Gigabit Networking: Digitized Radar Data Transfer and Beyond

Sangeetha L. Bangolae, Anura P. Jayasumana, V. Chandrasekar Department of Electrical and Computer Engineering Colorado State University Fort Collins, CO 80523, USA Email: {sang, anura, chandra}@engr.colostate.edu

Abstract - Gigabit networking makes possible the remote access to expensive and specialized facilities that were inaccessible in the past due to limited bandwidths. VCHILL project for digitized radar data transfer is one such real-time application that will tap the Next Generation Internet technology to provide interactive access to real-time and stored data generated by weather radars, thus revolutionizing the way experiments are carried out. The design and implementation of the UDP-based virtual radar application and the deployment of an efficient congestion control algorithm for this application are presented. The TCP-friendly Rate Adaptation Based On Loss (TRABOL) algorithm is a source-based rate control mechanism that controls the transmission rate based on the feedback about losses experienced by the client station. The performance results show that the deployment of this algorithm makes the application TCPfriendly.

Keywords – High-bandwidth application; Radar data transfer; gigabit network; NGI.

I. INTRODUCTION

With gigabit networking, it is now possible to deliver huge amounts of data to multiple users with low latency. Its impact can be seen in enhanced network-based applications in areas such as graphic visualization, virtual reality, desktop video, CAD/CAM design, and satellite image transfer. Gigabitnetworking technology is thus revolutionizing not only the data delivery process, but also the way research and experimentation are carried out in many fields.

A crucial aspect of the Next Generation Internet (NGI) initiative is gigabit-networking applications. Digital Earth [14], Biomedical Tele-Immersion [15], and Virtual CHILL [13] for Digitized Radar Data Transfer are some of the NGI based projects that have become viable with these data rates. NASA has demonstrated a few prototype gigabit applications, including Virtual MechanoSynthesis (VMS) and Digital Sky projects [12]. VMS is a 3-D simulator that allows users to see, move and even feel molecular structures, while Digital Sky is an astronomical data treasure that will enable viewing of tens of terabytes of archived space images created from data stored at remote sites.

Some of the gigabit applications have to rely on UDP to meet their real-time requirements since retransmissions in TCP introduce unacceptable delays, and characteristics of window-based flow-control at the transport layer may not match the requirements of the application. The tolerance to data loss is application dependent and hence, with certain applications, error-recovery may be achieved more efficiently at the application level. However, UDP protocol does not implement congestion control. Since TCP-based applications constitute a majority of the Internet traffic, it is imperative that these emerging non-congestion controlled (UDP based) applications, especially those demanding high bandwidths, co-exist in a 'friendly' manner with the TCP traffic. This requires the UDP applications to implement a form of end-to-end congestion control mechanism by which congestive collapse could be prevented in the Internet.

In this paper, we discuss the design and implementation of the digitized radar data transfer protocol for Virtual CHILL (VCHILL) radar project. This project involves the transfer of data generated by the radar over the NGI to remote sites as well as application programs for remote radar operation. It thus aims at a paradigm shift in radar and remote sensing research by exploiting the Internet technology to make the data available remotely in real-time, and also facilitate virtual control of the radar operation [13]. The deployment of the TCP-friendly Rate Adaptation Based On Loss (TRABOL) congestion control algorithm overcomes the limitations of TCP and UDP for the radar application. TCP-friendliness of TRABOL is evaluated in [2]. The data transfer model currently underway has demonstrated sustained transfer speeds of about 600 Mbps within a campus network.

Section II of this paper outlines the design of the digitized radar data transfer protocol while section III discusses the deployed congestion control mechanism and its performance evaluation. We conclude and present the future work in section IV.

II. VCHILL (VIRTUAL CHILL) RADAR

The CSU-CHILL Radar system [3,13] is a state-of-the-art national radar facility routinely used by researchers around the United States for remote sensing of atmospheric phenomena. The radar system consists of two transmitters and two receivers for vertical and horizontal polarization. The transmitter transmits a pulse that is reflected back by the target object and received by the receiver. The received signal is then sampled, and the sampled signal is referred to as the Digitized

This research was supported in part by DARPA Next Generation Internet (NGI) and NSF Information Technology Research (ITR) grant no. 0121546

Radar Signal (DRS). The sampling rate determines the realtime bandwidth requirement for the radar operation. Currently, the required data rate could vary between 64 Mbps and 384 Mbps. The collection of all the samples between two pulses is termed as a 'ray'. A complete set of rays constitutes a 'sweep'.

The objective of the VCHILL project is to allow remote users connected by high-speed networks to access the radar data and perform signal calculations for research and educational purposes at their local sites. Before the onset of NGI, researchers had to take extensive time off their activities to be at the radar site to gather the gigabytes of data and transfer it using tapes. NGI thus provides an opportunity to not only make the data widely and remotely available, but also to reform the radar control operations to achieve a new geographically diverse interactivity [13]. It is also essential to meet the application-level QoS demands. Although the end-toend transport protocol has to provide the functionality desired to the applications with sufficiently high bandwidth, low latency and low overhead, even with a gigabit backbone, it becomes complicated due to the exponentially increasing number of applications and users demanding the resources on the network.

A. VCHILL DRS Transfer Architecture

Fig. 1 depicts the block diagram of the radar application showing transfer of DRS data over the NGI. In the DRS server, the data generated by the radar is acquired by the data acquisition process and transferred over the gigabit link by the data transmission process using UDP protocol. At the remote end, the DRS client receives the data through the data receive process while the signal parameter estimation thread processes the data received and calculates the different radar signal parameters and passes them to the graphical display unit for display to the end user. In addition to gigabit link speeds, the server and client machines being used should be capable of handling huge amounts of data and be able to do highperformance computations [6,7].

The goal of the virtual radar application design is to achieve DRS data transmission over gigabit networks. At the remote end, the data is received and the various radar signal parameters that represent the atmospheric phenomena are calculated using complex digital signal processing algorithms and displayed by the display unit. The required applicationlevel QoS such as high bandwidth and real-time operation at the end user poses a challenge for the implementation. Our objective is in developing a generalized protocol that will be suitable for a variety of radar systems. The initial implementation in this initiative consisted of the 'archived' compressed data transmission using the TCP transport protocol. This implementation marked a major milestone in the process of realizing the end goal.

Extensive Research is being done for real-time multimedia applications such as audio and video conferencing that require relatively high bandwidth, low end-to-end delay and smoothness of delay variation. However, the characteristics of importance of real-time radar data transfer application are different and are summarized as follows:

- High-Bandwidth requirement for best operation; hence a high responsiveness to available bandwidth.
- Satisfactory operation with a minimum bandwidth threshold possible; yet increase in bandwidth provides a better display image.
- Tolerance to losses and end-to-end delay high, compared to audio and video streaming media.
- Smoothness of jitter not critical for proper functioning, unlike with media applications.

UDP transport protocol is used for the digitized radar data transfer; although it is unreliable, the absence of retransmissions makes it suitable for real-time operation. The UDP-based application [1], however, may strain the underlying network infrastructure by utilizing most of the available bandwidth and starving the neighboring TCP streams. Hence, there arose a need for an efficient congestion control mechanism in the application. The control loop in the reverse direction (fig. 1) aids in the implementation of the congestion control algorithm, which is based on the feedback sent by the receiver to the source about the network status and receiver capability. The radar display image consists of rays, and this allows us to send the feedback on a ray-by-ray basis.



Fig. 1. Block Diagram of digitized radar data transfer over the Internet



Fig. 2. Radar display image at the client end with no loss

Weather conditions such as storm, hail, snow, and clear-air in the region of coverage of the radar could be determined from the radar signal parameters. The radar image in fig. 2 shows an important radar signal parameter called *reflectivity* that signifies the power of the target. This image thus contains data that helps relate the radar data to the real world [9]. The color bars on the image indicate different magnitudes of the parameter. This is one of the many such parameters of interest derived from the radar signal data. Although 5 to 10 % loss could be tolerated in the transfer of the data from the server to the client end, huge amount of losses renders the data useless and this condition of 90% losses is depicted in fig. 3. The losses that occur during the transmission could be due to congestion in the network, low receiver capacity or low bandwidth links.



Fig. 3. Radar display image at the client end with 90% loss

III. CONGESTION CONTROL AND TCP-FRIENDLINESS

As the number of non-congestion controlled applications increases, the need for some form of end-to-end congestion control mechanism such as the one in TCP, has increased, in order to maintain the stability in the Internet. This concept led to the study of TCP-friendliness of a UDP flow. TCP-friendly behavior means that a real-time flow exhibits convergence to fairness property over the same timescale as that of a neighboring TCP session, under the same network conditions, along with an efficient utilization of the available bandwidth. With the inception of the idea of TCP-friendliness came several proposals [5,8,11] for congestion control of non-TCP based transfers, especially real-time audio and video streaming applications. A method to achieve fairness between TCP and UDP would be the implementation of packet scheduling algorithms such as class based queuing (CBQ), and weighted fair queuing (WFQ), or resource reservation mechanism that involve control or changes to be made at the intermediate nodes. Application based control is an alternative and easier mechanism that involves control at the end points only. In this manner, the characteristics of the UDP application are taken into account, thus making it possible to obtain a degree of application-level QoS.

A. TRABOL Algorithm

A TCP-friendly Rate Adaptation Based On Loss algorithm (TRABOL) has been designed and implemented for the radar application to perform congestion control at the application-level. According to TRABOL algorithm, the transmission rate

of the server is continuously adjusted, in a TCP-friendly manner, based on losses in the network. When there is congestion, the server backs off aggressively and probes for any available bandwidth once congestion is not noticed. The sending rate increase and decrease policies are applied according to AIMD algorithm, which has been shown to efficiently converge to fairness [4].

An application level packet consists of a ray header and a part of the ray data. Each ray is thus divided into a number of data packets and sent over the network to the client end. A feedback packet is sent by the receiver for every ray of data received. Based on the feedback message about loss rate, the sender decides the transmission rate and sends data thereon at that rate. Considering that the loss rate fairly completely characterizes the system, and an accurate estimate of the loss rate is provided by the destination, this scheme does not rely on the round-trip time to estimate congestion. However, the implementation can be further extended to accommodate the information provided by RTT variation, if necessary.



Fig. 4. Radar Data packets transfer with congestion control.

The design criteria for severe-AIMD could be summarized as:

- 1. Decrease the sending rate aggressively upon congestion, by D-factor.
- 2. Increase the sending rate aggressively when the congestion indication returns a 0, by I-factor.

The decrease and increase policies are applied by either increasing or decreasing, the gap between groups of packets in ray and the number of packets sent in a ray. The inter-group gap and the number of packets to send in a group are determined by the corresponding transmission rates. This functionality is shown in fig. 4 wherein a delay is introduced between groups of packets at the sender end.

Packets are thus formed and sent to the receiver over the network and the receiver upon receiving the packets in a ray sends a feedback message to the sender. The feedback manager at the sender end collects and processes feedback information, while the congestion manager applies the specific TRABOL algorithm based on severe-AIMD. Let 'L_r' represent the loss rate calculated at the sender end based on the feedback received from the destination. This loss rate is compared with the pre-defined threshold loss rates, which helps in checking the occurrence of congestion.

If ' L_r ' is greater than the maximum loss rate, MAX_LOSS, then the sending rate is decreased aggressively by D-factor. The drop occurs during the beginning of the 'congestion cycle' and is followed by an increase policy determined by the I-factor if subsequent losses are within the threshold. If the loss rate on the other hand is less than or equal to the minimum loss rate, MIN_LOSS, it is checked if there was a congestion event (marked by a decrease policy) before or not.



Fig. 5. (a) Sending Rate and Loss Rate at the destination with no rate control. (b) Loss rates at the destination with and without rate control. (c) Transmission rate (Mbps) and Loss rate (%) of the UDP-based radar application with rate control, D-factor = 0.45 and Initial Increase Rate = 20 Mbps.



Fig 6. Description of congestion events

If there was a decrease policy, an increase policy given by Ifactor is applied; otherwise the rate is maintained constant. Dfactor is a value chosen between 0.3 and 0.45 for aggressive reduction of transmission rate. These congestion events are described using the equations that follow in fig. 6. If IIR is the Initial Increase Rate, the I-factor is calculated using the formula given in (2) and if the current rate of transmission is R_{old} , then the new rate R_{new} is given by (3). Using the severe-AIMD approach, it can be shown that a neighboring TCP flow will get a fair share of bandwidth if it exists over a time period at least equal to one 'congestion epoch'. Thus we can argue that this end-to-end congestion control algorithm is efficacious, since it ensures convergence to fairness and efficiency at least for certain timescales.

B. Performance Evaluation

We performed experiments to evaluate the proposed congestion control scheme and to determine the TCPfriendliness of the radar application. The performance test topology consisted of a high-speed source machine connected to an equally high-speed receiver over an *internal* gigabit link. The maximum transmission rate attained is about 650 Mbps for UDP-based data transfer. Some of the performance results are shown in fig. 5. Figure 5(a) shows the server rate in Mbps and the loss rate in % as a function of time for the case when there is no rate control at the server end. The server throttles the receiver and the network by sending data at the maximum rate possible, in this case about 500 Mbps. At the destination, we perceive a fast moving poor quality radar image. Swamping of the Internet resources occurs since there is no congestion control. The overall quality is very low, and the loss rate is almost always above 35%. The vast difference in the loss rates between the cases when there is rate control and when there is no rate control is illustrated in fig. 5(b). When the rate control is on, the loss rate consists of short living peaks not exceeding 25%, but for the initial transient. The reason for a decrease in the loss rate is also due to the fact that during congestion in the network, even if one packet is lost in ray, the entire ray is lost, while the TRABOL algorithm ensures that the ray can still be reconstructed with as many packets as that arrive at the destination. Due to this, there is an increase in the performance of the application with the deployed algorithm.

The implementation of the TRABOL algorithm using severe-AIMD to adapt the server rate is depicted in fig. 5(c) where we show the transmission rate of the server in Mbps and the corresponding loss rates in % as a function of time in seconds. The target rate that is the pre-defined maximum rate at which the application starts the data transfer is set as 150 Mbps and the minimum rate as 45 Mbps. MIN_LOSS was 5% and MAX_LOSS was 10%. Since smoothness is not required during rate adaptation, the algorithm is highly responsive towards any available bandwidth and highly aggressive to perceived congestion. Consequently, the transmission rate oscillates between the target rate and the minimum rate.

To evaluate the TCP-friendliness of the DRS data transfer application, we created an internal network as shown in fig. 7 having a bottleneck link with an available bandwidth of about 92 Mbps.



Fig. 7. TCP-friendliness test bed configuration

 TABLE I

 Average Throughput Of A Tcp Flow Under Various Conditions

Flow CD	Throughput of flow AB
TCP - Netperf	46 Mbps
UDP - Netperf	8 Mbps
UDPunctrl	17 Mbps
UDP _{ctrl}	43 Mbps

Consider two streams of traffic from A and C, going to B and D respectively during the same time period. Under ideal conditions, fair allocation of bandwidth between the two streams would mean that the bottleneck link bandwidth is shared equally. Each stream should thus end up with an average bandwidth of about 46 Mbps. Throughout the experiments, a TCP stream was run between hosts A and B while a TCP or a UDP stream was run in parallel between hosts C and D and the bandwidth utilization by the TCP stream between hosts A and B is analyzed using *Netperf.* Care was taken to set the socket buffer sizes and message size the same in both TCP and UDP applications.

The results of the experiments are tabulated in table 1, which shows the average throughput of the TCP flow AB under four test conditions. When both the streams AB and CD are TCP-based, the bottleneck link bandwidth is shared equally between them, because of the efficient congestion control algorithm implemented in TCP. When an UDP-based *Netperf* stream is run between C and D, the throughput of the TCP flow AB is around 8 Mbps, and when the non-congestion controlled radar application (UDP_{unctrl}) is run, this throughput is about 17 Mbps. But when the congestion-controlled radar application (UDP_{unctrl}) is run between C and D, the throughput of the TCP flow AB increased to about 43 Mbps over a period of 18 seconds. Thus we conclude that the radar application with the congestion control is TCP-friendly.

The experiences with the design and implementation of the UDP-based server and client applications for digitized radar data transfer over the NGI and the deployment of a congestion control mechanism for this application have shown that there is room for major enhancements in these areas. With the emergence of gigabit networking applications that will revolutionize the access to vast databases, has arisen the necessity to tune the end-to-end transport protocols. TCP's slow start and congestion avoidance mechanisms will perform poorly for short-time flows on a gigabit link, while UDP without a congestion control mechanism would swamp the resources on the Internet. Hence there is a need for a better, generalized congestion control algorithm, which should include the advantages of both TCP and UDP protocols

IV CONCLUSION AND FUTURE WORK

In this paper we discussed the emerging gigabit networking applications and in particular, the VCHILL radar project. The end goal of the project is to transfer the digitized radar data over the Next Generation Internet in real-time. To achieve this goal, a UDP-based application has been developed. Due to the unfairness this non-TCP based high bandwidth data transfer application shows towards a neighboring TCP stream, there was a need to deploy an end-to-end congestion control mechanism. A TCP-friendly Rate Adaptation Based on Loss (TRABOL) algorithm has been implemented and tested for the VCHILL application. This algorithm is an application-level, source based rate control mechanism using the AIMD approach and it has been shown to be TCP-friendly.

As the number of real-time applications besides audio and video streaming media is increasing, a general algorithm

similar to the RTP/RTCP protocol [10], which is more suitable for audio and video streams, would considerably benefit these next-generation network applications. Also, for scalability, the application should migrate to IP Multicasting to be able support a large number of clients without overloading the server. The phenomenon of TCP-friendliness has seen a number of researches in the recent past and in their works, TCP-friendliness means 'a non-TCP based flow should receive the same average bandwidth as that of a corresponding TCP flow traversing along the same link'. In future, if there is an application-limited bandwidth issue rather than a networklimited bandwidth issue, this statement has to be redefined. A proportional bandwidth allocation needs to be deployed for fairness rather than an equitable share among all the existing flows. The digitized radar data transfer application represents a class of emerging Internet applications that cannot function well over TCP. This class of applications differs from other next-generation network applications in its requirements and we aim at generalizing the rate control algorithm and hence the TCP-friendliness constraint for all such applications.

References

- [1] S. Bangolae, "End-to-end congestion control mechanism for Real-time high-bandwidth applications: A case study with digitized radar data transfer", Master's Thesis, Colorado State University (In progress).
- [2] S. Bangolae, A. Jayasumana, and V. Chandrasekar, "TCP-friendly congestion control mechanism for a UDP-based "high-speed application and characterization of fairness", IEEE ICCS 2002, Nov. 2002.
- [3] D. Brunkow, V. N. Bringi, P. C. Kennedy, S. A. Rutledge, E.A. Mueller, and R. K. Bowie, "A description of the CSU-CHILL National Radar Facility", Journal of Atmospheric and Oceanic Technology, Vol. 17, pp. 1596-1608, 2000.
- [4] D. Chiu and R. Jain, "Analysis of the increase and decrease algorithm for congestion avoidance in computer networks, Journal of Computer Networks and ISDN, 17(1):1-14, June 1989.
- [5] S. Floyd, M. Handley, J. Padhye, and J. Widmer, Equation-Based congestion control for unicast applications", SIGCOMM 2000, August 2000.
- [6] S. Kaiya, "Radar data transfer protocol for display of CSU-CHILL digitized radar signals", Master's Thesis, Colorado State University, May 2002.
- [7] A. Kumar, "TCP, UDP, and IP Multicast based applications for Real-Time access of CSU-CHILL radar data", Master's Thesis, Colorado State University, Feb. 2000.
- [8] R. Rejaie, M. Handley, D. Estrin, "RAP: An end-to-end rate-based congestion control mechanism for real-time streams in Internet", In proc. of IEEE INFOCOM 1999, March 1999.
- [9] R. E. Renehart, "Radar for meteorologists" ISBN 0-9658002-0-2: 1999.
- [10] H. Schulzrinne, S. Cassner, R. Frederick, and V. Jacobson, RTP: A Transport Protocol for real-time applications; IETF, RFC 1889 (Standards Track), January 1996.
- [11] D. Sisalem, H. Schulzrinne, "The Loss Delay Based Adjustment Algorithm: A TCP friendly adaptation scheme", In proc. of NOSSDAV, Cambridge, UK, July 8-10 1998.
- [12] Tomorrow's Networking technologies, HPCC NASA, Website: <u>http://ct.gsfc.nasa.gov/brochures/2000/nren.htm</u>
- [13] "Virtual CSU-CHILL National Radar Facility", Website: http://www.engr.colostate.edu/ece/Research/vchill/vchill.html
- [14] "Visualizing the Digital Earth", Website: <u>http://www.ai.sri.com/digital-earth/</u>
- [15] "Bio-medical Tele-Immersion", Website: <u>http://www.nlm.gov/research/</u> ngisumphase2.html