19.11 Radar Networking: Considerations for Data transfer Protocols and Network Characteristics

Tarun Banka, Aditya Maroo, Anura P. Jayasumana*, V. Chandrasekar,
Nitin Bharadwaj, SathishKumar Chittibabu
Colorado State University, Fort Collins, CO

Abstract

QoS requirements for real-time time-series data transfer over Internet are significantly different from those of traditional applications like voice and video streaming. We evaluate and compare the efficacy of three transport protocols, TCP, UDP and TRABOL for radar data streaming. The impact of network dynamics on the quality of radar data is investigated as well. TRABOL, an application level transport protocol combined with a time-series sample selection scheme is proposed to meet the QoS requirements. Performance results obtained using radar emulation test-bed shows that proposed strategy meets the application QoS requirements without overly impacting the performance of other cross-traffic streams on the network.

I. Introduction

QoS requirements for real-time digitized radar data transfer are significantly different from other traditional applications such as voice or video transfer on the network. Radar time-series data is required to detect a meteorological signal and make estimates of the fundamental moment parameters like reflectivity, doppler velocity and spectral width. Depending on the application requirements, different transport protocols are used to transfer data over the Internet; most common of them are UDP and TCP. UDP protocol is used when application performance is constrained by timely delivery of data and application is tolerant to the loss and reordering of the data. On the other hand TCP protocol is used when applications cannot tolerate any loss and reordering of the data, but are not constrained by a strict real-time data delivery requirement.

There are different ways in which time-series data can be requested by the end applications. It is required to stream time-series radar data in real-time or in non real-time. For real-time time-series data streaming the bandwidth requirement could be orders of magnitude higher than common Internet applications like Voice/Video streaming. In case of voice transfer, bandwidth requirements can vary from 6 Kbps to 128 Kbps, and for video streaming, bandwidth requirement can vary from 50 Kbps to 6 Mbps. Bandwidth requirement of real-time time-series radar data is in the order of tens of Mbps to hundreds of Mbps.

Number of time-series samples received for a resolution volume determines the accuracy of the end moment parameters; higher the number of samples higher is the accuracy. Many end algorithms have a limit on the errors that they can tolerate in the moment parameters. Maximum acceptable error in the moment parameters determines the minimum number of time-series samples required per resolution volume. Due to real time requirements of end algorithms, it is necessary to deliver the minimum number of samples in a bounded time. This determines the minimum bandwidth required for the end algorithm. Therefore it is necessary for protocols to always transmit data at or above this minimum rate for a particular algorithm. It is possible that network becomes a bottleneck due to limited bandwidth availability. Under these scenarios, transport protocol should not transmit data at a rate that network cannot support, that would not only aggravate the network congestion but may lead to high losses for the end applications as well. Similarly depending on the resources available at the destination, an application can dictate maximum rate at which it can receive data from the radar server.

An end algorithm may dictate sample requirements depending on the acceptable error in the moment parameters. These application specific sample requirements are explained in Section V. It is also important to make sure that requirements of radar applications are met while not degrading the performance of other streams on the network. In this paper we consider these requirements as a benchmark for evaluating the performance of different transport protocols when used for radar data streaming.

We perceive that wider distribution of the time-series data requires transmission over the Internet. Thus for time-series radar streaming over the Internet it is necessary to play by the rules of the Internet. A transport protocol TRABOL (TCP friendly Rate Adaptation Based On Loss) is proposed in Bangolae (2002,2003) that can be used for the digitized radar data transfer. In the current paper we present a new mechanism for sample selection schemes as per the end application radar data quality requirements. This sample selection scheme complements the TRABOL by considering the application requirement for time-series samples at the time of transmission.

**Corresponding Author*: Prof. Anura P. Jayasumana
Dept. of Electrical and Computer Engineering
Colorado State Univ., Fort Collins, CO, 80521
Email: Anura.Jayasumana@colostate.edu

In this paper we compare the performance of UDP, TRABOL (with sample selection scheme) and TCP (without sample selection scheme) against the benchmark defined above. Section II briefly describes the TRABOL protocol. Section III explains the concept of TCP friendliness and fairness. Section IV shows different performance metrics used for the evaluation. Section V discusses different sample selection schemes. The radar network emulation test bed is described in section VI. Section VII presents the emulation results and conclusions are presented in the Section VIII.

II. TRABOL (TCP-friendly Rate Adaptation Based On Loss)

TRABOL protocol was developed to meet the requirements of applications like radar data streaming. This protocol uses UDP as an underlying transport protocol with functionality of transmission rate adaptation under network congestion Bangolae (2002, 2003) and Handley (2003). Rate adaptation is done such that it meets the minimum and maximum rate requirements of the application while remaining friendly to the cross traffic on the network.

There are three main phases in the operation of TRABOL: (i) Rate decrease phase, (ii) Rate increase phase, and (iii) Memory phase. Figure 1 and figure 2 explains TRABOL rate control mechanism in detail. Figure 1 gives the flowchart explaining different phases and figure 2 shows the real-time dynamics of transmission rate control.

Whenever packet loss for the last ray at the receiver exceeds a threshold (referred as acceptable loss in figure 1), the protocol goes into rate decrease phase; in this phase transmission rate is reduced by a decrease factor or to the minimum rate. Reason for selecting this approach is that TRABOL tries to prevent congestion in the network by acting aggressively during the rate decrease phase. Protocol remembers the last rate at which packet loss exceeded the acceptable loss. Available network bandwidth is dynamic in nature; TRABOL probes the available bandwidth conservatively by increasing the transmission rate additively by the increase factor. During this phase, known as the rate increase phase, transmission rate is increased additively till the transmission rate become equal to last congestion rate - Δ, where Δ is very small number (below last rate at which packet loss exceeded acceptable threshold). At this time, protocol transitions to the memory phase, during memory phase, rate increase is halted for a constant amount of time (referred as M epochs in figure 1) in order to avoid further congestion of the network. After the memory phase, TRABOL once again transitions to rate increase phase. It remains in the rate increase phase as long as losses suffered are below a threshold and transmission rate does not exceed the maximum rate requirement of the application.

In figure 1 and figure 2, ‘A’ indicates rate decrease phase, there is sudden fall of sending rate to the minimum rate, which is 120Mbps for this case. Rate increase phase is indicated using marker ‘B’ during which sending rate increases additively and it never exceeds maximum rate, which is 220 Mbps in this case. Memory phase of TRABOL is shown by ‘C’ during which sending rate remains constant for a short duration of time.

Note that for radar application, in all three phases of TRABOL, data is transmitted as per the sample requirements of the end algorithms. As mentioned before, an algorithm may have maximum and minimum receive rate requirement for real-time operations. Different rates are achieved by sending variable number of samples within a dwell time (time to generate all the samples of all resolution volumes).

Minimum numbers of samples are always transmitted within dwell time to meet the minimum rate requirement of the application. Increasing number of samples within dwell time increases rate. When it is not possible to send all time-series samples within dwell time due to network or end application limitations, then samples are transmitted as per the sample selection scheme described in section V. None of the existing transport protocols have the ability to meet such radar specific sample requirements. In this paper we have demonstrated the use of sample selection scheme along with TRABOL that meets radar data quality requirement.

III. TCP Friendliness and Fairness

In order to stream radar data over the Internet, it is necessary that transport protocol used not cause any degradation in the performance of the other applications sharing the bottleneck link. Majority of the Internet traffic is TCP thus impact of transport protocol could be studied using its TCP friendliness measure. The traditional TCP friendliness definition Padhye (1998), Floyd (1999, 2000) is based on long term throughput of a flow, which is related to its loss rate, round trip time, and maximum packet size according to formula Floyd (1999), Mathis (1997), and Padhye (1998):

$$R = 1.22M / (RTT\sqrt{p}) \quad (1)$$

R is the average TCP throughput, p is the TCP packet loss rate and M is the TCP maximum segment size. If non-TCP flows long term average throughput is same as that given by equation 1 then that flow is said to be TCP friendly. Multiple applications streams can share the bottleneck link at the same time. Depending on the transport protocol used, traffic streams can get different shares of the bottleneck bandwidth. It is also desired that all the streams passing through

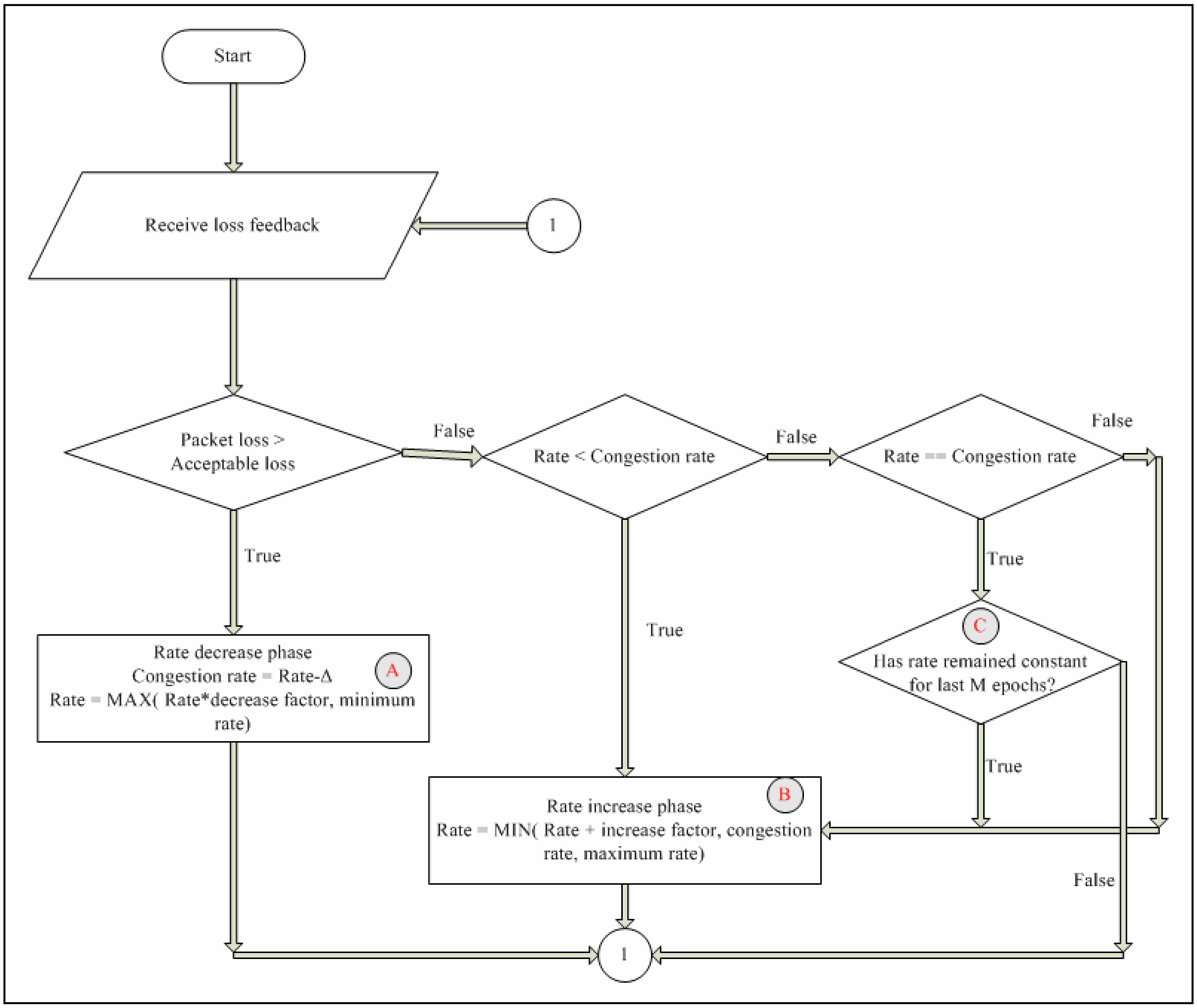


Figure 1: TRABOL flowchart for rate control mechanism

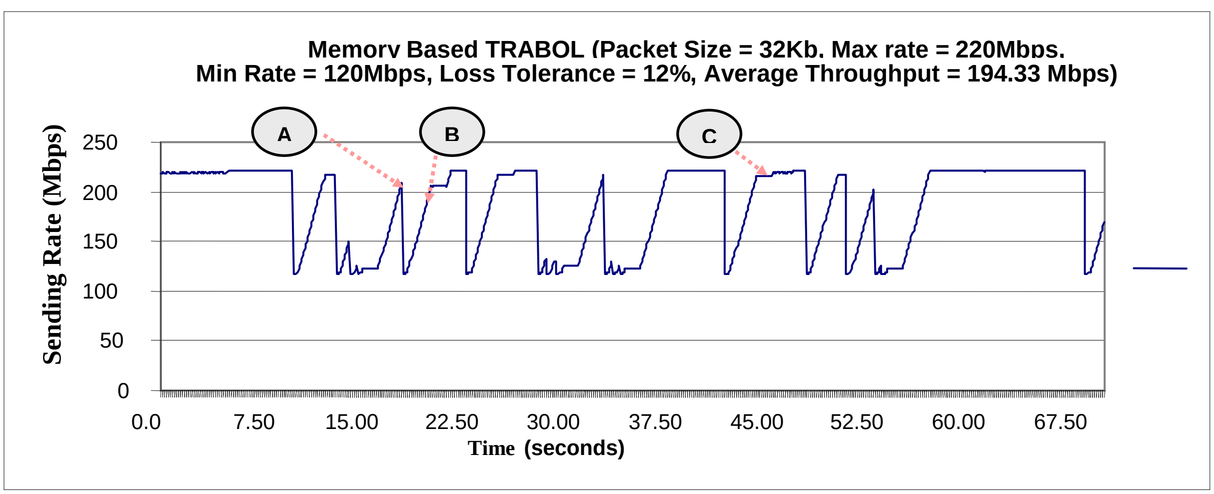


Figure 2: Snapshot of effect of TRABOL algorithm on the sending rate

the bottleneck link get their required share of the bandwidth. This behavior of the transport protocol can be evaluated using fairness measure. Section IV explains the fairness measure. Pure UDP based application is not fair, as it tend to occupy its required bandwidth at the expense of other cross-traffic streams on the network. By doing so it becomes unfriendly to the TCP cross-traffic and causes congestion in network, which in turn may turn out to be detrimental to the application itself. TRABOL on the other hand tries to be TCP friendly by operating between minimum rate and maximum rate whenever there is congestion on the network. There is a limit to which TRABOL can become TCP friendly, since it needs to ensure that sending rate doesn't fall below the minimum rate requirement of the end application. TCP shares the bottleneck bandwidth fairly with the other TCP streams on the network, and thus is friendly to other TCP streams on the network.

IV. Performance Metric

Quality of radar data received by the end application and TCP friendliness are important criteria for evaluating the performance of the transport protocols for radar applications. Following metrics are used for performance evaluation and comparison: (i) Standard deviation in moment parameters, and (ii) Fairness measure

(i) Standard deviation in moment parameters: Moment parameters have minimum standard deviation when all samples of resolution volume are used for the computation. As the number of samples decreases, it leads to an increase in the standard deviation. Standard deviation in moment parameters also depends on the factors like, whether single samples, pairs of adjacent samples, or triplets of adjacent samples are used for the computation. An other factor that may impact is how these different samples are separated in time. For e.g. for reflectivity computations it is required to use single samples within a resolution volume. Similarly for velocity computations it is desired to use pairs of adjacent samples within a resolution volume. We can thus claim that accuracy of moment parameters and quality of time-series radar data received can be determined by computing the standard deviation in the moment parameters.

Under ideal conditions, when there is no congestion on the network, then end user would receive all samples of all the resolution volumes with high probability. In reality packet loss is a common phenomenon on the Internet; under network congestion time-series radar data can suffer variable losses. Thus it is imperative to understand the impact of dynamics of the network on the quality of data in terms of standard deviation in the moment data. Transport protocols may have distinct behaviors under similar network conditions, thus it is possible to receive different samples using different protocols for the same resolution volume that can lead to different standard deviation in the moment parameters. Therefore standard deviation in the moment parameters can be used to compare performance of transport protocols.

(ii) Fairness measure: This measure captures the extent to which streams in a bottleneck link receive their required share of bandwidth. When all streams in the network dictate their rate requirements then fairness measure (F) is defined as Chiu (1989) and Jain (1984):

$$F = (\Sigma x_{i/}y_i)^2 / (n\ (\Sigma(xi/yi)^2) \qquad (2)$$

where, n = number of streams in the bottleneck
x_i = throughput of the i^{th} stream.
y_i = requirement of the i^{th} stream

When all the streams have the same requirement, then fairness measure reduces to

$$F = ((\Sigma x_i)^2 / (n\ \Sigma x_i^2) \qquad (3)$$

F is the fairness measure. F varies between (0,1], when all the streams in the bottleneck link get their required share then fairness index is 1

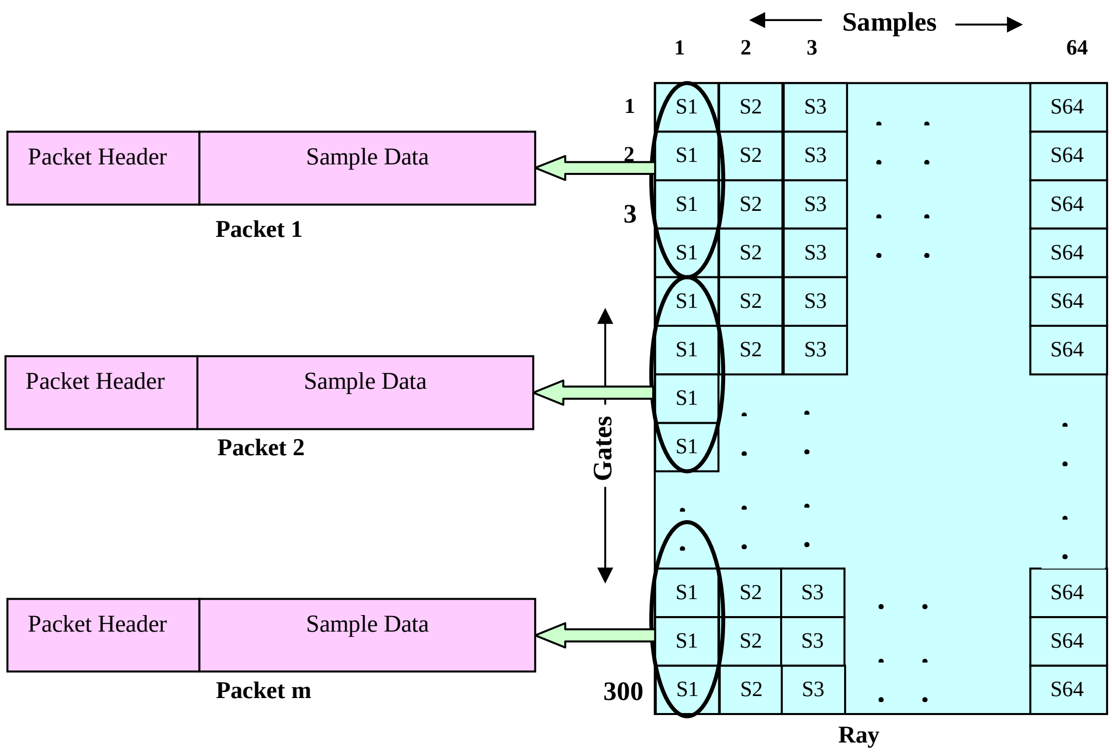
V. Sample Selection Schemes

Different radar applications may have different sample requirements e.g. velocity computation needs adjacent samples and reflectivity computation can be done using single sample at a time. We call this requirement as sample group requirements of the application. In this paper following sample groups are considered: (i) Single sample and (ii) Sample pairs

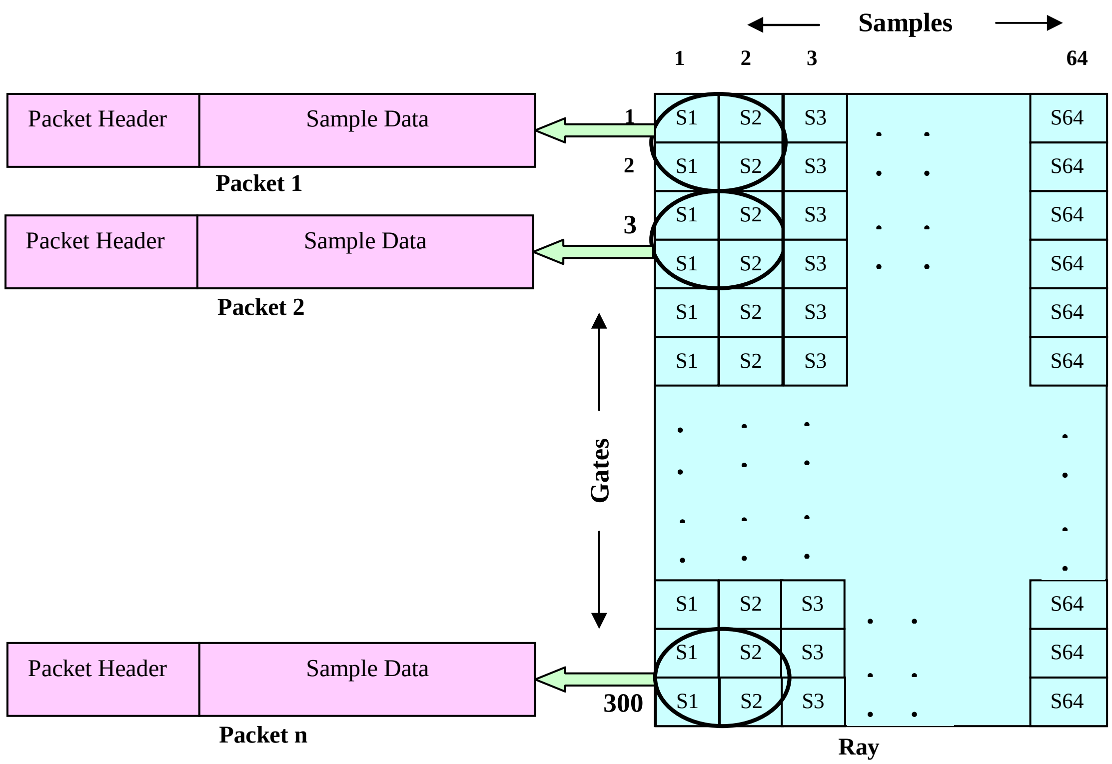
(i) Single sample group: This sample group is required for application that need single sample at a time for the computation for e.g. reflectivity computation. As shown in figure 3(a), a single sample from multiple resolution volumes is included in the same packet. Advantage of the proposed approach is that under lossy network conditions, when a packet is dropped only one sample for any resolution volume is lost thus not impacting the quality of the end result significantly.

(ii) Sample pair group: This sample group is required for applications that need two adjacent sample at a time for the computation e.g. velocity computation. As shown in figure 3(b), a pair of adjacent sample for multiple resolution volumes is included in the same packet. In case of packet loss, samples are dropped in pairs for a particular resolution volume. Whenever packets are received, they always have samples in pairs.

Besides sample groups, pattern of received samples in time also impacts the accuracy of the end



(a)



(b)

Figure 3: Sample group schemes: (a) Single sample selection (b) Pair Sample selection

results. When all the samples cannot be transmitted because of network bandwidth or client end limitation then it is required to transmit less number of samples to the destination. Thus there is need to drop some samples at the sender end, there are different ways in which samples can be dropped e.g. uniform drop, drop in contiguous group or random drops. Each of these samples drops scheme may have different impact on the accuracy of the moment parameters. So algorithms may specify different sample patterns schemes requirement that would minimize the errors in the moment parameters.

In this paper we analyze two sample pattern schemes (i) Uniform pattern, and (ii) Contiguous pattern. These sample pattern schemes are used to select samples to be dropped for each resolution volume. Figure 4 explains the sample pattern schemes, each arrow in figure 4 represents a transmitted sample, adjacent arrows (dashed or solid) represent samples that are included in the same packet and dot represents a sample that is dropped at the sender end.

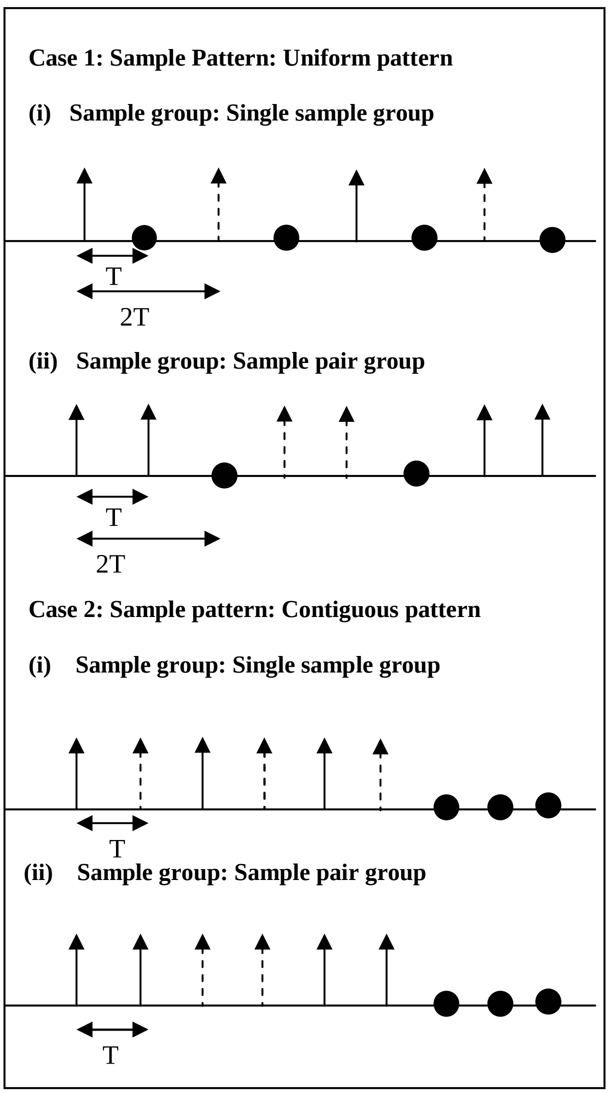


Figure 4: Sample pattern schemes

(i) Uniform pattern: In this sample pattern scheme, sample drops are uniformly distributed among the total samples of the given resolution volume. In figure 4, case 1 shows sample drops using uniform pattern scheme. All the samples that are selected for transmission using this sample pattern scheme also meet the sample group requirement of the application. This means that, while samples are dropped uniformly and at the same time they are transmitted either as single sample group or sample pair group as shown in Figure 4 under case 1.

(ii) Contiguous pattern: In this pattern scheme, a single cluster of adjacent samples is dropped. Number of samples to be dropped in a single cluster is determined by the rate at which data is to be transmitted. Remaining samples are transmitted using the user specified sample group scheme. Figure 4, case 2 shows the sample drops using contiguous group scheme while meeting the sample group requirement of the application.

In case of TCP, all samples are delivered to the destination; therefore sample selection scheme is not required for the transmission. In case of UDP, sample group requirement is considered during the transmission. UDP does not drop packets at the sender end to avoid network congestion, thus no sample pattern requirements are considered with the UDP. However, it is possible to select both sample group and sample pattern as per the end application needs with TRABOL. Since TRABOL can dynamically adapt its transmission rate, it increases its transmission rate by increasing the number of samples to be transmitted as per the sample group and sample pattern requirement. Similarly transmission rate can be reduced by sending less number of samples while considering both sample group and sample pattern requirements.

VI. Radar Network Test Bed Emulator

Experiments are conducted using a radar network emulation test-bed. There are five components of this test bed as shown in figure 5: (i) Radar emulator, (ii) Network emulator, (iii) Client application, (iv) Cross traffic generator (CTG), and (v) Cross traffic receiver (CTR).

Radar emulator, network emulator, and client machine are the Dual Xeon processor 3.06GHz server machines with 2GB RAM. Radar emulation is done using archived time-series data. On the client machine, all meteorological algorithms are executed and performance analysis is done. Network emulator NISTNet is used to emulate different network

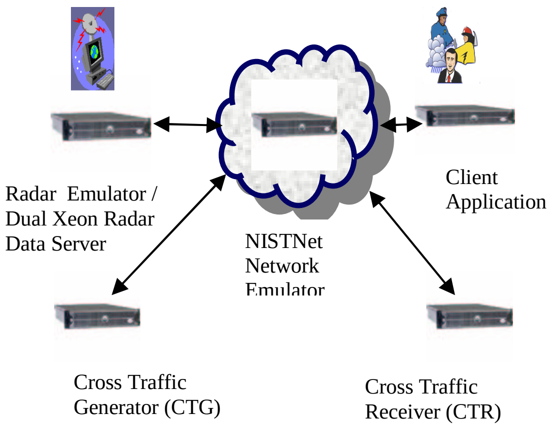


Figure 5: Emulation test-bed

bandwidth and loss scenarios. Since radar emulation is a disk intensive operation, RAID 0 functionality is used to enhance the read and write performance of the disks in server machine.

Radar emulator generates data at 90 Mbps, with 300 gates per ray and each gate has 64 samples. Different network dynamics like bottleneck bandwidth, packet losses etc. are emulated using cross traffic generator (CTG) and cross traffic receiver (CTR). Under different network conditions like variable packet losses, impact of different transport protocols and different sample selection schemes is studied on the end algorithms. Performance comparison of impact of UDP, TCP and TRABOL on TCP cross traffic is done.

VII. Results

Two sets of experiments are performed using the radar network emulation test bed; first set of experiments is done to investigate the effect of different sample selection schemes with UDP and TRABOL on the moment parameter computations. Second set of experiments compares the effect of TCP, UDP and TRABOL based radar data transmission on the TCP cross traffic on the bottleneck link while meeting the real-time requirement of the application.

It is possible that data packets get lost during the transmission on the network; different transport protocols deal with this scenario differently. In case of UDP, sender side does not take any corrective action in terms of re-transmission or reducing its sending rate to avoid further loss of information due to congestion in the network. This nature of UDP not only is unfair to the other traffic streams on the network, but can further aggravate the congestion on the network thus leading to higher probability of losing more radar data information during transmission. Since the data losses are random in nature, the end application may not receive the required samples. This can lead to unacceptable standard deviations in many computed moment parameters. Thus UDP may not meet the application requirement of minimum acceptable quality of data under varying network conditions.

TCP can detect and recover from packet losses by re-transmitting the lost packets. It is possible that re-transmitted data may arrive after the deadline for the data has passed at the receiver end, making it useless. Thus TCP may not be a suitable choice for real-time application because it does not guarantee to meet minimum rate requirement when network is under congestion. One of the main advantages of TCP, however, is that for every detected loss it takes preventive action by reducing the transmission window size to 1 and then increases sending rate using AIMD scheme; This approach helps in preventing further congestion on the network. By doing so TCP also becomes fair to the cross traffic on the network but at the expense of reducing the throughput of radar stream drastically.

(i) Radar data quality:

Radar data is transmitted using different sample selection schemes as explained in section V. Radar data quality is determined by estimating standard deviation in the moment parameters, computed using the received samples for a particular resolution volume. In case of TCP, since all the samples are received, the standard deviation in the moment data is minimal. For a fair comparison of TRABOL and UDP, it is desired that radar data quality is compared under same network loss conditions. Same sample groups are sent using both protocols, i.e., either single sample or sample pairs are sent using UDP and TRABOL. For the experimental results, available bottleneck bandwidth is set to 45, 60, 70 and 80Mbps; that correspond to 50%, 35%, 25% and 10% packet loss respectively. When UDP is used for radar data streaming, there is no mechanism for rate control, thus data is always transmitted at the rate at which data is generated by the radar emulator, i.e., 90Mbps. Since available bandwidth is less than the 90Mbps, the packets losses introduced are random in nature for the radar data. Experiments are performed for a case when end application has target rate requirement of 90Mbps and minimum rate requirement of 50Mbps. TRABOL adapts transmission rate dynamically between maximum rate 90Mbps and minimum rate 50Mbps as per the available bandwidth. In case of TRABOL overall same amount of information is lost as in the case of UDP but TRABOL drops most of the packets at the sender end deterministically as per the end applications sample group and sample pattern requirements. Case when bottleneck link is 45Mbps, TRABOL receiver throughput is below its minimum rate.

Comparison results of radar data quality under different data loss conditions using different sample

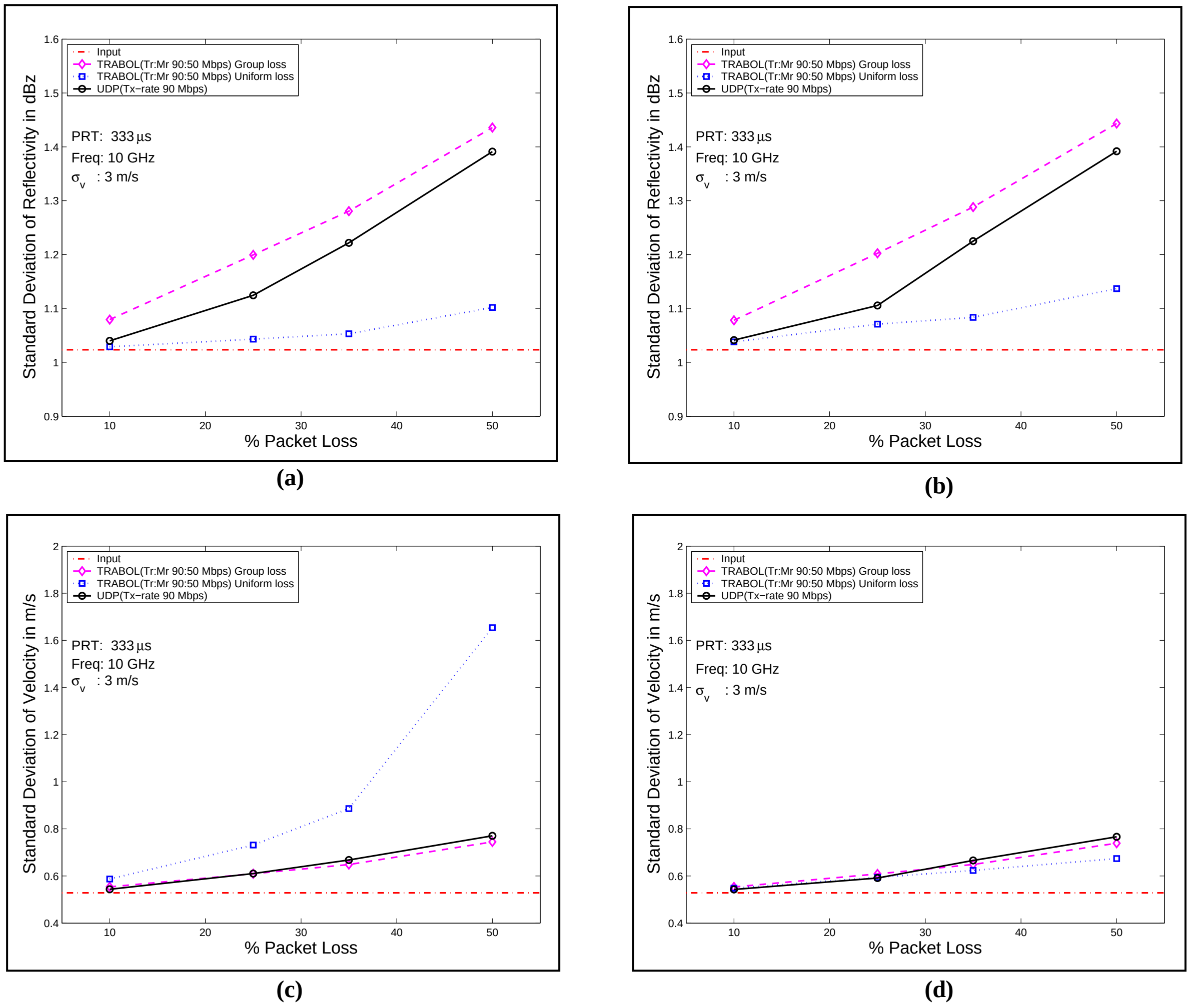
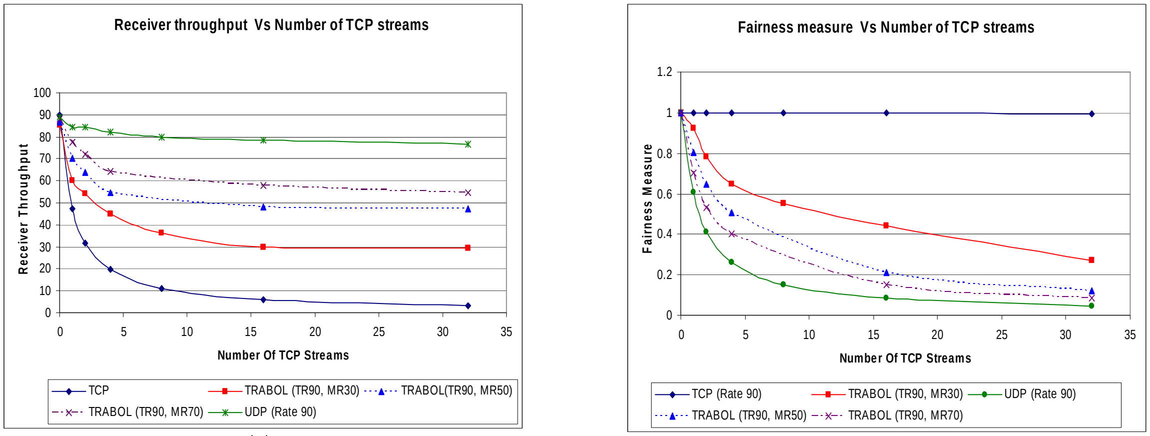


Figure 6: Impact of sample selection schemes using UDP and TRABOL on the radar data quality (a) SD in reflectivity with single sample group under uniform and contiguous pattern packet loss (b) SD in reflectivity with sample pair group under uniform and contiguous pattern packet losses (c) SD in velocity with single sample group under uniform and contiguous pattern losses (d) SD in velocity with sample pair group under uniform and contiguous losses

selection schemes are shown in figure 6. Experimental results in figure 6 show those deterministic losses introduced by TRABOL along with sample selection scheme mentioned in section V help in improving the accuracy of the moment parameters under most conditions. UDP packet losses are random in nature due to network dynamics, thus standard deviation is high in most cases when compared under same receiver throughput conditions for both UDP and TRABOL. In figure 6(a), single sample group is used to transmit data using both UDP and TRABOL. It can be seen that TRABOL with deterministic uniform losses has minimum standard deviation in reflectivity when compared to UDP with random losses and TRABOL with contiguous losses. Figure 6(b) shows a case when sample pair type is used by both UDP and TRABOL. Same behavior is observed as in figure 6(a), i.e., TRABOL with uniform loss of sample pairs has minimum standard deviation for reflectivity compared to UDP with random loss and TRABOL with contiguous loss.

In figure 6(c), single sample group is used by both UDP and TRABOL. TRABOL with contiguous loss and UDP with random loss performances are quite similar. TRABOL with uniform loss with single sample group shows worst performance because uniform pattern of loss with single sample group leads to minimum number of sample pair delivery, increasing standard deviation in end parameters. Figure 6(d) shows a case when sample pair group is used; in this case once again TRABOL with uniform loss of sample pairs performs better than the rest. It is thus evident that, TRABOL along with deterministic sample selection scheme can out perform UDP in terms of better quality moment data under same network conditions. Note that in case of TCP, since all samples are received so quality of data is superior to



(a) **(b)**

Figure 7: Comparison of TCP friendliness and throughput of UDP,TCP and TRABOL

either UDP or TRABOL under lossy network conditions. In figure 6, dotted line corresponding to input can be taken as a measure of standard deviation in moment parameters due to TCP protocol.

(ii) TCP friendliness and throughput:

An other benchmark for comparing transport protocol performance is the TCP friendliness measure of the protocols. In the previous section, we saw that TCP outperforms both TRABOL and UDP in terms of radar data quality under similar network conditions. In this section, we compare the TCP friendliness of three protocols and their performance in meeting the real-time requirement of the end application. TCP friendliness is captured using the fairness measure as explained in section IV. Real-time performance of the protocol is determined using long-term and short-term throughput at the receiver end under different network cross traffic conditions. Radar stream and cross traffic share a bottleneck bandwidth emulated using NISTNet network emulator. Performance results are taken by varying the number of TCP cross traffic streams and observing the receive throughput of the radar stream. Fairness measure is computed for different cross traffic conditions. Radar data is streamed using TCP, UDP and TRABOL from radar data server to client application as shown in figure 5.

Figure 7 shows experimental results for the receiver throughput and fairness properties of TCP, UDP and TRABOL. Performance results are shown for TRABOL with target rate (TR) 90Mbps and minimum rates (MR) 30Mbps, 50Mbps and 70Mbps. UDP streams are always transmitted at 90Mbps, i.e., rate at which, the radar emulator generates data. Note that loss tolerance threshold is 0% in case of TRABOL, which means that TRABOL immediately goes into rate decrease phase on detection of a single packet loss. Also all the throughput measurements are taken at the client end.

Figure 7(a) shows the receiver throughput of different transport protocols with cross traffic variation. When radar data is transmitted using UDP on a network with TCP cross traffic then UDP protocol captures all the bandwidth at the expense of throughput of TCP traffic. On the other hand, when radar data is transmitted using TCP, receiver throughput decreases as the number of cross-traffic TCP streams increases. It is possible that TCP throughput falls below the minimum rate requirement of the application. In case of TRABOL, receiver throughput keeps on falling as the number of cross-traffic streams increases but it does not decrease beyond the minimum rate requirement of the end application. In figure 7(b) it is seen that UDP is least fair to the other cross traffic streams, and TCP has maximum fairness. Fairness of TRABOL lies in between there of UDP and TCP. Fairness of TRABOL decreases with the increase in number of cross traffic streams or increase in the minimum rate requirement. If receiver receives radar data below the minimum rate then it can be inferred that transport protocol, network conditions or both are not suitable for meeting the rate requirement of the radar application.

VIII. Conclusions

This paper compares the performance of TCP, UDP and TRABOL for sending digitized radar data. Radar data quality results show that for same sample selection scheme and under same loss condition, quality of data received using TRABOL is better than the UDP case. At the same time, TRABOL is friendlier to other cross traffic streams on the network when compared to the UDP protocol. TCP on the other hand delivers all the radar data to the destination, but its throughput falls significantly under network congestion thus it does not meet the real-time streaming requirements of the radar data. After evaluating performance results, we can conclude that TRABOL, along with the different sample selection schemes, is able to meet the radar quality requirement without overly degrading the performance of other applications on the network.

Acknowledgements

This work was supported in part by the Engineering Research Center program of the National Science Foundation under NSF award number 0313747 and NSF Information Technology Research (ITR) grant number 0121546.

References

Bangolae S., Jayasumana A.P., and Chandrasekar V., Nov. 2002:
“TCP-friendly Congestion Control Mechanism for a UDP-based High-Speed Radar Application and Characterization of its Fairness,” Proc. 8th IEEE International Conference on Communication Systems (ICCS), Singapore, pp.164-168.

Bangolae, S., A. P. Jayasumana and V. Chandrasekar: May 2003:
“Gigabit Networking: Digitized Radar Data Transfer and Beyond,” Proc. IEEE International Conference on Communications (ICC'03), Vol. 1, pp. 684-688, Anchorage

Bangolae, S., Fall 2003:
“A TCP-friendly Congestion Control Mechanism for High Bandwidth Radar Application”, MS Thesis, Colorado State University

Carson M. and Santay D., 2003:
NIST Net: A Linux-based Network Emulation Tool. *ACM SIGCOMM Computer Communications Review*, 33(3): 111–126

Chiu D. and Jain R., June 1989:
"Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks". *Journal of Computer Networks*

Floyd S., Handley M., and Padhye, J., 2000:
“Equation-Based Congestion Control for Unicast Applications". SIGCOMM

Floyd S., and Fall K., August 1999:
"Promoting the use of End-to-End Congestion Control in the Internet" IEEE/ACM Transactions on Networking, August 1999.

Floyd S., Handley M. and Padhye J., May 2000:
“A comparison of Equation-based and AIMD Congestion Control” *ACIRI Technical Report http://www.aciri.org/tfrc/aimd.pdf*

Handley M., Floyd S., Padhye J., Widmer J., January 2003:
“TCP Friendly Rate Control (TFRC): Protocol Specification”, RFC 3448, *http://www.ietf.org*

Jain R., Chiu D., and Hawe W., Sept 1984:
“A Quantitative Measure of Fairness and Discrimination for Resource Allocation In Shared Computer Systems”, *DEC Research report TR-301*

Mathis M., Semke J., and Mahdavi J., July 1997:
"The Macrosopic Behavior of the TCP Congestion Avoidance Algorithm". *ACM Computer Communication Review, vol 27 num 3*.

Padhye J., Kurose J., Towsley D., and Koodli R. Oct. 1998:
"A Model Based TCP-Friendly Rate Control Protocol". UMass-CMPSCI Technical Report TR 98-04.

Padhye J., Firoiu V., Towsley D. and Kurose J., 1998:
"Modeling TCP Throughput: A Simple Model and its Empirical Validation". *ACM SICOMM'98*