7

Research on Network Tomography Measurement Technique

Cai Wandong, Yao Ye and Li Yongjun
School of Computer Science, Northwestern Polytechnical University, Xi'an City, 710072, China

1. Introduction

Network measurement depends on certain measurement method, technique and standard to obtain measurement sample based on measurement devices or tools, which applies the network performance analysis model to identify network topology architecture, and to infer performance parameter and traffic characteristics that provides the scientific decision for network resources optimization deployment, network management, failure point position, and so on[1~3]. For the wired network with solid infrastructures, such as Internet, it often adopts a interior direct measurement method that is also defined as traditional measurement technique in the chapter.

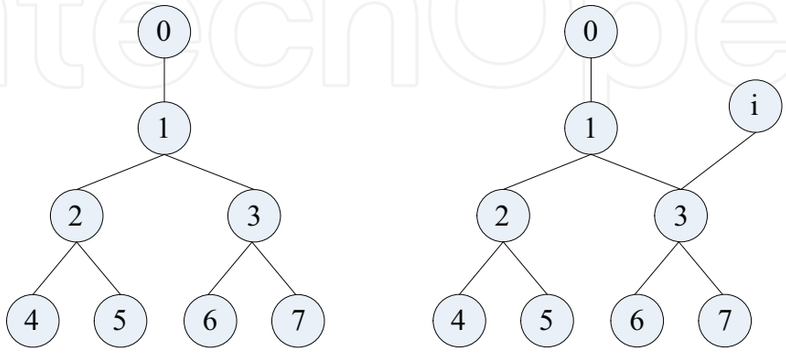
During the middle period of 90 years in last century, NT measurement technique was brought forward by Y. Vardi[4],which used the end-to-end measurement sample to infer network link performance parameters. Traditional network measurement technique is often applied in Internet with solid infrastructure, which does not need the interior nodes to collaborate with each other in the same autonomous area, but requires some IP network standard protocols to help, such as SNMP, ICMP, and so on. NT measurement technique could adopt the active or passive measurement method, and analyzes statistically the end-to-end network performance sample to infer link performance parameters, topology architecture or traffic characteristics. The objective of NT measurement technique mainly focuses on link delay or loss rate inference, link bandwidth and throughput inference, network topology architecture identification and traffic matrix estimate[6~13].

The measurement process in NT technique consists of three steps[14,15]. At first, measurement system model must be built on, including measurement topology model and performance analysis model, which generally adopts logic tree network topology model, and makes use of the relationship between nodes in measurement topology model and packet transmission behavior to build on performance analysis model. Secondly, active or passive measurement method is used to obtain the end-to-end measurement sample, then to evaluate the temporal and spacial independence of measurement sample. At last, the mathematics and statistics theory are used to analyze and evaluate the measurement sample based on performance analysis model to infer link performance or to identify topology architecture, etc.

1.1 NT measurement topolgy model

Measurement topology model is the basis on NT measurement technique. If the number of source node which has the chance to send measurement probes, is only one in measurement

process, that of leaf nodes collecting measurement sample is more than one, where exists one-to-many relationship between source node and leaf nodes, this measurement style is often called as single-source measurement model, and often uses tree topology measurement model to descript as the figure 1(a). Otherwise, if the source nodes and leaf nodes exit many-to-many relationship, this measurement style is generally called as multi-source measurement and often uses the non-loop graph topology measurement model to descipt as the figure 1(b).



(a) single-source measurement system model (b) multi-source measurement system model

Fig. 1. NT measurement topology model

In the tree topology measurement model, Let T=(V, L) denotes a reverse tree with the node set V and link set L. V could be finely classified as $V = \{S, M, R\}$, where S denotes the set of source nodes, M the set of interior forwarding nodes and R the set of leaf nodes(or receiver nodes). As in figure 1(a), $S = \{0\}$, Because there is only one source node to send the probes. However, leaf nodes 4,5,6,7 has the chance to collect the measurement sample. The link set contains ordered pairs (i, j) such that node i sends its data to node j directly, destined for the leaf node r($r \in R$). The link (i, j) is simply denoted by $l_{i,j}$ ($l_{i,j} \in L$). Howerver, the path from the node i to j is denoted by $P_{i,j}$, Let f(i) denote the father set of the node i. The ancestor set of node i could be denoted as: $F(i) = \{f^1(i), f^2(i), ..., f^n(i) \mid f^n(i) \in S\}$, noted that there exists the following rules: $f^0(i) = i$, $f^1(i) = f(i)$ and $f^n(i) = f(f^{n-1}(i))(n \geq 1)$. In the multi-source measurement model as in figure 1(b), there are more than one source nodes which has the chance to send probes, such as $S = \{0, i\}$. If the number of source nodes and leaf nodes are M and N respectively, the network architecture in multi-source measurement is called as M-by-N topology architecture[16].

1.2 NT measurement analysis model

NT measurement analysis model mainly consists of performance analysis model and network topology architecture identification model, the former focuses on link loss rate and delay inference, and the latter on topology architecture identification.

1.2.1 Link loss rate analysis model

It is to use the mathematical method to describe the relationship between the link and path performance. For example, Bernoulli model[17,18] and Gilbert model[19,20] are often used

in link loss rate inference. The former supposes that the loss of packets in one mobile node is independent of each other, which actually is a Bernoulli stochastic process. Stochastic process $X=(x_r)$ ($r \in R$) is used to describe state of the leaf node r receiving probes, $x_r=1$ denotes node r receiving a probe, otherwise $x_r=0$. For the N probes, the receiving state of leaf node r could be denoted as $X_r=\{x_r^{(n)}\}$ ($1 \le n \le N$). If the link loss rate parameter is presented as $\alpha=(\alpha_{l_r})(l_r \in L)$, where α_r is the loss rate of link l_r , the aim of Bernoulli model is to obtain the maximum pre estimate: $\alpha^*=\arg Max_\alpha P(X_{r\in R}\mid \alpha,T)$. However, the latter considers that there exits time dependence correlation between the consecutive probes. For instance, if the probe with sequence one is lost in one mobile node, the probability of probe with sequence two in the same mobile node being lost is higher. Gilbert model uses two states Markov process to describe this temporal dependence, 1 denotes probe loss and 0 not loss. In Gilbert model as in figure 2, p denotes that the probability of current probe is not lost where the one after which is lost, while q denotes that the probability of current probe is lost where the one after which is not lost. If $p+q=1$ is satisfied, Gilbert model could be changed into Bernoulli model.

1.2.2 Link delay analysis model

In link delay anlysis model, we often suppose that the system clocks in each nodes are synchronous, and discrete delay mode and continuous delay one are often used. In general, the discrete delay model adopts the discrete time method to study the probability distribution of link delay based on NT. However, the continuous delay time model often uses the cumulate generating function (abbreviated as CGF) to infer link delay parameters. Owing to using the logarithmic operation in CGF for its un-linear correlation, there exists some variances in the inference result, and even sometimes the variance is high. In order to reduce and correct the variance, Yolanda et al. [21] adopts a linear optimization method to correct the variance estimation of inference results. Network delay includes the fixed delay time and variational one, the sending delay(T_t)and transmission one (T_g) composes the former, and the process delay(Tp) and queuing delay (Tq)the latter. Link delay analysis model could be presented as the formula 1, where m is the number of link, $T_{t,0}$,$T_{g,0}$ denotes the sending delay of source node and transmission delay of the first link respectively.

$$Delay = T_{t,0}+T_{g,0}+\sum_{n=1}^{m}(T_{t,n}+T_{g,n}+T_{p,n}+T_{q,n})+T_{q,d} \tag{1}$$

1.2.3 Network topology inference analysis model

Network topology inference analysis model is founded on the basis of the following hypothesis, that the correlative degree between brother nodes is stronger than that between non- brother nodes. [22,23] bring forth a bias relationship of probe receiving to infer network topology architecture, which defined a hamming distance of probes receiving between node i and j as the formula 2. where n is the number of measurement.

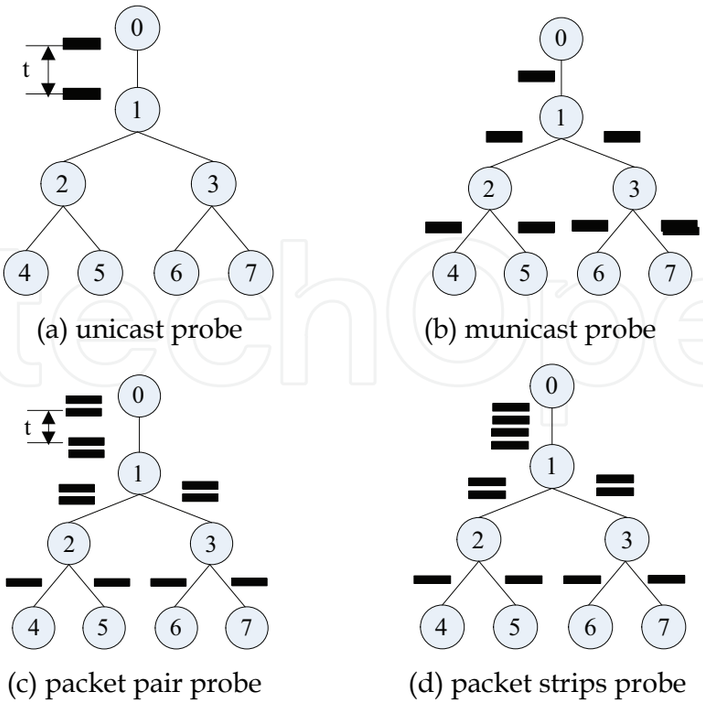
$$d(i,j)=\sum_{m=1}^{n}(x_i^m \oplus x_j^m), i,j \in V \tag{2}$$

If $d(i,j) < \varepsilon$ is satisfied, node i and j are deemed to brother node, and ε is a liminal value. Therefore, network topology architecture could be inferred through computing the $d(i,j)$ between nodes, which is a bin-tree architecture. However, a tree topology architecture could be inferred by expanding the method above.

1.3 NT measurement probes

unicast probe Unicast probe is transmitted by the source node to the leaf nodes according to a certain sample rule as in figure 2(a). Link loss rate and delay could be inferred on the basis of the number of unicast probes and that the leaf node receiving, end-to-end delay, and so on. Owing to unicast communication is supported by many networks, the merit of unicast probe is its broad application scope. Although the interval between unicast probes accords with a certain sample rule, which could reduce the influence brought by active measurement in a certain extent, it will destroy the correlation of the two conterminous unicast probes and reduce the precision of measurement. As in figure 2(a), the source node 0 sends unicast probe, since the leaf node 3 and 4 receive unicast probe dependently, if the node 3 receive a unicast probe, but node 4 not, it is difficult to judge where the unicast probe is lost.

multi-cast probe In order to settle the limitation of unicast probe, the multicast probe is put forward in network measurement. As in figure 2(b), the source node 0 transmits the multicast probe to a group of leaf nodes, such as node 4,5,6 and 7. Since the multicast probes have the same communication characteristic in the shared path, it will resolve the problem the correlation of probes and improve the precision of measurement. If node 3 receives the multicast probe, but node 4 not, it is easy to infer that the probe is lost in the link l_4. Of course, there are much limitation on munical probes, one is that some network devices,



(a) unicast probe (b) munical probe

(c) packet pair probe (d) packet strips probe

Fig. 2. NT measurement probes

such as switcher and router, do not support or configure multicast communication protocols, which will influent its application scope, another is some network devices adopt difference process method on unicast and multicast, which will also affect the measurement precison in some extent.

packet pair probe Nowak Robert et al. brings forth to using packet pair to measurement network performance as in figure 2(c). Packet pair comprised of two unicast probes with small interval, which is smaller than that between different packet pairs. In figure 2(c), source node 0 sends one packet pair to node 3 and 4, if the first unicast probe arrives at node 3 successfully, the we could safely guess that the probability of node 4 receiving the second unicast probe is near to 100%. Therefore, packet pair not only has the properties of multicast probe, but also extends the application scope of unicast probe. However, packet pair only takes into account the correlation between unicast probe, it is just used for bin-tree measurement analysis model.

packet stripes probe In order to resolve the limitation of packet pair, N.G. Duffield introduce the packet strips into network measurement, which extends the number of unicast probes from two to many as in figure 2(d). From the other point of view, the packet strip could be considered as many packet pairs, which supports the different packet pairs with correlation in the shared path. However, when the number of unicast probes is more enough, packet strip could be changed as unicast probes.

1.4 NT measurement inference method

NT measurement inference method is to use end-to-end network performance measurement sample to infer the probability distribution of link performance based on measurement analysis model and performance analysis model, which mainly composed of Maximum Likelihood Estimate(MLE), Expectation Maximization method(EM) and Bayesian estimate.

Maximum Likelihood Estimate Method MLE[24] is one of the elementary method on parameter estimate, which supposes that link performance parameter accords with distribution $f(X;\Theta)$, where $\Theta=(\theta_1,\theta_2,\cdots,\theta_n)$ is the estimated parameter. If end-to-end measurement sample is denoted as $\{y_1,y_2,\cdots,y_n\}$, supposing that they follows the same distribution rule independently, the distribution function of path performance parameter Y could be expressed as $Y=p(Y;\Theta)$, then the pseudo function follows the formula 3

$$L(Y;\Theta)=\prod_{i=1}^{n}p(y_i;\Theta) \tag{3}$$

The objective of MLE is to find the value of the parameter Θ when $L(Y;\Theta)$ obtains its maximum value, which could be denoted as $\hat{\Theta}=\arg MaxL(Y;\Theta)$. Nevertheless, it is difficult to find the transcendent distribution function $f(X;\Theta)$ of network link performance parameter X. Even though it was founded, there are high computing complexity degree of pseudo parameter estimate for the complexity of pseudo function with large network scale.

Expectation Maximization method EM algorithm[25,26] is to use partial measurement sample to infer maximum pseudo value of link performance distribution function, including two procedures, that is, E-step and M-step. The main problem about EM algorithm is that it could obtain the partially optimized solution, not the unitary optimized one. For the sake of computing complexity increasing by the scale of network, Pseudo-EM Algorithm[4] decomposes a large scale problem to several small scale ones. The maximum likelihood of

these small scale problems could be expressed as formula 4, where S is set of all small scale problems.

$$L\ (Y_1, Y_2, ..., Y_n; X) = \prod_{i=1}^{n} \prod_{s \in S} P^s(Y_i^s; X^s) \tag{4}$$

Bayesian estimate method It uses the transcendent probability distribution of link performance to infer the posterior one. However, how to get the former probability distribution is a difficult work. It is also difficult for Bayesian estimate method to obtain the link performance parameter with large network scale for its computing complexity. In order to solve this problem, Markov Chain Monte Carlo method is brought forth to infer link performance parameters by using Gibbs and Metropolis-Hasting sample rule based on Bernoulli and Gilbert probability model[27].

In short, MLE and Bayesian estimate methods needs to know the transcendent distribution, but it is very difficult to obtain in practice. EM resolves the problem of computing the estimated parameter of network link performance in math, but it is easy to converge on a partially optimized solution.

2. NT measurement technique in WSNs

Recent technological advances have made the development of low cost sensor nodes possible, and this allows the deployment of the large-scale sensor network to be feasible. The accurate network performance plays an important role in the successful design, deployment and management of sensor networks. However, the inherent stringent bandwidth and energy constraints of sensors create challenging problems in the network performance measurement. Motivated by the needs of accurate sensor network performance measurement and the inherent constraint of sensor network, in this section, we concentrate on: (1) the problem of efficiently estimating the internal link loss Cumulant Generating Function (CGF); (2) the problem of efficiently estimating the internal link loss rate from the passive end-to-end measurement.

There has been much research in the field of network tomography for the wireless sensor network .In [28], Li et al. proposed a simple method based on the hamming distance of sequences on receipt/loss of aggregated data between each pair of parent-child node to identify the lossy nodes. Under the assumptions that the link losses are mutually independent, Li et al. [29] formulated the problem of link loss estimation as a Bayesian inference problem and propose a Markov Chain Monte Carlo algorithm to inferring the internal link loss characteristics from passive end-to-end measurement. In [30] this problem was formulated as a Maximum-Likelihood Estimation problem and used the Expectation-Maximization algorithm to solve it. Almost existed methods used the iterative approximating approach to estimate the loss rate that requires a long execution time. In addition, iterative approach may trap into a local maximum. To overcome this problem, a simple up-bottom approach [31] and a bottom-up [32] to estimate loss rate in wireless sensor network were proposed, which identifies parameters of loss probability model based on the observations collected in the sink node. Knowledge of sensor network topology is a crucial component of sensor network tomography techniques. Based on the partial ordering relation on the packet receipt/loss between a node and its descendant nodes in the data aggregation process, Li et al. [33][34] formulated the problem of sensor network topology identification as a topological sorting problem and proposed a topological sorting algorithm

to solve it. In [35], an algorithm that named hamming distance and hop count based classification algorithm (HHC), to infer network topology by using end-to-end data in sensor network.

2.1 Loss cumulate generating function inference method

Each link loss CGF preserves all the statistical information of the loss since it is the log of the Fourier transform of the link loss probability density function. We can accurately infer many features of the link loss distribution from loss CGF[36].

2.1.1 Cumulate generating function

We suppose the link losses X_i are mutually independent, $i = 1,\cdots,n$. Define the end-to-end loss cumulate generating function (CGF) of the path i $K_{Y_i} = \log E\left[e^{tY_i}\right]$ and the link loss CGF $K_{X_i} = \log E\left[e^{tX_i}\right]$, with CGF parameter t, $t \in (-\infty,\infty)$. The CGF of $\boldsymbol{Y}$ can therefore be expressed as

$$\begin{aligned} K_{Y_i}(t) &= \log E\left[e^{tY_i}\right] = \log E\left[e^{t(\sum j\in M_i X_j)}\right] = \log\left\{\prod_{j\in M_i} E\left[e^{tX_j}\right]\right\} \\ &= \sum_{j\in M_i} \log E\left[e^{tX_j}\right] = \sum_{j=1}^{m} a_{ij}\cdot K_{X_j}(t) = A_{(i)}\cdot K_X(t) \end{aligned} \tag{5}$$

where $A_{(i)}$ denotes the *ith* row of the matrix A and $K_X(t) = \left[K_{X_1}(t),\cdots,K_{X_n}(t)\right]^T$ (^{T} denotes transpose). Thus the vector of end-to-end CGF's $K_Y(t) = \left[K_{Y_1}(t),\cdots,K_{Y_n}(t)\right]^T$ can be expressed by the following linear relation

$$K_Y(t) = A\cdot K_X(t) \tag{6}$$

There are n links and n paths in the sensor network, so the matrix A is full rank. The relation (2) is invertible and the link loss CGF $K_X(t)$ can be determined from the end-to-end loss CGF $K_Y(t)$ as the following equation

$$K_X(t) = \left(A^T A\right)^{-1} A^T K_Y(t). \tag{7}$$

Let $B = \left(A^T A\right)^{-1} A^T$, then we have

$$K_{X_j}(t) = \sum_{i=1}^{n} b_{ji}\cdot K_{Y_i}(t). \tag{8}$$

Define the end-to-end loss moment generating function (MGF) of path i $M_{Y_i} = E\left[e^{tY_i}\right]$, and the loss MCF of link i $M_{X_i} = E\left[e^{tX_i}\right]$. Similarly, we can get the relationship between M_{X_i} and M_{Y_i} $M_{X_j}(t) = \sum_{i=1}^{n} b_{ji}\cdot M_{Y_i}(t)$.

2.1.2 Loss CGF inference

Let N be the number of data collection trial, then the estimated value of M_{Y_i} can be obtained using the following equation

$$\hat{M}_{Y_i}(t) = \frac{1}{N}\sum_{k=1}^{N} e^{tY_i^k} \tag{9}$$

where Y_i^k is the end-to-end loss of path i in the kth data collection trial. We obtain estimates of the vector $K_X(t)$ from $\hat{M}_Y(t) = \left[\hat{M}_{Y_1}(t), \cdots, \hat{M}_{Y_n}(t)\right]^T$. Note that $\hat{M}_{Y_i}$ is an unbiased estimate of the MGF M_{Y_i}. According to equation (8), we have

$$\hat{K}'_{X_j} = \sum_{i=1}^{n} b_{ji} \cdot \log\left(\hat{M}_{Y_i}(t)\right). \tag{10}$$

As mention in [37], $\hat{K}'_{X_j}$ is biased estimate of K_{X_i} due to non-linearity of the log. We apply the technique adopted in [37] to obtain a bias corrected estimator for K_{X_i}.

$$\begin{aligned}
&\hat{K}'_{X_j} = \sum_{i=1}^{n} b_{ji} \cdot \log\left(\hat{M}_{Y_i}(t)\right) \\
&= \log\left\{\Pi_{i=1}^{n}\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}\right\} \\
&= \log\left\{\Pi_{i=1}^{n} E^{b_{ji}}\left[\hat{M}_{Y_i}(t)\right] - \left(\Pi_{i=1}^{n} E^{b_{ji}}\left[\hat{M}_{Y_i}(t)\right] - \Pi_{i=1}^{n}\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}\right)\right\} \\
&= \log\left\{\Pi_{i=1}^{n} E^{b_{ji}}\left[\hat{M}_{Y_i}(t)\right]\left(1 - \left(1 - \frac{\Pi_{i=1}^{n}\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}}{\Pi_{i=1}^{n} E^{b_{ji}}\left[\hat{M}_{Y_i}(t)\right]}\right)\right)\right\} \\
&= \log\Pi_{i=1}^{n} E^{b_{ji}}\left[\hat{M}_{Y_i}(t)\right] + \log\left(1 - \left(1 - \frac{\Pi_{i=1}^{n}\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}}{\Pi_{i=1}^{n} E^{b_{ji}}\left[\hat{M}_{Y_i}(t)\right]}\right)\right) \\
&= \log\Pi_{i=1}^{n} E^{b_{ji}}\left[\hat{M}_{Y_i}(t)\right] + \log\left(1 - \omega_j\right) \\
&= K_{X_j}(t) - \omega_j - \frac{1}{2}\omega_j + H.O.T. \\
&\approx K_{X_j}(t) - \omega_j - \frac{1}{2}\omega_j
\end{aligned} \tag{11}$$

where $\omega_j = 1 - \dfrac{\Pi_{i=1}^{n}\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}}{\Pi_{i=1}^{n} E^{b_{ji}}\left[\hat{M}_{Y_i}(t)\right]}$. This suggests that we can correct the bias using the following equation:

$$\hat{K}_{X_j} = \sum_{i=1}^{n} b_{ji} \cdot \log\left(\hat{M}_{Y_i}(t)\right) + \hat{E}\left[\omega_j\right] + \frac{1}{2}\hat{E}\left[{\omega_j}^2\right] \tag{12}$$

where $\hat{E}(\cdot)$ denotes empirical average,

$$\hat{E}\left[\omega_j\right] = 1 - \frac{\prod_{i=1}^{n} \hat{E}\left[\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}\right]}{\hat{M}_{X_j}(t)} \tag{13}$$

$$\hat{E}\left[\omega_j^2\right] = 1 - \frac{2 \cdot \prod_{i=1}^{n} \hat{E}\left[\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}\right]}{\hat{M}_{X_j}(t)} + \frac{\prod_{i=1}^{n} \hat{E}\left[\left(\hat{M}_{Y_i}(t)\right)^{2b_{ji}}\right]}{\hat{M}_{X_j}^2(t)} \tag{14}$$

$\hat{M}_{X_j}(t)$ is an estimate of the loss moment generating function of link j, which can be obtained from

$$\hat{M}_{X_j}(t) = \prod_{i=1}^{n}\left(\hat{M}_{Y_i}(t)^{b_{ji}}\right). \tag{15}$$

The empirical average $\hat{E}\left[\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}\right]$ can be obtained by implementing a sliding window method with window size W and step size S [9]. Define the number $N_w = \left\lfloor \frac{N-W}{S} \right\rfloor$ of windows increments

$$\hat{E}\left[\left(\hat{M}_{Y_i}(t)\right)^{b_{ji}}\right] = \frac{1}{N_w}\sum_{l=1}^{N_w}\left(\frac{1}{W}\sum_{k=(l-1)S+1}^{(l-1)S+W} e^{tY_i^k}\right)^{b_{ji}}. \tag{16}$$

We obtain the empirical average $\hat{E}\left[\left(\hat{M}_{Y_i}(t)\right)^{2b_{ji}}\right]$ in a similar manner,

$$\hat{E}\left[\left(\hat{M}_{Y_i}(t)\right)^{2b_{ji}}\right] = \frac{1}{N_w}\sum_{l=1}^{N_w}\left(\frac{1}{W}\sum_{k=(l-1)S+1}^{(l-1)S+W} e^{tY_i^k}\right)^{2b_{ji}}. \tag{17}$$

2.1.3 Simulation study and application

The ns2 simulator was extended to perform the simulation of the sensor network.and simulate the data flow through sensor network. For each data collection round, whether a node successfully received data sent to it by its child nodes was determined randomly but with a specified intended loss rate for each link. That is, as the number of data collection rounds increases the actual loss rate of each link should converge to the intended loss rate. Two networks were used in the simulations. One consisted of 120 nodes while the other contained 9 nodes. Figure 3 shows the topology of the 9-node network. An intended success

rate of 0.9 was chosen for all normally links in the simulation network. Each simulation consisted of 1200 data collection trials. Once all of the data was collected, each link loss CGF was inferred using the approach presented in Section 4. To estimate the loss CGF, we set the window size W to be 400, and the window shift step size S to be 10.

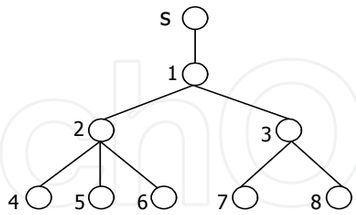


Fig. 3. A 9-node data aggregation tree

Two possible scenarios were simulated that may occur in a real sensor network. These scenarios were: (1) Equal losses throughout the network; (2) Cascaded losses. i.e., Heavy losses at links on the same path to the sink. The cascaded losses scenario was simulated by setting the intended success rates of links 2 and 5 to be 0.7.

Because each internal link loss CGF preserves all the statistical information of the link loss, we can accurately estimate many features of the link loss distribution from the link loss CGF. Here we give an example of lossy link detection. We define a lossy link in sensor network as the link whose loss rate exceeds a predefined threshold δ. In practical application, we can infer a link as the lossy link when the probability of a link loss rate exceeding δ exceeds a predefined threshold P. By the Chernoff Bound [38],

$$P\left(X_i \geq \delta\right) \leq e^{-t\delta} E\left[e^{tX_j}\right] = P_j$$

By appropriately selecting the threshold δ and threshold P close to 1, we can detect a lossy link by testing whether $P_j > P$. In Table 3, we show the Chernoff Bounds for $P(\boldsymbol{X}_j \geq 0.3)$ which were estimated from the simulation in Cascaded losses scenario. By setting the threshold P to 0.95, we can identify link 2 and 5 as the lossy link. This accord with the simulation configures.

Link	1	2	3	4	5	6	7	8
P(Xj ≥ 0.3)	0.66952	0.99671	0.67041	0.66942	0.99918	0.67060	0.66826	0.66684

Table 1. Chernoff Bound for each link in Cascaded losses scenario

2.2 Loss temporal dependency characteristic inference method

Here we concentrate on the problem of efficiently estimating the internal link loss rate from the passive end-to-end measurement. We use the Bayesian inference problem to formulate the sensor network loss inference problem and use the Metropolis-Hastings Sampling to find out link-level characteristics.

2.2.1 Loss inference based on gilbert model

In our proposed approach, firstly the unobservable data is inferred based on the link relationship and the observable data collected at the sink node. Once the unobservable data

has been identified, for every link this proposed approach uses Metropolis-Hastings algorithm to generate a sequence of samples of Gilbert model parameter. We iterate the unobservable data inference process and the sampling process until it reaches the given number of the samples. The following two subsections are used to detail the proposed algorithm for unobservable data and sampling algorithm, respectively. Without loss of generality, we also take the figure 1 for instance in this subsection.

2.2.2 Unobservable data inference

We employ the up-down approach to infer the unobservable data. Firstly, we infer the unobservable data of the node 1, the reception or loss of the packets sent from the children node of the node 1 to node 1, and then we move one level down to estimate the unobservable data of the lower level nodes that are children node of node 1. The process is continued until it reaches the leaf nodes.

Assume that one of children nodes of node i is node j. Using the similar method as presented above, we have the conditional posterior distribution of $y_{\mathrm{j,i}}^{m}$,

$$p(y_{j,i}^{m}=0 \mid \boldsymbol{X}, \boldsymbol{Y}_{j,i}^{[-m]}, \boldsymbol{\Theta}) = \begin{cases} 0 & \text{if } x_j^m = 1 \\ p(y_{j,i}^{m}=0 \mid y_{j,i}^{m-1}) \cdot p(y_{j,i}^{m+1} \mid y_{j,i}^{m}=0) & \\ = [y_{j,i}^{m-1} \cdot p_j + (1-y_{j,i}^{m-1})(1-q_j)] \cdot [y_{j,i}^{m+1} \cdot q_j + (1-y_{j,i}^{m+1})(1-p_j)] & \text{if } x_j^m = 0 \end{cases} \tag{18}$$

$$p(y_{j,i}^{m}=1 \mid \boldsymbol{X}, \boldsymbol{Y}_{j,i}^{[-m]}, \boldsymbol{Y} \setminus \boldsymbol{Y}_{j,i}, \boldsymbol{\Theta}) = \begin{cases} 1 & \text{if } x_j^m = 1 \\ p(x_j^m = 0 \mid \{y_{k,f(k)}^{m}, k \in a(i)\}, y_{j,i}^{m}=1) \cdot p(y_{j,i}^{m}=1 \mid y_{j,i}^{m-1}) \cdot p(y_{j,i}^{m+1} \mid y_{j,i}^{m}=1) & \\ = [(1 - \prod_{k \in a(i)} \frac{q_k}{p_k + q_k}] \cdot [y_{j,i}^{m-1} \cdot (1-p_j) + (1-y_{j,i}^{m-1}) q_j] \cdot [y_{j,i}^{m+1}(1-p_j) + (1-y_{j,i}^{m+1}) \cdot p_j] & \text{if } x_j^m = 0 \end{cases} \tag{19}$$

According to the conditional posterior distribution of $y_{j,i}^{m}$ as described above, we can draw a sequence of samples of $y_{j,i}^{m}$.

2.2.3 Loss performance parameter inference

We infer the Gilbert model parameters $\boldsymbol{\Theta}$ according to the samples of $y_{j,i}^{m}$ and the observable data $\boldsymbol{X}$. As the problem formulation describes, the estimated value $\hat{\boldsymbol{\Theta}}$ should agree with the posterior distribution $p(\boldsymbol{\Theta} \mid \boldsymbol{X}, \boldsymbol{Y})$. However, the posterior distribution $p(\boldsymbol{\Theta} \mid \boldsymbol{X}, \boldsymbol{Y})$ is not a closed-form expression. That is, the value of $\hat{\boldsymbol{\Theta}}$ can't be calculated from the data $\boldsymbol{X}$ and $\boldsymbol{Y}$ directly. In this paper, we consider the Metropolis-Hastings algorithm for sampling the parameters $\{(p_k, q_k), k \in \boldsymbol{V}\}$. Here we do not pay much attention on choosing the proposal distribution and the initial value of parameters, but concern that how to sample the parameters using Metropolis-Hastings algorithm. In [39], it is discussed in detail that choosing the proposal distribution and the initial value of parameters

We can choose a random walk proposal distribution for the proposed sampler, e.g.

$$g(p_k^{(j-1)}, p_k^{(j)}) \sim U(p_k^{(j-1)} - \sigma, p_k^{(j-1)} + \sigma) \tag{20}$$

That is, we draw a sample $p_k^{(j)}$ based on the above proposal distribution and accept it with probability

$$\alpha(p_k^{(j)}, p_k^{(j-1)}) = \min\left\{1, \frac{p(p_k^{(j)} \mid \boldsymbol{X}, \boldsymbol{Y}^{(j)}, \Theta \setminus p_k^{(j)})}{g(p_k^{(j)}, p_k^{(j-1)})} \cdot \frac{g(p_k^{(j-1)}, p_k^{(j)})}{p(p_k^{(j-1)} \mid \boldsymbol{X}, \boldsymbol{Y}^{(j-1)}, \Theta \setminus p_k^{(j-1)})}\right\} \quad (21)$$

where by assuming uniform prior on p_k, we have

$$p(p_k \mid \boldsymbol{X}, \boldsymbol{Y}^{(j)}, \boldsymbol{\Theta} \setminus p_k) \propto p(\boldsymbol{X}, \boldsymbol{Y}^{(j)} \mid \boldsymbol{\Theta}) \propto p(y_{k,f(k)}^0) \cdot p_k^{n_{1,0}} \cdot (1-p_k)^{n_{1,1}} \quad (22)$$

where n_{uv} is the number of occurrences of the adjacent pair (u, v) in the sequence $\boldsymbol{Z}_{k,f(k)}$, $u, v \in \{0,1\}$. As the loss model describes, each node tries to send data in each round. Thus the marginal distribution on $y_{k,f(k)}^0$ can be given by $p(y_{k,f(k)}^0 = 0) = \frac{p_k}{p_k + q_k}$ and $p(y_{k,f(k)}^0 = 1) = \frac{q_k}{p_k + q_k}$.

Using the formula (21)(22)(23), we can draw the random samples of p_k based on the samples of the unobservable data $\boldsymbol{Y}$ and observable data $\boldsymbol{X}$. Similarly, we can also draw the random samples of q_k where

$$p(q_k \mid \boldsymbol{X}, \boldsymbol{Y}^{(j)}, \boldsymbol{\Theta} \setminus q_k) \propto p(\boldsymbol{X}, \boldsymbol{Y}^{(j)} \mid \boldsymbol{\Theta}) \propto p(y_{k,f(k)}^0) \cdot q_k^{n_{0,1}} \cdot (1-q_k)^{n_{0,0}} \quad (23)$$

The proposed sampler iterates between sampling $y_{j,i}^m$ from the observable data $\boldsymbol{X}$ and sampling the Gilbert model parameters (p_k, q_k) based on the above sampler. After the sample procedure is finished, we can calculate the estimated value of $\hat{\Theta} = \{(p_k, q_k), k \in \boldsymbol{V}\}$. For a general sensor network, we can similarly infer link loss rate as in this simple example described above, and expand the sampling strategy as an up-bottom approach where we start from the child node of the sink node, followed by their child nodes, and so on, until we reach the leaf nodes.

2.2.4 Algorithm description

Suppose the total number of samples is $J=J_0+J_1$, where J_0 is the number of samples as 'burn-in' period and J_1 is the number of samples used to infer link loss parameters. Denote $\boldsymbol{\Theta}^{(i)}$ and $\boldsymbol{Y}^{(i)}$ as the ith round sample value.

Initialization: Draw random samples $\boldsymbol{\Theta}^{(0)}$ and $\boldsymbol{Y}^{(0)}$ from their perior.

Sample: for $j = 1, 2, \cdots J$ do

- Given $\boldsymbol{\Theta}^{(j-1)}$, for each $k \in V \setminus \{\{s\} \cup d(s)\}$, and $m = 1, 2, \cdots, N$, draw a sample $(y_{k,f(k)}^m)^{(j)} \sim p(y_{k,f(k)}^m \mid x_{k,f(k)}^m, \{(y_{k,f(k)}^i)^{(j-1)}, i = m-1, m+1\}, \{(p_n^{(j-1)}, q_n^{(j-1)}), n \in \{k\} \cup a(k)\})$.
- Given $\boldsymbol{Y}^{(j)}$, for each $k \in V \setminus \{s\}$, draw a random sample of $p_k^{(j)}$ based on $p_k^{(j-1)}$

$$g(p_k^{(j-1)}, p_k^{(j)}) \sim U(p_k^{(j-1)} - \sigma, p_k^{(j-1)} + \sigma)$$

and accept it with probability $\alpha(p_k^{(j)}, p_k^{(j-1)})$.

Inference : Calculate $\hat{\boldsymbol{\Theta}}$ from $\left\{\boldsymbol{\Theta}^{(J_1)}, \boldsymbol{\Theta}^{(J_1+1)}, \cdots, \boldsymbol{\Theta}^{(J)}\right\}$

Output : $\hat{\boldsymbol{\Theta}}$

Denote the size of sensor network as $|\boldsymbol{V}|$. From the algorithm described as above, we can get the time complexity of this proposed algorithm is $O(J \times N \times |\boldsymbol{V}|)$.

2.2.5 Simulation study

NS2 was used to perform the simulation of the sensor network. The ns2 was extended to simulate the data flow through sensor network. For each data collection round, whether a node successfully received data sent to it by its child nodes was determined randomly but with a specified intended loss performance for each link. The inference algorithm is implemented in MATLAB.

Two networks were used in the simulations. One consisted of 120 nodes while the other contained 9 nodes. Figure 3 shows the topology of the 9-node network. We used the Gilbert error model to model the link loss performance with parameters (p, q) as (0.1, 0.85) for all normally links in the simulation network. Each simulation consisted of 1000 data collection trials.

In the 9-node simulation network, we simulated two possible scenarios that may occur in a real sensor network. These scenarios were: 1) Equal losses throughout the network; 2) Heavy losses at some links. The second scenario was simulated by setting the loss parameters of links 2, 5 and 7 to be (0.15, 0.80).

Four plots of the inferred and sampled internal link loss performance parameters for all links are shown in Fig.4-Fig.7, respectively. The inferred link loss performance value is very close to the sampled link loss performance value. In the second scenario the error was significant since some of the losses that should have been attributed to link 2 were instead attributed evenly amongst link 2's child links. However, it is still possible to infer that these lossy links is in fact experiencing the heavy losses.

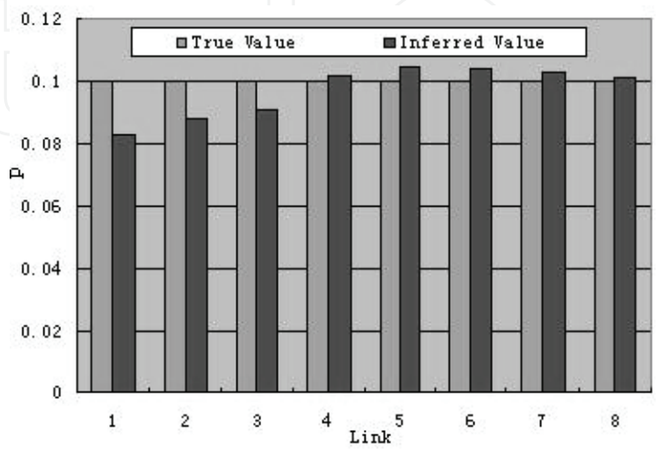


Fig. 4. True Value vs. Inferred Value in the equal loss scenarios for p

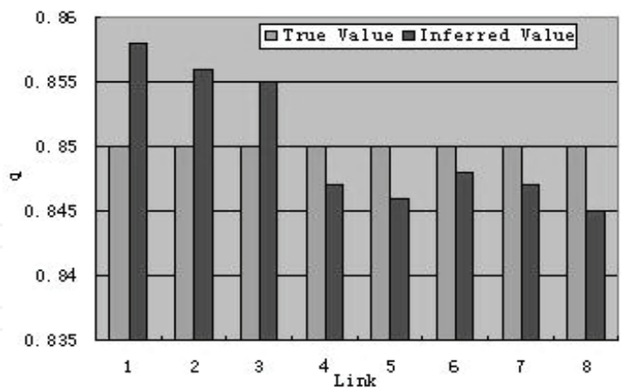


Fig. 5. True Value vs. Inferred Value in the equal loss scenarios for q

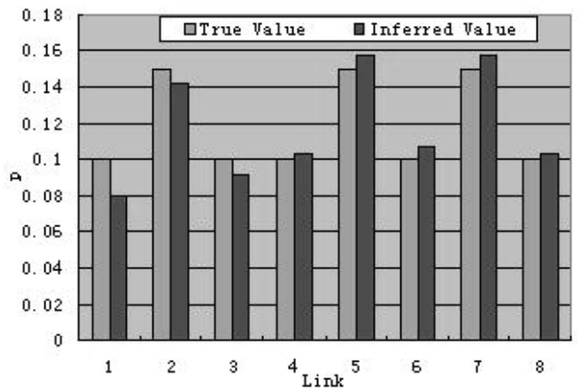


Fig. 6. True Value vs. Inferred Value in the heavy loss scenarios for p

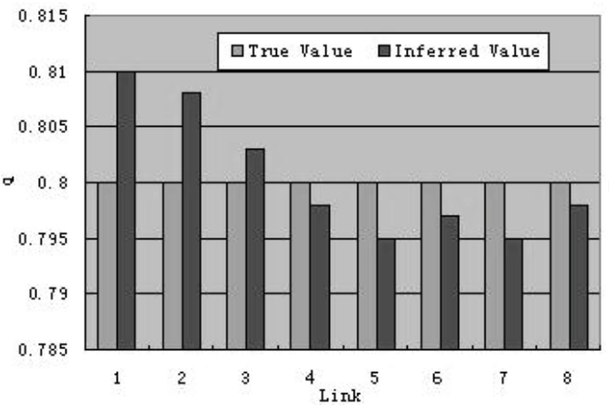


Fig. 7. True Value vs. Inferred Value in the heavy loss scenarios for q

Take the link 2 for instance. Figure 8 shows the relationship between the convergence speeds of the estimated loss performance value and the number of samples. Before the burn-in period was over, the error between the estimated value and the true value is significant. With the sample number increases, the estimated value is approaching to the true value.

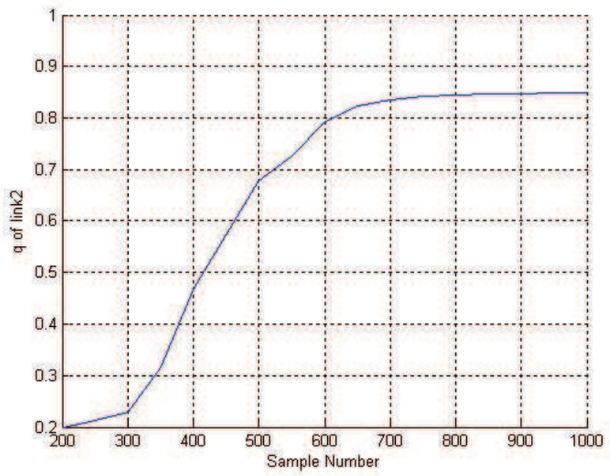


Fig. 8. Inferred Value vs. Sample Number in equal loss scenarios for q of link 2

Table1 provides the simulation result in 120-node network. It shows that the inferred link loss performance value is close to its true value. In the two simulation scenarios, the maximum error of link loss estimation is only 0.027 and 0.0312, respectively. These results show that our loss rate inference algorithm scales well.

Equal losses	Mean Error	p	0.043
		q	0.021
	Max Error	p	0.070
		q	0.052
Heavy losses on some links	Mean Error	p	0.058
		q	0.027
	Max Error	p	0.089
		q	0.061

Table 2. Absolute errors: 120-node network

3. NT measurement technique in ad hoc network

NT measurement technique adopts Edge nodes not only as the source sender to send the measurement packets, but also as the receivers to receive the measurement data sample used for inferring link performance parameters in Ad Hoc network. Since it is independent of network infrastructure and protocols, NT measurement outweighs internal network measurement in Ad Hoc network. Of course, there will appear new problems for introducing the NT technique to Ad Hoc network measurement.

The dynamic characteristic of Ad Hoc network topology is the main obstacle to use NT technique in Ad Hoc network measurement, because it effects the correctness not only of the measurement results, but also of the link performance parameters inference results. Therefore, the following problems must be resolved at first: (1) to put forward a feasible analysis method on dynamic characteristic of Ad Hoc network so as to meet the requirement of NT technique. (2) to found the Ad Hoc network measurement topology architecture and link performance inference model. (3) to chose the proper measurement method so as to

obtain measurement sample of performance parameters based on End-to-End. (4) to bring forth a link performance inference method so as to infer the link performance parameters by using measurement data sample, link performance inference model , mathematical and statistical theory.

3.1 Ad hoc network topology dynamic characteristics

Although researches have focused on the dynamic characteristics of mobility models in Ad Hoc network and taken much achievements recently[40], little attention was paid on the link topology dynamic characteristics of mobile models. Narayannan Sadagopan et al. [41]puts forward a statistical method to obtain the dynamic characteristic of MM, which includes how to obtain the probability density distribution of link and path connection time. Nevertheless, the research mainly focus on the viewpoint of the influence of dynamic characteristic on the performance of active network protocols, not on that of the Ad Hoc network measurement. At the same time, statistical analysis method is only applicable for the certain mobility models with one time to change its' velocity or direction in one second, such as RPGM, Freeway and Manhattan mobility model, not for the other mobility models in NS-2 tool, such as RW and RWP. Although Tian et al. [42]brings forward a link connection time model which could be used to compute the link connection minimum time, and further to obtain the minimum value of network topology lifetime. However, the computing model is too complicated for not being simplified. Besides, it is only adaptable for the RWP mobility model, not for the other mobility models in Ad Hoc networks. Wang et al. [43] brings forth a circle mobility model, in which when the initialization position of mobile nodes is known, the network topology architecture of Ad Hoc network could be computed according to the rules of nodes' movement. Specially, the minimum of network topology lifetime could also be obtained statistically. However, this research on NT measurement technique in Ad Hoc network mainly focus on circle mobility model, it fails to be useful for other mobility models. Therefore, How to put forward a analysis technique on the dynamic characteristic of Ad Hoc network topology , which could be used for all the mobility model as are supported in NS-2 tool, is an interesting issue to be solved.

In order to resolve the above problem, Yao et al.[44] presents a network topology snapshots capture method to obtain the Ad Hoc network topology architecture at any moment on the basis of analysis on the scene files of mobility models in Ad Hoc network. Through analyzing on the Ad Hoc network topology snapshots, the times of network topology in steady state or unsteady state during a certain time t could be obtained statistically, as well as the durative time of network topology in steady state or unsteady state during the whole simulation time. Furthermore, Yao et al.[45] adopts the discrete time and continuous time Markov stochastic process theory to predict the probability of the network topology invariability event happening and that of the network topology variability event happening, and the experiential formula of the probability of the network topology invariability and variability was deduced. The simulation result shows that the statistical analysis technique on Ad Hoc network topology dynamic characteristic not only is effective, but also has the general attribute, which could be used in the statistical analysis technique on Ad Hoc network topology dynamic characteristic under any mobility model.

3.1.1 Formalized description on mobility model

All the mobility models supported by NS-2 [46]have the same format of scene files produced by setdest tool. Through analysis on the scene files we could arrive at the conclusion that

there is a certain spatial relativity among mobile nodes. That is, the destination position of node j at time i is its current position at time $i + 1$ on condition that V_i^j equals zero, where V_i^j denotes the velocity of node j from time i to $i + 1$. Furthermore, during the period from time i to $i + 1$, node j moves along a line at the velocity of V_i^j from $C_i^j = (c_{x,i}^j, c_{y,i}^j)$ to $D_i^j = (d_{x,i}^j, d_{y,i}^j)$, where C_i^j denotes current position of node j at time i, $c_{x,i}^j$ and $c_{y,i}^j$ the x position and y position respectively of node j at time i, D_i^j the destination position of node j at time i. Then the spatial relativity of mobile nodes could be expressed as formula (24).

$$\begin{cases} C_{i+1}^j = D_i^j, & \text{if } C_i^j \neq D_i^j \ \& \ v_i^j != 0 \\ C_{i+1}^j = C_i^j, & \text{if } v_i^j = 0 \end{cases} \tag{24}$$

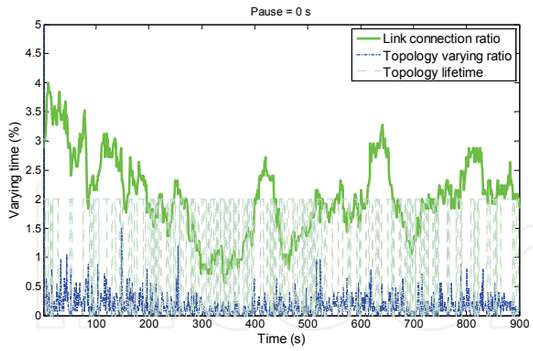
If let γ denote the snapshot time slot, the relativity between velocity and spatial position could be expressed as formula (25).

$$\begin{cases} c_{x,i+1}^j = c_{x,i}^j + v_{x,i}^j \times \gamma \\ c_{y,i+1}^j = c_{y,i}^j + v_{y,i}^j \times \gamma \end{cases} \tag{25}$$

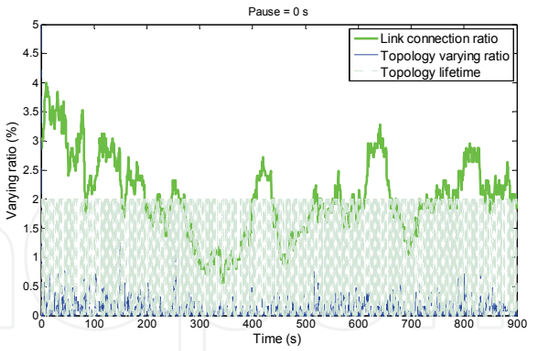
where $v_{i,x}^j$ and $v_{i,y}^j$ denote the x-axis and y-axis value of speed v_i^j at time i, which could be obtained by using position C_i^j, D_i^j and v_i^j. Thus it can be seen, the state information of node j at time i could be expressed as a three tuple $\langle C_i^j, D_i^j, V_i^j \rangle$. Furthermore, position snapshots of mobile nodes at any moment could be derived from formula (25). The method how to get physical topology snapshot is to compute the Euclid distance R between node j and $l(l \in V \setminus \{j\})$ at each time, where V denotes the node set of Ad Hoc network. If R is smaller than the transmission range of mobile node denoted as r, illuminating that there is a chance for the node j and l to build up a wireless connection at link layer, the state of link between node j and l could be set as 1, otherwise, as 0. If the same operation is implemented between any mobile nodes at each snapshot time, we could achieve the physical topology snapshot. At last, the steady and un-steady period of Ad Hoc network topology can be obtained by computing all the physically topology snapshots statistically.

3.1.2 Simulation study

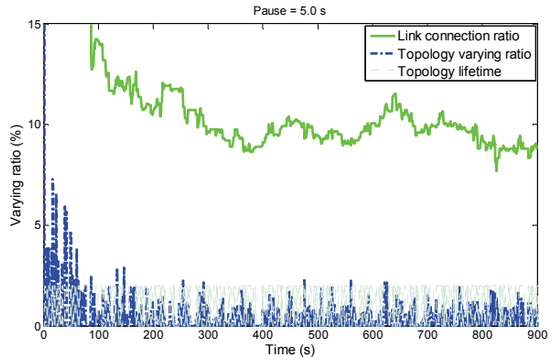
Through analyzing on the Ad Hoc network topology snapshots with RW and RWP mobility model, the relation of the link topology in steady or un-steady state and link topology varying ratio varying with time are shown as in Fig. 9(a~d). Next, we will explain the three concepts used in Fig. 9. Link connection ratio is the ratio of the links having a wireless connection with each other to all links in Ad Hoc networks in each one topology snapshot. Topology varying ratio is the ratio of the number of links that the state of which has varied



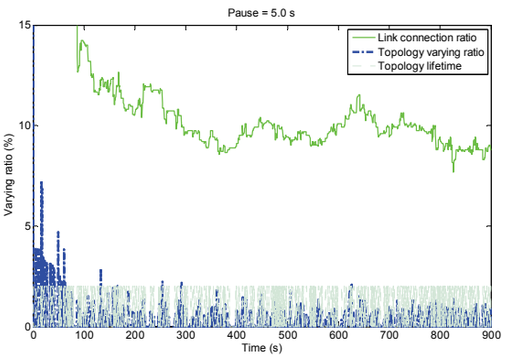
(a) snapshot time = 1.0s in RW



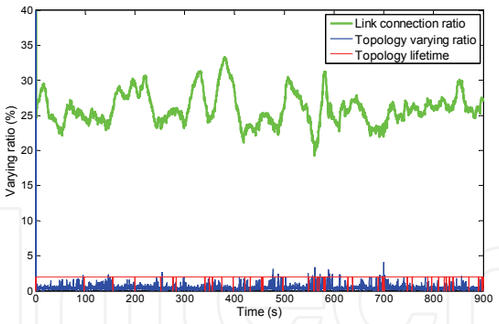
(b) snapshot time = 0.5s in RW



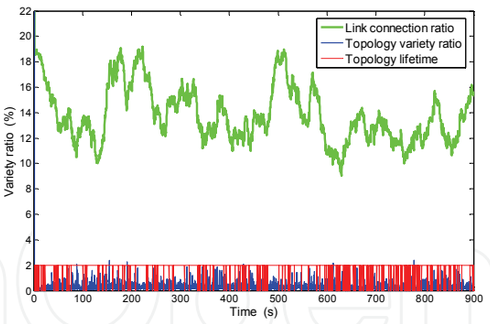
(c) snapshot time =1.0s in RWP



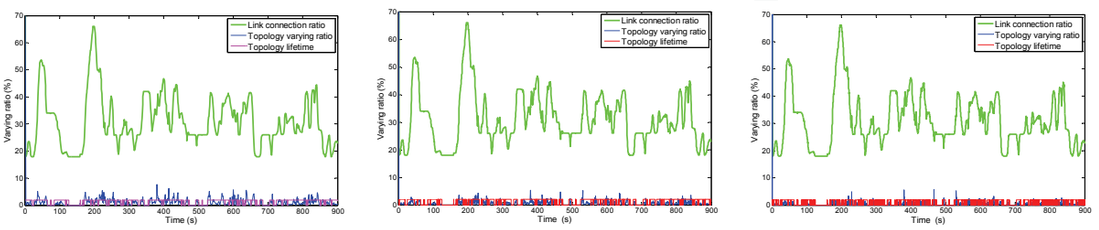
(d) snapshot time = 0.5s in RWP



(e) snapshot time = 0.25 in Freeway



(f) snapshot time = 0.25 in Manhattan



(g) snapshot time =1.0s in RPGM (h) snapshot time =0.5s in RPGM (i) snapshot time =0.25s in RPGM

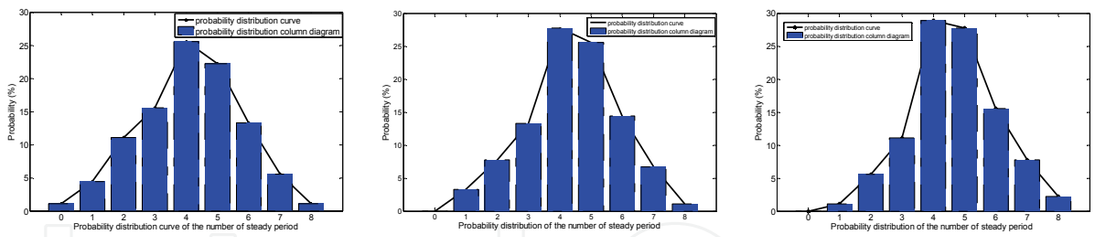
Fig. 9. Topology dynamic characteristic

to all links between the two consecutive topology snapshots. Topology lifetime is the time during which the Ad Hoc network topology does not vary. Actually the curve of topology lifetime is equivalent to that of the topology varying ratio in Fig.9, since when the value of topology varying ratio between the two consecutive topology snapshots is not equal to zero, the topology lifetime is set as two, otherwise set as zero to denote that the Ad Hoc network topology does not vary between the two consecutive topology snapshots. The mobile scene is set as the following parameters in NS-2: There are all 50 mobile nodes, and the stop time is 0s in RW and 5s in RWP respectively. The maximum velocity of mobile nodes is 20m/s, simulation being 900s, and the scene covers a square area with 1200m*1200m. The wireless communication coverage range is set as a circle with radius being 250m.

According to the result of analysis on the RW, RWP mobility model as in Fig. 9(a~d)[47], and that on the Freeway, Manhattan and RPGM mobility model in Fig.9(e~i)[48], we could safely arrive at the conclusion: The steady and un-steady period appear in turn during all simulation time, and the number of the steady and un-steady state, and the duration time in each state vary with different mobility models and the parameters of movement scenes.

3.1.3 Statistical characteristic of the steady period number

In a certain time t, the number of steady period (or un-steady period) is a discrete stochastic variable X. Through analyzing on the stochastic variable X, we could obtain the frequency of the steady period (or un-steady period) appearing in a certain time. We used the data in Fig. 9(d) as an example to obtain the probability distribution chart of the number of steady period appearing in $t = 10$ s, $t = 15$ s and $t = 20$ s as in the Fig. (a), (b) and (c) respectively.



(a) $t = 10$ s (b) $t = 15$ s (c) $t = 20$ s

Fig. 10. Probability distribution chart of the number of steady period

From the Fig. 10, we could likely arrive at the inconclusive hypothesis that the number of steady period appearing in a certain time approximately follows the poison distribution, and for different time there exists different parameter λ. Next, we will use χ^2 Fit hypothesis testing method to verify this hypothesis. At first, we put forward the following hypothesis test problem:

H_0 : The number of steady period follows the poison distribution,

H_1 : The number of steady period does not follow the poison distribution.

If the statistical time is set as $t = 10$ s, that is, we will count the number of steady period once per 10 seconds. Through processing the data in Fig. 10(a), about 90 statistical data is obtained as in the table 3.

i	0	1	2	3	4	5	6	7	8
$n\widehat{p_i}$	1.43	5.93	12.28	16.95	17.54	14.52	10.02	5.93	3.07
m_i	1	4	10	14	23	20	12	5	1
$\lvert m_i - n\widehat{p_i}\rvert$	0.44	1.93	2.28	2.95	5.46	5.48	1.98	0.93	2.07
$\frac{(m_i - n\widehat{p_i})^2}{n\widehat{p_i}}$	0.13	0.63	0.42	0.51	1.70	2.07	0.39	0.15	1.40

Table 3. χ^2 fit hypothesis testing table about the number of steady period

And then, we discuss how to verify the hypothesis test problem in the following three steps.

Step 1. To compute parameter λ in poison distribution by using the maximum likelihood estimate under the condition that hypothesis H_0 is true.

If the sample of stochastic variable X is denoted as x_i , $i = 0,1,...,n(n = 89)$, the maximum likelihood function about parameter λ could be expressed as formula (26).

$$L(\lambda)=\prod_{i=1}^{n}(\frac{\lambda^{x_i}}{x_i!}e^{-\lambda})=e^{-n\lambda}\frac{\lambda^{\sum_{i=1}^{n}x_i}}{\prod_{i=1}^{n}(x_i!)} \tag{26}$$

Though implementing the logarithmic operation on both sides of the formula (26), the logarithmic maximum likelihood function could be expressed as formula (27).

$$\ln L(\lambda) = -n\lambda + \ln\lambda\sum_{i=1}^{n}x_i - \sum_{i=1}^{n}\ln(x_i!) \tag{27}$$

In order to let the formula (27) equal to its maximum, we implement the differential coefficient operation for parameter λ on both sides of formula (27), and let it equal to zero as the formula (28).

$$\frac{d\ln L(\lambda)}{d\lambda} = -n + \frac{1}{\lambda}\sum_{i=1}^{n}x_i = 0 \tag{28}$$

Through computing the formula (28), the maximum likelihood estimate of parameter λ in poison distribution could be expressed as the following:

$$\hat{\lambda} = \frac{1}{n}\sum_{i=1}^{n}x_i .$$

Noted that the maximum likelihood estimator of parameter λ has the attributes, such as, an un-bias and effective estimate. According to the data in table 1, we could easily obtain the estimate value of parameter λ as $\hat{\lambda}$:

$$\hat{\lambda} = \frac{1}{n}\sum_{i=1}^{n}iv_i = 4.14 .$$

Step 2. To compute the test statistic variable V as is expressed in formula (29).
According to analysis on the data in table 1, we could obtain the value of the test statistic variable: $v = 7.40$.

$$V = \sum_{i=1}^{n} \frac{(m_i - np_i)^2}{np_i} \tag{29}$$

Step 3. Under the condition that significance level α equal to 0.05, we could get the in-equation relation between the theoretical value and statistical one as the following:

$$\chi_{\alpha}^{2}(r-1) = \chi_{0.05}^{2}(9-1) = 15.507 > 7.40$$

This in-equation relation means that the test statistic variable v does not belong to the reject range, therefore, we have to accept the hypothesis H_0, and to refuse another hypothesis H_1. It is reasonable for us to believe that the number of steady period appearing in 10 seconds follows the poison distribution with $\lambda = 4.21$, when we choose RWP mobility model in a certain mobile scene as our research object.

At the same time, that the number of un-steady period appearing in 10 seconds follows the poison distribution with $\lambda = 4.21$ could also be verified as the method above. When the statistical time is equal to different values, such as 15s, 20s, and so on, or when we choose other different mobility models, such as RW, Freeway, Manhattan and RPGM, we could also safely arrive at the conclusion that the number of steady or un-steady period appearing in a certain time also follows the poison distribution with different parameter λ. The paper does not discuss these for the limit to its length.

3.1.4 Statistical characteristic of the steady or un-steady duration time

When Ad Hoc network topology is in the steady state, the duration of which is called as steady duration time, otherwise, called as un-steady duration time. Because the steady duration time is a continuous stochastic variable, the statistical analysis method on the data about steady duration time in Fig. 9(d) is different from that on the number of steady period appearing in a certain time. Therefore, we divide the analysis method into three steps as the followings.

Step 1. To coordinate the data.
At first, we should coordinate the data about steady duration time, such as $x_1, x_2, \ldots, x_n$, in the sort ascending order as $x_{(1)} \le x_{(2)} \le \ldots \le x_{(n)}$, where n is the scale size of data sample about steady duration time, $x_{(1)}$ is the minimal value of the steady duration time, and $x_{(n)}$ the maximal one.

Step 2. To discrete the zone $[x_{(1)}, x_{(n)}]$.

Secondly, the zone $[x_{(1)}, x_{(n)}]$ is discrete to l smaller zones or groups as $I_i (1 \le i \le l)$ according to the scale size of data sample about steady duration time n. In general, if $n \ge 100$, the value of l belongs to the zone $[10, 20]$; when n is equal to 50 or so, l usually is set as 5 or 6. Since in Fig.1(d), $n = 332 \ge 100$ comes into existence, we set the value of l as 10. The case of small zones about the data in Fig.1(d) is processed and analyzed as in table 4.

Zone number: i	1	2	3	4	5	6	7	8	9	10
Zones	(0.0, 1.0]	(1.0, 2.0]	(2.0, 3.0]	(3.0, 4.0]	(4.0, 5.0]	(5.0, 6.0]	(6.0, 7.0]	(7.0, 8.0]	(8.0, 9.0]	(9.0, ∞]
$n\widehat{p_i}$	157.0	87.6	48.8	27.2	15.2	8.5	4.7	2.6	1.5	1.9
m_i	172	77	35	18	13	6	5	2	2	2
$\lvert m_i - n\widehat{p_i}\rvert$	15.0	10.6	13.8	9.2	2.2	2.5	0.3	0.6	0.5	0.1
$\frac{(m_i - n\widehat{p_i})^2}{n\widehat{p_i}}$	1.43	1.28	3.90	3.11	0.32	0.74	0.02	0.14	0.17	0.01

Table 4. χ^2 fit hypothesis testing table about steady duration time

Step 3. To analyze on the steady duration time

According to the data in the anterior three lines in table 4, the probability distribution of steady and un-steady duration time in Fig. 9(d) is shown as the Fig. 11 and Fig. 12 respectively. If we connect the middle points in the upper side line of the each rectangle to construct a fold line, when n and l are big enough, the fold line is approximate to the PDF curve of the stochastic variable, the steady or un-steady duration time, according to the probability statistic theory as in Fig. 11 and 12.

The larger is the scale size of data sample, the steady duration time, the smaller is the each zone, and PDF curve of the steady duration time of Ad Hoc network topology is more precise. According to the curve in Fig. 11, we could also likely arrive at the inconclusive hypothesis that the steady duration time approximately follows the exponential distribution. Next, we will use χ^2 fit hypothesis testing method to verify this hypothesis. At first, we put forward the following hypothesis test problem.

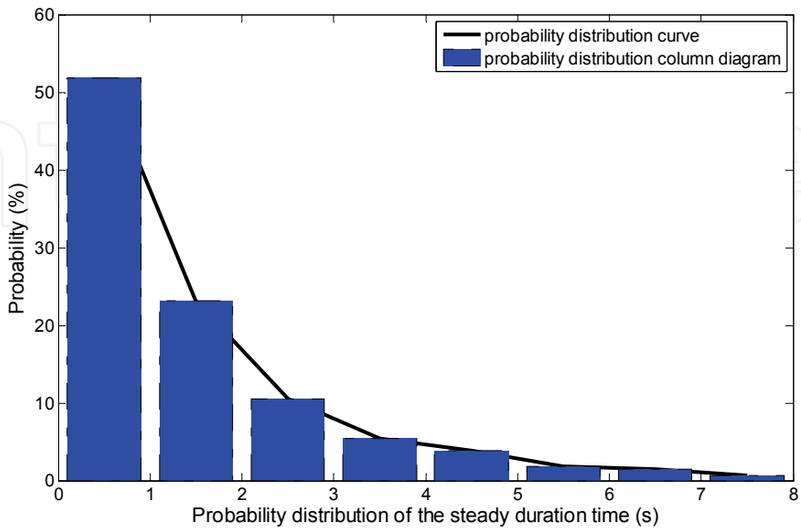


Fig. 11. PDF of steady duration time

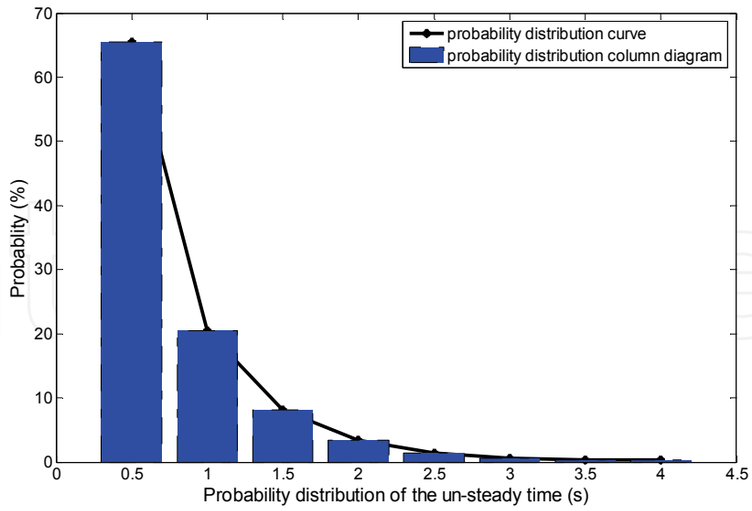


Fig. 12. PDF of un-steady duration time

H_0 : The steady duration time follows the exponential distribution,

H_1 : The steady duration time does not follow the exponential distribution.

According to analysis on the data in Fig. 9(d), we could obtain 332 data sample about the steady duration time. The analysis result about the 332 data sample is shown as in the table 2. Next, we will discuss the hypothesis test problem in the following three steps as the similar to that in section 3.1.

Step 1. To compute parameter λ in exponential distribution by using the maximum likelihood estimate under the condition that hypothesis H_0 is true.

If the sample of stochastic variable X , the steady duration time, is denoted as x_i , $i = 0,1,...,n(n = 331)$, the maximum likelihood function about parameter λ could be expressed as formula (30).

$$L(\lambda) = \lambda^n \exp(-\lambda \sum_{i=1}^{n} x_i) \tag{30}$$

Though implementing the logarithmic operation on both sides of formula (30), the logarithmic maximum likelihood function could be expressed as formula (31).

$$\ln L(\lambda) = n \ln \lambda - \lambda \sum_{i=1}^{n} x_i \tag{31}$$

In order to let the formula (31) equal to its maximum, we implement the differential coefficient operation for parameter λ on both sides of formula (31), and let it equal to zero as the formula (32).

$$\frac{d \ln L(\lambda)}{d(\lambda)} = \frac{n}{\lambda} - \sum_{i=1}^{n} x_i = 0 \tag{32}$$

Through computing the formula (32), the maximum likelihood estimator of parameter λ in exponential distribution could be expressed as formula (33).

$$\hat{\lambda} = \frac{1}{\frac{n}{\sum_{i=1} x_i}} = 1 \Big/ X_n \tag{33}$$

Noted that the maximum likelihood estimate of parameter λ also has two attributes, such as, an un-bias and effective estimate. To the limit of this paper, we ignore its proof. According to analysis on data in table 4, we could easily obtain the estimate value of parameter λ as $\hat{\lambda}$:

$$\hat{\lambda} = \frac{1}{\frac{1}{n}\sum_{i=1}^{n} iv_i} = 1 \Big/ 1.713$$

Step 2. To compute the test statistic variable: $V = \sum_{i=1}^{n} \frac{(m_i - np_i)^2}{np_i}$. According to analysis on the data in table 4, the value of the test statistic variable could be obtained as $v = 11.12$.

Step 3. Under the condition that significance level α is equal to 0.05, there exists the inequation relation between the theoretical value and the statistical one as $\chi_\alpha^2(r-1) = \chi_{0.05}^2(9-1) = 15.507 > v$.

This in-equation relation means that the test statistic variable v does not belong to the reject range, therefore, we have to refuse the hypothesis $H1$, and accept another hypothesis $H0$. It is reasonable for us to believe that the steady duration time follows the exponential distribution with the $\hat{\lambda} = 1 \Big/ 1.713$ =0.584, when we choose RWP mobility model in a certain mobile scene as our research object. At the same time, we could also prove that the un-steady duration time in the whole simulation time follows the exponential distribution with $\hat{\lambda} = 1.276$. In the same way, when the statistical time is equal to different values, such as 15s, 20s, and so on, or when we choose other different mobility models, such as RW, Freeway, Manhattan and RPGM, we could also safely arrive at the conclusion that the steady or un-steady duration time follows the exponential distribution with different parameter λ. The paper does not discuss these for the limit to its length.

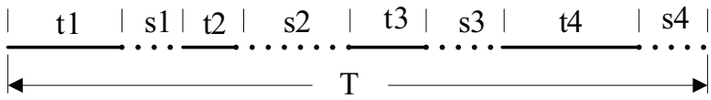
3.1.5 Markov stochastic process analysis method

According to the analysis result above, the dynamic characteristic of Ad Hoc network topology mainly embodies the following two points: one is that there is two states about Ad Hoc network topology, that is, the steady state and the un-steady state. Specially, the number of steady state or un-steady state appearing in a certain time follows the poison distribution with parameter λ. Another is that the steady or un-steady duration time follows the exponential distribution with parameter λ'. Therefore, we could easily arrive at the theorem 1.

Theorem 1 The dynamic varying process of Ad Hoc network topology is actually a continuous time and discrete state Markov stochastic one.

Proof:

When the data of Ad Hoc network topology snapshots with the snapshot time set as 0.25s is compared with that snapshot time set as 0.125s about RWP, RW, RPGM, Freeway and Manhattan mobility models, we find that the absolute error between them is less than 1%. Therefore, it is reasonable for us to consider that when the snapshot time is small enough, the states of the two consecutive Ad Hoc network topology snapshots does not vary except of the skip varying of state. This shows that time of MM is comprised of a serial of the steady and un-steady duration time periodically as in Fig. 13, where t1~t4 represent each different steady duration time, and s1~s4 the different un-steady ones , which could be achieved by counting the states of all the Ad Hoc network topology snapshots with small snapshot time.



T: time, t1~t4: the steady duration time, s1~s4: the un-steady duration time

Fig. 13. Time sequence

If the state space is set as $I = \{i_n, n \geq 0\}(i_n \in \{0,1\})$, where "0" denotes the steady state and "1" the un-steady state, for any time $0 \leq t_1 < t_2 < ... < t_n < t_{n+1}$ and the its corresponding states $i_1, i_2, ..., i_n, i_{n+1} \in I$, there exists the following formula:

$$P\{X(t_{n+1}) = i_{n+1} \mid X(t_1) = i_1, X(t_2) = i_2, ..., X(t_n) = i_n\}$$

$$= P\{X(t_{n+1}) = i_{n+1} \mid X(t_n) = i_n\} \tag{34}$$

According to formula (34), the state of Ad Hoc network topology snapshot is not only correlative merely to that of its former one, but also a discrete stochastic variable. Further more, the duration time of each state, that is, the steady or un-steady duration time is a continuous ones. Therefore, the theorem 1 is proved.

According to the analysis results in above section, if the Ad Hoc network topology is in steady state (denoted as "0")now, after a steady duration time in this state, it transfers to the un-steady state (denoted as "1"), and the un-steady duration time keeps to the exponential distribution with parameter λ_1. However, the steady duration time follows the same distribution with parameter λ_2. Therefore, the density matrix of this Markov stochastic process could be denoted as the following Q.

$$\begin{pmatrix} -\lambda_1 & \lambda_1 \\ \lambda_2 & -\lambda_2 \end{pmatrix}$$

According to the forward differential equation of continuous time Markov stochastic process[19,20], $P'(t) = P(t)Q$, the following differential equations (35) could be obtained.

$$\begin{cases} p'_{00}(t) = -\lambda_1 p_{00}(t) + \lambda_2 p_{01}(t) \\ p'_{01}(t) = \lambda_1 p_{00}(t) - \lambda_2 p_{01}(t) \\ p'_{10}(t) = -\lambda_1 p_{10}(t) + \lambda_2 p_{11}(t) \\ p'_{11}(t) = \lambda_1 p_{10}(t) - \lambda_2 p_{11}(t) \end{cases} \tag{35}$$

According to the probability theory, there exists the following restriction condition.

$$\begin{cases} p_{00}(t) = 1 - p_{01}(t) \\ p_{11}(t) = 1 - p_{10}(t) \end{cases}$$

If we use the equation $p_{01}(t) = 1 - p_{00}(t)$ to replace the $p_{01}(t)$ in the first differential equation of formula (35), then the following equation could be obtained.

$$p'_{00}(t) = \lambda_2 - (\lambda_1 + \lambda_2) p_{00}(t)$$

Let $Q_{00}(t)$ be equal to $e^{(\lambda 1+\lambda 2)t} p_{00}(t)$, that is, $Q_{00}(t) = e^{(\lambda 1+\lambda 2)t} p_{00}(t)$,Then to implement the differential coefficient operation on both sides of this equation for the parameter t , we could get the formula (36).

$$Q'_{00}(t) = (\lambda_1 + \lambda_2) e^{(\lambda 1+\lambda 2)t} p_{00}(t) + e^{(\lambda 1+\lambda 2)t} p'_{00}(t) \tag{36}$$

To multiply the first equation of the formula (35) by $e^{(\lambda 1+\lambda 2)t}$ on its both sides, the formula (36) could be simplified as the following formula (37).

$$Q'_{00}(t) = \lambda_2 e^{(\lambda 1+\lambda 2)t} \tag{37}$$

Through implementing the integral operation on the both sides of the formula (37) and adopting the initial condition: $p_{00}(0) = 1$, we could finally obtain the following forecast experimental formula (38) and (39).

$$p_{00}(t) = \frac{\lambda_2}{\lambda 1+\lambda 2}(1 + e^{-(\lambda 1+\lambda 2)t}) \tag{38}$$

$$p_{11}(t) = \frac{1}{\lambda 1+\lambda 2}(\lambda_1 + \lambda_2 e^{-(\lambda 1+\lambda 2)t}) \tag{39}$$

Formula (38) means that if the Ad Hoc network topology is in the steady state now, after time t, the probability that it is still in steady state is $p_{00}(t)$. Formula (39) means that if the Ad Hoc network topology is in the un-steady state now, after time t, the probability that it is still in un-steady state is $p_{11}(t)$. Therefore, formula (38) and (39) are called as the Ad Hoc network topology steady and un-steady duration time forecast experimental formula

respectively. Next, we use the concept of opposite events in probability theory to obtain warning experimental formal (40) and (41).

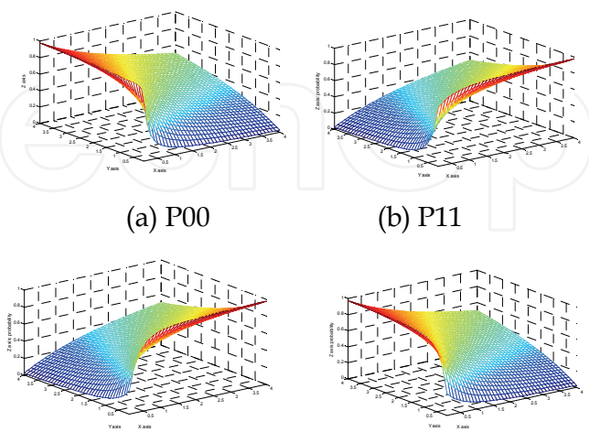
$$p_{01}(t)=1-p_{00}(t)=\frac{\lambda_1}{\lambda_1+\lambda_2}(1-e^{-(\lambda_1+\lambda_2)t}) \tag{40}$$

$$p_{10}(t)=1-p_{11}(t)=\frac{\lambda_2}{\lambda_1+\lambda_2}(1-e^{-(\lambda_1+\lambda_2)t}) \tag{41}$$

Formula (40) means that if the Ad Hoc network topology is in the steady state now, after time t, the probability that its state varies as un-steady one is $p_{01}(t)$. While formula (41) means that if the Ad Hoc network topology is in the un-steady state now, after time t, the probability that its state varies as steady one is $p_{10}(t)$. When time is set as 4s and 10s respectively, the experimental probability about state keeping invariable and varying is shown as the Fig. 14,15and 16 according to the forecast formula (38),(39)and the warning formula (40),(41), where x axis denotes the parameter of exponential distribution λ_1, y axis the parameter λ_2, and z axis denotes the probability value.

In order to understand the rule that the parameter of exponential distribution , λ_1 and λ_2, varies with time , we set λ_2 as 1.276 and λ_1 as 0.584 which are the same values as the analysis results in section 3.2, the experimental probability about state keeping invariable and varying with λ_1 or λ_2 and time could also be obtained according to the forecast formula (38),(39), and the warning formula (40),(41), but they are not shown for the limited length of paper.

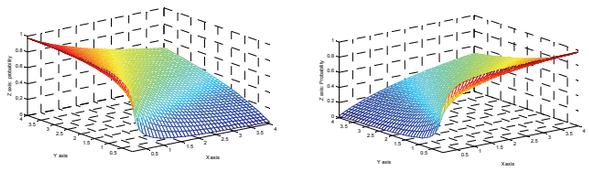
As shown in Fig 14~16, we could safely arrive at the following conclusion: (1) P01 and P11 increases, while P00 and P10 decrease with the increment of parameter λ_1. (2) P00 and P10 increase, while P01 and P11 decrease with the increment of parameter λ_2. (3) P01 and P10 increases, while P00 and P11 decrease with the increment of time t.



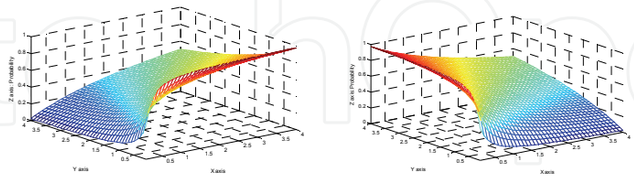
(a) P00 (b) P11

(c) P01 (d) P10

Fig. 14. Experimental probability about forecast and warning formula with t=4s

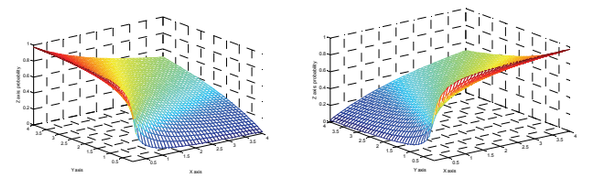


(a) P00 (b) P11

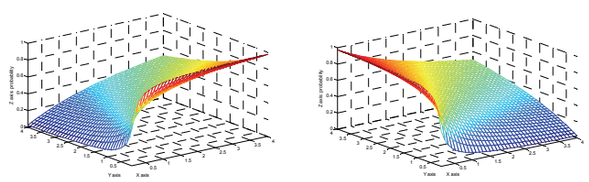


(c) P01 (d) P10

Fig. 15. Experimental probability about forecast and warning formula with t=8s



(a) P00 (b) P11



(c) P01 (d) P10

Fig. 16. Experimental probability about forecast and warning formula with t=10s

If the number of steady period appearing in a certain time is larger than 2, with the increment of parameter λ_1 in passion distribution, the number of steady period appearing will becomes smaller according to the progression theory, that is, the probability of Ad Hoc network topology keeping steady period will decrease. Therefore, P01 and P11 increases, while P00 and P10 decrease with the increment of λ_1 in a certain time. If steady duration time is lager than 1.0s, with the increment of parameter λ_2 in exponential distribution, the steady duration time will become larger according to the progression theory, that is, the probability of Ad Hoc network topology keeping steady period will increase. Therefore, P01 and P11 decreases, while P00 and P10 increase with the increment of parameter λ_2 in a certain time. With the increment of time t, the probability of Ad Hoc network topology keeping its former state(i.e., steady state or un-steady state) will become smaller. Therefore, P01 and P10 increase, while P00 and P11 decrease with the increment of time t with a certain parameters λ_1 and λ_2.

In a practical Ad Hoc network application system, we could use GPS or other position location technology to obtain the position of mobile nodes in any moment, instead of analyzing on the scene file. Next we could use the computing and analysis method in the paper to obtain the dynamic characteristic of Ad Hoc network topology, which could be used for performance evaluation and optimization of Ad Hoc network.

3.2 Performance inference method in ad hoc network

At present, Ad Hoc network performance measurement mainly focus on traditional network intra-measurement technique. [49][50] bring forth to use active probing in Ad Hoc network to measure available bandwidth. [51] puts forward a DEAN (Delay Estimation in Ad Hoc Networks) protocol, in which neighbor nodes uses Hello message to exchange delay time between each other, that is, to measuring delay time needs the collaboration of intra-nodes. All the research above actually is traditional intra-measurement, which has many faults as described above. Above all, there are many theory problems to be solved in Ad Hoc network measurement. At fist, how to deal with the influence of dynamic characteristic of network topology on performance measurement is a key issue. Second, measurement model and inference method is another key issue to be dealt with in NT measurement of Ad Hoc network.

3.3 Performance inference based on linear analysis model

In the process of the performance measurement on Ad Hoc network based on End-to-End measurement technology, the dynamic characteristic of link topology directly influences the measurement results. Yao et al. [52] consider that if the measurement could be completed under the condition that link topology remains relatively invariable, which maybe improve the veracity and precision of performance measurement. In his pervious works, the positions of mobile nodes in Ad Hoc networks at any moment could be obtained through link topology snapshots capturing algorithms according to analyzing on the scenario files of mobility models, and then the serials of snapshots of physical topology could be archived. The different periods during which physical topology is invariable can be gained by analyzing on the snapshots statistically, which is called as measurement window time in this paper. According to the results of analysis on the scenario files of RW, RWP, RGMP, and Manhattan mobility models, we could safely arrive at the conclusion: measurement window time will appear periodically in the whole simulation time.

During measurement window time, since the state of link in Ad Hoc network could not vary, the inference results of link performance based on the samples of End-to-End measurement could reflect the interior link characteristics effectively. Yao et al.[52] call this phenomenon as time validity in the measurement of Ad Hoc network. The next section presents a interior link delay reference algorithm of Ad Hoc network on the basis of End-to-End measurement method[53]. The main content of this algorithm is as followings: First to obtain the measurement time window through a link topology snapshot algorithm, Second to build up a measurement model and linear delay analysis model for Ad Hoc networks, Third to complete End-to-End measurement, Forth to refer interior link delay of Ad Hoc network according to measurement data sample, correlation among mobile nodes in Ad Hoc network topology, linear delay analysis model and mathematical statistics theory.

3.3.1 Link delay linear analysis model

On the assumption that we have done the measurement experiments m rounds. Each round we could get the End-to-End delay vector of receiver i denoted as $Y_i = \{y_{i,1}, y_{i,2}, \cdots, y_{i,m}\}$

($1 \le i \le n$), where n is the number of leaf node, and $y_{i,j}(1 \le j \le m)$ is the sample of stochastic variable $Y_i(i \in [1,n], Y_i \in [0,\infty])$. After the m times experiments have finished, the delay probability distribution of the End-to-End measurement could be obtained as: $P(Y) = \{P(Y_1), P(Y_2), ..., P(Y_n)\}$. If the estimated link delay probability distribution is denoted as $P(X) = \{P(X_1), P(X_2), ..., P(X_v)\}$, then the maximum likelihood function could be expressed as formula (42):

$$L(Y;X) = P(Y_1, Y_2, ..., Y_n; X_1, X_2, ..., X_v) = \prod_{i=1}^{n} P(Y_i;X) \tag{42}$$

When Formula (42) equals to the maximum, we use $\hat{X} = \arg_X \max \prod_{i=1}^{n} p(Y_i;X)$, where $\hat{X} = (\hat{X}_1, \hat{X}_2, ..., \hat{X}_n)$, as the estimated value of link delay X. However, the maximum likelihood estimation algorithm is very difficult to obtain the estimated value of link delay X for computing complexity. In order to obtain the link delay X, [25] adopts the expectation maximum (EM) algorithm including two procedures: E-step and M-step. The main problem about EM algorithm is that it could obtain the partially optimized solution, not the unitary optimized one. For the sake of computing complexity increasing by the scale of network, Pseudo-EM Algorithm[26]decomposes a large scale problem to several small scale ones. The maximum likelihood of these small scale problems could be expressed as:

$$L\ (Y_1, Y_2, ..., Y_n; X) = \prod_{i=1}^{n} \prod_{s \in S} P^s(Y_i^s; X^s) \tag{43}$$

where S is set of all small scale problems. The method to get the solution for Formula (43) is similar to that for Formula (42). The Bayesian estimation method uses the former probability distribution of link delay to infer the posterior one, however, how to get the former probability distribution is a difficult work.

The linear analysis model of delay will be presented next. As we all know, the delay of $path(i \to j)$, denoted as $d(i,j)$, is the sum of all link delay along this path, denoted as $d(k)$ ($k \in F(j) \cup \{j\}$), that is,

$$d(i,j) = \sum d(k)(k \in F(j) \cup \{j\}) \tag{44}$$

In the End-to-End measurement of Ad Hoc networks, the node i belongs to the set of S, and node j to the set of R. The task of link delay inference is to infer $d(k)$ according to the measurement samples of $d(i,j)$. If we only utilize one formula, it is impossible to infer $d(k)$. In order to obtain the link delay, we must use multi-formula and constitute simultaneous equations to resolve the link delay. The simultaneous equations could be expressed as formula (45).

$$Y = AX + \varepsilon \tag{45}$$

Formula (45) is referred to as interior link delay linear analysis model of Ad Hoc network in this paper. In Formula (45), Y is the delay of path which could be obtained or observed in End-to-End measurement procedure. cis the traffic matrix, and ε is the noisy which is ignored in this paper. X is the interior link delay of Ad Hoc network. Our task is that on the condition of Y and A known, ε ignored, how to resolve X. The solution of X is concerned with the types of A. In the next section, we will present the algorithm of link delay inference on the condition of A being square traffic matrix and non-square traffic matrix.

3.3.2 Algorithm of link delay inference

To compute the formula (45) is equivalent to resolve a non-homogeneous linear equations according to linear algebra theory. According to different type of traffic matrix A, we will divide two types(i.e., square traffic matrix and non-square traffic matrix) to discuss how to resolve the solution space for the formula (45) in this section.

Square Traffic Matrix When the traffic matrix is a square one, the solution for the non-homogeneous linear equations as formula (45) is concerned with the rank of the traffic matrix A. If the rank of traffic matrix A is full, there is a unique solution for the non-homogeneous linear equations, otherwise, the question of solution for the equation in section 3.1 is translated to that of a non-square traffic matrix problem. Now we only consider the A as a full rank traffic matrix. At first we could obtain the reverse matrix of A denoted as A^{-1}, the interior link delay can be expressed as formula (46).

$$X = A^{-1} \times Y \tag{46}$$

If the sender node sends N probes to every leaf nodes in Fig. 1(a) respectively, then every link delay in Ad Hoc networks could be achieved according to Formula (46) at different N time. However, we do not care about the link delay at different time, but are concerned about the link delay probability distribution during measurement window time, which could be obtained through analyzing on the link delay statistically during measurement window time based on the discrete link delay time. In practice, it is not possible to construct a square traffic matrix A in Ad Hoc networks. There is only one case that if there are N mobile nodes in Ad Hoc network, only one node is the sender, the other N-1 nodes are all leaf nodes. Under this condition, it is not necessary to use End-to-End measurement technology to infer the link delay, since there is only one step between the sender and leaf nodes, we could obtain the link delay directly through measurement.

Non-square Traffic Matrix When the rank of traffic matrix A is not full, or the traffic matrix A is a non-square matrix, the problem in section A is translated to how to resolve a non-homogenous linear equations. We will discuss this problem from the following two sides. (1) When the rank of the traffic matrix A is not equal to that of its augmentation matrix(i.e., $A|Y$), there is no solution for the non-homogenous linear equations. (2) When the rank of the traffic matrix A is equal to that of its augmentation matrix, there is a solution space for the non-homogenous linear equations. If the traffic matrix A is denoted as $A = (a_{i,j})_{m \times n}$, and rank(A)=rank($A|Y$)=r($r < n$), then the solution space is composed of n-r characteristic solutions (i.e., $\{\eta_i\}(1 \le i \le n-r)$) for the homogenous linear equations and one special solution(i.e., β) for the non- homogenous linear equations. Therefore, the solution space of the non-homogeneous linear equations could be denoted as the following formula (47).

$$S = \sum_{i=1}^{n-r} k_i \times \eta_i + \beta \tag{47}$$

Since the solution space S comprised of infinite solutions, it is necessary to limit the scale of solution space. The link l ($l \in L$) delay inference result is as formula (47), which is shared by χ paths. If the End-to-End delay of the χ paths is denoted as $\mathrm{T}_j (1 \le j \le \chi)$, the minimum delay of the χ paths Γ could be expressed as the following formula (48).

$$\Gamma = \min\{\mathrm{T}_j (1 \le j \le \chi)\} \tag{48}$$

Then the solution space S could be reduced to Ω: $\Omega = \sum k_j \times \eta_j + \beta (\Omega \subset S, 0 \le k_j \times \eta_j + \beta \le \Gamma)$. Next it is similar to the section A that we could obtain any link delay probability distribution through analyzing on the N times of solution space based on the discrete delay time. The unique difference between the square traffic matrix and non-square traffic matrix is that the lessen solution space maybe belongs to many discrete bins, but unique solution only to one bin. The algorithm of interior link delay probability distribution is as the following Algorithm

Step 1. To discrete the link delay time.

Step 2. $Count = 0$, and to compute the rank of traffic matrix A and augmentation matrix $A|Y$. If
$rank(A) \neq rank(A|Y)$ is true, Goto step10.

Step 3. To compute the characteristic solution for the homogenous linear equations as $\{\eta_i\}(1 \le i \le n-r)$

Step 4. To compute the special solution for the non-homogenous linear equations as β

Step 5. To construct the solution space for the non-
Homogenous linear equations as $S = \sum_{i=1}^{n-r} k_i \times \eta_i + \beta$

Step 6. To reduce the scale of S to Ω.

Step 7. $Count++$.

Step 8. If $Count < N$ (N is the times of End-to-End measurement), Go to Step3.

Step 9. To compute the link delay probability distribution through analyzing on the link delay in all N times statistically based on discrete delay time.

Step 10. Finish.

Delay time discrete method Let Θ be a set of finite delay, and link delay time $\theta_j (1 \le j \le 15)$ is discretized to Θ, then θ_j takes a value in Θ. If we suppose that discrete parameter is α, then bin size of delay time is $1/\alpha$, and the set Θ could be defined as following formula (49) based on the fixed bin size delay time discrete model.

$$\Theta = \{0, 1/\alpha, 2/\alpha, \ldots i/\alpha \ldots, 1\} (i \in [0, \alpha]) \tag{49}$$

Then discrete function of delay time could be defined as the following formula (50)

$$Discrete-Function(\theta_j)=\begin{cases} 0 & \theta_j \in [0, 1/2\alpha] \\ i/\alpha & \theta_j \in (\frac{i}{\alpha}-\frac{1}{2\alpha}, \frac{i}{\alpha}+\frac{1}{2\alpha}] \\ 1 & \theta_j \in (\frac{2\alpha-1}{2\alpha}, 1] \end{cases} \quad (i \in [0,\alpha], j \in [1,\infty]) \tag{50}$$

The value of α is an important factor to influence the reference accuracy and computing complexity. If α is small, although more discrete delay time zone and reference accuracy could be obtained, the computing complexity will increase quickly. Otherwise, in despite of computing complexity being reduced, discrete delay time zone and reference accuracy will be reduced. Therefore, it is necessary to make a compromise between computing complexity and reference accuracy according to difference application requirement.

3.4 Link performance inference based on multi-sources measurement

Yao et al.[54] presented a interior link loss rate reference algorithm of Ad Hoc network on the basis of End-to-End and multi-sources & multi-destinations measurement method. The main content of this algorithm is as followings: First to obtain the measurement time window through a link topology snapshot algorithm, Second to build up a measurement model and link loss analysis model for Ad Hoc networks, Third to complete End-to-End and multi-sources & multi-destinations measurement, Forth to refer interior link loss rate of Ad Hoc network according to measurement data sample, correlation among mobile nodes in Ad Hoc network topology, link loss analysis model and mathematical statistics theory. Results of simulation indicate that the loss rate reference algorithm based on multi-sources & multi-destinations measurement is not only better than on one-source & multi-destinations measurement, but also the former has short computing time, which is very adaptable to interior link performance reference for Ad Hoc networks.

3.4.1 Methodology and measurement framework

We make the following assumption on routing behavior[55], (1) The routes from the sources to the destinations are fixed during the measurement period. (2) There is a unique path from each source to each destination. (3) Two paths from the same source to different receivers take the same route until they branch. Two paths from different sources to the same receiver use exactly the same set of links after they join. (4) The routers and switches in the topology obey a first-in first-out policy for packets of the same class. In order to make the assumption A1 more reasonable, we seek to limit probing and keep the measurement period as short as possible. The assumptions A1 and A2 are motivated by the shortest-path nature of routing in the Internet and the situations of the load balancing and multiple-paths are not considered in the paper. The assumption A4 is reasonable as the measurement probes is steady flow from one source to one destination.

Let $P[a,b]$ devotes the path from a to b; $H(p)$ devotes the hop count of the path of p; $SP[s;i,j]$ devotes the shared path of paths from the source s to the destinations i and j; $SP[i,j;d]$ devotes the shared path of paths from the sources i and j to the destination d; $P^{\lceil h \rceil}[a,b]$ devotes the portion path of $P[a,b]$ with hop count is h and the path begins from a;

$P^{\lfloor k \rfloor}[a,b]$ devotes the portion path of $P[a,b]$ with hop count is *h* and the path ends with *a*; $\ell^{\lceil h \rceil}[a,b]$ devotes the hth link in the path $P[a,b]$. Let $\psi(p)$ ($\varphi(p)$) devotes the minimal delay of the large packets (the small packet) which probe the path *p*. The minimal delay (also called stable delay) includes the propagation delay, transmission delay and the stable processing delay and does not include the queuing delay. In this paper, the probe with the minimal delay is called the valid probe, and the size of the small packet is set 56 bytes (the minimum packet size in IP) and the size of the large packet is set 1500 bytes (the maximum packet size in IP). Let $\lambda(p)$ devotes the minimal delay difference of the path *p* measured by the large packet and the small packet, so $\lambda(p) = \psi(p) - \varphi(p)$. Then $\lambda(p)$ is a path metric and has monotonicity and separability properties.

The main process of the new methodology to identify the routing topology includes four steps. Firstly, we calculate the hop count of the path from the each source to each destination. Secondly, we infer the hop count of the share path for every 1-by-2 component. Thirdly, we infer the hop count of the share path for every 2-by-1 component. Fourthly, the routing topology is constructed by the topology construction algorithm based on hop count information

Hop count of a path In this step, we calculate the hop count of the path from the source *i* to the destination *j* by subtracting the left TTL value of a packet received by destination *j* (devoted by ttl_j) from the initial TTL value (devoted by ttl_0).

$$H(P[i,j]) = ttl_0 - ttl_j$$

One source to two destinations In this step, which contains two sub steps, we consider a single source (devoted by s_0, $s_0 \in S$) transmitting probes to two destinations (devoted by *i* and *j*, $i, j \in R$). In first step, as depicted in Figure 3(a), s_0 sends back-to-back packet pairs with the large packet destined for *j* and the small packet destined for *i*, in which the large packet is followed closely by the small packet. As the large packet and the small packet will be separate at the branching node, the share path of the large packet and the small packets is $SP[s_0;i,j]$. If the packet pair do not suffer the queuing delay, then

$$d_{s_0;i,j} = \psi(SP[s_0;i,j]) + \varphi(P^{\lfloor H(P[s_0,i]) - H(SP[s_0;i,j]) \rfloor}[s_0,i])),$$

where $d_{s_0;i,j}$ devotes end-to-end stable delay of the small packet.

In second step, we measure the delay difference of very physical link in the path from the source to the destination using a serial of the back-to-back packet pair, in which the large packet with the initial TTL value ttl_0 from 0 to $H(P[S,i])$ (specially , when the ttl_0 is set 0 the source does not send the large packet)is followed closely by the small packet with the initial TTL value larger than $H(P[S,i])$. As the large packet will be discarded by the internal node when the TTL number is reduced to zero, the share path of the large packet and the small packets is $P^{\lceil ttl_0 \rceil}[s_0,i]$. If the packet pair do not suffer the queuing delay, we get the relationship of hop count and the delay difference,

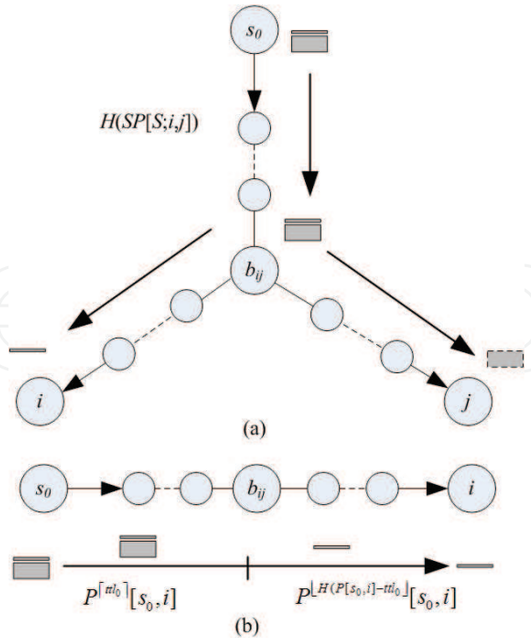


Fig. 17. Packet pair probes for every 1-by-2 component. (a) The meathead of probing the delay difference of the share path $SP[s_0;i,j]$ of each 1-by-2 component in the first sub step. (b) The meathead of probing the delay difference of every physical link in the path from s_0 to i in the second sub step.

$$d_{s_0,i}^{ttl_0} = (\psi(P^{\lceil ttl_0 \rceil}[s_0,i]) + \varphi(P^{\lfloor H(P[s_0,i])-ttl_0 \rfloor}[s_0,i])),$$

$$\begin{aligned}\lambda(\ell^{\lceil h \rceil}[s_0,i]) &= \psi(\ell^{\lceil h \rceil}[s_0,i]) - \varphi(\ell^{\lceil h \rceil}[s_0,i]) \\ &= (\psi(P^{\lceil h \rceil}[s_0,i]) + \varphi(P^{\lfloor H(P[s_0,i])-h \rfloor}[s_0,i])) \\ &\quad -(\psi(P^{\lceil h-1 \rceil}[s_0,i]) + \varphi(P^{\lfloor H(P[s_0,i])-(h-1) \rfloor}[s_0,i])) \\ &= d_{s_0,i}^{ttl_0} - d_{s_0,i}^{ttl_0-1}\end{aligned}$$

where $d_{s_0,i}^{ttl_0}$ devotes end-to-end stable delay of the small packet with the TTL value of the large packet is ttl_0. Meanwhile, as depicted in Figure 4, we can get the follow formula:

$$d_{s_0;i,j} = d_{s_0,i}^{H(SP[s_0;i,j])}.$$

So we can infer the hop count of the share path by

$$H(SP[s_0;i,j]) = \arg_{ttl_0 \in [0,H(P[s_0,i])]} \min\{\left| d_{s_0;i,j} - d_{s_0,i}^{ttl_0} \right|\}.$$

Let $M(i,ttl_0,K)$ devotes the *digging* measurements process, in which s_0 sends packet pair destined to i and large packets with initial TTL value ttl_0 and K measurements are collected in total. For each measurement $k = 1,2,...,K$, let $x_{s_0,i}^{ttl_0}(k)$ denotes measured delay time, then

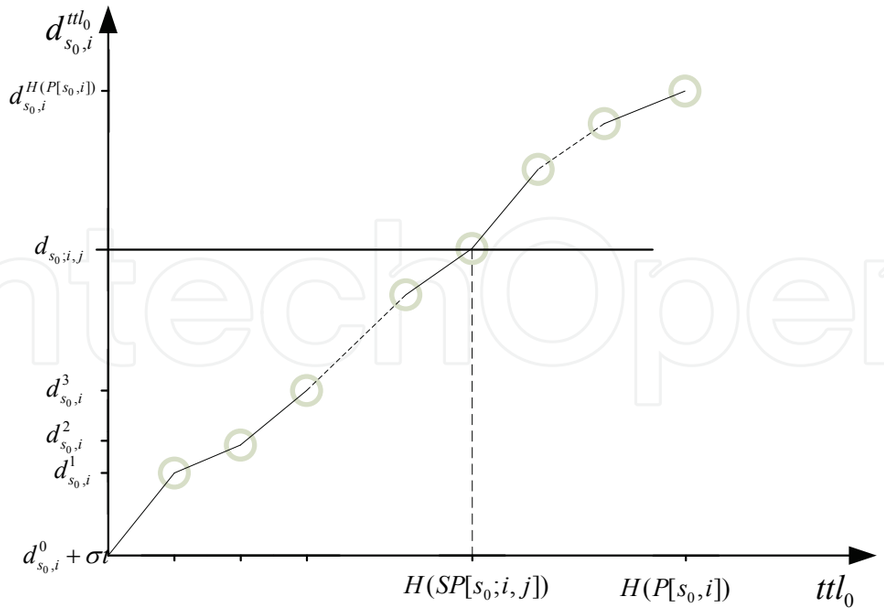


Fig. 18. The relationship of the ttl_0 and the delay difference. As the internal nodes may have different bandwidth, the increase values of delay difference produced at internal nodes may be not equal. So the points corresponding to delay differences are not placed on a straight line.

$$x_{s_0,i}^{ttl_0}(k) = d_{s_0,i}^{ttl_0} + q_{s_0,i}^{ttl_0}(k) + \sigma t_{s_0,i},$$

Where $q_{s_0,i}^{ttl_0}(k)$ devotes the queuing time, $\sigma t_{s_0,i}$ devotes the clock difference between the nodes s_0 and i. Similarly, let $x_{s_0;i,j}(k)$ denotes measured delay time of the share path in the first sub step, then

$$x_{s_0;i,j}(k) = d_{s_0;i,j} + q_{s_0;i,j}(k) + \sigma t_{s_0,i}.$$

For each type measurement we assume that these measurement results are independent and identically distributed; this assumption is reasonable if the probes are sufficiently separated in time. Then the hop count of the share path can be inferred by ttl_0 which makes the difference of minimal delay of packets (meathead 1) or the difference of the mean delay (meathead 2) reach the minimum value.

$$\hat{H}(SP[s_0;i,j]) = \arg_{ttl_0 \in [0,H(P[s_0,i])]} \min\{\left|\min_{k'=1,2,\ldots,K'}\{x_{s_0;i,j}(k')\} - \min_{k=1,2,\ldots,K}\{x_{s_0,i}^{ttl_0}(k)\}\right|\}$$

$$\hat{H}(SP[s_0;i,j]) = \arg_{ttl_0 \in [0,H(P[s_0,i])]} \min\{\left|\frac{1}{K'}\sum_{k'=1}^{K'} x_{s_0;i,j}(k') - \frac{1}{K}\sum_{k=1}^{K} x_{s_0,i}^{ttl_0}(k)\right|\}$$

To simply the inference process and reduce the probing traffic load, we use the binary search algorithm to search $\hat{H}(SP[s_0;i,j])$ as the $\min_{k=1,2,\ldots,K}\{x_{s_0,i}^{ttl_0}(k)\}$ and $\frac{1}{K}\sum_{k=1}^{K} x_{s_0,i}^{ttl_0}(k)$ have monotonicity property when K is large enough.

Two sources to one destination For two sources (devoted by i and j $i,j \in S$), the main process of measurement in our new methodology to infer the hop count of the share path to one destination (devoted by d_0, $d_0 \in R$) also includes two sub steps and the first sub step is the same to the second sub step in second step above to measure $d_{i,d_0}^{ttl_0}$.

In the second sub step, we measure the stable delay difference of the share path $SP[i,j;d_0]$. As depicted in Figure 5, the sources i and j send small packet and the large packet destined for d_0 periodically.

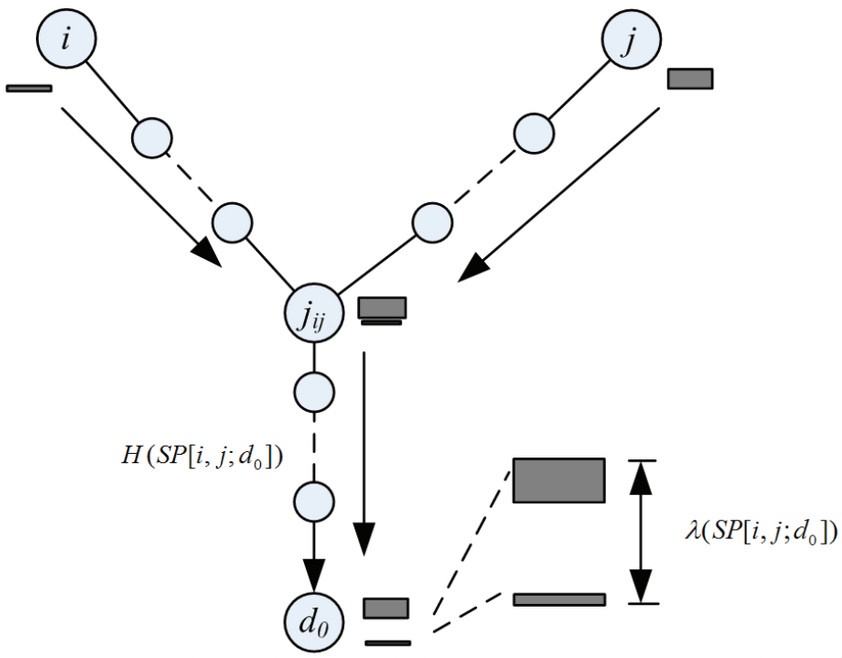


Fig. 19. The meathead of probing the delay difference of the share path of each 2-by-1 component. The sources i and j send small packet and the large packet destined d_0. If the large packet reach the joining node just after the small packet, then the interval equals the delay difference when they reach d_0.

Let sm_i and sm_j devotes the sending moment of the packet by the node i and the node j in the measurement periods; rm_{is} (rm_{il}) and rm_{js} (rm_{jl})devotes the receiving moment of the small packet (the large packet) sent by the node i and the node j; $x_s(k)$ ($x_l(k)$) devotes the measured delay time of the small packet (the large packet).

If the small valid packet and the large valid packet reach the joining node almost synchronously (this mean the interval time between and two packet is smaller than the minimal link delay difference in the path from the source to destination.), then difference of the received moment equals the stable delay difference on the share path.

$$\lambda(SP[i,j;d_0]) = rm_{is} - rm_{jl}$$

$$H(SP[i,j;d_0]) = \arg_{ttl_0 \in [0,H(P[s_0,j])]} \min\{\left|(rm_{is} - rm_{jl}) - (d_{j,d_0}^{H(P[s_0,j]} - d_{j,d_0}^{ttl_0})\right|\} .$$

Then we can infer the hop count of the share path by

$$\hat{H}(SP[i,j;d_0]) = \arg_{ttl_0 \in [0,H(P[s_0,j])]} \min\{\left|(rm_{is} - rm_{jl}) - (\min\{x_{j,d_0}^{H(P[s_0,j]}(k)\} - \min\{x_{j,d_0}^{ttl_0}(k)\})\right|\} .$$

or

$$\hat{H}(SP[i,j;d_0]) = \arg_{ttl_0 \in [0,H(P[s_0,j])]} \min\{\left|(rm_{is} - rm_{jl}) - (\frac{1}{K}\sum_{k=1}^{K} x_{j,d_0}^{H(P[s_0,j]}(k) - \frac{1}{K}\sum_{k=1}^{K} x_{j,d_0}^{ttl_0}(k))\right|\}$$

To synchronize the valid packets of the same size from the two sources to the destination to reach the joining node synchronously, one source only need to adjust the sending time forwards (or backwards) by the difference of the received moment of the two packets, because if packets reach the destination synchronously, they must have reached the join node synchronously.

To synchronize the valid packet of different size, we use the *synchronization measurement process* to adjust the sending moment. As the order of the valid packets reaching the joining node will remain to the destination, the destination can tell which valid packet reached the joining node firstly (secondly), and then inform the source to adjust the sending time backwards (forwards). In the synchronization measurement process, we keep the same sm_i and change sm_j using binary search algorithm to make the small valid packet and the large valid packet reach the joining node closely enough. To accelerate search process and to reduce the probing traffic load, we make *advance measurements* and use the measurement result to set the appropriate upper bound and the lower bound of sm_j.

In the advance measurements, as depicted in Figure 20, *j* sends small packets and large packets alternately, and *i* sends only small packets with the same period.

If we change the sending moment of *j* from sm_j to $sm_j + rm_{is} - rm_{js}$, the small valid packets from the two sources will reach the joining node at the same. Meanwhile in another period the small packet from *i* will reach the joining first than the large packet from *j*. So the upper bound of sm_j can be set $sm_j' + rm_{is} - rm_{js}$. In the same way, if we the change sending moment of node j to $sm_j' + rm_{is} - rm_{jl}$, then the small packet from *i* and the large packet from *j* will reach the destination at the same time, that means the large packet reaches the joining packet firstly. So the lower bound of sm_j can be set $sm_j' + rm_{is} - rm_{jl}$. After the advance measurements , the range of the sm_j is limited to $\lambda(P[j,d_0])$.

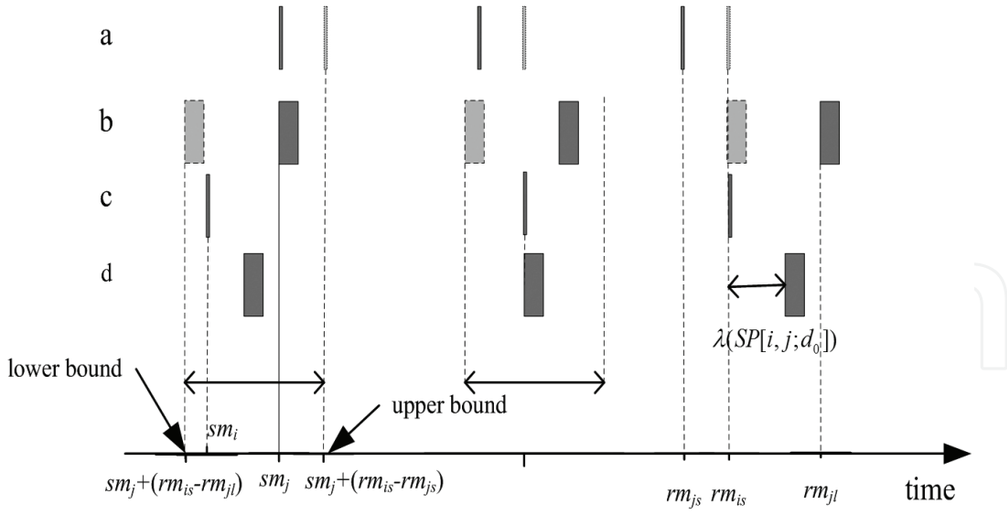


Fig. 20. The sending moment at the *i* and *j*, reaching moment at the joining node and the receiving moment at the d_0 of valid probes in the advance measurements. The packets in lines a, b and c are the small packet sent by *j*, the large packet sent by j and the small packet sent by *i*. The packet in line d is the synchronized large packet sent by *j*.

Algorithm computer the $\hat{\lambda}(SP[i,j;d_0])$

Input: the sources *i*, *j* and the destination d_0 .

Output: $\hat{\lambda}(SP[i,j;d_0])$

Precess:

1. Set the initial sending moment sm_i and sm_j , and let *j* sends small packets and large packets alternately, and *i* sends only small packets with the same period. Make measurements for K times.
2. Set $high = sm_j' + \min_{k\in[1,K]}\{rm_{is}(k)\} - \min_{k\in[1,K]}\{rm_{js}(k)\}$,

   $low = sm_j' + \min_{k\in[1,K]}\{rm_{is}(k)\} - \min_{k\in[1,K]}\{rm_{jl}(k)\}$, $sm_i = sm_i'$,.

   While ($high - low < \min_{h\in[1,H(P[j,d_0])]} \hat{\lambda}(\ell^{\lceil h \rceil}(j,d_0))$) Do

   $mid = \lceil (low + high)/2 \rceil$;

   $sm_j = mid$;

   Let *j* sends small packets and *i* sends small packets periodically for K times;

   If ($\min_{k\in[1,K]}\{rm_{is}(k)\} < \min_{k\in[1,K]}\{rm_{jl}(k)\}$)

   Then $high = mid$;

   Else $low = mid$

   End If

   End While

   $sm_j = low$;

Let *j* sends small packets and *i* sends small packets periodically for K times;

Return $\min_{k\in[1,K]}\{rm_{jl}(k)\} - \min_{k\in[1,K]}\{rm_{is}(k)\}$;

Algorithm topology Identification

Input: the source set S and the destination set D; $\hat{H}(P[i,j])$, $i \in S$, $j \in D$; $\hat{H}(SP[i,j;d])$, $i,j \in S, i \neq j$, $d \in D$; $\hat{H}(SP[s;i,j])$, $s \in S$, $i,j \in D, i \neq j$; topology $G =< S \bigcup D, \varnothing >$.

Output: identified topology G

Precess:

1. For (each node i in S)
   For (each node j in D) Do
   Inset $\hat{H}(P[i,j]) - 1$ nodes and $\hat{H}(P[i,j])$ links in the path from i to j;
   End For
   End For
2. For (each node s in S)
   For (each two nodes i and j in D) Do
   Merge $P^{\lceil \hat{H}(SP[s;i,j]) \rceil}[s,i]$ and $P^{\lceil \hat{H}(SP[s;i,j]) \rceil}[s,j]$;
   End For
   End For
3. For (each node d in D)
   For (each two nodes i and j in S) Do
   Merge $P^{\lfloor \hat{H}(SP[i,j;d]) \rfloor}[i,d]$ and $P^{\lfloor \hat{H}(SP[i,j;d]) \rfloor}[j,d]$;
   End For
   End For
   Return G

Delay Measurement and Clock Synchronization The methodology above need the condition that the clock of the measurement node have higher timing precision than the size of table delay difference of one-hop in the path from the source to destination. Furthermore, the system errors (such as the location errors) will be eliminated we computer the table delay difference, so only the random error influence the methodology accuracy. If the maximal bandwidth of link in the path is 1Gb/s, the timing precision should be higher than 10us which can be realized based on general PC[17]. So our methodology can be applied widely and has lower measurement cost than the meathead that need the assumption that the source and the destination have the strict clock synchronization, as to satisfy the assumption need deploy costly GPS receivers for every measurement node.

Probing Traffic Load For the M-by-N network, the probe number of the probe packet can be computed by the follow formula approximately:

$$probing\ number = MN + 2KM\binom{N}{2} + 2KMNE(h) + 2KN\binom{M}{2}\log_2(E(h)E(\frac{\max wd(\ell)}{\min wd(\ell)}))$$

where $E(h)$ devotes the average hop count of the paths from the sources to the destinations and $E(\frac{\max wd(\ell)}{\min wd(\ell)})$ devotes the average ratio of the maximal bandwidth and minimal

bandwidth of the links in the path the sources to the destinations. As usually during several seconds measurement time we can get the stable delay with a high probability [18, 19, 20], the value of K can be set from 50 to 100. In many case, the value of $E(h)$ ranges from 5 to 20 and the value of $\log_2(E(h)E(\frac{\max wd(\ell)}{\min wd(\ell)}))$ ranges from 5 to 10.

3.4.2 Simulation study

Firstly, we make simulations for the 1-by-2 component and 2-by-1 component, using the simulation tool *ns-2*. The hop count of every logical link ranges from 3 to 10. The physical link bandwidths range from 100Mb/s to 1000Mb/s. The background traffic added to every physical link is poisson traffic or self-similar traffic generated by three pareto traffics with $\alpha = 1.9$. Simulations were conducted in a low utilization scenario, a medium utilization scenario and a higher utilization scenario (by varying background traffic). In the first scenario, the average utilization over every physical link and runs was 10%, with a range of 5-15%; in the second scenario the average was 30 %, with a range of 10-50%; in the last scenario, the average was 50 %, with a range of 30-70%. As there are many physical links between the source and the destination, the average utilization of every logical link in three scenario are 45%, 90%, 99%.

The packet size in background ranges from 56byte to 1500byte; the small packet size of probe is 56 bytes and the large packet size of probe is 1500bytes. For every scenario, the simulation runs 200 times. The correctness of identification is depicted in Figure 8, Figure 9, and Figure 10. Generally speaking, the correctness increases quickly with the number of the probe packet and tend to 100%. The correctness of 1-by-2 component identification is higher than the correctness of 2-by-1 component identification, which can be improved by increasing the synchronization measurement process in the binary search algorithm. The background traffic become more unstable, the meathead1 adapt it better than meathead2.

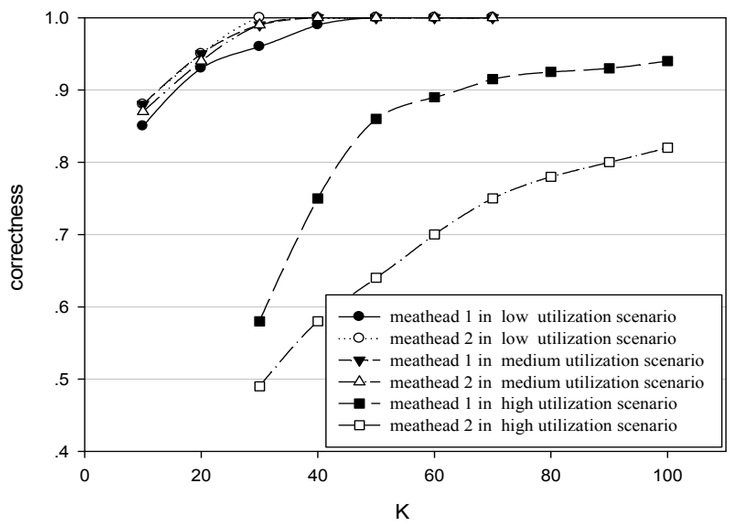


Fig. 21. The correctness of the identification vs. the number of the probes for 1-by-2 component with poisson background traffic.

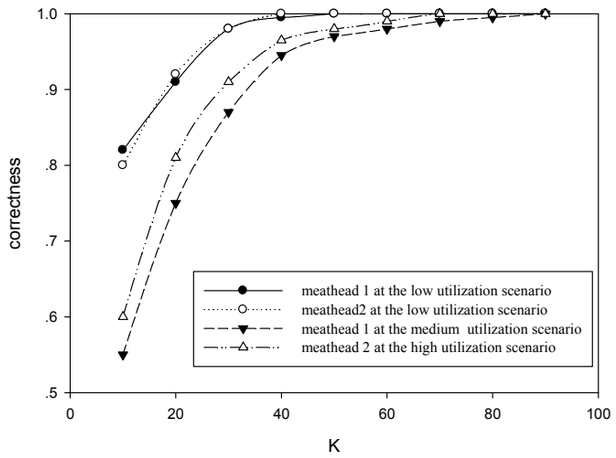


Fig. 22. The correctness of the identification vs. the number of the probes for 2-by-1 component with poisson background traffic.

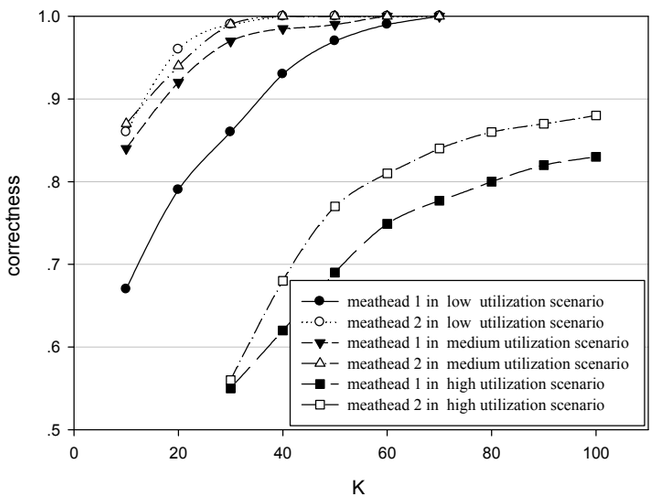


Fig. 23. The correctness of the identification vs. the number of the probes for 2-by-1 component with self-similar background traffic.

Secondly, we make simulations for a 3-by-4 network depicted in the Fig. 1(a) with the medium utilization poisson background traffic. We using meathead1 and set K=50. The simulation runs 200 times and the correctness of identification is 98%.

3.5 Ad hoc network delay tomography based on circle mobility model

The circle mobility model (CMM) is proposed by Wang[56], which is suited to patrolling periodically for gathering information in the military area or forest fireproofing. The assumptions are as follows. First, the location of SN is known to other nodes. Second, the MN knows the direction and how far it will move to next destination. This is true in real

situations where nodes know their destinations. The process, representing the movement of a node within a circular area A with radius R, can be described as follows. A SN (n_0) is placed at the point O, the centre of the circle. Initially, MN (n_i, n_j and n_k) are placed at points over A, see Fig.1-a. Without losing the generality，n_i is in initial position P_{i0} with radius r_i. Then a destination point P_{i1} is chosen from the circle with radius r_i and the node moves along arc L_i with constant velocity v_i and central angle θ_i. Once n_i reaches P_{i1}, n_i stays a pause time t_i and a new destination point P_{i2} is drawn, ...$P_{i(n-1)}$,P_{in}.... Obviously, the step time $ST_i = \theta_i / v_i + t_i$ and the step length $L_i = \theta_i r_i$. The nodes (n_i and n_j) with same radius have identical mobile properties, such as $\theta_i = \theta_j$, $v_i = v_j$, $t_i = t_j$, $ST_i = ST_j$ and $L_i = L_j$, otherwise they might have different mobile properties.

3.5.1 Circle mobility model

The MANET with CMM could be denoted as a dynamic logical tree $\Psi = (V, L(T))$ with the node set V and link set $L(T)$ at time T [2]. A source node to probe is called the root. A set of receivers, which called leaves, is denoted as $RCE \subset V$. The nodes between the source and receivers represent internal nodes. The tree model is defined by the set of paths. Each path, which is from the root to an end receiver denoted by $rce \in RCE$, comprises one or more links (direct connections with no intermediate nodes). A logical link is referred to as a subpath in which every internal node has only one child. In Ψ, the interal links are the logical links that link these branch nodes at T. Each node k, apart from the root, has a parent $f(k)$ such that link $(f(k), k) \in L(T)$ could be denoted as link k. The physical topology and the logical topology are depiced in Fig. 24(c) and Fig.24(d) respectively. The Fig.24(d) shows a typical binary tree in which the root and the receivers are 1, 3 and 4.

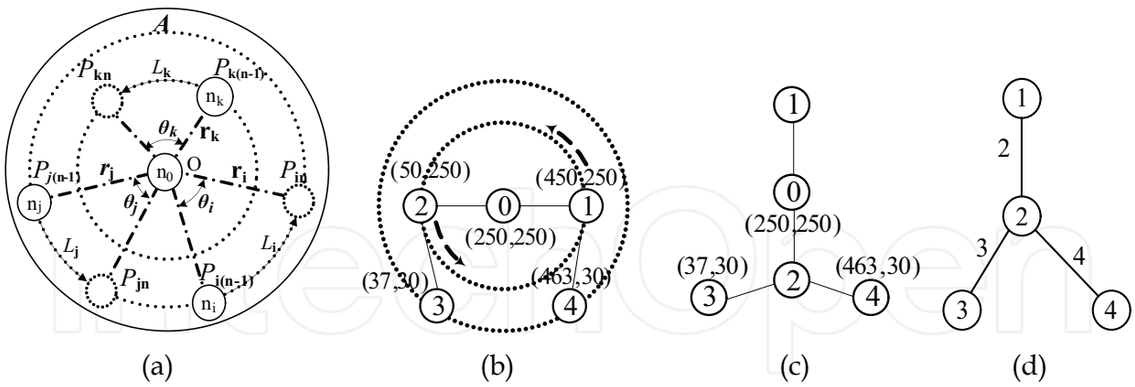


Fig. 24. The CMM mobility model and the topology used for the NS2 measurements (a) CMM mobility model. (b)Initial topology and nodes coordinates. (c) Physical topology within IT_1 and IT_2. (d) Logical topology inferred within IT_1 and IT_2.

There are many random components determining link delay, such as propagation delay, queuing at the node, node packet servicing delay and dropped event due to the overload of finite output buffer of the node or link breakage. The key assumption of our delay model is that the individual delays between different links and packets should be considered independently within IT. IT means the period of time during which the topology is

relatively stable under CMM, to overcome the stubborn topology changes. A series of *IT*s can be calculated during the simulation period due to the breakage and comebacke along the paths. The internal link delay could be inferred which is associated with the corresponding *IT* by probing two closely time-spaced packets (back-to-back packet pair) from the source to two different receivers. This is the difference between our delay model and previous works [4]. Each members of a packet pair passes through a common set of links in their paths, but diverge at some branch node to arrive to the respective destination. Apparently, if the gap between the two members is on the order of the machine's smallest unit of time, the difference between the delay experienced by the first and the second member of pair crossing same link can be ignored. Approximatively, the two packets experience identical network conditions along the shared links and any delay experienced will be identical for both probes.

During the *i*-th *IT* denoted by $IT_i = t_{end}^{(i)} - t_{start}^{(i)} > 0$ ($t_{start}^{(i)}, t_{end}^{(i)} \in T$), two members of the packet pair (i, j) are sent to destination rce_i and rce_j, respectively. Since the round trip paths of the probe are unsymmetrical, a measurement represents the E2E one way delay (OWD) of the couple of packets, denoted by $X_{ij}^{IT_i} = (X_i^{IT_i}(1), X_j^{IT_i}(2))$. Where $X_i^{IT_i}(1)$ and $X_j^{IT_i}(2)$ are the delays from source to the two end receives, respectively. The sending time is stamped on every packet by the sender, and the OWD is calculated at the receiver. An experiment consists in sending n packets pairs (i, j) for each pair of rce_i and rce_j. The set of measurements, the cumulated delay along the respective paths are associated with $X^{IT_i} = (X_{rec_i}^{IT_i(m)}(1), X_{rec_j}^{IT_i(m)}(2))_{m=1,\ldots,n;rec_i \neq rec_j \in RCE}$ for each couple of end receivers. The complete set of measurement X^{IT_i} is obtained by combining all possible pair of distinct rce_i and rce_j in Ψ winthin IT_i.

3.5.2 Link dealy probability distribution inference method

Let $D_{1k}^{IT_i}$ and $D_{2k}^{IT_i}$ represent the estimated value of delay over link *k* in MANET for the first member and for the second one of the packet pair during the IT_i. Since the distribution of a link delay is unknown, the characterization of the variable delay is obtained by non-parametric discrete distributions. The delay could be quantified as a finite set of possible delay $Q = \{0, q, 2q, ..., Mq, \infty\}$, where q, M and ∞ denote the width bin, a positive integer and the lost, respectively. The bin associated to $iq \in Q$ is the interval $[iq - q/2, iq + q/2]$, where $i = 0,1,2\ldots, M$. Many similar delays are grouped in a unique interval. The estimation of $D_k^{IT_i}$ is the probability of these intervals, denoted by $\alpha_k^{IT_i} = (P[D_k^{IT_i} = d])_{d \in Q}$. Our goal is to estimate $\alpha^{IT_i} = (\alpha_k^{IT_i})_{k \in V}$. Let D^{IT_i} be the set of delays experienced by the packet pairs along each link. It is possible to define the log-likelihood function for the pair (X^{IT_i}, D^{IT_i}) of the measurement X^{IT_i}, which is the complete data for inference problem:

$$L(X^{IT_i}, D^{IT_i}, \alpha^{IT_i}) = \log P_\alpha[X^{IT_i}, D^{IT_i}] = \sum_{k \in V} \sum_{d \in Q} n_k(d) \log \alpha_k^{IT_i}(d) \tag{51}$$

Where $n_k(d)$ is the number of packet pairs with delay *d* over link *k*. We estimate $\hat{\alpha}_k^{IT_i}(d) = n_k(d) / \sum n_k(d) = n_k(d)/n$ could be estimated by formula (51) with Maximum

Likelihood Estimate. Although $n_k(d)$ is an unknown value, the maximum of formula (51) could be estimated by using the Expectation Maximum algorithm.

① ***Initialization***. Calculate the IT_i to infer Ψ according to CMM and select the initial delay distribution $\hat{\alpha}^{IT_i(0)}$. First, the positions of the end nodes (source and receivers) are calculated by movement parameters along respective circles at time $t \in T$. At the same time *t*, the coordinates of the other MN between source and recivers can be obtained. Second, comparing the distance between nodes and the radio link range, the topology whose life time is from $t_{start}^{(i)}$ to $t_{end}^{(i)}$ could be inferred. The distribution of $\hat{\alpha}^{IT_i(0)} = P[D_k^{IT_i} = d]$ is the initial distribution for the iterative EM algorithm.

② ***Expectation.*** Let X^{IT_i} be discretized to the set Q. The measurement $x_{rce_i,rce_j}^{IT_i}$ depends on rce_i and rce_j, simply $x_{rec}^{IT_i}$. Using theorem of Bayes, $\hat{n}_k(d)$ could be derived as the following:

$$\begin{aligned}\hat{n}_k(d) &= \sum_{h=1}^{n} P_{\hat{\alpha}^{(l)}}[D_k^{IT_i} = d \mid X^{IT_i} = x_{rec}^{IT_i(h)}] \\ &= \sum n(x_{rec}^{IT_i})(P_{\hat{\alpha}^{(s)}}[X^{IT_i} = x_{rec}^{IT_i} \mid D_k^{IT_i} = d] \Big/ P_{\hat{\alpha}^{(s)}}[X^{IT_i} = x_{rec}^{IT_i}])\hat{\alpha}_k^{IT_i(s)}(d)\end{aligned} \tag{52}$$

Where $n(x_{rec}^{IT_i})$ is the number of times of the same discretized measurement in $x_{rec}^{IT_i}$. The count $\hat{n}_k(iq)$ for each $iq \in Q$ can be calculated in the formula (2). The iterative algorithm is expressed by the distribution $\alpha_k^{IT_i}$ computed at step *s*.

③ ***Maximization.*** The conditional expectation calculation of $\hat{\alpha}^{IT_i(s+1)}$ maximizes the function $L(X^{IT_i}, D^{IT_i}, \alpha^{IT_i})$, given X^{IT_i} and $\hat{\alpha}^{IT_i(s)}$. It is possible to obtain the new estimate at *(s+1)*-th step, using the $\hat{n}_k(d)$ in ②.

④ ***Iteration***. The joint application ② and ③ gives the stationary solution of the maximization, and $|\hat{\alpha}^{IT_i(s+1)} - \hat{\alpha}^{IT_i(s)}| < threshold$, where *threshold* allows the algorithm to know if the maximum is reached. Although the smaller *threshold* means the estimations are more precise, complicated calculations will be produced. In our simulations (Section 4), $threshold = 0.01$.

3.5.3 Simulation study

The NS2 simulator could be extended to simulate the traffic through CMM model with simulation time of 150s. MANET with two-receiver (3 and 4) is depicted in the Fig. 1-b. We simulate 2 scenarios by the CMM in a rectangular field (500m × 500m) with 5 nodes and *R*=250m. We let 3 and 4 be static (v_3=v_4=*0*) for simplification, but these nodes is mobile for scalability and inferences algorithm. The MN (1 and 2) with $r = 200$, $\theta = 10^{\circ}$, $v = 12m/s$ and $t = 1s$. Radio propagation range for each node is 250 meters and channel capacity is 1.5 *Mb/s*. We can easily infer Ψ throughout $IT_1 = 8s$ ($t_{start}^{(1)} \approx 32s$, $t_{end}^{(1)} \approx 40s$) and $IT_2 = 8s$ ($t_{start}^{(2)} \approx 136s$, $t_{end}^{(2)} \approx 144s$), see Fig.1-d. A link between two nodes shows that the two nodes can hear each other within IT_1 and IT_2. The probes comprise packet pairs with a 0.15*s* inter-pair time. The packet pairs were CBR with an inter-packet time of 0.1 microseconds by periodically sent to 3 and 4.

In scenario 1, background traffic consists of 2 TCP connections (2 to 1, 4 to 3) and the source is 0 and *q=0.1s*. In scenario 2, the source is 1, *q=0.05s*, and background traffic consists of 2 TCP connections (0 to 3, 4 to 1). The typical initial delay probability of every link can be chosen by the uniform distribution from 0 to *M*.

Fig.25 shows the simulation results plotted by Matlab6.5 along links (2,3 and 4). From left to right show results for link 2, link 3 and link 4. The estimated delay and actual delay are indicated with white and black, respectively. Obviously, internal links' actual average delays with high probability (>0.1) accord with estimated average delay. Since the complexity of the analysis is a function of the numbers of bins, a small *q* to ensure a desired level of accuracy results in excessive computational costs.

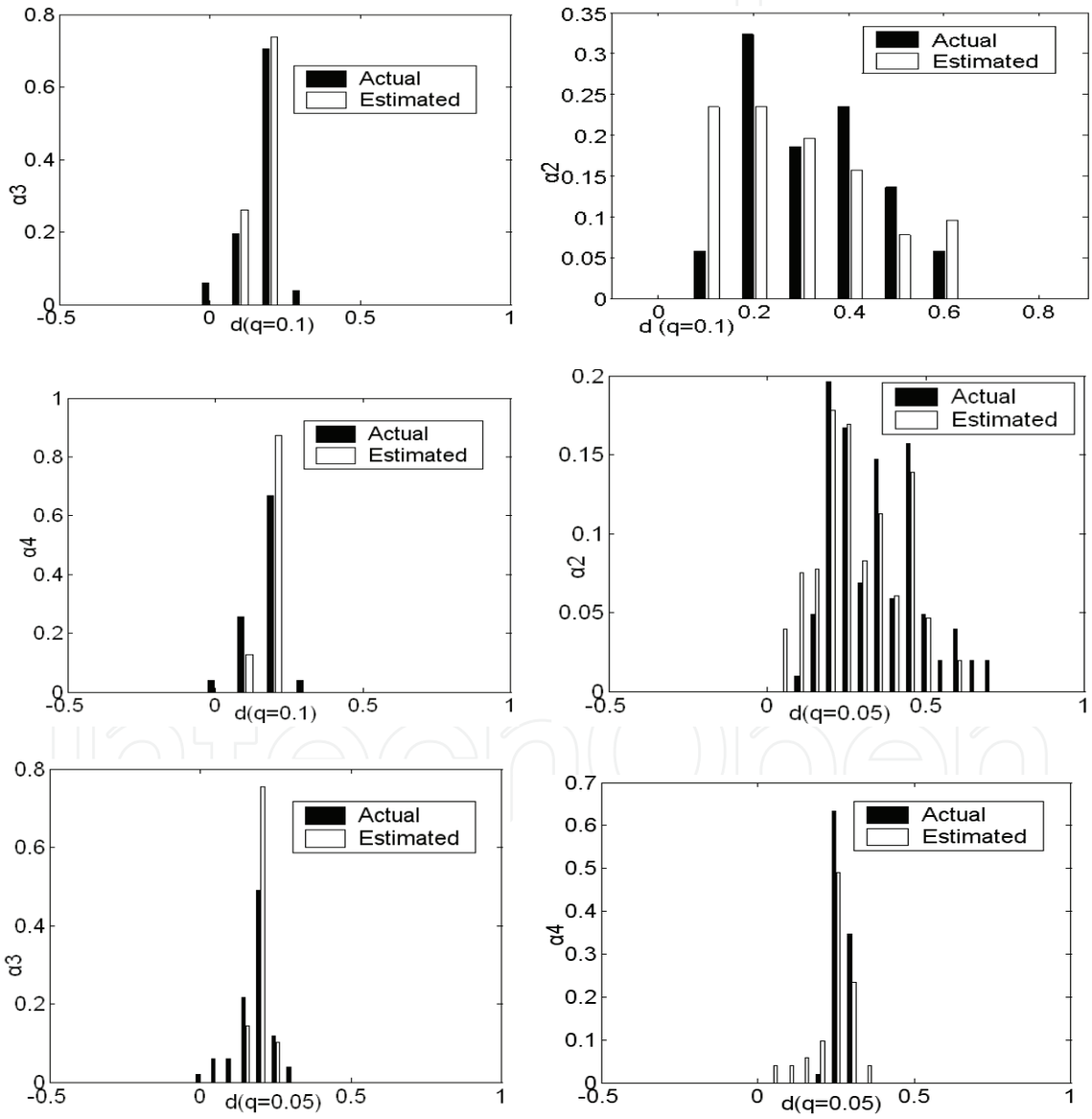


Fig. 25. Estimated vs. actual delay probability distributions from each scenario.

4. Acknowledgment

The research is supported in part by the National high technology research development plan (863 plan) of china (Grant No. 2009AA01Z424), and the Ph.D. Programs Foundation of Ministry of Education of China (Grant No. 200806990030), and the Innovative Foundation of Northwestern Polytechnical university in P.R.China(2008KJ02028).

5. References

[1] Duan Qi, Cai Wanddong. BC_EM: A Link Loss Inference Algorithm for Wireless Sensor Network. 5th International Conference on Wireless Communications, Networking and Mobile Computing(WiCom '09), p1–5, 2009.9,Beijing

[2] Duan Qi, Cai Wanddong. A Simple Graph-structure Network Tomography Topology Identification Method. 2009 International Joint Conference on Artificial Intelligence(JCAI '09), p 337-340, 2009.4,Haikou

[3] Li Guishan, Cai Wandong et al. Link Delay Estimation in Network with Stochastic Routing. 2009 WRI World Congress on Computer Science and Information Engineering, v2, p 110-114,2009.3, Los Angeles USA.

[4] Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data.J. Amer. Stat. Assoc., 91(433): 365-377, 1996.

[5] Yongjun Li, Wandong Cai et al. Wireless Sensor Network Topology Identification based on Data Aggregation. Journal of Computational Information Systems, 3(6), p2359-2365, 2007

[6] LI Yong-jun CAI Wan-dong WANG Wei TIAN Guang-li, Research on network topology identification algorithm based on end-to-end loss performance, JOURNAL ON COMMUNICATIONS, 28(10), 2007

[7] Zhao tao, Cai wandong, Li huixian, Sensor network level-topology inference based on Hamming distance, Journal of Huazhong University of Science and Technology(Nature Science) , 36(10): 71-74, 2008

[8] Tao Zhao, Wandong Cai et al. Topology control for wireless sensor network. 2007 IFIP International Conference on Network and Parallel Computing(NPC 2007),p343-348,2007.9, Dalian

[9] Li Yongjun, Cai Wandong, Wang Wei, Tian Guangli, Topology Identification Based on End-to-End Link Utilization, Journal of System Simulation, 18(22)

[10] Yongjun Li, Wandong Cai et al. A Fast Multicast-based Approach to Inferring Loss Performance. Journal of Communication and Computer, 2006.3

[11] Yongjun Li, Wandong Cai et al. Loss Temporal Dependency Tomography in Wireless Sensor Network. 2007 IEEE International Conference on Wireless Communications, Networking and Mobile Computing, p2352-2355, 2007.9, Shanghai

[12] Zhao tao, Cai wandong, Li Yongjun, A Method for Link Loss Inference Based on End-to-End Measurement, Journal of northwestern polytechnical university, 26(02): 158-161, 2008

[13] TIAN Guang-li,CAI Wan-dong,YAO Ye,ZHAO Zuo, Research on Topology Duration of Ad Hoc Networks and Limit Time of End-to-end Measurement, Journal of System Simulation,Vol 16, 2009.

[14] Guangli Tian , Wandong Cai et al. Routing Topology Identification Based on End-to-end measurements. 2008 IEEE International Conference on Information and Automation (ICIA 2008), p 1595-1598, 2008.6, Zhang JiaJie

[15] Wei Wang, Wandong Cai et al. The factor graph approach for inferring link loss in MANET. 2008 International Conference on Internet Computing in Science and Engineering (ICICSE 2008), 2008.1, Harbin

[16] Duan Qi, Cai Wanddong. Topology Identification Based on Multiple Source Network Tomography. IIS'2009, p125-128, 2009

[17] Duffield Nicholas G., orowitz Joseph, Lo Presti Francesco, et al.. Explicit loss inference in multicast tomography. IEEE Transactions on Information Theory, August, 2006, 52(8): 3852-3855.

[18] Nick Duffield. Network Tomography of Binary Network Performance Characteristics. IEEE Transactions on Information Theory, 2006, 52(12): 5373-5388.

[19] Guo Dong, Wang Xiaodong. Bayesian inference of network loss and delay characteristics with

applications to TCP performance prediction. IEEE Transactions on Signal Processing, 2003, 51(8): 2205-2218.

[20] Arya Vijay, Duffield N.G., Veitch Darry. Multicast inference of temporal loss characteristics. Performance Evaluation, 2007,vol (64): 9-12.

[21] Yolanda Tsang, Coates M., Nowak R.D. Network delay tomography, IEEE Transactions on Signal Processing, Aug. 2003, 51(8): 2125 – 2136

[22] Bestavros Azer, Byers John. Harfoush Khaled. Inference and labeling of metric-induced network topologies. IEEE Transactions on Parallel and Distributed Systems, November, 2005, 16(11): 1053-1065.

[23] Tian Hui, Shen Hong. Multicast-based inference for topology and network-internal loss performance from end-to-end measurements. Computer Communications, 2006, 29(11): 1936-1947.

[24] N.G. Duffield, F. Lo Presti, V. Paxson, et al.. Inferring link loss using striped unicast probes.IEEE INFOCOM 2001, Anchorage, Alaska, 2001, vol 2: 915-923.

[25] Liang, B. Yu. Maximum pseudo likelihood estimation in network tomography. IEEE Transactions on Signal Processing, 2003, 51(8): 2043-2053.

[26] Wang Wei, Cai Wandong, Wang Beizhan, et al.. Mobile Ad hoc Network Delay Tomography. Proceedings of 2007 IEEE International Workshop on Anti-counterfeiting Security, Identification. Xiamen University, Xiamen, Chian, April, 2007, pp: 365-370.

[27] Jianzhong Zhang. Origin-Destination Network Tomography with Bayesian Inversion Approach. Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, 2006, pp: 38-44

[28] Yongjun Li Wandong Cai Wei Wang Guangli Tian, Lossy Node Identification in Wireless Sensor Network, Proceeding of Fifth IEEE International Symposium on Network Computing and Applications, Cambridge, Massachusetts, USA, NCA 2006, v 2006, p 255-258, July 2006

[29] Yongjun Li Wandong Cai Guangli Tian Wei Wang, Loss Tomography in Wireless Sensor Network Using Gibbs Sampling, Proceeding of EWSN 2007, LNCS 4373, p 150–162, Delft, Netherlands, Jan 29-31 2007.

[30] Tao Zhao, Wandong Cai, Yongjun Li. Sensor network loss inference using end-to-end measurement. Journal of Computational Information Systems, 3(6):2383-2388, 2007

[31] Yongjun LI Wandong CAI Wenli JI Tao Zhao, A Fast Inference of Loss Rate in Wireless Sensor Network, Journal of Computational Information Systems, 3(1):125-132, 2007

[32] Tao Zhao, Wandong Cai, Yongjun Li. Bottom-up inference of loss rate in sensor network, Journal of Computational Information Systems, 4(4):1429-1434, 2008

[33] Yongjun LI Wandong CAI Wenli JI Tao Zhao, Wireless Sensor Network Topology Identification based on Data Aggregation, Journal of Computational Information Systems, 3(6):2359-2365, 2007

[34] Tao Zhao, Wandong Cai, Yongjun Li. MPIDA: A sensor network topology inference algorithm, International Conference on Computational Intelligence and Security, Harbin, Heilongjiang, China, 451-455, Dec, 2007

[35] Tao Zhao, Wandong Cai, Yongjun Li. Using End-to-End Data to Infer Sensor Network Topology, The 7th IEEE International Symposium on Signal Processing and Information Technology, Cairo, Egypt, 99-103, Dec, 2007

[36] Yongjun Li, Wandong Cai et al. Loss Cumulate Generating Function Inference in Wireless Sensor Network. 2006 IEEE International Conference on Wireless Communications, Networking and Mobile Computing, p1-4, 2006.9, Wuhan

[37] Meng-fu Shih, Alfred Hero, "Unicast inference of network link delay distributions from edge measurements," Proceedings of 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01), p3421-3424, 2001

[38] "The chernoff bound explained", available: http://people.deas.harvard.edu/~ho/ DEDS/OO/Idea/ Slide03.html

[39] B. Walsh. Markov Chain Monte Carlo and Gibbs Sampling, http://nitro.biosci.arizona.edu/ workshops/Aarhus2006/ pdfs/ Gibbs.pdf

[40] Yao Ye, Cai Wanddong, Koukam Abder, Hilaire Vincent. A Multi-hierarchical Group Mobility Model for Tactical Mobile Wireless Networks. Journal of Computational Information Systems, 5(1):275-282, February 2009

[41] N Sadagopan, F Bai, B Krishnamachari, A. Helmy. Paths：Analysis of path duration statistics and their impact on reactive MANET Routing Protocols. Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing(MobiHoc'03), Annapolis, p245~256,2003.

[42] Guangli Tian , Wandong Cai et al. Topology Variety Model for Mobile Ad Hoc Networks. Mobilware'08, 2008.2, Innsbruck, Austria.

[43] Wang Wei, Cai Wandong, Wang Beizhan, et al., Research on a Mobility Model Based on Circle Movement in Ad Hoc Network. Journal of Computer Research and Development , 44(6), p 932-938, June 2007

[44] Yao Ye, Cai Wandong, Tian Guangli, A Method of Snatching Ad Hoc network Link Topology Snapshot, Science paper Online of China, http://www.paper.edu.cn/index.php/default /releasepaper/content/200903-3, 2009.03

[45] Yao Ye, Ad Hoc Networks Measurement Model and Methods Based on Network Tomography[Ph.D], Northwestern Polytechnical University, 2010.

[46] Information Sciences Institute of University of Southern California. "The Network Simulator 2," available:www.isi.edu/nsnam/ns2.

[47] Yao Ye, Cai Wanddong, Hilaire Vincent, Koukam Abder. Research on Physical Topology Steady Degree of RWP Mobility Model on Ad Hoc Network. Journal of Computational Information Systems, 5(4): p1203-1211, August 2009

[48] Yao Ye, Cai Wanddong et al. Research on the Link Topology Lifetime Of Mobility Model in Ad Hoc Network. 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing(NSWCTC' 2009), v 1, p103-107, 2009.4, Wuhan

[49] Andreas Johnsson, Mats Björkman, Bob Melander. A Study of Dispersion-based Measurement Methods in IEEE 802.11 Ad-hoc Networks. The International Conference on Communication in Computing, Las Vegas, June, pp.227-230, 2004

[50] L. J. Chen, T. Sun, G. Yang, M. Y. Sanadidi and M. Gerla. Ad Hoc Probe: Path Capacity Probing in Wireless Ad Hoc Networks. The first IEEE International Conference on Wireless Internet (WICON 2005), Budapest, Hungary, 2005.

[51] Cheikh Sarr and Isabelle Guerin Lassous. Estimating Average End-to-End Delays in IEEE 802.11 Multihop Wireless Networks. Technical Report 1, INRIA, July 2007.

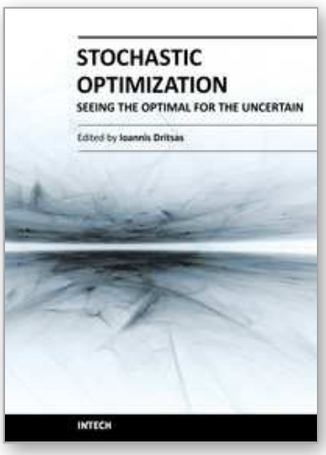
[52] Yao Ye, Cai Wanddong. Ad Hoc Network Measurement Based on Network Tomography: Theory, Technique, and Application. Journal of Networks, 5(6): 666-674, June 2010

[53] Yao Ye, Cai Wanddong. Interior Link Delay Reference of Ad Hoc Networks Based on End-to-End Measurement: Linear Analysis Model of Delay. Journal of Networks, 4(4), p 244-253, June 2009

[54] Yao Ye, Cai wandong, Tian guangli , A Link Loss Rate Inference Method for Ad Hoc Networks Based on End-to-End Measurement, Journal of northwestern polytechnical university, 28(1), p82-86, February 2010

[55] Guangli Tian, Cai Wanddong. Routing Topology Identification Based on Multiple Sources End-to-end Measurements. APWCS'2010, p322-325, 2010

[56] Wei Wang, Wandong Cai et al. Mobile ad hoc network delay tomography. 2007 IEEE International Workshop on Anti-counterfeiting, Security and Identification (ASID), p365-370 2007.4, Xiamen



Stochastic Optimization - Seeing the Optimal for the Uncertain
Edited by Dr. Ioannis Dritsas

ISBN 978-953-307-829-8
Hard cover, 476 pages
Publisher InTech
Published online 28, February, 2011
Published in print edition February, 2011

Stochastic Optimization Algorithms have become essential tools in solving a wide range of difficult and critical optimization problems. Such methods are able to find the optimum solution of a problem with uncertain elements or to algorithmically incorporate uncertainty to solve a deterministic problem. They even succeed in â€œfighting uncertainty with uncertaintyâ€. This book discusses theoretical aspects of many such algorithms and covers their application in various scientific fields.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Cai Wandong, Yao Ye and Li Yongjun (2011). Research on Network Tomography Measurement Technique, Stochastic Optimization - Seeing the Optimal for the Uncertain, Dr. Ioannis Dritsas (Ed.), ISBN: 978-953-307-829-8, InTech, Available from: http://www.intechopen.com/books/stochastic-optimization-seeing-the-optimal-for-the-uncertain/research-on-network-tomography-measurement-technique

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen