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

C. Chi, D. Huang, D. Lee and X. Sun

September 12, 2001

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 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 reduces the data communication network congestion. On the other hand, it has negligible effect on network performance.

1 Introduction

In a distributed network such as Internet and all optical network, resource and network topology information have 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 messages (LSA's) to be flooded, causing serious problems for the network stability and scalability. Various techniques have been proposed to cope with this well-known *LSA-squared problem* [1]. Note that LSA's are flooded only when there is a topology update and that nodes do not flood information of the adjacent link load status.

In an all optical network such as Lucent Lambda Router Network [2], 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 all the involved channel status is changed from being available to occupied and vice versa, 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, leading to the network instability and also making it difficult to scale up. As a matter of fact, as lightpaths are set up and torn down, a huge volume of information is flooded while OSPF does not flood the link load update information. Therefore, the congestion caused by flooding in the DCN of an all optical network is even worse than Internet.

To cope with this optical networks DCN congestion we propose a new technique: lazy flooding. It significantly reduces the number of flooded messages for the network resource and topology information update without affecting the lightpath construction and, consequently, path protection and restoration. The idea is simple. Suppose that there are k channels in each 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 doe not flood for each channel update, and we call it *lazy flooding*. Different flooding decisions give different lazy flooding methods, and we study three in this paper:

(1) Threshold Flooding. For a pre-determined threshold value $0 < T \leq k$, whenever the available number of channels in a link is less than T, the number of the available channels is flooded after a channel update. Otherwise, there is no flooding.

(2) Geometric Flooding. For a geometric sequence of numbers above the threshold: $0 < T \leq g_1 < g_2 < \cdots < g_r \leq k$, the link status information is flooded whenever the available number of channels in the link is less than T or is 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 link status information is flooded whenever the available number of channels in the link is less than T or is 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 now discuss informally the cost, i.e., the possible disadvantages of not flooding with each link status update:

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

(2) Blocking Network. 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 algorithms assign a weight on each link for the computation, and the weight is determined by the link load [3]. Similarly, certain lightpath computation algorithms also use shortest path algorithms and assign a weight on each link, which is typically the occupied channels. The rational is: avoid routing through a link where there are almost no channels available. Obviously, lazy flooding does not keep this information updated for each OXC on the network, and Threshold Flooding is the worst. The other two methods 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 Geometric Flooding when the available channel number is low and the channel status is more sensitive to the link and network blocking. Our analysis and simulation will show that both Fibonacci and Geometric 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 do 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 out-performs the case when each link status update is flooded. This anomaly has also been observed with the Internet performance.

Section 2 contains mathematical analysis of various flooding techniques. Section 3 reports the simulation results of two different routing algorithms: one is insensitive to the link channel status and the other is.

2 Mathematical Analysis

We first analyze different lazy flooding techniques on a single link. We consider a discrete time and discrete state communication link with a total bandwidth B. Assume that the available capacity $\{c_t, t = 0, 1, 2, ...\}$ follows a Markov chain (MC) with the transition probability

$$\Pr\left(c_{t+1} = j | c_t = i\right) = p_{ij}, \quad T = [p_{ij}]_{i,j=0,1,...,B} \,.$$

We also assume that this MC is Ergodic.

The all-flooding scheme has the flooding probability $p_{ij} = 1$ for $|i-j| = 1$, $i, j = 0, 1, \cdots, B$. To show the gain of a particular lazy flooding scheme, we calculate its flooding probability when the MC achieve stationary states.

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.

2.1 Flooding Methods

A straightforward method, which is used in Internet OSPF, is all flooding: the link capacity is flooded whenever there is a change. Lazy flooding is different; it only floods for each link capacity change when the capacity is below a threshold value: $0 < L \leq B$. Specifically, the link capacity $c_t \leq L$ is flooded whenever the capacity has a change. If the link capacity $c_t > L$, it is not always flooded for each link capacity change. There are three different techniques, each of which has its own merit for different optical networks and with different network loads.

Threshold Flooding There is no flooding if the available link capacity $c_t > L$. Otherwise, it floods whenever there is a link capacity change.

Exponential Flooding

Let

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

where $K = \lfloor \log_2 (B - L) \rfloor + L$, $\lfloor x \rfloor$ denotes the largest integer no more than x. It floods if and only if a link capacity changes from a value b_k.

Fibonacci Flooding Let

$$f_k = k,\ k = 0, 1, ..., L, \quad f_k = f_{k-1} + f_{k-2} - L + 3,\ k = L+1, L+2, ...K,$$

where K is the largest index k such that $f_k \leq B$. It floods if and only if a link capacity changes a value from f_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.

2.2 Death and Birth Service Loss Model and its Stationary Distribution

A widely used Markov chain model in communications is M/M/B with possible loss when the link is full. Using the capacity c_t instead of loading, which corresponds to the number of customers, we have the transition probability:

$$p_{ij} = \begin{cases} \lambda & \text{j=i-1} \geq 0, \\ 1 - \lambda - (B - i)\,\mu & \text{i=j} \neq 0, \\ 1 - B\mu & \text{i=j=0}, \\ (B - i)\,\mu & \text{j=i+1} \leq B, \\ 0, & \text{otherwise.} \end{cases} \qquad i, j = 0, 1, 2, ..., B.$$

Let $\rho = \frac{\lambda}{\mu}$, it follows from the formula (3.45) in [1] that the stationary distribution is given by the following:

$$\begin{aligned} \pi_B &= \frac{1}{\sum_{k=0}^{n} \frac{\rho^k}{k!}}; \\ \pi_k &= \frac{\rho^{B-k}}{(B-k)!}\pi_B,\ k = 0, ..., B-1. \end{aligned}$$

This is the so-called Earlang distribution. For a large B, this distribution can be approximated by a Poisson or a normal pdf with mean $(B - \rho)$ and variance ρ.

2.3 Probabilities for Different Flooding Techniques

Let $0 \leq L \leq B$. When $c_t \leq L$, we do the flooding if there is a change in capacity. When $c_t > L$, we consider the following three policies:

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

$$P_t = \Pr\left[c_{t+1} \neq c_t, c_t \leq L\right]$$

(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

$$P_e = \Pr\left\{\cup_k \left[c_t = b_k, c_{t+1} \neq b_k\right]\right\}.$$

(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 $f_k \leq B$. So, the flooding probability is

$$P_f = \Pr\left\{\cup_k \left[c_t = f_k, c_{t+1} \neq f_k\right]\right\}.$$

We calculate these probabilities by finding: (i) Stationary distribution of the model; and (ii) Probabilities of the interested events on the basis of the stationary distribution and transition probabilities of the model.

Threshold Flooding Probability

For this model, it is easy to see that the flooding probability is

$$P_t = \Pr\{c_{t+1} \neq c_t, c_t \leq L\} = B\mu\pi_0 + \sum_{k=1}^{L} [(B-k)\mu + \lambda]\pi_k.$$

Exponential Flooding Probability

For the exponential flooding policy, we have

$$\Pr\{\cup_k [c_t = b_k, c_{t+1} = b_k - 1]\} = \lambda \sum_{k=1}^{K} \pi_{b_k},$$

and

$$\Pr\{\cup_k [c_t = b_k, c_{t+1} = b_k + 1]\} = \mu \sum_{k=0}^{K} (B - b_k)\pi_{b_k}.$$

So,

$$P_e = B\mu\pi_0 + \sum_{k=1}^{K} [\lambda + \mu(B - b_k)]\pi_{b_k}.$$

Fibonacci flooding probability

Similar to Exponential flooding, we have

$$\Pr\{\cup_k [c_t = f_k, c_{t+1} = f_k - 1]\} = \lambda \sum_{k=1}^{K} \pi_{f_k};$$

and

$$\Pr\{\cup_k [c_t = f_k, c_{t+1} = f_k + 1]\} = \mu \sum_{k=0}^{K} (B - f_k)\pi_{f_k}.$$

So,

$$P_f = B\mu\pi_0 + \sum_{k=1}^{K} [\lambda + \mu(B - f_k + 1)]\pi_{f_k}.$$

2.4 A Comparisons

Figure 1 shows the flooding probabilities (the benefit) of the three flooding polices over different model parameter $\rho = \frac{\lambda}{\mu}$ and threshold L. For all-flooding, i.e., flooding whenever, the flooding probability is always one.

It is clear that the larger the threshold L is, the more the flooding will be. Also, recall that approximately the mean of capacity is $(B - \rho)$, if the ρ is large, the available channel capacity will be small, and that results in more flooding. Overall the three lazy flooding methods have a similar flood probability and are much smaller than the all-flooding method.

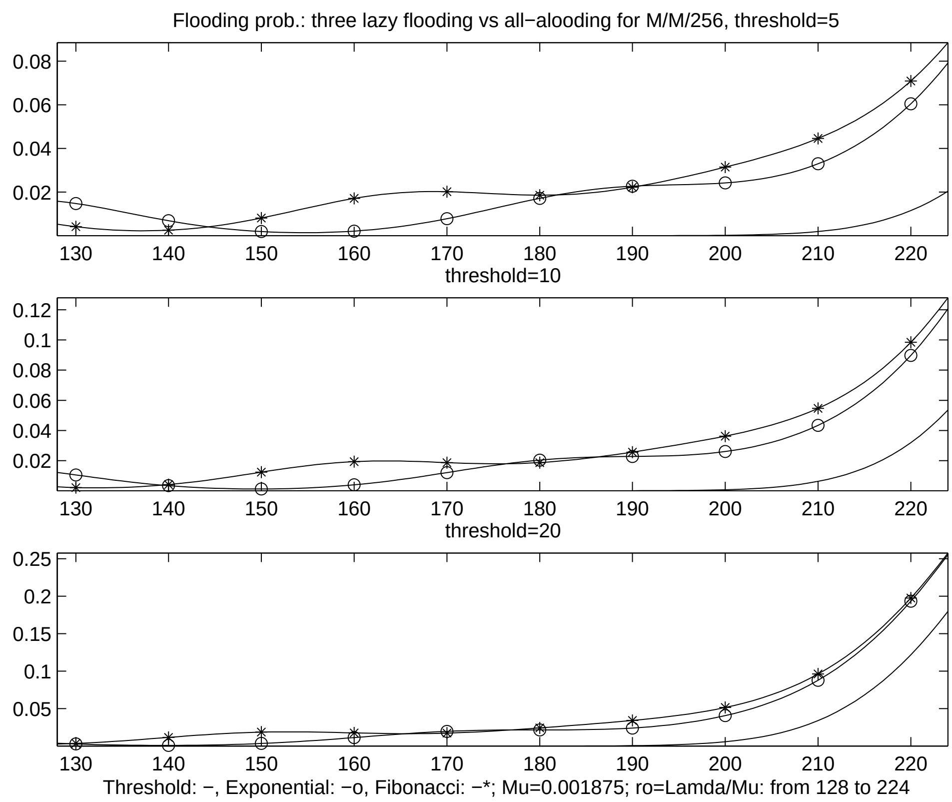

Figure 1: Flooding probabilities

2.5 Mean and Variance of Information Distortion

We now estimate the cost of the three lazy flooding methods that is measured by the difference of the true capacity and the flooded information, denoted by D_t. Since the load on a link is random, so is D_t. On the basis of the stationary distribution of c_t we can find the corresponding distribution of D_t for different flooding methods, and then the mean and variance of D_t.

2.5.1 Threshold flooding

We have

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

So,

$$\begin{aligned} E(D_t) &= \sum_{k=L+1}^{B} (k-L)\,\pi_k; \\ Var(D_t) &= \sum_{k=L+1}^{B} \left[k - L - E(c_t)\right]^2 \pi_k + \sum_{k=0}^{L} \left[E(c_t)\right]^2 \pi_k. \end{aligned}$$

2.5.2 Stationary Distribution of Augmented Vector Process

For exponential and Fibonacci flooding, the situation is more complicated. Since the calculations of the stationary distributions of the two schemes are similar, we only show how to derive the formulae for exponential flooding here.

Let $\{\hat{c}_t, t = 0, 1, ...\}$ be the known available capacity at time t on the link by remote nodes when the exponential flooding is used. Then $\hat{c}_t = b_k$ if and only if there is an integer j such that $c_{t-j} = b_k$ and $b_{k-1} < c_s < b_{k+1}$ for all $t - j < s \le t$.

It is clear that $\{\hat{c}_t, t = 0, 1, ...\}$ is *not* a Markov Chain. We have to consider its augmented vector process $X_t = [c_t, \hat{c}_t]$ that is a 2-dimensional MC. Its transit probabilities are given by:

For $b_{k-1} + 1 < j < b_{k+1} - 1$,

$$\begin{aligned} \Pr(c_t = j+1, \hat{c}_t = b_k | c_{t-1} = j, \hat{c}_{t-1} = b_k) &= (B-j)\,\mu; \\ \Pr(c_t = j, \hat{c}_t = b_k | c_{t-1} = j, \hat{c}_{t-1} = b_k) &= 1 - \lambda - (B-j)\,\mu; \\ \Pr(c_t = j-1, \hat{c}_t = b_k | c_{t-1} = j, \hat{c}_{t-1} = b_k) &= \lambda. \end{aligned}$$

For $j \le b_{k-1} + 1$,

$$\Pr(c_t = j-1, \hat{c}_t = b_k | c_{t-1} = j, \hat{c}_{t-1} = b_k) = 0.$$

For $j \ge b_{k+1} - 1$,

$$\Pr(c_t = j+1, \hat{c}_t = b_k | c_{t-1} = j, \hat{c}_{t-1} = b_k) = 0.$$

We also have initial conditions:

$$\begin{aligned} \Pr\left(c_t = j, \hat{c}_t = b_k\right) &= 0, \quad j \le b_{k-1} or j \ge b_{k+1}; \\ \Pr\left(c_t = b_k, \hat{c}_t = b_k\right) &= \Pr\left(c_t = b_k\right) = \pi_{b_k}. \end{aligned}$$

Thus, let $p_{k,j} = \lim_{t\to\infty} \Pr\left(c_t = j, \hat{c}_t = b_k\right)$, $b_{k-1} < j < b_{k+1}$, we have the following system of two equations:

$$\begin{aligned} p_{k,j} &= (B-j+1)\,\mu p_{k,j-1} + \left[1-\lambda-(B-j)\,\mu\right] p_{k,j} + \lambda p_{k,j+1},\ b_{k-1} < j < b_k; \\ p_{k,b_{k-1}} &= 0, \qquad p_{k,b_k} = \pi_{b_k}; \end{aligned} \tag{1}$$

and

$$\begin{aligned} p_{k,j} &= (B-j+1)\,\mu p_{k,j-1} + \left[1-\lambda-(B-j)\,\mu\right] p_{k,j} + \lambda p_{k,j+1},\ b_k < j < b_{k+1}; \\ p_{k,b_k} &= \pi_{b_k}, \qquad p_{k,b_{k+1}} = 0. \end{aligned} \tag{2}$$

2.5.3 A Closed Form for the Joint Distribution

To solve the equation system (1) and (2), we need the following lemma:

Lemma: For $b_{k-1}+1 < j < b_k$, we have

$$p_{k,j} = \left(1 + \sum_{l=1}^{j-b_{k-1}-1} \prod_{m=1}^{l} \frac{B-j+m}{\rho}\right) p_{k,b_{k-1}+1} \tag{3}$$

To use this lemma, first we need to calculate $p_{k,b_{k-1}+1}$. Since

$$\pi_{b_k} = p_{k,b_k} = \left[1 + \sum_{l=1}^{b_k-b_{k-1}-1} \prod_{m=1}^{l} \frac{B-b_k+m}{\rho}\right] p_{k,b_{k-1}+1},$$

we have

$$p_{k,b_{k-1}+1} = \frac{\pi_{b_k}}{1+\sum_{l=1}^{b_k-b_{k-1}-1}\prod_{m=1}^{l}\frac{B-b_k+m}{\rho}}.$$

Then, substituting the above formula into (3), we have

$$p_{k,j} = \frac{1+\sum_{L=0}^{j-b_k-1}\prod_{m=1}^{L}\frac{B-j+m}{\rho}}{1+\sum_{L=0}^{b_{k+1}-b_k-1}\prod_{m=1}^{L}\frac{B-b_k+m}{\rho}}, \quad b_{k-1} < j < b_k.$$

Finally, for $b_k < j < b_{k+1}$, we have

$$\begin{aligned} p_{k+1,j} &= \pi_j - p_{k,j} \\ &= \pi_j - \frac{1+\sum_{L=0}^{j-b_k-1}\prod_{m=1}^{L}\frac{B-j+m}{\rho}}{1+\sum_{L=0}^{b_{k+1}-b_k-1}\prod_{m=1}^{L}\frac{B-b_k+m}{\rho}}\pi_{b_{k+1}}. \end{aligned}$$

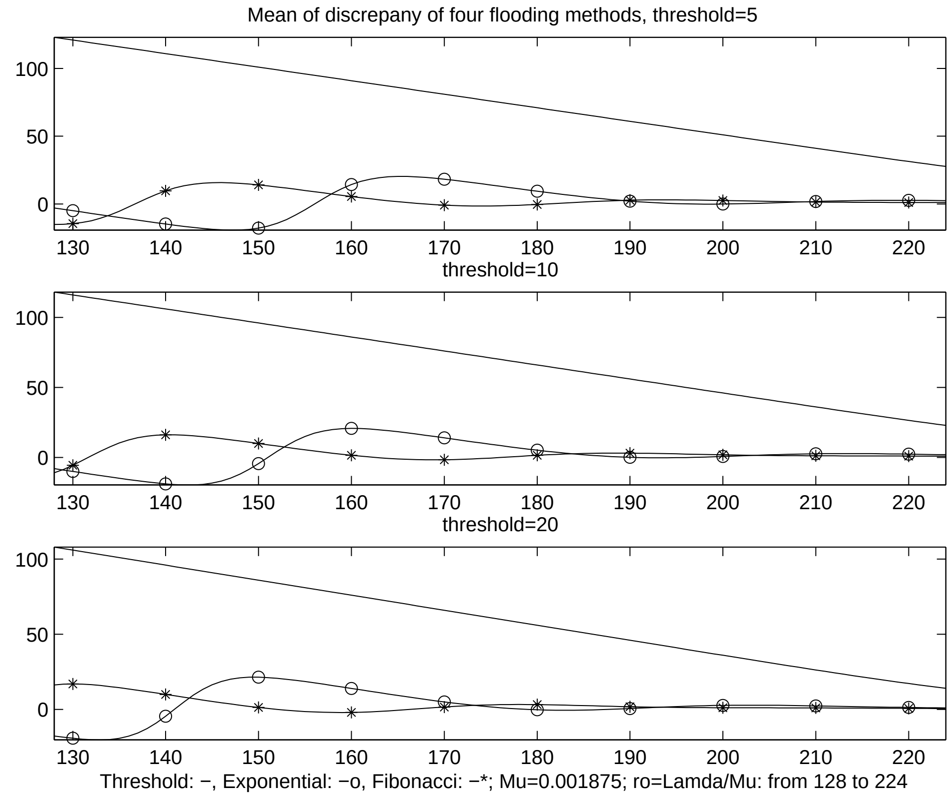

Figure 2: Mean of the difference

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

$$E(c_t - \hat{c}_t) = \sum_{k=L}^{K} \sum_{j=b_{k-1}+1}^{b_{k+1}-1} (j - b_k)\, p_{k,j}, \tag{4}$$

$$Var(c_t - \hat{c}_t) = \sum_{k=L}^{K} \sum_{j=b_{k-1}+1}^{b_{k+1}-1} \left[j - b_k - E(c_t - \hat{c}_t)\right]^2 p_{k,j}. \tag{5}$$

For Fibonacci flooding, we only need to replace b_k by f_k in the formulae.

2.5.4 A Comparison and Analysis

We plot the mean and variance of the three lazy flooding methods with different threshold l against different $M/M/B$ model parameter ρ. For all three cases, the parameter μ is fixed. See Fig.2 and Fig.3 below.

Since $\rho = \frac{\lambda}{\mu}$, larger the ρ is, less the capacity will be, then we have more flooding. So it is not surprising that for large ρ, the mean and variance of the distortion 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 then decreases against ρ. This is because as ρ increases, the stationary distribution $\{\pi_k\}$

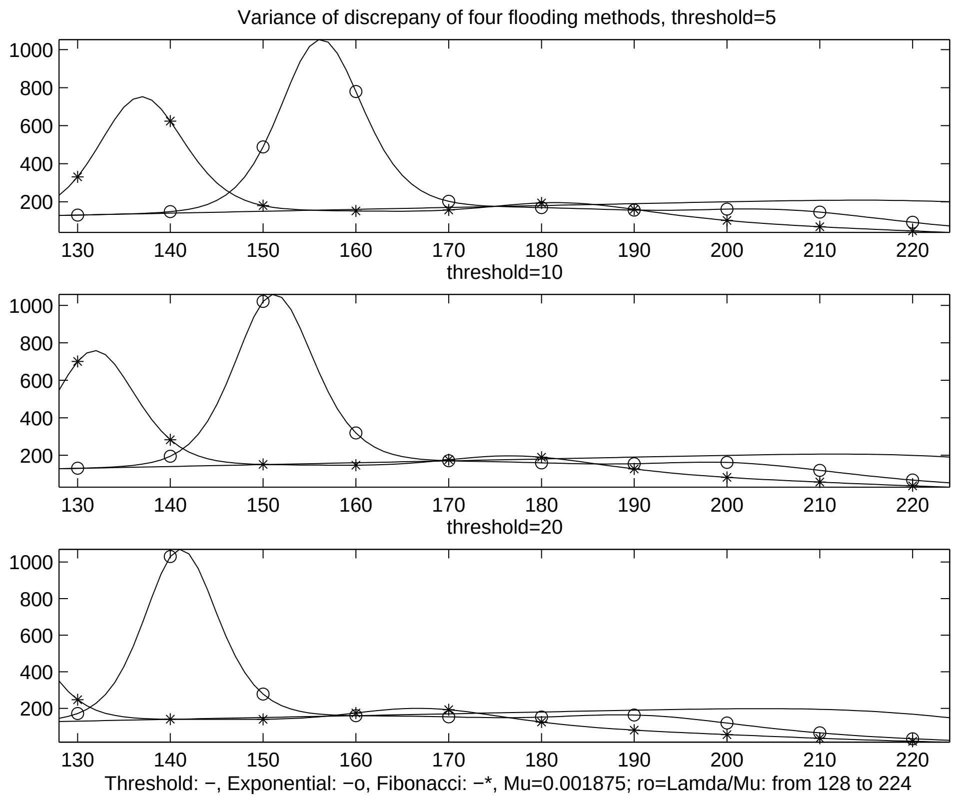

Figure 3: Variance of the difference

will concentrate on the lower index end, i.e., the most possible available capacity is small. Then we flood more frequently. This reduces both mean and variance. There are rapid changes in both means and variances for exponential flooding and Fibonacci flooding. This is because in the left and right hand sides of the peaks, $\hat{c}_t$ equals to different values with a very large probability. So the variance is not too large. However, for exponential flooding when the value ρ makes

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

the variance will be large. Similarly, for Fibonacci flooding, the peak happens when

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

3 Simulation Study for Flooding Schemes

In previous section, we have analysized 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 on a link and the information received by remote nodes through lazy flooding, which is an indication of the information discrepancy due to lazy flooding; it turns out that it is negligible.

For the overall network performance, we conduct a simulation study. We want to estimate the link and network blocking rates when all-flooding and lazy flooding schemes are used.

3.1 Routing Algorithms

In this simulation, two routing algorithms are used 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 not.

- **Algorithm 1:** Every link is assigned $weight = 1$, then we find a path with the least number of hops by Dijkstra shortest path 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. Then we find a path, in which every link weight exponentially increases with its traffic load, by Dijkstra shortest path algorithm.

3.2 Simulation Environment

The simulation is based on a given 14 nodes network which has a similiar topology as NSFnet in U.S.A. The network is shown in Fig.4, each link is bi-directional and the capacity of each link is 20. In this simulation, we study the cases with different network load. When a path is set up or torn down, network status messages are flooded out if the

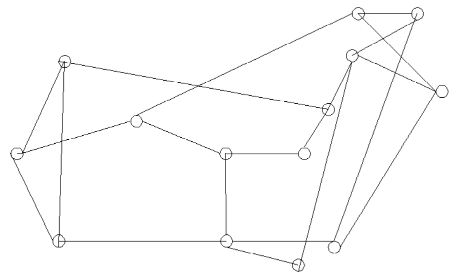

Figure 4: NFSnet

link capacity in the path satisfies the flooding condition. The delay of these messages is constant in each simulated case, which is $0.5, 1, ..., 5$ times 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 4 parameters which are $link\_block\_rate$, $network\_block\_rate$, $flood\_frequency$, and $inaccurate\_block$. When the change of these parameters over different flooding delay 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. The simulation stops and re-runs when 40 connections are rejected.

$Link\_block\_rate = \frac{\sum_{i=1}^{M} R_i}{\sum_{i=1}^{M} C_i}$, where M is the total number of links in the network, R_i is the number of connections which are rejected because link i has no channels, C_i is the total capacity of link i. In the simulation, R_i varies with different flooding schemes and routing algorithms.

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

$Flooding\_frequency$ is the ratio of the number of floods generated by the lazy flooding schemes comparing with that of all-flooding scheme in the sample period.

$Inaccurate\_block$ is the block rate resulted by inaccurate network information, which includes two parts: a path is calculated for a connection request, but the network has no capacity to admit such a connection, so the path is in fact a false one, and the connection is rejected in the end; the network has a path for a connection request, but based on the database information, no path is found for it, so the connection is rejected. The two cases are all resulted by the inaccurate network status information, and the $Inaccurate\_block$ is the rate of the connection requsts that are rejected because of above reasons.

3.3 Simulation Results for Link Blocking Rate

The following subsections show the simulation results for link blocking rate when algorithm 1, algorithm 2 are used. Due to flooding delay or lazy flooding, there are still some

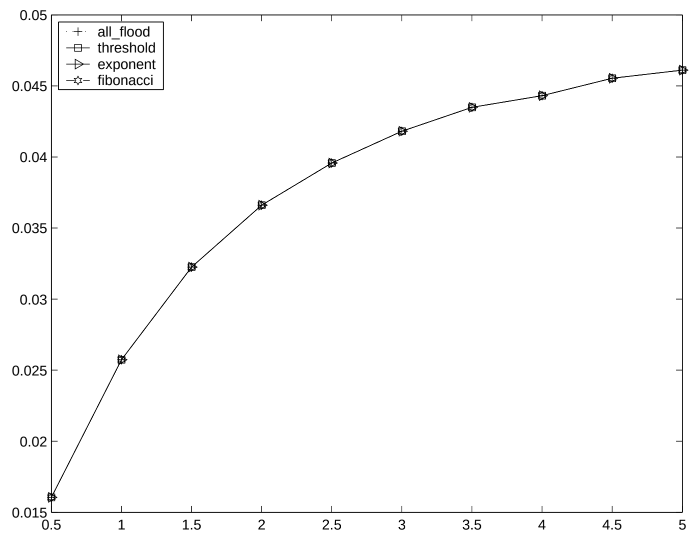

Figure 5: Algorithm1:Link Blocking Rate vs. Flood Delay, Threshold=2

connection requests asking for link (u, v) which has no capacity any more. According to the information in the database that there are channels still available on (u, v) and choose to route via it, and this connection is blocked in the end.

In the figures below, the x-axis represents the flooding delay of the messages, while the y-axis is the link blocking rate of the network.

3.3.1 Link Blocking Rate for Algorithm 1

Fig.5 and Fig.6 show the mean of the link blocking rate over average network flooding delay, which is $0.5, 1, ..., 5$ times of the average arrival interval of the connection requests and threshold are set to 2 and 4, respectively. In this figure, the four curves are identical. So we have:

Remark 1 *When algorithm 1 is used as the routing algorithm in a network, the link blocking rate is the same no matter what flooding scheme is used to advertise the network status.*

Exponential and Fibonacci flooding have much less floods than all-flooding but with the same link blocking rate.

The main reason is: 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. In another word, 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 link blocking rate when algorithm 1 is used as the routing algorithm.

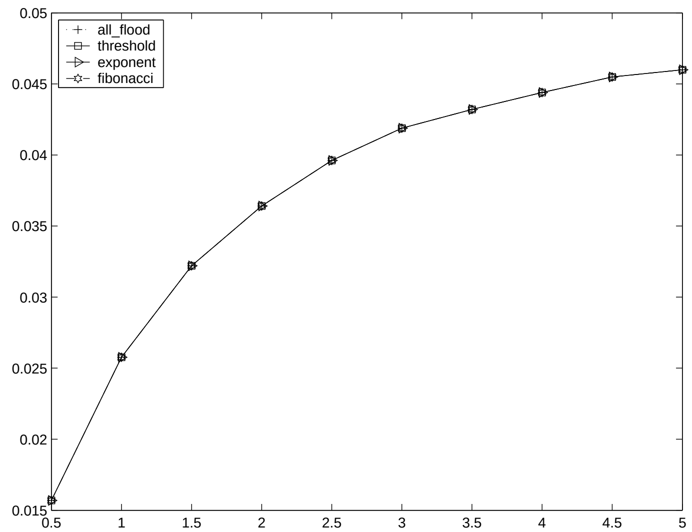

Figure 6: Algorithm1:Link Blocking Rate vs. Flood Delay, Threshold=4

3.3.2 Link Blocking Rate For Algorithm 2

Fig.7 and Fig.8 show the link blocking rate vs flooding delay when threshold is 2 and 4, respectively. Threshold flooding has the most link blocking rate. The difference among all-flooding, Fibonacci flooding, and exponent flooding is no more than 0.01%,which is intangible from the figure. This indicates that algorithm 2 makes good use of the network status information to route the connection requests, and more network information help algorithm 2 to optimize the utilization of the network resource.

Remark 2 *When algorithm 2 is used as the routing algorithm in the network, the more accurate the network status, the less link blocking rate is.*

Exponential and Fibonacci flooding have much less floods than all-flooding but with approximately the same link blocking rate.

3.4 Simulation Results for Network Blocking Rate

The following subsections show the simulation results for network blocking rate when algorithm 1, algorithm 2 are 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 allocated along the network, otherwise the connection is blocked.

In the figures below, 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, while the y-axis is the network blocking rate.

3.4.1 Network Blocking Rate for Algorithm 1

Fig.9 shows the mean of the network blocking rate over the flooding delay in the network. In this figure, the four curves are identical. So we have the following comments:

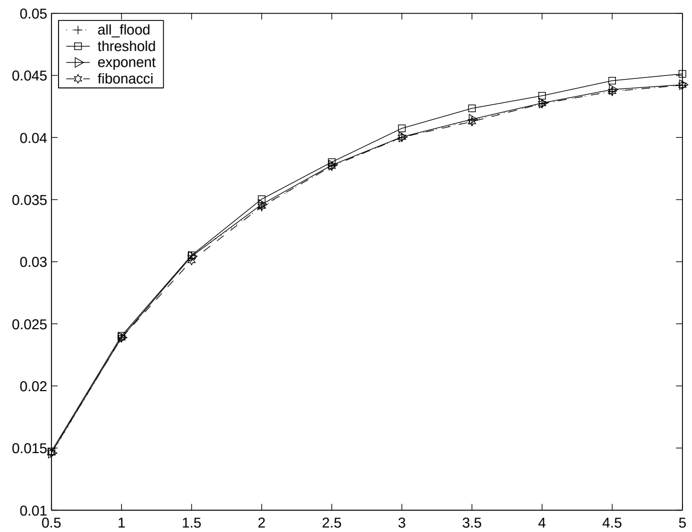

Figure 7: Algorithm2:Link Blocking Rate vs. Flood Delay, Threshold=2

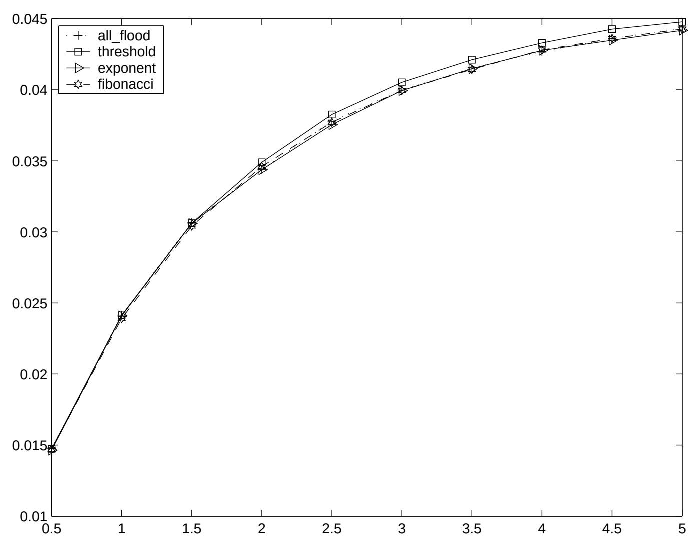

Figure 8: Algorithm2:Link Blocking Rate vs. Flood Delay, Threshold=4

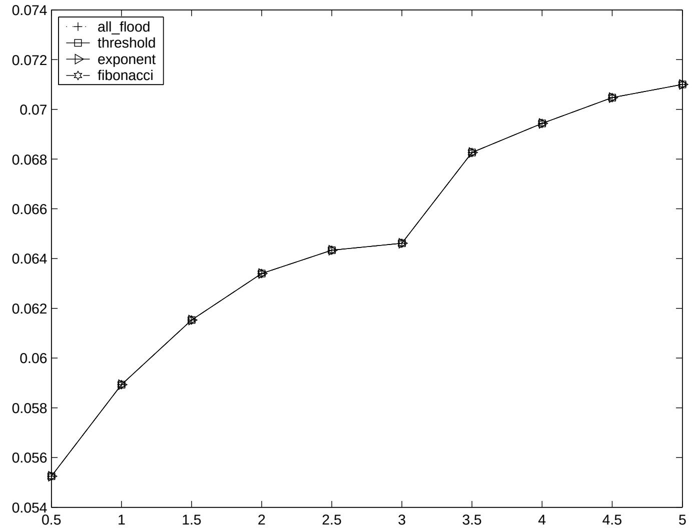

Figure 9: Algorithm1: Network Blocking Rate vs. Flood Delay, Threshold=2

Remark 3 *When algorithm 1 is used as the routing algorithm in a network, the network blocking rate is the same no matter what flooding scheme is used to advertise the network status.*

Exponential and Fibonacci flooding have much less floods than all-flooding but with the same link blocking rate.

The main reason for above comment is the same as the remark 1.

3.4.2 Network Blocking Rate For Algorithm 2

Fig.11 and Fig.12show that the more flooded information, the less network blocking rate.

Remark 4 *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 has 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.

3.5 Simulation Results For Flooding Frequency

The following figures and remark show the simulation results for the flooding frequency when algorithm 1, algorithm 2 are used.

Note that the flooding probability given in figure 1 is based on a single link network, while the flooding probability here is based on the simulated 14-node network. In the figures below, the x-axis represents the Load of the network, which is $180, 200, ..., 320$ Erlang,while the y-axis is the flooding frequency of the lazy flood schemes.

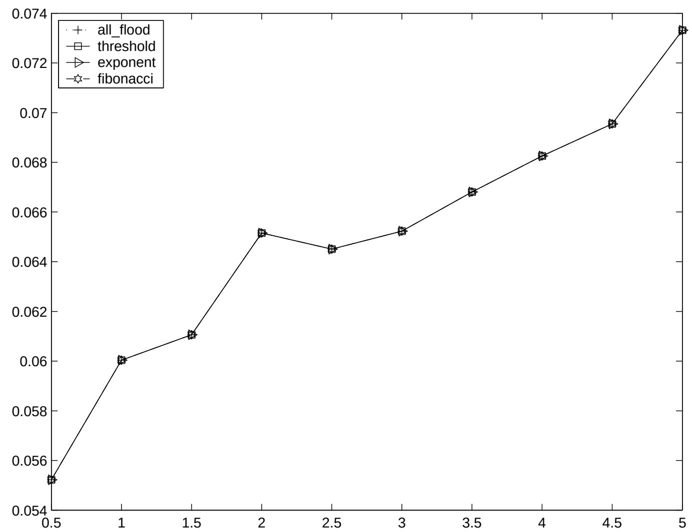

Figure 10: Algorithm1: Network Blocking Rate vs. Flood Delay, Threshold=4

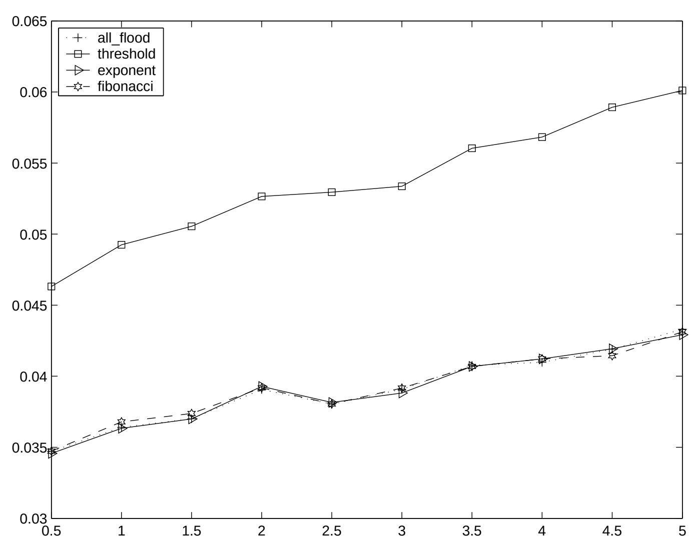

Figure 11: Algorithm2:Network Blocking Rate vs. Flood Delay, Threshold=2

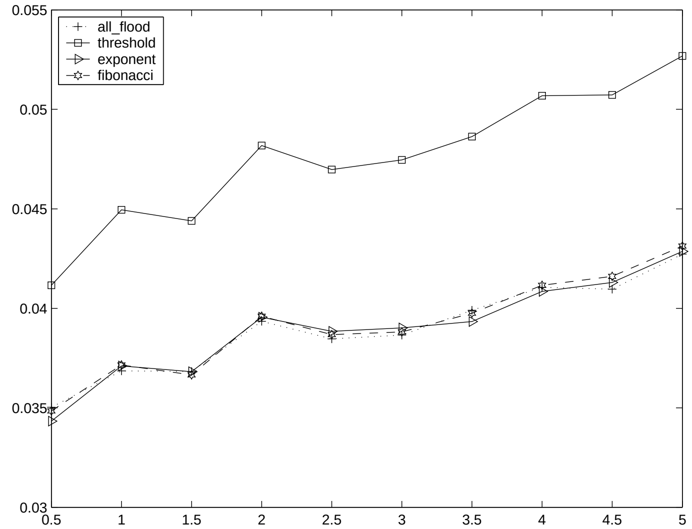

Figure 12: Algorithm2:Network Blocking Rate vs. Flood Delay, Threshold=4

Remark 5 *For the flooding frequency, $all\_flood > Fibonacci > exponential > threshold$ stands for any algorithm at any case.*

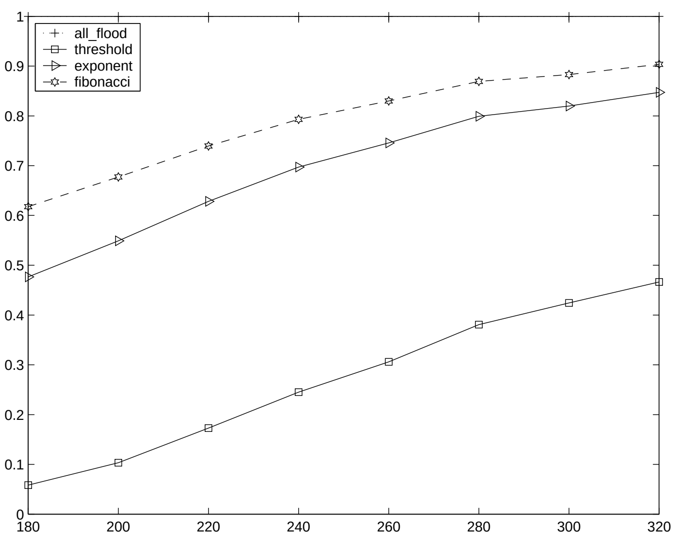

Figure 13: Algorithm1: Flooding Frequency vs. Network Load, Threshold=2

3.6 Simulation Results for Inaccurate Blocking Rate

The following figures and remarks show the simulation results for the *inaccurate_block* rate when algorithm 1 and algorithm 2 are used.

Remark 6 *When algorithm 1 is used as the routing algorithm in a network, the blocking*

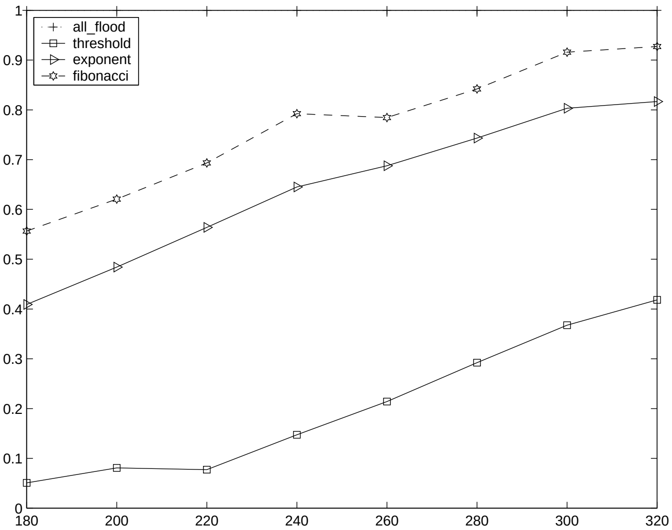

Figure 14: Algorithm2: Flooding Frequency vs. Flood Delay,Threshold=2

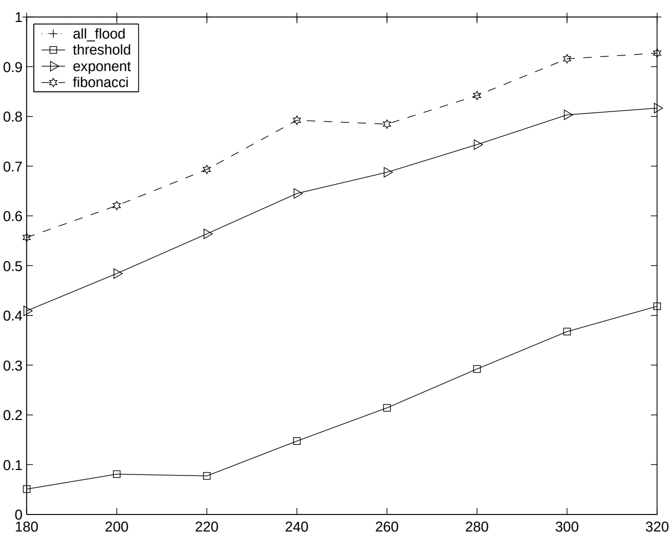

Figure 15: Algorithm2: Flooding Frequency vs. Flood Delay,Threshold=4

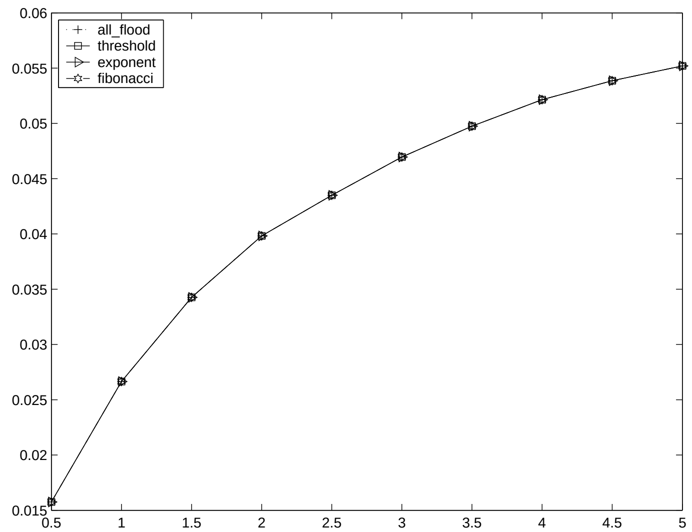

Figure 16: Algorithm1: Inaccurate Block vs. Flood Delay,Threshold=2

resulted by inaccurate network information is the same no matter what flooding scheme is used to advertise the network status.

Remark 7 *When algorithm 2 is used as the routing algorithm in a network, threshold flooding has the most inaccurate_block among all the flooding schemes.*

Comparing with all-flooding, Exponential and Fibinacci flooding have much less floods but almost the same inaccurate blocking.

4 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.

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 observed 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 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.

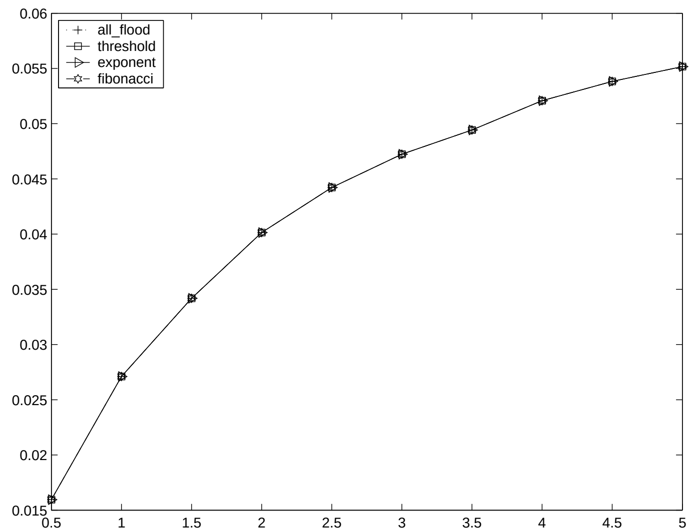

Figure 17: Algorithm1: Inaccurate Block vs. Flood Delay,Threshold=4

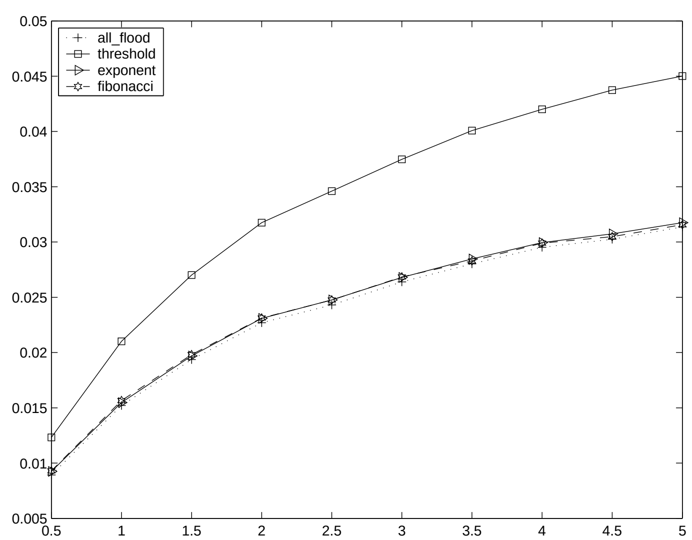

Figure 18: Algorithm2: Inaccurate Block vs. Flood Delay,Threshold=2

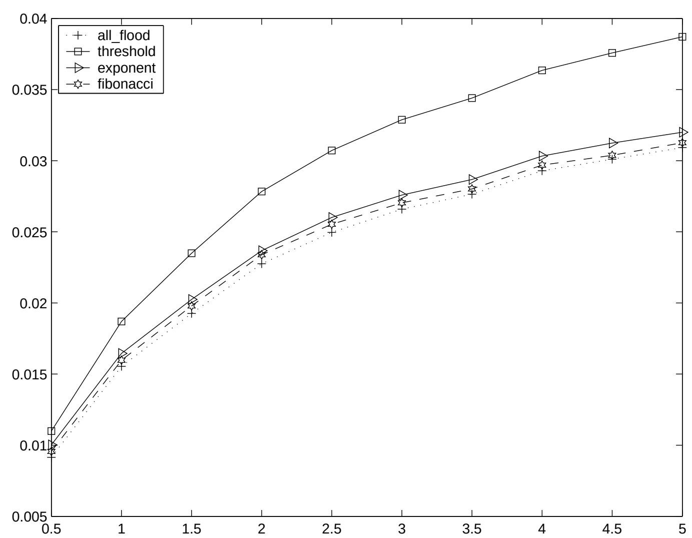

Figure 19: Algorithm2: Inaccurate Block vs. Flood Delay,Threshold=4

References

[1] A.Aho,and D.Lee, "Hierarchical Networks and the LSA N-Squared Problem in OSPF Routing", Proc. Globecom 2000.

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

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

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

[5] J.A.Bondy and U.S.R.Murry (1976) Graph theory and its applications, p19, THE MACMILLAN PRESS LTD.