INFORMATION DYNAMICS APPLIED TO LINK-STATE ROUTING[1]

Hyeonsang Eom, Ashok K. Agrawala, Sam H. Noh[2], A. Udaya Shankar
({hseom, agrawala, noh, shankar}@cs.umd.edu)
Institute for Advanced Computer Studies
Computer Science Department
University of Maryland
College Park, 20742

CS-TR-4297
UMIACS-TR-2001-75

November 1, 2001

Abstract

Information Dynamics [Agrawala, 2000] is an information-centric framework that provides a sufficient understanding of the characteristics of information used in systems for better system design and implementation. In this paper, we describe how to improve link-state routing based on this framework. Link-state routing protocols such as OSPF (Open Shortest Path First) [Moy, 1991] are currently used in many networks. In link-state routing, routes are determined based on link-delay estimates, which are periodically flooded throughout the network. This flooding of link-delay estimates is done without considering the relevance of these estimates to routing quality, i.e. without taking into account the usefulness of the link-delay information. We have developed a new approach that improves link-state routing by estimating future link delays and flooding these estimates only to the extent that they are relevant. This means that we consider the dynamics of the link-delay information and its usefulness. Simulation studies suggest that our approach can lead to significant reductions in routing traffic with noticeable improvements of routing quality in high-load conditions, demonstrating the effectiveness of the framework. We plan to further investigate the conditions where our information-dynamics approach is better than the standard approach.

---

[1] This work is supported partly by DARPA/Rome Labs, Department of the Air Force, under contract F306020020578 to the Department of Computer Science at the University of Maryland. The views, opinions, and/or findings contained in this report are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the Department of Air Force, DARPA, DOD, or the U.S. Government.

[2] Dr. Sam H. Noh is currently with the University of Maryland Institute for Advanced Computer Studies (UMIACS) and the School of Information and Computer Engineering, Hong-Ik University, Seoul, Korea.

1. Introduction

Information plays a major role in the operation of systems. In general, such information used in or generated by systems is also dynamic in nature. The Information-Dynamics framework [Agrawala, 2000] provides a new perspective for systems with a focus on information, information usefulness (or "value"), and the changes of information and its usefulness over time. Hence, with the framework, we can better understand the interactions between different components of a system that uses information. Such better understanding provides a basis for better system design and implementation. In this paper, we apply the information-dynamics framework to network systems. In particular, we focus on link-state routing where we show that the dynamic nature of link-delay information plays a key role in determining the dissemination of this information, and that the understanding of this role eventually leads to more efficient routing.

In link-state routing, each node in the network maintains a view of the current state of the network. The view is essentially a graph with vertices corresponding to the network nodes, edges corresponding to network links, and for each link, a cost representing an estimate of the current delay on the link. Each node makes (periodic and/or event-driven) measurements of the state of each of its outgoing links. It periodically constructs an estimate for the current delay on the link from these state measurements, and floods these link-delay estimates to all other nodes in the network [Peterson, 1996; Rosen, 1980] so that other nodes can update their views. Each node periodically uses its view to compute least-cost paths to all other nodes, where the cost of a path is the sum of the costs of all the links in the path. When a node receives a workload packet, it forwards the packet to the neighbor that is the next node in the least-cost path to the destination node of the packet.

The information-dynamics framework defines the interactions of entities (the basic building blocks of a system) in terms of information, thereby providing guidance on how a link-state routing system can be improved. In the framework, agents are defined as active entities that have capabilities to autonomously perform operations or actions, and that can also initiate actions. The nodes in a network are agents because they initiate routing activities (series of actions), i.e. periodic view and route updates.

Each agent has its perceived reality, i.e. its view of the *world*. Each node as an agent maintains its perceived reality that includes its network-state view, routes, and route costs. A context of an agent is a relevant part (to given information) of the agent's perceived reality that includes a goal and the cost involved in achieving the goal. In link-state routing, given link-delay information, each node has a context. The goal of each node in a link-state routing system is to route workload packets toward minimizing the end-to-end delays. Each node takes actions with the information, such as using and broadcasting information, in order to accomplish its goal. The main cost involved is the overhead of broadcasting link-delay estimates. The information-dynamics framework allows us to consider this context of each node in improving link-state routing.

The key concept in this whole framework is the notion of information dynamics, that is, the fact that the usefulness of information as well as information itself may change over time. This notion is a basis for improving link-state routing because in a network, the delays of links in a network are dynamic and the confidence level of link-delay information propagated to other nodes in link-state routing decreases over time. To help make use of the usefulness aspect of such dynamic information, the framework associates the notion of information utility to an agent, which is the benefit that the agent can receive by using the information. The utility function of the agent quantifies the benefit. Thus, the concept of information utility provided by the framework helps understand how the nodes in a link-sate routing system can evaluate link-delay information that they exchange. This understanding allows us to improve link-state routing by considering the utility.

With its utility function, an agent is bound to take actions toward maximizing the utility. The context of the agent for information is the domain of the utility function. Since the purpose of link-state routing is to provide information for accurate estimation of the current link delays at low cost, the utility of to-be-sent or received link-delay information may be determined by the closeness of the information to the current delay and the cost of the broadcast.

The usefulness of information to an agent in the context is the difference between the utility achieved with the information and the utility without it. Based on the usefulness of information, an agent decides whether or not to request, send, receive, store, or use the information. The information-dynamics framework allows us to understand that the usefulness

of link-delay information to a node in the context of link-state routing is decided by considering its utility, i.e. based on its contribution to the accurate estimation of the current link delays and its overhead.

This research is motivated by the fact that in link-state routing, each node floods its link-cost estimates without regard to whether the estimates will lead to less costly paths. From an information-dynamics perspective, the usefulness of the link-cost information is not considered. This could result in significant unnecessary routing traffic. We have developed a new approach that allows each node to disseminate link-cost information only when necessary (for estimating the current link delays i.e., when the information is useful), thereby leading to routing-traffic reduction. This reduction is the primary benefit of our approach.

Ideally, a workload packet should be routed based on the delays it will encounter at each link of the path at the time the packet gets to the link. That is, for each link along a potential route, the node doing the routing needs an estimate of the link delay at the (future) time when the workload packet would arrive at the link. We refer to this future delay as **encountered delay**. In the standard link-state routing, the encountered delay of a link is estimated by the exponential average of past link-state measurements. In our approach, each node estimates the encountered delay of a link based on a model of the dynamic change of the expected link delay given an instantaneous link-delay measurement. This estimation technique allows us to consider the dynamics of not only the link-delay information but also its usefulness. We expect that this estimation technique can improve the workload performance (e.g., delay, throughput). This improvement is an additional benefit of our approach.

The remainder of the paper is as follows: in Section 2, we give a more formal description of the problem that is addressed. We then present the approach that we take in Section 3. The experiments are presented in Section 4. In this section, we describe the network configuration and scenarios for our simulation studies, and present the preliminary comparison results obtained from these studies. Section 5 briefly surveys major related works. Finally, Section 6 concludes our work and summarizes our future work.

2. Problem Formulation

For a link, we treat the delay x at time t as a stationary stochastic process $\{x(t)\}$. Thus, the mean and variance of $x(t)$ are constant (independent of time t). Let m and s^2 denote the mean and variance, respectively. Also, the autocorrelation function $E[\{x(t)-m\}\{x(t+t)-m\}]/s^2$ depends only on the lag t and not on time t. Let $r(t)$ denote this autocorrelation function.

Consider the instantaneous conditional mean and variance, respectively, of the delay given a measurement x_0 at time t_0:

$$E\{x(t) \mid x(t_0) = x_0\}, \text{ where } t_0 < t$$

$$Var\{x(t) \mid x(t_0) = x_0\}, \text{ where } t_0 < t$$

If no other measurement is available, we expect the instantaneous conditional mean to change from x_0 towards m over time. Similarly, we expect that the instantaneous conditional variance to change from zero to s^2 over time. When the measurement is made, the conditional variance is zero because the measurement is valid at that time.

Our approach is to develop estimates for the functions $E\{x(t) \mid x(t_0) = x_0\}$ and $Var\{x(t) \mid x(t_0) = x_0\}$. Then we will use these estimates to do selective broadcasts and determine least encountered-cost paths.

3. Approach

We assume that the conditional mean decays exponentially over time to its steady-state value. Based on this assumption, we use

$$\hat{m}(x_0, t_0, t) = x_0 + (m - x_0)(1 - e^{-a(t-t_0)}) \quad (t \geq t_0)$$

as an estimate of $E\{x(t) \mid x(t_0) = x_0\}$, where a is a non-negative constant to be determined. This is illustrated in Figure 1. Similarly, we use an exponential-decaying estimate $\hat{s}^2(x_0, t_0, t)$ for $Var\{x(t) \mid x(t_0) = x_0\}$, as illustrated in Figure 2.

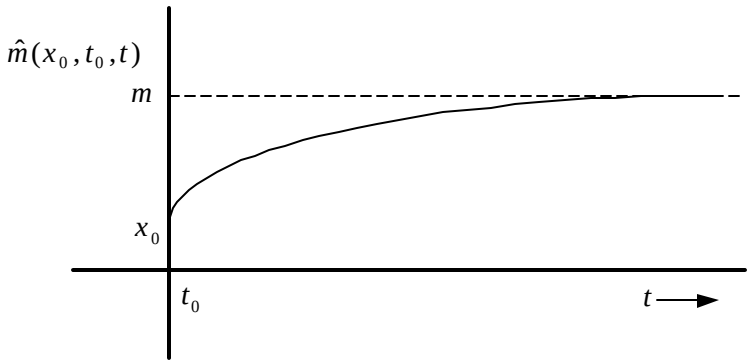


Figure 1 Evolution of the instantaneous conditional delay-mean estimate

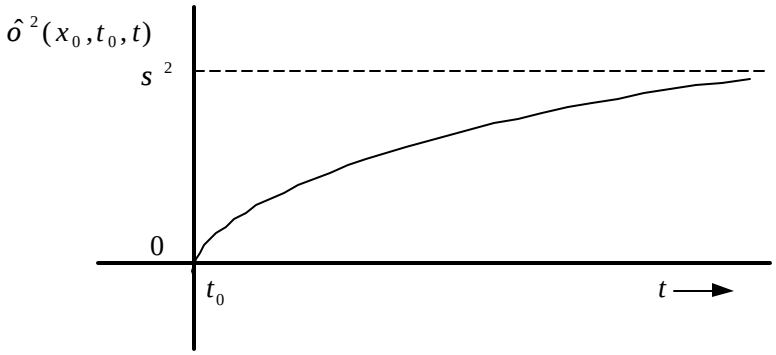


Figure 2 Evolution of the instantaneous conditional delay-variance estimate

Given these estimation functions, a node computes the encountered delay of a packet on a path as follows. Let the path have links $l_1, l_2, ..., l_n$, and let the node send the packet into the path at time t_0. Let $\hat{m}^{l_i}(t)$ be the function estimating the encountered delay on link l_i at time t. The estimated encountered delay for the packet on link l_1 is $\hat{m}^{l_1}(t_0)$. The estimated encountered delay for the packet on link l_2 is $\hat{m}^{l_2}(t_0 + \hat{m}^{l_1}(t_0))$, and so on. So the estimated encountered delay for the packet on the path is given by:

$$\hat{m}^{l_1}(t_0) + \hat{m}^{l_2}(t_0 + \hat{m}^{l_1}(t_0)) + \hat{m}^{l_3}(t_0 + \hat{m}^{l_1}(t_0) + \hat{m}^{l_2}(t_0 + \hat{m}^{l_1}(t_0))) + ... + \hat{m}^{l_n}(...)$$

Computing path costs in this way, the node would route the packet on the path with the least encountered-delay estimate. Each node determines the least-cost paths using the standard shortest-path algorithm [Dijkstra, 1959] as in link-state routing. Thus, routing is as in standard link-state routing except for the path-cost computation. To do this computation, each node maintains a view as in link-state routing except that a measured delay and measurement time are kept for each link. The node updates its view of a local link (i.e., a link outgoing from itself) whenever a workload packet is sent on that link. The node updates its view of a remote link whenever it receives a measurement update for the link. View updates are not periodic.

At each view update, each node broadcasts the updated delay information to its neighbors only if the estimated encountered delay on the corresponding link at that time is significantly different from the steady-state mean. This is how the node determines the dynamic usefulness of the updated delay information with respect to routing-quality improvement, and broadcasts only useful information. We assume that every node knows the steady-state value of each link. Hence, no propagation of link-delay measurements is required beyond some point; if a node does not receive any measurement for a link, it will use the steady-state value.

Each node maintains a routing table that indicates the next hop for each destination. With its view for all links, each node updates its routing table by computing the least-cost paths to all the other nodes just before it decides which of its outgoing links to send the packet onto when it receives a workload packet. We refer to this update technique as the "just-in-time route-update" method. This method allows each node to determine the least-cost paths for the most recent time using the most recent delay information for each link. Note that the periodic-update scheme used in link-state routing is not suitable for our approach. If the periodic-update scheme were used, the link delays estimated using our approach would be used without any change until the next route-update time. The problem with this is that these estimates could be different from the steady-state values.

4. Simulation

To show the overall applicability of this approach to link-state routing, we compared via simulation a routing scheme using our approach with SPF (Shortest Path First), a link-state routing technique. We call our routing scheme the InfoDyn (Information-Dynamics) scheme.

For simulation studies, we used MaRS, the Maryland Routing Simulator [Alaettinoglu, 1994; Shankar, 1992]. We tried SPF with two kinds of link-cost functions, a delay cost function and a hop-normalized-delay function [Khanna, 1989].

4.1 Network configuration and scenarios

We conducted studies for the NSFNET-T1-backbone topology. In this configuration, there are 14 nodes connected via 21 links. Each link represents two one-way channels. Each node can process a data packet of 544 bytes in 1 ms, and each link channel has 183 KB (1.4 Mbps) bandwidth. We initially assume that there is no propagation delay for each link.

In this network, a workload is generated by FTP source and sink pairs. These sources and sinks are connected to nodes. FTP is regulated by a flow-control mechanism and an acknowledgement mechanism with retransmission. The flow-control mechanism is a static window-based scheme implemented in MaRS. This scheme consists of two windows: produce and send windows. We set the produce-window size to infinity, and the send-window size to eight. Also, we initially use 120 seconds as the total simulated time.

There are two kinds of FTP flows: regular and on-off flows. In each regular flow, the source starts transmitting packets at time 0, and sends as many packets as possible with an inter-packet production delay of 1 ms. For each on-off flow, there are alternating constant-length on and off intervals. Each on-off flow starts at a different time (from 0 to 24 seconds), and has a different length (from 20 to 120 seconds). Also, a certain number of packets are produced at once at the beginning of on intervals while no packets are produced during off intervals. The number of packets for each on interval is determined so that the packets of that number would be successively transmitted during the on interval without any flow-control mechanism and without any other flow. Specifically, the number is the length of an on interval divided by the transmission time of a data packet, where the transmission time is the packet size divided by the link bandwidth.

We initially consider five scenarios in this network configuration: N0 – N4. The level of queuing delay of these scenarios is high: in the best cases (lowest-average-delay cases after parameter tuning) of using SPF with 1 second route-update intervals, the average queuing-delay portions of the average round-trip delay per packet are around 94 %. Also, the

utilizations (the average fractions of the time when packet queue size > 1) range from 0.73 to 0.74. There are 121 FTP flows in Scenario N0 to N3, and 131 flows in Scenario N4. Table 1 shows the differences between the scenarios. In particular, Scenario N4 has two hot spots (each of which receives packets from every other node).

Table 1 Differences in the FTP-flow characteristics between scenarios

Scenario	*Number of Regular Flows*	*Number of On-Off Flows*	*Length of On-Off Intervals (Seconds)*
N0	60	61	5
N1	60	61	10
N2	60	61	15
N3	0	121	5
N4	55	76	5

4.2 Preliminary results

For the InfoDyn scheme, we used an exponential-change-rate (α) value and a threshold value for the selective broadcast of routing packets, during each simulation run for each scenario. We tried seven threshold values. Also, we tried eight α values across the full value range in each of these different-threshold-value cases. As the steady-state value of each link in each simulation run using the InfoDyn scheme, we used the sample delay mean of the corresponding link computed in a simulation run using SPF with 1 second route-update intervals for the same scenario.

The use of the InfoDyn scheme without any routing-packet broadcast (thereby with only local-link view update) is called the InfoDyn Short-Term Steady-State (STSS) case. Hereafter, "best" means leading to the lowest average round-trip delay per packet.

4.2.1 InfoDyn Short-Term Steady-State (STSS) case

The InfoDyn STSS case with the best α of the exponential model results in 3 to 8 % reduction in the Average (Avg) Round-Trip (RT) delay per packet and 4 to 22 % reduction in the standard deviation (STD) in all scenarios compared with the best cases of using SPF with 1, 10, and 30 second route-update intervals - we obtained the best result of using SPF for each combination of a route-update-interval length and a scenario by tuning several cost-function parameters. Figure 3 shows these reductions. Note that there are no routing packets sent out in this InfoDyn case while 75,642, 7,602, and 2,562 routing packets are sent out with 1, 10, and

30 second route-update intervals, respectively, in the SPF cases. These results imply that when every node knows the "long-term" steady-state delay-mean values of all links and uses our routing approach, flooding requirements can be significantly reduced with noticeable reductions in the average delay and the variance, compared with the standard link-state routing approach where each node periodically broadcasts "short-term" steady-state values (exponential averages) for link delays.

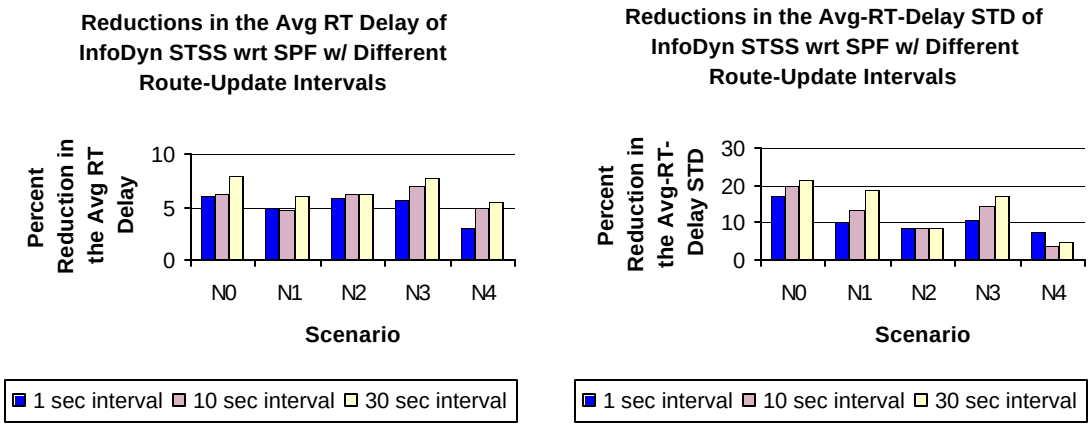


Figure 3 Reductions in the average round-trip delay and STD of InfoDyn STSS

4.2.2 Impact of routing-packet broadcast

For each scenario with a fixed α value, the average delay and STD are almost the same across simulation runs with different threshold values except for those runs where a very large number of routing packets are broadcast. For example, Figure 4 shows the impact of varying the threshold value in Scenario N3 (the all-on-off case) when the best α is used. There are three charts. The left-most and middle charts indicate the changes in the average delay and STD, respectively, depending on the threshold value used. The right-most chart shows the numbers of routing packets used for different threshold values. The smaller the threshold value, the more routing packets are sent out. When about 40,000 routing packets are used (the 100 ms threshold-value case), there are 0.53 ms and 0.66 ms increases in the average delay and STD, respectively, compared with 152.03 ms average delay and 92.24 ms STD of the best case (the 130 ms threshold-value case). Similar impacts of routing-packet broadcast are observed for the other scenarios. Figure A1 (Pages 16 and 17) in APPENDIX shows the same three charts in each row for each of the other scenarios. As in the figure, for the threshold values that correspond to less than 100,000 routing packets in each scenario, the variation of the

average delays is within 1 ms and that of STDs is within 5 ms. Note that these numbers are the scale units in the delay and STD charts, respectively.

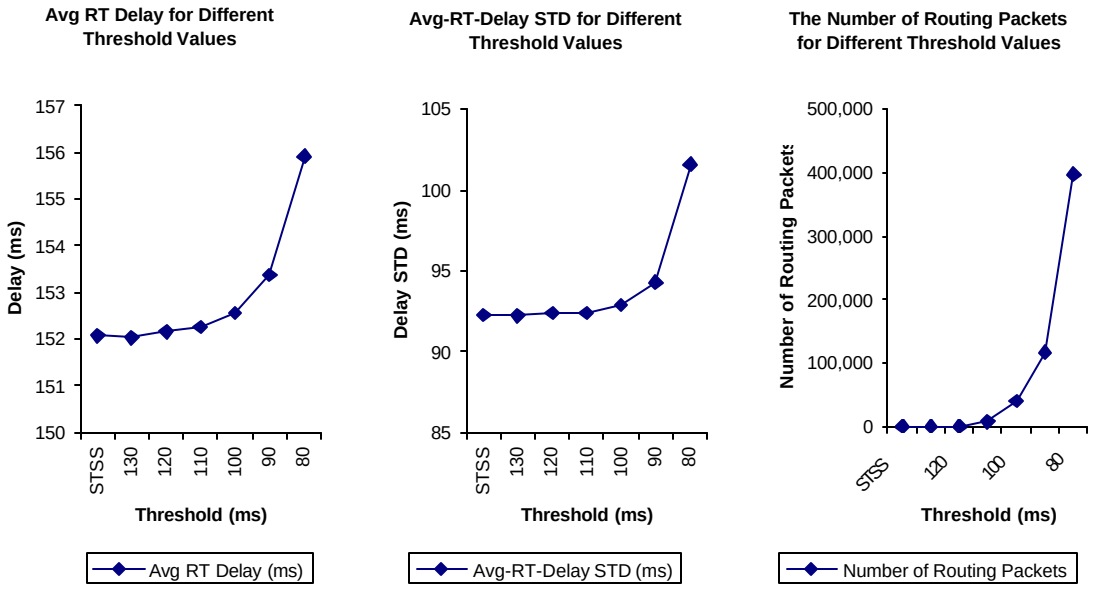


Figure 4 Impact of varying the threshold in Scenario N3 (when using the InfoDyn scheme w/ the best a)

The average delay and STD range from 152.0 to 161.9 ms and from 89.7 to 109.1 ms, respectively, in the best cases of using the InfoDyn scheme (with the best a) in all scenarios when routing packets are broadcast. Compared with this best case for each scenario, the InfoDyn STSS case with the same best a leads to increases in the average delay and STD by up to 0.1 ms and 0.3 ms, respectively. The reason why these increases are small is that the impact of a routing packet on routing quality is transient: the encountered link delay estimated by the receiving node using the delay measurement contained in the packet soon becomes the steady-state value. These results indicate that each node may not need to broadcast link-delay measurements when using the InfoDyn scheme.

4.2.3 Impact of varying the a value

There are two possible sources for the routing-quality improvement: use of the long-term steady-state link-delay means and link-delay estimation with the exponential delay-mean change. To see the influence of each of these factors, we first set a to infinity. Then, the link-delay means are used without any change in route determination. In the InfoDyn STSS case,

this Static-Routing case leads to up to 5 % and 18 % increases in the average delay and STD, respectively, in four scenarios and 2 % and 9 % decreases, respectively, in one scenario compared with the best cases of using SPF. These results mean that the use of the link-delay means is not a source of routing-quality improvement in most cases. However, the use of the best α results in 4 to 11 % and 7 to 22 % decreases in the average delay and STD, respectively, in all scenarios compared with these Static-Routing cases. These results indicate that the selection of the α value is crucial for routing-quality improvement.

The best routing quality is achieved with the same α across all scenarios in the case of using the same threshold value or in the InfoDyn STSS case. For example, Figure 5 shows the effects of using different α values in Scenario N3 in the InfoDyn STSS case. The left and right charts indicate the changes in the average delay and STD, respectively, depending on the α value. As in the figure, the average delay and STD increase as the α value used digresses in both directions from the value (1000) for the best result. Similar trends are observed for the other scenarios. Figure A2 (Pages 18 and 19) in APPENDIX shows the same two charts in each row for each of the other scenarios. Therefore, if we can find the best or a near-best setting in one case, we may reduce the average delay and STD by using the same setting in other cases. In fact, routing quality is improved for a wide range of α values. Table 2 shows the α ranges of the InfoDyn STSS cases in all scenarios that lead to decreases in the average delay with respect to the best SPF cases. Therefore, the parameter tuning is not required.

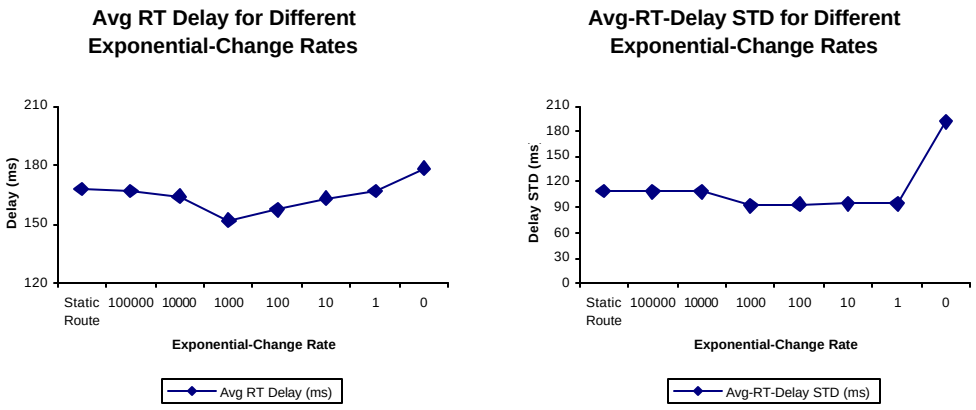


Figure 5 Impact of varying the α value in Scenario N3 (in the InfoDyn STSS case)

Table 2 *a* Ranges of the InfoDyn STSS cases leading to decreases in the average RT delay wrt the best SPF cases

Scenario	*Rate (Circled if Routing Quality is Improved)*							
	Static Route	*100,000*	*10,000*	*1,000*	*100*	*10*	*1*	*0*
N0	O	O	O	O	O	O		
N1		O	O	O	O	O		
N2				O	O	O		
N3				O	O			
N4				O	O			

5. Related Work

Typically, delays vary and change rapidly in a network. For example, at a fine-grained level, the characteristics of the Internet are highly dynamic [Agrawala, 1998]. Such dynamics in networks make it difficult to estimate encountered link delays. Many researchers have investigated the dynamic behavior of networks such as the dynamics of end-to-end Internet packet delays. [Agrawala, 1998; Labowitz, 1998; Paxson, 1999; Pointek, 1997; Sanghi, 1993].

For statistical uncertainty modeling concerning information estimation, there are two basic approaches: modeling based on past observations followed by extrapolation, and modeling via the analysis of factors that determine the information at the target estimation time. An example of the first modeling approach is a time-series model such as an AutoRegressive Integrated Moving Average (ARIMA) model [Box, 1994; Chatfield, 1984]. An example of the second is a regression model for factor(s)-and-effect information pairs (or tuples). The parameters of both modeling approaches can be estimated using least-squares fitting [Trivedi, 1982].

6. Conclusion and Future Work

Our preliminary results indicate that our approach is promising. When we compared our routing scheme based on a new link-delay-estimation technique with SPF via simulation for various FTP-workload scenarios with the NSF-T1-backbone network topology, we found that our routing scheme could achieve 100 % reductions in routing traffic with 3 to 8 % decreases of the average round-trip delay per packet in high-load conditions.

These routing-traffic reduction and routing-quality improvement resulted from the estimation of future (encountered) link delays based on the dynamics of the expected link delay given an instantaneous link-delay measurement, and from the consideration of the dynamic usefulness of the link-delay measurement via this estimation. Hence these benefits demonstrate the effectiveness of the information-dynamics framework.

We plan to characterize the situations where we can improve link-state routing by using our information-dynamics approach. For this research, we plan to further investigate the effectiveness of our routing scheme via extensive simulation studies with different patterns of dynamic workload and/or with different parameter settings for the network. We will try random scenarios created by enabling random parameters such as the average number of packets per FTP connection and the average delay between connections. In addition to FTP workload, we will use other types of workload. Also, we will create and try scenarios with different levels of load condition to investigate the relationship between load level and the benefit of using our scheme. In addition, we will use higher and/or different link-bandwidth and propagation-delay values. Based on the results of these studies, we will determine the characteristics of the situations that lead to significant routing-traffic reductions with routing-quality enhancements in the case of using our approach, compared with standard link-state routing.

Each node needs to estimate the steady-state value of each link in order for our approach to be practical. Our preliminary results indicate that each node may compute the sample mean of the delay of each local link using standard link-state routing for a long period of time, and flood the sample mean periodically (but, at a lower frequency) so that all other nodes can use it as the steady-state value. We will also study different ways to compute the sample mean and provide a guideline for the computation.

7. Reference

[Agrawala, 2000] A.K. Agrawala, R. Larsen, and D. Szajda "Information Dynamics: An Information-Centric Approach to System Design," *Proceedings of the International Conference on Virtual Worlds and Simulation*, January 2000.

[Agrawala, 1998] A.K. Agrawala, The NetCalliper Project, http://www.cs.umd.edu/projects/sdag/netcalliper, 1998.

[Alaettinoglu, 1994] C. Alaettinoglu, A.U. Shankar, K. Dussa-Zieger, and I. Matta, "Design and Implementation of MaRS: A Routing Testbed," *Journal of Internetworking: Research & Experience*, 5 (1):17-41, 1994.

[Box, 1994] G.E.P. Box, G.M. Jenkins, and G.C. Reinsel, *Time Series Analysis: Forecasting and Control,* Prentice Hall, 1994.

[Chatfield, 1984] C. Chatfield, *The Analysis of Time Series: an Introduction*, Chapman and Hall, London & New York, 1984.

[Dijkstra, 1959] E.W. Dijkstra, "A Note on Two Problems in Connection with Graphs," *Numerische Mathematik*, 1, pp. 269-271, 1959.

[Khanna, 1989] A. Khanna and J. Zinky, "The Revised ARPANET Routing Metric," *Proceedings of ACM SIGCOMM*, pp. 45-56, 1989.

[Labowitz, 1998] C. Labowitz, G.Malan, and F. Jahanian, "Internet Routing Instability," *IEEE/ACM Trans. Networking*, 6 (5):515-528, October 1998.

[Moy, 1991] J. Moy, *Open Shortest Path First v.2*, RFC 1247, 1991.

[Paxson, 1999] V. Paxson, "End-to-End Internet Packet Dynamics," *IEEE/ACM Trans. Networking*, 7 (3):277-292, June 1999.

[Peterson, 1996] LL.Peterson and B.S. Davie, *Computer Network: A Systems Approach*, Reprographic Services, 1996.

[Pointek, 1997] J. Pointek, F. Shull, R. Tesoriero and A.K. Agrawala, "NetDyn Revisited: A Replicated Study of Network Dynamics", *Computer Networks and ISDN Systems*, 29(7):831-840, August 1997.

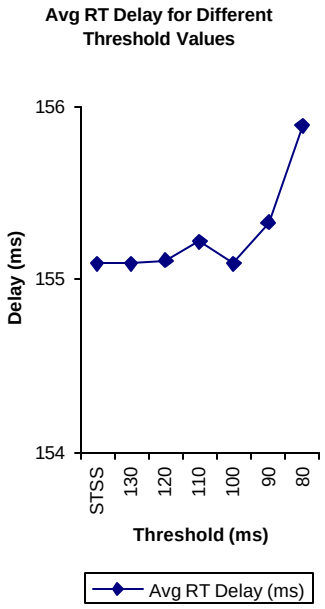
[Rosen, 1980] E.C. Rosen, "The Updating Protocol of ARPANET's New Routing Algorithm," *Computer Networks*, 4(1), pp. 11-19, 1980.

[Sanghi, 1993] D. Sanghi, O. Gudmundsson, and A.K. Agrawala, "A Study of Network Dynamics," *Computer Networks and ISDN Systems*, 26(3), pp. 371-378, November 1993.

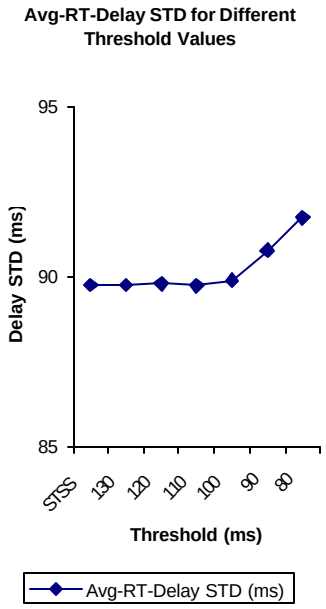
[Shankar, 1992] A.U. Shankar, C. Alaettinoglu, I. Matta, and K. Dussa-Zieger, "Performance Comparison of Routing Protocols using MaRS: Distance-Vector versus Link-State," *Proceedings of ACM SIGMETRICS,* pp. 181-192, 1992.

[Trivedi, 1982] K.S. Trivedi, *Probability & Statistics with Probability, Queuing and Computer Science Applications*, Prentice Hall, 1982.

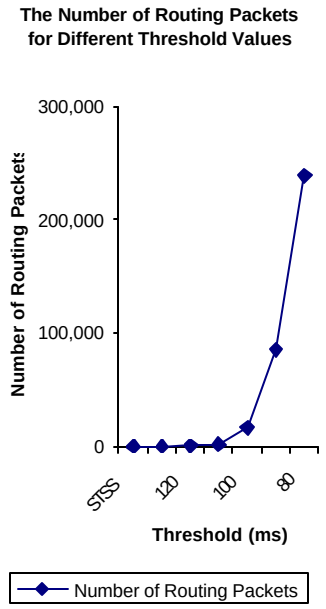
APPENDIX Supplementary Figures



[a]

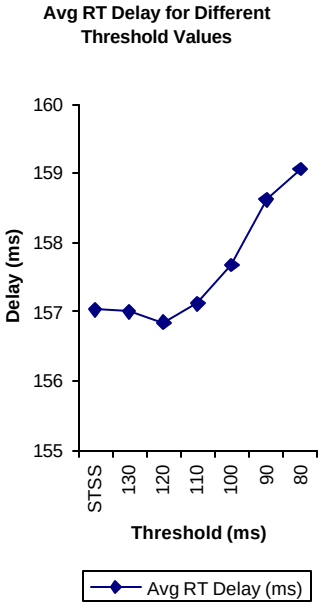


[b]

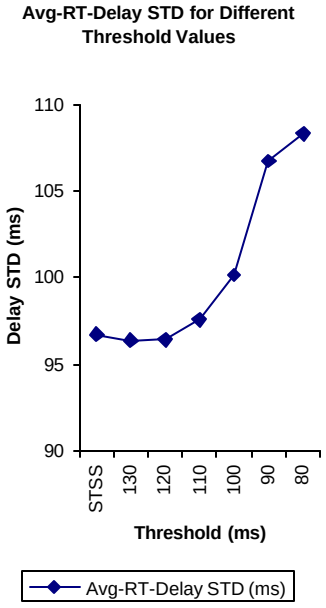


[c]

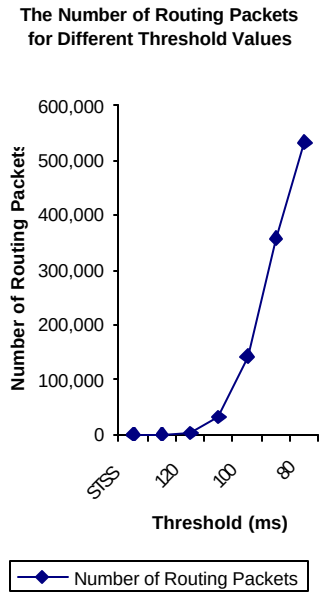
([a], [b], and [c]: Scenario N0)



[d]

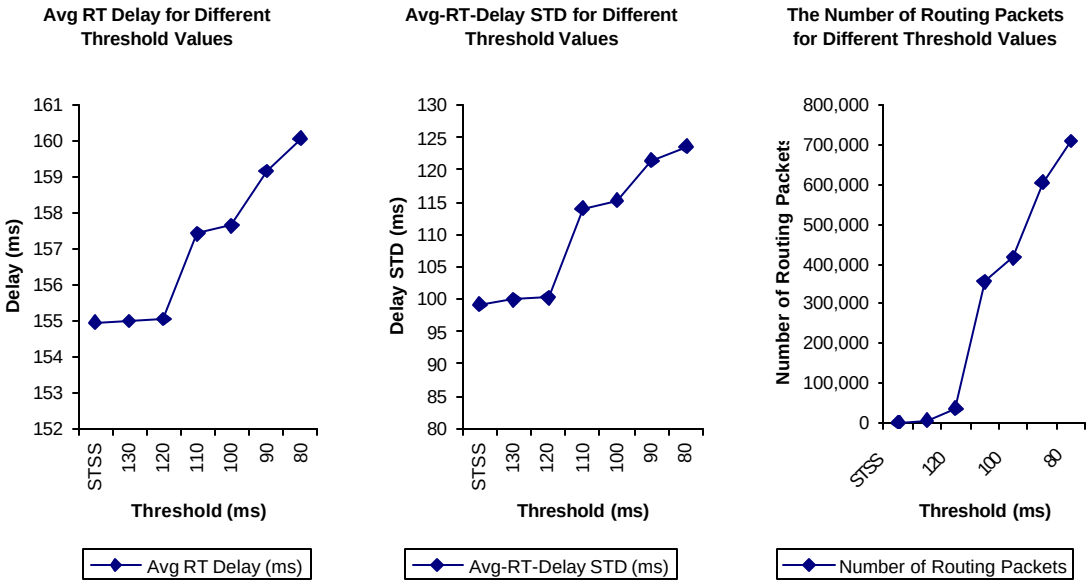


[e]



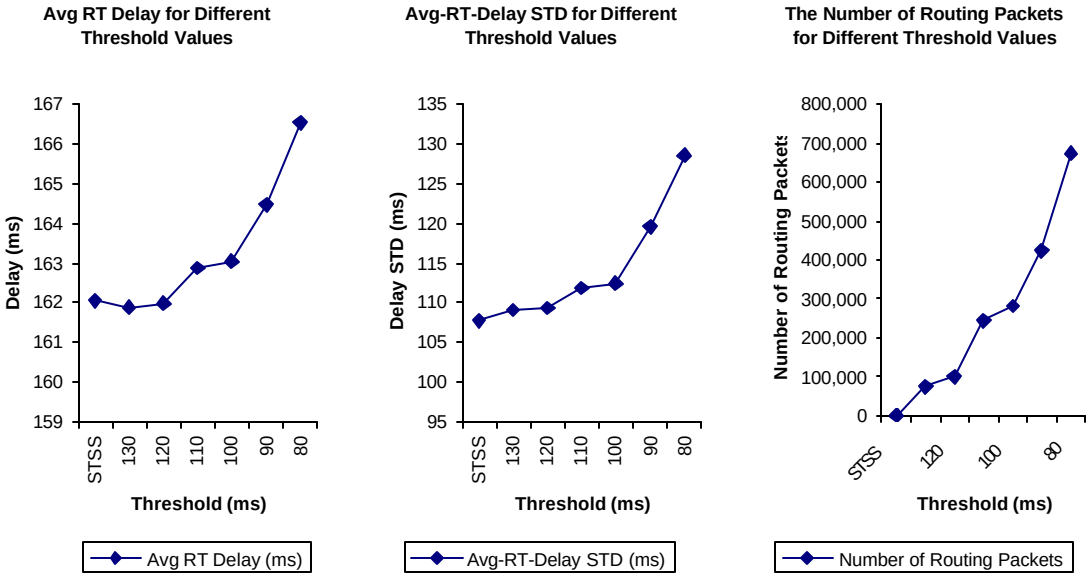
[f]

([d], [e], and [f]: Scenario N1)



[g] **[h]** **[i]**

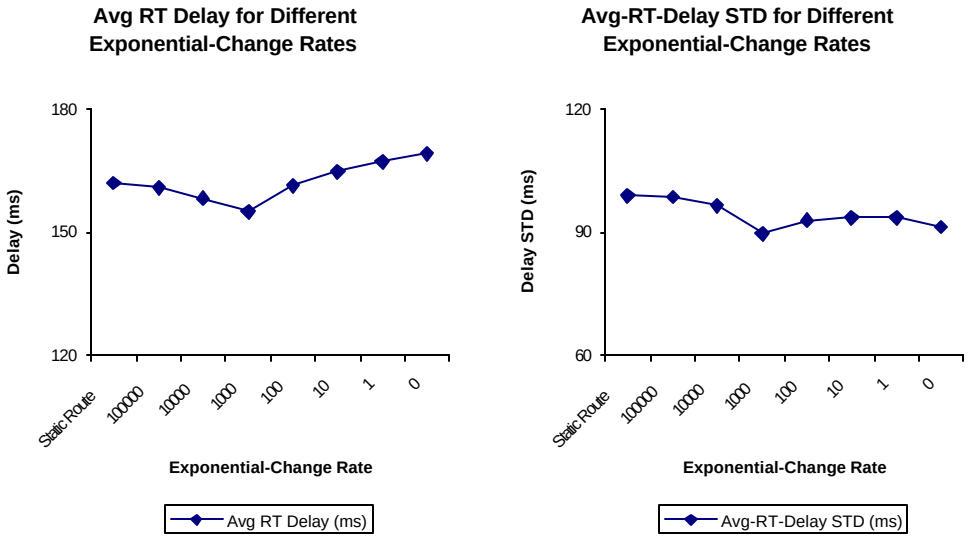
([g], [h], and [i]: Scenario N2)



[j] **[k]** **[l]**

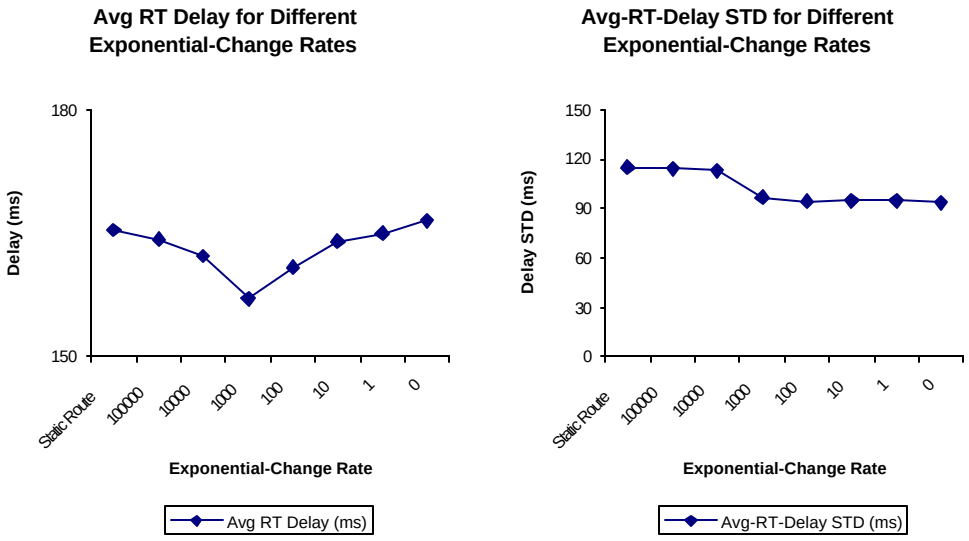
([j], [k], and [l]: Scenario N4)

Figure A1 Impact of varying the threshold in Scenarios N0, N1, N2, and N4 (when using the InfoDyn scheme w/ the best a)



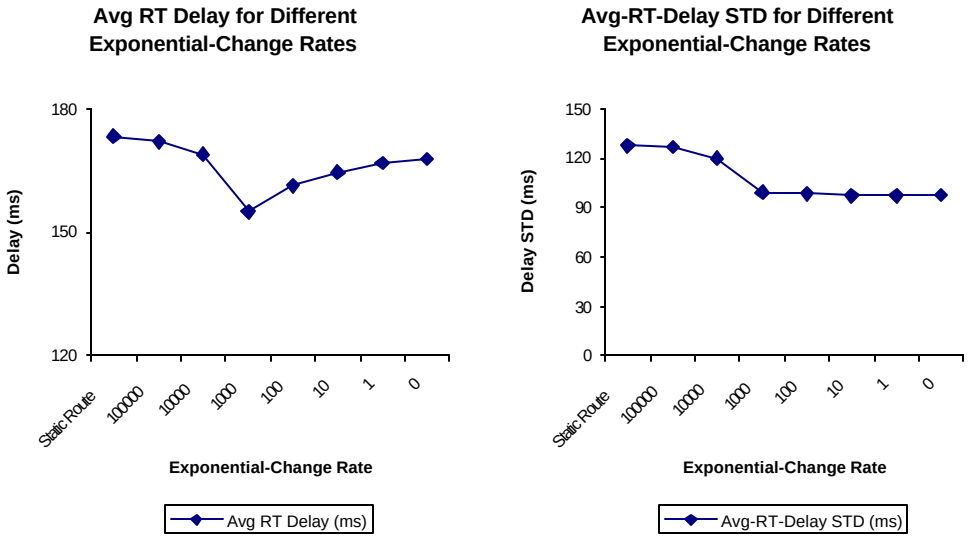
[a] **[b]**

([a] and [b]: Scenario N0)



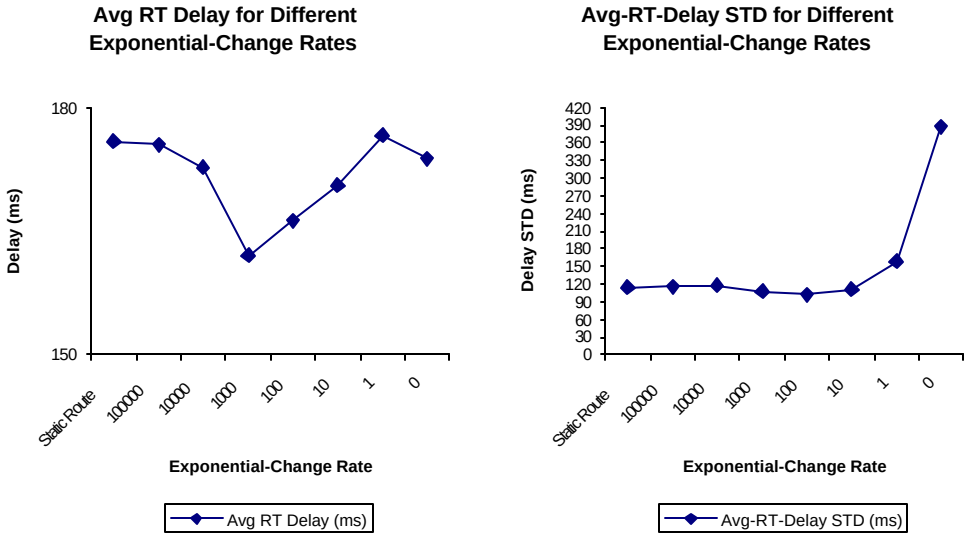
[c] **[d]**

([c] and [d]: Scenario N1)



[e] **[f]**

([e] and [f]: Scenario N2)



[g] **[h]**

([g] and [h]: Scenario N4)

Figure A2 Impact of varying the a value in Scenarios N0, N1, N2, and N4 (In the InfoDyn STSS cases)