路由器日志序列模式挖掘*)

庄 军[1] 郭 平[1] 周 杨[1] 周 劲[1] 蔡日旭[2]
(重庆大学计算机学院 重庆 400045)[1] (中国建设银行重庆市分行 重庆 400010)[2]

摘 要 随着网络技术的发展,人们对网络质量的要求也越来越高,作为网络传输中重要环节之一的路由器的工作状态的变化直接影响到网络运行质量。从路由器日志中挖掘出的知识既可用于评价网络质量,又可用于改善网络信息服务。本文分析了路由器日志中一些常见信息,采用序列挖掘方法对日志进行了挖掘,并对挖掘结果进行了解释和分析。

关键词 路由器日志,序列模式,数据挖掘

Mining Sequence Pattern of Router Logs

ZHUANG Jun[1] GUO Ping[1] ZHOU Yang[1] ZHOU Jin[1] CAI Ri-Xu[2]
(College of Computer Science, Chongqing university, Chongqing 400044)[1]
(Branch of Chinese Construction Bank Chongqing, Chongqing 400010)[2]

Abstract With the development of network technology, more higher qualities ofnetwork are required. Routers as a kind of important device for network transmission, the change of its state directly influence the network qualities. The knowledge mined from router logs can not only be used to evaluate the qualities of network, but also to improve services of network. This paper first analyzes some information in router logs, then sequential pattern method is used to mine router logs, and finally the mining results are analyzed, corresponding explanations are also given.

Keywords Router log, Sequence pattern, Data mining

1 引言

路由器是网络信息传输的枢纽,被广泛用于网络建设中,承担着局域网之间及局域网与广域网之间的连接。路由器工作时会通过 syslog 机制在日志主机上形成日志文件,内容包括链路建立(失败)信息、包过滤信息等。网络管理员可以通过日志了解日志事件,进而进行故障定位、排除和网络管理。

然而大多数网络管理人员仅是在网络通讯发生故障时才去查找日志,而且只查找引起故障的那一小部分日志,日志的其余部分被忽略掉了。通过对路由器日志的挖掘,可以获得许多有关网络性能及效率的知识,网络管理员利用这些知识可以通过改善网络布局和结构等措施提高网络安全性和网络通信质量,同时发现和查找影响网络安全的潜在故障点,以提高网络性能。

2 项目背景及相关工作

目前,众多路由器产品中 Cisco 是应用最广泛的。Cisco 设备的日志输出分为 7 个等级:等级 0 表示 emergency(紧急事件)、等级 1 表示 alter(警报)、等级 2 表示 critical(临界状态)、等级 3 表示 error(错误)、等级 4 表示 warming(警告)、等级 5 表示 notification(通知)、等级 6 表示 informational(报告)、等级 7 表示 debug(调试信息)。通常将路由器日志输出等级设置在等级 4 或 5 以下。常见的日志信息主要有表示路由器端口的 line 和 line protocol 的两种状态:

(1)line 的状态为 up,表示路由器接收该线路接口的 DCD 信号为高;line 的状态为 down,表示线路接口的 DCD 信号为低。

(2)line protocol 的状态为 up,表示该线路接口的线路协议匹配成功;line protocol 的状态为 down,表示线路协议匹配失败。

通信系统在运行中可能出现一些故障,如何迅速地找出故障所在,并及时修改是维持系统正常运行的关键。例如对日志中以太端口的分析,可以用来检查一条链路的状态,如下所示:

- Ethernet 0 is up, line protocol is up →链路工作正常;
- Ethernet 0 is up, line protocol is down →连接故障,路由器未接到 LAN 上;
- Ethernet 0 is down, line protocol is down (disable)→接口故障;
- Ethernet 0 is administratively down, line protocol is down →接口被人为地关闭。

就路由器日志内容来看,它显示各端口的状态:在特定时刻的上线(up)或下线(down)。然而,当日志的量积累到一定程度时(如几兆或更多),其中就包含了一些规律。发现和利用这些规律对网络管理和网络维护都有很大的意义,但是目前尚未见到这类研究的相关报到。

数据挖掘技术中序列挖掘方法的研究与应用为路由器日志挖掘提供了技术支持。序列挖掘即是要挖掘出时间序列或者伪时间序列中经常出现的序列模式,称为频繁序列。序列挖掘是一类重要的数据挖掘问题,它有着广泛的应用前景并

*)基金项目:国家自然科学基金项目(编号:50378093)。庄 军 硕士研究生,主研方向:Data Mining。郭 平 副教授,主研方向:AI、GIS、Data Mining。

且已提出了许多算法。这些算法基本上可以分为两类。第一类是以高速内存中的操作代替对序列数据库的频繁读写。这类算法将序列数据库中的数据读出并经过压缩后存储在内存特定的数据结构中，该结构保持了数据之间的相关性。挖掘过程中，通过对内存数据结构的操作代替对序列数据库的操作从而提高算法的效率，如 FP-growth 算法[1]、SPADE 算法[2]等。第二类是直接读写序列数据库。这类算法首先需要生成频繁序列的候选序列，再通过查询序列数据库以确认候选序列是否为频繁序列，最终获得频繁序列。如 Apriori all、Apriori some、Dynamic Some[3]、GSP[4]、SCG[5]等算法即是如此。本文选用了 SCG 算法。

3 路由器日志挖掘的方法

路由器日志挖掘需要解决序列的集成以及时间序列挖掘算法等问题。

为叙述方便，这里先给出几个相关的概念。

定义 1 可辩识的对象 $a_j(1\leqslant j\leqslant m)$称为项目，它们的集合 $I=\{a_1,a_2,\cdots,a_m\}$称为项目集。$t\in 2^I$ 称项目集 I 上的序列，$D=\{t|t\in 2^I\}$称为待挖掘的数据集或序列数据库。

定义 2 给定常数 ξ称为最小支持度。对序列 $\eta\in D$

$$s(\eta)=\frac{|\{t|\eta\subseteq t \text{ and } t\in D\}|}{|D|}$$

称为 η在序列数据库 D 上的支持度。其中：$|X|$为集合 X 中元素的个数。如果

$$S(\eta)\geqslant\xi$$

称 η为 D 上的频繁序列模式或频繁序列。

3.1 时间序列集成和时间窗口的划分

从路由器日志中抽取出适合序列挖掘算法挖掘的序列数据库是进行挖掘的第一步。路由器日志文件中的每条数据由于含有时间戳，使它保持着严格的时间顺序。对这类时间序列信息的处理通常有两种方法：一是将它转换成不带实时约束的时间序列，即通过设置时间窗口，将在同一时间窗口中出现的事件看作是同时发生的，不同时间窗口中的事件被看成是有时序关系的；另一种是每个事件带有实时约束，即考虑每个事件间有时序关系。

我们采用第一种方法来集成路由器日志序列数据库。因为我们挖掘的目的是找出频繁 up 和(或)down 的端口的组合，并由此分析相应线路的状况。通过设置不同的时间窗口，可以获得出现 up 和(或)down 的端口的不同组合，有利于挖掘达到挖掘目标。另一方面，如果日志序列被转换成带实时约束的 up 和(或)down 序列，将难于获得不同端口的组合，不便于达到挖掘目的。

在实际的路由器日志处理中，设时间窗口的长度为 w，对日志的处理如下：

(1)读取日志文件的第一条记录，记它的生成时间为 t_1；

(2)顺序读取日志文件中的 $k-1$ 条记录，第 k 条记录的生成时间为 t_k。要求：

$$t_k-t_1\leqslant w$$

且对第 $k+1$ 条记录有：

$$t_{k+1}-t_1>w$$

其中 t_{k+1}是第 $k+1$ 条记录生成的时间。将这 k 条记录对应的 up 和(或)down 及端口的组合作为同时发生的事件存入日志序列数据库中。

(3)对第 $k+1$ 开始的记录做类似的处理，最后形成待挖掘的日志序列数据库。

3.2 SCG 算法

SCG 算法是文[5]中提出的用于挖掘序列数据库的一个序列挖掘算法。首先，通过 CTG 算法将序列数据库 D 转换成对应的图 G 和矩阵 T(算法 CTG)；其次，将与 G 中完全子图对应的序列作为 D 上频繁序列的候选序列；最后，通过搜索 D 从候选序列中确定频繁序列。该算法将候选序列的生成与图的完全子图联系起来，有效地解决了候选序列生成的效率问题。

记长度为 k 的频繁序列为 L_k，C_k 记长度为 k 的候选序列，V 记 G 的顶点集合，E 记 G 的边集合。

算法 CTG

输入：序列数据库 D，最小支持度 ξ；

输出：图 G，矩阵 T；

步骤：

// 生成矩阵 T

第 1 步：统计 D 中出现的项目并计算每个项目的支持度。将支持度不小于 ξ 的项目作为矩阵 T 的行和列；

第 2 步：矩阵 T 的元素值初始化为 0；

第 3 步：对 D 中的每一个序列，在 T 中记录各项目对出现的次数。设$\{x_1,\cdots,x_n\}$是 D 中的序列，将 T 中 x_i 所对应的行与 x_j 所对应列交叉处的元素(记为：$T(x_i,x_j)$)的值加 1，$1\leqslant i,j\leqslant n$。

第 4 步：修改 T 的元素的值。设 x,y 是 T 的行列对应的项目，重新给 $T(x,y)$赋值为：

$$T(x,y)=\begin{cases}0 & \dfrac{T(x,y)}{|D|}<\xi\\ 1 & \text{其它}\end{cases}$$

// 生成图 G

第 6 步：生成以 T 为邻接矩阵的图 G。图 G 的顶点对应于 D 中的项目。

将与 G 中完全子图对应的序列作为 D 上频繁序列的候选序列得到算法 SCG 中候选序列生成算法如下。

算法 SCG 中候选序列生成算法 // 由长度为 k 的频繁序列集合 L_k 生成长度为 $k+1$ 的候选序列

输入：图 G，候选序列长度 $k+1$，长度为 k 的频繁序列集合 L_k

输出：长度为 $k+1$ 的候选序列集合 C_{k+1}

步骤：

第 1 步：if $k=1$ then $C_1=\{v|v\in V\}$；return C_1；

第 2 步：if $k=2$ then $C_2=\{\{u,v\}|u,v\in V \text{ and } uv\in E\}$；return C_2；

第 3 步：$C_{k+1}=\phi$；

第 4 步：for$\forall S_1,S_2\in L_k$ and $|S_1\cap S_2|=k-1$ do

```
{
  {u,v}=(S1∪S2)-(S1∩S2);
  if{u,v}∈L2 then
    Ck+1=Ck+1∪(S1∪S2); // 将(S1∪S2)作为长度为k+1的候选序列
}
```

第 5 步：return C_{k+1}

利用上述算法生成的候选序列，SCG 算法可以得到最大频繁序列集。下面我们讨论利用 SCG 进行路由器日志挖掘的过程。

4 路由器日志挖掘的过程

挖掘实验基于国内某大型企业广域网中的一台 Cisco 路由器(型号 7513)的日志,时间标记为 2004 年 6 月 18 日至 2004 年 10 月 18 日,总计 9 千余条记录。以下是日志中截取的一段:

Jun 18 06:53:53: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial5/1/0:4, changed state to down

Jun 18 06:53:53: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial5/1/0:6, changed state to up

Jun 18 06:53:53: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial5/0/0:8, changed state to up

Jun 18 06:53:53: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial5/0/0:9, changed state to down

Jun 18 06: 54: 04: % LINK-3-UPDOWN: Interface ATM8/1/0, changed state to down

Jun 18 07:06:53: %FIB-4-RPPREFIXINCONST2: RP missing prefix for 66.0.233.60/30 (present in routing table)

4.1 数据清理

删除日志中与数据挖掘不相关的数据,挖掘路由器日志的目的是发现各端口的 up 和 down 的变化模式,其它信息暂时作为无用记录予以删除。

如上述日志中的最后一条记录(记录时间为 Jun 18 07:06:53)作为无用记录删除。

4.2 分类

由于路由器的日志变化反映与路由器各端口相连接的网络其它设备的工作状况,于是将经过数据清理后的数据按照分类 ATM (Asynchronous Transfer Mode 异步转换模式)和 Serial 得到不同的两类数据集。下面,我们仅以 Serial 对应的数据集的处理为例来讨论。

4.3 时间窗口设置

针对不同时间窗口的设置可以获得不同的序列数据库。经过分析和比较,Serial 对应数据集的时间窗口选择 15 分钟较适合,由此可以获得相应的序列数据库,以下是数据库中的几条记录:

0/0:9up;0/0:2up;1/0:9up;0/0:2down;1/1:1down;1/0:1down;0/1:3down;1/0:9down;0/0:2up;1/1:1up;0/0:9down;1/0:1up;0/1:3up;1/0:9up;0/0:9up;1/0:4down;0/0:8down;0/0:8up;1/0:4up;

0/0:8up;1/0:4down;0/0:8down;0/0:8up;1/0:4up;1/1:1up;1/0:4down;1/0:4up;

0/0:9down;0/0:9up;0/0:8down;0/0:8up;0/0:9up;0/0:8down;0/0:9down;0/0:8up;0/0:9down;0/0:8up;0/1:1up;0/0:9up;0/0:9down;1/1:1down;

1/1:1down;0/1:1down;0/0:8up;0/0:1up;0/0:3up;0/1:1up;1/1:1up;0/0:9up;0/0:8down;0/0:9down;0/0:8up;1/0:3up;

由于该路由器的串口(用于广域网)位置全部位于 5 号插槽上,因此为了方便程序中的数据处理,我们将其中每条记录长度进行了缩减。例如:%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial5/0/0:8, changed state to up 被简写成 0/0:8up,表示串口位于 5 号插槽 0 号模块 0 号端口 8 号子端口的状态变化为 up。

4.4 序列挖掘结果和分析

对 4.3 节中获得的序列数据库,在支持度为 9%时,使用 SCG 算法挖掘得到长度为 4 的频繁序列有 2 个,分别是:

1:0/0:9up—0/0:8down—0/0:9down—0/0:8up

2:1/0:4down—0/0:8down—0/0:8up—1/0:4up

第一个序列反映路由器端口 Serial5/0/0:9 和 Serial5/0/0:8 的 line protocol 状态变化的顺序,即 Serial5/0/0:9 的状态变化伴随着 Serial5/0/0:8 的状态变化,并且这种伴随关系在日志集中频繁出现。第二个序列反映 Serial5/1/0:4 和 Serial5/0/0:8 的 line protocol 状态变化的顺序,即 Serial5/1/0:4 的状态变化伴随着 Serial5/0/0:8 的状态变化,并且这种伴随关系同样在日志集中频繁出现。

针对上述的挖掘结果,我们检查路由器的连接发现 Serial5/0/0:9、Serial5/0/0:8 和 Serial5/1/0:4 这三个端口分别连接到三个比较偏远的下属企业,并且使用的恰好是网络运营商的同一台中间设备。通过对该中间设备的设置进行调整后,路由器上述端口的日志量有了大幅下降,这三条线路的运行状况也有明显的改善。

结束语 使用序列挖掘算法发现路由器日志内包含的有意义的信息,并将这些信息运用在实践中可以辅助网络管理人员做好网络的监测工作,改进网络运行状况,从而提高网络的运行效率。本文针对路由器日志所做的挖掘研究仅是对路由器日志数据处理的一种尝试,如何合理、充分地利用路由器日志中包含的信息为网络维护服务还需要做进一步的研究工作。

参 考 文 献

1 Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation. In :Proc. of the ACM SIGMOD Intl. Conf. on Management of Data,2000. 1～2

2 . Zaki M. SPADE: An efficient algorithm for mining frequent sequences. Machine Learning, 2001,40:31～60

3 Agrawal R, Srikant R. Mining sequential patterns. In :Proc. of the Eleventh Intl. Conf. on Data Engineering, 1995. 3～14

4 Srikant R, Agrawal R. Mining quantitative association rules in large relational tables. In :Proc. of the ACM SIGMOD Conf. on Management of Data, 1996. 1～12

5 郭平, 刘潭仁. 基于图结构的候选序列生成算法. 计算机科学, 2004,31(1),136～139

6 刘同明. 数据挖掘技术及其应用. 国防工业出版社, 2001

7 Agrawal R, Srikant R. Fast algorithms for mining association rules. In :Proc. of 20th Intl. Conf. on Very Large Data Bases, 1994. 487～499

8 Han J , Kamber M. 数据挖掘——概念与技术. 北京:高等教育出版社,2001