交换式互连局域网的流量分割算法

丁伟　　吴桦
东南大学计算机科学与工程系

摘　　要

本文提出了一个适用于大流量、多网段环境的交换式互连局域的网段划分算法,其基本原则是流量均衡。算法具体又分为二分划分和普通划分两部分。二分划分适用于网段数为二的整数次幂的划分；普通划分的原理基于二分划分，但适用于更一般的情况。文中对算法的正确性和时间复杂性等问题进行了比较详细的分析和讨论。

关键词　计算机网络；网络设计；交换式局域网；算法；算法分析

交换式互连网(SIN-Switched Internets)是目前各类企业网、校园网和机关网组网的一种结构。这类网络的核心设备是交换器(SWITCH)，其逻辑结构如图1所示。它的高速(一般在1 Gb/s以上）通过能力使各网段分别享有10Mb/s以上的带宽,从而可以满足基于客户机/服务器计算模式的网络应用和各类多媒体应用的需求。在总体上SIN的星型结构便于向ATM等更先进的技术过渡[1-2]。

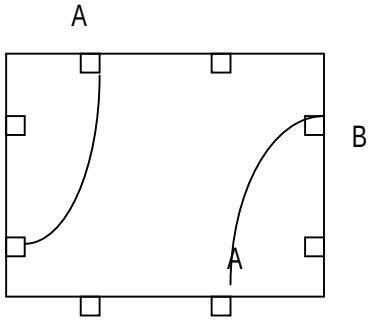


图1 交换器的逻辑结构

尽管目前交换器内部总线的带宽已达到若干个Gb/s,但由于任何穿越它的信息仍要在至少两个10 Mb/s的以太网上传递，因此用交换器设计网络时，端口的划分问题就

成为一个影响网络性能的重要问题。比较理想的划分方案是一个服务器/客户群位于一个端口，但在实际中更一般的情况是每个客户可同时是多个服务器的客户，而每个服务器也可能是其它服务器的客户机。在这样复杂的情况下,如何划分各端口的流量,从而使其具有更合理的结构呢？从理论上讲，这是一个拓扑设计的问题，但传统的算法无法适应SIN的情况,本文将给出一个划分网段的算法，其基本原则是流量均衡,它包括二分划分和普通划分两部分，前者适用于网段数为二的整次幂的情况，后者适用于一般情况.

1 定义和假设

在讨论该算法并对其进行分析之前，先给出一些形式化的假设和定义。

假设1:在用交换器互连的i段以太网上的信息传递延迟d_i，在带宽一定的条件下，与且仅与该子网上的流量x_i有关，即$d_i=f(x_i)$，且f是单调递增函数。

定义1:设一个交换以太网共有n个端口，其延迟分别为d_1,$d._2$..d_n，则该交换以太网的延迟为$D=d_i=Max\{d_1,d_2...dn\}$，并称i为最大延迟子网。

定义2:设一个交换以太网共有n个端口,各端口所连接子网的流量分别为$x_1,x_2...x_n$，若$x_j=Max\{x_1,x_2...x_n\}$,则称j为最大流量子网。且令$X=x_j$为最大流量。

定义3:在确定条件下，设共有k种流量分割方法，其延迟分别为$D_1,D_2...D_k$，$D_1=Min\{D_1,D_2...D_k\}$,则称l为最优划分。

根据定义3流量分割的过程实际上就是寻找该定义中最优划分l的过程。

结论1:若假设1成立，定义1中的最大延迟子网同时也是定义2中的最大流量子网。

该结论的证明从略，其意义在于将寻找最优划分时的延迟条件转化为流量条件，而对后者的计算要简单得多。由此可得以下推论:

推论 定义3中的最优划分l同时使$X_1=Min\{X_1,X_2...X_k\}$成立。

2 二分划分

由于目前交换器的端口数一般为4,8,16,32等二的幂次方个,因此对这种规律的端口数目可采用逐级一分为二的划分方法。

在任何情况下，将一个子集划分为a,b两个子集时原子集内部的流量被分割为三部分，即a子集内部的流量$X_{a'}$，b子集内部的流量$X_{b'}$和两个子集之间的流量X_{ab}。由于X_{ab}实际上将穿过交换器的两个端口，因此a子网上的实际流量为

$$X_a = X_{a'} + X_{ab} \quad (1)$$

同样

$$X_b = X_{b'} + X_{ab} \quad (2)$$

在将S二分划分为a和b的过程中，若让a为S的子集，则b为余集。根据集合论中的幂集定理[3],一个容量为n的集合S的子集数为2^n，在考虑到a和b的对称性以及它们不可能为空集和全集的情况下,将上述S二分为两个子集的方法多达2^{n-1}-1个。根据定义3,此时k=2^{n-1}-1，若穷尽这2^{n-1}-1划分，将可找到S的最佳划分I，然而当n=50时，2^{n-1}-1≈5.6295*10^{14}，在100万次/秒的工作站上约需执行5630天。因此用穷举法寻找最佳二分是不可能的。下面提出一种将集合二分划分的方法，它在一定的条件控制下可以达到最优，同时还可保证良好的时间复杂性。该方法的主要思路是将元素逐个从b向a过渡，此时b中的流量逐步减少，a中的流量逐步增大，当达到一定程度时，两者趋于平衡，则算法结束。根据使相关信息尽量在一个网上流动的基本原则，从a向b的过渡总是选择b中与a流量最大的元素。具体描述如下:

1）令a=Φ，b=S，X_a=0，X_b=X_s，最大流量子集c=b，最大流量X_c=X_b;

2）从b中任选一点放入a中,此时|a|=1，|b|=n-1;

3）按(1)式和(2)式计算X_a和X_b，并根据定义2求最大流量子集c';

4）若$X_{c'}$>=X_c则认为无改善,记录不修改,结束;否则记录本次修改，同时在b中选择与a流量最大的一点加入a,转3。

该算法的一个形式化描述如下:

```
procedure binary-division(n:int;F:array[1..n,1..n] of real);
      /* 将n台主机划分成两个子集,F为流量矩阵*/
var a,b:set of  1..n; /* 用来存放划分的结果*/
      fab:array[1..n]  of  int; /* 记录b中各点与a 的流量*/
      x_a,x_b,real; /*(1) 和(2) 中的X_a 和 X_b*/
begin   所有变量置初值;
      repeat
      until put-atob(fab,a,b, x_a,x_b );
```

/*put-atob 是一个布尔函数，它将b中与a流量最大的一个
元素加入a，如果修改后的最大流量X有所改善，则用
新的划分取代原有划分，并返回'真'，否则返回'假' */
输出a,b的返回结果
end;

put-atob 的形式化描述：
function put-atob(var fab:array[1..n] of real;var x_a,x_b:real;var a,b:set of 1..n);
begin 各变量置初值； put-atob=true;
寻找b中与a流量最大的元素k;
预修改x_a； 预修改x_b；
if 修改后最大流量有所改善
then 用新划分代替旧划分
else put-atob=false
end;

下面通过几个定理来分析上述算法的正确性和时间复杂性。

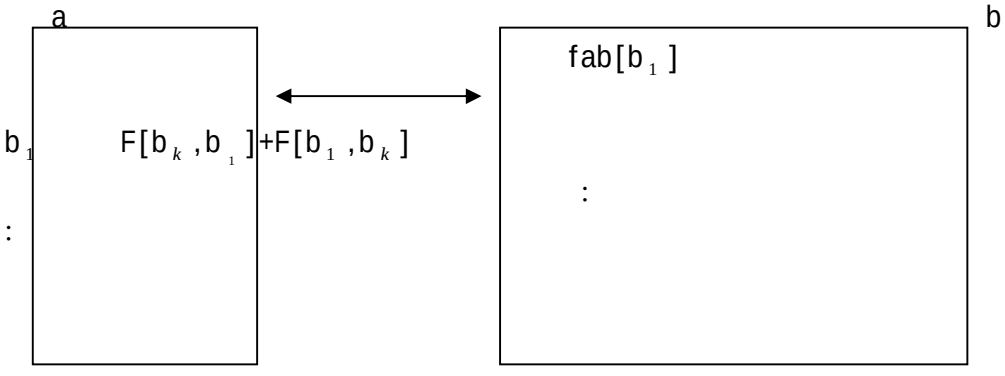
定理1：若fab矩阵中有fab[k]>0（即a和b之间的流量不为0），且put-atob为真，则b中的流量x_b单调减少，即若令x_{b0}为执行前的流量，x_{b1}为执行后的流量，且$x_{b1}<x_{b0}$.

证明：设修改前b=（b_1,b_2,...b_k...b_m），修改后将b_k移入a，则图2和图3描述了修改前和修改后流量的变化过程，图中各流量符号以及它们之间的关系符合（1）和（2）式的定义。根据（2）式和图2、图3，有：

$$x_{b1}=x_{b'}+x_{ab1}=x_{bo'}-\sum+fab[b_k]=x_{b0'}+x_{ab0}-fab[b_k]=x_{b0}-fab[b_k]$$

因为 $fab[b_k]>0$

所以 $x_{b1}<x_{bo}$ 证毕。



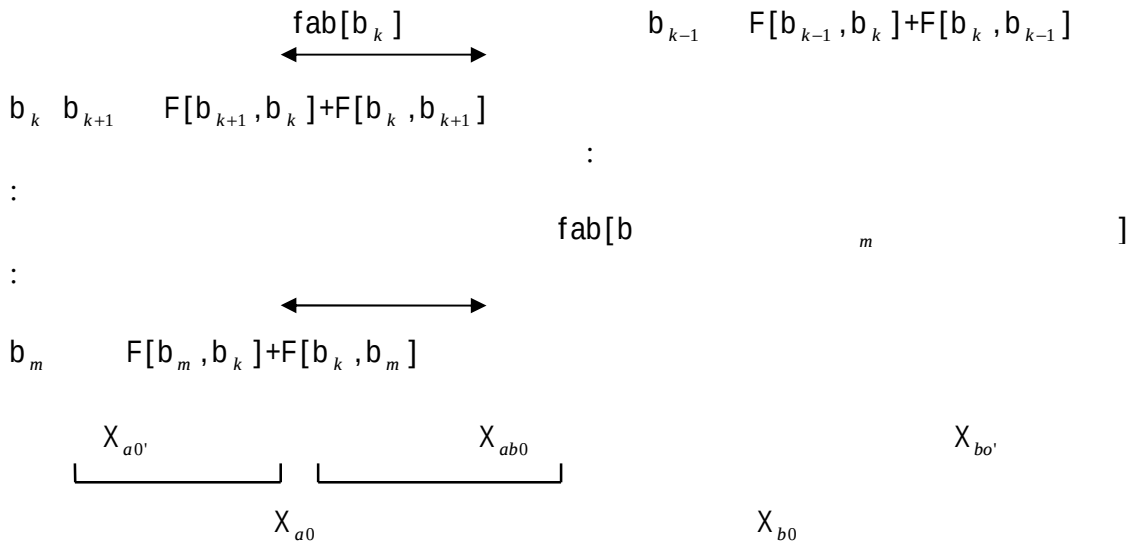
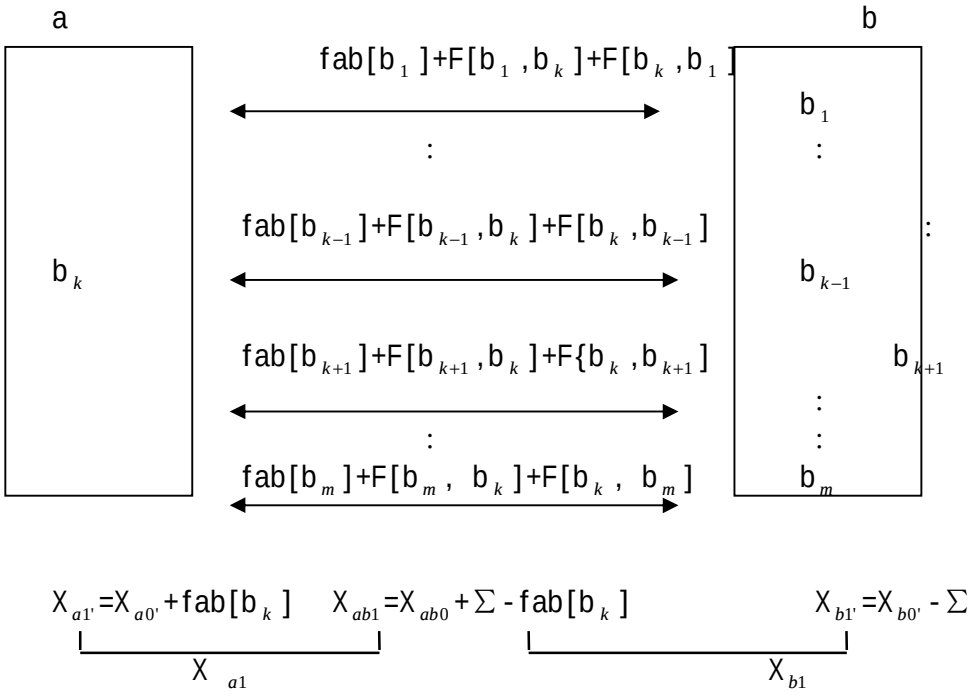


图2　划 分 前 的 流 量 分 布



$$\Sigma = \sum_{i=1}^{m} \left(F[b_i, b_k] + F[b_k, b_i] \right)$$

图3　划 分 后 的 流 量 分 布

定理2: 若n(主机总数)>3,且至少有两对主机间的流量大于0,则put-atob函数不会使b为空。

由于篇幅原因,该定理的证明从略,其意义在于它证明put-atob函数能够终止。

定理3: 算法put-atob的时间复杂性为O(|b|)。

定理4: 算法binary-division的时间复杂性为$O(|n^2|)$。

上述两定理证明从略。二分划分算法(binary-division)的每一步仅取决于当前的划分状况,因此有可能无法达到穷举法所能达到的最佳划分。在上述算法中,由于划分的第一步(即当a为空时,将b中的一个元素选入a),是整个划分的基础,因此是非常重要的。但这个选择在上述算法中却是任意的。因此,如果将binary-division算法增加一层循环,使其从每个点起始一次(可通过对a置初值实现),则可 以使其在更大的范围内选择最优。此时,算法的时间复杂性增加一个数量级,即从n^2变为n^3,但仍属可以容忍的范围。

3 普通划分

二分划分只能将结点的集合一分为二。虽然连续使用时可将其
分为2的幂次方个子集,在大多数情况下可以满足需求,但目前已有一些交换器为非2的幂次方个端口(如6个,12个等),将来这样的产品会更多。因此,下面将在二分划分的基础上讨论将结点的集合划分为任意多个子集的情况。

由于普通划分的情况较二分划分更复杂,因此同样无法用穷举法寻找最优。设m为所要划分的子集数,则普通划分算法的主要思想为:在m个集合中选择流量最大和流量最小的子集,将流量由大向小流动至平衡。

该算法在实现时可以有两种方式,其一是直接划分,它从起始时就设立m个集合,其中一个为全集,其它m-1个为空集,然后按上述划分思想重复进行划分,一次达到平衡;另一个是迭加划分,它从二分划分开始,每次达到平衡后再增加一个集合,直到集合数达到m为止。具体如下:

直接划分算法:

STEP1 $S_1=(1,2,...n);S_2=S_3=...=S_m=0;$

STEP2 在S_1到S_m中寻找流量最大的子集S_{max}和最小的子集S_{min}；

STEP3 将S_{max}中的一个与S_{min}流量最大的结点放入S_{min}中；

STEP4 重复STEP3 至平衡；

STEP5 重复STEP2、STEP3 和STEP4 至平衡。

迭加划分算法：

STEP1 S_1=(1，2，。。。)；k=2; /*k 是迭加控制变量*/

STEP2 S_k=0;

STEP3 在S_1到S_k中寻找流量最大的子集S_{max}和最小的子集S_{min}；；

STEP4 将S_{max}中的一个与S_{min}流量最大的结点放入S_{min}中；

STEP5 重复STEP4 至平衡；

STEP6 k++； 若k>m 则结束， 否则转STEP2。

在上述两算法中，结点的流动方式和平衡原则与二分划分相同，所要注意的是一个结点的移动不仅仅会使原集和目标集之间的流量发生变化，也会使这两个集合与其它集合之间的流量关系发生变化，与原集的有关流量应转到目标集，这一点与二分划分不同。但是可以看出，除了移动结点所涉及的两个集合外，其它集合的实际流量(即网内流量加网间流量，类似于二分划分中(1)式和(2)式的定义)，不会发生变化。

上述两算法在SUN 工作站上用C 实现后，通过多组仿真数据的比较，发现它们之间在比较重要的分组段(6-12 组)上没有明显的差别。

4 结论

随着计算机使用的普及，许多情况下小范围内的计算机密度可以达到很高。在用交换器等新设备设计互连局域网以支持客户/服务器计算模式和多媒体等新技术时，所面临的问题与当初广域网设计的情况相比有了较大的变化。交换网正逐步代替传统的共享网而逐步成为当今企业网、校园组网技术的主体。但目前适应这种新形势的、实用的网段划分方法尚不多见。该问题已引起人们的重视，有关的研究正在成为热点[4]。同时各种新技术，如仿真、系统工程和人工智能等，也被尝试用来研究该

领域中的问题[5]。本文以流量均衡为基本原则,提出了一个网段分割算法,它适用上述环境并具有较好的实用性。

参考文献

1 Robert Mandeville. Ethernet Switches Evaluated. DATA COMMUNICATION, McGRAW-HILL'S NETWORKING TECNOLOGY MAGZINE. 1994 (3): 8~10

2 Bradley F. Shimmin. Comparing Three Methods of Ethernet Switching.LAN TIMES, McGRAW-HILL'S INFORMATION SOURCE FOR NETWORK COMPUTING, VOL 1994, 11(1): 25~30

3 K.Kuratowski & A.Mostowski. Set Theory. New York:North-Holland,1976,35~40

4 L.J.Leblanc and S.Narasimhan.Topological expansion of metropolitan area network. Computer Network and ISDN Systems. 1994, 26(9): 1235~1248

5 H. Saito and T. Asaka. Traffic aspect of personal telecommunications in intelligent network. Computer Networks and ISDN Systems, 1994, 26(9): 1089~1100

A Traffic Distribution Algorithm for Switched LANs

Dingwei Wuhua
Computer Department of Southeast University

ABSTRACT

The paper gives out some latest development issues in the field of switched LANs.An algorithm is proposed which can be used in above environment with heavy traffic and multi-segments.The main principle of the algorithm is distributing the whole traffic to each segment as average as possible.The algorithm consists of binary division and ordinary division.When the number of segments to be divided equals to powers of 2,binary division is used;ordinary division is based on binary division but suitable to more common cases.Both correctness and time complex of the algorithm are discussed in detail.

Key words:Computer Network,Network Design,Algorithm,Switched LAN