Lazy Flooding: A New Technique for Information Dissemination in Distributed Network Systems

C. Chi*, D. Huang*, D. Lee‡ and X. Sun*
* Bell Laboratories, Lucent Technologies
E-mails:{chic, dhuang4, xsun1}@lucent.com
‡Ohio State University, Columbus, Ohio
E-mails: lee@cse.ohio-state.edu

***Abstract*— Flooding is a commonly used technique for network resource and topology information dissemination in the data communication networks. However, due to the well-known N-squared problem it causes network delay in response or even congestion. We propose a new flooding technique, called Lazy Flooding; it floods only when links reach a certain status. It significantly cuts down the number of floods and thus improves the data communication network response time. On the other hand, it has negligible effect on the network performance due to the selected flooding.**

I. Introduction

In a distributed network such as Internet and all-optical network, resource and network topology information has to be constantly updated at each network node, i.e., IP router and optical switch, such that correct routing or lightpath decision can be made in a dynamic and distributed networking environment.

To update this information IP protocol OSPF uses flooding; whenever there is a network topology change, i.e., link or node up or down, information is flooded throughout the whole network. Suppose that there are N nodes in an OSPF network. Then each update requires on the order of N^2 Link State Advertisement (LSA) messages to be flooded, causing serious problems for the network stability and scalability. Various techniques have been proposed to cope with this well-known *LSA N-squared* problem [1].

Since flooding consumes excessive communication resources, it poses a serious problem for the scalability of the link-state protocols. Based on an analysis of the consumption of the link bandwidth by LSAs, hierarchical OSPF network is described in [2] to reduce the flooded LSAs throughout the whole Internet. Observing that network aggregation can reduce the management packets traffic yet increase the inaccuracy of network information, an analytical model is developed in [3] for investigating the impact on QoS routing from information inaccuracy resulted by network aggregation. As a trade-off between performance and complexity, decision theory is applied [4] to study the best achievable performance in assessing wavelength availability using partial management information. There is quite an effort to study the trade-off between the quality of routing and messaging overhead. Rather than flooding complete link-state information to all routers, [5] proposes to selectively diffuse the link-state information based on a distributed computation of preferred paths. The flooding problem is also investigated for multicast routing protocols [6], and a number of procedures are proposed, including: reducing the message overhead such as Time To Live (TTL) protocol [7][8], Distributed Spanning Join (DSJ) protocols, and Directed Reverse Path Join (DRPJ) protocol. Due to the limited bandwidth in wireless networks, controlling the messaging overhead is an important design criterion. On-demand Multicast Routing Protocol (ODMRP), which is designed for mobile ad hoc networks, delivers packets to destination(s) on a mesh topology using scoped flooding of data[9], again for reducing the flooding of messages. Two flooding methods are proposed in [10]: self pruning and dominant pruning, which utilize neighbor information to reduce redundant transmissions in ad hoc networks. For reducing the message flooding traffic, particularly in QoS routing, it is proposed to control the conditions of updates. Different update policies, such as threshold based updates, equal class based updates and exponential class based updates, are provided in [11]. On the other hand, the resulting inaccuracy in network information and the corresponding impact on the performance of the QoS routing algorithms are also investigated in [11]. A closely related work is [12] where the relation between the link state information and the size and connectivity of the underlying network topology is studied.

In an all-optical network such as Lucent Lambda Router Network [13], the topology information, which includes the Optical Cross Connect (OXC) up and down and the fiber (link) cut and repair, is flooded throughout a separate Data Communication Network (DCN), which is a signaling network. In addition, when a lightpath is set up and torn down the channel status of the involved links is changed; the number of the available channels on the link is decreased and increased, respectively, and this information is also flooded throughout the DCN network. Obviously, for a network with N nodes (OXC), each channel status change results in an order of N^2 messages to be flooded via the DCN, and hence each lightpath set up (tear down) needs an order of LN^2 messages flooded where L is the number of hops the lightpath. The large number of flooded messages may lead to the signaling network delay in response and instability and also make it difficult to scale up. As a matter of fact, as lightpaths are set up and torn down, a huge volume of information is to be flooded while OSPF does not flood the link load update information resulted by routine paths being set up and torn down. Therefore, the delay in network response caused by flooding in the DCN of an all-optical network is even worse than that of Internet.

The signaling message flooding problem that we are dealing with in all-optical networks is apparently different than what has been studied in [5]-[10]. We take Internet message flooding as a comparison. First, most of the published works are on in-band signaling where signaling messages are flooded in the data network, such as Internet, competing for resources with the data traffic. For all-optical networks, such as [13], resource information messages are flooded in a different dedicated signaling network than the data network; there is no competition for resources with the data traffic. Second, traffic load information helps for routing in the Internet, particularly QoS routing, and the inaccuracy of the information may affect the performance. However, for all-optical network, if all the channels on a link have been allocated yet other nodes are not informed, false lightpaths may be constructed - dead lightpaths, which occupy precious resources yet do not transmit any data. Finally, the signaling network of all-optical networks is extremely sensitive to the delay; any congestion or even unexpected delay from excessively flooded signaling messages in the DCN of all-optical network is intolerable, since the response time expected from the DCN for lightpath construction and tearing down is on an order of micro-seconds in the worst case; lightpath construction, protection and restoration time is crucial for commercial optical networks. Consequently, we want to reduce the DCN traffic as much as we can, so long as it does not result in false lightpaths nor wasting channel capacities by leaving them idle.

To cope with this message flooding problem of optical networks we propose a new technique: Lazy Flooding. It significantly reduces the number of flooded messages for the network resource and topology information update with little affecting on the lightpath construction and, consequently, path protection and restoration. Given the different nature of our signaling network flooding problem, the focus of our analysis and the criteria of our experiments are different than that in most of the published literature. Our focus is on: (1) Reducing the flooded resource information messages as much as possible to guarantee the response time of the signaling network; (2) Prevent the construction of lightpaths over a link where there is no available channels; and (3) The inaccurate link capacity information does not affect load sensitive lightpath computation. On the other hand, with due modifications our technique could also be useful for applications other than optical network signaling.

The idea of Lazy Flooding is simple. Suppose that there are k channels in a link (fiber). Instead of flooding for each channel status change, i.e., from being available to being occupied or vice versa, we hold it until it reaches certain points. Apparently, this method is “lazy”, i.e., it does not flood for each channel update, and we call it *Lazy Flooding*. In contrast, we call it *All Flooding* if each channel status change is flooded. Different flooding decisions give different Lazy Flooding methods, and we study three in this paper:

(1) Threshold Flooding. For a pre-determined threshold value $0 < T \leq k$, the number of the available channels in a link is flooded if and only if when it is less than T after a channel update.

(2) Exponential Flooding. For a geometric sequence of numbers above the threshold: $0 < T \leq g_1 < g_2 < \cdots < g_r \leq k$, the number of the available channels in a link is flooded if and only if when it is less than T or equal to g_i, $i = 1, \cdots, r$, after a channel update.

(3) Fibonacci Flooding. For a Fibonacci sequence of numbers above the threshold: $0 < T \leq f_1 < f_2 < \cdots < f_r \leq k$, the number of the available channels in a link is flooded if and only if when it is less than T or equal to f_i, $i = 1, \cdots, r$, after a channel update.

We shall show that these Lazy Flooding techniques significantly reduce the number of messages flooded throughout the DCN by both mathematical analysis and simulation. We first discuss informally the cost, i.e., the possible disadvantages of not flooding with each link status update.

(1) Link blocking. Suppose that there are no channels available on a link. However, due to Lazy Flooding a remote node still believes that a certain number of channels is available in this link according to its own database information, and, consequently, may still construct lightpaths passing through this link, leading to a blocking link on a lightpath. To cope with this problem, we set a threshold value T in all the Lazy Flooding techniques. When the number of available channels is getting smaller than the threshold value T we flood the information for each channel update to inform all the nodes of the situation. As will be shown later by analysis and simulation, the blocking probability caused by Lazy Flooding is negligible.

(2) Network blocking. Suppose that channels in a link are released from tearing down the lightpaths. For a same reason as in (1), a remote node does not know about it and believes that there are no channels available in this link, and, consequently, it may not be able to find any lightpath for a specified destination node. This scenario blocks the network for a lightpath computation, and, as a matter of fact, there are indeed resources available. Similarly, it can be shown that this probability caused by Lazy Flooding is also negligible

(3) Discrepancy in link load. OSPF uses shortest path routing. Certain shortest path algorithms assign a weight to each link for the computation, and the weight is determined by the link load [14]. Similarly, certain lightpath computation algorithms also use shortest path algorithms and assign a weight to each link, which is typically the number of the occupied channels. The rationale is: avoid routing through a link where there are almost no channels available. Obviously, Lazy Flooding does not keep this information updated with each link status update for each OXC on the network. Threshold Flooding is the worst; there is no information at all except when the available capacity of a link is below a threshold. For applications such as path/link protection in all-optical network, we need more information of the channel status of all the links for protection

path routing computation. Fibonacci and Exponential Flooding aim at providing approximate information of the channel status of all the links yet without flooding for each channel status update. Note that Fibonacci Flooding tends to flood more than Exponential Flooding when the number of the available channels is low and the channel status is more sensitive for the link and network blocking. Our analysis and simulation will show that both Fibonacci and Exponential Flooding lead to rather negligible discrepancies in the link load while significantly reduce the number of floods.

(4) Network global optimization. Due to Lazy Flooding all the nodes in the network may not have the updated and accurate information of the network topology and resources. Apparently, the overall lightpath computation is not optimized since the available information is approximate. Note that even with the exact information available the overall optimization is NP-hard when it is off line, and an on-line optimization is even harder. Surprisingly, our experiments show that in general Lazy Flooding does not cause degradation in network overall performance. Under certain circumstances, it even outperforms All Flooding. This anomaly has also been observed with the Internet performance.

The Lazy Flooding techniques we propose can be used for both the all-optical network and Internet signaling. For clarify, in most of our explanations in the sequel, we follow the all-optical networking scenarios.

The idea similar to Threshold Flooding and Exponential Flooding has been proposed in [11], but no formal definition and analytical model are provided to quantitatively analyze the time discrepancy that is resulted from the delayed updates and how much traffic is saved by these flooding policies. This paper uses an embedded Markov chain to model the link capacity change process in an all-optical network for a formal analysis. Extensive simulation is performed to support the analytical result in a single link and also throughout a whole network.

In section 2, we propose an analytical model to investigate above problems in the context of single link network. Some experimental results are also reported to check if the model is compatible with reality. It is difficult to give a model to analyze the impact of different flooding schemes on network performance with routing algorithms involved. In section 3, we use discrete event simulation to study the relationship between these flooding schemes with network performance. The experimental results on two different routing algorithms are reported: one is sensitive to the link channel status and the other is not. The paper concludes in section 4.

II. Mathematical Analysis

For a mathematical analysis, we consider a single link. A widely used model is the $M/M/B$ queueing system: the arrival rate of requests for channels on a link follows a Poisson distribution with parameter λ and the service rate, at which lightpaths are cleared from a link, follows an exponential distribution with parameter μ. The total capacity (channels) of a link is B.

This model represents a continuous time process $\{X(t)\}$ with values in the number of the used channels on this link: 0,1,...B. We are interested in the flooding probabilities while an OXC floods when there is a change in a link capacity. For this, we introduce an imbedded Markov chain (MC) in $\{X(t)\}$. Let $\{\tau_n\}$ be the stopping time sequence, at which the occupied capacity on a link changes, i.e.,

$$\begin{aligned} \tau_0 &= \min\{t>0; X(t) \neq X(0)\} \\ \tau_n &= \min\{t>\tau_{n-1}; X(t) \neq X(\tau_{n-1})\}, \\ & \quad n = 1,2,... \\ x_n &= X(\tau_n),\ n = 0,1,2,... \end{aligned}$$

According to the classical results in queueing theory [15] $\{x_n\}$ is a stationary Markov chain.

The All Flooding scheme has a flooding probability 1 since it floods at each time τ_n. We now estimate the flooding probabilities of all the Lazy Flooding schemes to show their gain. On the other hand, due to Lazy Flooding, other nodes in the network do not know the exact capacity on this link. To measure the missing information, we compute the mean and variance of the difference between the exact link capacity and the capacity flooded to the network.

A. Flooding Methods

A straightforward method is All Flooding: the link capacity, i.e., the number of available channels, is flooded whenever there is a change. Lazy Flooding is different; it floods after a link capacity change if the available channels $c_n = B - x_n \leq L$. For $c_n > L$, it is flooded selectively. There are three different selection criteria, each of which has its own merit for different networks.

Threshold Flooding After a link capacity change, the number of the available channels c_n is flooded if and only if $c_n \leq L$.

Exponential Flooding

Let

$$b_k = k, k = 0,1,...,L, \quad b_k = L + 2^{k-L}, k = L+1,...K,$$

where $K = \lfloor \log_2(B-L) \rfloor + L$ and $\lfloor x \rfloor$ denotes the largest integer no more than x.

After a link capacity has changed, the number of the available channels c_n is flooded if and only if it equals to one of the values $b_k, k = 1,...K$.

Fibonacci Flooding

Let

$$\begin{aligned} f_k &= k, k = 0,1,...,L, \\ f_k &= f_{k-1} + f_{k-2} - L + 3,\ k = L+1, L+2,...K. \end{aligned}$$

where K is the largest index k such that $f_k \leq B$.

After a link capacity has changed, the number of the available channels c_n is flooded if and only if it equals one of the values $f_k, k = 1,...K$.

We first compute the stationary distribution of the Ergodic MC, then estimate the probabilities of these three flooding methods on the basis of the stationary distribution and transit probabilities of the MC.

B. Stationary Distribution for the Imbedded Markov chain

Stationary distributions of imbedded Markov chains have been studied in queueing theory [15][16]. However, only two types of imbedded discrete time Markov chains were studied: only at **either arriving epoch or departure epoch**. In both cases, the stationary distribution of the imbedded discrete time Markov chain is exactly the same as the continuous time process $\{X(t)\}$, namely the Erlang distribution [16].

However, for link capacity changes by OXCs, we have to consider the imbedded Markov chain at **both arriving and departure epochs**, since in both cases the OXC floods a piece of information. Therefore, the existing results cannot be used to estimate $\{x_n = B - c_n\}$ directly. To obtain the probability structure of $\{x_n\}$ and its stationary distribution, we need the following lemmas.

Lemma 1: The transition probabilities of $\{x_n\}$ are given by

$$\begin{aligned}
p_{0,j} &= \Pr(x_n = j | x_{n-1} = 0) = \begin{cases} 1 & j = 1 \\ 0 & \text{otherwise} \end{cases} \\
p_{B,j} &= \Pr(x_n = j | x_{n-1} = B) = \begin{cases} 1 & j = B-1 \\ 0 & \text{otherwise} \end{cases} \\
p_{i,j} &= \Pr(x_n = j | x_{n-1} = i) \quad i = 1, ..., B-1 \\
&= \begin{cases} \frac{\lambda}{\lambda + i\mu} & j = i+1 \\ \frac{i\mu}{\lambda + i\mu} & j = i-1 \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{1}$$

Proof:

Let Y be the random variable for the time difference between a stopping time, τ_{n-1}, and the first request arriving time after τ_{n-1}. Let Z be the random variable for the time difference between the same stopping time τ_{n-1} and the first service finishing time after τ_{n-1}. Assume that at τ_{n-1} the system is in the state i. Then, since both arriving process and service process are memoryless, the two times follow exponential distributions with parameters λ and $i\mu$, respectively. Since the exponential distribution is continuous, there are two or more arrivals or finished services at any time epoch zero. Therefore,

$$\begin{aligned}
p_{0,j} &= 0 \ \text{ if } \ j > 1; \\
p_{B,j} &= 0 \ \text{ if } \ j < B-1; \\
p_{i,j} &= 0 \ \text{ if } |i-j| \neq 1 \ \text{ for } \ i \neq 0, B.
\end{aligned}$$

Equation (1) follows from:

$$\begin{aligned}
p_{i,i+1} &= \Pr(x_n = i+1 | x_{n-1} = i) \\
&= \Pr(Y < Z) = \int_0^\infty \int_y^\infty \lambda i\mu e^{-\lambda y - i\mu z} dz dy \\
&= \frac{\lambda}{\lambda + i\mu}
\end{aligned}$$

■

Lemma 2: Let $\rho = \frac{\lambda}{\mu}$. The stationary distribution of $\{x_n\}$ is given by

$$\begin{aligned}
w_0 &= \frac{1}{2\sum_{k=0}^{B-1} \frac{\rho^k}{k!}}, \\
w_k &= \left(1 + \frac{\rho}{k}\right) \frac{\rho^{k-1}}{(k-1)!} w_0 \\
&= \left(\frac{\rho^k}{k!} + \frac{\rho^{k-1}}{(k-1)!}\right) w_0, \\
k &= 1, 2, ..., B-1. \\
w_B &= \frac{\rho^{B-1}}{(B-1)!} w_0.
\end{aligned} \tag{2}$$

Proof: It follows from Lemma 1 that

$$[w_0, w_1, ..., w_B] = [w_0, w_1, ..., w_B] \times \begin{bmatrix} 0 & 1 & 0 & ... & 0 & 0 \\ \frac{\mu}{\lambda+\mu} & 0 & \frac{\lambda}{\lambda+\mu} & ... & 0 & 0 \\ 0 & \frac{2\mu}{\lambda+2\mu} & 0 & ... & 0 & 0 \\ . & . & . & ... & . & . \\ 0 & 0 & 0 & ... & \frac{\lambda}{\lambda+(B-2)\mu} & 0 \\ 0 & 0 & 0 & ... & 0 & \frac{\lambda}{\lambda+(B-1)\mu} \\ 0 & 0 & 0 & ... & 1 & 0 \end{bmatrix}$$

One can simply verify that (2) holds for $k = 1$ and 2. Assume that (2) holds for $k-2$ and $k-1$, then

$$\begin{aligned}
w_k &= \frac{\lambda + k\mu}{k\mu} \left[w_{k-1} - w_{k-2} \frac{\lambda}{\lambda + (k-2)\mu} \right] \\
&= \left(1 + \frac{\rho}{k}\right) \frac{\rho^{k-1}}{(k-1)!} w_0.
\end{aligned}$$

We have:

$$\begin{aligned}
w_B &= w_{B-1} - w_{B-2} \frac{\lambda}{\lambda + (B-2)\mu} \\
&= \frac{\rho^{B-1}}{(B-1)!} w_0.
\end{aligned}$$

Since $\sum_{k=0}^{B} w_k = 1$, we have

$$w_0 = \frac{1}{2\sum_{k=0}^{B-1} \frac{\rho^k}{k!}}.$$

This completes the proof. ■

Remark: The Erlang distribution in an $M/M/B$ system is given by

$$\gamma_0 = \frac{1}{\sum_{k=0}^{B} \frac{\rho^k}{k!}}, \quad \gamma_k = \frac{\rho^k}{k!} \gamma_0, \quad k = 1, 2, ..., B.$$

When ρ is not too large or too small in comparison to B, the mode of the stationary distribution is not at the two ends. Then $w_0 \approx \frac{\gamma_0}{2}$, $w_B \approx \frac{\gamma_B}{2}$, while $w_k \approx \gamma_k, 0 < k < B$. ■

Fig. 1 displays the stationary distributions with a comparison to the Erlang distribution where $\{w_k\}$ is very close to $\{\gamma_k\}$ except for ρ close to 0 and $B = 32$.

C. Probabilities for Different Flooding Techniques

Let $0 \leq L \leq B$. After a link capacity has changed, if the resulting $c_t = B - x_t \leq L$, link information is flooded.

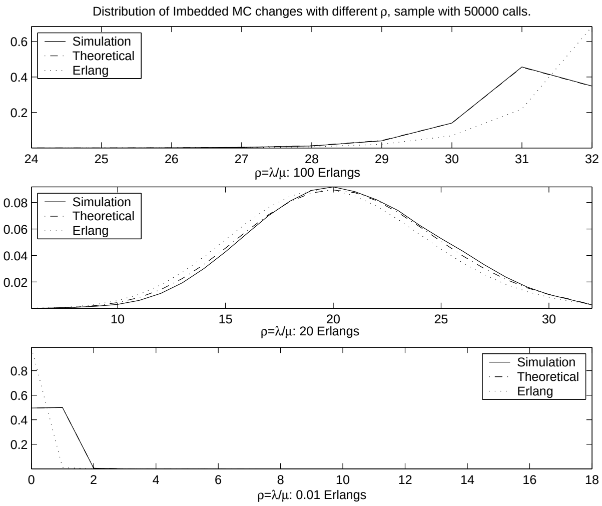

Fig. 1. Distribution of Imbedded MC

When $c_t > L$, we consider different flooding protocols for: Threshold, Exponential and Fibonacci Floodings. Let

$$\pi_k = \Pr(c_n = k) = w_{B-k}, \quad k = 0, 1, 2, ..., B.$$

Then

(i) Threshold Flooding: no flood for $c_t > L$. Therefore, the flooding probability is

$$P_t = \Pr\{[c_t \le L]\} = \sum_{k=0}^{L} \pi_k.$$

(ii) Exponential Flooding: Let

$$b_k = \begin{cases} k, & k = 0, 1, ..., L \\ L + 2^{k-L}, & k = L+1, ...K \end{cases}$$

where $K = \lfloor \log_2(B - L) \rfloor + L$. We need to calculate the flooding probability

$$P_e = \Pr\{\cup_k [c_t = b_k]\} = \sum_{k=0}^{K} \pi_{b_k}.$$

(iii) Fibonacci Flooding: Let

$$f_k = \begin{cases} k, & k = 0, 1, ..., L \\ f_{k-1} + f_{k-2} - L + 3, & k = L+1, ...K \end{cases}$$

where K is the largest index k such that $f_k \le B$. Therefore, the flooding probability is

$$P_f = \Pr\{\cup_k [c_t = f_k]\} = \sum_{k=0}^{K} \pi_{f_k}.$$

Remark: In this paper, we consider the flooding when a link "enters" a predetermined capacity. We can also consider the case when a link "leaves" a predetermined capacity. In this case, the probability of flooding at the pre-determined value $b_k(f_k)$ is the same:

$$\Pr\{c_{t-1} = b_k, c_t = b_k + 1\} + \Pr\{c_{t-1} = b_k, c_t = b_k - 1\} = \Pr\{c_{t-1} = b_k\}. \quad \blacksquare$$

D. Mean Values of Floodings

Another quantity showing the saving of Lazy Flooding is the average flooding numbers. We calculate the mean for All Flooding first, and then derive that for Lazy Flooding.

Assume that the time unit for the Poisson and exponential parameters in the $M/M/B$ model is Δ. Therefore, the mean of the requests for channels for lightpath set-up in a given time period Δ is λ.

For All Flooding, let N_1^{all} be the average number of floodings upon requests for channels in the time period Δ, and let $u_i, 0 \le u_i \le u_{i+1}, i = 1, 2, ...,$ be the time epoch that the i-th request arrives. Assume that the capacity of the channel at time epoch t is $C(t)$. Define an index function

$$I(A) = \begin{cases} 1, & \text{event } A \text{ is true} \\ 0, & \text{otherwise} \end{cases}$$

Assume that the system achieves its equilibrium status. Since event $[u_i < \Delta]$ and event $[C(u_i) < B]$ are independent,

$$\begin{aligned} N_1^{all} &= E\sum_{k=1}^{\infty} I(u_i < \Delta, C(u_i) > 0) \\ &= \Pr\{C(t) > 0\} \sum_{k=1}^{\infty} E[I(u_i < \Delta)] \\ &= \lambda \Pr\{C(t) > 0\}. \end{aligned} \tag{3}$$

Since u_i is the arriving epoch only, $\Pr\{C(t) > 0\}$ can be calculated by (Erlang):

$$\Pr\{C(t) > 0\} = 1 - \Pr\{C(t) = 0\} = 1 - \frac{\frac{\rho^B}{B!}}{\sum_{k=0}^{B} \frac{\rho^k}{k!}}.$$

Now let N_2^{all} be the number of floodings when a link acquires released channels from lightpaths tearing down in Δ. When the system achieves its equilibrium status we must have

$$N_2^{all} = N_1^{all}; \tag{4}$$

otherwise the capacity will strictly increase or decrease, and it contradicts the equilibrium status assumption.

Thus, the mean of the flooding numbers is

$$2\lambda \left(1 - \frac{\frac{\rho^B}{B!}}{\sum_{k=0}^{B} \frac{\rho^k}{k!}}\right). \tag{5}$$

For Lazy Flooding at the given capacities: $S = \{s_1, s_2, ..., s_n\}, 0 \le s_i, n \le B$, let N_1^{lazy} be the average number of floodings from requests for channels in Δ. Similarly,

$$\begin{aligned} N_1^{lazy} &= E\sum_{k=1}^{\infty} I(u_i < \Delta, C(u_i) - 1 \in S) \\ &= \Pr\{C(t) - 1 \in S\} \sum_{k=1}^{\infty} E[I(u_i < \Delta)] \\ &= \lambda \Pr\{C(t) - 1 \in S\}. \end{aligned} \tag{6}$$

Substituting the formulas in Section II-C into (6), we can calculate the average number of floodings when the capacity enters a flooding point.

Let N_2^{lazy} be the average number of floodings in Δ when the capacity leaves a flooding point. With ω in the σ-algebra generated by $\{C(t), 0 \le t < \infty\}$, let $k_i^{(1)}(t,\omega)$ be the number of floodings in the time period $(0,t]$ when the capacity leaves the flooding point s_i, and let $k_i^{(2)}(t,\omega)-$ be that when the capacity enters the flooding point, $i = 1, 2, ..., n$. Then

$$\left| k_i^{(1)}(t,\omega) - k_i^{(2)}(t,\omega) \right| \le 1. \tag{7}$$

Since the process is Ergodic, we have

$$\begin{aligned} \lim_{m\to\infty} \frac{1}{m} \sum_{i=1}^{n} k_i^{(1)}(m\Delta,\omega) &= N_1^{lazy}, \\ \lim_{m\to\infty} \frac{1}{m} \sum_{i=1}^{n} k_i^{(2)}(m\Delta,\omega) &= N_2^{lazy}. \end{aligned}$$

It follows from (7) that

$$\frac{1}{m} \left| \sum_{i=1}^{n} k_i^{(1)}(m\Delta,\omega) - \sum_{i=1}^{n} k_i^{(2)}(m\Delta,\omega) \right| \le \frac{n}{m}.$$

Then we have

$$\left| N_1^{lazy} - N_2^{lazy} \right| = 0,$$

i.e.,

$$N_2^{lazy} = \lambda \Pr\{C(t) - 1 \in S\}.$$

Thus, the total mean of floodings when the capacity enters a flooding point is

$$N_1^{lazy} + N_2^{lazy} = 2\lambda \Pr\{C(t) - 1 \in S\}.$$

The savings by Lazy Flooding in average flooding numbers are significant:

$$\left(1 - \frac{\Pr\{C(t) - 1 \in S\}}{\Pr\{C(t) > 0\}}\right).$$

Calculation result on the flooding probability is given in next subsection.

E. A Comparison

Fig. 2 shows the flooding probabilities of the three flooding methods over different model parameter $\rho = \frac{\lambda}{\mu}$ and threshold L. For All Flooding, the flooding probability is always one.

It is clear that the larger the threshold L is, the more the flooding will be. Recall that approximately the mean of capacity is $(B - \rho)$. Hence if ρ is large, the number of the available channels will be small, and that results in more flooding. Overall the three Lazy Flooding methods have a similar flood probability that is much smaller than All Flooding.

Fig. 3 shows the saving in percentages in the mean of the flooding numbers of the three Lazy Flooding methods in comparison to All Flooding. The savings are significant, especially when the channel is not too busy ($\rho = \frac{\lambda}{\mu}$ is small in comparison to B). Also, the savings are in a decreasing order of: Threshold, Exponential and Fibonacci Flooding. Even in the worst case, we still have 85% savings.

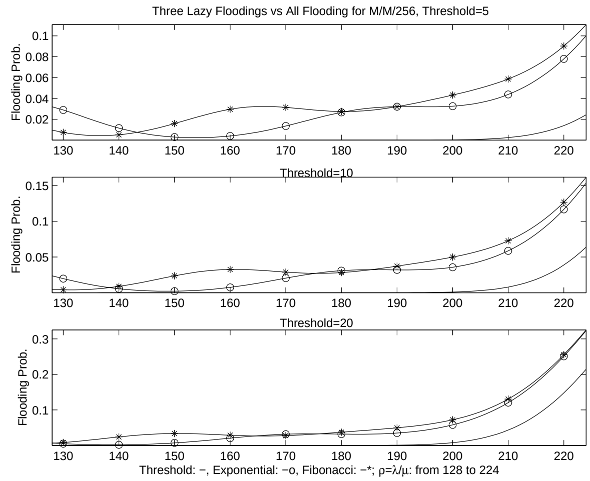

Fig. 2. Flooding Probabilities

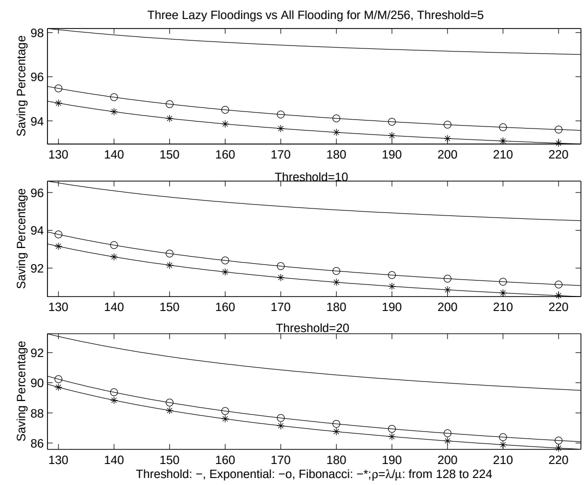

Fig. 3. Flooding Mean: Lazy Flooding vs. All Flooding

F. Mean and Variance of Information Distortion

We now discuss how much information distortion the three Lazy Flooding methods have resulted for a single link case. The information discrepancy is measured by the difference between the true and the flooded numbers of the available channels, denoted by D_n, which is also a random variable. However, on the basis of the stationary distribution of c_n we can find the corresponding distribution of D_n for different flooding methods and hence its mean and variance.

1) Threshold Flooding: For the case, we have

$$D_n = \begin{cases} 0, & c_n \le L, \\ c_n - L, & c_n > L. \end{cases}$$

Therefore,

$$\begin{aligned} E(D_n) &= \sum_{k=L+1}^{B} (k - L)\,\pi_k; \qquad (8) \\ Var(D_n) &= \sum_{k=L+1}^{B} [k - L - E(D_n)]^2\,\pi_k \end{aligned}$$

$$+ \quad [E(D_n)]^2 \sum_{k=0}^{L} \pi_k. \qquad (9)$$

2) Stationary Distribution of Augmented Vector Process: For exponential and Fibonacci Flooding, the situation is more complicated. Let $\{\hat{c}_n, n = 0, 1, ...\}$ be the known capacity at time τ_n on a link by remote nodes when the Exponential Flooding is used. It is clear that $\{\hat{c}_n, n = 0, 1, ...\}$ is not a Markov chain. We have to consider its augmented vector process $Y_n = [c_n, \hat{c}_n]$ that is a 2-dimensional Markov chain. Let

$$p_{k,j} = \lim_{n\to\infty} \Pr(c_n = j, \hat{c}_n = b_k), b_{k-1} < j < b_{k+1}. \qquad (10)$$

We have

Lemma 3:

$$(i)\pi_j = p_{k,j} + p_{k+1,j}, b_k < j < b_{k+1}. \qquad (11)$$

(ii) For $0 \le k < L$,

$$p_{k,j} = \begin{cases} \pi_k, & j = b_k, \\ 0, & \text{otherwise.} \end{cases} \qquad (12)$$

(iii) For $k = L$,

$$p_{L,k} = \begin{cases} \pi_L, & k = L \\ 0, & k < L. \end{cases} \qquad (13)$$

(iv) For $L < k < b_K$,

$$p_{k,j} = \begin{cases} \pi_k, & j = b_k, \\ 0, & j \le b_{k-1} \quad or \quad j \ge b_{k+1}. \end{cases} \qquad (14)$$

(v) For $k = b_K$,

$$p_{k,j} = \pi_j, \quad b_K \le j. \qquad (15)$$

Proof: Since $\hat{c}_n = b_k$ if and only if there is a non-negative integer $j \ge 0$ such that $c_{n-j} = b_k$ and $b_{k-1} < c_s < b_{k+1}$ for all $n - j \le s \le n$. We have

$$\Pr(c_n = j) = \Pr(c_n = j, \hat{c}_n = b_k) + \Pr(c_n = j, \hat{c}_n = b_{k+1}), b_k < j < b_{k+1}.$$

Equation (11) follows from (10), and (12)-(15) can be derived similarly.

Therefore, we only need to consider (i) $p_{L,j}$ for $L < j < b_{L+1}$; (ii) $p_{k,j}$ for both $b_{k-1} < j < b_k$ and $b_k < j < b_{k+1}$ when $L < k < b_K$; (iii) $p_{K,j}$ for $b_{K-1} < j < b_K$. In summary, we want to calculate $\{p_{k,j}, j = b_k + 1, ..., b_{k+1} - 1\}$ and $\{p_{k+1,j}, j = b_k + 1, ..., b_{k+1} - 1\}$ for $k = L, L+1, ...K - 1$.

For $k > L$, $b_k - b_{k-1} > 1$ and $b_{k+1} - b_k > 1$, and hence if $\hat{c}_n = b_k$ and $c_n \ne b_k$, we have $\hat{c}_{n-1} = b_k$. Thus, for $b_k < j < b_{k+1}$ we have

$$\begin{aligned} & \Pr(c_n = j, \hat{c}_n = b_k) \\ = & \Pr(c_n = j, \hat{c}_n = b_k, \hat{c}_{n-1} = b_k) \\ = & \Pr(c_n = j, \hat{c}_n = b_k, c_{n-1} = j - 1, \hat{c}_{n-1} = b_k) \\ + & \Pr(c_n = j, \hat{c}_n = b_k, c_{n-1} = j + 1, \hat{c}_{n-1} = b_k). \end{aligned}$$

Furthermore, when $b_k < j < b_{k+1}$ we have

$$\begin{aligned} & \Pr(c_n = j, \hat{c}_n = b_k, c_{n-1} = j - 1, \hat{c}_{n-1} = b_k) \\ = & \Pr(c_n = j, c_{n-1} = j - 1, \hat{c}_{n-1} = b_k) \\ = & \Pr(c_n = j | c_{n-1} = j - 1, \hat{c}_{n-1} = b_k) \\ & \Pr(c_{n-1} = j - 1, \hat{c}_{n-1} = b_k) \\ = & \Pr(c_n = j | c_{n-1} = j - 1) \\ & \Pr(c_{n-1} = j - 1, \hat{c}_{n-1} = b_k) \\ = & \frac{B - j + 1}{\rho + B - j + 1} \Pr(c_{n-1} = j - 1, \hat{c}_{n-1} = b_k). \end{aligned}$$

Similarly,

$$\begin{aligned} & \Pr(c_n = j, \hat{c}_n = b_k, c_{n-1} = j + 1, \hat{c}_{n-1} = b_k) \\ = & \frac{\lambda}{\lambda + (B - j - 1)\mu} \Pr(c_{n-1} = j + 1, \hat{c}_{n-1} = b_k). \end{aligned}$$

Therefore,

$$\begin{aligned} & \frac{B - j + 1}{\rho + B - j + 1} p_{k,j-1} - p_{k,j} \\ + & \frac{\rho}{\rho + B - j - 1} p_{k,j+1} = 0, \\ & b_k < j < b_{k+1}. \end{aligned} \qquad (16)$$

Consequently, we can calculate $\{p_{k,j}, j = b_k + 1, ..., b_{k+1} - 1\}$ for $k = L, L + 1, ...K - 1$ by

$$\begin{bmatrix} p_{k,b_k+1} \\ p_{k,b_k+2} \\ p_{k,b_k+3} \\ \vdots \\ p_{k,b_{k+1}-2} \\ p_{k,b_{k+1}-1} \end{bmatrix} = \begin{bmatrix} -1 & \frac{\rho}{\rho+B-b_k-2} & ... & 0 \\ \frac{B-b_k-1}{\rho+B-b_k-1} & -1 & ... & 0 \\ 0 & \frac{B-b_k-2}{\rho+B-b_k-2} & ... & 0 \\ . & . & ... & . \\ 0 & 0 & ... & \frac{\rho}{\rho+B-b_{k+1}} \\ 0 & 0 & ... & -1 \end{bmatrix}^{-1} \begin{bmatrix} -\frac{(B-b_k)\pi_{b_k}}{\rho+B-b_k} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}.$$

Finally, $\{p_{k+1,j}, j = b_k + 1, ..., b_{k+1} - 1\}$, $k = L, L + 1, ...K - 1$, can be obtained by

$$p_{k+1,j} = \pi_j - p_{k,j}.$$

On the basis of the joint distribution of $\{c_n, \hat{c}_n\}$, we can derive the mean and the variance of the discrepancy by the following formulae:

$$E\left(c_{n}-\hat{c}_{n}\right) = \sum_{k=L+1}^{K} \sum_{j=b_{k-1}+1}^{b_{k+1}-1}\left(j-b_{k}\right) p_{k, j} \quad (17)$$

Set $E(c_n - \hat{c}_n) = m$, the variance of $c_n - \hat{c}_n$ is:

$$\begin{aligned} & Var\left(c_{n}-\hat{c}_{n}\right) \\ = & E\left(c_{n}-\hat{c}_{n}-m\right)^{2} \\ = & \sum_{k=L+1}^{K} \sum_{j=b_{k-1}+1}^{b_{k+1}-1}\left(j-b_{k}\right)^{2} p_{k, j}-m^{2} \end{aligned} \quad (18)$$

3) A Comparison: We display the mean and variance of the information distortion D_n of the three Lazy Flooding methods with different threshold values $L = 5, 10$ and 20 and with different $M/M/B$ model parameters $\rho = 128$ to 224 with step size equal to 1. For all three cases, the parameter μ is fixed. See Fig.4 and Fig.5 below.

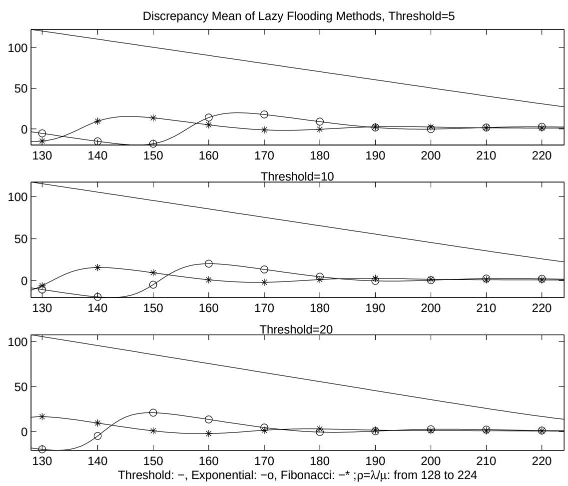

Fig. 4. Mean of the Discrepancy

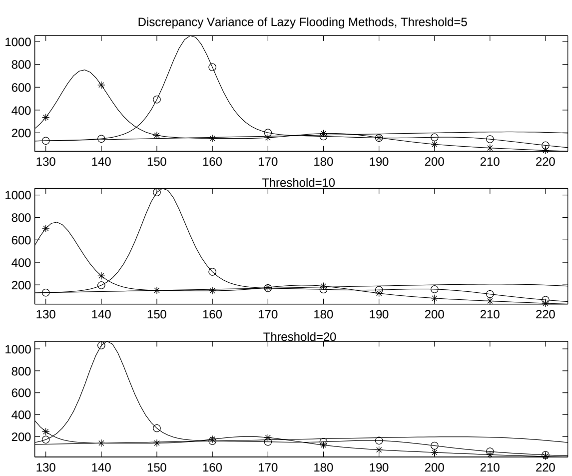

Fig. 5. Variance of the Discrepancy

Since $\rho = \frac{\lambda}{\mu}$, the larger the ρ is, the less the available capacity will be, resulting in more flooding. Therefore, it is not surprising that for large ρ, the mean and variance of the distortion D_n become small. Also, the effects of the threshold L are basically a shift of the curves towards the left. From the plots we can see that the variance of the distortion in Threshold Flooding increases and then decreases against ρ; as ρ increases, the stationary distribution $\{\pi_k\}$ will concentrate on the lower index end, i.e., the most possible available capacity is small, we flood more frequently, and this reduces both the mean and variance. There are rapid changes in both means and variances for Exponential Flooding and Fibonacci Flooding. This is because on both sides of the peaks, $\hat{c}_t$ is equal to different values with a very large probability. Therefore, the variance is not too large. However, for Exponential Flooding when the value ρ makes

$$\Pr\left(\hat{c}_t = b_k\right) \approx \Pr\left(\hat{c}_t = b_{k-1}\right),$$

the variance will be large. Similarly, for Fibonacci Flooding, the peak occurs when

$$\Pr\left(\hat{c}_t = f_k\right) \approx \Pr\left(\hat{c}_t = f_{k-1}\right).$$

Remark: In this section, the savings and the mean and variance of information distortion of different flooding schemes are analyzed based on a condition that the signaling network is out-of-band. In such a signaling network, the signaling messages are transmitted in a different physical media than the data traffic, and, consequently, the traffic load of signaling messages does not compete for network resources with the data traffic.

When signaling/flooding information is carried in a same physical media as data traffic, and no special resources are reserved for signaling information, we call the signaling network as an in-band signaling network. For in-band signaling network, signaling traffic competes with data traffic for network resources, such as in Internet where LSAs are transmitted, sharing the same link as IP packets. Usually, a dedicated channel is reserved for in-band signaling, and in this case it has no effect on the available channels for the data traffic. If in-band signaling traffic shares channels with data traffic, then Lazy Flooding has direct significance in saving channels and increasing the total throughput. At present, there are no standards on in-band signaling have been given, and an analysis of the effect of Lazy Flooding schemes on in-band signaling network needs further study. ■

G. Experiment Results for Single Link

We have analyzed the flooding probability of different Lazy Flooding schemes, which reflects the gain of Lazy Flooding: it floods much less than All Flooding. On the other hand, we have estimated the mean and variance of the difference between the true available capacity of a link and its information received by remote nodes through Lazy Flooding, which is an indication of the information discrepancy due to Lazy Flooding.

In this section, we conduct a discrete event simulation to validate the analytical results in the previous section.

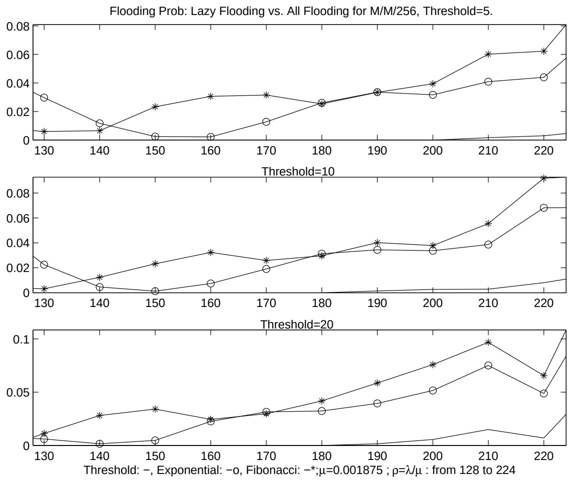

Fig. 6. Simulation Result of Flooding Probability

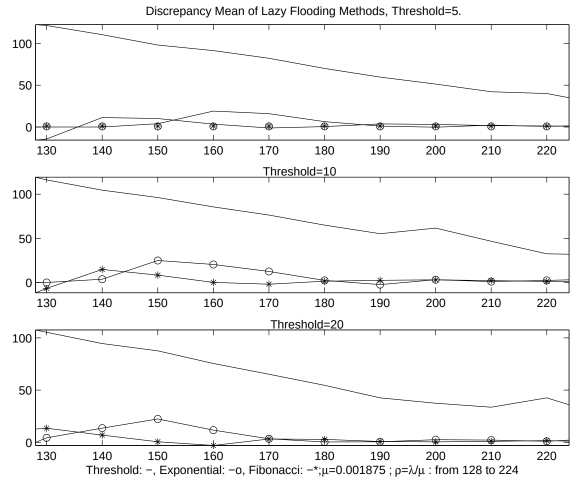

Fig. 7. Simulation Result of Discrepancy Mean

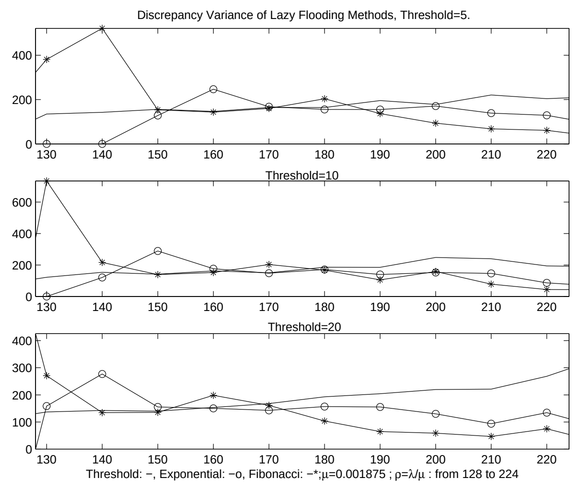

Fig. 8. Simulation Result of Discrepancy Variance

1) Simulation Environment: We consider a link with 256 channels. Requests for channels are generated based on a Poisson distribution, and the duration of each request conforms to an exponential distribution with fixed mean same as the value used in the case study of the theoretical calculation. The traffic load ρ, which is the ratio of the arrival rate and that of the departure rate of the requests, ranges from 128 to 224. The simulation does not take into consideration the flooding information propagation delay; we want to study the discrepancy from the intentional delay from the Lazy Flooding schemes. For all the Lazy Flooding schemes, we set the flooding threshold to be $L = 5, 10, 20$. The simulation begins with an empty link, and the data are collected with 5000 connection set-up and tear-down requests. We mainly consider flooding probability and mean and variance of capacity information discrepancy. We investigate the difference between analytical and experimental results. It turns out that it is negligible. Matlab is used for the simple computation.

2) Single Link Simulation Result: $Flooding\ Probability$ is the ratio of the number of floods generated by the Lazy Flooding schemes over that of All Flooding scheme.
$Discrepancy\ Mean$ is the mean of the capacity information discrepancy, and is estimated in (8) and (17) for Threshold Flooding and Exponential (Fibonacci) Flooding, respectively.
$Discrepancy\ Variance$ is the variance of the capacity information discrepancy, and is estimated in (9) and (18) for Threshold Flooding and Exponential (Fibonacci) Flooding, respectively.

Fig.6 displays the simulation result for flooding probability, Fig.7 and Fig.8 give the simulation result for the mean and variance of discrepancy. Comparing the experimental results in Fig.6, Fig.7 and Fig.8 with that of the analytical results in Fig.2, Fig.4 and Fig.5 respectively, it is clear that they are very close.

III. Simulation Study for Networks

In previous section, we have given an analytical model for flooding schemes in single link scenario. It is difficult to give a close form analytical result for the overall network performance. In this part, we use discrete event simulation to study the impact of different flooding schemes on network performance.

A. Simulation Environment

The network simulation is on a 14 nodes network that has a similar topology as NSFnet in U.S.A. The network is shown in Fig.9. Each link of the two networks is bi-directional and the capacity of each link is 20, i.e., it contains 20 channels. In this simulation, we study the cases with different network load. The requests for a path are generated randomly based on Poisson distribution with specified mean arrival interval. The duration of each path conforms to exponential distribution. In optical network, the rejection rate of a connection request should not be greater than 5%, so the mean of path duration is selected such that network block rate is no more than 5%. The source and destination node of the requested connection are generated randomly from all the nodes in the network. When a path is set up or torn down, link status messages are flooded out if the link capacity in the path satisfies the flooding condition. These messages are supposed to reach all nodes in the network after a constant flooding delay which is $0.5, 1, ..., 5$ times

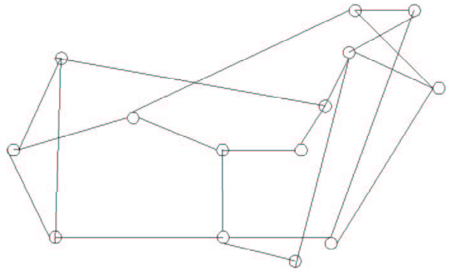

Fig. 9. NSFnet

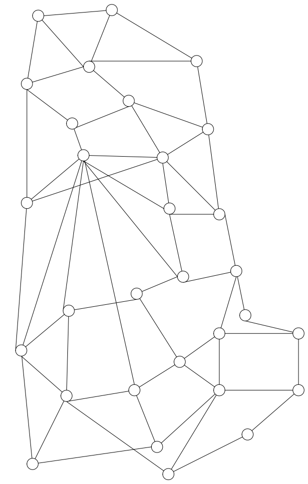

Fig. 10. European Fiber Backbone Structure

of the average arrival interval of the connection requests. For Lazy Flooding schemes, we set the flooding threshold to be $L = 2, 4$. In this simulation, we study 3 parameters which are $network\_block\_rate$, $flood\_frequency$, and $discrepancy\_block$. When the change of these parameters over different flooding delays is studied, the network load is gradually increased to near 80% of the total network capacity. When the change of these parameters over different traffic load is studied, flooding delay of the network is set to be the same as the average arrival interval of the connection requests.

The simulation begins with an empty network, and the data are collected after 3,000 connection set up and tear down requests have been dealt with and the network has accommodated about 260 connections. In optical network, the rejection rate of a connection request should not be greater than 5%, the simulation stops when $800 \times 5\% = 40$ connections are rejected.

Lazy Flooding techniques may affect the blocking rate when a network becomes large. To study such impact we conduct simulation on a 30 nodes network that has the same topology as a European Fiber Backbone[17], as shown in Fig.10. Each link in Fig.10 has 40 channels.

$Network\_block\_rate = \frac{R}{N}$ where N is the number of connection requests to the network and R is the number of connections rejected.

$discrepancy\_block$ is the block rate resulted from inaccurate network information, that is, the discrepancy in the available link capacity: (1) A path is computed for a connection request, but one or more links on the path have no capacity to accommodate such a connection, resulting in a false path, and, consequently,the connection is rejected; (2) the network has a path for a connection request, but based on the inaccurate database information, no path is found for it, and, consequently, the connection request is rejected. Both cases are the consequences of the discrepancy in network link status information.

$Flooding\_frequency$ is the ratio of the total number of floods generated by the Lazy Flooding schemes in the whole network over that of All Flooding scheme. It measures the reduction in floods using Lazy Flooding.

B. Routing Algorithms

Two routing algorithms are used in the simulation and both are based on the shortest path algorithms. The difference is the weight assignment: one is insensitive to the link load and the other is.

- **Algorithm 1:** Every link is assigned $weight = 1$. We find shortest paths using Dijkstra's algorithm.
- **Algorithm 2:** Every link is assigned $weight = (2 \times H + 1)^{\alpha}$ where $\alpha = 1 - \frac{C_t}{B}$ is the load of the link and H is the network diameter. Hence the link weight increases exponentially with its traffic load. We find shortest paths using Dijkstra's algorithm.

C. Simulation Results for Algorithm 1

Algorithm 1 is a topology driven algorithm, which calculates a path only on the basis of the network topology and does not vary according to the network resource. A flooded message has impact on the routing decision only when the link capacity reaches zero, while every flooding scheme floods out the message at this case, so the four flooding schemes have identical network block rate and discrepancy blocking rate when algorithm 1 is used as the routing algorithm.

We have simulated the case for Lazy Flooding when the system "enters", "leaves", and both "enters" and "leaves" a predetermined capacity. The flooding frequency for Lazy Flooding when the system "enters" or "leaves" are similar and less than both "enters" and "leaves" case; while the three policies do not have much difference. Figure 11 shows the flooding frequencies of the "leaves" methods when algorithm 1 is used in the network. The x-axis represents the load of the network, which is $180, 200, ..., 320$ Erlang, and under such a network load, the network block rate is no more than 5%. The y-axis is the flooding frequency of the Lazy Flooding schemes comparing with All Flooding. Note that this flooding frequency is normalized with All Flooding scheme as defined by $Flooding\_frequency$ in section III-A. This figure illustrates how much flooding traffic can be saved by Lazy Flooding schemes and how the saving to be reduced with traffic load increases.

D. Simulation Results for Algorithm 2

The following subsections $Network\_block\_rate$ and $discrepancy\_block$ rate are illustrated when algorithm 2 is used. The changes of these parameters over different flooding

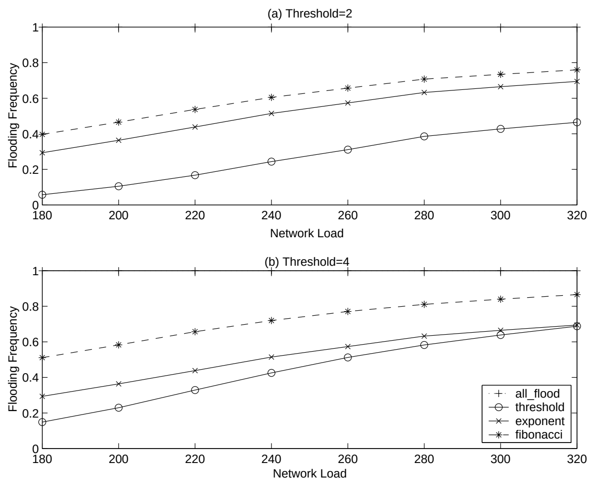

Fig. 11. Algorithm1: Flooding Frequency vs. Network Load, (a)Threshold=2. (b)Threshold=4

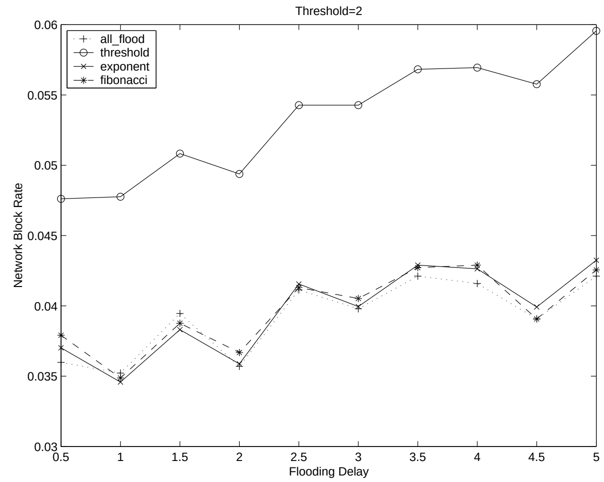

Fig. 12. Algorithm2: Network Blocking Rate vs. Flooding Delay of Fig.9

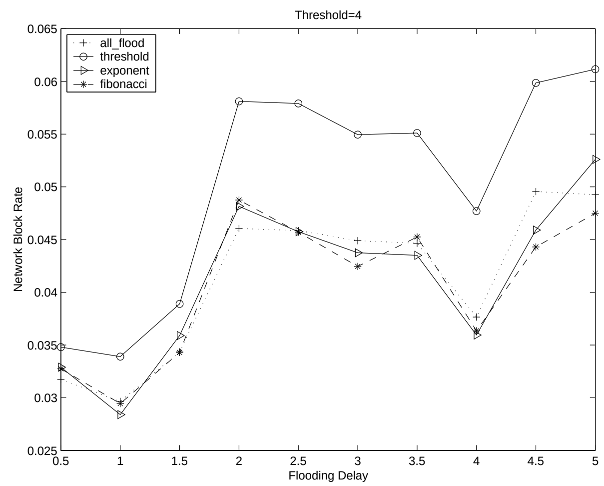

Fig. 13. Algorithm2: Network Blocking Rate vs. Flooding Delay of Fig.10

delay and over different network load are all given. To study if Lazy Flooding has more impact on these parameters in larger sized network, simulation result on network Fig.10 is also reported.

1) Network Blocking Rate For Algorithm 2: This section shows the simulation results for network blocking rate when algorithm 2 is used. For a connection request, on the basis of the information database, a route is calculated. If every link in the route has enough capacity, this connection is established and resources acquired, otherwise the connection is blocked.

Fig.12 shows that the more information is flooded, the less network blocking rate is. The x-axis represents the flooding delay of the messages, which is $0.5, 1, ..., 5$ times of the average arrival interval of the connection requests, and the y-axis is the network blocking rate. Network load is 250 Erlang under different flooding delay such that network block rate is not higher than 5%. Flooding delay reflects the network convergence time and Fig.12 also shows that for any flooding scheme, the longer convergence time the higher blocking rate.

Fig.13 is the network blocking rate vs. flooding delay get on network Fig.10. Network load is specified at 500 Erlang under different flooding delay. Fig.13 shows that when network becomes large, the effect of different flooding schemes is the same as a small network. The behavior of Lazy Flooding does not change with the scale of network. In a larger-sized network, flooding delay is longer than that in a small network, but such a delay has the same impact on Lazy Flooding and All Flooding. So comparing with All Flooding, the impact of Lazy Flooding on network block rate in a large network does not increase with the scale of network.

Remark: When algorithm 2 is used as the routing algorithm in the network, for most of the time, the more accurate the network status is known, the less network blocking rate is.

With slightly more floods Exponential and Fibonacci Flooding have much lower network blocking rate than Threshold Flooding. On the other hand, comparing with All Flooding, they have much less floods but almost the same network blocking rate. ■

2) Simulation Results for Discrepancy Blocking Rate: In an optical network, a connection request can be blocked in two phases: path selection and path allocation. Inaccurate network resource information can misguide a path selection decision to increase network block rate. $Discrepancy\_block$ is the block rate resulted by inaccurate network information, and reflects how much Lazy Flooding schemes impact on network blocking rate. The following figures show the simulation results for the $discrepancy\_block$ rate when algorithm 2 is used.

Fig.14 and Fig.15 gives the discrepancy block resulted by different flooding schemes on Fig.9 and Fig.10. Flooding delay in a network has impact on any flooding schemes, even All Flooding scheme can result in discrepancy blocking. When a network becomes larger, discrepancy block resulted by Lazy Flooding does not increase any further.

Remark: When algorithm 2 is used as the routing algorithm in a network, Threshold Flooding has the most $discrepancy\_block$ among all the flooding schemes.

Comparing with All Flooding, Exponential and Fibonacci Flooding have much less floods but almost the same discrepancy blocking. ■

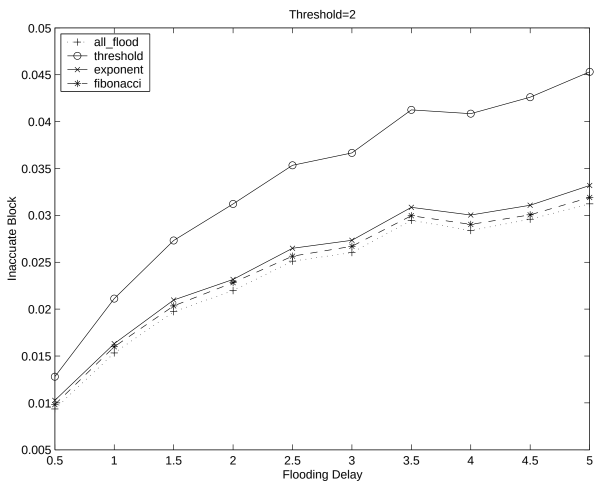

Fig. 14. Algorithm2: Discrepancy Block vs. Flooding Delay of Fig.9.

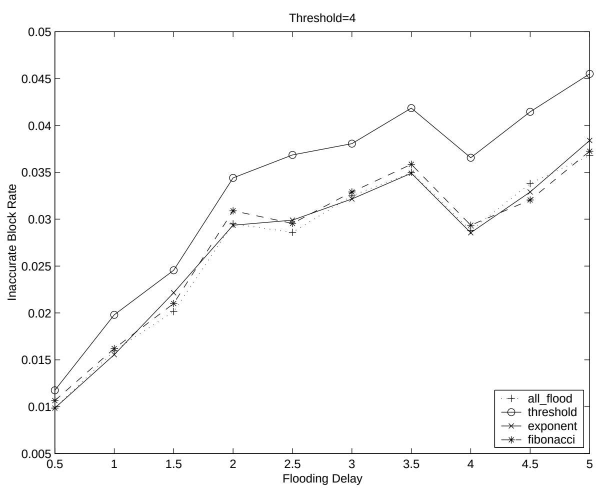

Fig. 15. Algorithm2: Discrepancy Block vs. Flooding Delay of Fig.10.

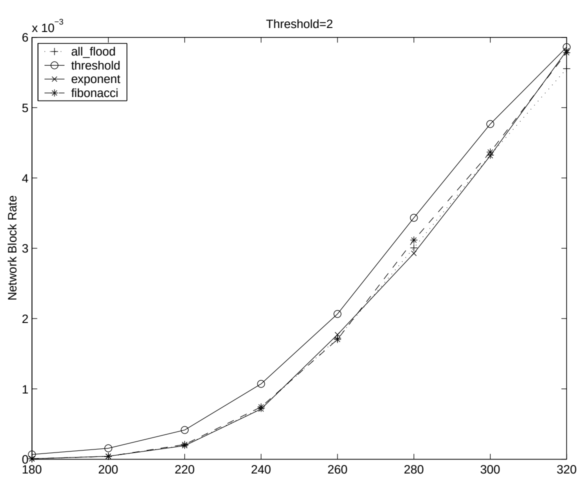

Fig. 16. Algorithm2: Network Blocking Rate vs. Network Load of Fig.9

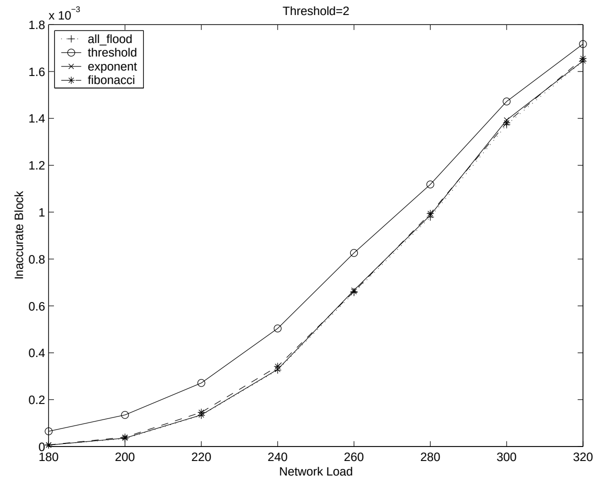

Fig. 17. Algorithm2: Discrepancy Block vs. Network Load of Fig.9.

Both network block rate and discrepancy block rate have been studied when network load varies from 180 to 320 in Fig.9 and 700 to 1000 in Fig.10. As shown in Fig.17 and Fig.16, remarks for network block rate and discrepancy block rate are all held when x-axis is network load instead of flooding delay.

IV. Conclusion

We have studied three Lazy Flooding methods: Threshold, Exponential, and Fibonacci. They significantly reduce the number of floods in the DCN networks. On the other hand, our analysis and simulation show that Lazy Flooding has negligible effect on network performances in terms of link and network blocking rate. Which one is the best among the three Lazy Flooding schemes depends on user's motivation and criteria.

Apparently, more floods provide more accurate network and resource information and would naturally lead to better utilization of the network resources and thus result in less link and network blocking. Some of our preliminary simulations exhibit abnormal behaviors of the network for certain routing algorithms, which are based on the shortest paths and on the weights on the links, which are sensitive to the traffic load. More specifically, we have occasionally observed that more floods lead to more link and network blocking rate. This has also been observed in the Internet study. Lightpath or Route flapping could be an explanation. However, for all-optical networks, further study is needed to understand the problem.

Overall, what is the best lazy flooding technique? There is no simple answer. Different applications have different constraints and requirements on the network. The main motivation of our work is the DCN signaling OSPF network (for Lucent Lambda Router All-optical Network) response time that should be below 50ms. Given this constraint, All Flooding is not appropriate; due to the large number of LSAs flooded in the network, one cannot guarantee the 50ms response time. Threshold Flooding seems to work well. However, we do not have the link status information for path/link protection computation. That has led us to Exponential Flooding. Based on the mathematical analysis and simulation, it significantly cuts down the flooding traffic and still provides needed link status information. However, this is a rather special application and it would be misleading to the readers if we conclude that Exponential Flooding is the best under all the circumstances. Therefore, we propose several easy-to-implement Lazy Flooding techniques with both mathematical analysis and simulation results, showing that they cut down the flood traffic and maintain link status information. We believe that our techniques cover a range of applications. If none of our approaches are applicable for a particular application, one can design his own lazy flooding techniques, conduct mathematical analysis and experiments, and determine which one is the best, taking our approaches as a guide or reference.

ACKNOWLEDGEMENT

We take this opportunity to thank all those who have reviewed this paper. Their valuable comments have helped to enhance and polish this paper.

REFERENCES

[1] A.Aho and D.Lee, "Hierarchical Networks and the LSA N-Squared Problem in OSPF Routing," *GLOBECOM 2000*, Nov. 27 - Dec. 1, San Francisco, CA

[2] J.T.Moy, RFC 1245, "OSPF Protocol Analysis," July, 1991.

[3] R. Guerin and A. Orda, "QoS Routing in Networks with Inaccurate Information: Theory and Algorithms," *IEEE/ACM Trans. on Networking*, Vol. 7, No. 3, June 1999.

[4] G. Liu, C. Ji and V. Chen, "Network Management Information for Light Path Assessment: Trade-off between Performance and Complexity," INFOCOM, Sanfrancisco, March, 2003.

[5] Jochen Behrens, J.J.Garcia-Luna-Aceves, "Distributed,Scalable Routing Based on Link-State Vectors," in *Proc.ACM SIGCOMM 94*, London, UK, Aug.1994.

[6] H. Fujinoki and K.J.Kristensen , "The Directed Reverse Path Join (DRP) Protocol: An Efficient Multicast Routing Protocol," *Computer Communications*, V24, 12, Jul. 15, 2001, p1121-1133.

[7] K. Carlberg and J. Crowcroft, "Building Shared Trees Using a One-to-Many Joining Mechanism," *Computer Communication Review*, 27 (1) January (1997) 5 - 11

[8] S. Deering and D. Cheriton, "Multicast Routing in Datagram Internetworks and Extended LANs," *ACM Transactions on Computer Systems*, 8 (2) May (1990) 85 - 110.

[9] S.J. Lee, W. Su, M.Gerla, "Wireless Ad Hoc Multicast Routing with Mobility Prediction," *Mobile Networks and Applications*, V6,4, 2001, p351-360.

[10] Hyojun Lim, Chongkwon Kim, "Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks," *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, 2000, p61-68.

[11] G. Apostolopoulos, R. Guerin, S. Kamat and S. Tripathi, "Quality-of-Service Based Routing: A Performance Perspective, " *ACM SIGCOMM*, August 1998.

[12] A. Shaikh, J. Rexford and K. G. Shin, "Evaluating the Impact of Stale Link State on Quality-of-Service Routing," *IEEE Transactions on Networking*, Vol. 9, April 2001.

[13] "Optical Network Management Protocol," *Bell Labs Tech Memo.* August 2001.

[14] D.Bertsekas and R.Gallager, *Data Networks*, 2nd ed., Englewood Cliffs, NJ:Prentice Hall, 1992.

[15] D.G. Kendall,"Some problems in the theory of queues," *Journal of Royal Statistical Socienty*, Serial B, p151-185, Vol13, 1951.

[16] L. Kleinrock (1975),*Queueing systems, Volume 1: Theory,* p105-106, John Wiley & Sons.

[17] Wayne D. Grover, *Mesh-Based Survivable Networks: options and strategies for optical, MPLS, SONET and ATM Network*, PRENTICE HALL PTR, UPPER SADDLE RIVER, NJ07458.