Computer Communications 26 (2003) 155–163
computer communications
www.elsevier.com/locate/comcom

Using Bayesian network on network tomography

Weiping Zhu

School of Computer Science, ADFA, The University of New South Wales, New South Wales ACT2600, Australia

Received 18 December 2001; revised 24 April 2002; accepted 1 May 2002

Abstract

Network tomography aims to obtain link-level performance characteristics, such as loss rate and average delay on each link, by end-to-end measurement. The obtained information can help us to understand the dynamic nature of networks. A number of methods have been proposed in recent years, which can be divided into two classes: multicast-based and unicast-based. In this paper, we propose an approach in the multicast class that uses the Bayesian network to carry out statistical inference. Simulations based on the network simulator 2 (ns2) were conducted, which shows our approach produced almost identical result as that produced by the maximum likelihood estimator previous proposed.

Keywords: Bayesian networks; Network tomography; Statistical inference

1. Introduction

To successfully design, control and manage networks, we must have a better understanding of network characteristics, in particular at the data link level [1]. Link level characteristics, such as packet loss rate and average delay of each link within a target network that can be a part of the Internet, can help us to understand the impact of aggregated traffic of various applications on transmission links. Knowing these, we can take necessary measures to improve network design, and if these characteristics can be obtained in real-time, we can even take actions to overcome related problems, i.e. network congestion. Two approaches can be used to obtain those information: (1) collecting statistics at the internal node; and (2) studying end-to-end communication patterns to evaluate internal network condition, so called network tomography. The first approach requires internal nodes, routers and/or switches, to dynamically collect the required data and deliver the data to interested parties, which not only requires to add extra workload and functionality to internal nodes, but also add extra traffic on networks. While, the second approach treats the target network as a *black box*, it adds probe packets from a node or nodes, called source, to the target network, those probes travels with the ongoing traffic, called background traffic later, to their destinations that can be a number of receivers. The paths from the source to the receivers consist of the target network. The arrivals and losses of the probe packets at the receivers indicate the traffic conditions on the corresponding links.

Several groups started to investigate methods to infer internal network condition based on end-to-end network measurements [2–6]. The methods proposed from those studies can be divided into two classes: multicast-based and unicast-based. A multicast-based method, as it named, multicasts probes from a source to all receivers based on a multicast tree that covers the target network on a periodical or exponential basis. The receivers record which probe arrived and which did not, and provide the data for further analysis. The analysis focuses on discovering correlations between receivers, these correlations shows the traffic conditions on related links. While, the unicast-based methods target those networks that do not support multicasting. Instead, they use packet-pair technique to create correlation between two receivers, two packets separated by ϵ in time sent from a source to two receivers that share a part of their paths from the source create correlation between the two receivers, then based on observations at the two receivers, similar techniques as those used in multicast-based methods are applied to identify traffic conditions on the shared path and not shared paths. Both approaches rely on statistical inference to obtain link-level characteristics.

In this paper, we propose to use the Bayesian networks (BN) to carry out the inference and use the Expectation–Maximization (EM) algorithm [18] to find out link-level characteristics. The EM algorithm is an iterative procedure

E-mail address: weiping@cs.adfa.edu.au (W. Zhu).

0140-3664/03/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0140-3664(02)00131-7

switching between Expectation and Maximization until it is converged. The BN is an active research area by itself that has been producing various methods to overcome the limitation of classical maximum likelihood estimator (MLE), such as, sparse data and overfitting [7]. In addition, many stochastic approximation methods, e.g. Markov chain Monte Carlo (MCMC) and bound and collapse [8], have been developed for fast convergence, which makes it a competitor to classic MLE. Over the last 20 years, especially in the last 10 years, the BN has become a popular representation for encoding uncertain knowledge in expert systems [9]. More recently, researchers have developed methods for learning the uncertain knowledge from data, which is a new and fast evolving technique, and more importantly, even from incomplete data, the parameters and structure of a BN can be derived. All these make the BN an ideal method to systematically discover hidden or missing information. Even without advance knowledge about traffic models, a prior can be assigned to each link on the basis of the Bayesian hypothesis. Then, using the data supplied from observations, a BN can identify link-level characteristics. Further, the BN is capable of identifying network structure from end-to-end measurement, which will free us from knowing the multicast structure in advance.

We carried out a preliminary study that used the network simulator 2, *ns2* [10], to simulate a network with 8 nodes and 7 links that connect the 8 nodes into a tree structure. A combination of TCP and UDP traffics are added to different nodes as background traffic. Probing packets are sent from the root to the 4 leaf nodes in a regular basis. By collecting the arrivals of probe packets at the 4 leaf nodes and using a BN to infer the link-level characteristics at the 7 links in terms of loss rate, we found the inferred result indicates the loss rate of the background traffic. In addition, we compared our results with the results obtained from using the MLE proposed in Refs. [6,11,12], and found after eliminating the abnormality of producing negative loss rate on some links from the MLE, the two results are almost identical. This reveals the feasibility and accuracy of our approach.

zThe rest of the paper is organized as follows. In Section 2, we focus on the related work and our contribution. In Section 3, we provide the fundamental of the BN and apply it to discover link-level characteristics by end-to-end measurement. We then present our study based on simulation in Section 4 that covers the details of traffics and compares the results obtained from incomplete observation with the actual data collected from the simulator. Section 5 is devoted to concluding remark, it also contain our current and future work in line of network tomography.

2. Related work and our contribution

Recently, Cáceres et al. successfully used the multicast-based approach to obtain data loss rate and average delay [6, 11,12]. They applied a classical MLE on the collected data to estimate the loss rates of the background traffic at each link. Both simulation and experiment study on the Mbone show the feasibility and potential of this approach. Meanwhile, Yajink et al. studied the temporal dependence of packet loss in Ref. [13]. They analyzed a 76 hours stationary trace and found the correlation timescale is approximately 1 second or less. Further, they used various 2^k-state Markov chain models to match the loss occurred in the trace, the results show higher order models tends to be more accurate than lower order ones.

In contrast to the above which assume an advance knowledge about the network structure for multicast, Ratnasamy and McCnne [14] proposed a way to discover the network structure using multicast-based loss estimator. They group multicast receivers and based on loss on the shared path between groups and/or between receivers in the same group to find path information. They had an algorithm for inferring binary trees and proposed an ad hoc approach for higher branching ratio trees.

More recently, Harfoush et al. proposed to use a unicast-based approach to discover link-level performance [15]. Their simulation confirms the usefulness of their method. Similarly, Coates and Nowak also used the packet-pair technique to estimate link-level characteristics. They used EM algorithm to estimate the correlation between packet pair, and then loss characteristics on related links [16].

The method we propose in this paper is a generic one that fits to any network, either multicast-enabled or multicast-disabled. The method uses the BN, called probabilistic network or belief network by some people, to infer the link-level characteristics. However, in the rest of the paper, our discussion is focused on multicast-enabled networks since inferring link-level characteristics by multicasting probes is more difficult than that using unicast. This is because a multicast tree can have multiple branches, and can be expended to multiple levels that create much more complicated correlation and dependency between nodes and links than that of a packet-pair approach, which only needs to consider the correlation between the two receivers. Subsequently, the inference used in our study is more complicated than [16]. In addition, based on observation, the BN can discover network structure. Therefore, The assumption of knowing the multicast tree structure in advance can be removed.

Our estimation process is repeated every time when new observations become available, which ensures the method to have a high adaptability since all the information from prior to the current observation is summarized into the information contained in the current observation.

3. Bayesian network and parameter learning

Formally, a BN is defined by a three-element tuple (X, S, P), where $X = \{X_1, X_2, \ldots, X_n\}$ is a set of variables, S

is the structure of the BN that defines the causal influence among the variables in X, the strength of these influences are quantified by the conditional probabilities, P.

The structure of a BN can be illustrated by a direct acyclic graph (DAG) in which each variable X_i, $i \in (1, \ldots, n)$ corresponds to a node, and an arrow (arc) from variable i to variable j denotes the cause–effect relationship between them, where i is a parent of j. A node can have a number of parents if the node is influenced by a number of variables (causes). The influences are measured by a set of conditional probabilities, $p(X_i|\mathbf{Pa}_i)$, for each state of a child variable X_i, given a configuration of its parent variables, denoted by $\mathbf{Pa}_i$. If a child and its parents are discrete variables, the conditional probability is represented by an $m \times n$ conditional probability table, where m is the number of possible configurations of the parents' states and n is the number of states of the child node. As stated, the structure S describes independence and dependence between variables. Given a sample of X, $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, the joint probability can be obtained by

$$p(X = \mathbf{x}) = \prod_{i=1}^{n} p(X_i = x_i|\mathbf{Pa_i} = \pi_{ij}), \tag{1}$$

where π_{ij} denotes the state of $\mathbf{Pa_i}$ in $\mathbf{x}$. If a node, say i, has no parent, then $p(X_i = x_i|\mathbf{Pa}_i = \pi_{ij}) = p(X_i = x_i)$. Formula (1) reveals that no matter how complicated a BN is, given the state of all nodes, the joint probability is represented by the product of conditional probabilities of all links. This decomposition eliminates the consideration of impact between independent variables, which significantly reduces the complexity of the calculation of the joint probability.

3.1. Learning link-level characteristics from observation

No matter which method, multicast-based or unicast-based, is used to send probe packets to receivers, the problem in nature can be solved by a BN. In fact, there is an 1-to-1 mapping between the BN used to infer link level characteristics and the network used to send probe packets (NPP). These two networks employ the same structure, each variable of the BN corresponds to a switch (router) of the NPP, similarly, each arc corresponds to a link, and the conditional probability of an arc corresponds to the link-level characteristics, such as the loss probability. Further, each probe sent to leaf nodes can be viewed as a *trial*, the result of a trial is a *configuration* of corresponding events occurred at each link, $d = \{d_1, d_2, \ldots, d_n\}$. If each link has only two possible states for a trial, 1 and 0, in which 1 denotes the link passed the probe, and 0 denotes the link did not pass the probe either because the probe did not reach the link or because the probe is lost on the link due to errors (including congestion). The complete sample space for a system with n links has 2^n possible configurations. If considering the structure described by the corresponding DAG, the number of possible configurations is substantially less than 2^n. For example, assume a network as shown in Fig. 2 has 4 nodes. When a probe is multicast from X_0 to X_2 and X_3, there are only 5 eligible configurations, which are {0,0,0}, {1,0,0}, {1,1,0}, {1,0,1}, and {1,1,1}, instead of $2^3 = 8$.

If we are able to obtain the states from all links for a large number of trials, we can either use classic MLE or the Bayesian method to find the conditional probabilities for all links. Statistically, this process can be represented by log-likelihood,

$$L(\Theta) = \log Pr\,(D; \Theta) = \log \prod_{d^i \in D} Pr(d^i; \Theta)$$

where D denotes a set of observations, $D = \{d^1, d^2, \ldots, d^m\}$, one for a trial; Θ denotes a set of parameters, $(\theta_1, \ldots, \theta_n)$, one for a link that determines the corresponding loss probability for the link. D can be further divided into groups according to configurations and each member of a group has the same configuration. Let $n(d)$ denote a function that returns the size of a group with configuration d. All eligible configurations form a configuration space, the space is represented by Ω_C. We aims to find Θ that can maximize $L(\Theta)$ from observations, i.e.:

$$\arg\max_{\Theta} L(\Theta) = \arg\max_{\Theta} \sum_{d \in \Omega_C} n(d) \log Pr(d; \Theta) \tag{2}$$

However, in practice there are many situations that prohibit us from having complete dataset, such as the network tomography problem in which the states of the internal links are not available, at least not directly. In some situations, we can uncover internal states easily from incomplete (partial) data. For instance, if a receiver receives a probe, we can conclude all links on the path from the source to the receiver must successfully pass the probe to their children. Nevertheless, it is not easy to infer the cause of a simultaneous loss, which refers to such a situation that a multicast probe is lost completely by all receivers attached to a subtree that has at least two descendent receivers. The difficulty is due to there are multiple causes that can lead to a simultaneous loss, which could be a simple loss at the link connecting the subtree to its parent, or a parallel loss at all links of a *cut* that can separate the subtree into two parts, in which one part consists of parents and ancestor of the other. For instance, Fig. 2 shows a simple subtree that has four nodes (X_0–X_3) and three links (l_1–l_3), where $(\{l_1\}, \{l_2, l_3\})$ are the two *cuts* of the subtree. Obviously, finding loss probabilities from the above situation is more difficult than that from complete dataset since we must consider the impacts from all possible causes that can lead to the same observation. Even though, a number of methods based on different principles, e.g. neural net, expectation–maximazation (EM), etc., have been developed to solve the problem.

In the following discussion, we will focus on the EM algorithm due to its popularity, and will use the capital letters to represent links and the corresponding lowercases for an assignment of the links, for instance, Z denotes a set

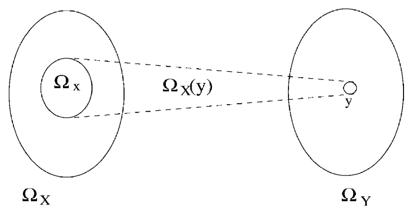

Fig. 1. The relationship between observable space Ω_Y and hidden space Ω_X, and an observation and its corresponding causes.

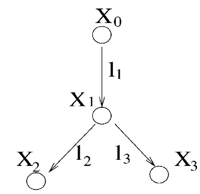

Fig. 2. A subtree.

of links, while z is an assignment of Z. The links that correspond to the multicast tree can be divided into two sets, let X be the set of hidden links, Y the set of observable links, and Ω_X and Ω_Y the set of values or sample spaces for these two sets, respectively. The product of Ω_X and Ω_Y can be thought of as the complete sample space. Correspondingly, (x, y) is a complete sample (dataset), and y is the observable part.

To learn loss probabilities from observation, y, we need first to identify all possible internal state configurations, Ω_x, that can produce y. Let $\Omega_X(y)$ be a function that maps an observation y to a set of internal configurations that are possible causes of y. Then, we have

$$\Omega_x = \Omega_X(y)$$

Fig. 1 illustrate the relationship among the symbols defined above. If the size of Ω_x, $|\Omega_x|$, equals to 1, the task is the same as we have complete data since x becomes redundant. If $|\Omega_x| > 1$, that means there are $|\Omega_x|$ internal state configurations that can lead to observation y. In this situation, we must consider the contribution of each configuration and we will use the conditional expectation of Ω_x on y, $E_{\Omega_x}(x|y)$, to estimate the probability of y. For example, assume a simultaneous loss occurred at the receivers attached to X_2 and X_3 of Fig. 2. Then, we have an observation $y = \{0,0\}$, and correspondingly $\Omega_x = (\{0\},\{1\})$, then,

$$p(y = \{0,0\}|\Theta) = \sum_{x\in\Omega_x} Pr(x, y|\Theta)$$

$$= Pr(\{0,0,0\}|\Theta) + Pr(\{1,0,0\}|\Theta) = Pr(l_1 = 0|\theta_1)$$

$$+Pr(l_1 = 1|\theta_1)Pr(l_2 = 0|\theta_2)Pr(l_3 = 0|\theta_3) \qquad (3)$$

where $Pr(d|\Theta)$, as previously defined, is the joint probability. The first term represents the loss occurred at link l_1, while the second term represents the parallel losses at l_2 and l_3. The two causes are related by formula (3), increasing one leads to the decrease of the other, a good example of 'explaining away' [17].

The EM algorithm is based on the same principle to estimate hidden states. Formally, it lets $f(x, y|\Theta)$ specify a family of functions, one of which governs the generation of complete samples, and let $g(y|\Theta)$ specify a family of functions one of which governs the generation of incomplete samples. f and g are related, if f is continuous, we have,

$$g(y|\Theta) = \int_{x\in\Omega_X} f(x, y|\Theta)\mathrm{d}x \qquad (4)$$

otherwise,

$$g(y|\Theta) = \sum_{x\in\Omega_X} f(x, y|\Theta)$$

To find an assignment of Θ that can maximize formula (2), the EM method does it by making essential use of the associated family $f(x, y|\Theta)$ [18], it is an iterative process that operates on the means (expected values) of hidden/missing variables, which are obtained from expected sufficient statistics. Mathematically, it can be described by the following equation;

$$\arg\max_{\Theta} L(\Theta) = \arg\max_{\Theta} \sum_{y\in\Omega_Y} n(y) \log g(y|\Theta)$$

$$= \arg\max_{\Theta} \sum_{y\in\Omega_Y} n(y) \log \sum_{x\in\Omega_X(y)} f(x, y|\Theta)$$

$$= \arg\max_{\Theta} E_{x|y}\Theta)] \qquad (5)$$

Eq. (5) shows a two-step process to update the loss probabilities, namely E-step and M-step. The E-step is the computation of the expectation of the unknown variables and the M-step maximizes the probability with respect to Θ, i.e.

$$Q(\Theta|\Theta') = \arg\max_{\Theta} L(\Theta|\Theta').$$

This is a monotonic increase process that is stopped when Θ is converged. The EM algorithm can be used to identify the loss probabilities in various BNs in which a variable can have multiple parents and multiple roots.

If multicast is used to send a probe to receivers, formula (4) turns to following form:

$$g(y|\Theta) = \prod_{y_i\in y} Pr(y_i|Pa(y_i), \theta_i)Pr(Pa(y_i)) \qquad (6)$$

where y is a binary vector of m elements, $y = (y_1, y_2, \ldots, y_m)$ that represents the observation at receivers. Each element corresponds to a receiver; if $y_i = 1$, $i \in [1, \ldots, m]$, receiver i received the probe, if $y_i = 0$ receiver i did not receive the probe. Since the probe is propagated through a tree structure, $Pa(y_I) \in x$ can be represented by the product of the loss probabilities of its ancestors, so do the ancestors except for the root which always has state 1. Therefore,

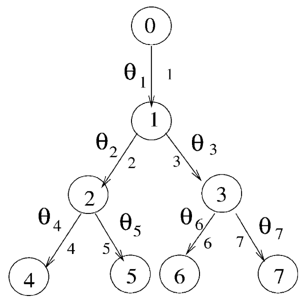

Fig. 3. Network structure.

$g(y|\Theta)$ equals to formula (1), but with hidden variables. If the distributions of these loss probabilities ($Pr(\cdot)$) exist, apart from selecting x, we need to select parameters (Θ) for the loss probabilities to maximize $g(y|\Theta)$, we can use the Bayesian inference to achieve this. If we do not know the form of the loss probabilities, the Bayesian learning, acting similar to classic methods identifies the physical probabilities instead.

Multicasting creates a strong correlation between parent and children, and between sibling brothers. If we can take advantage of those correlations, the learning process can be speedup since we can substantially reduce the number of internal configurations, i.e. $|\Omega_X|$. The two most important correlations are

- if a node receives a probe, its parent must receive the probe, i.e. $Pr(Pa_i = 1|X_i = 1) = 1$.
- if a receiver did not receive a probe but at least one of its sibling brothers did, we can conclude a loss occurs on the link from its parent to the receiver.

These two steps can greatly reduce the uncertainty from observations, and change the learning process from a system with hidden nodes to a system with missing data. In addition, our experience shows most simultaneous losses ($>99\%$) are created at the link connecting the corresponding subtree to the rest of the network. The reason is obvious, similar to the reliability analysis between parallel circuits and serial circuits.

4. Experimental result

To demonstrate its effectiveness of the Bayesian method, we conducted a series of tests on a simulation environment built on ns2 that has 8 nodes connected by 7 links, named 1–7, into a tree structure, as shown in Fig. 3. Link 1 had 3 Mbps of bandwidth, 2 ms of propagation delay; link 2 and 3 also had 3 Mbps of bandwidth, but 10 ms of propagation delay; the other 4 links had 1.5 Mbps of bandwidth and 10 ms propagation delay. All nodes has a FIFO queue and except node 1 has a queue with a limit of 20 packets, all other nodes can at most queue 10 packets at a time. The droptail policy is employed by all nodes to handle congestion, i.e. when a queue is full, newly arrived packets were dropped. Probe packets, 40 bytes each, were periodically multicasted from the root to the receivers attached to the leaf nodes. The background traffic consists of:

1. two TCP streams with window size = 50 and packet size = 1KB flow from node 0 to node 4 and 5, respectively;
2. one burst stream with burst period = 400 ms, idle period = 300 ms, bit_rate = 1000k, and packet size = 200 B flows from node 0 to node 4;

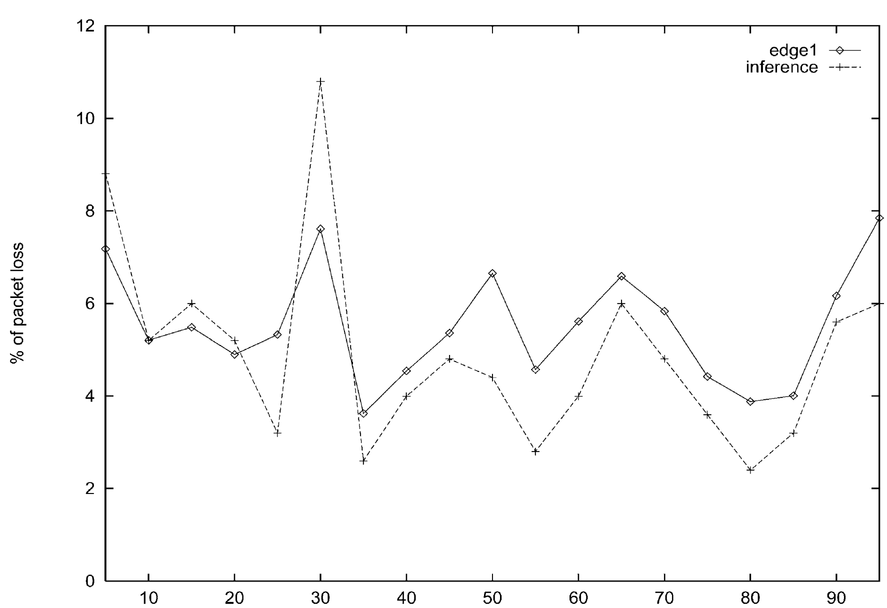

Fig. 4. Loss rate on link 1 with probe interval = 0.02 s.

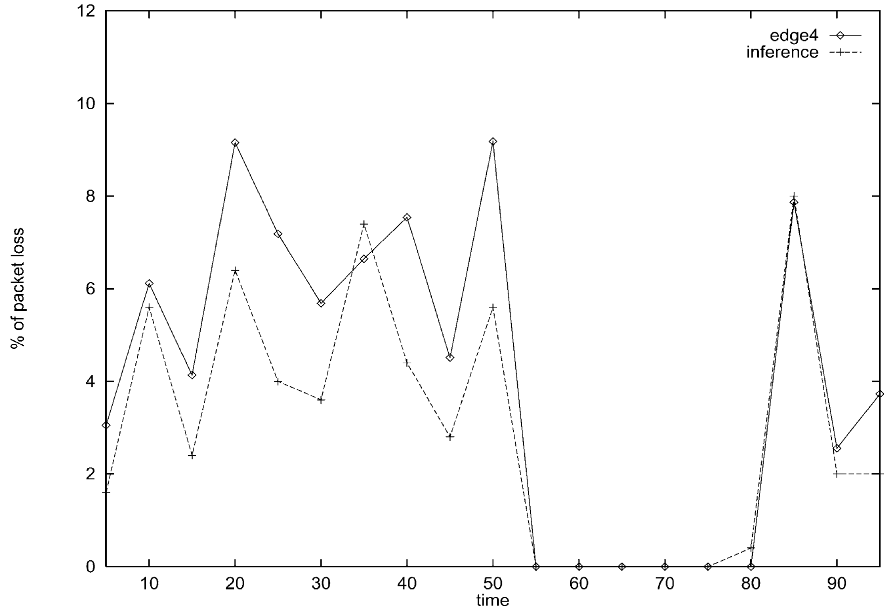

Fig. 5. Loss rate on link 4 with probe interval = 0.02 s.

3. one burst stream with burst period = 300 ms, idle period = 300 ms, bit_rate = 800k, and packet size = 200 B flows from node 0 to node 5;
4. one burst stream with burst period = 300 ms, idle period = 200 ms, bit_rate = 400k, and packet size = 500 B flows from node 1 to node 6;
5. one burst stream with burst period = 200 ms, idle period = 200 ms, bit_rate = 400k, and packet size = 500 B flows from node 1 to node 7;
6. one ftp stream flows from node 0 to node 4 with window size = 60 and packet size = 600; and
7. three ftp streams flow from node 0 to node 5, 6 and 7, respectively, with window size = 60, and packet size = 800.

where the burst periods and idle periods yield exponential distribution, and the numbers provided above are the means of the corresponding exponential distribution. Except 2 that started at 0 s and suspended at 50 s, and resumed at 80 s, all other streams started flow from 0 to 95 s.

What we were interested in this study is to find out the packets loss rate at each link segment by end-to-end measurement. θ_i, $i \in \{1, \ldots, 7\}$ in Fig. 3 represents the loss rate at link i. A multicast agent is added to the root node (0) to multicast probe packets on a regular basis to the 4 leaf nodes. A sequence number is attached to each probe packet, which enables a receiver to identify which probe packet is lost.

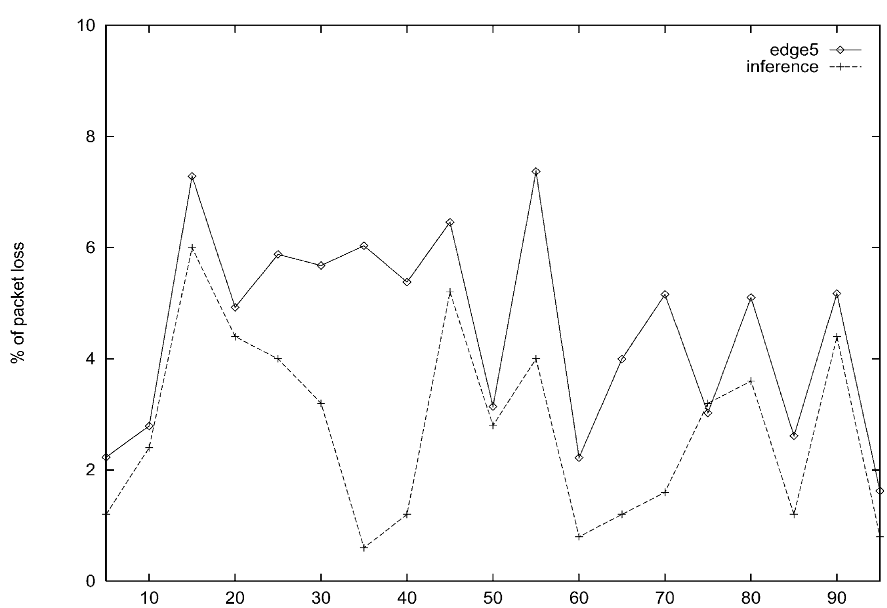

Fig. 6. Loss rate on link 5 with probe interval = 0.02 s.

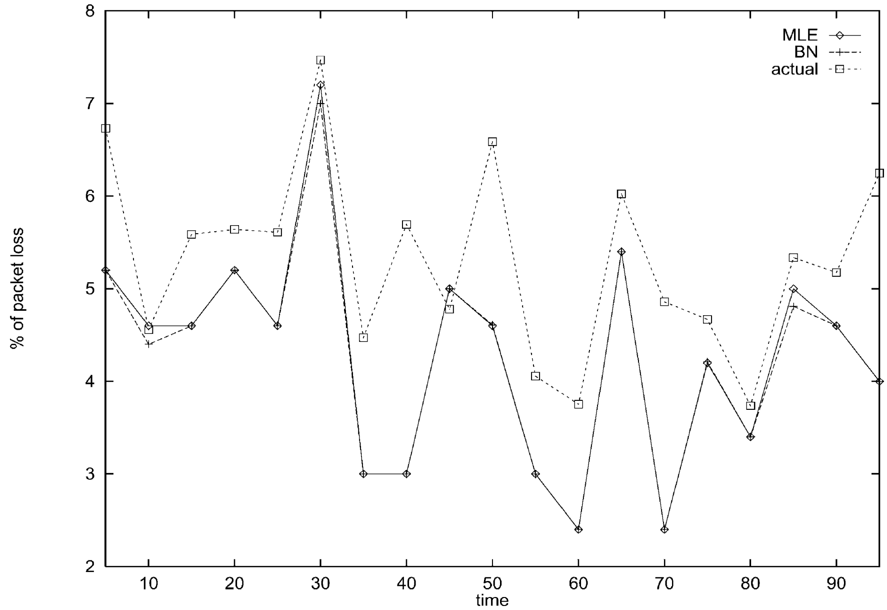

Fig. 7. Loss rate on link 1 and probe interval = 0.01 s.

Two set experiments were carried out on the simulation environment, the difference between them is the interval between two probe packets, one was 0.02s, and other was 0.01s. During the experiments, we conducted an inference every 5 s based on the data collected by the four receivers, the simulator uses the same interval to collect the actual link-level data, packet sent and dropped, at every node. We call the data collected by the simulator actual result.

In the two set experiments, the inferred results on link 2, 3, 6 and 7 match the actual results perfectly since these links were lightly loaded. Figs. 4–6 shows the difference between inferred result and actual result on the other 3 links, i.e. link1, link4 and link5, respectively. Although there are some differences between the inferred and actual results in Figs. 4 and 5, the inferred results correctly show the loss trend of the background traffic, in particular, when stream 2 was suspended. However, for link 5, there are clear differences between the inferred result and the actual result in two time ranges, i.e. [25:40] and [60:70]. We investigated the cause and found these differences were not caused by inference but the data collected by the receivers, the collected data is very different from the actual data. In

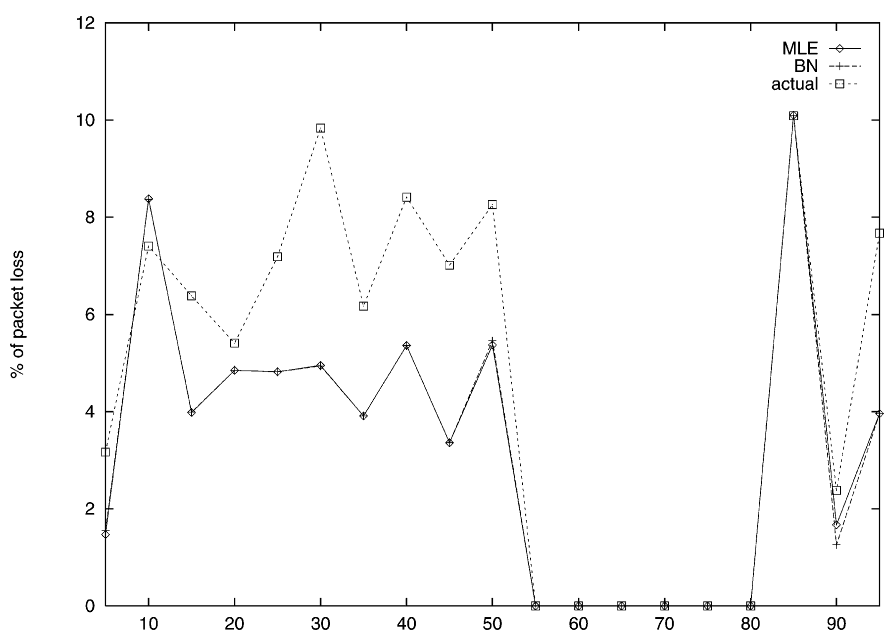

Fig. 8. Loss rate on link 4 with probe interval = 0.01 s.

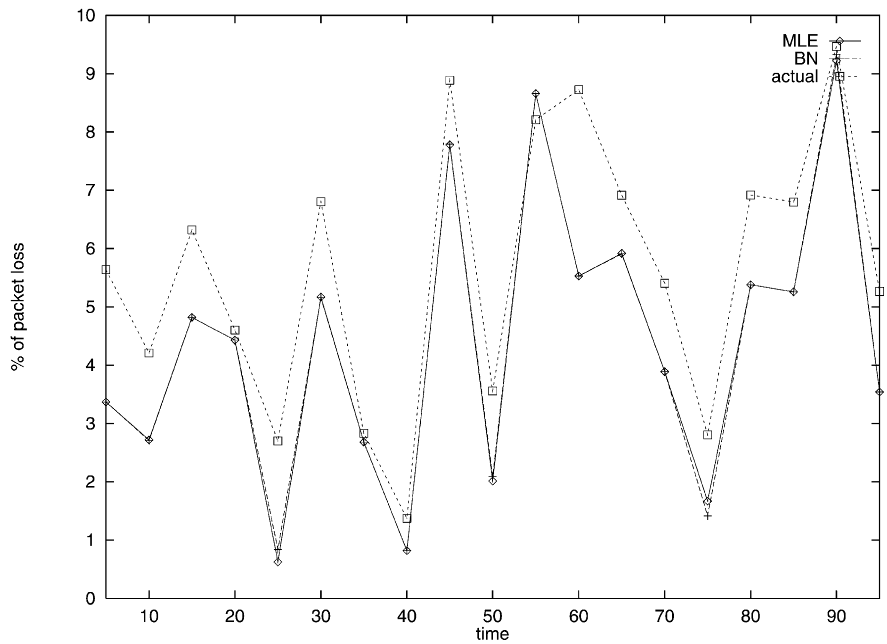

Fig. 9. Loss rate on link 5 with probe interval = 0.01 s.

terms of statistics, the sample collected (probe) in these two periods is inconsistent with population. There is only a small portion of the probe packets, compared with background traffic, that were lost in these periods. This suggests us to increase sampling frequency. We then carried out another round simulations in which the probe interval was reduced from 0.02 to 0.01 s. The second round results on these three links are shown in Figs. 7–9, respectively. To compare with the MLE proposed in Refs. [6,11,12], we included the results produced by using the MLE to infer the loss rates in these figures. As shown these two methods produce almost identical results. However, due to overfitting the MLE constantly produced pass rate >1 for link 2 (loss rate $= 1 -$ pass rate), the results presented above are the results after we corrected those errors. In addition, this shows our approach share all statistical features listed in Refs. [6,11, 12], including convergence to actual loss rate given a large number of samples, and agreement with theoretical intervals. In general, the second round results are more accurate than the first round. Comparing Figs. 6 and 9, one is able to see the obvious difference.

We observed another interesting feature in these two round tests, which shows the reaction of the background streams to traffic increase. Although these two round experiments had the same background traffic, their curves have noticeable differences, which shows the sensitivity of TCP flow control to the slightly traffic increase, i.e. TCP streams reduced their window sizes immediately when traffic becomes heavier that makes the curves of actual results different in these two set experiments. This sensitivity makes it hard to have a model to describe loss characteristics, such as Bernoulli. In fact, based on our observation, using *nam*, we found packet loss has a strong temporal relation, e.g. if a packet is lost at a link, the next packet on the link has a higher probability to be lost than a random selected packet.

5. Conclusion

In this paper, we present a new approach to infer the link-level characteristics by end-to-end measurement. The inference method is built on the BN, which maps the network tomography problem into a BN learning problem. By exploiting the learning ability of a BN on discovering uncertain/hidden knowledge from incomplete data, the link-level characteristics can be uncovered. The initial result presented in this paper shows the Bayesian based method can produce reasonable accurate results.

We believe this technique is an addition to the current study of network tomography. The proposed method as the MLE proposed by MINC [19] can be used in either multicast-enabled or multicast-disabled networks; however, the proposed method uses a different inference technique that relies on the BN which is mathematically sound and attractive. The attractive lies on its ability of learning from observations for both link-level characteristics and network structure. In this paper, we demonstrated the first ability in inferring loss rate in a balanced tree. However, much work remains to be done in this direction. We are investigating the methods to learn structure from data and methods other than multicast to obtain samples about background traffic.

References

[1] V. Paxson, End-to-end internet packet dynamics, IEEE/ACM Transactions on Networking 7 (3) (1999).

[2] Felix: Independent monitoring for network survivality, in ftp://ftp.bellcore.com/pub/mwg/felix/index.html.

[3] IPMA: Internet performance measurement and analysis, in http://www.merit.edu/ipma.

[4] J. Mahdavi, V. Paxson, A. Adams, M. Mathis, Creating a scalable architecture for Internet measurement, INET'98.

[5] Surveyor, in http://io.advanced.org/surveyor.

[6] R. Caceres, N.G. Duffield, J. Horowitz, D. Towsley, Multicast-based inference of network-internal loss characteristics, IEEE Transactions on Information Theory (1999) 45.

[7] W. Buntine, A guide to the literature on learning probabilistic networks from data, IEEE Transactions on Knowledge and Data Engineering 8 (2) (1996).

[8] M. Ramoni, P. Sebastiani. Efficient parameter learning in Bayesian networks from incomplete databases, Report KMI-TR-41, The Open University, 1997.

[9] D. Heckerman, A tutorial on learning with Bayesian networks, Technical Report MSR-TR-95-06, 1995.

[10] The network simulator 2, in http://www.isi.edu/nsnam/ns2.

[11] R. Caceres, N.G. Duffield, S.B. Moon, D. Towsley, Inference of internal loss rates in the mbone, IEEE/ISOC Global Internet'99 (1999) 99.

[12] R. Caceres, N.G. Duffield, S.B. Moon, D. Towsley, Inferring link-level performance from end-to-end multicast measurements, Technical Report, 1999.

[13] M. Yajnik, S. Moon, J. Kurose, D. Towsley, Measurement and modeling of the temporal dependence in packet loss, IEEE Infocom (1999).

[14] S. Ratnasamy, S. McCanne, Inference of multicast routing tree topologies and bottleneck bandwidths using end-to-end measurements, IEEE Infocom'99, New York (1999).

[15] K. Harfoush, A. Bestavros, J. Byers, Robust identification of shared losses using end-to-end unicast probes, Technical Report BUCS-2000-013, Boston University, 2000.

[16] M. Coates, R. Nowak, Unicast network tomography using EM algorithms, Technical Report TR-0004, Rice University, September 2000.

[17] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[18] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, B (1977).

[19] N. Duffield, V. Paxson, D. Towsley, MINC: Multicast-based inference of network-internal characteristics, in http://www.research.att.com/duffield/minc/.