Maximizing End-To-End QoS Guarantees under Multipath Transmission for Video over Ad-Hoc Networks

EE359 Project Report under the guidance of Professor Andrea Goldsmith
Fall 2003

December 9, 2003

Abstract

In this project we do a theoretical analysis for transmission of video over an adhoc network using multiple paths. The aim of the analysis is to maximize Quality of Service (QoS) which in the present context is defined to be the percentage (%) of video packets that meet their hard [1] delay deadline where delay arises for two reasons 1) queueing delay at the (buffers of) various nodes (before the packet gets a chance to be transmitted) that make up a path and 2) due to retransmissions of the packet if it is not correctly received. The analysis is fairly general in the sense that it holds for any arrival process and any service distribution (statistical and not deterministic) as long as it is the same for all nodes in the network, specifically random variables characterizing the arrival and service time distribution for all the nodes are independent and identically distributed (i.i.d.). The final analytical resutls have been worked out for the case when the arrival follows a Poisson distribution and the service times are exponentially distributed (the most common case). However the analysis is restricted in the sense that the results hold under specific network conditions - rates transmitted are close the maximum rates that the link can support which is true for busy networks and the number of hops in each path are many (to be qualified later) which is true for an adhoc networks of a decent size.

The optimal (in the sense that it maximizes the QoS defined above) strategy for distribution of the rates over the multiple hops is first derived for the case when the network has a single active source destination (S-D) pair. The result of the single active S-D pair are then used to develop a methodology for a distributed algorithm when multiple S-D pairs are active in the adhoc network. The choice of the initial set of flows greatly reduces convergence time (or rather assures convergence which is not always true). All this was done for queueing delay alone. Then we studied the effect of retransmissions arising from the packets being randomly corrupted.

Finally we show the performance of the distributed algorithm developed through simulations.

1 Introduction

Supporting multimedia (high rate data) with stringent delay constraints over a hostile wireless channel in a multiple access scenario is a technical challenge [2].

[1] In video transmission there is also a notion of soft deadlines, which states that a packet arriving after its deadline also has a utility value [1]. However in the present context we consider only hard deadlines i.e. a packet which arrives after its deadline has no use.

The same problem manifests itself on a larger scale in an adhoc wireless network where in a multiple access scene there is the additional headache where not only the channel is an enemy (voluntarily) but even the other nodes become enemical (involuntarily) when they are trying to push their own data simultaneously as there is no central authority to coordinate their transmission and the packets may either collide in air or the link capacities are greatly reduced to residual capacityowing to a MAC protocol in operation.

The perfect (theoretically optimal) solution to transmission in a general wireless adhoc network situation is a open research problem conveying that the problem is a tough problem. Nevertheless, recently the research community has done a lot in this field. A more detailed description of what has been done in this field is given in the next section. This project is also an effort in this direction and basically tries to come up with a theoretically optimal solution (in closed form) for a 'small' bit - a special case of the 'bigger' problem. It considers a case when there are many nodes in the network (to give multiple paths) and takes advantage of this large number of nodes to provide statistical QoS guaratees for multimedia transmission. The optimality is obtained by parititioning the total rate on these paths to maximize QoS assuming delay is due to queuing alone (perfect wireless channel and no multiple access interference which, if not there, might have led to retransmissions and so an increased delay). The 'optimal' strategy for this case is then used as a tool to develop 'good' strategies for the imperfect channel and multiple access cases. In the next two paragraphs we justify and motivate the work done in this project.

The ad-hoc network setting to derive the optimal optimal analytical solution is justified in the sense that for adhoc networks where there are many nodes and small capacity links, the strongest reason for packets not meeting their deadlines is the queuing delay in the tranmitter buffers which is only enhanced in a multiple access scene where this small capacity is further reduced to residual capacity[2]. Other important reason is random packet errors in a wireless scenario which necessitate retransmissions. This problem is argued to be a special case of the first problem and so the same solution applies.

Since error free transmission rate for a single link is really low compared to the large rates that the node has to support to push in real time video, nodes take advantage of the presence of multiple similar nodes in the network and do a multipath-multihop transmission of the data. Multipath-multihop transmission is not only a better solution ([3], [4]) but sometimes it turns out to be the only solution possible. For instance in applications like sensor networks where each node has limited power and so cannot send directly to a destination node which may be far off. Even if the node could send at high power, it prefers to send at low power to a close-by node (relying on the fact that the packet will eventually reach its destination) to reduce interference to other ongoing transmissions. Now one may argue that in effect the total power that is spent to make the packet reach its destination (sum of the power of all the nodes that forward the packet) may be higher than the power that the source node might need to send the packet directly to destination, so where is the saving of power[3] ?!!. Well the answer to this is that multihop transmission reduces interference, but even if it may not, think what would the savings of power be of use if the source node (of a sensor network) which might in the future sense some important phenomemon (and so be the most critical node) finishes its limited power and the intermediary nodes save their power forever as the application itself is over !!!

With this motivation, we did the analysis for multipath-multihop transmission of video

[2]There is generally some MAC protocol operating the network which is used by all nodes to allow error free transmission for other nodes and avoid collisions. Due to this MAC protocol nodes cannot transmit for all times and queueing delays increase or the capacity is reduced to residual capacity.

[3]We came across this interesting contradiction in power saving during our work. But due to lack of time we have not analyzed this which seems to be an interesting research problem. However, we do give some intuition regarding this in the line that follows.

over an adhoc network.

2 Related Work

Most of the work in literature related to providing delay guarantees to video traffic is for networks where there is set of given path for a node to transmit the data and the data rates are sufficiently high to be sent on the available paths and simultaneously meet their deadlines. So there is traffic shaping [5], smoothening of the traffic at the ingress [6] with traffic modelled as fluid [7] and node buffers as leaky buckets [8]. Essentially its optimal transmission of video over a *limited* number of available paths. The results given in these work are generally applicable to both wireless networks and (traditional) wired networks as the analysis just considers the delay in the queues.

Given this it may appear that the work done in this project is hypothetical in the sense that we consider that we have *many* paths to be chosen from (which might seem a luxury) and the aim of the analysis is to find a set of path that maximize the QoS criterion defined. The answer is a *yes* for traditional networks; however, in adhoc networks this situation is as realistic as adhoc networks themselves. The results derived in the referenced given above do not generally apply to adhoc networks where due to limited bandwidth and potential node mobility, traditional networking principles may not hold. This is especially true for routing. In this regard, authors in [9] have developed a strategy for congestion-optimized multipath routing of video., in which the total rate is divided among a set of (*many*) available paths with the aim to minimize total network congestion.

Routing algorithms generally share the goal of minimizing the number of hops between source and destination [10]. In [9], the authors contend that this may not be the best strategy and develop a strategy to minimize the total network congestion. In this project we start with the aim of maximizing QoS (% of packets meeting their deadline which would intuitively increase if network congestion is reduced, same aim as in [9]) and end up showing that for queues with Poisson arrivals and exponential servce time distribution the optimal strategy is to put more rate on paths with lesser number of hops and in this way select a set of paths that have minimum number of hops. We, thus, show that the two aims in [10] and [9] are essentially identical.

For the case of rate-partitioning in a distributed algorithm, several distributed routing protocols exist and have been tailored to wireless adhoc networks [11], [12]. In this project, we develop a simple distributed algorithm based on our intuition derived from the analytical results obtained for the single S-D case.

The rest of the project is organized as follows. In Section 3, we outline the basic assuptions used in the project. Section 4 provides the derivation of the result for optimal flow partitioning in the S-D case, its extension to the distributed algorithm with multiple active S-D pairs and the changes made to it when there are retransmissions due to random packet loss/corruption. In Section 5, we show the performance of the distributed algorithm developed through simulations. Concluding remarks are given in Section 6.

3 Assumptions

We make the following assumptions in our analysis:

1. Nodes have infinitely large buffers.

2. Network is a Jackson Network [13].

3. Network is operating in steady state so Burke's theorem holds [13].

4. Each packet has one source and one destination (unicasting).

5. Source nodes know about the number of available paths and the number of hops on each path and link capacities of the link. This is required to obtain a closed form solution for the flow partitioning in the single S-D pair case. In the distributed algorithm, we will relax all these assumptions on hte knowledge of the source node.

6. Packet sequencing at the source is FIFO.

7. Neglect all the resequencing delays at the receiver.

8. Multiple paths are assumed to have no common links.

9. Nodes do no adapt power while transmitting on different paths (equal power on all paths). Other than these standard assumptions, we make two more assumptions for which the analytical result in the single S-D case holds:

10. Network is a busy network.

11. The number of hops on each path is *many* (say greater than 3).

4 Derivation of Optimal Flow Partitioning

In this section we first derive the expression for optimal flow for single active S-D pair. Using ideas from this derivation, we will find a simple way for implementing the distributed algorithm. The cases considered are shown in Fig. 1

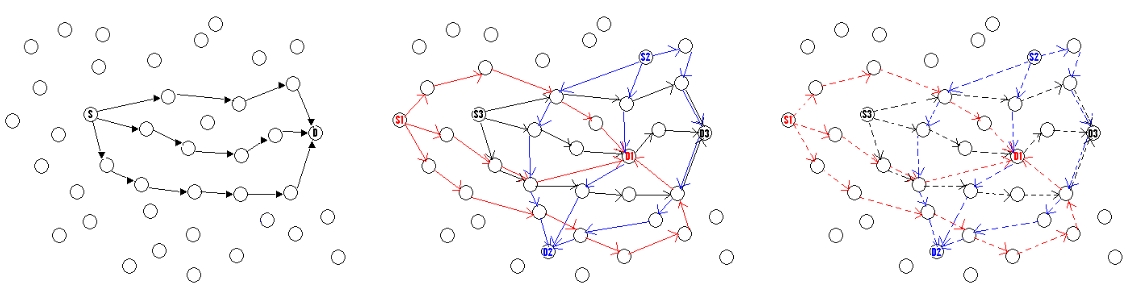

Figure 1: Different Cases considered for analysis. a) Single S-D, b) Multiple S-D, c) Multiple S-D on a Wireless Link

4.1 Single Source-Destination

Consider a source and destination node with N paths between them known to the source. The number of hops on each path is given by the vector $\mathbf{R} = [r_1, r_2, \ldots, r_N]$. As pointed out in Section 3, we can assume that there are no common links between these paths. This can be assured by using one of many algorithms proposed in literature. Traffic arrival in the buffer is Poisson in nature and the service time is exponentially distributed. The queueing delay at a single node is exponentially distributed and is given as [14]:

$$f_{t_A,link} = \frac{1}{\alpha} e^{\frac{-t_A}{\alpha}} \tag{1}$$

where t_A >0 is the queueing delay (time in seconds) and $\alpha = 1/(\mu - \lambda)$, μ being the mean of the exponential distribution (service time) and λ is the rate of the Poisson process (arrival process).

For a steady state Jackson network, by Burke's theorem what comes out has the same rate as what goes in and so traffic leaving the node again is Poisson distributed and has the same rate λ. Assumig that all links in the network have the same μ for the service time (the i.i.d. assumption given in Section 3), the delay profile over the entire path is given as the convolution, which turns out to be the Gamma distribution [15]:

$$f_{t_A} = \frac{t_A^{r-1} e^{\frac{-t_A}{\alpha}}}{\Gamma(r)\alpha^r} = \Gamma_{t_A}(r, \alpha) \tag{2}$$

where r is the number of hops on this path, and $\Gamma(r)$ is the standard Gamma function. Note that since it is the single active pair case, there is no other traffic on the path and so λ and α are the same on all links (as μ is same).

Given this the probability that a packet meets its deadline is given as:

$$Prob(d_0) = \int_0^{d_0} f_{t_A}(t_A) dt_A = \int_0^{d_0} \Gamma_{t_A}(r, \alpha) dt_A \tag{3}$$

If we consider the system to be stationary and ergodic (which is true for steady state operation), then the percentage of packets that meet their deadline d_0 is the same as $Prob(d_0)$ as given in (3). Using the delay profile on all the available paths, a good formulation of the problem (Lagrangian) to maximize QoS is given as:

$$\mathcal{L} = min \left\{ \int_0^{d_0} \Gamma(r_1, \frac{1}{(\mu - \lambda_1)}), \int_0^{d_0} \Gamma(r_2, \frac{1}{(\mu - \lambda_2)}), \ldots, \int_0^{d_0} \Gamma(r_N, \frac{1}{(\mu - \lambda_N)}) \right\} + \nu(\sum \lambda_i - \lambda) \tag{4}$$

There is additional constraint of rate positivity, which is taken care by itself as the Lagrangian in (4) has to be maximized. The resulting flows from the solution to this problem guarantee the minimum QoS on each path in the statistical sense.

As evident (4) is tough to differentiate for reasons for formulating the *min* as a continuous function. An alternative formulation which is easier to differentiate and so maximize, is one which guarantees a QoS measure in the average sense and is given as:

$$\mathcal{L} = \frac{\sum \lambda_i \int_0^{d_0} \Gamma\left(r_i, \frac{1}{(\mu - \lambda_i)}\right)}{\lambda} + \nu(\sum \lambda_i - \lambda) \tag{5}$$

We note that the QoS on each path is weighted by the traffic flow alloted to that path λ_i and the sum is divided by the total flow λ to get the average %. Substituting for the Gamma distribution, we get:

$$\mathcal{L} = \frac{\sum \lambda_i \int_0^{d_0} \frac{t_A^{r_i - 1} e^{\frac{-t_A}{\alpha}}}{\Gamma(r_i)\alpha^{r_i}}}{\lambda} + \nu(\sum \lambda_i - \lambda) \tag{6}$$

Once again, the expression is complicated and at this stage can be simplified by invoking the Central Limit Theorem (CLT) which is very accurate for values of $r_i > 3$ or so as shown in Fig.2 where the two curves are indistinguishable.

So (6) is further simplified as:

$$\mathcal{L} = \frac{\sum \lambda_i \int_0^{d_0} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(t_A - \mu)^2}{2\sigma^2}} dt_A}{\lambda} + \nu(\sum \lambda_i - \lambda) \tag{7}$$

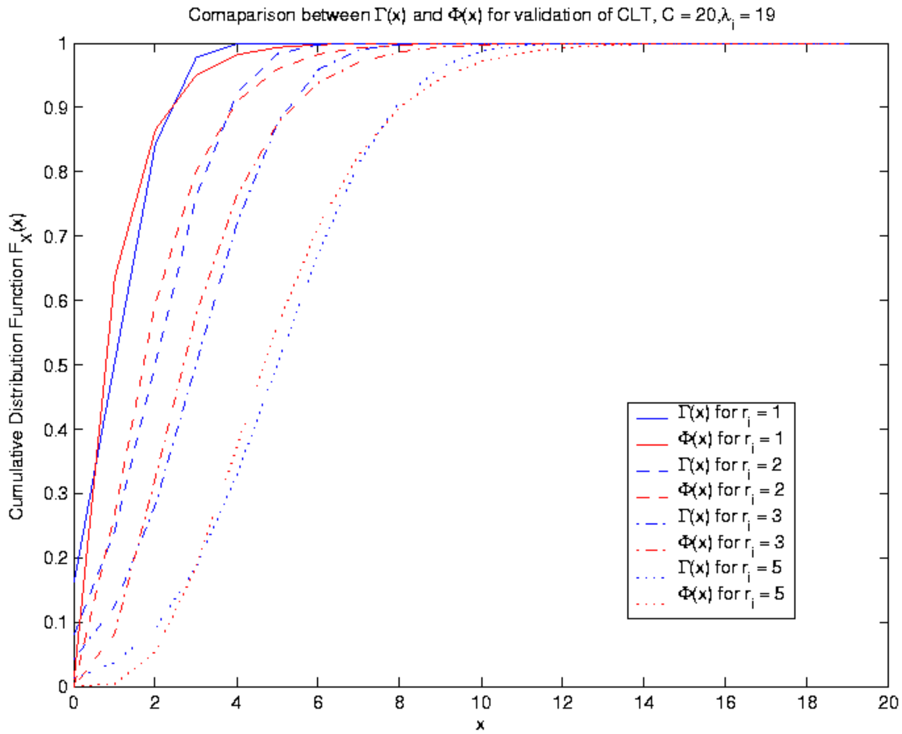

Figure 2: Validation of Central Limit Theorem

where $\mu = \frac{r_i}{(\mu-\lambda_i)}$ and $\sigma = \frac{r_i}{(\mu-\lambda_i)^2}$. Since the cdf of a Gaussian has a well known form (7) can be rewritten as:

$$\mathcal{L} = \frac{1}{\lambda}\sum \lambda_i \Phi\left((\mu-\lambda_i)d_0 r_i^{-\frac{1}{2}} - r_i^{\frac{1}{2}}\right) + \nu(\lambda - \sum \lambda_i) \tag{8}$$

Differentiating (8) and equating it to 0:

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = 0 \tag{9}$$

Rearranging terms we get:

$$-\lambda_i \Phi'\left((\mu-\lambda_i)d_0 r_i^{-\frac{1}{2}} - r_i^{\frac{1}{2}}\right) d_0 r_i^{-\frac{1}{2}} - \left((\mu-\lambda_i)d_0 r_i^{-\frac{1}{2}} - r_i^{\frac{1}{2}}\right) = \nu\lambda \tag{10}$$

which can be written as:

$$g(\lambda_i) = \nu\lambda \tag{11}$$

where $g(\lambda_i)$ is same as the L.H.S. of (10). Now we need to factor out λ_i to get the optimal solution for λ_i, but the equation obtained after so many simplifications is still transcedental and cannot be simplified to get a closed form solution for λ_i and so no intuitions !!!. In our effort to obtain a closed form solution to (10), we do a Taylor series expansion of $g(\lambda_i)$ near μ or we make the assumption of a busy network so the rates to be transmitted are close to maximum data rate supported on the link (link capacity)[4]. Essentially, we express g as:

$$g(\lambda_i) = g(\mu) + (\lambda_i - \mu)g'(\mu) \tag{12}$$

[4]We note that this is actually the network condition when it handles Video-on-Demand (VoD) systems or in general for a systems supporting high-rate video data [16].

From here on we replace μ by C, to emphasize the fact that we are operating close to link capacity. Equating the corresponding terms we get the linear equation as:

$$g(\lambda_i) = \frac{-Cd_0}{\sqrt{r_i}}\frac{1}{\sqrt{2\pi}}e^{-\frac{r_i}{2}} + (\lambda_i - C)\left[\frac{-d_0}{\sqrt{r_i}}\right]\frac{e^{\frac{-r_i}{2}}}{\sqrt{2\pi}}[-Cd_0 + 1] - \frac{1}{\sqrt{2\pi}}e^{\frac{-r_i}{2}} \tag{13}$$

Neglecting the $+1$ in comparison to $-Cd_0$, we get:

$$g(\lambda_i) = \frac{-Cd_0}{\sqrt{r_i}}\frac{1}{\sqrt{2\pi}}e^{-\frac{r_i}{2}} + (\lambda_i - C)\frac{1}{\sqrt{2\pi}}e^{\frac{-r_i}{2}}\left[\frac{d_0^2 C}{\sqrt{r_i}} - 1\right] \tag{14}$$

Neglecting -1 in comparison to $d_0^2 C/\sqrt{r_i}$, we get:

$$g(\lambda_i) = \lambda_i\left(\frac{1}{\sqrt{2\pi}}e^{\frac{-r_i}{2}}\frac{d_0^2 C}{\sqrt{r_i}}\right) - \frac{1}{\sqrt{2\pi}}e^{\frac{-r_i}{2}}\frac{Cd_0}{\sqrt{r_i}}[1 + Cd_0] \tag{15}$$

Again neglecting $+1$ in comparison to Cd_0 and making the substitution:

$$f(r_i) = \frac{1}{\sqrt{2\pi}}\frac{d_0^2 C}{\sqrt{r_i}e^{\frac{r_i}{2}}} \tag{16}$$

gives the simplified expression for g as:

$$g(\lambda_i) = f(r_i)\lambda_i - Cf(r_i) \tag{17}$$

We see that all this neglections make sense if the application is a video over an adhoc network with typical values of these parameters as $C = 20$, $d_0 = 1$ and $3 < r_i < 10$ [16].

Substituting this in (11), we get a closed form expression for $\lambda_{i,opt}$ in terms of the Lagrangian constant ν (which is to be found) as:

$$\lambda_{i,opt} = C + \frac{\nu\lambda}{f(r_i)} \tag{18}$$

Using the flow-conservation constraint $\sum \lambda_i = \lambda$, we get ν as:

$$\nu = \frac{\lambda - N_{th}C}{\lambda\sqrt{2\pi}\frac{1}{d_0^2 C}\sum_{i=1}^{N_{th}}\sqrt{r_i}e^{\frac{r_i}{2}}} \tag{19}$$

Substituting this in (18), we get:

$$\lambda_{i,opt} = C - \frac{\sqrt{r_i}e^{\frac{r_i}{2}}}{\sum_{i=1}^{N_{th}}\sqrt{r_i}e^{\frac{r_i}{2}}}(N_{th}C - \lambda) = C - h(r_i) \tag{20}$$

where N_{th} is analogous to the γ_0 for the optimal power allocation in the waterfilling expression to maximize capacity as given in [17]. It is the minimum number of paths required so that $(N_{th}C - \lambda) > 0$ and so λ_i is both positive and $< C$ for all i. Well we have already spelt out the name of the scheme this scheme is analogous to - *waterfilling*, as verified by:

$$\lambda_{i,opt} + h(r_i) = C \tag{21}$$

Since $h(r_i)$, increases with the number of hops r_i on a path, the result in (21) suggests that the optimal flow partitioning on the different multipaths is analogous to the waterfilling phenomenon, where we put more rate on the path which has a lesser value of $h(r_i)$ (the counterpart of fading in this context). Hence the optimal flow partitioning in effect also minimizes the number of hops as pointed out in the beginning.

At this point we make a note that the solution obtained is the optimal solution only if the Lagranagian $\mathcal{L}$ is a convex function of the flow λ_i for all i. We show this with the help of Fig. 3.

Before we go the to the multiple node case, we point out that the result derived, since it used the CLT, is valid even if the arrival process and service distribution are not Poisson and exponential respectively but statistical and i.i.d. for CLT to hold. The only term that changes in (21) is $f(r_i)$.

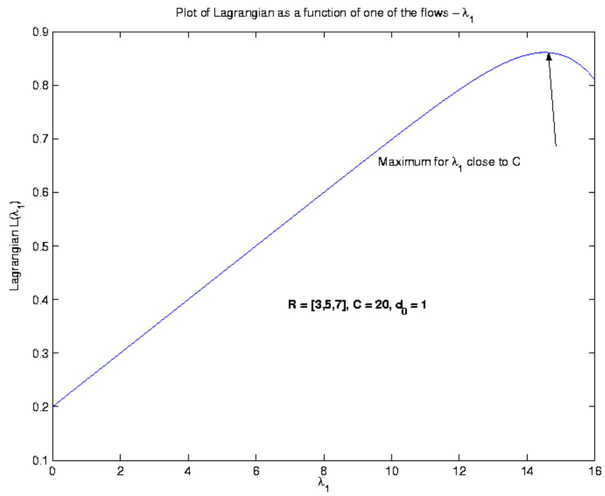

Figure 3: Plot of lagrangian for a Single S-D

4.2 Multiple Source Destination: Distributed Algorithm

A global formulation of the Lagrangian that maximizes the QoS for all the active sources and destinations can be given as:

$$\mathcal{L} = \sum_j \left\{ \frac{1}{\lambda_j} \sum_k \left(P_{kj}\lambda_k \left[\Phi\left(\frac{d_0 - \mu}{\sigma}\right) - \Phi\left(\frac{-\mu}{\sigma}\right) \right] \right) + \nu_j (\sum_k P_{kj}\lambda_k - \lambda_j) \right\} \tag{22}$$

where μ is the mean (different from what it meant in the previous section) of the delay profile on the kth path and σ^2 is the variance; these are given as:

$$\mu = \sum_i \frac{R_{ik}}{C - \sum_l R_{il}\lambda_l} \tag{23}$$

$$\sigma^2 = \sum_i \frac{R_{ik}}{(C - \sum_l R_{il}\lambda_l)^2} \tag{24}$$

where j sums over the number of active S-D pairs, k sums over the number of paths used by all of them for transmission and l sums over the number of links in a path. These are linked through the P and R coefficients defined as:

1. $P_{kj} = 1$ if source j can use path k for transmission, 0 otherwise.
2. $R_{ik} = 1$ if the kth path consists of the ith link, 0 otherwise.

This seems to be the direct extension of the formulation in previous section for the single S-D case and can be solved using MATLAB to obtain optimal flow-partitioning for all the sources for the paths that are available (as evident a closed form solution for (22) would be very complicated) as was done for a similar problem formulation in [9].

But we are interested in the distributed algorithm where there can be no global optimization. So we take the part from (22), specific to the jth source as:

$$\mathcal{L}_j = \frac{1}{\lambda_j} \sum_k \left(\lambda_k \left[\Phi\left(\frac{d_0 - \mu}{\sigma}\right) - \Phi\left(\frac{-\mu}{\sigma}\right) \right] \right) + \nu_j (\sum_k \lambda_k - \lambda_j) \tag{25}$$

where the P coefficients have been dropped with the understanding that the paths in the summation are specific to the jth source. The effect of the presence of other nodes is captured through the μ and σ expressions as given in (23) and (24).

At this point we stop and make a note that partially differentiating (25) might give the optimal solution as in the analysis for previous section and so a distributed algorithm. We tried this, however, the derivative of the Φ function gives all kinds of *ugly* expressions (as in the previous section) which may not be possible for the source node to know especially if we relax the assumptions that the source node has perfect knowledge of the number of hops on each path, link capacities as we discussed in Section 3. So we not only wish to come up with a distributed algorithm but one with a really simple implementation or rather a practical implementation such that quantities involved in the expression are either trivially known to the sender or can be easily fed back by the destination (here we assume a flawless feedback path).

With this in mind, we rewrite (25) as:

$$\mathcal{L}_j = \frac{1}{\lambda_j} \sum_k \lambda_k \text{qos}_k + \nu_j (\sum_k \lambda_k - \lambda_j) \tag{26}$$

where qos_k is the simply the Φ terms in (25) which represent precisely the quality of service (qos) measure that we started with i.e. the percentage of packets sent on the kth path that meet their deadline d_0.

Differentiating (26) and equating it to 0, we get:

$$\frac{\partial \mathcal{L}_j}{\partial \lambda_k} = \frac{1}{\lambda_j} \left[\lambda_k \frac{\partial \text{qos}_k}{\partial \lambda_k} + \text{qos}_k \right] + \nu_j = 0 \tag{27}$$

but this is a very standard method in the literature referred to as the *Gauss-Newton* method used for obtaining optimal solution by linearizing non-linear function about the point of operation.

Solving for ν_j using the flow conservation constraint, we get an expression for the optimal flow partition on kth path as:

$$\lambda_{k,opt} = \frac{\lambda_j}{\partial q_k} \left[\frac{1}{\sum_k \frac{1}{\partial q_k}} \right] \tag{28}$$

where $\partial q_k = \partial \text{qos}_k / \partial \lambda_k$. We notice the following about this result:

1. ∂q_k is always -ve $\forall k$. Hence $\lambda_{k,opt}$ is positive.

2. ∂q_k denotes the fall in QoS on kth path when we try to push in more rate through that path by marginally increasing λ_k. This happens as either the path has some ambient traffic already or has a limited capacity and so increasing λ_k results in congestion and so reduced number of packets achieving their deadlines.

3. (28) suggests that we put more rate on paths where ∂q_k is less (or that path is less congested), which agrees with intuition.

4. the term in [] in (28) is like the equivalent resistance of a set of resistors in parallel and so is less than the minimum of all the quantities in the summation.

5. hence the coefficient of λ_j is <1 which implies $\lambda_k < \lambda_j$ and $\sum_k \lambda_k = \lambda_j$, satisfying the definition of partition.

To develop the distributed strategy substitute (28) back into (27), to get:

$$\frac{\partial \mathcal{L}_j}{\partial \lambda_k} = \frac{\lambda_k}{\lambda_j} \partial q_k - \frac{1}{\sum_k \partial q_k} \tag{29}$$

Given this, a good distributed strategy which converges to the optimal, when it converges is:

$$\lambda_k(t+1) = \lambda_k(t) - \alpha \left[\frac{\lambda_k(t)\partial q_k(t)}{\lambda_j} - \frac{1}{\sum_k \frac{1}{\partial q_k(t)}} \right] \tag{30}$$

where t denotes a time frame over which a set of flows are maintained on each path and their resulting QoS is measured and fed back to the sender by the destination. The algorithm works as follows:

1. $\alpha > 0$ is small and dictates convergence. Details about α are covered in 4.2.1.

2. when the term in the [] in <0 , we increase the k flow and when it is >0, we reduce it so that the flow converges to the optimal as given in (28), if it converges.

3. source j starts with some initial flows on a set of given K paths ($k = 1, \ldots, K$). We take K to be given to all the source nodes which is true in practice where each node knows the possible set of paths available to it.

4. for the first two time frames (30) is not used. For the first time frame each source j chooses some initial flows on each path (which should be a good initial guess). In the next time frame it changes all these flows marginally (maintaining flow conservation, so some of the flows are increased and some decreased). The destination node feeds back the marginal change in QoS (the change in the % of packets meeting their deadline).

5. during this discussion, we have implicitly assumed that over a time frame the flows of the other sources remain constant so that the QoS measure is a reliable number or nodes are synchronized. This may not be true always in which case convergence is slowed down.

6. due to a discretization going on at the destination (number of packets being not a real number but an integer), there is a rounding off in the number of packets meeting their deadline, which can make ∂q_k 0 at times (say for N_0 paths), in which case (30) involves a division by 0 and is meaningless. From (28) and the discussion that follows it, a 0 ∂q_k means that the entire λ_j should be put on the kth path. But, taking into account the random effects that may take place during convergence (due to changes in flow by the other active sources on common paths/links), λ_k is not instantly made equal to λ_j. So we do the following:

   1) Handling $\partial q_k = 0$: $\lambda_k(t)$ is increased by α and (30) is used for the remaining paths using a modified λ_j reduced by the amount $N_0\alpha$.

   2) Satisfying the Bounds: Since increasing λ_k in this way can exceed the capcity C. We set $\widetilde{\lambda_k}(t+1) = \min(C^-, \lambda_k(t) + \alpha)$, where we use C^- as exactly close to C delay blows up.

   3) Satisfying Flow conservation: At last we make sure that flow is conserved, for this we set:

$$\lambda_k(t+1) = \widetilde{\lambda_k}(t+1) \frac{\lambda_j}{\sum_k \widetilde{\lambda_k}(t+1)} \tag{31}$$

7. we now discuss the issues relating to a good initial guess, convergence rate and convergence.

4.2.1 Convergence

The following can be said about convergence:

1. the value of α decides the rate of convergence. A very small value of α means slower convergence but ensures convergence (threoretically) for all values of initial conditions as the systems remains stable [18]. We have classified the statement just said because in our case, due to the discretization, a very small value of α makes ∂q_k 0.

2. large value of α means faster convergence, but convergence takes place only for a good initial guess. For a bad initial guess and large α, the flows quickly diverge (as shown in Section 5)

3. so the choice of α is a design issue.

4. to choose appropriate initial conditions, we analyze the expression for ∂q_k obtained from (25):
$$\partial q_k = \frac{\partial}{\partial \lambda_k}\left[\Phi\left(\frac{d_0 - \mu}{\sigma}\right) - \Phi\left(\frac{-\mu}{\sigma}\right)\right] \tag{32}$$
where the partial derivative of the first term is given as:
$$\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{d_0 - \mu(\lambda_k)}{\sigma(\lambda_k)}\right)^2} \left\{\frac{-\sigma(\lambda_k)\mu'(\lambda_k) - \sigma'(\lambda_k)(d_0 - \mu(\lambda_k))}{\sigma^2(\lambda_k)}\right\} \tag{33}$$
where ' denotes partial differentiation and the partial derivative of the second term can be obtained similarly. A close analysis of (32) gives that both the exponential and the rational terms in (33) increase as $\mu(\lambda_k)$ or $\sigma(\lambda_k)$ increase, which increase with the number of hops (23), (24). Effectively ∂q_k increases with the number of hops on the kth path.

5. hence a good initial partition of the flows is made according to the number of hops, which was precisely the result found in the previous section for a single S-D pair.

Once again, we show the validity of the solution obtained by showing that the Lagranagian $\mathcal{L}$ is a convex function of the flow λ_k for all k. We show this with the help of Fig. 4.

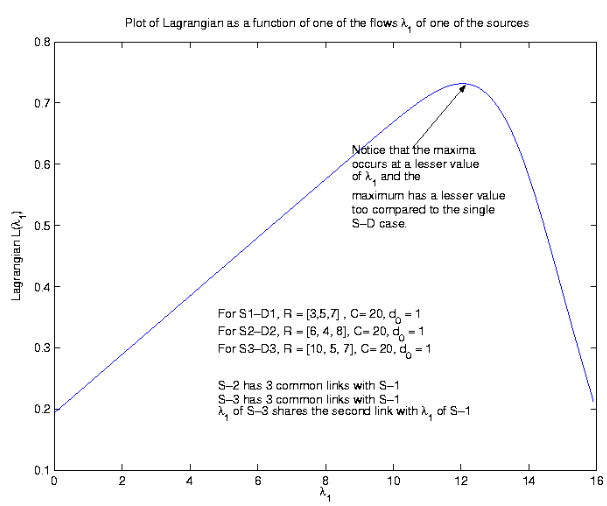

Figure 4: Plot of lagrangian for Multiple S-D's

4.3 Retransmissions

In this section, we consider the effect of a fading wireless channel where there is an increase of delay for packets arising due to corruption/loss of packets and subsequent retransmissions. To analyze this we make the simplyfying assumption that the wireless channel (link between two consecutive nodes in the multihop chain) is modelled by a single parameter p, which denotes the probability with which a packet transmitted by a node is successfully received at the other node (note that p can be different for the different links in the multihop chain). Given this, the number of *retransmissions* n_s till the packet is successfully transmitted between two nodes has a probability mass function given as:

$$P_{N_s}(n_s) = p^{n_s}(1-p) \tag{34}$$

which is the standard *Geometric* distribution, for which the mean (number of retransmissions) is given as:

$$\text{mean of} N_s = \mu_{N_s} = \frac{1}{p} \tag{35}$$

Assuming that a retransmitted packet is handled in a way identical to the transmitted packet, we effectively get an increase in the number of hops by μ_{N_s} i.e. the link between the nodes which was equivalent to a single hop is now equivalent to $\mu_{N_s} + 1$ hops.

At this point we note that p could also be modelled as an effective increase in the arrival traffic rate at the source node which now becomes $\lambda_k/(1-p)$. However, since our concern in this project is to find optimal flow partitioning of a constant flow over a given set of paths, we model the effect of p into the number of hops for which we also have a nice result that the flow on a path is waterfilling over the number of hops. This is intuitively also satisfying as we will send less rate on a path which has a small p (or for our case more number of hops). Hence the analysis for both the single and multiple S-D cases can be applied for imperfect wireless links with p going inside the QoS expression in (7) or (26) as the number of hops.

5 Simulations

The distributed algorithm developed is simulated for the network topology shown in Fig. 1(b) in Section 4.

We run the distributed algorithm for three cases:

1. Case I: α is small and our initial guess is good, specifically generated using the result of Section 4 for the single S-D case - initial rates are given as waterfilling over the number of hops. For this case, we see that the algorithm converges in barely 15 time frames to within 10^{-5} accuracy on the flows.

2. Case II: α is large (50 times the earlier value) but the initial guess is good. For this case, we see that the algorithm diverges.

3. Case III: α is large again and the initial guess is also bad. For this case, as expected, we see that the algorithm diverges.

The results are shown in figures Fig. 5 to Fig. 7. A close look at the figures reveal the closeness to which the results derived in this project are valid.

6 Conclusion

Thus, in this project we:

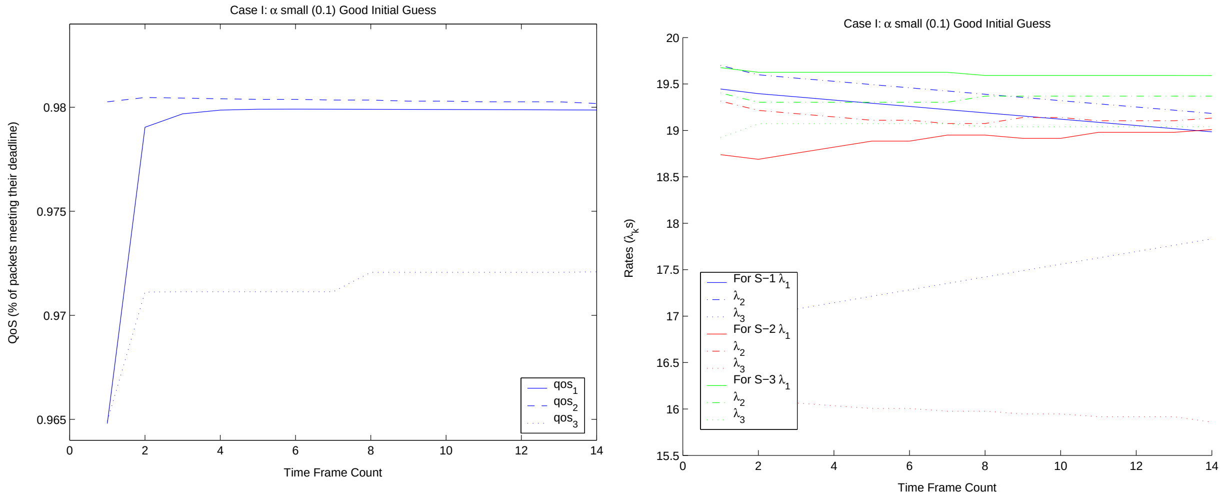

Figure 5: Case I

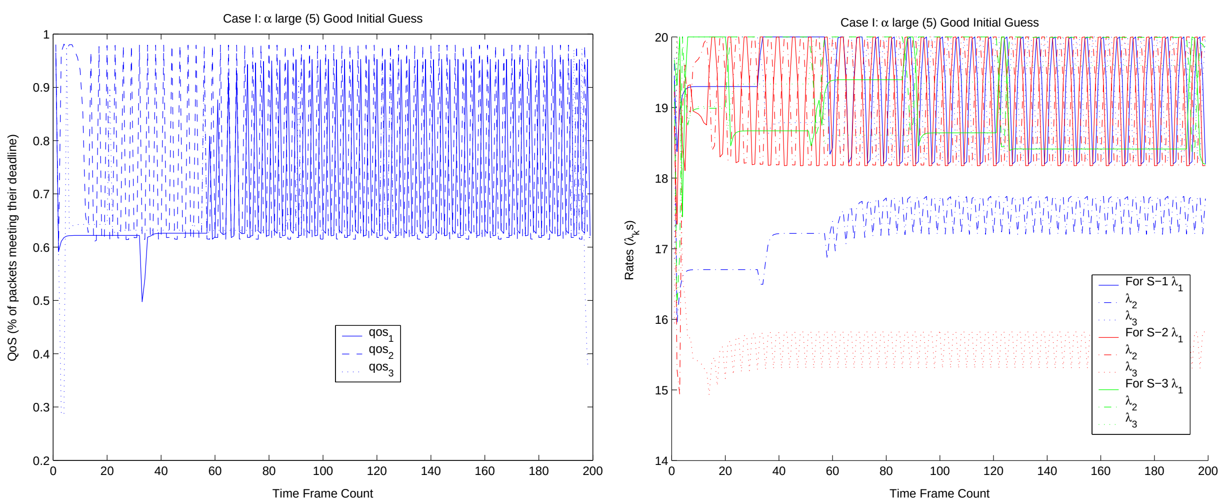

Figure 6: Case II

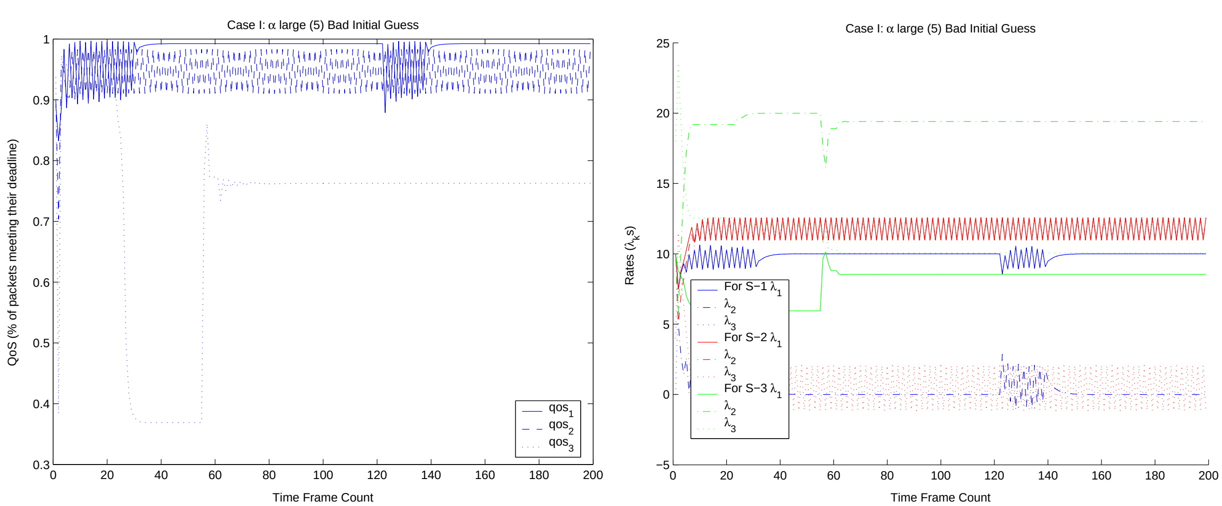

Figure 7: Case III

1. found the optimal strategy of flow partitioning between the available multipaths to maximize QoS when a single source-destination (SD) pair is active.

2. developed a distributed, isotropic flow partitioning algorithm and studied its rate of convergence.

3. studied the effect of packet loss and retransmissions on the optimal strategy.

which are precisely the same as the objectives that we started with in the EE359 project proposal.

References

[1] G. Bernat, A. Burns and A. Llamosi, *"Weakly Hard Real-Time Systems"*, IEEE Trans. Comp., 50(4), 308-321, 2001.

[2] S. Prameswaran, *"Wireless Ad-Hoc Networks"*, available at http://www.ece.msstate.edu/research/titl/presentations/AdHoc.pdf

[3] Eric Setton, Y. J. Liang and Bernd Girod, *"Multiple Description Video Streaming over Multiple Channels with Active Probing"* , ICME-2003.

[4] A. Jain, R. Agarwal, S. K. Bose, *"Distributed Video Streaming with Multi-Dimensional Multiple Description Coding"*, submitted to IEEE Comm Letters.

[5] Hemant Chaskar and U. Madhow, *"Wireless Link Shaping for Serive Guarantees"*, ISIT 1998.

[6] Vijay Sivaraman and Fabio Chiussi, *"End-to-End Delay Guarantees using Earliest Deadline First (EDF) Packet Scheduling"*, Globecom-99. 1, 1307-1312, 1999.

[7] Martin Reisslein, K. W. Ross and Srini Rajagopal, *"Guaranteeing Statistical QoS to Regulated Traffic: The Multiple Node Case"*, IEEE Conf. on Decision and Control, 1, 531-538, 1998.

[8] A. Elwalid, D. Mitra and R. H. Wentworth, *"A New Approach for Allocating Buffers and Bandwidth to Hetrogenous Regulated Traffic in an ATM Node"*, IEEE JSAC, 13(6), 1115-1127, Aug 1995.

[9] Eric Setton, Xiaoqing Zhu and Bernd Girod, *"Congestion Optimized Multipath Routing of Data and Video over Ad Hoc Networks"*, presented at the First SNRC Project Meeting at Stanford University, Fall 2003.

[10] Woo-Jin Choi and S. Tekinay, *"An Efficient Table Driven Routing Algorithm for Wireless Ad Hoc Networks"*, IEEE 54th VTS, 4, 2604-2608, Fall 2001.

[11] Elizabeth M. Royer and Chai-Keong Toh, *"A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks"*, IEEE Personal Comm., 46-55, Apr 1999.

[12] Sung-Ju Lee, Mario Gerla and Chai-Keong Toh, *"A simulation study of Table-Driven and On-Demand Routing Protocols"*, IEEE Network, 48-54, 1999.

[13] Sanjay K. Bose, *"An Introduction to Queueing Systems"*, Kluwer/Plenum Publishers, 2001.

[14] Shantanu Rane, *"Modeling End-to End Packet Delays for Networks Employing Link Layer Retransmission"*, EE359 Course Project Fall 2002, available at http://www.stanford.edu/ srane/courses/ee359/ee359report.pdf.

[15] *"Gamma Distribution on the Mathworld Website"*, available at http://mathworld.wolfram.com/GammaDistribution.html.

[16] Shu-Ngai Yeung and J. Lehoczky, *"End-to-end Delay Analysis for Real-Time Networks"*, 22nd IEEE Real-Time Systems Symposium, 299-309, 2001.

[17] Professor Andrea J. Goldsmith, *"EE 359 Course Reader"*, Stanford University, Fall 2003.

[18] Professor Stephen Boyd, *"EE263 Notes"*, Stanford University, Fall 2003.