Evidence Combination for Traffic Adaptive Routing

M. Balasubramanian
Computer science, LSU
madhu@csc.lsu.edu

Louise A. Perkins
Computer science, USM
louise.perkins@usm.edu

S.S. Iyengar
Computer science, LSU
iyengar@csc.lsu.edu

Sumeet Dua
Computer science, LaTech
sdua@coes.latech.edu

Donald H. Kraft
Computer science, LSU
kraft@csc.lsu.edu

Abstract

We propose a novel traffic adaptive routing algorithm to achieve equal path utilization in dynamic networks. We choose traffic measured at each router and the trip-time measured between networks as the indicators of the network condition. We derive the states of the equal-cost paths by combining the evidences pointed by the traffic and trip-time measures using Dempster-Shafer (DS) theory of evidence. The DS theory computes a belief measure and takes into account any ignorance present in the information sources acquiring the evidences and the inherent uncertainty in the network paths due to their dynamic nature. Based on the states of the equal-cost paths determined using DS theory, the routers route data packets among the equal-cost paths proportional to their load conditions. For example, if equal-cost paths P_1 and P_2 as seen from router R are in low and high load condition respectively, the router R will route $2/3^{rd}$ of the data packets through P_1 and $1/3^{rd}$ through P_2. We modified the LINUX IP stack for adaptive routing and traffic measurements. The routing tables are updated every 30 seconds based on the states of the equal-cost paths determined using the DS theory. We demonstrate the algorithm in an interconnected network with LINUX machines configured as routers and show that the proposed evidence combination framework will result in improved path utilization and compare the performance with the per-packet mode of traffic distribution.

1. Introduction

The modern digital data communication architecture has its roots from the packet switching standards developed during 1970s. Data originating from the source computer is usually divided into packets. The routing elements, referred as routers, route the data packets to the destination network. When there are multiple paths available to a destination, each packet may travel in different routes and may arrive at the destination in an order different from the original data. The receiving computer reorganizes the data using the control information in the packets.

The Internet can be viewed as a web of sub-networks, wired or wireless, kept alive through the routers. The routers identify the next routing point for a packet using a set of predefined routes known as static routes or learn the paths dynamically. RIP, OSPF, IGRP and EIGRP [1, 2] are some of the standard dynamic routing protocols used in dynamic networks. The dynamic routing protocols define a cost parameter for each available path to a destination network using quality metrics such as number of hops and quality of service. Multiple paths with equal cost to

Proceedings of the 18th International Conference on Systems Engineering (ISCEng'05)
0-7695-2359-5/05 $20.00 © 2005 IEEE

a destination network are referred as equal-cost (EC) paths. EC paths provide an opportunity to distribute the data packets to a destination. Figure 1 shows a schematic representation of two EC paths P_1 and P_2 to a destination network net_B.

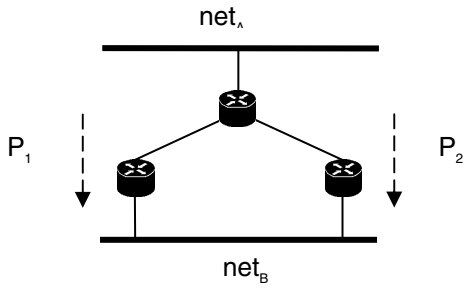

Figure 1. Equal-cost paths P_1 and P_2

The static per-packet and per-destination modes of router traffic distribution attempts to achieve an equal utilization of the EC paths. The per-packet mode routes the data packets in a round-robin fashion among the available EC paths and per-destination mode alternates among the EC paths for each destination network. The static traffic distribution schemes assume that the paths are under equal utilization. For example, if path P_1 is under higher utilization than path P_2, an equal traffic sharing among the EC paths will not improve the state of the path P_1. Hence there is a clear need for the routers to consider the state of the paths while routing.

Several adaptive routing schemes exist that utilize the knowledge about the states of the paths in determining an optimal path to route a packet. The authors of [3] adjust the link costs according to the traffic measured at various routing points and use an adaptive distance vector routing protocol to update the routing tables. Agent based routing algorithms inspired by the intelligent behavior of certain insects use forward and backward agents to identify the states of various paths [4, 5]. ARPANET [6] uses a trip-time measure in choosing an optimal path.

The traffic-metric is usually measured in the routing elements and it does not account for the status of the router buffer. A buffer-full condition in the routers will cause packet drops and result in resending the packets. Hence the traffic-rate measured at the routers cannot completely represent the states of the paths.

The trip-time through a network path depends on: 1. Router buffer capacity, 2. Router buffer status, and 3. Service rate at the routers, which depends on the CPU utilization and the traffic in the path. Thus the trip-time/path-delay measured can represent the state of the network at a higher confidence level. The network trip-time metrics are usually measured using special control packets called forward and backward agents. In dynamic networks, the states of the paths change and hence the trip-time needs to be measured frequently to derive a more accurate state of the paths. Sending more routing agents through the paths may introduce additional overhead in the network. The confidence in the delay metrics used to represent the path utilization will be reduced when the trip-time measurement frequency is low. Also due to the dynamic nature of the networks, there is inherent uncertainty about the states of the paths.

We propose to estimate the states of the paths with improved confidence by complementing the traffic evidences with the trip-time evidences using Dempster-Shafer (DS) theory of combination. DS theory uses a belief measure instead of a conventional probability measure to represent the confidence in the evidences and accounts for the uncertainty present due to the lack of a complete knowledge by considering the ignorance. The confidence in the overall decision is improved by combining multiple evidences when they are available. We

demonstrate the proposed framework in an interconnected network with LINUX machines configured as routers. *Routed*, dynamic routing daemon process, is modified to incorporate the DS rule of combination in deciding an optimal route. An experimental network is setup to have two EC paths to study the traffic distribution. We compare the performance with per-packet mode of traffic distribution.

Section 2 explains the DS framework for traffic adaptive routing. Section 3 discusses the experimental setup and the results. Section 4 concludes this research and provides direction for future research.

2. Framework for traffic adaptive routing

In this section we provide a brief overview of the DS theory of evidence and combination and present a DS framework for evidential routing.

2.1 Dempster-Shafer theory of evidence

Dempster-Shafer (DS) theory of evidence is a theory of uncertainty developed by Dempster [7] and extended by Shafer [8]. It gives us an opportunity to combine various available evidences and account for the ignorance in a decision system. DS theory has been studied to improve model-independent systems such as automatic traffic incident detection algorithm [9] in road transportation system by combining various evidences, medical diagnosis application [10], word recognition algorithm [11], automatic image segmentation algorithm [12] by combining information from multimodality images, for crop classification [13], for target identification system using multi-sensor data fusion [14] and improved document retrieval systems [15, 16].

DS framework consists of a frame of discernment Θ that contains a set of mutually exclusive and exhaustive hypothesis that represent the states of the system. It is similar to the universe of discourse in probability theory. For example, to represent the states of each available path, we select $\Theta = \{H_{Low}, H_{Medium}, H_{High}\}$. The hypotheses H_{Low}, H_{Medium} and H_{High} represent a network path in a state of light, medium and high path utilization respectively. Using the evidences acquired by the information sources, a probability like measure known as *mass m* is assigned to various subsets of Θ that measure the belief on the hypothesis as pointed by this evidence. The mass function *m* obeys the following properties.

$$0 \le m(H) \le 1.0$$

$$\sum_{H \subseteq \Theta} m(H) = 1.0$$

The confidence on each hypothesis is represented by a belief measure Bel(*H*), using the mass function as follows.

$$\mathrm{Bel}(H) = \sum_{A \subseteq H} m(A)$$

The belief measure Bel(*H*) identifies the least amount of belief placed on the hypothesis *H*. Similarly a plausibility measure defined as

$$\mathrm{Pls}(H) = \sum_{A \cap H \neq \phi} m(A)$$

measures the extent to which the evidences do not contradict the hypothesis *H*. Thus the confidence on any hypothesis *H* is represented by an uncertainty interval [Bel(*H*), Pls(*H*)]. For

example, a [1, 1] interval indicates absolute certainty about *H*, while [0, 1] interval indicates no certainty. This interval can be viewed as a lower and upper probability measures on *H*.

Further when multiple evidences become available, DS theory of combination aggregates them into a single body of evidences as follows.

$$\text{Bel}(H) = \sum_{A \cap B = H} m_1(A).m_2(B) \Bigg/ \left(1 - \sum_{A \cap B = \phi} m_1(A).m_2(B)\right)$$

where, m_i is the mass assigned using the i^{th} evidence or information source.

2.2. DS framework for traffic adaptive routing

The primary goal of the DS inference engine is to identify the network path utilization in approximate real-time to achieve traffic adaptive routing. We define the path-state hypotheses of low path-utilization (H_{Low}), medium path-utilization (H_{Medium}) and high path-utilization (H_{High}) to represent the states of the network paths. The traffic and trip-time measurements are used to assign belief measures to the hypotheses.

Table 1a. Traffic threshold

Traffic (Bytes)	Path state
< 20000	Low utilization
> 20000 and < 30000	Medium utilization
> 30000	High utilization

Table 1b. Trip-time threshold

Trip-time (μSec)	Path state
< 1500	Low utilization
> 1500	Medium or high utilization

The natural placement of the routers in the network paths allows the routers to measure the traffic in the path through them without additional network overheads. We used LINUX PCs as routers (by setting a '1' in */proc/net/ipv4/ip_forward* file) and measured the traffic from the Ethernet device statistics. We derived an empirical estimate of traffic thresholds for the experimental network shown in figure 2 to classify the traffic measured as an indicator of low, medium or high path-utilization. Table 1a shows the traffic thresholds used to classify the traffic measurements. Belief measures were assigned to the path-state hypotheses using the traffic measured as follows:

Let, N_{TR} – Number of traffic measurement samples obtained during a 30 second interval
X_{TR} – Number of low path-utilization samples observed during a 30 second interval
Y_{TR} – Number of medium path-utilization samples observed during a 30 second interval
Z_{TR} – Number of high path-utilization samples observed during a 30 second interval

$$m_1(\{H_{Low}\}) = X_{TR}/N_{TR}$$
$$m_1(\{H_{Medium}\}) = Y_{TR}/N_{TR}$$
$$m_1(\{H_{High}\}) = Z_{TR}/N_{TR}$$

Control packets referred as routing agents are used to measure the trip-time through the paths. Frequent trip-time measurements will increase the network overheads as opposed to the in-place traffic measurement in the routers. After several experimental runs, we derived trip-time threshold estimates to classify the network path-state as in $\{H_{Low}\}$ or $\{H_{Medium}, H_{High}\}$. Due to the limitation on the number of trip-time samples that can be obtained without increasing the network overhead (in terms of number of routing agents), it was observed that the trip-time measures could not clearly differentiate a medium and a high path-utilization. Therefore we

assign belief measures to the hypotheses combination $\{H_{Medium}, H_{High}\}$. Table 1b shows the trip-time thresholds used to categorize the trip-time measurements. Belief measures were assigned to the path-state hypotheses as follows.

Let, N_{TT} - Number of trip-time measurement samples obtained during a 30 second interval

X_{TT} - Number of low path-utilization samples observed during a 30 second interval

Y_{TT} - Number of medium or high path-utilization samples observed during a 30 second interval

$$m_2(\{H_{Low}\}) = X_{TT}/N_{TT}$$
$$m_2(\{H_{Medium}, H_{High}\}) = Y_{TT}/N_{TT}$$

The mass values computed using traffic and trip-time measurements are combined to derive the overall beliefs on the path-state hypothesis as follows.

$$m(\{H_{Low}\}) = m_1(\{H_{Low}\}).m_2(\{H_{Low}\})$$
$$m(\{H_{Medium}\}) = m_1(\{H_{Medium}\}).m_2(\{H_{Medium}, H_{High}\})$$
$$m(\{H_{High}\}) = m_1(\{H_{High}\}).m_2(\{H_{Medium}, H_{High}\})$$
$$m(\{\phi\}) = m_1(\{H_{Low}\}).m_2(\{H_{Medium}, H_{High}\}) + m_1(\{H_{Medium}\}).m_2(\{H_{Low}\}) + m_1(\{H_{High}\}).m_2(\{H_{Low}\})$$
$$\text{Bel}(\{H_{Low}\}) = m(\{H_{Low}\})/(1 - m(\{\phi\}))$$
$$\text{Bel}(\{H_{Medium}\}) = m(\{H_{Medium}\})/(1 - m(\{\phi\}))$$
$$\text{Bel}(\{H_{High}\}) = m(\{H_{High}\})/(1 - m(\{\phi\}))$$

Once the evidences are combined, the path-state hypothesis with highest belief measure was chosen as the probable state of the network path. Section 2.3 will explain how the path-state hypothesis assigned to the network path will be used in routing.

2.3. Evidential routing

LINUX IP layer uses a kernel routing table also known as forwarding information base (FIB) and a routing cache to choose the next routing point for a data packet. The dynamic routing daemon process *routed* implements a simple adaptive distance vector protocol that learns the network changes. We modified *routed* to: 1. Share the traffic measured at the router with the neighboring routers (every 30 seconds), 2. Combine the network traffic and the trip-time measurements using DS theory and identify the most probable states of the paths available from the router and 3. Update the FIB and the routing cache in the LINUX IP layer with the new routing decision based on the states of the network paths.

The LINUX IP stack was modified to evaluate the current path utilization to distribute the traffic among the available multiple EC paths to the destination. Detailed review about LINUX IP stack is available in [17, 18].

3. Experiment and results

Figure 2 shows the experimental network to study the performance of the proposed algorithm. R_1, R_2 and R_3 are the LINUX PCs configured as routers with the proposed changes to the IP stack and *routed*. C_{TG} is a traffic generator to create artificial traffic in the route through R_2. C_{TT1}, C_{TT2} and C_{TT3} are used to measure the trip-time in the paths from the sub-network net_A to the sub-network net_B through routers R_2 and R_3. To facilitate analysis, traffic to be routed is

originated from the computer C_S in net_A and destined at computer C_D in net_B. A constant traffic of 150kBps is routed through R_2. All the routers in the network are of same characteristics and the links were of equal capacity. From router R_1, routers R_2 and R_3 provide two EC paths to net_B. Router R_1 assesses the paths through R_2 and R_3 by combining their traffic and trip-time measurements every 30 seconds for adaptively routing the traffic originating from net_A. Table 2 shows the various possible combinations of states of the paths through routers R_2 and R_3 and the proportion of traffic routed through them from router R_1 during the next 30 second interval.

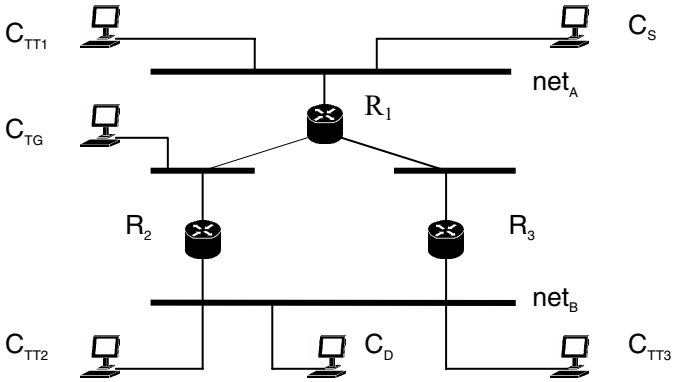

Figure 2. Experimental network

Let P_1 denote the network path from net_A to net_B through routers R_1 and R_2 and P_2 denote the network path from net_A to net_B through routers R_1 and R_3. The states of these paths are determined using the DS framework as explained earlier. When the paths P_1 and P_2 are under equal utilization, router R_1 equally distributes any new traffic from net_A, among the paths P_1 and P_2. When one of the paths, say P_1, is under higher utilization than the path P_2, router R_1 will route more data packets through P_2 during the next 30 seconds. The exact proportion of the traffic routed through each path is decided by the combination of the states of the paths P_1 and P_2 as shown in the table 2.

Table 2. Traffic distribution rules

State of network paths		Proportion of traffic routed from Router R_1	
P_1	P_2	Through Router R_2	Through Router R_3
$\{ H_{Low} \}$	$\{ H_{Low} \}$	1/2	1/2
$\{ H_{Low} \}$	$\{ H_{Medium} \}$ or $\{ H_{High} \}$	2/3	1/3
$\{ H_{Medium} \}$ or $\{ H_{High} \}$	$\{ H_{Low} \}$	1/3	2/3
$\{ H_{Medium} \}$	$\{ H_{Medium} \}$	1/2	1/2
$\{ H_{Medium} \}$	$\{ H_{High} \}$	2/3	1/3
$\{ H_{High} \}$	$\{ H_{Medium} \}$	1/3	2/3
$\{ H_{High} \}$	$\{ H_{High} \}$	1/2	1/2

The traffic generator C_{TG} sends 150kBps traffic to the sub-network net_B via router R_2 throughout the experiment. The purpose of this artificial traffic is to have different traffic conditions in paths P_1 and P_2. To study the performance of the proposed method at router R_1, an experimental traffic is sent from the source computer C_S in sub-network net_A to the

destination computer C_D in sub-network net_B. Figure 3 shows a typical traffic distribution among the paths P_1 and P_2 using the proposed method and the static per-packet mode of traffic distribution. It is evident that the proposed method provides a more equal traffic distribution among paths P_1 and P_2 indicated by a narrow traffic distribution curves.

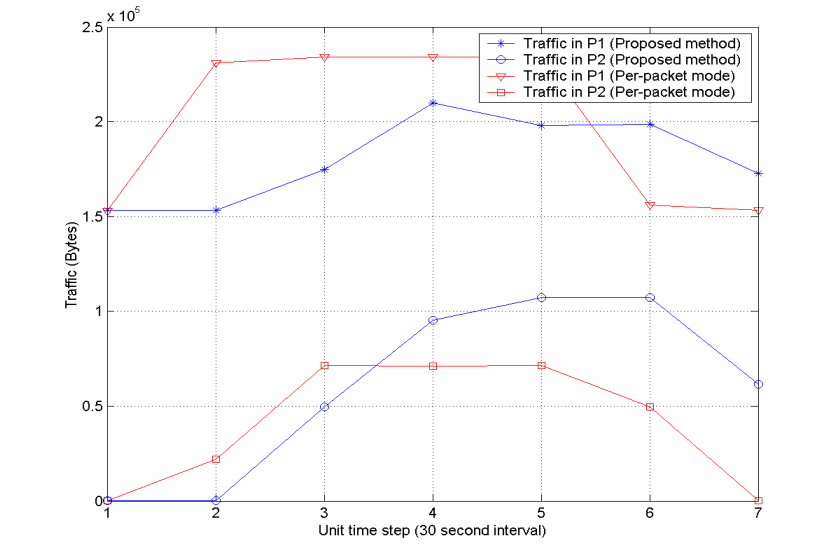

Figure 3. Traffic distribution among paths P_1 and P_2

As given in table 1a, traffic measurements less than 200000 bytes indicate low path utilization. Under steady state condition, path P_1 carries a traffic of approximately 150000 bytes per second (due to the artificial traffic) while path P_2 is at no load condition. Therefore both the paths P_1 and P_2 are identified to be in low path utilization state. Since both the paths are in the same path-states, router R_1 equally distributes the experimental traffic from the source computer C_S. Once the traffic in path P_1 reaches 200000 bytes (time step 4 in figure 3), path P_1 is identified to be in medium path utilization state while path P_2 continues to be in low utilization. Now the router R_1 routes $2/3^{rd}$ of the traffic through the path P_2 and $1/3^{rd}$ through the path P_1 using the traffic distribution rules given in table 2. Path P_2 continues to share a higher proportion of the traffic originating from net_A than path P_1 until both the paths come into the same path-utilization states. Once the traffic in the path P_1 comes under 200000 bytes both the paths P_1 and P_2 are in the state of low path utilization. Now the router R_1 equally distributes the traffic among P_1 and P_2 behaving like a per-packet mode of traffic distribution. An equal and adaptive routing can be achieved by properly altering the traffic and trip-time thresholds defined in table 1a and 1b.

4. Conclusion

A novel DS framework for traffic adaptive routing is developed. The result presented here demonstrates that the framework proposed is suitable to handle the dynamic nature of the network traffic and path conditions by combining the path-state evidences. The thresholds in tables 1a and 1b and the traffic distribution rules in table 2 were developed for assigning belief measures for the framework presented in this work. However, any information source that can assign belief measures to the path state hypotheses can be used in place of tables 1a and 1b and the traffic distribution rules can be changed accordingly. We demonstrated the framework in an experimental network as opposed to a simulation technique used in many dynamic traffic

routing algorithms. If more evidences such as packet drop, collision and retransmission rates can be acquired without increasing the network overheads, the confidence in the routes chosen for dynamic routing can be significantly improved by combining these additional evidences.

5. Acknowledgement

This publication was made possible through a financial support in part from NSF-JFAP fund with Dr.S.S. Iyengar, LA Board of Regents Health Excellence Fund HEF(2000-05)-01 and NIH grant number P20 RR16456. The contents presented in this publication are solely the responsibility of the authors and do not necessarily represent the official views of the funding agencies. Authors thank Dr. Elias G. Khalaf for arranging the hardware resources.

6. References

[1] C. Huitema, Routing in the Internet, Prentice Hall, Englewood Cliffs, NJ, 1995.
[2] Z. Wang, J. Crowcroft, "Analysis of Shortest-Path Routing Algorithms in a Dynamic Network Environment", Computer Communication Review, Vol. 22, No. 2, 1992, pp. 63-71.
[3] M. Balasubramanian, Dinakar Nethi, Gregory A. Smith, Louise A. Perkins, "Load Balancing through Adaptive Traffic Routing", AGEM Winter scholar symposium, Mississippi, 2002, pp. 16.
[4] Martin Heusse, Dylvain Guerin, Dominique Snyers, Pascale Kuntz, "Adaptive Agent-Driven Routing and Load Balancing in Communication Networks", Advances in Complex Systems, Vol. 1, N 2-3, 1998, pp. 237-234.
[5] I.N. Kassabalidis, M.A. El-Sharkawi, R.J. Marks II, P. Arabshahi, A.A. Gray, "Swarm Intelligence for routing in Communication Networks", IEEE Globecom 2001, pp. 3613-3617.
[6] J.M. McQuillan, I. Richer, E.C. Rosen, "The New Routing Algorithm for the ARPANET", IEEE Transactions on Communication, COM-28, pp. 711-719.
[7] A.P. Dempster, "Upper and Lower Probabilities Induced by Multivalued Mappings", Annals of Mathematical Statistics, Vol. 38, 1967, pp. 325-329.
[8] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, 1976.
[9] B. Ahn, S. Byun, D.B. Choi, "Traffic incident detection using evidential reasoning based data fusion", In proceedings of the 6th World Congress on Intelligent Transport Systems, Toronto, Canada, 1999.
[10] J. Yen, "GERTIS: a Dempster-Shafer approach to diagnosing hierarchical hypotheses", Communications of the ACM, Vol. 32, Issue 5, May 1989, pp. 573-585.
[11] R.L. Bowles, R.I. Damper, "Application of the Dempster-Shafer Theory of Evidence to Improved-Accuracy Isolated-Word Recognition", In proceedings of Eurospeech'89, pp. 384-387.
[12] M. Rombaut, Y.M. Zhu, "Study of Dempster-Shafer theory of image segmentation applications", Image and Vision Computing, Vol. 20, 2002, pp. 15-23.
[13] H. Ganster, A. Pinz, "Active Fusion using Dempster-Shafer Theory of Evidence", OEAGM/AAPR Workshop, 1996, pp. 37-46.
[14] P.L. Bogler, "Shafer-Dempster Reasoning with Applications to Multisensor Target Identification Systems", IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-17, No. 6, 1987, pp. 968-977.
[15] M. Lamas, I. Ruthven, "Representing and retrieving structured documents using Dempster-Shafer theory of evidence: Modeling and Evaluation", The Journal of Documentation, Vol. 54, No. 5, 1998, pp. 529-565.
[16] M. Lalmas, "Dempster-Shafer's Theory of Evidence applied to Structured Documents: modeling Uncertainty", ACM SIGIR Conference on Research and Development in Information Retrieval, PA, 1997, pp. 110-118.
[17] G. Herrin, "LINUX IP Networking: A Guide to the Implementation and Modification of the LINUX Protocol Stack", http://www.cs.unh.edu/cnrg/gherrin/linux-net.eps.
[18] S.T. Satchell, H.B.J. Clifford, LINUX IP Stacks Commentary, CoriolisOpen Press, 2000.