IEEE Trans. on Networking, 7(6):885-896, Dec. 1999

Analysis of Multi-Path Routing

Israel Cidon[1], Raphael Rom[2], Yuval Shavitt[3]

Abstract

In connection oriented networks, resource reservations must be made before data can be sent along a route. For short or bursty connections a selected route must have the required resources to ensure appropriate communication with regard to desired QoS. For example, in ATM networks the route set-up process considers only links with sufficient resources and reserve these resources while it advances towards the destination. The same concern for QoS routing appears in datagram networks like the Internet when applications with QoS requirements need to reserve resources along pinned routes.

In this work we analyze the performance of multi-path routing algorithms and compare them to single path reservation that might be persistent, i.e., retry after a failure. The analysis assumes that the routing process reserves resources while it advances towards the destination, thus there is a penalty associated with a reservation that cannot be used.

Our analysis shows that while multipath reservation algorithms perform comparably to single path reservation algorithms, either persistent or not, the connection establishment time for multi-path reservation is significantly lower. Thus, multi-path reservation becomes an attractive alternative for interactive applications such as WWW browsing.

1 Introduction

Broadband integrated services digital networks (B-ISDN) are aimed to transport all electronic communication formats from e-mail to phone calls to home video. Communication in such high-speed networks is connection-oriented, i.e., before data can be transferred a connection should be established.

Data applications are bursty in nature. Thus either connections are short or data is transferred in bursts spread over time. E.g., WWW browsing establish many connections each for the delivery of a single entity in a URL.

In order to use the network resources efficiently, bandwidth reservations must be made to ensure high probability of data arrival to its destinations. In ATM PNNI standard [For96], reservations are performed while a search for a feasible route is conducted. If the search process reaches a point where sufficient resources are not available for reservation, it cranks-back several hops and then the search is continued from some intermediate point on the route. In the Internet, reservations for connection-oriented traffic can be done using RSVP [ZDE$^+$93] along the shortest path between routers. If a reservation cannot be made a new path might be calculated to try and accommodate the requested resources [ZES97].

In both ATM and the Internet the failure to set up a connection results in delay in the set-up process. In PNNI the delay is due to the time it takes for the reservation process to crank back, and the recalculation of the alternative route at the point where the search starts. In RSVP, the delay is due to the time-outs associated with the reservation process, and the need to restart the reservation process.

Recently [CRS97, Sha96], we suggested a family of multipath reservation algorithms that use multiple reservation processes concurrently for the same connection. The concurrency in the reservation process has the following merits

- A reservation failure in one (or more) links does not slow down the reservation process in other links.
- If several routes are available for reservation, the one that meets the application requirements the most can be chosen

In this work we analyze and compare the performance of multi-path algorithms with single path algorithms. The analysis is general in the sense that it does not take into account the design of any specific algorithm. In particular, the analysis does not capture the ability of the suggested algorithms [CRS97, Sha96] to work in any directed subgraph of the communication network. We look at two main performance measures: network throughput (goodput) and connection establishment delay.

It is important to note that, the throughput analysis given in this paper for multi-path algorithms serves as a loose lower bound of their performance. In particular the analysis assume that multiple routes considered for routing are node disjoint. The algorithms we suggest in [CRS97] benefit from node sharing between routes as this enable

[1] Dept. of Electrical Engineering, Technion, Haifa, Israel. E-mail: cidon@ee.technion.ac.il.

[2] Dept. of Electrical Engineering, Technion, Haifa, Israel; and Sun Microsystems, Mountain View, CA 94043 E-mail: rom@ee.technion.ac.il.

[3] Corresponding author. Bell Laboratories, Lucent Technologies, room 4G-627, 101 Crawfords Corner Rd., Holmdel, NJ 07733-3030. E-mail: shavitt@ieee.org. This author work was done in part while he was with the Technion, Haifa, Israel.

the *early release* of reserved resources while the reservation process is under way. Link sharing is not captured by the analysis as well.

Our analysis shows that multipath routing has slightly better throughput than single path routing, when no retries are allowed, and slightly worse when one or two retries are used. However, when the expected connection establishment time is compared multi-path routing has significantly shorter expected delay than single path routes. Since, as we mentioned above, the throughput analysis for multipath routing is a worst case lower bound, we believe that this paper shows that multi-path routing has an important role in future bursty applications. Hwang et al. [HKT95] have also found that blocking probabilities for sequential and parallel reservation algorithms are similar. However note that they use a different mathematical model as they concentrate mainly on the processing delay, and their algorithms can only run on tree subgraphs.

The competitiveness of multi-path routing might seems contra-intuitive. For telephone networks, it has been shown [Kel91, GKK95] that even trying alternative routes that are longer from the shortest route, may result in performance degradation in the form of higher block probability. However the telephone model network model does not apply to general networks. Most of the work done for routing in telephone network assume fully connected networks. In such networks there is no penalty in trying a blocked route as the information is available at the source, there is also no need to attempt multiple reservations since the model assumes knowledge of the resource availability in in all the relevant links at the source switch.

In data networks, where the number of hops in a route may be in the double-digit zone, these results do not hold. In particular, there is a significant cost both in increased blocking and in time delay for a failed attempt to reserve a route, since the blocking may occur close to the destination when most of the route is already reserved for the connection.

The rest of the paper in organized as follows. In the next section we describe the model of the network we analyze. In section 3 we analyze multi-path routing for the case where each route can support a single connection, in section 4 we analyze the case where consecutive retries are permitted upon failure. In the following section we look at the case where a route can support several connections, and in section 6 we analyze the connection establishment time. Section 7 holds the numerical results from the analysis of the previous sections. In the final two sections we discuss the implication of the results and point to algorithmic solutions triggered by this work.

2 The System Model

As mentioned in the introduction, the multi-path reservation algorithms can benefit from routes sharing nodes and links. However, in this analysis we assume that n disjoint routes are available between source and destination nodes. The competitiveness of multi-path algorithms under this worst case scenario gives way to promote the use of such algorithms. The advantage of this model is that we can easily quantify the effect of parallelism on the performance of the reservation algorithms.

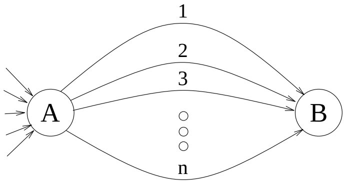

Figure 1: The analyzed system

Consider a source-destination pair of nodes that are connected by n disjoint routes, each of which can support m connections at a time. Let the source node be A and the destination node be B (see figure 1). The connection-request arrival process is Poisson with intensity λ. When a connection-request arrives at node A, there is no knowledge about the availability of the routes, thus one (or more) routes are selected randomly to attempt a reservation. The overall period of the reservation and the connection duration time is exponentially distributed with mean $1/\mu$.

Note that asymptotically, when the load is approaching infinity, all reservation methods have the same throughput. When the system is heavily congested, it will always fill to its capacity whether we use a single route reservation, or a multi-path reservation. In such high loads, the system is mostly moving between two states: full system and single vacancy. As soon as the system moves out of the full state it returns to this state again, since the high rate of incoming requests ensures that any available bandwidth is occupied at once.

When multi-path routing is used, the reservation algorithm selects randomly k $(1 \leq k \leq n)$ of the routes and tries to capture (reserve bandwidth) them, each of the routes has an equal probability to be selected. If more than one route is captured, only a single one is used for the connection while the others are released. The period until an unused route is released is exponentially distributed with mean $1/\theta$ $(1/\theta \leq 1/\mu)$. $1/\theta$ describes the average time it takes the system to signal the release of such a redundant reservation. Note that, $\theta = \mu$ models the case where the connection is short with regard to the reservation process. Thus, if several reservations succeed all the routes appear to be used (resources are reserved), while the destination is using only one of them, ignoring the rest. This happens, for example, when a short burst is sent preceded by a reservation request that tries to reserve sufficient resources on-the-fly [Tur92, BT92].

For single-path reservation algorithms, we also analyze the case where upon a failure to reserve the route, κ additional attempts are made. The time between successive additional attempts is exponentially distributed.

Table 1 indexes the cases considered in the paper and refer the reader to the relevant section for each case.

m	k	κ	other parameters	section
1	1	1		3.1.1
1	n	1	$\theta = \mu$	3.1.2
1 1	general 2	1 1		3.2 3.2
1	1	∞		4.1
1	1	2		4.2
1	1	3		4.3
general	1, 2, 3	1	$\theta = \mu$, $n = 3$	5
general	general	1		6.1
general	1	2,3		6.2

Table 1: An index for the cases considered in the paper.

3 Analysis of the Minimal Capacity Case

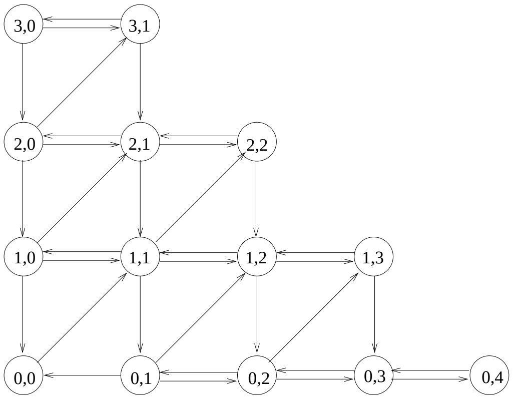

Figure 2: A Markov chain for a system with $n = 4$, $m = 1$, and $k = 2$

We concentrate in our analysis on the case when $m = 1$, i.e., the case where each route can support a single connection, as the relative computational simplicity of this case makes it possible to examine more aspects of the system. For this case, the above system can be modeled by a continuous-time Markov chain with $n(n+3)/2$ states as illustrated in figure 2 for the case where $n = 4$ and $k = 2$. Each state is represented by the ordered pair (t, s), where s is the number of routes that are used for transmission, and t is the number of redundant routes that were captured and are not used for transmission. The infinitesimal transition rates from state (t, s) to state (t', s'), $q_{t,s,t',s'}$ are (see Figure 3)

$$\begin{aligned} q_{t,s,t-1,s} &= t\theta \\ q_{t,s,t,s-1} &= s\mu \qquad (1) \\ q_{t,s,t+i,s+1} &= b(i, t+s)\lambda \\ & \qquad (0 \le i \le \min\{k-1, n-(s+t+1)\}) \end{aligned}$$

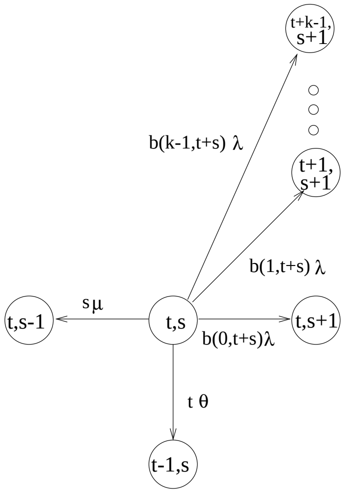

Figure 3: The transitions out of a state in the Markov chain for a system with $m = 1$ and general n and k values

where

$$b(i, \nu) = \binom{n-\nu}{i+1}\binom{\nu}{k-(i+1)} \Big/ \binom{n}{k}$$

We are interested in the connection reservation success probability, P_{suc}, that is proportional to the system throughput. P_{suc} is given by the ratio between the rate of the accepted requests, λ_{out}, and the rate of incoming requests, $\lambda_{in} = \lambda$. Thus, we can write

$$P_{suc} = \frac{\lambda_{out}}{\lambda_{in}} = \frac{\bar{N}\mu}{\lambda} = \rho^{-1} \sum_{t=0}^{n-1} \sum_{s=1}^{n} s \cdot \pi_{t,s} \qquad (2)$$

where $\bar{N}$ is the average number of served connections in the system, and $\rho = \lambda/\mu$. The second transition in Eq. 2 is due to Little's Law.

To obtain P_{suc}, the system steady state probabilities, $\pi_{t,s}$, should be found by solving the system equilibrium equations, $\vec{\pi}Q = 0$ (Q is derived directly from the infinitesimal transition rates, $\vec{\pi}$ is the vector of steady state probabilities), together with the probability conservation relation, $\sum_{(t,s)} \pi_{t,s} = 1$. This numerical solution requires $O(n^{2(2+\alpha)})$ basic operations, where $O(x^{2+\alpha})$ is the number of operations used by the matrix inversion algorithm for an $x \times x$ matrix (for the best known matrix inversion algorithm $\alpha > 0.5$). I.e., the solution requires $O(n^5)$ operations. In the following sections, we shall describe methods to make the problem more tractable. For two special cases: when $k = 1$ and when $k = n$ and $\theta = \mu$ we present a closed form solution. For other cases we present a recursive solution that requires only $O(k \cdot n^3)$ operations.

3.1 Analysis of Special Cases

3.1.1 Single-path reservation

Single path-reservation is the case when $k = 1$. In this case, t always equal to 0, thus the system can be modeled

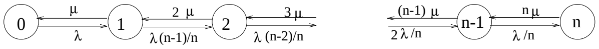

Figure 4: A Markov chain for a system with $k = 1$

by an $n+1$-state birth-death process with transition rates

$$\begin{aligned} \lambda_s &= \frac{n-s}{n}\lambda \\ \mu_s &= s\mu \end{aligned} \tag{3}$$

as depicted in figure 4. The average number of active connections is given by

$$\bar{N} = \sum_{s=1}^{n} s \cdot \pi_s = \frac{\rho n}{n+\rho}$$

where $\rho = \lambda/\mu$. Thus,

$$P_{suc} = \frac{\lambda_{out}}{\lambda_{in}} = \frac{\bar{N}\mu}{\lambda} = \frac{n}{n+\rho} \tag{4}$$

where the second transition in Eq. 4 is due to Little's Law. The throughput in given by

$$\mathcal{T} = \lambda P_{suc}$$

3.1.2 Greedy reservation with maximal penalty

When the penalty for over-reservation is maximal, i.e., $\theta = \mu$, the system can be modeled by a single number that represents the total amount of occupied routes, i.e., by an $n+1$-state Markov chain with transition rates

$$\begin{aligned} q_{s,s-1} &= s\mu \\ q_{s,u} &= b(\max\{0, s-(u+1)\}, s)\lambda \\ &\qquad \max\{k, s+1\} \le u \le \min\{n, s+k\} \end{aligned}$$

This system can be solved with $O(n^{2+\alpha})$ operations for any k.

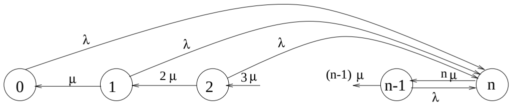

Figure 5: A Markov chain for a system with $\mu = \theta$ and $k = n$

A greedy reservation algorithms tries to reserve in all n paths which is captured in our model by setting $k = n$. For this case, (see figure 5), we can write the equilibrium equations

$$(\lambda + s\mu)\pi_s = (s+1)\mu\pi_{s+1} \qquad s = 0, 1, \ldots, n-1$$

that yield

$$\pi_s = \frac{(\rho+s-1)(\rho+s-2)\cdots\rho}{s(s-1)\cdots 2\cdot 1}\pi_0 = \binom{\rho+s-1}{s}\pi_0 \tag{5}$$

where the last transformation is by definition [Knu73, sec. 1.2.6]. Substituting Eq. 5 in the probability conservation equation, gives

$$1 = \pi_0 + \pi_0 \sum_{s=1}^{n} \binom{\rho+s-1}{s} = \pi_0 \sum_{s=0}^{n} \binom{\rho+s-1}{s} \tag{6}$$

which yields a closed form solution for π_0 (and the other the steady state probabilities),

$$\pi_0 = 1 \bigg/ \binom{\rho+n}{n}$$

and for the success probability,

$$P_{suc} = 1 - \pi_n = \frac{n}{n+\rho} \tag{7}$$

Comparing this result with Eq. 4, yields that when $\theta = \mu$, i.e., when the penalty for capturing more than one link is maximal, a system where the reservation algorithm attempts to capture one link performs identically to a system where the algorithm attempts to capture all the links.

3.2 Reducing the Analysis Complexity Using Recurrence

For the general case, we can reduce the computation complexity of section 3 by using recursion. Our aim is to write the steady state probabilities of all the system states, $\pi_{t,s}$, as functions of $\pi_{0,s}$, $0 \le s \le n$. Then, we can write n equilibrium equations and together with the probability conservation equation we obtain $n+1$ linear equations that can be solved with complexity of $O(n^3)$. For clarity, we shall first demonstrate each step in the computation process for the case where $k = 2$, and then give the general solution. Table 2 presents the correlation between the expressions for $k = 2$ and general k.

$k = 2$	general k
(8)	(11)
(9)	(12)
(10)	(13)
(16)	(17) & (18)
(19)	(20)

Table 2: The correlation between expressions for $k = 2$ and general k.

Using the Markov chain illustrated in figures 2 and and the transition rates of Eq. 1 (illustrated in figure 3), we can write the following $n(n+1)/2 - 1$ equilibrium equations

$$\begin{aligned} q_{t,s,t,s}\pi_{t,s} &= q_{t-1,s-1,t,s}\pi_{t-1,s-1} + q_{t,s-1,t,s}\pi_{t,s-1} \\ &\quad + q_{t+1,s,t,s}\pi_{t+1,s} + q_{t,s+1,t,s}\pi_{t,s+1} \qquad (8) \\ &\qquad\qquad 2 \le t \le n-2,\ 0 \le s \le n-(t+1) \\ q_{0,s,0,s}\pi_{0,s} &= q_{0,s-1,0,s}\pi_{0,s-1} + q_{1,s,0,s}\pi_{1,s} \\ &\quad + q_{0,s+1,0,s}\pi_{0,s+1} \qquad 1 \le s \le n-1 \\ q_{0,0,0,0}\pi_{0,0} &= q_{1,0,0,0}\pi_{1,0} + q_{0,1,0,0}\pi_{0,1} \end{aligned}$$

where $q_{t,s,t,s}$, the transition rate out of state (t,s), is given by

$$q_{t,s,t,s} = t\theta + s\mu + \sum_{l=0}^{\min\{n-(t+s+1),1\}} b(l,t+s)\lambda \qquad (9)$$

Now, we can write the following recursion relations for $\pi_{t,s}$, $t > 0$:

$$\begin{aligned}
\pi_{1,0} &= (q_{0,0,0,0}\pi_{0,0} - q_{0,1,0,0}\pi_{0,1})/q_{1,0,0,0} \qquad (10)\\
\pi_{1,s} &= (q_{0,s,0,s}\pi_{0,s} - q_{0,s-1,0,s}\pi_{0,s-1} - q_{0,s+1,0,s}\pi_{0,s+1})\\
&\quad /q_{1,s,0,s} \qquad s = 1,2,\ldots,n-1\\
\pi_{t,s} &= (q_{t-1,s,t-1,s}\pi_{t-1,s} - q_{t-2,s-1,t-1,s}\pi_{t-2,s-1} -\\
&\quad q_{t-1,s-1,t-1,s}\pi_{t-1,s-1} - q_{t-1,s+1,t-1,s}\pi_{t-1,s+1})\\
&\quad /q_{t,s,t-1,s} \quad t = 2,3,\ldots,n \;\; s = 1,2,\ldots,n-t
\end{aligned}$$

For a general value of k, Eq. 8 takes the form

$$\begin{aligned}
q_{t,s,t,s}\pi_{t,s} &= q_{t+1,s,t,s}\pi_{t+1,s} + q_{t,s+1,t,s}\pi_{t,s+1} \qquad (11)\\
&+ \sum_{l=\max\{0,t-(k-1)\}}^{t} q_{l,s-1,t,s}\pi_{l,s-1}
\end{aligned}$$

where $q_{t,s,t,s}$, the transition rate out of state (t,s), is given by

$$q_{t,s,t,s} = t\theta + s\mu + \sum_{l=0}^{\min\{n-(t+s+1),k-1\}} b(l,t+s)\lambda \qquad (12)$$

and the recurrence takes the form

$$\begin{aligned}
\pi_{t,s} &= (q_{t-1,s,t-1,s}\pi_{t-1,s} - q_{t-1,s+1,t-1,s}\pi_{t-1,s+1} \qquad (13)\\
&- \sum_{l=\max\{0,t-k\}}^{t-1} q_{l,s-1,t-1,s}\pi_{l,s-1})/q_{t,s,t-1,s}\\
&\qquad t = 1,2,3,\ldots,n \;\; s = 1,2,\ldots,n-t
\end{aligned}$$

The above recurrence suggests that all $\pi_{t,s}$ can be written as functions of $\pi_{0,s}$, i.e.,

$$\pi_{t,s} = \sum_{l=0}^{n} C_{t,s}(l)\pi_{0,l}, \qquad (14)$$

It is easier to calculate the recurrence for the coefficients, $C_{t,s}(l)$, rather than directly for $\pi_{t,s}$. First, we calculate the coefficients of $\pi_{1,s}$ by

$$\begin{aligned}
C_{1,s}(s) &= q_{0,s,0,s}/q_{1,s,0,s} \quad s = 0,1,2,\ldots,n-1 \; (15)\\
C_{1,s}(s-1) &= -q_{0,s-1,0,s}/q_{1,s,0,s} \quad s = 1,2,\ldots,n-1\\
C_{1,s}(s+1) &= -q_{0,s+1,0,s}/q_{1,s,0,s} \quad s = 0,1,2,\ldots,n-1\\
C_{1,s}(l) &= 0 \quad |l-s| > 1
\end{aligned}$$

Next, we calculate the coefficients of $\pi_{t,s}$ for $t = 2,3,\ldots,n-1$. For $k = 2$ the recurrent calculation is done by

$$\begin{aligned}
C_{t,s}(m) &= (q_{t-1,s,t-1,s}C_{t-1,s}(m) \qquad (16)\\
&-q_{t-1,s+1,t-1,s}C_{t-1,s+1}(m)\\
&-q_{t-1,s-1,t-1,s}C_{t-1,s-1}(m)\\
&-q_{t-2,s-1,t-1,s}C_{t-2,s-1}(m))/q_{t,s,t-1,s}
\end{aligned}$$

Now, for a general value of k, the recurrent calculation of the coefficients $C_{t,s}(l)$, $t > 1$ takes the form:
For $t > k$

$$\begin{aligned}
C_{t,s}(m) &= (q_{t-1,s,t-1,s}C_{t-1,s}(m) \qquad (17)\\
&-q_{t-1,s+1,t-1,s}C_{t-1,s+1}(m)\\
&-\sum_{l=1}^{k} q_{t-l,s-1,t-1,s}C_{t-l,s-1}(m))/q_{t,s,t-1,s}
\end{aligned}$$

For $t \leq k$

$$\begin{aligned}
C_{t,s}(m) &= (q_{t-1,s,t-1,s}C_{t-1,s}(m) \qquad (18)\\
&-q_{t-1,s+1,t-1,s}C_{t-1,s+1}(m)\\
&-\sum_{l=0}^{t-1} q_{l,s-1,t-1,s}C_{l,s-1}(m))/q_{t,s,t-1,s}
\end{aligned}$$

The recurrence calculation requires $O(k \cdot n^3)$ operations ($1 \leq k \leq n$). $n+1$ equilibrium equations are not used to derive the recurrence, thus n of them can be used together with the probability conservation equation, in equation system 19, to achieve the following $n+1$ linear equation system, whose solution complexity is lower than $O(n^3)$.

$$\begin{aligned}
q_{t,n-t,t,n-t}\pi_{t,n-t} &= q_{t,n-(t+1),t,n-t}\pi_{t,n-(t+1)} + \qquad (19)\\
&\quad q_{t-1,n-(t+1),t,n-t}\pi_{t-1,n-(t+1)}\\
&\qquad 1 \leq t \leq n-1\\
q_{0,n,0,n}\pi_{0,n} &= q_{0,n-1,0,n}\pi_{0,n-1}\\
\sum_{(t,s)} \pi_{t,s} &= 1
\end{aligned}$$

For a general value of k, Eq. 19 takes the form:

$$\begin{aligned}
&q_{t,n-t,t,n-t}\pi_{t,n-t} = \qquad (20)\\
&\sum_{l=\max\{0,t-(k-1)\}}^{t} q_{l,n-(t+1),t,n-t}\pi_{l,n-(t+1)} \quad 1 \leq t \leq n-1
\end{aligned}$$

Using the recurrence on the coefficients we can write Eq. 20 as

$$\begin{aligned}
&\sum_{m=0}^{n} q_{t,n-t,t,n-t}C_{t,n-t}(m)\pi_{0,m} =\\
&\quad \sum_{m=0}^{n} \sum_{l=\max\{0,t-(k-1)\}}^{t} q_{l,n-(t+1),t,n-t}C_{l,n-(t+1)}(m)\pi_{0,m}\\
&\qquad 1 \leq t \leq n-1
\end{aligned}$$

and rewrite the probability conservation equation as

$$\sum_{t=0}^{n-1}\sum_{s=0}^{n-t}\sum_{m=0}^{n} C_{t,s}(m)\pi_{0,m} = 1$$

4 Consecutive Trials

In this section we analyze the performance of the system when only one route is examined at a time, but upon failure additional κ attempts are made to reserve resources. Note that by doing so, the arrival rate λ_e the system observes is higher than λ the arrival rate of requests from the outside. We make the standard assumption [RS90, sec. 3.1] that the time period between consecutive retries is exponentially distributed and that the combined arrival process of new incoming request and repeating requests is Poisson.

For simplicity, we assume that upon failure, the next route to be selected for reservation is selected randomly, and that all the n routes have the same probability to be selected regardless of the route previously checked. If n is large enough this assumption does not introduced a large error. Even for small value of n, checking the same route again, might be useful as it may be free after an exponential time passed from the previous attempt.

4.1 Infinite re-trials

We first analyze the case where re-trials are performed until success is achieved. If the arrival rate λ is larger than $n\mu$ this system in unstable, thus it is not a practical strategy and is only brought as a reference to what can be achieved in some conditions.

The effective arrival rate to such a system is given by

$$\lambda_e = \lambda \sum_{i=1}^{\infty} P_{suc} \cdot i \cdot (1 - P_{suc})^{i-1} = \lambda / P_{suc} \qquad (21)$$

substituting λ_e in equation 4 gives us the success probability per trial:

$$P_{suc} = 1 - \frac{\lambda}{n\mu} \qquad (22)$$

which is smaller than the success probability for $k = 1$ (Eq. 4) with one trial (as expected).

The system is stable when $\lambda_e < n\mu$. λ_e is given by substituting P_{suc} from equation 22 in equation 21:

$$\lambda_e = \lambda / P_{suc} = \frac{\lambda \mu n}{\mu n - \lambda} = \lambda + \frac{\lambda^2}{\mu n - \lambda}$$

Thus, stability is achieved for $\lambda < \frac{1}{2} n\mu$. Of course, if the system is stable the probability for a connection to eventually capture bandwidth is 1.

4.2 Two Trials

If the number of retrials is limited to one, the effective arrival rate is:

$$\lambda_e = \lambda + \lambda(1 - P_{suc}) = \lambda(2 - P_{suc})$$

substituting λ_e in equation 4 yields the equation

$$\rho P_{suc}^2 - (n + 2\rho) P_{suc} + n = 0$$

and its solution is given by

$$P_{suc} = 1 - \left(\sqrt{\left(\frac{n}{2\rho}\right)^2 + 1} - \frac{n}{2\rho} \right) \qquad (23)$$

The throughput is thus given by

$$\mathcal{T} = \lambda_e P_{suc} = \lambda(2 - P_{suc}) P_{suc}$$

4.3 Three Trials

If the number of retrial is limited to two the effective arrival rate is

$$\lambda_e = \lambda \left[1 + (1 - P_{suc}) + (1 - P_{suc})^2 \right] = \lambda(3 - 3P_{suc} + P_{suc}^2)$$

substituting λ_e in equation 4 yields the equation

$$\rho P_{suc}^3 - 3\rho P_{suc}^2 + (n + 3\rho) P_{suc} - n = 0$$

and its solution is given by

$$\begin{aligned} P_{suc} \quad = \quad & 1 - \frac{\frac{2}{3}^{\frac{1}{3}} n}{\sqrt{r} \left(-9\, r^{\frac{3}{2}} + \sqrt{3} \sqrt{4\, n^3 + 27\, r^3} \right)^{\frac{1}{3}}} \qquad (24) \\ & + \frac{\left(-9\, r^{\frac{3}{2}} + \sqrt{3} \sqrt{4\, n^3 + 27\, r^3} \right)^{\frac{1}{3}}}{18^{\frac{1}{3}} \sqrt{r}} \end{aligned}$$

The throughput is thus given by

$$\mathcal{T} = \lambda_e P_{suc} = \lambda(3 - 3P_{suc} + P_{suc}^2) P_{suc}$$

5 General Link Capacity

To reduce the complexity of the analysis, we assume here that the penalty for capturing more than a single route is maximal, i.e., $\mu = \theta$. We analyze a system with 3 routes ($n = 3$) and let m, the link capacity, change. A continuous-time Markov chain with $(m + 1)^n$ states can be used to model the system. Since the Markov chain for $n = 3$ is quite complicated to depict, we show in figure 6 the case for $n = 2$, $m = 3$, and $k = 2$, and in figure 7 the case for $n = 2$, $m = 3$, and $k = 1$. Each state is represented by the tuple $\langle t_1,\ \ldots,\ t_n \rangle$, where t_i represents the number of connections currently using route i.

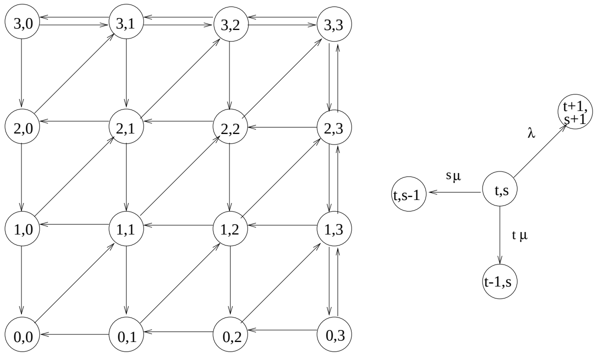

Figure 6: A Markov chain for a system with $n = 2$, $m = 3$, and $k = 2$

To write the infinitesimal transition rates between states we use the following notations. Let Γ be the tuple $\langle t_1,\ t_2,\ t_3 \rangle$,

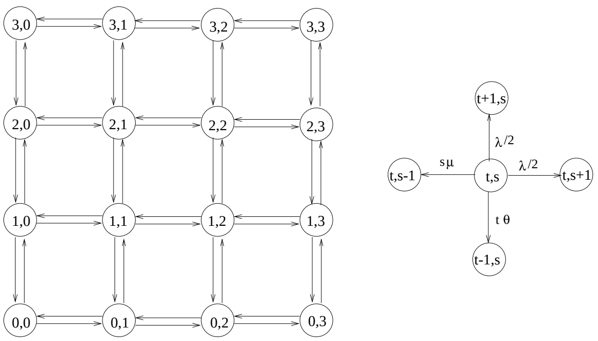

Figure 7: A Markov chain for a system with $n = 2$, $m = 3$, and $k = 1$

then Γ^+ represents a tuple where exactly one of the components is greater by one from the corresponding component of the tuple Γ. Γ^{++} represents a tuple where exactly two of the components are greater by one from the corresponding components of the tuple Γ. Γ^{+++} represents a tuple where all three components are one larger than the ones in Γ. We define Γ^-, Γ^{--}, and Γ^{---} similarly except that the components are smaller by one.

The infinitesimal transition rates between states for $k = 1$ are

$$\begin{aligned} q_{\Gamma,\Gamma_i^-} &= t_i\mu \\ q_{\Gamma,\Gamma^+} &= \lambda/3 \end{aligned}$$

The infinitesimal transition rates between states for $k = 2$ are

$$\begin{aligned} q_{\Gamma,\Gamma_i^-} &= t_i\mu \\ q_{\Gamma,\Gamma^{++}} &= \lambda/3 \\ q_{\Gamma,\Gamma^+} &= 2\lambda/3 \quad \text{if two components of } \Gamma \text{ are equal to } m \\ q_{\Gamma,\Gamma^+} &= \lambda/3 \quad \text{if one component of } \Gamma \text{ is equal to } m \end{aligned}$$

The infinitesimal transition rates between states for $k = 3$ are

$$\begin{aligned} q_{\Gamma,\Gamma_i^-} &= t_i\mu \\ q_{\Gamma,\Gamma^{+++}} &= \lambda \\ q_{\Gamma,\Gamma^+} &= \lambda \quad \text{if two components of } \Gamma \text{ are equal to } m \\ q_{\Gamma,\Gamma^{++}} &= \lambda \quad \text{if one component of } \Gamma \text{ is equal to } m \end{aligned}$$

For all cases $q_{\Gamma,\Gamma}$ is calculated by summing the negation of all the transition rates out of state Γ.

We solve for the steady state probabilities π_Γ as described in section 3. For the case when $k = 1$, P_{suc} is given by

$$\begin{aligned} P_{suc} = & \sum_{t_1,t_2,t_3<m} \pi_{\langle t_1,t_2,t_3\rangle} + \\ & \sum_{\text{for exactly one } i,\ t_i=m} \frac{2}{3}\pi_{\langle t_1,t_2,t_3\rangle} + \\ & \sum_{\text{for exactly two values of } i,\ t_i=m} \frac{1}{3}\pi_{\langle t_1,t_2,t_3\rangle} \end{aligned} \tag{25}$$

For the case when $k = 2$, P_{suc} is given by

$$\begin{aligned} P_{suc} = & \sum_{\text{for at most one } i,\ t_i=m} \pi_{\langle t_1,t_2,t_3\rangle} + \\ & \sum_{\text{for exactly two values of } i,\ t_i=m} \frac{2}{3}\pi_{\langle t_1,t_2,t_3\rangle} \end{aligned} \tag{26}$$

For the case when $k = 3$, P_{suc} is given by

$$P_{suc} = 1 - \pi_{\langle m,m,m\rangle} \tag{27}$$

6 Reservation Time Analysis

In this section we assume that the duration of a reservation request process along a single route is exponentially distributed with mean τ regardless of whether it succeeds or fails. This time includes the propagation delay and the queueing delay of the control messages, and the processing delay of the requests in the switches.

6.1 Multi-Path Reservation

When k multiple reservations are done in parallel, the time until the first one terminates is τ/k as it is a competition between k exponential processes. The success probability for each of the reservation processes is bounded below by P_{suc}/k where P_{suc} is the overall reservation success probability as computed in section 3.2. The expected time to successfully reserve a route is thus bounded below by

$$\begin{aligned} T_{suc} = \frac{1}{1-(1-P_{suc}/k)^k}\cdot & \left[\frac{\tau}{k}\frac{P_{suc}}{k} + \right. \\ & \frac{\tau}{k-1}\left(1-\frac{P_{suc}}{k}\right)\frac{P_{suc}}{k} + \cdots + \\ & \left. \tau\left(1-\frac{P_{suc}}{k}\right)^{k-1}\frac{P_{suc}}{k}\right] \end{aligned} \tag{28}$$

where P_{suc} is calculated by equation 2, and $1/(1-(1-P_{suc}/k)^k)$ is a normalization factor that ensures that $\sum_{i=0}^{k-1}(1-P_{suc}/k)^i \cdot (P_{suc}/k) = 1$.

6.2 Successive Trial Reservation

For successive reservation, we assume no delay between a connection rejection and the next trial, which yields

$$\begin{aligned} T_{suc} &= \frac{1}{1-(1-P_{suc})^\kappa}\sum_{i=1}^{\kappa} i\cdot\tau P_{suc}(1-P_{suc})^{i-1} \\ &= \frac{1-\kappa(1-P_{suc})^\kappa P_{suc}-(1-P_{suc})^\kappa}{P_{suc}(1-(1-P_{suc})^\kappa)}\tau \end{aligned} \tag{29}$$

where P_{suc} is calculated by equation 23 or 24.

7 Numerical Results

Throughout this section, we compare normalized throughput, which is the throughput divided by the number of routes, $\mathcal{T}/n$. This way the maximum throughput is always 1 regardless of the system size. Recall that $\rho \stackrel{\triangle}{=} \lambda/\mu$.

We start with case where each route can hold, at most, one connection, i.e., when $m = 1$. Figure 8 and tables 3 - 6 show the normalized throughput as a function of ρ for $\mu/\theta = 1$. If retries are not permitted, single path routing is always below multipath routing. The best throughput is achieved for $k \cong n/2$ (the bolded number in each row is the maximum throughput among the nonpersistent algorithms), but the differences are not major. When retries are permitted higher throughput is achieved, and the number of allowed reservation attempts increase the throughput increases, as expected. Maximal penalty ($\mu = \theta$) represents the case where short bursts are sent along best effort routes, possibly using on-the-fly reservation [BT92, Tur92].

When the penalty for using more than a single route decreases, the throughput achieved by multi-path algorithms increases as the overhead of over-reservation decreases. Figure 9 depicts the throughput when the penalty is 1/10. For this set of parameters, using multi-path routing that attempts to reserve two paths ($k = 2$) yields the same throughput as using single path routing with one retry. Multi-path routing with $k \geq 3$ achieves higher throughput than single path reservation with a single retry. The low penalty case represents long bursts or short term connections that use three-way reservation [BT92]. If several paths are captured the source selects only one to be used for transmission and releases the reserved resources from the rest of the routes.

For all the calculated parameters, $k = 1$ always achieves the lowest success probability when compared with other values of k, as can be clearly seen by the solid line that is always the lowest in the presented graphs. Note that the value $k = 1$ represents the classic case where reservation for a connection is attempted along a single route, while the values $k > 1$ represent cases where reservation is attempted along several routes.

Figure 10 shows the expected time to successfully reserve a route, T_{suc}, as a function of the arrival rate λ for $n = 9$. Recall that a single reservation attempt on a single route takes an average of τ time units. Since this average is not a function of the load, it translates to a horizontal line at Y=1 in the figure.

Successive trial reservation is shown to increase the expected reservation time by up to 25% when one retry is permitted and by 75% when two retries are permitted. Note the the maximum plotted load is around 1. Multipath reservation decreases the expected reservation time by more than 30% for $k = 2$, and by almost 50% for $k = 3$. Higher k values farther decrease the expected reservation time.

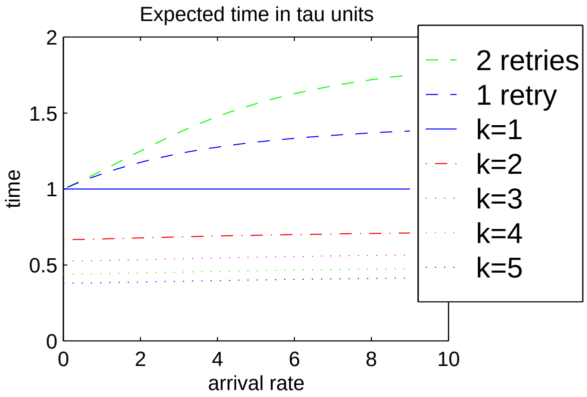

Figure 10: T_{suc} as a function of the load for $n = 9$ and $m = 1$.

The reason for the big increase in the expected reservation time when retries are used can be explained by looking at the success probability per attempt (figure 11). As the number of permitted retries increases the actual loads of requests arriving to the system, λ_e, increases and the success probability per trial decreases.

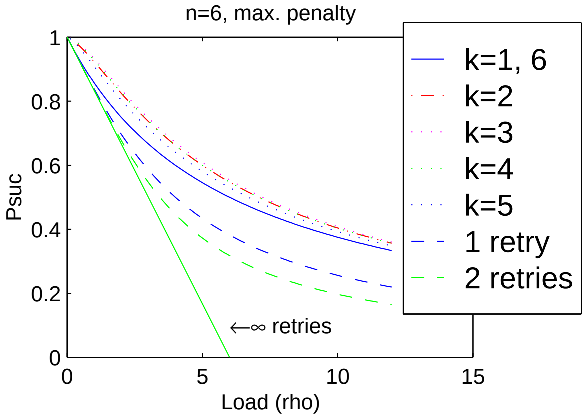

Figure 11: The success probability per trial, P_{suc}, as a function of the load for $n = 6$ and $m = 1$.

Next we check the effect of increasing the capacity of the paths. Figures 12 and 13 show the success probability for a system with three routes for two cases: $m = 1$ and $m = 2$. Figures 14, 15, and 16 show the success probability for a system with four routes for $m = 1$, 2, and, 3. Two phenomena can be observed from the figures. As the link capacity increases, the relative performance of multipath routing decreases. However, as the number of possible routes increases the relative performance of multi-path routing increases. This implies that the use of multi-path reservation is more attractive when resources are scarce and connectivity is high.

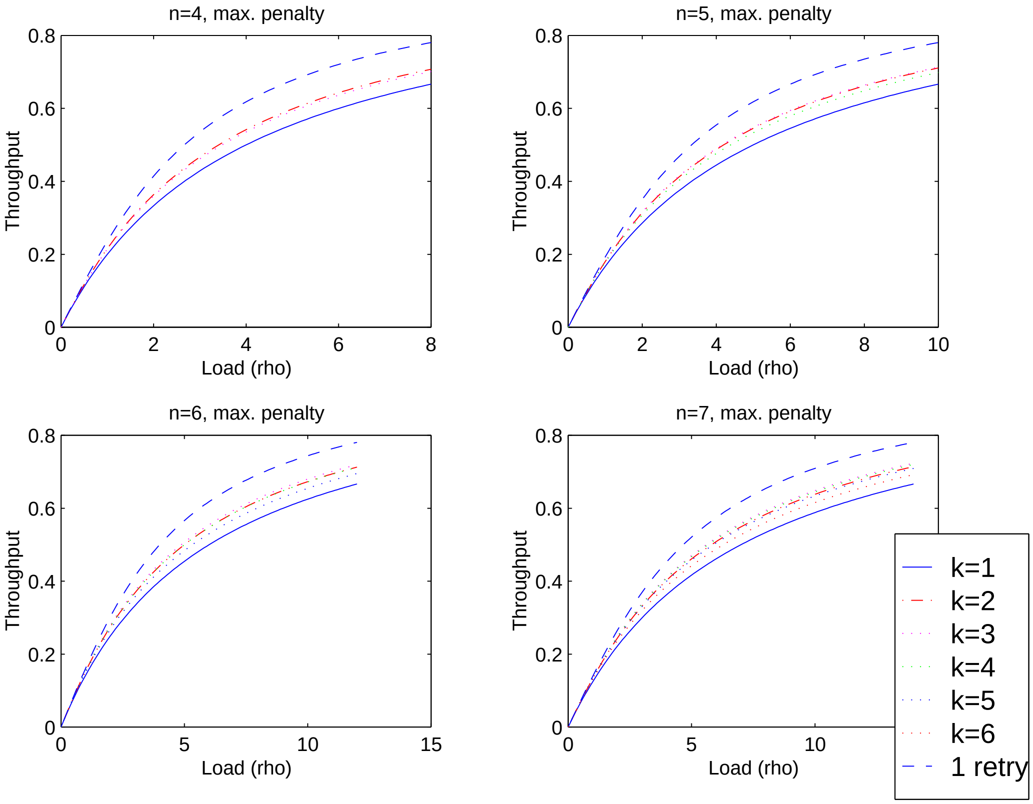

Figure 8: The throughput for the case where $n = 4, 5, 6, 7$, $m = 1$, and $\mu = \theta$.

ρ	multi-path routing ($\kappa = 0$)						multi-trial ($k = 1$)	
	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$\kappa = 2$	$\kappa = 3$
0.1	0.0163934	0.0165988	**0.0166255**	0.0166227	0.0165913	0.0163934	0.016662	0.0166666
1	0.142857	0.153846	**0.155515**	0.154545	0.151515	0.142857	0.162278	0.165906
10	0.625	0.673077	**0.68**	0.672269	0.654762	0.625	0.744031	0.803165

Table 3: The throughput for the case where $n = 6$, $m = 1$, and $\mu = \theta$.

8 On Multi-Path Reservation Implementation

Overall, the results above serve as a motivation to reserve routes in parallel. Even in the worst case, when the penalty for over-reservation is maximal, the throughput of multi-path reservation is comparable with that achieved by a persistent single path reservation attempt. However, the expected connection establishment time for multi-path routing is about half the one for persistent reservation even under medium load conditions. This makes multi-path reservation an attractive solution for applications that require fast set-up.

The analysis in this paper is restricted to disjoint routes. This makes the analysis in particular conservative because it maximizes the penalty for the excess reservation. In practice, as demonstrated below, when routes share links it reduces the amount of excess reservation, and when routes share nodes the time excess reservation is held may decrease significantly.

In the following we shortly describe some multi-path reservation algorithms and show that this algorithm family takes advantage of shared links and nodes, and for which the analysis represents a worst case performance. In particular, the implementation of early bandwidth release at joint nodes reduces the penalty for excess reservation, and the ability to have joint links increases the flexibility in choosing a good collection of routes.

The algorithms are based on a flooding algorithm that attempts to reserve bandwidth along several possible routes. Generally, searching from a scratch for a route between two nodes in the entire network is inefficient in terms of communication cost and set-up time. Thus, we assume that a topology-update algorithm informs the nodes about the (slow) changes in the network topology and about the cost of the links. When a node wishes to establish a connection, it searches for the best route in a subgraph of the network that contains links that lead to the destination

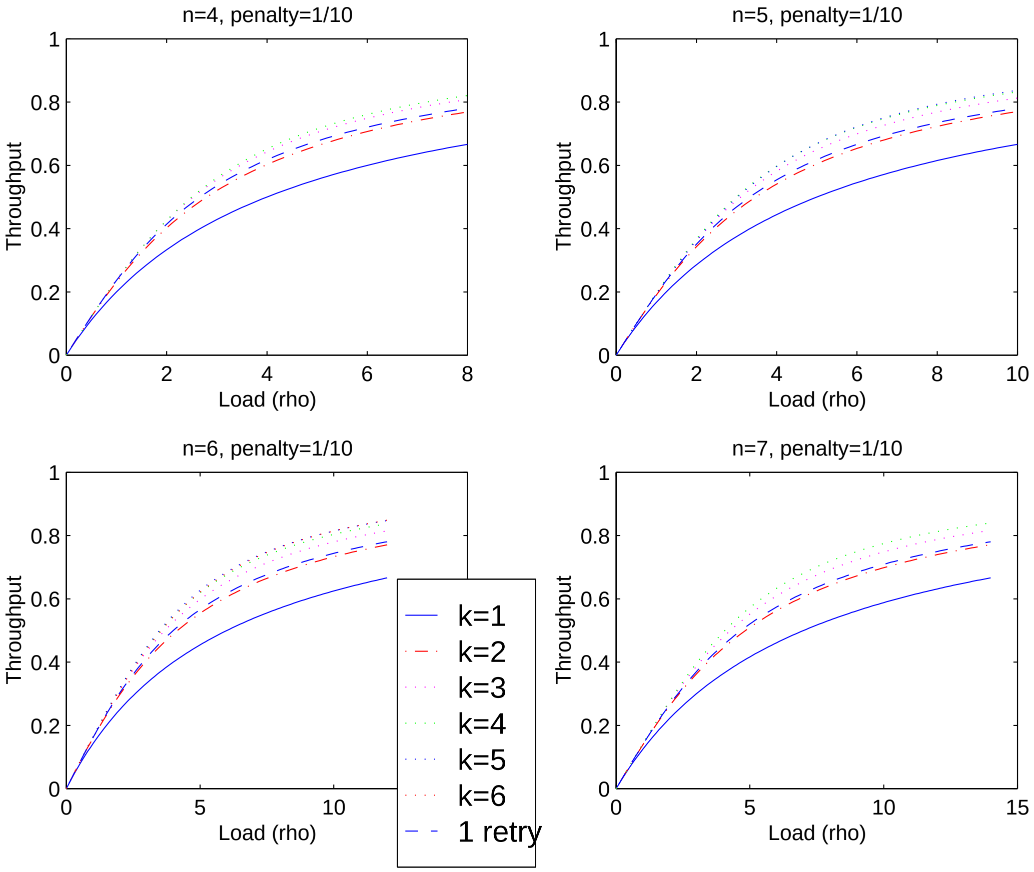

Figure 9: The throughput for the case where $n = 4, 5, 6, 7$, $m = 1$, and $\theta/\mu = 10$.

ρ	multi-path routing ($\kappa = 0$)							multi-trial ($k = 1$)	
	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$\kappa = 2$	$\kappa = 3$
0.1	.0140845	.0142434	.0142641	**.0142664**	.0142602	.0142364	.0140845	.0142828	.0142857
1	.125	.134172	**.135836**	.135644	.134409	.131868	.125	.140055	.142444
10	.588235	.638402	**.648404**	.644824	.633609	.615672	.588235	.709481	.771799

Table 4: The throughput for the case where $n = 7$, $m = 1$, and $\mu = \theta$.

and that have a “reasonable” cost. We call this restricted subgraph a *diroute*. The selection of the diroute can be made by the source node or, in a distributed manner by the nodes of the graph [CRS96, Sha96]. To avoid reservation of resources in the entire diroute until the best route is chosen, the algorithms release resources from segments of the diroute as soon as they learn that these segments are inferior to another segment where reservation was made. The implementation of this *early release* of bandwidth is possible since a node in the diroute that receives two or more reservation messages from different links, can locally select the best one, and can locally decide to release the bandwidth from the other incoming paths.

Three sub-families of algorithms are presented in [CRS97, Sha96]:

Fast algorithms where the reservation message travels to the destination as fast as possible, but the best possible route might not be the one selected.

Slow algorithms where the reservation message travels to the destination at the speed of the slowest path, but the selected path is guaranteed to be the best in the diroute and the message complexity is linear in the number of diroute links.

Superfast algorithms where the reservation message from the source to the destination and the positive acknowledgment from the destination to the source, both travel as fast as possible. Similar to the fast algorithms, the selected path might not be the best. The superfast algorithms use initial multicast connections that are gradually pruned to a unicast connection.

The main thrust of the algorithms is to reach the destination with a feasible path (using a flooding-like approach), altering the path if better alternatives are found in time, and releasing superfluous reserved bandwidth as soon as it

ρ	multi-path routing ($\kappa = 0$)								multi-trial
	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	$\kappa = 2$
0.1	.0123457	.0124718	.0124875	**.0124903**	.012489	.0124836	.0124656	.0123457	.012498
1	.111111	.118841	.120347	**.120455**	.119901	.118761	.116667	.111111	.123106
10	.555556	.606759	**.618846**	.618018	.610525	.59816	.580694	.555556	.677033

Table 5: The throughput for the case where $n = 8$, $m = 1$, and $\mu = \theta$.

ρ	multi-path routing ($\kappa = 0$)								multi-trial
	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	$\kappa = 2$
0.1	.010989	.0110914	.0111034	**.0111058**	.0111057	.011104	.0110997	.0110859	.0111097
1	.1	.106583	.107901	**.10812**	.107877	.107292	.106308	.104575	.109772
10	.526316	.577819	.591326	**.592441**	.587454	.578447	.565933	.549313	.646586

Table 6: The throughput for the case where $n = 9$, $m = 1$, and $\mu = \theta$.

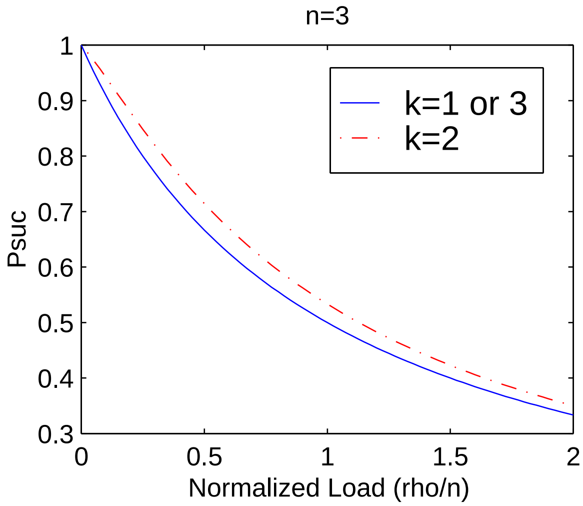

Figure 12: The success probability per trial, P_{suc}, as a function of the load for $n = 3$ and $m = 1$.

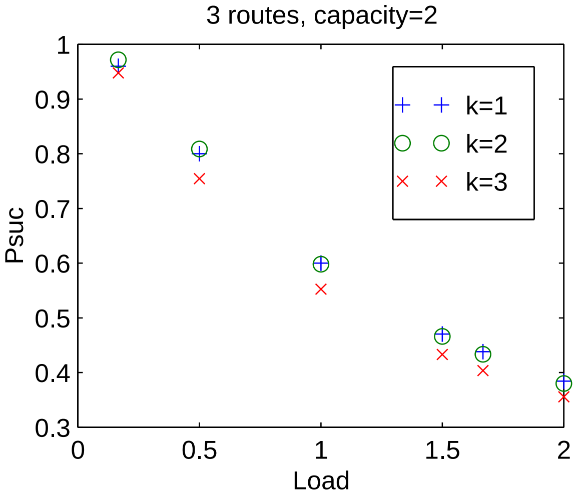

Figure 13: The success probability per trial, P_{suc}, as a function of the load for $n = 3$ and $m = 2$.

is identified.

The forward flooding is implemented by *Request* messages that carry the cost of the sub-route from the source to the node they arrive at. This cost is used by the intermediate node to select the best current in-coming sub-route if several exist, and to release the resources from the rest. Only a single reservation in made in a link even if it is shared by several sub-routes.

A destination node that receives, at least, one *Request* message starts the second stage of the algorithm by sending an *Accept* message. This message travels backwards along the reserved route and fixes its selection, i.e., a node that receives an *Accept* message cannot change its sub-route selection any more. In the super-fast algorithm there is an additional backward flooding message to signal the source that a route has been found and that data transmission can be started [Sha96].

These algorithm represent different trade-offs between the speed the search advances and the quality of the resulted route. All of them use the early release mechanism to release redundant resources (bandwidth) as soon as possible. We expect this work to trigger future development of multi-path reservation algorithms.

9 Concluding Remarks

This paper analyzes the performance of multi-path routing algorithms that reserve resources along the paths considered for routing. The analysis is based on the Poisson model which is no longer used for packet level analysis, but is still considered a good estimation of the burst (or session) level analysis presented in this paper. Also unlike packet generation where an ON-OFF model is considered as a common extension to the Poisson there is no general consensus on alternative bursty call generation processes or even if it is required. This is a very interesting open ques-

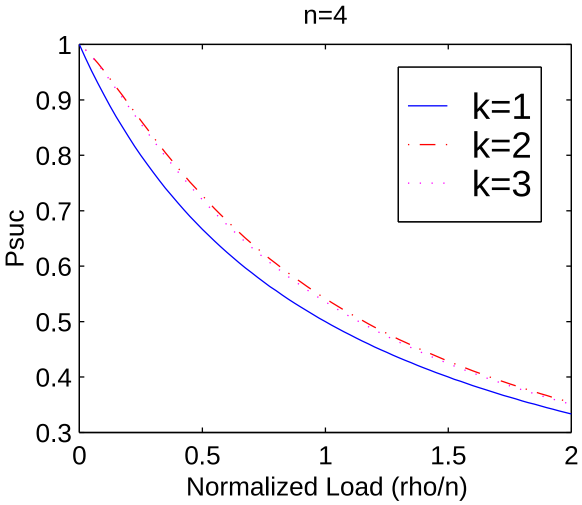

Figure 14: The success probability per trial, P_{suc}, as a function of the load for $n = 4$ and $m = 1$.

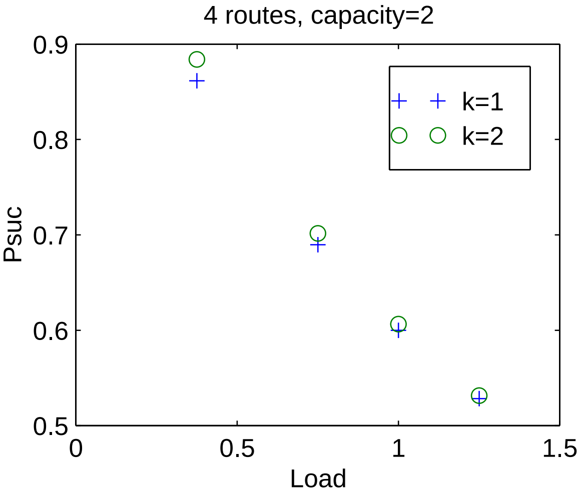

Figure 15: The success probability per trial, P_{suc}, as a function of the load for $n = 4$ and $m = 2$.

tion. Note that in this abstraction level, the independence assumption is also a good estimation.

The results presented here show that most of the gain due to multipath reservation is achieved when one or two paths are searched in addition to the traditional one path. This fact together with other practical consideration will most likely discourage implementors of multipath reservation algorithms from using more than two or three routes in parallel. Another consideration in using only a few paths in parallel is that real networks behavior may deviate from some of the assumption made in the analysis, and thus the optimal value of k, the number of paths to try, may change in practice. However, remember that analysis is conservative in two main points: the assumption that routes are disjoint, and the selection of maximal penalty for the excess reservation.

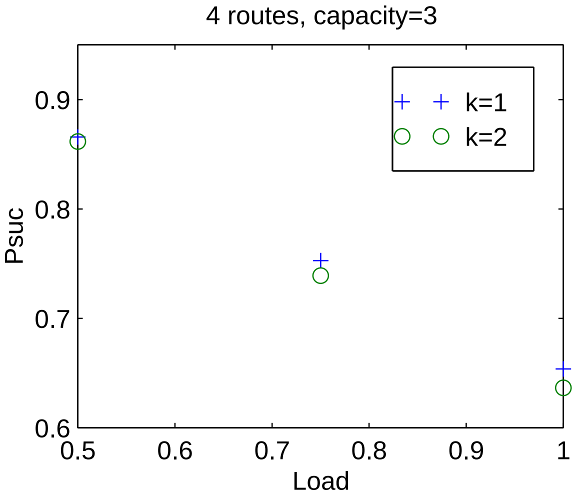

Figure 16: The success probability per trial, P_{suc}, as a function of the load for $n = 4$ and $m = 3$.

References

[BT92] Pierre E. Boyer and Didier P. Trachier. A reservation principle with applications to the ATM traffic control. *Computer Networks and ISDN Systems*, 24:321 – 334, 1992.

[CRS96] Israel Cidon, Raphael Rom, and Yuval Shavitt. Multi-path routing combined with resource reservation. Center of Communication Report #242 Faculty of Electrical Engineering, Technion – Israel Institute of Technology, Haifa, Israel, April 1998.

[CRS97] Israel Cidon, Raphael Rom, and Yuval Shavitt. Multi-path routing combined with resource reservation. In *IEEE INFOCOM'97*, pages 92 – 100. IEEE, April 1997.

[For96] ATM Forum. Private network network interface (PNNI) v1.0 specifications, June 1996.

[GKK95] Richard J. Gibbens, Frank P. Kelly, and Peter B. Key. Dynamic alternative routing. In Martha E. Steenstrup, editor, *Routing in Communications Networks*, pages 13 – 47. Prentice Hall, 1995.

[HKT95] Ren-Hung Hwang, James F. Kurose, and Don Towsley. On-call processing delay in high speed networks. *IEEE/ACM Transactions on Networking*, 3(6):628 – 639, December 1995.

[Kel91] F. P. Kelly. Loss networks. *The annals of applied probability*, 1(3):319 – 378, 1991.

[Knu73] Donald E. Knuth. *The Art of Computer Programming*, volume 1. Addison-Wesley, second edition, 1973.

[RS90] Raphael Rom and Moshe Sidi. *Multiple Access Protocols: Performance and Analysis*. Springer-Verlag, 1990.

[Sha96] Yuval Shavitt. *Burst Control in High-Speed Networks*. PhD thesis, Technion – Israel Institute of

Technology, Electrical Engineering Dept., Technion City, Haifa 32000, Israel, June 1996.

[Tur92] Jonathan S. Turner. Managing bandwidth in ATM networks with burtsy traffic. *IEEE Network*, 6(5):50 – 58, September 1992.

[ZDE+93] Lixia Zhang, Stephen Deering, Deborah Estrin, Scott Shenker, and Daniel Zappala. RSVP: a new resource ReSerVation protocol. *IEEE Network Magazine*, 7(5):8 – 18, September 1993.

[ZES97] Daniel Zappala, Deborah Estrin, and Scott Shenker. Alternate path routing and pinning for interdomain multicast routing. Technical Report 97-655, USC CS, June 1997.

Israel Cidon Israel Cidon received the B.Sc. (summa cum laude) and the D.Sc. degrees from the Technion - Israel Institute of Technology in 1980 and 1984, respectively, both in electrical engineering. From 1984 to 1985 he was with the faculty of the Electrical Engineering Department at the Technion. In 1985 he joined the IBM T.J. Watson Research Center, NY, where he was a Research Staff Member and the manager of the Network Architectures and Algorithms group involved in various broadband networking projects such as the Paris/Planet Gigabit networking testbeds, the Metaring/Orbit Gigabit LAN and the IBM BroadBand Networking architecture. In 1994 and 1995 he was with Sun Microsystems Labs in Mountains View, CA, as manager of High-Speed Networking founding various ATM projects including Openet - an open and efficient ATM network control platform. Since 1990 he is with the Department of Electrical Engineering at the Technion.

He was a founding editor for the IEEE/ACM Transactions on Networking. between 1992 and 1997. Previously he served as the Editor for Network Algorithms for the IEEE Transactions on Communications and as a guest editor for Algorithmica. In 1989 and 1993 he received the IBM Outstanding Innovation Award for his work on the PARIS high speed network and topology update algorithms respectively.

His research interests are in networks architecture, distributed network applications and algorithms and mobile networks.

Raphael Rom Raphael Rom received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion–Israel Institute of Technology, Haifa, Israel, and the Ph.D. degree in computer science from the University of Utah, Salt Lake City, UT. He was a Senior researcher on the research staff of SRI International in California, and subsequently joined the Faculty of Electrical Engineering in the Technion, Haifa, Israel. Since 1989 he is also with Sun Microsystems where he lead and managed the high speed networking group of SunLabs and is engaged in modeling and analysis of communication networks. In addition he held visiting positions in IBM T.J. Watson Research Center and Stanford University. Dr. Rom is the coauthor of the book "Multiple Access Protocols: Performance and Analysis." His areas of interest are algorithms for, and performance analysis of data communication and wireless networks and the design of general data communication systems.

Yuval Shavitt Yuval Shavitt received the B.Sc. in Computer Engineering (cum laude), M.Sc. in Electrical Engineering and D.Sc. from the Technion — Israel Institute of Technology, Haifa in 1986, 1992, and 1996, respectively.

From 1986 to 1991, he served in the Israel Defense Forces first as a system engineer and the last two years as a software engineering team leader. He spent the summer of 1992 as summer student at IBM T.J. Watson Research Center, NY. After graduation he spent a year as a Postdoctoral Fellow at the Department of Computer Science at Johns Hopkins University, Baltimore, MD. Since 1997 he is a Member of Technical Stuff in the Network & Service Management Research Department at Bell Labs, Lucent technologies, Holmdel, NJ. His recent research focuses on active networks and their use in network management, QoS routing and partitioning, and location problems.