面向会话的负载均衡简化算法[1]

龚 俭、陆 晟、芮苏英

（东南大学计算机科学与工程系　南京 210096）

摘要： 负载均衡算法被广泛应用于并行处理、服务集群等环境中。一些基于网络报文内容相关性的应用，例如 IDS 和 IPv6 的 Anycast 服务等要求在对报文进行负载均衡分配时要保持网络会话的相关性，即相关的报文要分配到同一个处理节点，否则其语义不能得到正确处理。传统的负载均衡算法对于这类服务需要在会话的上下文信息规模和会话完整度之间权衡，对于会话数量很大的情况通常开销也很大。基于位熵的概念，本文提出了一种可满足会话完整性的负载均衡简化算法——域分类算法。该算法不需要各处理机之间内部通信协调工作，也不需要在调度节点保持会话的上下文，在满足报文或会话相关性要求的同时，仍能保持较好的宏观平衡度和微观平衡度。

关键字：负载均衡、报文、位熵、位流熵、域分类

中图分类号：TP393.3　**文献标识码：**A

1　引言

随着数据传输技术的发展和网络规模的日益扩大，互联网的信道带宽不断增大，流量也随之剧增。因此，高速信道上的数据处理要求和设备处理能力之间的矛盾越来越凸现出来，而在处理能力不能满足需求时，采用负载均衡算法来缓解处理能力和待处理量之间的矛盾是一种常见的做法。

负载均衡的概念被广泛应用于许多领域，从并行计算到集群服务等，并发展出了静态和动态两类，包括几何算法、图论算法、节点变换的负载均衡算法等多种负载均衡算法[1]。这些负载均衡算法一般使用于分布式内存并行计算机，讨论的是任务分担和分布式调度问题。同时负载均衡也用于许多服务集群环境，例如分布式 FTP 服务器、Web 服务器集群等等，这些服务器往往使用一些无内部通信的算法或者是较少依赖于内部通信的算法进行负载均衡处理。这些算法包括[2][3]：random、round-robin、weighted round-robin、High Availability 等，以及某些可编程算法，这些算法可以依据 IP 地址或其它信息进行服务器选择。

但是存在一些基于网络报文内容相关性的应用，例如 IDS 和 IPv6 的 Anycast 服务要求在对报文进行负载均衡分配时要保持网络会话的相关性，即相关的报文要分配到同一个处理节点，否则其语义不能得到正确处理。例如，由于入侵检测的特殊需求，需要保证会话或相关信息的完整。如果有的攻击为了消除入侵检测系统所检测的特征，对会话和报文进行了分片，则入侵检测系统就需要将这些被人为拆分的报文重新组合后进行分析；此外，攻击行为之间的关联性分析也需要检测数据具备相关信息的完整性。这决定了入侵检测中的负载均衡不能够使用 Round Robin 等传统的负载均衡算法，而需要满足报文相关性等特殊需求并且任务分担平衡度较好的、新的负载均衡算法。

本文基于位流熵[4]的概念提出一种能够保持会话特性的负载均衡算法——域分类算法，该算法能够较好的保证处理机任务分配的均匀性，从而适用于入侵检测系统或相关的需要保持会话特性或流特性的高速网络数据处理环境。

---

[1]本文内容受国家自然科学基金重点课题“面向大规模网络的入侵检测与预警模型”（90104031）和 973 课题“网络动态行为理论”（2003CB314803）资助。作者简介：龚俭，工学博士，东南大学计算机系教授、博士生导师，主要研究方向为网络行为学、网络安全监测、网络体系结构。

本文的第 2 节给出了域分类算法的基本定义，第 3 节讨论了域分类算法中参数的选择，第 4 节和第 5 节分别讨论了这个算法的宏观均衡性和微观均衡性，并第 6 节将这个算法与传统的负载均衡算法进行了比较。

2 域分类负载均衡算法

在网络测量研究中可知，报文流可以通过特定的报文字段（以下称为域）加以标识和区分，例如通过源宿地址和源宿端口号，因此网络会话也可以通过类似的方式进行区分。但是从负载分配的角度看，在建立会话标识之后，需要在调度节点保持当前会话的上下文，以保证相关的报文被分配到相同的处理节点。但在会话数量较大的情况下实现的开销也会很大，如果限制当前保持的会话上下文数量，则长会话的完整性将不能得到保证。因此对于高速网络大流量的情形，其实用性往往受到限制，可扩展性不好。如果将会话标识直接作为负载分配的调度依据，则可以保证会话的完整性，而且也不需要在调度节点保持会话的上下文。但是由于会话长度的不均匀性，这种方法不能保证负载分配的均衡性。[4]在研究网络流量抽样测量模型时发现，IP 报文中某些域（例如分组标识字段）的值具有很好的随机性，因此将这种特性与会话的传统标识方法结合起来，可以获得一种无上下文，且可保持会话完整性的负载均衡方法，即借鉴报文抽样算法中对于随机性分析的思想，依据报文某些比特位对数据进行基于报文特征的负载分离，这就是域分类负载均衡算法的基本思想。

利用[5]中对于域(Field)、规则表(Rule List)的定义，基于报文的域分类负载均衡算法形式化定义如下[2]：

域分类算法 Algo：Algo(*O*, Λ, 〃)->i, i∈{1, 2, …, m}

其中：

- ***分类域生成运算*** *O*：O(F_1, F_2, …, F_n)->F
  F_1，F_2，…，F_n为报文 P 中抽取的 n 个域；O 为定义在 F_1~F_n上的运算；F 为一比特流，是 O 的运算结果，不妨视为一个特殊的域。
- ***分类操作***Λ： Λ(ℜ, F)->R
  ℜ是处理规则集，它限定了处理节点对 F 的处理范围；R 是对于规则集ℜ使 F 域为真的位流集。
- ***映射操作*** *M*：M(R)->i
  M 将 R 映射到节点标号 i 上；i∈{1, 2, …, m}。该操作将保证负载的均衡性。

为了保证算法的有效性，要求分类操作Λ的运算结果 R 的度必须为 1，即|R|=1，以保证负载分配结果的唯一性，即会话的完整性；要求域 F=O(F_1, F_2, …, F_n)必须保证相关性不丧失，即相关的报文必须具有相同的 F 值；这些要求可以通过对算法参数的选择来实现。根据上述的定义，域分类负载均衡算法可表为：

1) 从报文 P 中获取域 F_1、F_2、…、F_n的值；
2) 作用运算 O 于 F_1、F_2、…、F_n获得结果域 F；
3) 根据分类规则集ℜ对 F 进行分类操作Λ，得到分类结果 R；
4) 根据映射操作 M 将分类结果 R 映射到一个处理节点编号 i；
5) 根据分类编号 i，分配报文 P 到相应编号的处理节点。

注：步骤 3）中的分类操作可以直接使用常见的报文分类算法[6]。

从上可以看到，域分类算法是一种不需要处理机内部通信的负载均衡算法，适用于基于

[2]此处没有完全使用[5]中的定义，因为[5]中表述过强，而算法仅需要一个简单定义即可。

报文分配的服务集群环境。

3 域分类算法中参数的选择

域分类算法的操作过程是确定的，因此算法的核心在于域 F_i、操作 O、规则集$\Re$、分类操作 Λ 以及映射操作 M等参数的选择。对域分类算法选择不同的参数，得到的算法实例差异很大。这些算法实例的好坏，可以由负载平衡的平衡度加以衡量。本文将利用两个不同的测度作为负载平衡算法的比较依据：宏观平衡度和微观平衡度。在报文处理的负载均衡中，宏观平衡度与报文分配的随机性是等价的，因为均匀随机的等概率分配，能够保证各个处理机获得的报文数量在长时间粒度上非常接近，从而满足宏观平衡性的要求。本节将说明如何选择好的参数以保证域分类算法的宏观平衡性。

3.1 位熵和位流熵

熵是信息论中的基本概念，是各种随机试验不确定程度的度量。为了度量 IP 报文各比特的随机性，[4]将熵的概念推广应用到比特随机性的研究中，并定义其概念为：

定义 1 位熵是一个比特所表示的信息量，比特 b 具有 0、1 两种可能性，即 b={0,1}，且有 p(0)=p，p(1)=1-p，0≤p≤1，则位熵 H（b）定义为：

$$H(b) = -(p\log_2 p + (1-p)\log_2(1-p))$$

[4]中分析了位熵具有的物理含义：(1) 位熵 H（b）表示比特集合中事件出现的平均不确定度，是刻画比特串集合不确定的一个测度。(2) 位熵 H(b)表示变量 b 的随机性，变量随机性越小，则其熵越小，变量 b 取 p（0）和 p（1）是等概率的，其随机性大，它的熵也大。(3) 位熵 H(b)表示比特串输出每个符号所提供的平均信息量。因此位熵也是一种信息的测度。

在实际应用中，处理的往往是比特流，所以同样需要对比特流随机性定义的指标，[4]中采用了位流随机测度作为这个指标，而位流随机测度则是使用位流熵定义的。

定义 2 位流熵是比特流所表示的信息量，比特流 s 具有 $n=2^s$种可能性[3]，设其概率分别为 p_0、p_1、…、p_{n-1}，则位熵 H（s）定义为：$H(s) = -\sum_{i=0}^{2^s-1} p_i \log_2 p_i$ 。

从最大位流熵定理已知最大位流熵为 s，而位流随机测度被[4]定义为：

定义 3 位流随机测度（E）是位流熵 H（s）和最大位流熵 H_{max}（s）的比值，表示比特流随机程度，$E = H(s)/H_{\max}(s) = H(s)/s$。

3.2 域 F_i 和操作 O 的选择

位熵和位流熵分析主要用于选择合适的候选比特流，并由位流熵保证宏观平衡性的良好，然后使用负载平衡度分析以确定微观平衡度是否良好。

---

[3] [4]中定义的可能性为 $n+1=2^s$，可能性由 p_0到 p_n，不妨定义为 p_0到 p_{n-1}，则可能性为 n 种

在域分类负载均衡算法中所使用的位流熵不再仅仅是原始报文头中比特的位流熵，而是这些比特运算结果的位流熵。由于比特运算的结果仍旧是比特流，所以仍旧可以使用位流随机测度对比特的随机性进行分析，从中试图找出宏观平衡性好的比特流。现在所面对的问题就是如何选择参与运算的域以及这些域上的操作。

从最大位流熵定理[4]可知，只有各种情况等概率出现，位流熵才比较好，这也说明了如果作为位流熵构成成分的比特的位熵特性不好，则位流熵的性质也不会好。虽然这个结论对于运算后比特流不一定正确，但是以下讨论能够帮助确定好的域和域上的运算。

由于高位熵比特所组成的流的位流熵较高，首先分析各个比特运算后的位熵。虽然操作 0 允许任意运算操作，但是对于比特运算而言运算结果受制于真值表，实际上运算的种类是很有限的。因此可以通过分析两个比特之间的二目操作，归纳获得对多比特运算有参考意义的结论。两个比特之间的运算只有 $2^2\times2^2=16$ 种，如表 1 所示。

表 1 二目比特运算真值表

Table 1. Value table of Two-Element Operation

0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
A	B	0	&		A		B	$\oplus$	|		$\overline{\oplus}$	$\overline{B}$		$\overline{A}$			1

对于恒 0 运算和恒 1 运算显然生成比特的位熵为 0，无法保持良好的随机性，因此不能用。对于产生结果中 0 和 1 的比例为 3:1 或 1:3 的 8 种运算($C_4^1+C_4^3$)，如果 A 和 B 的位熵均较高，那么在 A 和 B 的相关性低(或无相关性)的情况下，生成出的结果中 0 和 1 的比例必然差异很大，所以位熵不会很高。当然，可以通过选择两个位熵很低的比特 A 和 B 以期望获得好的结果位熵。例如 A 和 B 都是非常可能出现 1 的，那么与(&)操作后的位熵反而可能得到很大改善。但是选择这样位熵特性的比特并不直观，因为难以确定这两个比特所应该具备的特性。

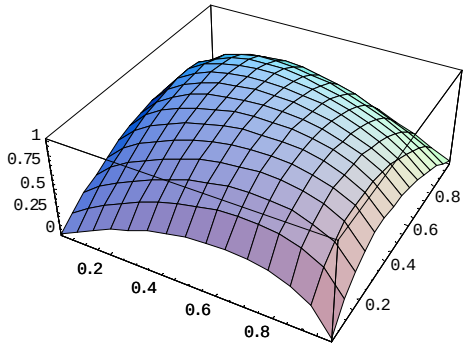

图 2 两值异或运算后的位熵

Fig. 2. Bit entropy after XOR operation

因此一般可以选择的操作为 0 和 1 的比例为 2:2 的六种(C_4^2)运算，对于 A、$\overline{A}$ 操作，其位熵等于 A 的位熵，因为 $H(\overline{A})=-((1-p)\log_2(1-p)+p\log_2 p)=H(A)$，同理 B 和 $\overline{B}$ 操作的位熵等于 B 的位熵。因此剩余异或操作和同或操作需要分析，而且可以发现这两种运算的位熵相同。令运算后出现 0 的概率为 p (求位熵后和出现 1 的概率计算的结果相同)，A 出现 1 的概率为 p_1，B 出现 1 的概率为 p_2，两者不相关，则：$p = p_1p_2 + (1 - p_1)(1 - p_2)$，其位熵为：

$$H=-(1-p_1)(1-p_2)\log_2(1-p_1)-p_1p_2\log_2 p_1-(1-p_1)(1-p_2)\log_2(1-p_2)-p_1p_2\log_2 p_2$$

在 $p_1\in(0,1)$ 和 $p_2\in(0,1)$ 上有如图 2 所示的位熵运算结果。

其中在 p1=p2=0.5 处有 H=1 为极值点。由于 $\frac{\partial^2 H}{\partial p_1} = \frac{p_1 + p_2 - 2p_1p_2}{-p_1 \ln 2 + p_1^2 \ln 2}$ 在以上区间中必小于 0，所以正如在图 2 中所看到的，H 为拱型结构。

但是如果 A 比特和 B 比特的相关性很大，则获得的位熵特性将无法保证。

所以，负载均衡算法需要选择的是具有良好位熵性质的报文头中无相关(或低相关)比特流进行运算所产生的比特流。幸运的是，一般情况下标识会话的域的位熵特性均比较好。例如源、宿地址、源、宿端口、IP 标识字段等[4]。

3.3 规则集$\Re$、分类操作 Λ 和映射操作 *M* 的选择

规则集$\Re$与处理语义有关，不能随意选择，但它不影响负载分配。

分类操作 Λ 就是根据报文 P 的具体域 Fi 以及规则集$\Re$所进行的一般分类处理，可以使用常用的报文分类算法。只要操作是确定的，相同的$\Re$和 F 总能得到相同的 R，从而保证会话的完整性。另外，如果将|R|限定为处理节点的数量 m，则分类操作同时完成了负载划分，因此映射操作 M 则可以简单使用等于操作，利用规则 R={R_i}中 R_i 所具体表示的值来获得分类编号 i。

在规则集$\Re$、分类操作 Λ 以及映射操作 *M* 确定之后，在分析域分类负载均衡算法的平衡度时，只需考虑域 F_i 以及操作 O 的实际选择方式。

4 域分类算法的宏观平衡度分析

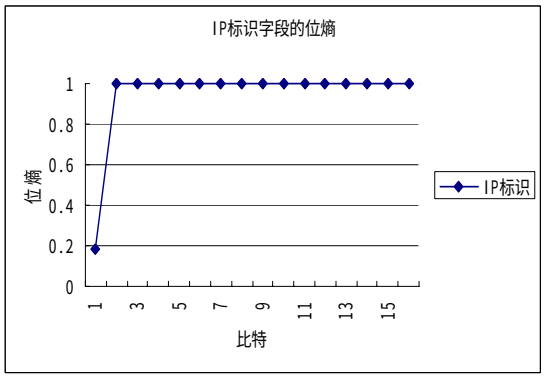

图 3 IP 标识字段的位熵

Fig. 3. Bit entropy of IP identification

宏观平衡度分析用于证明域分类算法在长时间粒度内，具有较好的平衡性；同时由于其运算简单，也是筛取域和操作的一种有效手段。本节通过从 CERNET 主干网络某处 67,870,553 个报文进行分析，来选择若干能够保证报文相关性的域分类算法实例。这些报文为 1 周内固定间隔采样获得，每 1 分 15 秒抽取 2 秒中内的全部报文。

由于本文的负载均衡算法要求被选用的域对未来需要考虑的报文相关性有决定性的意义，因此仅仅分析了部分在入侵检测中对报文相关性有贡献的 IP 字段的位熵和位流熵。[4]中虽然对 IP 报头中各个比特的位熵和各个常用域的位流熵均进行了分析，但是构造合理的入侵检测适用的域分类负载均衡算法却不能够简单的使用这些分析数据，而需要对流量比特的随机性进行应用扩展的相应分析，也就是综合考虑域和操作的结果进行分析。

算法 1 令 F_1 为 IP 标识字段，运算 O 为等于操作即：O: (=, F_1)[4]，得到域 F。

该算法使用的 IP 标识段在[4]中是被认为最优随机性字段。由图 3 可知，在选择规则集$\Re$所使用的比特时，不适宜选择最高比特，选择其余比特的效果均类似。

[4] O 的表示采用以操作符引导的 S 表达式说明，下同

算法 2 令 F_1 为源 IP 地址字段，F_2 为宿 IP 地址字段，运算 O 为异或操作即：O=(^, F_1, F_2)，得到域 F。

该算法使用的源 IP 地址和宿 IP 地址，也被认为有良好的随机性。图 4 是源 IP 地址、目的 IP 地址和它们异或关系的比特的位熵。

其中对于后十六比特的位熵及最后 7 比特(16 比特位上的点是第 16 比特的位流随机测度，15 比特位上的点是第 16 比特和 15 比特构成的流的位流随机测度，依次类推)的位流随机测度放大后如图 5 所示。

可以看到，运算后的位熵比较原来的生成流反而有了较大改善。这是由于图 2 中顶部比较平坦所带来的效应，表明异或运算可以在一定程度上改善报文比特的随机性。

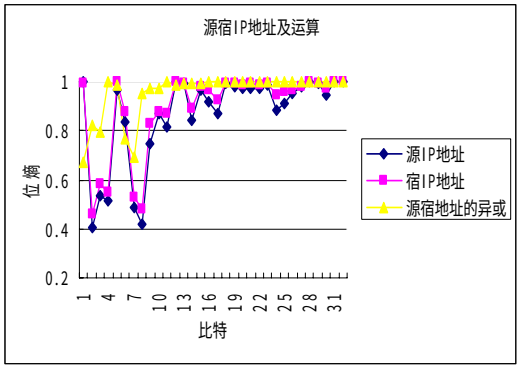

图 4 源宿 IP 地址及运算的位熵
Fig 4. Bit entropy of Source IP and Destination IP and their XOR

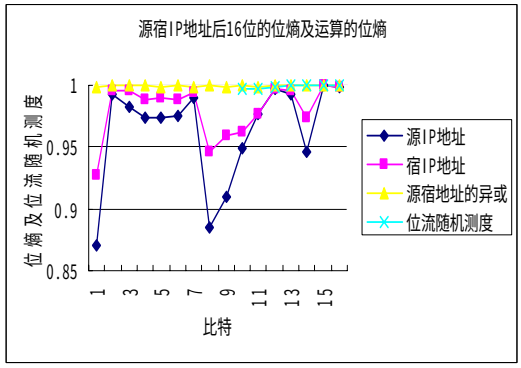

图 5 源宿 IP 地址后 16 位的位熵及异或运算的位熵
Fig. 5. Bit Entropy of Low 16bits in Source IP and Destination IP and their XOR

算法 3 令 F_1 为源端口字段，F_2 为宿端口字段，运算 O 为异或操作即：O=(^, F_1, F_2)，得到域 F。

算法 4 令 F_1 为源 IP 地址字段，F_2 为宿 IP 地址字段，F_3 为源端口字段，F_4 为宿端口字段，运算 O 为异或操作即：O=(^, F_1, F_2, F_3, F_4)，得到域 F。

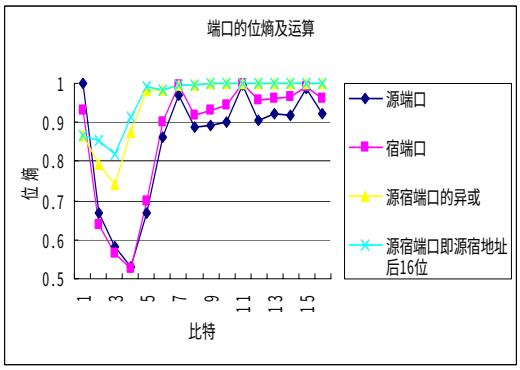

图 6 端口、端口运算、端口和地址运算的位熵
Fig. 6. Bit Entropy of Ports, XOR of Ports, Low 16bits XOR of IP

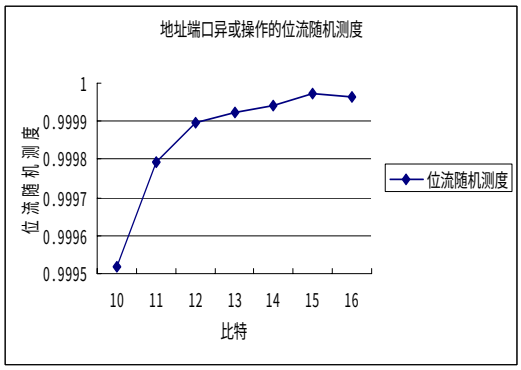

图 7 算法 4 运算后比特流的位流随机测度
Fig. 7. Bit flow randomicity metrics of Algorithm 4

算法 3 和 4 将端口字段引入，因此需要分析端口字段的位熵。端口的随机性在[4]中并未被分析。

图 6 是端口的位熵、源宿端口运算的位熵以及端口加地址运算后的位熵。可以发现，随着被引入的较好随机量的增加，生成出的随机量的随机性得到了改善。

计算第 16 比特向前倒推的各比特流的位流随机测度，可以得到如图 7 所示的结果。由图 7 可知，随着使用的比特数的增加，位流随机测度会下降，但仍旧在 0.9995 以上。这说明增加处理节点的个数到一定程度，宏观平衡性会略微下降。

在以上算法中，算法 2 能够保证任何相同源、宿 IP 地址的会话被分配到同一处理机中，而算法 4 可以保障同一个会话被负载均衡到同一处理机上，并且以上算法的平均性很好。因此该算法不但可以应用于象高速入侵检测系统以及其它具有类似需求的应用场合。然而算法 3 很难想象有何实际用途。

5 域分类算法的微观平衡度分析

好的位流熵特性保证了负载均衡后的各个处理节点的负载在宏观上(也就是大尺度时间范围内)保持平衡，但是并不能保证各个处理节点的负载在微观上(小尺度时间范围内)保持平衡。由于有面向会话的限制，域分类算法的微观平衡度不能简单使用并行处理中常用的各种指标，例如 Machine balance 和各种基准等[7]。参考[3]中针对 Web 服务器所定义的平衡度指标，本文定义了两种报文处理中的平衡性指标。

- $load_{i,j}$：表示第 i 台处理机(共有 n 台处理机)在第 j 个采样点(共有 m 个采样点)上的负载；
- $peak\_load_j$：表示在第 j 个采样点上，负载最大的处理机的负载；
- peak_to_mean(峰值到均值比)：$\frac{peak\_load_j}{(\sum_{i=1}^{n} load_{i,j})/n}$

LBM(Load Balance Metric)：

$$\frac{\sum_{j=1}^{m}\left(\frac{peak\_load_j}{(\sum_{i=1}^{n} load_{i,j})/n}\times\frac{\sum_{i=1}^{n} load_{i,j}}{n}\right)}{\sum_{j=1}^{m}\sum_{i=1}^{n} load_{i,j}/n}=\frac{\sum_{j=1}^{m} peak\_load_j}{(\sum_{j=1}^{m}\sum_{i=1}^{n} load_{i,j})/n}$$ **公式 3**

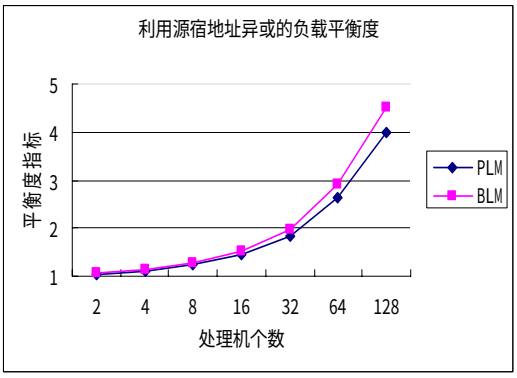

图 8 算法 2 的负载平衡度
Fig. 8. Load balance status of Algorithm 2

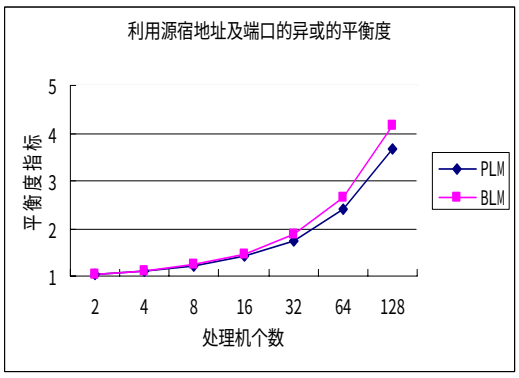

图 9 算法 4 的负载平衡度
Fig. 9. Load balance status of Algorithm

本文和[3]中的不同主要体现在 $load_{i,j}$ 的定义上：本文分别使用 $pps_{i,j}$(packet per second)和 $bps_{i,j}$(bits per sencond)表示对应的 $load_{i,j}$。因此，出现两种不同的 LBM，分别命名为 PLM(pps Load Balance Metric)和 BLM(bps Load Balance Metric)。这两个指标分别反映了主机在报文处

理机个数和比特处理个数上平衡性的差异。

对应于第 4 节中位流熵较好的选择域及运算，分别计算 PLM 和 BLM。其中对于算法 2，可以得到如图 8 的 PLM 和 BLM 曲线。

可以看到，随着采用比特数目的增加，处理节点个数增多，微观平衡性被很大程度的削弱了。

如果采用算法 4，得到的 PLM 和 BLM 则如图 9 所示。

对应于上节中关于位流随机测度和位熵的计算，可以发现平衡性比算法 2 有了一定的改善。这再次说明了引入新的、位熵较好的随机量，对域分类负载均衡算法有改善作用。

6 域分类算法和传统负载均衡算法的比较

本文提出的域分类算法用于把报文分配到不同的处理节点上，因此非常类似于集群服务中所采用的算法，而和并行计算中的负载均衡算法讨论的则不是相似的问题。

本算法针对高速信道的大流量报文处理任务，每一级处理节点都允许从单一处理节点变成处理机集群，而任务的到达速度是一般的服务器集群所无法比拟的，许多情况下根本没有时间由通信决定负载均衡的调度，因此内部通信量的多少也成为衡量报文负载均衡算法的重要指标。例如，High Availability 等算法，虽然内部通信也比较少，可以适用于 Web 服务或 FTP 服务等环境，但是对于报文负载均衡而言则无法满足要求。

保持报文相关度也是域分类算法的一个重要特性，这是象入侵检测这样的应用所必须要求的功能。

表 2 通过宏观平衡度、微观平衡度、内部通信量和报文相关度保证四个指标，对各种服务请求分配或报文分配类的负载均衡算法进行了比较。

表 2 服务请求分配或报文分配负载均衡算法比较

Table 2. Comparisons of load balance algorithms

算 法	算 法 说 明	宏观平衡度	微观平衡度	通信量	相关度保证
Random	采用均匀随机概率分配负载	好	取决于随机算法	无	无
Round-robin	循环分配负载	好	好	无	无
Weighted round-robin	处理机不同负载额度下，循环分配	可用于异构环境 不一定要保证	不需要保证	无	无
High availability	每一主机达到负载额度后，再分配到下一处理机	好	较差	较少	无
域分类	利用报文域进行分类，仅用于报文负载均衡	好 (取决于域和域上的运算)	较好	无	可保证

可以看到，只有域分类算法能够满足报文相关性的保证，其它算法虽然在有的方面强于域分类算法，但是却无法应用于高速入侵检测以及其它需要维持报文相关性的应用场合。而且域分类算法的微观平衡度也仅次于 Round-robin 算法。

7 结论

为了满足高速入侵检测以及其它需要维持报文相关性的应用的处理能力分布的需要，本文提出了基于 IP 报文头域分类的负载均衡算法。利用位熵和位流熵的计算寻找有潜力的 IP

报文头域和基于这些域的运算，本文证明了被选出的特定的负载平衡算法实例具有良好的平衡性，虽然比 round-robin 等传统负载均衡算法的平衡性略差，但是能够提供良好的相关性支持，并且不需要处理节点之间的内部通信。

本文提出的算法不仅可以用于高速入侵检测系统，同时可以满足各种需要基于报文相关性分析的场合，例如网络行为分析中的流生成、分析和处理等应用场合。

参考文献

[1] Y. F. Hu, R. J. Blake, “Load Balancing for Unstructured Mesh Applications”, Parallel and Distributed Computing Practices, Vol. 2, No. 3, September 1999

[2] “Growing Your E-Business with IBM Server Load Balancing and Caching embedded solutions”, IBM White Paper. http://www.networking.ibm.com/white/serverload.html

[3] Richard B. Bunt, Derek L. Eager, Gregory M. Oster, and Carey L. Williamson, “Achieving Load Balance and Effective Caching in Clustered Web Servers”, Proceedings of the forth International Web Caching Workshop, San Diego, California, April 1999, 159-169

[4] Guang Cheng,Jian Gong,Wei Ding,Network Traffic Sampling Measurement Model on Packet Identification,ACTA ELECTRONICA SINICA Vol.30 No.12A：89－93 Dec.,2002.(in Chinese)

[5] Sundar Lyer, Ramana Rao Kompella, Ajit Shelat, “ClassiPI: An Architecture for Fast and Flexible Packet Classification”, IEEE Network, March/April 2001, pp.33-41

[6] Pankaj Gupta, NIck McKeown, “Algorithms for Packet Classification”, IEEE Network, March/April 2001, pp.24.

[7] Kai Hwang, Zhiwei Xu, “Scalable Parallel Computing: Technology, Architecture, Programming”, China Machine Press, ISBN 7-111-07176-X, pp.91, May. 1999(in Chinese)

Session-Oriented Fast Load Balancing Algorithm

GONG Jian, LU Sheng, RUI Suying

Department of Computer Science and Engineering, Southeast University

Abstract: Load balancing is widely used in parallel computing and cluster computing environments. With the development of high-speed network-based Intrusion Detection System (IDS) load balancing is fetched into this area to help IDS work more efficiently. A load balancing algorithm named Dimension-based Classification Algorithm is introduced in this paper, which is designed for High-speed network applications such as IDS. This algorithm has a fairly good load balancing performance in both macroscopical and microscopical scopes. Above all, communications between process nodes are redundant while using this algorithm and it can keep the relativities among packets, which is necessary in many occasions.

Keywords: Intrusion Detection, Load Balance, Network, High Speed, Wide Bandwidth, Packet Classification, Bit Entropy, Flow Entropy