M.I.T. Laboratory for Computer Science

Request for Comments No. 257
June 12, 1984

Packet Trains for Network Workload Characterization

by Rajendra K. Jain

1. Overview

For network modeling and simulation, it is essential to know the right model for packet arrivals. The efficiency of protocols generally depends heavily on the arrival pattern. For example, windowing helps if traffic is bursty but it may not be useful if traffic is sparse.

This RFC proposes a possible model for network workload. Starting with Poisson and bursty Poisson packet arrivals which have been traditionally used for network modeling and analyses, we discuss their pitfalls and propose a more realistic model based on the concept of *packet trains.* The next two sections discuss the Poisson and bursty Poisson arrival respectively. The concept of packet trains is then introduced and described in detail.

2. Poisson Arrivals:

This is the most commonly used model in simulation and modeling studies. As shown in figure 1, in this model the inter-arrival time of packets is assumed to be memoryless (exponential). That is, knowing arrival time of a packet gives no clue about future arrivals.

While this may be an appropriate model for remote virtual terminal traffic, it is certainly not appropriate for file transfer traffic on a network. Successive packets on a network do not arrive independently but are mostly related. In fact, those sources which just sent a packet have much

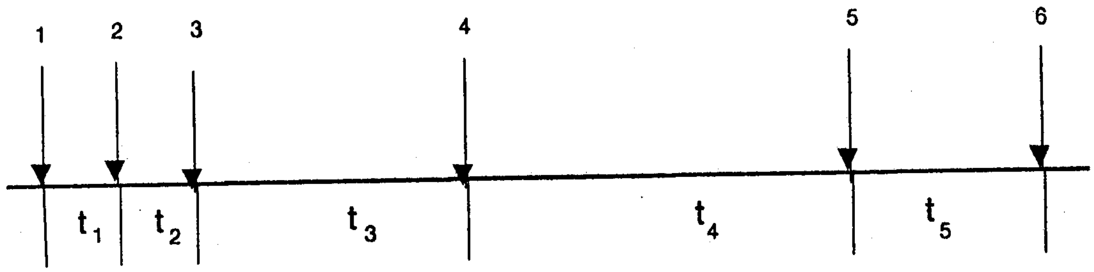

Figure 1: Poisson arrival pattern assumes single packet arrival. The interarrival time between packets is exponentially distributed.

higher probability of sending another packet than those that have been inactive for some time. Similarly, once we see a packet on the network, probability of seeing another packet very soon becomes high. In other words, packets arrive in bursts. The Poisson arrival model of memoryless independent arrivals, therefore, does not apply.

The measurements done at M.I.T. Lab for Computer Science on Version 2 Ring [Feldmeier 84] clearly show that the packet arrivals are not Poisson. A plot of ln(p(t)) as a function of t, taken from Feldmeier's thesis, is shown in Figure 2. Here, t is the inter-arrival time and p(t) is the probability density of t. If packet arrivals were Poisson, p(t) would be exponential and ln(p(t)) would be a linear function of t as shown below:

$$p(t) = \lambda e^{-\lambda t}$$
$$\ln(p(t)) = \ln(\lambda) - \lambda t$$

Figure 2 shows that the probability of short inter-arrival times is much higher that what would be in a Poisson arrival.

3. Bursty Poisson:

To model bursty arrival in other (non-networking) applications, such as customer arrivals at a bank, another model, called bursty Poisson is used. Under this model, a group (burst) of customers arrive at a service facility with exponentially distributed inter-arrival times. The burst size is random (see Figure 3).

If the burst size is geometrically distributed, then closed form expressions for the waiting times

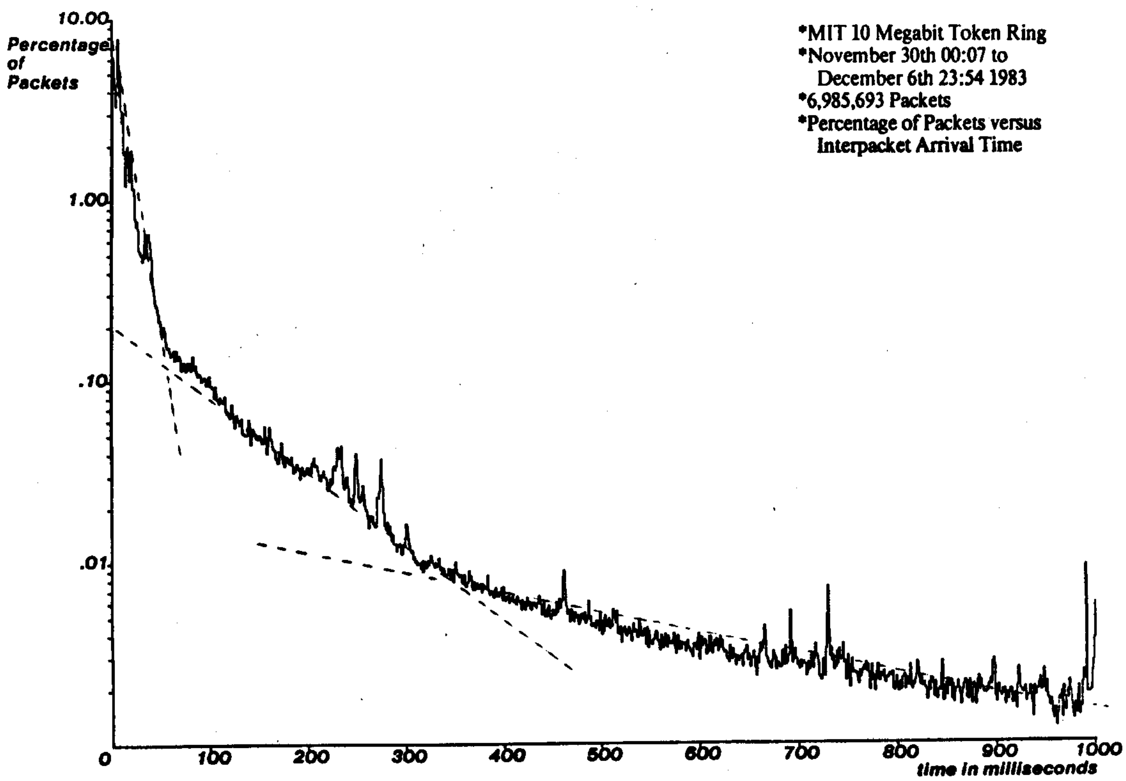

Figure 2: Histogram of packet inter-arrival times on a log-linear paper. Poisson Arrivals would result in a straight line on this paper.

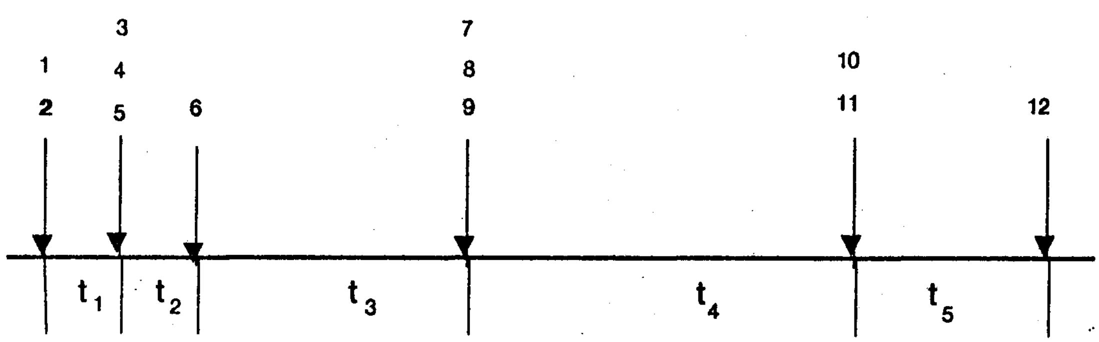

Figure 3: Bursty Poisson arrivals allow many packets to arrive simultaneously. The inter-burst time is exponentially distributed and burst size is random.

and queue sizes can be derived [Spirn 82]. However, even this model is not realistic for network traffic because the network interfaces generally have very few (usually 2) transmit buffers and therefore can never generate bursts of size more than two. In fact, after the first burst of size two, all further arrivals will be of size 1, because the higher level protocols take longer to supply the next packet which has to be read from a disk and formatted. Even if the interface has more transmit buffers, the same argument holds since file sizes are generally larger than the total buffer space in the interface.

4. Packet Trains:

The main drawback of the bursty Poisson model is the *simultaneous* arrival of all members of a burst. If we allow the members to arrive in a succession with a small inter-arrival interval we get what we call *packet trains*. Thus, as shown in figure 4, an omniscient observer looking at the network traffic would see a succession of trains of packets with relatively long idle periods between the trains.

Figure 4: Packet trains consist of a succession of packets with relatively small inter-packet gaps.

The packet train model, as proposed so far, has two inter-arrival parameters: Inter-train arrival interval between successive trains, and Inter-packet arrival interval between successive packets of a train. The trains themselves may arrive randomly, possibly in a Poisson pattern. However, after passing of a locomotive (the first packet of the train) the arrival distribution changes to one of shorter inter-arrival times between packets. The two parameters represent two different phenomena. Inter-packet interval depends upon the speed of the higher level network protocols as well as that of the secondary storage devices. The inter-train interval, on the other hand, depends upon the frequency of file transfers. The inter-packet interval has a small variance, whereas the inter-train interval is probably memoryless.

In order to detect trains in a real network traffic, we need to set an upper limit, say τ on inter-packet gap. Thus, a train is said to have ended as soon as no further packets from the same source are seen within τ seconds. We call this interval Maximum Allowed Inter-packet Gap (MAIG). Thus, with a MAIG of 10 ms, all packets from a single source within 10 ms of each other will be part of the same train. Obviously, larger the MAIG longer the trains. What is an appropriate value for MAIG? Well, it depends upon the turn around time of the higher level protocols. For example, if the disk on a node takes 40 ms to get a block off the disk, all packets belonging to a single file transfer would form a train obtained with a MAIG of 40 ms.

5. Bi-directional Trains:

The trains discussed so far are what we would now call *unidirectional trains* i.e., all packets are coming from the same source and going to the same destination. We have not made use of the fact that many protocols work in a request-response, or packet-acknowledgement, or character-echo manner. Given that we have just seen a packet going from A to B, there is a very high probability that we would next see a *reaction packet* (a response, an acknowledgement, or an echo character) going from B to A. From now on, we will use the term *reaction* to refer to packets flowing in the reverse direction.

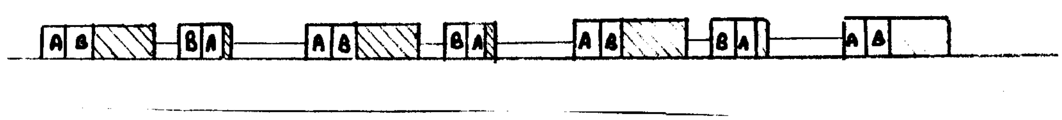

Figure 5: Bidirectional trains consist of packets going from a source A to destination B as well as those coming back from B to A.

Bidirectional trains consist of all packets flowing between two nodes (see Figure 5). The name *bidirectional* is a slight misnomer in the sense that in many local area networks, e.g., rings, packets going from A to B as well as those from B to A all travel in the same direction.

Bidirectional trains have three inter-arrival parameters: Inter-train arrival interval, Inter-packet arrival interval, and a reaction interval. The first two are defined in the same way as for unidirectional trains. The last one is the interval between a packet and its reaction. The *reaction* is generated by the node rather quickly without going to disk devices. The reaction interval is, therefore, the smallest of the three parameters.

Reaction Probability is the probability of a packet being followed by a reaction. The traffic measurements on M.I.T. LCS ring show that the number of packets received by a node is generally equal to the number transmitted. This seems to support our view that most protocols work in a packet-reaction (request-response) manner and that the reaction probability is close to one. However, a detailed analysis is required to verify this conjecture.

The three linear sections of the histogram of packet inter-arrival times (shown via broken lines in figure 2) seem to come from the three inter-arrival intervals of the bidirectional trains. The first line with small inter-arrival interval seems to be due to packet-reaction phenomenon. The second line with medium inter-arrival times may be due to inter-packet gaps in a single train. The last line with large inter-arrival times may represent inter-train intervals. Again, further analysis is required to verify this.

6. Hierarchy of Trains:

It is possible for a train to consist of several trains. For example, some operating systems may prefetch several blocks from a disk and send these in a quick succession with a small inter-packet gap. At the end of the batch, the gap may be a little longer. Each batch itself may look like a train. Sometimes this effect may also be due to the networking protocols, e.g., windowing may make a single file transfer look like several trains. Gateways may also create or destroy the train effects.

7. Train Parameters:

We need to analyze a real network traffic to produce a number of statistics related to trains. Some of these statistics are listed below. In each case, since distributions are expected to be skewed with long tails, it would be useful to know 95-percentile, and median along with mean.

1. **Inter-train Interval**
2. **Inter-packet interval**
3. **Reaction Interval**
4. **Reaction Probability**
5. **Train length** = # of packets in a train
6. **Train weight** = # of bytes in a train
7. **Train Duration:** Total time between the first bit of a train and the last bit of a train.
8. **Regeneration Cycle:** A train and the idle period following it define a regeneration cycle. One reasonable assumption is to assume that each regeneration cycle is independent of

others. That is, length of one train has no relation to length of the other trains. Similarly, length of one idle period is independent of other idle periods.

9. **Burstiness Index:** In order to compare traffic on two networks and to be able to say which one is more bursty, we need a measure for burstiness. Some function of train lengths would provide a measure of burstiness of the traffic. Finding the exact function requires more analytical and measurement work. Also, since larger MAIG will give longer trains, it may be necessary to include MAIG in the definition of the burstiness index so that the measured value of the index for a particular traffic does not depend upon the MAIG value used. This requires more study of the actual train length vs MAIG data.

8. Poisson and Bursty Poisson as Special cases of Trains:

It is obvious that unidirectional trains are special cases of bidirectional trains with zero reaction probability. What is more interesting is the observation that both Poisson and bursty Poisson are also special cases of packet trains. If all trains were one packet long and arrived with exponential inter-arrival time distribution, the traffic would be Poisson. Thus, Poisson is a special case of trains with a MAIG of zero and train length fixed at one. Allowing longer trains while keeping MAIG at zero produces bursty Poisson.

Many network design decisions are affected by the arrival patterns. For example, if it takes 40 ms to consume a packet (receive and write it out on a disk), only one buffer is required to be able to receive a train with a constant inter-packet interval of 40ms and still be able to guarantee no packet loss at 100% throughput. A poisson arrival pattern with a mean inter-packet arrival of 40 ms would imply 100% utilization but infinite average delay. The number of buffers required to guarantee no loss is also infinity. This is an extreme example but it helps illustrate the point. The trains make the next packet arrival (after the locomotive) highly predictable. The reduced uncertainty results in better performance with less resources.

Also a rate based flow control makes more sense for trains than for Poisson arrivals. For example, a receiver with a consumption rate of 40 ms per packet could ask the sender to space-out its packets by 40+ ms (a rate slightly less than 1 packet per 40 ms) whereas with a Poisson model, 100% throughput is simply not achievable.

Reservation based resource sharing, e.g., 1 slot every 40 ms on a satellite channel, is another example of network design which makes more sense for trains than for random Poisson arrivals.

9. Packet Processions:

Packet processions are yet another type of packet sequences that are of interest to network designers. These consist of a sequence of packets coming almost *back to back*, i.e., the gap between successive packets is too small to insert another packet (see Figure 6).

Figure 6: Packet processions consist of packets coming very close to each other.

For example, on a token ring, a sequence of packets with inter-packet gap less than the round trip delay around the ring would form a packet procession. Similarly, on an Ethernet[1] packets within a slot time of each other (some complete and some partial due to collision) would form a packet procession. The packets of a packet procession would generally belong to different source destination pairs.

Packet processions are important because presence of too many of these impacts network access time. The distribution of length of these processions also gives distribution of the network access time. To appreciate their importance, imagine waiting at a road intersection when a procession of cars is going by the crossing road. It is more useful to know the length and frequency of these processions than the fact that the utilization, say over a day, is only 0.05%.

10. Further Work:

The concept of trains is still a hypothesis. It looks like a reasonable model of network traffic. However, before we venture too much in to it, we need to verify that the trains do in fact exist on real networks. This requires analysis of packets flowing on a real network. Fortunately, for us the Feldmeier's ring monitor gives all the information that we need for such an analysis. The monitor collects first 16 bytes of each packet (which contain source and destination addresses and protocol type), time-stamps it, and sends it in batches to an analysis machine. Currently, the analysis machine throws away the data and stores only a summary. We need to write programs to analyze the data and study it to see train effects.

[1] Ethernet is a trademark of XEROX corporation.

Another important area to work on is analytical or simulation modeling of trains. We know that several Poisson streams when mixed together result in a stream which is still Poisson. What happens when packets of several trains are intermixed? Does a mixture of several bidirectional trains result in a tri-linear pattern for inter-arrival interval as shown in figure 2? Answers to these and other similar questions will help in modeling of networks with packet train traffic.

11. Summary:

Traditionally used models of packet arrivals, namely Poisson and bursty Poisson do not represent file traffic on networks. A better model is to represent arrivals in terms of packet trains, where all packets of a file transfer form a train. The inter-packet interval between successive packets of a train has distribution very different from inter-train interval. The former is a characteristic of the higher level protocols and system configurations while the later depends solely on user behavior.

In this paper, we described different parameters of trains that should be measured on a real network. Also, we showed that Poisson and bursty Poisson are special cases of the trains.

12. Acknowledgements:

I would like to thank Prof. Jerome Saltzer who carefully read an earlier version of this RFC and proposed the concept of bidirectional trains. Dr. David Clark provided useful comments and has promised to arrange help for further analysis. Dave Feldmeier and Lixia Zhang also reviewed the earlier draft and helped improve it.

References

[Feldmeier 84]
Feldmeier, David C.
Empirical Analysis of a Token RIng Network.
Massachusetts Institute of Technology technical memo MIT/LCS/TM-254, January, 1984.

[Spirn 82]
Spirn, Jeffrey.
Network Modeling with bursty traffic and finite buffer space.
Proc. ACM Comput.Network Perform. Symp., April, 1982.