:	20 Gb
Compact Disk:		650 Mb
Input device	:	Standard Keyboard and Mouse
Output device:		VGA and High Resolution Monitor

Software specification

Operating System:		Windows 2000 server Family
Techniques	:	JDK 1.5.0.01
Data Bases	:	Microsoft SQL Server 2000
Front End	:	Java Swing

TCP/IP Stack

The TCP/IP stack is shorter than the OSI one.

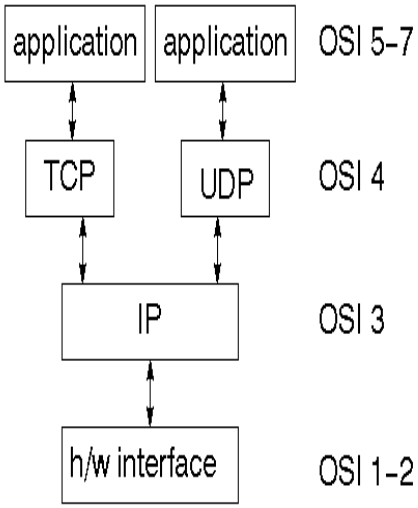
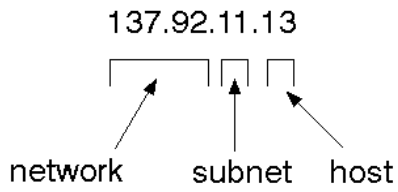


Figure 1: TCP - UDP Protocol

IP Datagram

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers (Fig 1). The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end[1].

Total address



The 32 bit address is usually written as 4 integers separated by dots.

Sockets

A socket is a data structure maintained by the system to handle network connections. A socket is created using the call socket. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with Read File and Write File functions.

```
#include <sys/types.h>
#include <sys/socket.h>
int socket (int family, int type, int protocol);
```

Here "family" will be AF_INET for IP communications, protocol will be zero, and type will depend on whether TCP or UDP is used. Two processes wishing to communicate over a network create a socket each. These are similar to two ends of a pipe[6].

Results and Discussion

Implementation is the stage of the work when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods[5].

Implementation is the process of converting a new system design into operation. It is the phase that focuses on user training, site preparation and file conversion for installing a candidate system. The important factor that should be considered here is that the conversion should not disrupt the functioning of the organization (Fig 2). The implementation can be preceded through Socket in java but it will be considered as one to all communication. For proactive broadcasting we need dynamic linking. So java will be more suitable for platform independence and networking concepts. For maintaining route information we go for SQL-server as database back end[1].

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner[2].

Implemented all the heuristics and observed Chains outperform DPDVB and DVMA in terms of execution time. The Chains heuristic also achieves the tightest delay variation bound along with DPDVB. The results are also presented to show that finding k shortest paths for all destinations is not a bottleneck in the solution. It is also observed that Chains require higher values of k to achieve the tightest delay variation when the graph becomes dense or when the end-to-end delay bound increases (Fig 3). For dynamic reorganization of the multicasting sub network with the tightest delay variation and bounded delay, it is noticed that the solution with Chains is more efficient than that of DVMA in terms of time-complexity[5].

In this research, the problem is considered of determining a multicasting sub network with end-to-end delay bound and delay variation bounded for collaborative applications on overlay network. As part of the future research work, link delays are considered as time-varying functions and required to develop an efficient heuristics for the DVBMN problem.

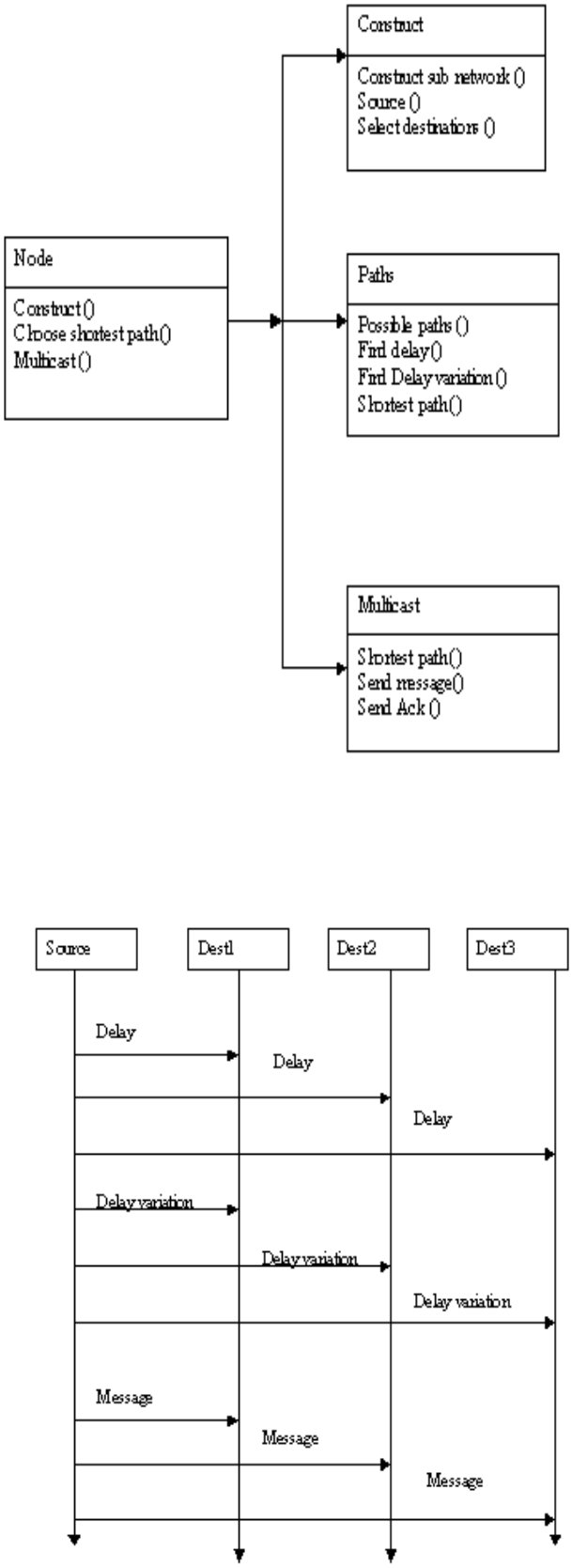
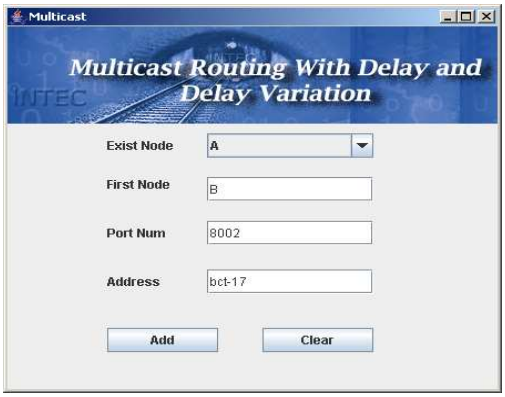


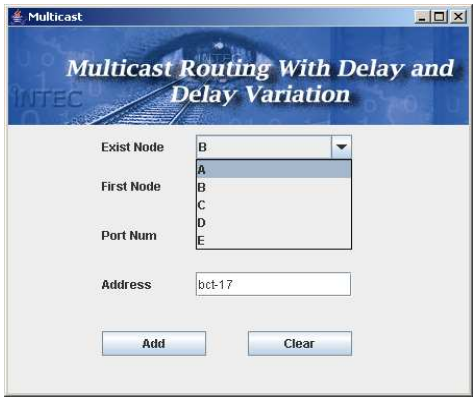
Fig 2: Class and Sequence of Multi cast overlay network

Screen 1

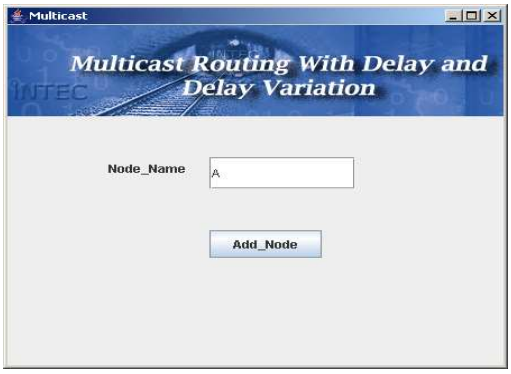
Screen 2



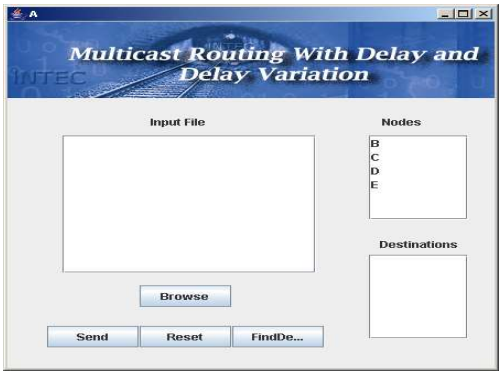
Screen 3



Screen 4



Screen 5



Screen 6

Fig 3: Sample Screens of Minimization of Delay and Delay Variance

References

1. Ballardie. (1997) “Core Based Tree (CBT) Multicast Routing Architecture,” Internet RFC 2201.
2. Baneerjee, C.Kommareddy, K.Kar, B.Bhattacharjee, and S.Khuller. (2003) “Construction of an Efficient Overlay Multicast Infrastructure for Real-Time Applications,” Proc. IEEE INFOCOM.
3. Banik, S.Radhakrishnan, and C.N.Sekharan. (2004) “Multicast Routing with Delay and Delay Variation Constraints for Multimedia Applications,” Proc. Seventh IEEE Int’l Conf. High Speed Networks and Multimedia Comm., pp. 399-411.
4. Bharath Kumar and JM Jaffe. (1983) “Routing to Multiple Destinations in Computer Networks,” IEEE Trans. Comm., vol. 31, no. 3, pp. 343-351.
5. Brosh and Shavitt. (2004) “Approximation and Heuristic Algorithms for Minimum Delay Application-Layer Multicast Trees,” Proc. IEEE INFOCOM.
6. Keshav (1997) “An Engineering Approach to Computer Networking” Addison-Wesley, Reading, MA.

