Average utilization analysis in path swapping of call routing

27/1/99

Abstract

We are investigating the cost of routing oscillations in connection-oriented networks and the quantitative effects of network and traffic parameters to it. We provide an analysis of the incurred under-utilization of a network path based on the path's capacity, the update rate of network state information and the arrival and departure processes of the traffic involved. We describe a prediction model for the under-utilization and compare it to experimental data.

Introduction

We are considering the call source routing model for a connection network. In a simplified model, the routing decision process consists of choosing the best path amongst a set of possible ones based on network state information that has reached the decision element. This decision process may be suboptimal for two reasons. Firstly, the optimality criteria may make it hard to find the right solution in the time allowed. Secondly, the network state information may be out-of-date since it arrives at the decision element with some delay and can be provided at some finite, small or big, rate. In the following, we will focus on this second parameter of the problem; the only quality request a call is allowed is the desired bandwidth for the connection, leading on a decision process of selecting the path with the highest available bandwidth. The duration of the calls, while known and used in the analysis is not used in the decision process.

The network update process is not the same in all existing systems. It can be uniform across the whole network, or different parts of it can transmit at different rates and with different delays. Also it can be static over time or vary depending on the level of system utilization. In the following, we are going to focus in the simplest case where there is a fixed update rate and information on the whole network arrives at the same time. Thus, there are update points, at which the information on the routing element is renewed. We also assume that there is no delay due to the physical restrictions of the network, thus the information is accurate at the update point. In real networks, the propagation delay is quite small compared to the update interval, thus playing minor role in the overall performance of the system.

The routing oscillations, we are interested in, occur when each time we reach an update point, the new information changes the best path for a pair of source-destination points, cycling through a smaller or larger set of possible paths. The oscillations are bounded to the update points: It is easy to deduct from the above paragraph that during an update period, the interval between two consecutive update time points, the decision process gives almost always the same path for a pair of source-destination points. The process can be approximated by choosing the best path in the beginning of the period and continuously using it until the next update point. Simpler oscillations occur when the possible paths are few and of similar capacity; however, under the 'best path' decision process, some kind of oscillation will always occur. In the following we will focus on a very simple form of such oscillation between 2 equal capacity paths. Hopefully, it works for more complicated cases also.

The metric we are analyzing, the under-utilization of a path, is inherent in path swapping in a routing scheme. We assume our clients are calls that need to be routed through one of possible paths. Normally, only one path can be selected at any given point, giving an under-utilization of $(k-1)/k$ where k the number of possible paths. However, in a connection-oriented network the path for each call is fixed upon its arrival and not affected by future routing decisions, even if the call spans more than one update periods. This allows us to achieve higher utilization by switching from more used (i.e. currently allocated) paths to less used. However, a not-used path will eventually empty out as calls allocated to it depart. This increases it's chances of being selected again. We are interested in this under-utilization oscillation both on average and as a real time process. In the following we describe how to predict the average under-utilization of the paths involved and suggest some approaches for describing the real time case. The distant goal is to create a decision process for an optimal update/switching rate based on network and traffic attributes.

Network and Traffic assumptions

- Calls have a constant arrival rate and a random holding time, given from an exponential distribution. Also their bandwidth requirement is a constant value. We want to alleviate this last assumption later. The constant arrival rate simulates poisson arrival of much finer granularity (a bigger number of calls with much smaller requirements). Note that for a time period t, the average number of arrivals through a poisson process is $\lambda * t$ where λ the poisson distribution parameter.

- Equal bandwidth paths. This makes with high probability every update point a path switching point; thus the update rate is equal to the switching rate. It's easy to see that from the symmetry of the network; we'll try to quantify the probability of deviation below. We also want to alleviate this assumption and generalize the results to more complicated networks.

- Only 2 available paths currently. With this and the previous assumption we can focus on only one path with the arrival process being switching periods with arrival rate λ and 0. More paths or paths with different capacities will lead to arrival processes with more inactive or active periods instead of an 1:1 analogy. Note that each path has an average arrival process rate of $\frac{\lambda}{2}$.

- In the phone network, call paths are usually not changeable once decided upon; the delay incurred from handoff of a connection to a new path is intolerable. We pose the same restriction by not allowing the redirection of a call based on new information.

- Calls that arrive at a full path are just dropped and lost.

Under-Utilization Analysis

In the rest of the report we will consider a single path, one element of a pair of paths that take part in the routing oscillation. Also, we will treat the path as a single link with capacity equal to the bottleneck capacity of the path.

The under-utilization of the link depends on the arrival and the departure processes of the set of incoming calls. As we said before, the arrival process are interleaving periods equal to the update period T, with arrival rate λ and 0. We'll refer to a pair of active and inactive periods as an update cycle. Modeling the departure process would allow us to model the under-utilization also. However, this is not trivial since the holding time is a random variable. It is thus impossible to create a real time model for the departure process. Our goal is to create a model for the 'average' departure process in every update cycle, so we can compute the average under-utilization of the system over a large number of update cycles.

There are two approximation approaches we can follow and as we show later their validity changes with the network state, leading to the need to use both for the overall description of the system.

In the first approach, we apply an averaging argument in the holding time process itself. Approximate it with a constant average holding time, say H. Then the departure process becomes deterministic and is the time shift of the arrival process by H, except in cases where that would violate the capacity restrictions of the link. Of course, this is not true in a real system. The existing variance would lead to a deviation from the predicted values. However, for a constant variance σ, the deviation of the stochastic holding time process from the constant value approximation process is analog to the magnitude of the average value. Small values of H lead to small error in the approximation. Note also that H is inversely analog to the maximum departure rate. The maximum number of calls in the path is determined by its capacity and the bandwidth request size and it's constant in this model. Thus, for small H, the maximum departure rate ρ_{max} is large. For $\rho_{max} \geq \lambda$, the departure process is not a bottleneck even in an active update period. It can 'follow' the arrival process and the utilization of the link depends on the time shift between the two processes, i.e. the value of H. This leads to the mathematical model (II.a) described below.

In the second approach, we approximate the departure process instead of the holding time process with a deterministic process. We make the assumption that, after the system stabilizes, the probability of a call departure is the same at all time points and not connected to the arrival process. Thus, we model the departure process as a constant rate process. Let that rate be ρ. The error in this approximation decreases as H becomes bigger. It is easy to see that for H bigger than or comparable to the update cycle time, the departure sequence for one cycle interferes with the following update cycles, leading to a smoothing out of the departure process. As H becomes bigger, ρ_{max} decreases. It becomes at first the bottleneck rate for an active period and for even larger H

the overall bottleneck rate for an update cycle (case (I) below).

As we mentioned before, we have to keep in mind another set of restrictions, based on the capacity of the path. No more than that can be allocated and of course, the path can empty out but then the departure rate drops to zero. In other words, the 'gap' between the two processes, arrivals and departures, can not be more than the capacity or less than zero. The chance of straining these restrictions, filling up or emptying out the path, increases as the update period increases. Of course, fill-ups occur in active periods and empty-outs in inactive ones. It is important to mention that this affects mainly the second approach described above. In the network states where the first approach is valid, the utilization of the path is directly analog to H and is not affected by the update rate.

Methodology

As it may already be obvious, we are treating the path at hand as a queuing system, where accepted calls correspond to job arrivals and call departures to job departures. The arrival rate carries through nicely. However, the departure rate doesn't because the call holding time corresponds to the sum of the waiting and service time in a simple queuing system. Indeed, calls start getting service immediately as they arrive in the system (routed through the path). Thus, the departure rate (isomorphically the service rate) depends both on the length of the queue and the actual waiting time that jobs (calls) have accumulated in the queue.

In the following, we display the 'cumulative' functions of the two processes, arrivals and departures. The values of these functions at a given point in time, give the total number of calls that have arrived (or departed) up to that point. Thus the actual values are not that important, but the difference between the two functions give the number of calls currently at the system. Thus, the area between them for a given time interval, denotes the total path resources used in this interval. From this it is easy to get the average utilization in this interval by dividing with the time elapsed.

Mathematical analysis

It is easy to see that $\rho = \frac{AvUtil}{H}$ (1) where $AvUtil$ the average utilization of the path and H the average holding time.

The average utilization of the path gives us the number of calls in the system on average. Since the calls remain on the system for H time units on average, again on average $\frac{AvUtil}{H}$ calls depart every time unit. Also, note that $\rho_{max} = \frac{Base}{H}$, where $Base$ the capacity of the path.
We distinguish the following cases:

(I) $\rho_{max} < \frac{\lambda}{2}$: ρ_{max} is the limiting factor in the one-path system. Calls depart slower than they arrive, thus we have call drops. The link doesn't need to be fully utilized because of the path switching. Since call drops are certain, we'll analyze the system from a full-path point in time.

a) $\rho < \frac{Base}{T}$: Note that $\rho * T$ is how much the system will empty during an inactive period. However, this can not be more than $Base$, since at that point the line is completely empty. First we look at the simple case when this doesn't happen. Since $\rho < \frac{\lambda}{2}$, the line will be filled by the end of the next active period.

Since the arrival process is unaffected by call drops, a full line is going to remain full until the end of an 'active' period. Assume t_a is such a point in time; then $[t_a, t_a + T]$ is an inactive period and $[t_a + T, t_a + 2 * T]$ is the next active period. As you can see in Fig.1, the link is expected to empty somewhat during the inactive period and then refill during the active period as overall demand is greater than the available capacity. The area between the arrival and the departure process corresponds to the under-utilization of the link. The size of this area is determined by λ, ρ and T and can be easily shown to be equal to

$$\frac{T^2 * \lambda * \rho}{2 * (\lambda - \rho)}$$

This is the under-utilization over a $2{*}T$ period, so the average under-utilization is $\frac{T*\lambda*\rho}{4*(\lambda-\rho)}$ per time unit. Let B be the capacity of the link, for short. Then

$$AvUtil = B - \frac{T * \lambda * \rho}{4 * (\lambda - \rho)}$$

However, as we seen before $\rho = \frac{AvUtil}{H}$. Replacing ρ in the previous equation, we get a formula from which we can compute $AvUtil$, as it's the single unknown. Thus we get:

$$AvUtil = \frac{(B + \frac{T*\lambda}{4} + \lambda * H) - \sqrt{(B + \frac{T*\lambda}{4} + \lambda * H)^2 - 4 * \lambda * B * H}}{2}$$

Let us define the above formula as $G1(H, \lambda, B, T)$.
Also $AvUnderUtil = B - AvUtil \equiv G1'(H, \lambda, B, T)$.
This formula is valid for $\rho < \frac{\lambda}{2}$, or equivalently $AvUtil < H * \frac{\lambda}{2}$,
or $G1(H, \lambda, B, T) < H * \frac{\lambda}{2}$.
This cutoff point can be computed for any of the parameters involved when considering the rest fixed. In the following we'll consider as H the changing parameter, because it is the one parameter directly connected to ρ.

In the cutoff point, $\rho = \frac{\lambda}{2}$, thus $0.5 * \lambda * H = AvUtil = G1(H, \lambda, B, T)$ and after some math the average duration at the cutoff point is

$$H_{cp} = \frac{2 * B}{\lambda} - \frac{T}{2}$$

Also $AvUtil_{cp} = B - \frac{T*\lambda}{4}$ and $AvUnderUtil_{cp} = \frac{T*\lambda}{4}$

b) $\rho > \frac{Base}{T} \Rightarrow \rho * T > Base$: At some point $t_1 = \frac{Base}{\rho}\ (modT)$ in the inactive period the line will be completely empty. This complicates a little the under-utilization area we are computing as we show below. Note that $\rho < \frac{\lambda}{2} \Rightarrow \lambda - \rho > \frac{\lambda}{2} > \rho > \frac{Base}{T}$. Thus, although the 'gap' at the end of the inactive period will be $Base$, it will be covered in the next active period.

Note also that in this case, we start the inactive period with a full path and finish with an empty path to begin the active period. We can equivalently start from the beginning of the active period with an empty path. This means that we get empty and full path time points with the same probability and thus, it is equivalent to compute either the under-utilization or the utilization depending which point we take as beginning of an update cycle.

As you can see from fig.2,

$$UnderUtil = \frac{B*B}{2*\rho} + B*(T - \frac{B}{\rho}) + \frac{B*B}{2*(\lambda-\rho)}$$

for a $2*T$ period and

$$AvUnderUtil = \frac{B}{2} - \frac{B^2}{4*\rho*T} + \frac{B^2}{4*(\lambda-\rho)*T}$$

Starting from the active period instead, there is some time point $t_2 = \frac{Base}{(\lambda-\rho)}\ (modT)$ where the link will fill up. After that point the arrival rate is equal to the departure rate; all extra calls are dropped. This complicates similarly the utilization area calculation. From fig.4 we can see that:

$$Util = \frac{B*B}{2*(\lambda-\rho)} + B*(T - \frac{B}{(\lambda-\rho)}) + \frac{B*B}{2*\rho}$$

for a $2*T$ period. Thus

$$AvUtil = \frac{B}{2} - \frac{B^2}{(4*T*(\lambda-\rho))} + \frac{B^2}{(4*T*\rho)}$$

which is equivalent to the formula for the under-utilization we computed above. Replacing ρ with $\frac{AvUtil}{H}$ we get (U denotes $AvUtil$):

$$U^3 - (\lambda*H + \frac{B}{2})*U^2 + (\frac{B*\lambda*H}{2} - \frac{B^2*H}{(2*T)})*U + \frac{(\lambda*B^2*d^2)}{(4*T)} = 0$$

(A1)

Solving it we get the corresponding curve in fig.3.

(II) $\rho_{max} > \frac{\lambda}{2}$: The arrival process is the limiting factor in the path. Calls depart faster than they arrive on general. Since the average arrival rate is $\frac{\lambda}{2}$, ρ can not be bigger than that. It may be smaller because calls drops are still possible (they are avoided completely only if $\rho_{max} > \lambda$ itself).

a) $\lambda - \rho < \frac{Base}{T}$: There is not enough time for the link to completely fill up during an active period.
While $\rho < \frac{\lambda}{2}$, the previous analysis holds because it doesn't matter how ρ is created (bottleneck congestion or transient congestion).

If $\rho = \frac{\lambda}{2}$, then from the first formula we get

$$AvUtil = \rho * H = \frac{\lambda * H}{2}$$

Thus

$$AvUnderUtil = Base - 0.5 * \lambda * H = G2(H, B, \lambda)$$

Note that

$$G2(H_{cp}, B, \lambda) = G1'(H_{cp}, \lambda, B, T)$$

(2)
since $G2$ is derived from (1) and $\rho = \frac{\lambda}{2}$ and the same hold for $H = H_{cp}$.

Note also that $G2$ is not affected from the value of T. The update rate doesn't affect the utilization when the duration of calls is small enough that the arrival rate is the bottleneck.

Another way to approach this is to consider the $MaxUtil$ in the link. Since $\rho \leq \frac{\lambda}{2}$,

$$MaxUtil = \rho * H \leq 0.5 * \lambda * H$$

Thus

$$MinUnderUtil = B - 0.5 * \lambda * H$$

which is $G2$ again.

b) $\lambda - \rho > \frac{Base}{T}$: At some time $t_2 = \frac{Base}{(\lambda - \rho)}$ $(modT)$ in the active period the link will fill up. After that point the arrival rate is equal to the departure rate; thus, the average arrival rate is smaller than $\frac{\lambda}{2}$ now, limiting also the departure rate. Thus the same analysis as in part (I) is valid.

FIGURE 3.a

FIGURE 3.b

FIGURE 3.c