

基于贝叶斯模型的 IP 网拥塞链路诊断算法

杜艳明^{1*}, 韩冰¹, 肖建华²

(1. 浙江工业职业技术学院 计算机学院, 浙江 绍兴 312000; 2. 南开大学 现代物流研究中心, 天津 300071)

(*通信作者电子邮箱 allanduym@163.com)

摘要:通过端到端路径的性能判断 IP 网络运行状态的方法可以以较小的代价诊断网络故障,但目前已有的端到端技术仍然存在两个主要问题:1)端到端的探测数量不足以准确定位每条链路的拥塞状态;2)随着网络规模的扩大,诊断所消耗的计算时间过长,无法达到实时性的要求。为解决以上问题,提出一种基于贝叶斯模型的高效拥塞链路诊断算法。所提算法将拥塞定位问题建立成贝叶斯模型,将模型进行二次化简,并限制了同时发生拥塞的链路个数,从而在保证一定准确度的基础上大大降低了推理的计算复杂度。通过仿真与 Planetlab 实验将所提算法与 Clink 算法进行了对比,实验结果证明,所提算法具有更高的诊断准确度和更短的诊断时间。

关键词:IP 网;故障诊断;端到端探测;贝叶斯网;拥塞链路定位

中图分类号:TP393.06 **文献标志码:**A

Congested link diagnosis algorithm based on Bayesian model in IP network

DU Yan-ming^{1*}, HAN Bing¹, XIAO Jian-hua²

(1. School of Computer, Zhejiang Industry Polytechnic College, Shaoxing Zhejiang 312000, China;

2. Research Center of Logistics, Nankai University, Tianjin 300071, China)

Abstract: In IP network, tomography method can perform fault diagnosis by analyzing the end-to-end properties with low costs. However, most existing tomography based techniques have the following problems: 1) the end-to-end detected number is not sufficient to determine the state of each link; 2) as the scale of the network goes up, the diagnosis time may become unacceptable. To address these problems, a new congested link diagnosis algorithm based on Bayesian model was proposed in this paper. This method firstly modeled the problem as a Bayesian network, and then simplified the network by two steps and limited the number of multiple congested links. Therefore, the proposed method could greatly reduce the computational complexity and guarantee the diagnostic accuracy. Compared with the existing diagnosis algorithm which is called Clink, the proposed algorithm has higher diagnostic accuracy and shorter diagnosis time.

Key words: IP network; fault diagnosis; end-to-end probe; Bayesian network; congestion link location

0 引言

IP 网的故障诊断技术对网络用户和网络服务提供商来说都是至关重要的。而随着 IP 网络规模日益扩大和网络结构的复杂多样,如何及时准确地发现并诊断网络故障成为一个富有挑战性的难题,也成为众多研究者的研究热点。

在 IP 网中,最常用的链路拥塞故障诊断方法为端到端测量法。即网络中的源终端节点向其他目的终端节点发送探针包,测量得到一组端到端路径的性能。通过分析路径与其经过链路之间的关联关系,定位存在拥塞故障的链路。然而,目前的端到端拥塞诊断技术均面临一个难题:在大多数的 IP 网中,能用到的所有端到端测量路径的个数不足以准确定位每一条发生拥塞的链路。

为了解决这个问题,目前已有的利用端到端路径测量技术定位拥塞链路的方法可以分为两大类:1)利于多播路由发送探针包,从而得到足够数量的端到端路径个数^[1-3];2)假设网络中每条链路拥塞概率分布是已知的,并且认为发生拥塞链路的个数尽可能少,从而确定一组最有可能拥塞链路^[4-8]。以上两类方法均存在一定的局限性:支持多播的路由目前还没有得到大量普及,而对链路拥塞概率分布的假设会导致诊断准确度的

下降。并且随着网络规模的扩大,已有的海量端到端数据的分析方法越来越无法满足实时性的要求。

本文针对目前基于端到端的拥塞链路诊断方法存在的不足,提出了一种基于贝叶斯模型的高效拥塞链路诊断算法。本文算法主要分为3步:1)将被管系统建立成贝叶斯模型,并在模型中设定探针与链路之间的关联关系;2)通过代数分析,精确确定部分链路丢包率,并从原模型中移除,此步骤大大降低了模型推理复杂度;3)通过少量额外端到端测量推理出贝叶斯模型中的先验概率,并代入贝叶斯公式求得剩余链路的一组最有可能状态。最后,本文通过实验验证了该算法的有效性和可行性。

本文主要介绍以下工作:

1)将拥塞链路定位问题建立为贝叶斯模型中的推理问题,并给出端到端路径丢包率与链路丢包率之间的概率关系式。将路径测量数据以及链路状态值分别建模为贝叶斯网中的证据节点和隐藏节点。

2)根据路径与链路状态之间的代数关系,可精确确定部分链路的拥塞状态。从模型中移除这部分链路可将原贝叶斯模型进行两次化简,使原模型规模减小 90% 左右,从而大大降低了推理的复杂度。

收稿日期:2011-07-06;修回日期:2011-08-22。 基金项目:国家自然科学基金资助项目(60903105)。

作者简介:杜艳明(1972-),男,湖北黄冈人,讲师,硕士,主要研究方向:计算机网络通信、图像处理;韩冰(1984-),女,吉林公主岭人,讲师,硕士,主要研究方向:Web 挖掘;肖建华(1979-),男,重庆人,讲师,博士,主要研究方向:智能计算、物流系统优化。

3)在简化后的贝叶斯模型基础上提出了一种高效的贝叶斯推理算法,该算法通过限制故障个数,在保证一定准确度的基础上进一步降低了贝叶斯推理的复杂度,可在多项式时间内完成贝叶斯推理过程。

4)实现了已有的链路拥塞诊断算法(Clink 算法)及本文提出的算法,通过仿真及 Planetlab 的实验验证,无论在计算时间还是诊断准确度上,本文算法的表现均优于 Clink 算法。

1 问题描述及建模

1.1 建立被管网络的贝叶斯模型

贝叶斯网络模型是一种表示因果关系的有向无环图。图中包含两个元素: $G = (V, E)$,其中: V 表示节点, E 表示连接节点之间的有向边。在贝叶斯网中,每个节点存储了一个条件概率表,当该节点为根节点时,此条件概率表为节点的先验概率分布。根据图中的因果关系以及已知的条件概率和先验概率,可由已知节点的状态(也称为证据节点)推理未知节点(也称为隐藏节点)的状态。贝叶斯网中所有节点的联合概率^[9]可由式(1)表示:

$$P(X_1, X_2, \dots, X_N) = \prod_{i=1}^N P(X_i | pa(X_i)) \quad (1)$$

其中 $pa(X_i)$ 为节点 X_i 在图中的父节点。

图1为具有5个终端的IP网,计算机图标为可发送和接收探针包的终端节点。表1列举了图1中的6个端到端路径。

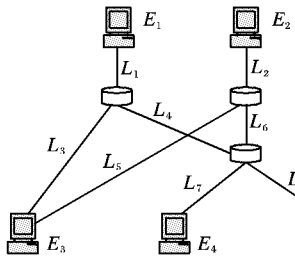


图1 5个终端的IP网络

表1 图1中的端到端探测

探针	路径	探针	路径
T_1	$L_1 \rightarrow L_2$	T_4	$L_2 \rightarrow L_5$
T_2	$L_1 \rightarrow L_4 \rightarrow L_7$	T_5	$L_2 \rightarrow L_6 \rightarrow L_7$
T_3	$L_1 \rightarrow L_4 \rightarrow L_8$	T_6	$L_2 \rightarrow L_6 \rightarrow L_8$

图2为根据表1所建立的贝叶斯网模型。其中上层节点 $X = (X_1, X_2, \dots, X_8)$ 表示图1中链路 $[L_1, L_2, \dots, L_8]$ 的随机变量;下层节点 $T = (T_1, T_2, \dots, T_6)$ 表示表1中端到端探针的随机变量。链路节点与探针节点之间的有向边表示该探针经过此链路。每个链路 X_i 有0和1两个取值:当 $X_i = 0$ 时表示链路正常;当 $X_i = 1$ 时表示链路处于拥塞状态。每个探针节点 T_i 同样有0和1两个取值:当 $T_i = 0$ 时表示该探针经过的路径正常;否则则说明该条路径拥塞。一般情况下,当某探针路径正常时,该路径上的所有链路均正常;若某条路径拥塞,则该路径至少经过一条拥塞的链路。因此,路径节点与链路节点之间存在如下的概率关系:

$$P(T_i = 0 | pa(T_i)) = \{0, 0, \dots, 0\} = 1 \quad (2)$$

$$P(T_i = 1 | \exists X_i = 1 \cap X_i \in pa(T_i)) = 1 \quad (3)$$

贝叶斯推理也就是根据模型中证据节点的状态来推理隐藏节点最有可能的状态。在本模型中,已知端到端探测 $T = (T_1, T_2, \dots, T_6)$ 的状态值后,求 $X = (X_1, X_2, \dots, X_8)$ 最有可能的一组取值,可用式(4)表示:

$$\max P(X_1, X_2, \dots, X_8 | T_1, T_2, \dots, T_6) = \max \frac{P(X_1, X_2, \dots, X_8, T_1, T_2, \dots, T_6)}{P(T_1, T_2, \dots, T_6)} = \max (X_1, X_2, \dots, X_8, T_1, T_2, \dots, T_6) \quad (4)$$

根据贝叶斯公式(1),得式(5):

$$P(X_1, X_2, \dots, X_8, T_1, T_2, \dots, T_6) = P(T_1 | pa(T_1)) P(T_2 | pa(T_2)) \dots P(T_6 | pa(T_6)) P(X_1) P(X_2) \dots P(X_8) \quad (5)$$

其中: $pa(T_i) \in X$ 为贝叶斯网中 T_i 的父节点; $P(X_j)$ 为链路 X_j 先验概率; $P(T_i | pa(T_i))$ 可通过式(2)和(3)计算出,而目前并没有关于 $P(X_j)$ 的信息,不能直接得到链路的先验概率。本文将在2.2.1节中介绍如何通过少量额外探测得到每条链路的先验概率,进而得到式(5)的值。

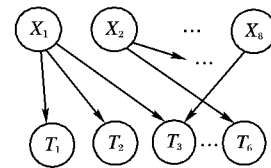


图2 建立图1的贝叶斯网

1.2 代数模型

令 $\varphi(X_i)$ 表示链路 X_i 的通过率,则其链路丢包率为 $1 - \varphi(X_i)$; $\varphi(T_i)$ 为路径 T_i 的路径通过率,其路径丢包率为 $1 - \varphi(T_i)$ 。假设当前IP网的路由矩阵为 R , R 中的每一行表示一条端到端路径,每一列代表一条链路。若路径 T_i 经过链路 X_j , 则元素 $R_{ij} = 1$; 否则 $R_{ij} = 0$ 。例如表1的路由矩阵 R 为:

$$R = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \quad (6)$$

路径通过率 $\varphi(T_i)$ 与链路通过率 $\varphi(X_j)$ 之间存在如下关系:

$$\varphi(T_i) = \prod_{j=1}^M [\varphi(X_j)]^{R_{ij}} \quad (7)$$

对式(4)两边取对数函数可得:

$$\lg \varphi(T_i) = \sum_{j=1}^M R_{ij} \lg \varphi(X_j) \quad (8)$$

令 $A = [\varphi(T_1), \varphi(T_2), \dots, \varphi(T_M)]'$, $B = [\varphi(X_1), \varphi(X_2), \dots, \varphi(X_N)]'$, 则求解式(8)的问题可转化为求解线性代数式(9):

$$RB = A \quad (9)$$

由式(9)可得出如下结论:若已知端到端的路径丢包率,则每条链路的丢包率可以被唯一确定的充要条件为路由矩阵 R 满秩。

但是,在大多数IP网中,路由矩阵 R 并不满足这一条件,因此无法得出唯一的一组链路丢包率值,从而无法准确定位所有拥塞链路。第2章将介绍本文提出的拥塞链路定位算法,该算法可利用现有的端到端探测以及少量额外探测,通过贝叶斯算法推理出每条链路的拥塞状态。

2 拥塞定位算法

2.1 确定大部分链路的精确状态

根据式(4)直接进行贝叶斯推理的计算复杂度极高,因

为要遍历每条链路的每一个状态。本文发现,在原模型中,大部分链路的拥塞状态是可以通过分析路由矩阵 \mathbf{R} 得到。在去掉已确定的链路后,原模型的规模可减小90%左右。

以图1为例,假设观测到的路径拥塞状态为 $\mathbf{T} = (1, 1, 0, 1, 1, 1)$,可根据以下两个步骤将模型进行两次化简。

1) 对任意端到端路径,路径通过率为其所经过链路通过率的乘积。因此,当某端到端路径通过率为1时,说明该路径上不存在拥塞链路。从式(9)中移除不丢包的路径及链路,得到简化后的式子如下:

$$\mathbf{R}_1 \mathbf{B}_1 = \mathbf{A}_1 \quad (10)$$

根据观测到的端到端测量值,式(6)中可去掉第3行以及第1,4,8列,原矩阵可化简为

$$\mathbf{R}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (11)$$

对应的贝叶斯模型可化简为如图3所示。

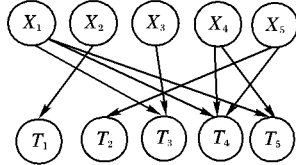


图3 一次简化后的贝叶斯模型

2) 将式(11)进行初等行变换,可得到上三角矩阵

$$\mathbf{R}_1' = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (12)$$

通过观察式(12)可发现,矩阵 \mathbf{R}_1' 中部分行仅有一列元素非零(第2,4行),因此该非零列(第2,5列)对应链路的丢包率可被唯一确定。假设 \mathbf{R}_1' 中第 j 行仅有第 j 列非零,则可求出该链路的状态即为经过初等行变换后的对应路径的状态。此时,经过行变换后,第2,5条链路对应的路径状态分别为 $\{1, 1\}$ 。

在进行了以上两个步骤后,可精确求得部分链路的拥塞状态。从原式中去掉这些链路和路径,可将式(10)简化为

$$\tilde{\mathbf{R}} \tilde{\mathbf{B}} = \tilde{\mathbf{A}} \quad (13)$$

例如:式(11)可化简为

$$\tilde{\mathbf{R}} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad (14)$$

同样可将原贝叶斯模型两次化简为如图4所示。

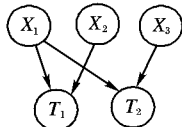


图4 两次简化后的贝叶斯模型

在本例中仅假设 T_3 为正常路径,而在一般网络中,正常路径往往多于非正常路径,因而可得到规模更小的 $\tilde{\mathbf{R}}$ 矩阵。

2.2 确定剩余链路拥塞状态

第1章已经提到,要进行贝叶斯推理需要得到上层隐藏节点的先验概率。本节首先利用少量额外端到端探测,获得剩余链路的先验概率,然后代入贝叶斯模型中进行推理,最终得到最有可能的拥塞故障集合。

2.2.1 学习剩余链路拥塞概率

假设两次化简后,贝叶斯模型中的剩余链路和端到端探测的个数分别为 N_X 和 N_T 。对每条路径 T_i 的状态测量 m 次,即 $1 \leq i \leq N_T, T_i = \{T_i^1, T_i^2, \dots, T_i^m\}$ 。对每个 T_i 取均值,用路径 T_i 上所经过的链路 X_j 的状态表示如下:

$$\begin{aligned} E(T_i) &= E\left(\bigcup_{j=1}^{N_X} X_j \bar{R}_{ij}\right) = \\ P\left(\bigcup_{j=1}^{N_X} X_j \bar{R}_{ij} = 1\right) &= \\ 1 - P(X_j \bar{R}_{ij} = 0, 1 \leq j \leq N_X) &= \\ 1 - \prod_{j=1}^{N_X} (1 - p_j)^{\bar{R}_{ij}} \end{aligned} \quad (15)$$

其中 p_j 为链路 X_j 的拥塞概率。对式(15)两边取对数,可得出

$$\lg(1 - E(T_i)) = \sum_{j=1}^{N_X} \bar{R}_{ij} \lg(1 - p_j); 1 \leq i \leq N_T \quad (16)$$

从式(16)可看出,若想得到剩余链路的拥塞概率,则要求矩阵 $\tilde{\mathbf{R}}$ 满秩。然而,大多数情况下 $\tilde{\mathbf{R}}$ 并不满足要求。因此,本文对式(16)进行如下扩展。

用 T_{ik} 表示路径 T_i 与路径 T_k 相合并后的路径拥塞状态。当 $T_{ik} = 1$ 时, T_i 与 T_k 至少一条拥塞; $T_{ik} = 0$ 时, T_i 与 T_k 均不拥塞。因此可以得到

$$\begin{aligned} E(T_{ik}) &= P\left(\bigcup_{j=1}^{N_X} X_j \bar{R}_{ij} = 1\right) \cup P\left(\bigcup_{j=1}^{N_X} X_j \bar{R}_{kj} = 1\right) = \\ 1 - \prod_{j=1}^{N_X} (1 - p_j)^{\bar{R}_{ij} \cup \bar{R}_{kj}} \end{aligned} \quad (17)$$

对式(17)两边取对数可得:

$$\lg(1 - E(T_{ik})) = \sum_{j=1}^{N_X} \{\bar{R}_{ij} \cup \bar{R}_{kj}\} \lg(1 - p_j) \quad (18)$$

对所有 $1 \leq i < k \leq N_T$ 求解式(18),可得到额外的 $N_T(N_T - 1)/2$ 个式子。结合式(16),总共可得到 $N_T(N_T + 1)/2$ 个线性方程,而 $N_T(N_T + 1)/2 \gg N_X$ 。实验证明,在大多数(90%以上)的IP网中,式(16)与(18)的结合,可以得出链路拥塞概率 p_j 的唯一组解。

$$\text{例如,对于式(14),扩展后的矩阵为} \tilde{\mathbf{R}}_{\text{expand}} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix},$$

显然, $\tilde{\mathbf{R}}_{\text{expand}}$ 为满秩矩阵。

2.2.2 拥塞链路的贝叶斯推理方法

要求出每条链路 X_i 的最有可能状态,也就是求:

$$\begin{aligned} \arg \max_{1 \leq i \leq N_X} P(X_1, X_2, \dots, X_{N_X} | T_1, T_2, \dots, T_{N_T}) &= \\ \arg \max_{1 \leq i \leq N_X} P(X_1, X_2, \dots, X_{N_X}, T_1, T_2, \dots, T_{N_T}) \end{aligned} \quad (19)$$

由贝叶斯公式(1)可得:

$$\begin{aligned} P(X_1, X_2, \dots, X_{N_X}, T_1, T_2, \dots, T_{N_T}) &= \\ \prod_{i=1}^{N_T} P(T_i | pa(T_i)) \prod_{j=1}^{N_X} P(X_j) &= \\ \prod_{i=1}^{N_T} P(T_i | pa(T_i)) \prod_{j=1}^{N_X} p_j \end{aligned} \quad (20)$$

尽管在2.1节中已将模型大大简化,但式(20)的计算仍需要遍历所有剩余 N_X 条链路状态,复杂度为 $O(2^{N_X})$ 。

为了进一步降低复杂度,应把不必要的拥塞状态忽略掉。本文限制拥塞故障同时发生个数为 k ,此时式(20)要计算的状态个数为 $\sum_{i=0}^k \binom{N_X}{i}$,计算复杂度可降低为多项式级: $O(N_X^{k+1})$ 。下面分析由此带来的诊断误差。

假设 p_{\max} 为链路发生拥塞故障的最大概率, 即 $p_{\max} = \max_{1 \leq i \leq N_X} p_i$, 则同时发生拥塞的链路个数为 k 以上的概率为:

$$P(|F| > k) = 1 - \sum_{|F|=0}^k \binom{N_X}{|F|} p_{\max}^{|F|} (1 - p_{\max})^{N_X - |F|} \quad (21)$$

其中 $|F|$ 为同时发生故障的个数。当剩余链路个数 $N_X = 50$, $p_{\max} = 0.1$, 同时发生故障的个数不超过 $k = 10$ 时, 则同时发生 10 个以上故障的概率为:

$$P(|F| > 10) = 1 - \sum_{|F|=0}^{10} \binom{50}{|F|} 0.1^{|F|} (1 - 0.1)^{50 - |F|} \approx 0.0094 \quad (22)$$

由式(22)可知限制故障个数而导致的误差是非常小的, 可以忽略不计。

由式(2)、(3)和(20), 可求得符合条件的所有拥塞状况的概率值, 因而也可求得使得式(19)达到最大的一组 X_1, X_2, \dots, X_{N_X} 状态, 此状态即为剩余链路的最有可能状态, 其中状态值为 1 的 X_i 变量为拥塞链路。

2.3 拥塞链路推理算法

本文所提出的拥塞链路推理算法的伪代码如下所示。

输入: 路由矩阵 R ; 路径通过率向量 $D = [D_1, D_2, \dots, D_M]$; 额外探测次数 m ; 故障发生个数门限 k ;

输出: 链路拥塞状态向量 $S = [S_1, S_2, \dots, S_N]$ 。

```

 $S_{\text{exact}} = \text{null};$  //存储精确计算出的链路状态
 $S_{\text{mostLike}} = \text{null};$  //存储贝叶斯推理出的链路状态
 $S_{\text{good}} = \text{findGoodLinks}(R, D);$  //找出部分不拥塞链路
 $[R_1, D_1] = \text{removeGoodLinks}(R, D);$  //对模型进行一次简化
 $R_1' = \text{upperTriangular}(R_1);$  //将  $R_1$  化为上三角矩阵
 $S_{\text{determined}} = \text{findSingleLink}(R_1');$  //从  $R_1'$  中找出可确定的链路
 $S_{\text{exact}} = [S_{\text{good}}, S_{\text{determined}}];$ 
//可精确计算的链路为  $S_{\text{good}}$  与  $S_{\text{determined}}$  的并集
 $[\tilde{R}, \tilde{D}] = \text{removeExactLinks}(R, D, S_{\text{exact}});$  //从  $R$  中移除  $S_{\text{exact}}$ 
 $\tilde{D}_{\text{snapshots}} = \text{getSnapshots}(\tilde{D}, m);$  //取  $m$  次额外探测
 $p = \text{learnProbabilities}(\tilde{R}, \tilde{D}_{\text{snapshots}});$ 
//学习剩余链路先验概率
 $S_{\text{mostLike}} = \text{BayesianRule}(\tilde{R}, p, \tilde{D}_{\text{snapshots}}, k);$ 
//通过贝叶斯推理出剩余链路状态
 $S = [S_{\text{exact}}, S_{\text{mostLike}}]$ 

```

2.4 算法性能分析

假设系统共包含 N 条链路, M 条端到端路径, 其中 M_0 条路径不丢包, M_0 条路径中共覆盖了 N_0 条链路。假设去掉不拥塞的 N_0 链路后, 有 N_1 条链路的状态可直接被确定。在剩余链路中同时发生拥塞的链路个数不超过 k 个。

2.4.1 时间复杂度分析

对于已有的基于贝叶斯推理的拥塞定位算法, 若要确定最有可能的一组链路状态, 需要遍历 $N - N_0$ 条链路状态, 从而所需的计算复杂度为 $O(2^{N-N_0})$ 。

对于本文提出的算法, 在提取出可被唯一确定的链路后, 对剩余链路限制了同时发生拥塞的个数不超过 k , 则需要计算的状态个数为 $\sum_{i=0}^k \binom{N - N_0 - N_1}{i}$, 因而计算复杂度可降低为 $O((N - N_0 - N_1)^{k+1})$ 。

2.4.2 诊断准确度分析

在诊断准确度上, 已有的基于贝叶斯推理的方法在一次

简化模型基础上进行推理, 因此诊断误差来自于一次简化后的诊断模型(见图 3)。

而本文中提出的算法在推理之前已精确确定了部分链路的拥塞状态, 因而一部分误差仅来自于二次简化后的模型(见图 4); 另一部分误差来自于限制了同时发生拥塞的个数 k 。而 2.2 节式(22)已证明, 限制同时发生拥塞个数所产生的误差微乎其微。

3 实验结果及分析

本文利用 Matlab 实现了文献[5]提出的 Clink 算法以及本文提出的算法, 分别在仿真环境与 Planetlab 搭建的真实网络环境中验证本算法的优越性。

3.1 仿真

3.1.1 实验环境搭建

考虑网络拓扑的形状在一定程度上会影响实验效果, 而当前研究发现, 大多数网络拓扑的形状均服从幂率规则^[10]。因此, 本文利用 Inet 拓扑产生器^[11]产生了规模分别为 100, 200, 300, 400 和 500 的服从幂率规则的网络拓扑。网络拓扑中的每条链路以概率 0.1 发生拥塞。本文利用 LLRD1 丢包率模型^[12]对每条链路分配丢包率: 当某条链路为拥塞链路时, 其丢包率服从 $[0.05, 1]$ 上的均匀分布; 当为正常链路时, 其丢包率服从 $[0, 0.01]$ 上的均匀分布。当为每条链路赋予了一个丢包率后, 链路的丢包服从 Gilbert 过程。即链路的状态以其丢包率的概率在“好”与“坏”之间波动。当处于“好”状态时, 不丢包; 当处于“坏”状态时, 全丢包。网络中的所有叶子节点作为终端节点, 端到端路径的选择服从最短路径的原则。取丢包率极限值 δ 为 0.05, 当计算得出的丢包率大于该值则认为此链路拥塞; 否则认为此链路正常。对每条需要重复测量多次的路径测量 50 次, 限制同时发生拥塞的链路个数不超过 10。

3.1.2 仿真实验结果

本节用两个标准来衡量算法的诊断准确度: 覆盖准确率 (Detection Rate, DR) 和误诊率 (False Positive Rate, FPR)。

$$DR = |F \cap X| / |F|, FPR = |X \setminus F| / |X|$$

其中: F 为实际发生拥塞的链路, X 为算法诊断出的发生拥塞的链路。

图 5、6 显示了在不同的网络规模下, Clink 算法与本文算法的 DR 与 FPR 值。从图中可以看出, 在网络规模较小的情况下, 两种算法表现相差不大; 而当网络规模增加时, 本文算法的 DR 与 FPR 明显优于 Clink 算法。

图 7 显示了在不同网络规模下, Clink 算法与本文算法执行诊断时间的比较。由于本文算法在第一阶段大大简化了计算式(6), 同时限制了同时发生拥塞的链路个数, 因此也节省了时间开销。

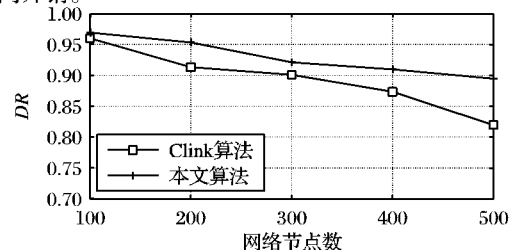


图 5 不同网络规模下两种算法的 DR 值比较

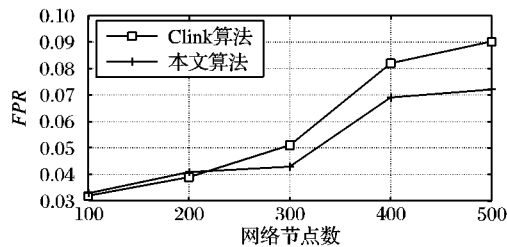


图6 不同网络规模下两种算法的FPR值比较

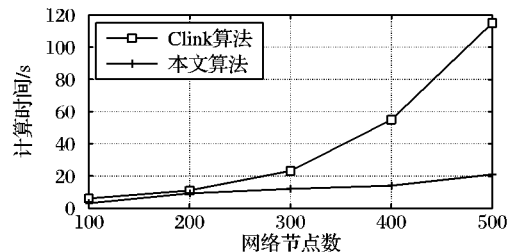


图7 不同网络规模下两种算法的计算时间比较

3.2 PlanetLab 实验

3.2.1 实验环境搭建

实验部署在分布于全球的115个Planetlab节点上。其中亚洲的节点个数为78个,占实验节点的大部分;其次为北美及欧洲节点。每个终端节点向其余所有节点发送探测包,总共有13110条路径。首先用tracert命令测量每条路径的路由,形成路由矩阵 R 。但是,出于某些安全原因,路径中的一些路由节点不响应tracert命令的请求,从而无法确定该路由节点IP。因此获得的路由矩阵不可避免地存在一定的噪声。在获得路由矩阵后,通过向每条路径发送UDP包来测量路径的丢包率。UDP包的大小为40B,其中20B的IP头,8B的UDP头和12B的包序列及发送时间,每个包发送的时间间隔为5ms,每条路径发送500个UDP包。在接收端计算路径的通过率,计算方法为:发送包的个数/接收到包的个数。包的传输速率控制在100KBps以内,每个终端节点每一轮随机地选择另一个终端节点发送探测包,以防止人为地产生拥塞。

Planetlab实验开始于2011年4月27日,持续大约20h,收集到0.7GB数据。

3.2.2 Planetlab 实验结果

由于Planetlab实验中无法得到每条链路真实丢包率,为了验证实验结果,本文采用文献[4]中提出的验证方法。即将所有路径随机地分为大小相等的两个集合:其中一个路径集合为推理集合;另一个集合为验证集合。若验证集合中的某拥塞路径包含至少一条推理出的拥塞链路,或者不拥塞路径不包含拥塞链路,则称此路径为“一致路径”。

图8显示了在不同个数的Planetlab终端节点的情况下,验证集合中的“一致的路径”占验证集合所有路径的百分比。随着终端节点个数的增加,路径个数也呈指数级增长,因此可用于推理的路径也相应地增加,两算法的诊断准确度均有所提高。而当用于推理的路径增加时,通过2.1节计算得到的精确链路个数也有较大幅度增长,因而较Clink算法,本文算法的准确度提高幅度也较大。

4 结语

为了解决目前基于端到端路径测量技术定位拥塞链路的方法定位准确度不够、诊断时间过长等问题,本文提出了一种基于贝叶斯模型的拥塞链路诊断算法。本文算法首先把拥塞定位问题建立成贝叶斯模型,并利用已有的端到端测量,精确

定位出大部分链路的拥塞状态,然后通过少量额外端到端探测学习剩余链路拥塞概率,最后利用贝叶斯公式求出剩余链路的最有可能拥塞状态。分别在仿真环境与Planetlab实验环境中比较了本文算法和已有的拥塞定位算法(Clink算法)。实验结果证明,本文算法具有更高的诊断准确度和更快的诊断速度。

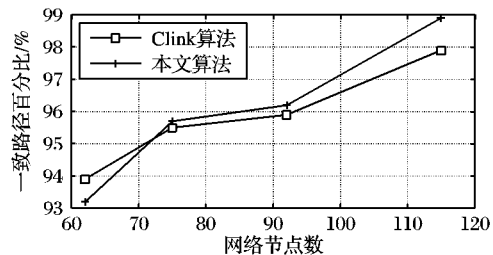


图8 包含不同Planetlab终端节点个数的网络中两算法准确度比较

参考文献:

- [1] CHEN A, CHAO J, BU T. Network tomography: Identifiability and Fourier domain estimation [C]// Proceedings of the 26th IEEE International Conference on Computer Communications. Piscataway: IEEE, 2007: 1875-1883.
- [2] BU T, DUFFIELD N, PRESTI F L, et al. Network tomography on general topologies [C]// Proceedings of the 2002 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems. New York: ACM, 2002: 21-30.
- [3] JOHNS A, BU T, FRIEDMAN T, et al. The use of end-to-end multicast measurements for characterizing internal network behavior [J]. IEEE Communications Magazine, 2000, 38(5): 152-159.
- [4] DUFFIELD N G. Network tomography of binary network performance characteristics [J]. IEEE Transactions on Information Theory, 2006, 52(12): 5373-5388.
- [5] NGUYEN H X, THIRAN P. The boolean solution to the congested IP link location problem: Theory and practice [C]// INFOCOM 2007: The 26th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies. Piscataway: IEEE, 2007: 2117-2125.
- [6] NGUYEN H X, ROUGHAN M. Rigorous statistical analysis of Internet loss measurements [J]. ACM SIGMETRICS Performance Evaluation Review, 2010, 38(1): 361-362.
- [7] GHITA D, NGUYEN H, KURANT M, et al. Netscope: Practical network loss tomography [C]// Proceedings of the 2010 IEEE International Conference on Computer Communications. Piscataway: IEEE, 2010: 1-9.
- [8] NGUYEN H X, THIRAN P. Network loss inference with second order statistics of end-to-end flows [C]// IMC'07: Proceedings of 2007 ACM Internet Measurement Conference. New York: ACM, 2007: 227-240.
- [9] PEARL J. Fusion, propagation, and structuring in belief networks [J]. Artificial Intelligence, 1986, 29(3): 241-288.
- [10] ODINTSOVA N, RISH I. Empirical study of topology effects on diagnosis in computer networks [C]// MASS'07: Proceedings of 2007 IEEE International Conference on Mobile Ad Hoc and Sensor Systems. Washington, DC: IEEE Computer Society, 2007: 1-6.
- [11] WINICK J, JAMIN S. Inet-3.0: Internet topology generator, CSE-TR-456-02 [R]. Ann Arbor: University of Michigan, 2002.
- [12] PADMANABHAN V N, QIU L, WANG H J. Server-based inference of Internet performance [EB/OL]. [2011-04-10]. <http://research.microsoft.com/en-us/um/people/padmanab/papers/msr-tr-2002-39.pdf>.