

基于模型的局域网诊断系统 An Diagnosis System of LAN Based on Model

黄燕勤¹ 黄燕钧²

Huang Yanqin Huang Yanjun

(1. 梧州市信息中心 梧州 543003; 2. 广西大学梧州分校 梧州 543002)

(1. Information Management Center of Wuzhou, Wuzhou, 543003;

2. Wuzhou Branch of Guangxi University, Wuzhou, 543002)

摘要 基于模型诊断的 HS-DAG 算法, 提出一个网络通信模型的诊断系统. 当局域网出现故障时, 能够根据观察结果判断故障出现的位置, 不会因为剪枝而丢掉部分最小碰集, 提高了故障诊断效率.

关键词 诊断系统 局域网 冲突集 碰集 HS-DAG 图算法

中图法分类号 TP393.1

Abstract Because of the complexity of the LAN management, a diagnosis system of network communication model, which adopts the HS-DAG algorithm based on model diagnosis is constructed to compute the diagnosis set where the faults happen. When the LAN network does not work, the diagnosis fault place on basis of the observed result according to this diagnosis system can be judged quickly, and the diagnosis efficiency is improved.

Key words diagnosis system, LAN, conflict-set, hitting-set, HS-DAG algorithm

随着社会信息化程度的不断加深, 各种网络故障也不断增加. 而且, 一旦出现通信故障, 要在星罗棋布的网络通信设备中找出故障需要花费很多的时间与精力, 尤其是当计算机处于大楼的不同层、不同房间时更是麻烦. 网络管理人员可以根据网络的特点来排查故障. 尽管如此, 一旦某台计算机不能与主机或其他计算机进行通信, 要想准确地定位发生故障的位置, 也要花费不少的时间和精力. 因此, 寻找一种快速有效的解决途径是当务之急. 基于模型的诊断系统可以将网络管理人员的经验与计算机网络通信原理结合起来, 可大大提高故障诊断的准确性与效率.

1 局域网虚拟模型

假设一个局域网络中的计算机分别处于办公楼的三层, 每层用一个 Hub 将计算机连接在一起, 如图 1 所示.

在网络故障中,网络通信软件相对较稳定,而硬件出现故障的几率较高。为方便研究,在模型中进行简化,只考虑网线两端的接口以及 Hub 的稳定性,并假设其他设备能够稳定运行。在该模型中: S 表示网络中的服务器; $C_1、C_2、C_3、C_4$ 表示域网络连接的客户机; $H_1、H_2、H_3$ 表示局域网中的 Hub; $L_1、L_2、L_3、L_4、L_5、L_6、L_7$ 表示连接相应设备之间的电缆。

2 基于模型的诊断系统及相关定义

一个诊断系统定义为一个三元组 $(SD, COMP, OBS)$,

其中: SD 是系统描述,描述系统部件的拓扑关系、每个部件的功能等,它是一阶谓词公式的集合。

图 1 的系统描述可表示为:

(1) 各部件之间的关系(输入与输出之间的关系)如下:

$$\text{out}(L_1) = \text{in}(H_1) \wedge$$

$$\text{out}_1(H_1) = \text{in}(L_2) \wedge \text{out}_2(H_1) = \text{in}(L_3) \wedge;$$

$$\text{out}_3(H_1) = \text{in}(L_4) \wedge \text{out}(L_2) = \text{in}(H_2) \wedge \text{out}(L_4) = \text{in}(H_3) \wedge;$$

$$\text{out}_1(H_2) = \text{in}(L_5) \wedge \text{out}_2(H_2) = \text{in}(L_6) \wedge \text{out}(H_3) = \text{in}(L_7).$$

注:其中 $\text{in}(L_1) = \text{in}(H_1)$ 表示一个式子, \wedge 是逻辑联接符号,连接两个表达式,可以去掉此符号。

(2) 系统行为 $()$:

$$\neg \text{ab}(L_1) \rightarrow \text{out}(L_1) = \text{in}(L_1);$$

$$\neg \text{ab}(L_2) \rightarrow \text{out}(L_2) = \text{in}(L_2);$$

$$\neg \text{ab}(L_3) \rightarrow \text{out}(L_3) = \text{in}(L_3);$$

$$\neg \text{ab}(L_4) \rightarrow \text{out}(L_4) = \text{in}(L_4);$$

$$\neg \text{ab}(L_5) \rightarrow \text{out}(L_5) = \text{in}(L_5);$$

$$\neg \text{ab}(L_6) \rightarrow \text{out}(L_6) = \text{in}(L_6);$$

$$\neg \text{ab}(L_7) \rightarrow \text{out}(L_7) = \text{in}(L_7);$$

$$\neg \text{ab}(H_1) \rightarrow \text{in}(H_1) = \text{out}_1(H_1) + \text{out}_2(H_1) + \text{out}_3(H_1);$$

$$\neg \text{ab}(H_2) \rightarrow \text{in}(H_2) = \text{out}_1(H_2) + \text{out}_2(H_2);$$

$$\neg \text{ab}(H_3) \rightarrow \text{in}(H_3) = \text{out}(H_3).$$

注:“ \neg ” 是一个整体函数符号。

$COMP$ 为系统的组成部分,是系统中待诊断对象。它是可能出异常的对象集合,是一个有限的常量集。另外,当部件 C 正常时, $\neg \text{ab}(C)$ 为真,当部件 C 不正常时, $\neg \text{ab}(C)$ 为假。

图 1 的系统部件可表示为:

$$COMP = \{H_1, H_2, H_3, L_1, L_2, L_3, L_4, L_5, L_6, L_7\}.$$

OBS 为一观察集,是系统的输入与输出值的集合,是一阶谓词公式的有限集,诊断的根据来自观测。在该模型中服务器作为输入端,客户机作为输出端。 OBS 就是描述输入端(服务

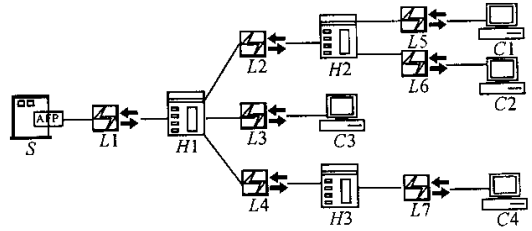


图 1 局域网虚拟模型

器)及输出端(客户机)的一种描述(从而判断他们能否正常通信)。

定义1 诊断系统: 一个诊断系统由系统描述 SD (一般是描述组件行为和系统结构的谓词逻辑语句的集合)和诊断部件集合 $COMP$ 组成。

定义2 诊断: 假设 $(SD, COMP)$ 是一个诊断系统, OBS 是一个观察集. 一个集合 $\Delta \subseteq COMP$ 是诊断系统 $(SD, COMP)$ 的一个诊断, 当且仅当 $SD \cup OBS \cup \{\neg ab(C) | C \in COMP \setminus \Delta\} \cup \{ab(C) | C \in \Delta\}$ 是不一致的(矛盾的)。

注: Δ 是有故障的部件, $COMP \setminus \Delta$ 表示除去出现故障的部件之外的正常部件。

从上述定义中推演出:

(1) 如果 Δ 是 $(SD, COMP, OBS)$ 的一个诊断, 那么对每一个 $C_i \in \Delta, SD \cup OBS \cup \{\neg ab(C) | C \in COMP \setminus \Delta\} \models ab(C_i)$;

注: \models 表示蕴含的意思。

(2) Δ 是 $(SD, COMP, OBS)$ 的一个诊断, 当且仅当 $SD \cup OBS \cup \{\neg ab(C) | C \in COMP \setminus \Delta\}$ 是一致的。

也就是说, 诊断是根据观察结果与系统描述的行为, 利用谓词逻辑理论, 推导出不正常的部件. 其原理如图2所示. 在进行诊断之前, 先给出相关的一些定义。

定义3 冲突集 CS 是一个部件的集合, 若这个集合中的全部部件均“正常”工作, 则与观测到的现象不一致. 或者说: 冲突集中至少有一个部件是有故障的. 如假设 C_1 不能正常通信, 则 $CS = \{L_1, H_1, L_2, H_2, L_5\}$, 而 $CS_1 = \{H_1, L_2, H_2, L_5\}$ 则不能是一个冲突集, 因为若 CS_1 中的部件均正常工作, C_1 也不能确定是正常工作, 因为 L_1 不能确定是否正常。

定义4 最小冲突集 是冲突集, 但是, 它的任意非空真子集均不是冲突集。

定义5 碰集 HS 给定一个集合簇(集合的集合), 若一个集合, 与这个集合簇中的全部集合的交集均不为空集, 则称该集合为此集合簇的碰集。

定义6 最小碰集 一个碰集的任意非空真子集均不是碰集, 则称它为最小碰集。

R. Reiter^[1]在“Artificial Intelligence”论文中证明了如下定理: 全部最小冲突集的最小碰集就是系统的诊断。

所以, 一个系统的诊断求解, 可以部分地转化为计算冲突集的最小碰集的问题. 因此, 冲突集和碰集, 就是基于模型诊断的主要概念. 计算冲突集的方法主要有:

(1) 基于假设的真值保持系统 (Assumption-based Truth Maintenance System, 简称 ATMS);

(2) 树结构;

(3) 程序诊断分片算法。

计算碰集算法有: (1) HS-树^[1]; (2) HS-DAG图^[2]; (3) BHS-树; (4) 布尔代数算法; (5) 遗传算法。

本文采用 HS-DAG 图的方法一边计算冲突集一边计算碰集。

定义7 (用 HS-DAG 图计算诊断) 假设 $TP(SD, COMP, OBS)$ 是一个诊断系统, F 是其冲突集的集合, 采用树结构计算系统的诊断的步骤如下:

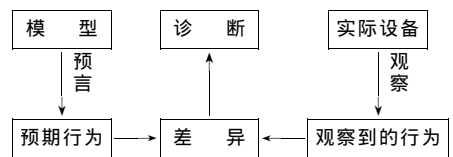


图2 诊断原理

(1) 生成一个根结点 n_0 , 定义 $H(n_0) = \{\}, F = \{\}$, 计算 $F_0 = TP(SD, COMP \setminus H(n_0), OBS)$, 如果 $TP(SD, COMP \setminus H(n_0), OBS)$ 是一致的, 则 $H(n_0)$ 就是系统的一个诊断, 否则, 可得到一个冲突集 F_0 , 将其并入 F , 即 $F = F \cup F_0$. 当系统有故障产生时, 第一次计算一定可得到一个冲突集;

(2) 对每一个 $a \in F_0$, 生成一个新的结点 n_i , 将 n_0 到 n_i 的边标记为 a , 同时计算 $H(n_i) = H(n_0) \cup \{a\}$, $F_i = TP(SD, COMP \setminus H(n_i), OBS)$, 如果是一致的, 则将该结点标记为 \surd , $H(n_i)$ 就是一个诊断. 否则, 将 $F(n_i)$ 并入 F 中, 即 $F = F \cup F(n_i)$;

(3) 如果新生成的结点 n_i 的集合 $H(n_i)$ 是某个标记为 \surd 的结点 n_j 的集合 $H(n_j)$ 的超集, 则将该结点关闭; 如果新生成的结点 n_i 的集合 $H(n_i)$ 与已存在的某个结点 n_j 的集合 $H(n_j)$ 相等, 则将该结点与已存在的结点合并; 如果新生成得结点 n_i 的集合 F_i 是已存在的某个结点 n_j 的集合 F_j 的子集, 则将结点 n_j 的所有 $a \in F_j - F_i$ 的边的子树删除;

(4) 如果存在既没有关闭又没有标记的结点, 则返回 (2). 否则, 最后存在的叶子结点就是所有的诊断.

3 局域网故障的诊断方法

在进行诊断之前, 先将局域网描述成一个 $(SD, COMP, OBS)$ 系统, 其中 $SD, COMP$ 如上所述, 假设 $OBS = \{in(L1) = in(S) \wedge out(C1) = in(S) \wedge out(C2) \neq in(S) \wedge out(C3) = in(S) \wedge out(C4) = in(S)\}$, 即在诊断系统故障时服务器发出一条信息, 如果局域网中各设备正常, 那么每个客户机都应该收到信息. 如果有客户机不能收到, 则说明存在故障, 需要诊断系统进行诊断.

根据系统行为与观察行为, 应用“定义7”的知识可以推断出各冲突集:

(1) 第1次计算 $TP(SD, COMP \setminus \{\}, OBS)$ 得到一个冲突集, 将其并入 F 使得 $F = \{\{L1, H1, L2, H2, L6\}\}$;

(2) 第2次计算 $TP(SD, COMP \setminus \{L1\}, OBS)$, $TP(SD, COMP \setminus \{H1\}, OBS)$, $TP(SD, COMP \setminus \{L2\}, OBS)$, $TP(SD, COMP \setminus \{H2\}, OBS)$, $TP(SD, COMP \setminus \{L6\}, OBS)$.

生成其下一代结点如图3所示,

其中 $n_0: H(n_0) = \{\}, F = \{\}, TP(SD, COMP \setminus \{\}, OBS) = \{L1, H1, L2, H2, L6\}$;

$n_1: H(n_1) = \{L1\}, F = \{\{L1, H1, L2, H2, L6\}\}, TP(SD, COMP \setminus \{L1\}, OBS) = \{H1, L2, L3, L4, H2, L5, L6, H3, L7\}$;

$n_2: H(n_2) = \{H1\}, F = \{\{L1, H1, L2, H2, L6, L4, H3, L7\}, \{H1, L2, L3, L4, H2, L5, L6, H3, L7\}\}; TP(SD, COMP \setminus \{H1\}, OBS) = \{L2, L3, L4, H2, L5, L6, H3, L7\}$;

$n_3: H(n_3) = \{L2\}, F = \{\{L1, H1, L2, L4, H2, L6, H3, L7\}, \{H1, L2, L3, L4, H2, L5, L6, H3, L7\}, \{L2, L3, L4, H2, L5, L6, H3, L7\}\}; TP(SD, COMP \setminus \{L2\}, OBS) = \{L1, H1, L3, L4, H2, L5, L6, H3, L7\}$;

$n_4: H(n_4) = \{H2\}, F = \{\{L1, H1, L2, L4, H2, L5, L6, H3, L7\}, \{H1, L2, L3, L4, H2, L5, L6, H3, L7\}, \{L2, L3, L4, H2, L5, L6, H3, L7\}, \{L1, H1, L3, L4, H2, L5, L6, H3, L7\}\}; TP(SD, COMP \setminus \{H2\}, OBS) = \{L1, H1, L5, L6, L4, H3, L7\}$;

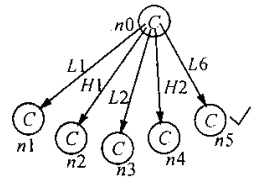


图3 下一代结点

$n_5: H(n_5) = \{L_6\}, F = \{\{L_1, H_1, L_2, L_4, H_2, L_5, L_6, H_3, L_7\}, \{H_1, L_2, L_3, L_4, H_2, L_5, L_6, H_3, L_7\}, \{L_2, L_3, L_4, H_2, L_5, L_6, H_3, L_7\}, \{L_1, H_1, L_3, L_4, H_2, L_5, L_6, H_3, L_7\}, \{L_1, H_1, L_5, L_6, L_4, H_3, L_7\}\};$

$TP(SD, COMP \setminus \{L_6\}, OBS) = CONSISTANT.$

(3) 完成这一步可以得到一个诊断 $H(n_5) = \{L_6\}$, 然后分别分析其它新生成的结点 n_i 的 $H(n_i)$:

$n_1: H(n_1) = \{H_1, L_2, L_3, L_4, H_2, L_5, L_6, H_3, L_7\};$

$n_2: H(n_2) = \{L_2, L_3, L_4, H_2, L_5, L_6, H_3, L_7\};$

$n_3: H(n_3) = \{L_1, H_1, L_3, L_4, L_5, L_6, H_3, L_7\};$

$n_4: H(n_4) = \{L_1, H_1, L_5, L_6, L_4, H_3, L_7\}.$

注意到这些结点的集合 $H(n_i)$ 都是标记为 \checkmark 的结点 n_5 的 $H(n_5)$ 的超集, 所以将它们都关闭. 树结构如图 4 所示.

最后计算得到的最小碰集为 $\{L_6\}$, 这就是局域网络模型诊断得到的产生故障部件.

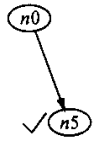


图4 树形结构

4 结束语

本文提出的局域网故障诊断是一种基于知识和经验的系统, 要求系统设计人员对相应的系统有充分的理解, 从而得到准确的系统描述(模型). 如果系统描述中不准确或有错误, 则很可能使诊断结果不可靠; 要对在该系统操作过程中得到的经验进行整理, 使得该诊断系统能够作出准确、高效的判断. 计算冲突集和碰集的算法比较多, 选择合适的算法对问题的解决有很大帮助. 局域网的组网模式虽然有很多种, 但多数可以将整个局域网结构划分为一个和多个树状结构的部分来分别进行故障诊断排除. 所以, 本文采用的 HS-DAG 图算法, 不失为一种较好的故障诊断办法, 且不会因为剪枝而丢掉部分最小碰集^[3].

参考文献

- 1 Reiter R. A theory of diagnosis from first principles. Artificial Intelligence, 1987, 32(1): 57~96.
- 2 Greiner R, Smith B A, Wilkerson R W. A correction to the algorithm in Reiter's theory of diagnosis (Research Note). Artificial Intelligence, 1989, 41(1): 79~88.
- 3 Wotawa F. A variant of Reiter's hitting-set algorithm. Information Processing Letters, 2001, (79): 45~51.

(责任编辑: 黎贞崇)