Economides, A.A.& Silvester, J.A.: **Optimal routing in a network with unreliable links.** Proceedings IEEE 1988 Computer Networking Symposium, pp. 288-297, IEEE 1988.

OPTIMAL ROUTING IN A NETWORK WITH UNRELIABLE LINKS

Anastasios A. Economides and John A. Silvester

Communication Sciences Institute
Electrical Engineering-Systems Department
University of Southern California
Los Angeles, CA 90089-0272

Abstract

This paper investigates the routing of packets in a network in which the link error rates vary. A queueing network model that incorporates the effect of the link error rates is developed and is used to find optimal routing assignments for fixed error rates. Single and multiple path dynamic routing algorithms, that minimize the average packet delay or the failure probability of packet transmission are proposed. In case the network state is not exactly known, stochastic learning automata are proposed to drive the routing process.

1. INTRODUCTION

In studies of network optimization, performance and reliability have been extensively discussed. Network performance evaluates how well the network performs, given it is functioning correctly. Network reliability analyzes the probability that the network has not failed by time t, given it was fully operational at time 0. These two fundamental issues in network design are strongly dependent. However only recently, have they been considered together, in a composite measure called performability [7].

Network components, as in any physical system, are subject to failures. Extensive literature exists on investigating the network connectivity among two or more nodes under network component failures, for example [3]. However little work exists on how to design protocols that make use of reliability information.

Networks provide the medium for information transmission from one site to another. A primary factor of network performance is route selection. Not suprisingly, much work exists on this very important problem, for example [1]. Since a network is a dynamic environment, i.e. its state changes over time, optimal route selection should be based on the current network state which is called dynamic or adaptive routing.

Dynamic programming has been used for the dynamic routing problem in a simple unreliable manufacturing system. In [10], optimallity conditions for two machines where only one can fail are derived. They conclude that closed form solution would be very hard to obtain for large systems. Also in [4,12], switching policies for an unreliable machine that routes external input to either of two other machines are derived. In these papers, failure corresponds to total loss of use of the component.

In this paper, we investigate the problem of routing packets when there are many unreliable outgoing links to the same destination (Fig. 1). The analysis can be easily extended to a general network with unreliable links. A queueing network model that considers the link error rate is developed, allowing the optimal traffic assignments for a static network to be found. For the non-static case, single and multiple path dynamic routing algorithms that tend to minimize the average packet delay or the packet failure probability using stochastic learning automata are proposed.

The paper is organized as follows : in section 2, we introduce a queueing model that incorporates packet failure probabilities. In section 3, optimal single or multiple path routing assignments are found for the case of known link error rates. In section 4, we use stochastic learning automata for single and multiple path dynamic routing that minimize the average packet delay or the failure probability of packet transmission. In section 5, we discuss how the link state may be estimated. Finally, in section 6, we draw some conclusions.

2. QUEUEING NETWORK MODEL

In this section, we develop a queueing network model that incorporates the effect of the link error rates on the average packet delay. The link error rate is modeled as a feedback branch in the queueing model, since when a packet transmission fails it is retransmitted (Fig. 2). Consider a network node with one incoming and L outgoing links to the same destination (Fig. 1) and Poisson packet arrivals with exponentially distributed service requirements. We are interesting in optimally routing the packets over these links. We make the following definitions about network traffic :

$1/\mu$: mean packet duration.

C_i : link i transmission capacity, i=1,...,L.

λ : packet arrival rate.

$e_i(t)$: packet error rate on link i at time t, $e_i(t) \in [0,1]$.

$P_i(t)$: routing probability to link i at time t, $P_i(t) \in [0,1]$, $\sum_{\forall i} P_i(t) = 1$, and $P_i^*(t)$ = optimum $P_i(t)$ with respect to some objective function.

$\bar{T}_i(t)$: steady state average packet delay on link i at time t.

Define the adjusted normalized capacity

$$\mathcal{C}_i(t) \triangleq [1 - e_i(t)] * \mu * C_i.$$

Thus $\mathcal{C}_i(t)$ is the link capacity in error free packets per second at time t.

Modeling each link i as an M/M/1 queue, assuming immediate knowledge of failure, we find

$$\bar{T}_i(t) = \frac{1}{\mathcal{C}_i(t) - \lambda * P_i(t)}$$

where of course : $\lambda * P_i(t) \leq \mathcal{C}_i(t)$.

3. OPTIMAL STATIC ROUTING

In this section, we develop optimal link assignments for a static network state, i.e. in which the link states, $e_i(t)$ and the routing probabilities, $P_i(t)$ are fixed and known, i.e. we can use $\lambda * P_i$ and e_i as the packet arrival rate and packet error rate on link i.

3.1 SINGLE PATH ROUTING

Here, we are constrainted to use a single path, it is assumed that there is (at least) one link that has sufficient capacity to handle the traffic.

i) MINIMIZE THE PACKET FAILURE PROBABILITY OF A SPECIFIC USER

In order to minimize the packet failure probability of a user, we should send each one of its packets over the minimum error rate link. A user example is the network operating system that sends control packets to the network nodes.

Algorithm#1 :

Send to link i, where $e_i = \min_{\forall j: \mathcal{C}_j \geq \lambda} \{e_j\}$

ii) MINIMIZE THE PACKET DELAY OF A SPECIFIC USER

In order to minimize the packet delay of a user, we should send its packets over the minimum packet delay link.

Algorithm#2 :

Calculate $\bar{T}_j$, *at* $P_j = 1 \quad \forall j$

Send to link i, where $\bar{T}_i = \min_{\forall j: \mathcal{C}_j \geq \lambda} \{\bar{T}_j\}$

3.2 MULTIPLE PATH ROUTING

Here, we want to minimize a global objective function, such as the packet failure probability or the overall network average packet delay.

i) MINIMIZE THE PACKET FAILURE PROBABILITY

Order the links in order of increasing error rates, i.e. $e_1 \leq e_2 \leq ... \leq e_L$ and let an acceptable loading factor be $0 < \rho < 1$. The following trivial algorithm sends the traffic to the most reliable links.

Algorithm#3 :

If $0 \leq \lambda < \rho * \mathcal{C}_1$, *then send the traffic to link 1.*

If $\rho * \mathcal{C}_1 \leq \lambda < \rho * (\mathcal{C}_1 + \mathcal{C}_2)$, *then send* $\rho * \mathcal{C}_1$ *of the traffic to link 1, and the rest* $\lambda - \rho * \mathcal{C}_1$ *of the traffic to link 2.*

...

If $\rho * (\mathcal{C}_1 + \mathcal{C}_2 + ... + \mathcal{C}_{L-1}) \leq \lambda < \rho * (\mathcal{C}_1 + \mathcal{C}_2 + ... + \mathcal{C}_L)$, *then send* $\rho * \mathcal{C}_1$ *of the traffic to link 1,* $\rho * \mathcal{C}_2$ *of the traffic to link 2,..., and the rest* $\lambda - \rho * (\mathcal{C}_1 + \mathcal{C}_2 + ... + \mathcal{C}_{L-1})$ *of the traffic to link L.*

If $\rho * (\mathcal{C}_1 + \mathcal{C}_2 + ... + \mathcal{C}_L) \leq \lambda$, *then send* $\rho * \mathcal{C}_1$ *of the traffic to link 1,* $\rho * \mathcal{C}_2$ *of the traffic to link 2,...,* $\rho * \mathcal{C}_L$ *of the traffic to link L and disregard the rest of the traffic.*

These flows can be achieved by a probabilistic per packet decision or round robin sceduling.

ii) MINIMIZE THE OVERALL NETWORK AVERAGE PACKET DELAY

The overall network average packet delay is

$$\bar{T} = \sum_{i=1}^{L} P_i * \bar{T}_i$$

Order the links in order of decreasing (adjusted) capacity, i.e. $\mathcal{C}_1 \geq \mathcal{C}_2 \geq ... \geq \mathcal{C}_L$.

Using an approach similar to [1], we have several cases to consider.

Case 1 : Single Path Routing

When one of the links has very large capacity, $\mathcal{C}_1$, and the traffic is light, all the traffic will be routed there. This occurs when :

$$\frac{\partial(P_1 * \bar{T}_1)}{\partial P_1} |_{P_1^*=1} \leq \frac{\partial(P_l * \bar{T}_l)}{\partial P_l} |_{P_l^*=0} \quad \forall l \neq 1.$$

which is true when

$$\frac{C_1}{\{C_1 - \lambda\}^2} \leq \frac{C_l}{\{C_l\}^2} \quad \forall l \neq 1$$

Rewriting, we have

$$\sqrt{C_1 * C_l} \leq C_1 - \lambda \quad \forall l \neq 1$$

finally

$$0 \leq \lambda \leq \sqrt{C_1} * \{\sqrt{C_1} - \sqrt{C_l}\} \quad \forall l \neq 1$$

Case 2 : Two Path Routing

In this case, 2 links are selected and the marginal costs on these two links will be balanced, i.e.

$$\frac{\partial(P_1 * \bar{T}_1)}{\partial P_1} |_{P_1^*} = \frac{\partial(P_2 * \bar{T}_2)}{\partial P_2} |_{P_2^*} \leq \frac{\partial(P_l * \bar{T}_l)}{\partial P_l} |_{P_l^*=0}$$

$\forall l \neq 1, 2.$

This will be true when

$$\frac{C_1}{\{C_1 - \lambda * P_1^*\}^2} = \frac{C_2}{\{C_2 - \lambda * P_2^*\}^2}$$

After some manipulation, we find

$$\begin{aligned} P_1^* &= \frac{\sqrt{C_1}}{\sqrt{C_1} + \sqrt{C_2}} + \\ &+ \frac{\sqrt{C_1 * C_2}}{\lambda * \{\sqrt{C_1} + \sqrt{C_2}\}} * \{\sqrt{C_1} - \sqrt{C_2}\} \\ P_2^* &= 1 - P_1^* \end{aligned}$$

The other links must satisfy a relationship similar to case 1, i.e.

$$0 \leq \lambda \leq \sqrt{C_1} * \{\sqrt{C_1} - \sqrt{C_l}\} + \sqrt{C_2} * \{\sqrt{C_2} - \sqrt{C_l}\}$$

$\forall l \neq 1, 2.$

Case 3 corresponds when three paths are used, and so on until Case L.

Case L : Full Multi-Path Routing

All paths are used and all marginal costs will be identical, i.e.

$$\frac{\partial(P_i * \bar{T}_i)}{\partial P_i} |_{P_i^*} = \frac{\partial(P_j * \bar{T}_j)}{\partial P_j} |_{P_j^*} \quad \forall i, j.$$

This will be true when

$$\frac{C_i}{\{C_i - \lambda * P_i^*\}^2} = \frac{C_j}{\{C_j - \lambda * P_j^*\}^2} \quad \forall i, j$$

The optimal P_i^* are thus given by :

$$\frac{P_i^*}{\sqrt{C_i}} = \frac{1}{\sum_{\forall l} \sqrt{C_l}} + \frac{\sqrt{C_i}}{\lambda} - \frac{\sum_{\forall l} C_l}{\lambda * \sum_{\forall l} \sqrt{C_l}} \quad \forall i$$

or

$$P_i^* = \frac{\sqrt{C_i}}{\sum_{\forall l} \sqrt{C_l}} + \frac{\sum_{\forall l} \sqrt{C_i * C_l}}{\lambda * \sum_{\forall l} \sqrt{C_l}} * \{\sqrt{C_i} - \sqrt{C_l}\}\}$$

where of course

$$0 \leq \lambda * P_i^* < C_i$$

The following algorithm will optimally assign the traffic on the links.

Define

$$\alpha_k \triangleq \sum_{l=1}^{k} \sqrt{C_l} * \{\sqrt{C_l} - \sqrt{C_{k+1}}\}$$

and the total capacity for the k best links

$$\mathbf{C}_k \triangleq \sum_{l=1}^{k} C_l$$

Algorithm#4 :

If $0 \leq \lambda \leq \alpha_1$, then send all the traffic to link 1.

...

If $\alpha_{k-1} < \lambda \leq \alpha_k$, then send

$$C_i - \frac{\sqrt{C_i} * (\mathbf{C}_k - \lambda)}{\sum_{l=1}^{k} \sqrt{C_l}}$$

of the traffic to link i, $1 \leq i \leq k$.

...

If $\alpha_{L-1} < \lambda < \sum_{l=1}^{L} C_l$, then send

$$C_i - \frac{\sqrt{C_i} * (\mathbf{C}_L - \lambda)}{\sum_{l=1}^{L} \sqrt{C_l}}$$

of the traffic to link i, $1 \leq i \leq L$.

In Fig. 3, we consider optimal routing for 2 links with $\mu = 1, C_1 = C_2 = 2$, and $\lambda = 1$. We plot the routing probability to link 1 versus the link 1 error rate, for different values of link 2 error rate. When the link 1 error rate increases, the routing probability to link 1 decreases. The curve for zero errors on link 2 ($e_2 = 0$) shows that for error rates on link 1 (e_1) higher than 0.75 all traffic is sent to link 2. The high error rate curve ($e_2 = 0.9$) stops at $e_1 = 0.6$ since the system can no longer accomodate the traffic.

In Fig. 4, we consider optimal routing for 2 links with $\mu = 1, C_2 = 2$, equal error rates $e_1 = e_2$, and $\lambda = 1$. We plot the routing probability to link 1 versus the link 1 capacity for different values of the error rate. When the link 1 capacity increases, the routing probability to link 1 increases. For low link capacities and high error rates the links cannot accomadate the offered traffic. For high link 1 capacity all the traffic is serviced by link 1.

Next, we consider routing to 2 links with $\mu = 1, C_1 = C_2 = 2$,and $e_2 = 0$. In Fig. 5 $\lambda = 1$, while in Fig. 6 $\lambda = 2$. We plot the average packet delay versus the link 1 error rate for the case where the error rates are ignored, T, i.e. half of the traffic is sent to each link and compare to the optimal assignments, T^*. We see tremendous improvement for high error rates.

Finally, in Fig. 7, we compare the average packet delays T and T^* for different arrival rates, $\mu = 1, C_1 = C_2 = 2$,and $e_2 = 0$. We see again tremendous improvement for high error rates.

Extensions of the queueing model can be done by considering M/G/1 queues. It can be shown [5, Problem 5.22] that the Laplace transform of the total service time in a M/G/1 queue with feedback is

$$\mathcal{B}_i^*(s) = \frac{(1-e_i)*B_i^*(s)}{1-e_i*B_i^*(s)}$$

where $B_i^*(s)$ is the Laplace transform of the service time in the M/G/1 queue without feedback. In our case, the $\mathcal{B}_i^*(s)$ transform corresponds to the total packet service time including retransmissions on link i. The $B_i^*(s)$ transform corresponds to the packet service time on a perfectly reliable link i (without retransmissions). Then the z-transform of the number of packets on link i is

$$Q_i(z) = \left(1-\frac{\lambda*\bar{X}_i}{1-e_i}\right)*\frac{(1-e_i)*(1-z)*B_i^*(\lambda*(1-z))}{(1-e_i+e_i*z)*B_i^*(\lambda*(1-z))-z}$$

From this we can find the average number of packets on link i and from that the average packet delay on link i

$$\bar{T}_i = \frac{2*\bar{X}_i*(1-\lambda*P_i*\bar{X}_i)+\lambda*P_i*\bar{X_i^2}}{2*(1-e_i-\lambda*P_i*\bar{X}_i)}$$

where $\bar{X}_i$ is the mean packet service time on a perfectly reliable link i (without retransmissions).

It is also straightforward to consider propagation delays across the links $\tau_{prop,i}$, and processing delays in the nodes τ_{proc} [6].

One numerical approach to finding P_i^* is using learning automata theory [2, 8, 9, 11]. A learning automaton works as follows. Whenever a performing action results in a favorable outcome, the probability of performing the same action is increased and the probabilities of the other actions are decreased. When it results in an unfavorable outcome, the probability of performing the same action is decreased and those of the other actions are increased. For our case, the learning automaton increases the routing probability of the minimum marginal delay link and decreases those of all other links at every step. It tends to equalize the marginal delays of the links. Define $P_i(k)$ to be the probability for link i at step k. Then the following iteration can be used.

Suppose link i was selected at step k, with probability $P_i(k)$.

Set

$$D_j(k) = \frac{\partial(P_j*\bar{T}_j)}{\partial P_j}\Big|_{P_j=P_j(k)} \quad \forall j$$

If $D_i(k) = \min_{\forall j}\{D_j(k)\}$ *then*

$$P_i(k+1) = P_i(k) + \alpha*[1-P_i(k)]$$

$$P_j(k+1) = P_j(k) - \alpha*P_j(k) \qquad \forall j \neq i$$

else

$$P_i(k+1) = P_i(k) - \beta*P_i(k)$$

$$P_j(k+1) = P_j(k) + \beta*\left[\frac{1}{L-1} - P_j(k)\right] \qquad \forall j \neq i.$$

where $0 < \beta \ll \alpha < 1$.

where $\bar{T}_j$ is any of the delay measures discussed above.

This approach to finding P_i^* works even in the case where the traffic processes do not follow any nice distribution. Also, we can use as D_j, the real measurement of the increase of the portion of the overall network delay corresponding to link j due to the addition of new traffic on this link.

4. DYNAMIC ROUTING

We now consider the case where the error rates are time dependent, i.e. $e_i = e_i(t)$. In section 5, we discuss how to estimate these error rates, for now we assume the information is available. Thus we have $\hat{e}_i(t)$ as the current estimate for $e_i(t)$.

If these time dependent error rate estimates are accurate at every time instant, then we can immediately use algorithms 1, 2, 3, and 4.

On the other hand, if we have some uncertainty about the error rate estimates, we do not want to overreact. A way to smooth wrong routing decisions due to error rate estimate inaccuracies is by randomization. For the multiple path dynamic routing, this is already done by the optimal traffic assignment. However for the single path dynamic routing the optimal decisions were deterministic. Thus we introduce learning automata algorithms to learn the optimum routing decisions These learning algorithms route the packets probabilistically over the alternate links and update these routing probabilities according to the outcome of their actions [2, 9, 11].

i) SELECT A LINK PROBABILISTICALLY w.r.t FAILURE PROBABILITIES

Let the time at which the k^{th} packet arrives be t_k. Routing decisions (i.e., which link to use) are made at update instants which for now we consider to be the same as t_k.

In order to increase the probability that the traffic will be correctly transmitted, the most reliable link is selected. This can be done either directly (choose probabilistically among the link reliabilities) or through the routing probabilities that will be updated by a learning automaton scheme. At time t_k, a link is selected for packet transmission. Whenever a packet is transmitted successfully, then the probability to choose the same link is increased, while the routing probabilities of the other links are decreased. Whenever a packet fails transmission, then the routing probability of the selected link is decreased, while the routing probabilities of the other links are increased.

Algorithm#5 :

Suppose link i was selected at time t_k.

If *successful packet transmission, then*

$$P_i(t_{k+1}) = P_i(t_k) + \alpha * [1 - P_i(t_k)]$$
$$P_j(t_{k+1}) = P_j(t_k) - \alpha * P_j(t_k) \quad \forall j \neq i$$

else

$$P_i(t_{k+1}) = P_i(t_k) - \beta * P_i(t_k)$$
$$P_j(t_{k+1}) = P_j(t_k) + \beta * [\frac{1}{L-1} - P_j(t_k)] \quad \forall j \neq i$$

$$0 < \beta \ll \alpha < 1$$

Another way to update the routing probabilities with less overhead, (but also less accuracy) is to update less frequently, for example at times τ_n (Fig. 8). We assume that link selection (based on the $P_i(t_k)$'s) is made at the update points. Knowing that $n_i(\tau_n)$ packets were successfully transmitted and $u_i(\tau_n)$ failed during $[\tau_n, \tau_{n+1})$, then we must increase the routing probability of the selected link $n_i(\tau_n)$ times and decrease it $u_i(\tau_n)$ times. In a similar way we must decrease and increase the routing probabilities of the other links.

Since we do not want to keep track of the exact sequence of occurence of the packet failures and successes, we assume such sequences. There are several ways to accomplish this, for example :

i) Increase P_i in $n_i(\tau_n)$ updates, then decrease it in $u_i(\tau_n)$ updates.

ii) Let $n_i(\tau_n) \leq u_i(\tau_n)$. Increase and decrease P_i in $n_i(\tau_n)$ updates, then decrease it in $u_i(\tau_n) - n_i(\tau_n)$ updates.

iii) Let $n_i(\tau_n) > u_i(\tau_n)$. Increase and decrease P_i in $u_i(\tau_n)$ updates, then increase it in $n_i(\tau_n) - u_i(\tau_n)$ updates.

iv) Decrease P_i in $u_i(\tau_n)$ updates, then increase it in $n_i(\tau_n)$ updates.

v) Let $n_i(\tau_n) \leq u_i(\tau_n)$. Decrease and increase P_i in $n_i(\tau_n)$ updates, then decrease it in $u_i(\tau_n) - n_i(\tau_n)$ updates.

vi) Let $n_i(\tau_n) > u_i(\tau_n)$. Decrease and increase P_i in $u_i(\tau_n)$ updates, then increase it in $n_i(\tau_n) - u_i(\tau_n)$ updates.

By a same approach as in [2], we can solve these recurrence equations and have $P_i(\tau_{n+1}) = Function\ (P_i(\tau_n), n_i(\tau_n), u_i(\tau_n))$. Thus instead of updating $P_i(\tau_n)$ at every packet transmission success or failure, we update at the times τ_n.

Another approach is to increase the probability of the link, if it had the best error rate performance. A learning automaton increases the routing probability of the selected link if it had the minimum error rate and decreases the routing probabilities of the other links. Otherwise it decreases the routing probability of the selected link and increases the routing probabilities of the other links.

Algorithm#6 :

Suppose link i was selected at time t_k*:*

If $\hat{e}_i(t_k) = \min_{\forall j}\{\hat{e}_j(t_k)\}$, *then*

$$P_i(t_{k+1}) = P_i(t_k) + \alpha * [1 - P_i(t_k)]$$
$$P_j(t_{k+1}) = P_j(t_k) - \alpha * P_j(t_k) \quad \forall j \neq i$$

else

$$P_i(t_{k+1}) = P_i(t_k) - \beta * P_i(t_k)$$
$$P_j(t_{k+1}) = P_j(t_k) + \beta * [\frac{1}{L-1} - P_j(t_k)] \quad \forall j \neq i$$

$$0 < \beta \ll \alpha < 1$$

ii) SELECT A LINK PROBABILISTICALLY w.r.t. MARGINAL PACKET DELAY

Here, we want to route incoming traffic in such a way as to minimize the average packet delay, $\bar{T}$, in the network.

$$\bar{T} = \sum_{\forall i} P_i * \bar{T}_i$$

In the simplest case this will happen when

$$\frac{\partial(P_i * \bar{T}_i)}{\partial P_i} = \frac{\partial(P_j * \bar{T}_j)}{\partial P_j} \quad \forall i, j$$

We propose a dynamic routing algorithm based on learning automata theory, that converges to optimal routing. Whenever a link is selected and it gives the smallest marginal delay, then the probability of selecting that link again is increased. If it is not the best link, then the probability of selecting it again is decreased. The algorithm tends to balance the first derivative packet delays on the links. We use a step size that depends on the environment state. The more unbalanced links the faster the convergence, while as the links tend to be balanced, the step size decreases to zero. The exponential function is a good candidate to achieve this. It has rapid derivative and as it tends to "steady state" it slows.

Algorithm#7 :

Suppose link i was selected at t_k *with probability* $P_i(t_k)$.

Set

$$D_j(t_k) = \frac{\partial(P_j(t_k) * \bar{T}_j(t_k))}{\partial P_j(t_k)}$$

$$D_{min}(t_k) = \min_{\forall j}\{D_j(t_k)\}$$

$$D_{Max}(t_k) = \max_{\forall j}\{D_j(t_k)\}$$

If $D_i(t_k) = \min_{\forall j} \{D_j(t_k)\}$ *then*

$$P_i(t_{k+1}) = P_i(t_k) + [1 - e^{\alpha*(D_i(t_k) - D_{Max}(t_k))}] * *[1 - P_i(t_k)]$$

$$P_j(t_{k+1}) = P_j(t_k) - [1 - e^{\alpha*(D_i(t_k) - D_{Max}(t_k))}] * *P_j(t_k) \quad \forall j \neq i$$

else

$$P_i(t_{k+1}) = P_i(t_k) - [1 - e^{\beta*(D_{min}(t_k) - D_i(t_k))}] * *P_i(t_k)$$

$$P_j(t_{k+1}) = P_j(t_k) + [1 - e^{\beta*(D_{min}(t_k) - D_i(t_k))}] * *[\frac{1}{L-1} - P_j(t_k)] \quad \forall j \neq i.$$

where α and β are scaling factors.

We could also consider other damping factors.

5. ERROR RATE ESTIMATES

In this section, we are concerned with how to estimate the time varying error rates.

5.1 DIRECT MEASUREMENTS

When nothing is known about the behavior of the link error rate, then the simplest way to estimate it is by counting the number of packet failures and successes. Let

$u_i(t)$: number of unsuccessful packets among the last M packets, that were sent through link i prior to time t.

i) Then an estimate of the packet failure probability on link i is

$$\hat{e}_i(t) = \frac{u_i(t)}{M}$$

ii) Another estimate is the exponential smoothing

$$\hat{e}_i(1) = \frac{u_i(1)}{M}$$

$$\hat{e}_i(t_n) = a * \hat{e}_i(t_{n-1}) + (1-a) * \frac{u_i(t_n)}{M} \quad 0 < a < 1$$

5.2 RELIABILITY MARKOV MODEL

There are cases where a link can be considered to be in two states of operation with a different error rate in each state. An example is a satellite communication link under normal and hazardous conditions (eg. rain). In such a case we can model each link as a two state Markov failure/repair model (Fig. 9). Depending on the link state, traffic will suffer different error rates with the extreme case of no errors when the link is good, and no throughput when the link is failed.

We assume that as soon as a link fails its repair starts and that failure and repair are Poisson processes. We use the following notation:

G : good state

B : bad (failed) state.

$\pi_i(G,t)$: probability that link i is in state G at time t.

$\pi_i(B,t)$: probability that link i is in state B at time t.

$e_i(G)$: error rate in state G, for link i.

$e_i(B)$: error rate in state B, for link i, with $e_i(G) \leq e_i(B)$.

η_i : failure rate for link i.

r_i : repair rate for link i.

We can also incorporate the measurements of the unsuccessful packets in the Markov model. So, let the random variables

$n_i(t-\tau)$: be the number of packets successfully transmitted during $[t-\tau, t)$, through link i; and

$u_i(t-\tau)$: be the number of packets failed transmission during $[t-\tau, t)$, through link i.

The probability that there will be $u_i(t-\tau)$ unsuccessful packets on link i, during the $[t-\tau, t)$ period, given link i is in the Bad state, (throughout the $[t-\tau, t)$ period), is

$$P[u_i(t-\tau) \quad / S_i(t-\tau) = B] = \binom{n_i(t-\tau) + u_i(t-\tau)}{u_i(t-\tau)} * *[e_i(B)]^{u_i(t-\tau)} * [1 - e_i(B)]^{n_i(t-\tau)}$$

Also the probability that there will be $u_i(t-\tau)$ unsuccessful packets on link i, during the $[t-\tau, t)$ period, given link i is in the Good state is

$$P[u_i(t-\tau) \quad / S_i(t-\tau) = G] = \binom{n_i(t-\tau) + u_i(t-\tau)}{u_i(t-\tau)} * *[e_i(G)]^{u_i(t-\tau)} * [1 - e_i(G)]^{n_i(t-\tau)}$$

The probability that there were $u_i(t-\tau)$ unsuccessful packets on link i, during the $[t-\tau, t)$ period is

$$P[u_i(t-\tau)] = P[u_i(t-\tau)/S_i(t-\tau) = G] * \pi_i(G, t-\tau) + +P[u_i(t-\tau)/S_i(t-\tau) = B] * \pi_i(B, t-\tau)$$

Then the probability that link i will be in the Bad state during the $[t, t+\tau)$ period, given that there were $u_i(t-\tau)$ unsuccessful packets during the $[t-\tau, t)$ period, is by Bayes rule

$$P[S_i(t) = B/u_i(t-\tau)] =$$

$$= P[S_i(t) = B, S_i(t-\tau) = G/u_i(t-\tau)] +$$
$$+ P[S_i(t) = B, S_i(t-\tau) = B/u_i(t-\tau)] =$$

$$= \frac{P[S_i(t) = B, S_i(t-\tau) = G, u_i(t-\tau)]}{P[u_i(t-\tau)]} +$$

$$+ \frac{P[S_i(t) = B, S_i(t-\tau) = B, u_i(t-\tau)]}{P[u_i(t-\tau)]} =$$

$$= \frac{P[u_i(t-\tau)/S_i(t-\tau) = G] * P[S_i(t) = B/S_i(t-\tau) = G]}{P[u_i(t-\tau)]} *$$

$$* P[S_i(t-\tau) = G] +$$

$$+ \frac{P[u_i(t-\tau)/S_i(t-\tau) = B] * P[S_i(t) = B/S_i(t-\tau) = B]}{P[u_i(t-\tau)]} *$$

$$* P[S_i(t-\tau) = B] =$$

$$= \frac{P[u_i(t-\tau)/S_i(t-\tau) = G] * \eta_i * \pi_i(G, t-\tau)}{P[u_i(t-\tau)]} +$$

$$+ \frac{P[u_i(t-\tau)/S_i(t-\tau) = B] * (1 - r_i) * \pi_i(B, t-\tau)}{P[u_i(t-\tau)]}$$

Similary, the probability that link i will be in the Good state, during the $[t, t+\tau)$ period, given that there were $u_i(t-\tau)$ unsuccessful packets, during the $[t-\tau, t)$ period is

$$P[S_i(t) = G/u_i(t-\tau)] =$$

$$= \frac{P[u_i(t-\tau)/S_i(t-\tau) = G] * (1 - \eta_i) * \pi_i(G, t-\tau)}{P[u_i(t-\tau)]} +$$

$$+ \frac{P[u_i(t-\tau)/S_i(t-\tau) = B] * r_i * \pi_i(B, t-\tau)}{P[u_i(t-\tau)]}$$

i) Eventually, we let

$$\begin{aligned} \hat{e}_i(t_n) = \quad & e_i(G) * P[S_i(t_n) = G/u_i(t_{n-1})] + \\ & + e_i(B) * P[S_i(t_n) = B/u_i(t_{n-1})] \end{aligned}$$

ii) Note also that we can use

$$\hat{e}_i(t_n) = \begin{cases} e_i(B) & \text{if } P[S_i(t_n) = B/u_i(t_{n-1})] > P[S_i(t_n) = G/u_i(t_{n-1})] \\ e_i(G) & \text{o.w.} \end{cases}$$

5.3 LEARNING AUTOMATON MODEL

Another approach to estimate $e_i(t_k)$, will be the use of learning automata [8]. In the following algorithm whenever a packet is successfully transmitted through a link, then the probability that this link is in the Good state is increased, otherwise it is decreased.

If successful packet transmission at time t_k, then

$$\pi_i(G, t_k) = \pi_i(G, t_{k-1}) + \alpha * \pi_i(B, t_{k-1})$$
$$\pi_i(B, t_k) = \pi_i(B, t_{k-1}) - \alpha * \pi_i(B, t_{k-1})$$

else

$$\pi_i(G, t_k) = \pi_i(G, t_{k-1}) - \beta * \pi_i(B, t_{k-1})$$
$$\pi_i(B, t_k) = \pi_i(B, t_{k-1}) + \beta * \pi_i(B, t_{k-1})$$

Where $0 < \alpha \ll \beta < 1$.

i) So we can use

$$\hat{e}_i(t_k) = e_i(G) * \pi_i(G, t_k) + e_i(B) * \pi_i(B, t_k)$$

ii) Finally another estimate will be

$$\hat{e}_i(t_k) = \begin{cases} e_i(B) & \text{if } \pi_i(G, t_k) < \pi_i(B, t_k) \\ e_i(G) & \text{o.w.} \end{cases}$$

5.4 NO STATE INFORMATION AVAILABLE

When no observations of packet successes and failures can be done, then we will depend exclusively on the two state Markov model. By solving the Markov Chain for $\pi_i(G, 0) = 1$, we have in steady state

$$\pi_i(G) = \frac{r_i}{\eta_i + r_i}$$

$$\pi_i(B) = \frac{\eta_i}{\eta_i + r_i}$$

i) Finally, we let

$$\hat{e}_i = \frac{e_i(G) * r_i + e_i(B) * \eta_i}{\eta_i + r_i}$$

ii) Another way will is to let

$$\hat{e}_i = \begin{cases} e_i(B) & \text{if } \eta_i > r_i \\ e_i(G) & \text{o.w.} \end{cases}$$

Thus, we have proposed several error rate estimates that can be used either when nothing is known about the failure behavior of the link, or when the link operates in two environments. We are currently working in extending these estimates in cases where we have some more information about the environment behavior.

Note also that wherever $P_i^*(t) = 0$, the source node continues to periodically send control traffic through this link and update the $u_i(t)$.

6. CONCLUSIONS

In this paper, queueing network, reliability Markov and Learning Automata models were introduced. Optimal static routing algorithms for routing packets in a simple network were proposed. It is shown that which links should be used is a function of the traffic loading. For low traffic and very bad links (high error rate), we find that these links will not be included in the set of routes selected. Single and multiple path dynamic routing algorithms were introduced.These algorithms minimize the failure probability of the packet transmission, or the overall network average packet delay. We considered two cases regarding the accuracy of the network condition measurements and several approaches for estimating the link states.

We modeled the error rate with a two state Markov model. This can be easily extended to a Markov model with many states. Also in a similar fashion, we can model the arrival rate (and service rate) with multiple state Markov models. In each state different arrival rate (service rate) will be assumed.

References

[1] D. P. Bertsekas and R. G. Gallager. *Data Networks.* Prentice Hall, 1987.

[2] A. A. Economides, P. A. Ioannou, and J. A. Silvester. Decentralized adaptive routing for virtual circuit networks using stochastic learning automata. *Proc. IEEE Infocom 88*, 1988.

[3] H. Frank and I. T. Frisch. *Communication, Transmission, and Transportation Networks.* Addison-Wesley, 1971.

[4] E. L. Hahne. Dynamic routing in an unreliable manufacturing network with limited storage. *Lab. for Information and Decision Systems, MIT, LIDS-TH-1063, OSP No. 87049*, pp. 1-242, Febr. 1981.

[5] L. Kleinrock. *Queueing Systems, Vol. 1: Theory.* J. Wiley & Sons, 1975.

[6] L. Kleinrock. *Queueing Systems, Vol. 2: Applications.* J. Wiley & Sons, 1976.

[7] J. F. Meyer. On evaluating the performability of degradable computing systems. *IEEE Trans. on Computers*, Vol C-29, pp. 720-731 Aug. 1980.

[8] K. S. Narendra and M. A. L. Thathachar. Learning automata : a survey. *IEEE Transn on Systems, Man, and Cybernetics*, Vol. SMC-4, No. 4, pp. 323-334, July. 1974.

[9] K. S. Narendra, E. A. Wright, and L. G. Mason. Application of learning automata to telephone traffic routing and control. *IEEE Transn on Systems, Man, and Cybernetics*, Vol. SMC-7, No. 11, pp. 785-792, Nov. 1977.

[10] G. J. Olsder and R. Suri. Time-optimal control of parts routing in a manufacturing system with failure-prone machines. *Proc. of IEEE 19th Conf. on Decision and Control*, pp.722-727, 1980.

[11] P. R. Srikantakumar and K. S. Narendra. A learning model for routing in telephone networks. *SIAM J. Control and Optimization*, vol-20, no 1, pp. 34-57, Jan. 1982.

[12] J. N. Tsitsiklis. Convexity and characterization of optimal policies in a dynamic routing problem. *J. of Optimization Theory and Applications:*, Vol. 44, No. 1, pp. 105-136, Sept. 1984.

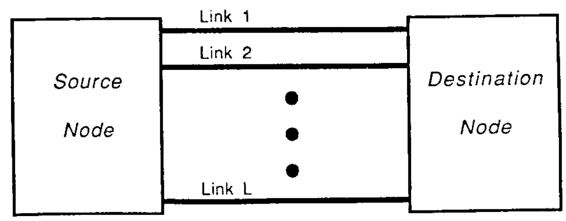

Fig. 1 Routing over L links to the same destination node

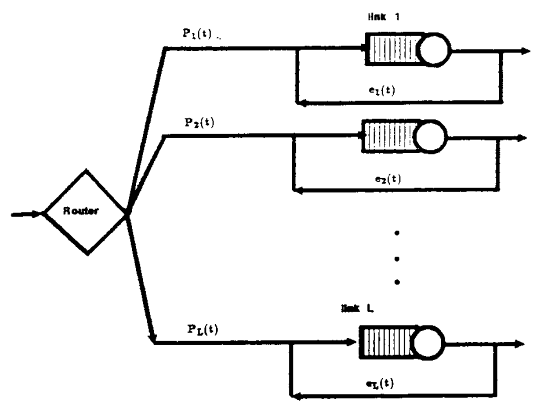

Fig. 2 Queueing network model that considers the link error rates

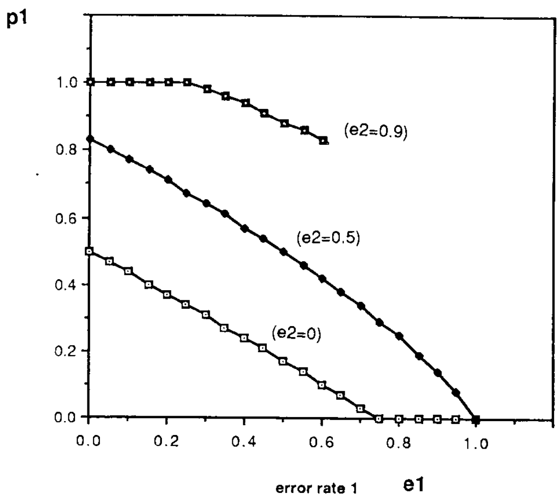

Fig. 3 Routing probability to link 1 versus link 1 error rate
$\mu = 1, C_1 = C_2 = 2, \lambda = 1$

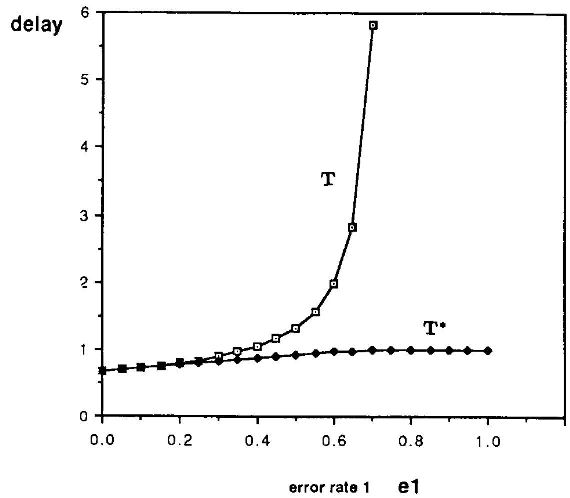

Fig. 5 Overall average packet delay versus link 1 error rate
$\mu = 1, C_1 = C_2 = 2, \lambda = 1, e_2 = 0$

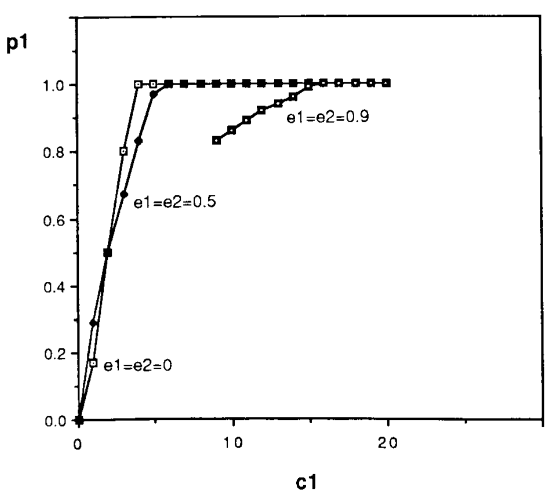

Fig. 4 Routing probability to link 1 versus link 1 capacity
$\mu = 1, C_2 = 2, \lambda = 1$

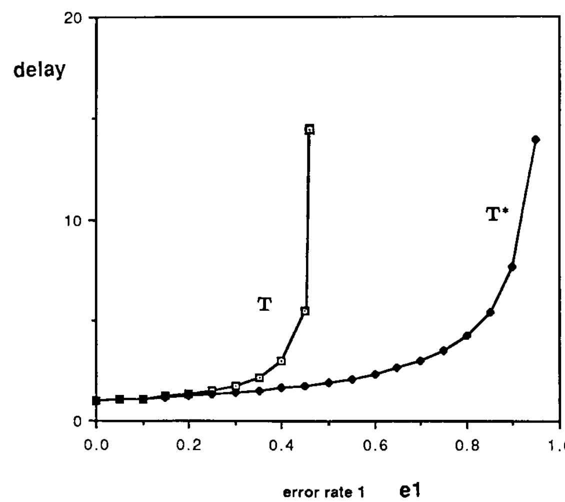

Fig. 6 Overall average packet delay versus link 1 error rate
$\mu = 1, C_1 = C_2 = 2, \lambda = 2, e_2 = 0$

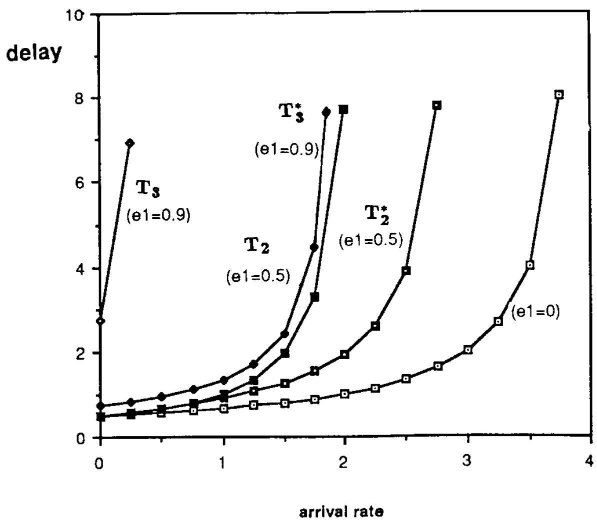

Fig. 7 Overall average packet delay versus arrival rate
$\mu = 1, C_1 = C_2 = 2, e_2 = 0$

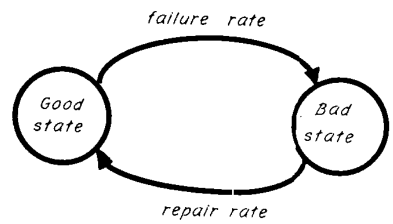

Fig. 9 Two state link error rate model

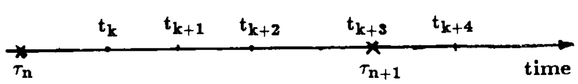

Fig. 8 Packet arrival, routing probability update and routing decision instants