基于LEAST的高速网络大流检测算法

徐 敏，夏靖波，申 健，陈 珍
(空军工程大学信息与导航学院，西安，710077)

摘要 针对大流漏检率过高，占用SRAM过大问题，提出了基于最少(LEAST)改进型大流检测算法。主要思想：利用LEAST淘汰机制将小流丢弃使得大流能够被保护，采用窗口-储备策略解决检测大流的公平性问题。通过相关组织所提供的实际互联网数据进行了实验比较，结果显示：与现有算法相比，新算法具有更高的测量准确性，平均大流漏检率降低至0%～0.13%。

关键词 网络测量；大流流量；LEAST淘汰机制；窗口-储备策略
DOI 10.3969/j.issn.1009-3516.2015.04.015
中图分类号 TP393 **文献标志码** A **文章编号** 1009-3516(2015)04-0062-04

An Elephant Flow Identifying and Measuring Algorithm Based on LEAST in High-speed Network Environment

XU Min，XIA Jingbo，SHEN Jian，CHEN Zhen
(Information and Navigation College，Air Force Engineering University，Xi′an 710077，China)

Abstract：In high-speed network environment，it′s very important to extract elephant flow timely and accurately for cognizing behavior and law of network. In order to reduce the elephant flow measurement missing rate and overmuch occupation of SRAM，an improved algorithm based on LEAST is proposed. By using LEAST elimination mechanism for discarding the mice flow，the elephant flow can be protected. And Window-Reserve strategy is adopted to ensure the fairness of identifying and measuring elephant flow. Finally，through the comparison between the simulation results and the actual flow data，the result shows that the new algorithm has a higher measurement accuracy and is more practicable，and the elephant flow on the average measurement missing rate is reduced to 0%～0.13%.

Key words：network measuring；elephant flow；LEAST elimination mechanism；window-reserve strategy

Estan及Varghese首先把大流检测问题引入网络测量领域，并给出“Sample and hold”[1-2]和“Multistage filters”[3]。前者实现简单但误差较高，后者具有较高的误判率(将小流误判为大流)，另外存在消耗大量空间、在实际中难以实现等问题。IETF推荐的流量测量理念是在路由器中创建并维护一组流表，将需要的流记录保存于此。Kim等人[4]再将LRU页面置换算法引入流表中检测大流，该算法实现简单，大流漏检率高。文献[5]和[6]均提出采用两级结构来检测大流，第1级都采用LRU结构进行大流预保护，不同的在于第2级，前者采用最少(LEAST)淘汰策略，后者仍使用LRU淘汰策略，比较结论得LEAST可以在有限的静态随机存储器(SRAM)更快地处理流量信息，具有很好的可扩展性。文献[7]、[8]表明，大流持续时间长且分组到达速率高，进而以较大的概率留在流记录表项中，所以可以只采用LEAST淘汰机制。

收稿日期：2015-03-04
基金项目：陕西省自然科学基金资助项目(2012JZ8005)
作者简介：徐 敏(1990－)，女，江苏盐城人，硕士生，主要从事网络流量测量研究.E-mail：376748496@qq.com

引用格式：*徐敏，夏靖波，申健，等. 基于LEAST的高速网络大流检测算法[J]. 空军工程大学学报：自然科学版，2015，16(4)：62-65. XU Min，XIA Jingbo，SHEN Jian，et al.An Elephant Flow Identifying and Measuring Algorithm Based on LEAST in High-speed Network Environment[J]. Journal of Air Force Engineering University：Natural Science Edition，2015，16(4)：62-65.*

1　网络流量分析

本文实验所用相关流量数据来自实际的互联网骨干链路中，由 MAWI 工作组和 CAIDA 组织所提供。文献[9]已证明：无论是在 OC-12 链路（带宽为 0.622 Gbit/s 传输链路）还是 OC-48 链路（带宽为 2.5 Gbit/s 传输链路）中，大流流量大约占据了链路总流量的 90%，即流量分布呈重尾分布，流量分布表明少量的大流产生大部分的流量，在实际网络传输中，这就要求大量的内存空间存储大流，为了降低大流的漏检率需将尽量多的内存空间分配给存储大流的表项数，再结合 LEAST 淘汰策略将最小流丢弃，腾出新的空间，基于此本文将配置最佳的 LEAST 列表用来检测大流。

2　算法检测机制

2.1　基本思想

当数据分组到达时，先判断是否在 LEAST 表项内有对应的流记录，若有所属表项便将 LEAST 对应表项中的字节数加上此分组的大小；否则，如果 LEAST 表项未满，重新建立一个表项记录该数据分组；若 LEAST 表项已满，淘汰 LEAST 表项数值最小的那项。最后，在测量结束时，LEAST 表中符合大流定义的流将被读出并保存。

2.2　性能分析

文献[5]表明：高速网络中随着流量的增加，数据分组到达间隔呈现泊松分布。分组归并流越多，泊松特性越明显。本文研究的对象就是单位时间内到达的数据分组。当 LEAST 表项数已经占用满并且又有新的数据分组要进入时，表中累积字节数最小的那条流记录将被丢弃出去，具体过程见图 1。

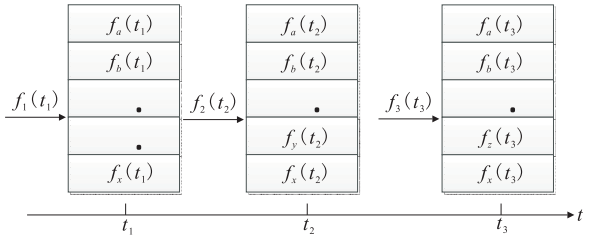

图 1　LEAST 策略的淘汰具体过程

Fig.1　The selection process of LEAST

其中，$f_1(t_1)$、$f_2(t_2)$、$f_3(t_3)$ 表示分别在时刻 t_1、t_2、t_3 对应有新流到达；$f_x(t_1)$、$f_y(t_2)$、$f_z(t_3)$ 表示分别在 t_1、t_2、t_3 时刻 LEAST 表中最小流字节数的流记录；$f_1(t_2)$、$f_2(t_3)$ 表示分别上一时刻到达的新流在下一时刻的流记录。

当 LEAST 列表已满且有新流到达，以 3 个时刻简化过程：

步骤 1　假设时刻 t_1 有新流 f_1 到来时，根据 LEAST 的淘汰规则 $f_x(t_1)$ 将被丢弃，同时新流 f_1 替换出 $f_x(t_1)$ 所占的表项；

步骤 2　假设时刻 t_2 又有新流 f_2 到来时，首先要选择丢弃哪条流的表项，实质在于比较 $f_1(t_2)$ 与 $f_y(t_2)$，如果前者小于后者，则 $f_1(t_2)$ 被丢弃，反之 $f_y(t_2)$ 被丢弃，同时新流 f_2 替换出 LEAST 列表中最小流所占的表项，同时由淘汰机制易得上一时刻 t_1 的最小流记录比下一时刻 t_2 的最小流记录要小即 $f_y(t_2) \geqslant f_x(t_1)$；

步骤 3　假设时刻 t_3 又有新流 f_3 到来时，同理如果 $f_2(t_3) < f_z(t_3)$，则 $f_2(t_3)$ 被丢弃，反之 $f_z(t_3)$ 被丢弃，同时新流 f_3 替换出列表中最小的表项，并且易得 $f_z(t_3) \geqslant f_y(t_2)$。

因为 $f_y(t_2) \geqslant f_x(t_1)$，且 $f_z(t_3) \geqslant f_y(t_2)$，即 $f_z(t_3) \geqslant f_x(t_1)$，流 f_1 可能被丢弃的阈值小于等于流 f_2 可能被丢弃的阈值，以此类推，后归并的大流与先归并的大流留在列表被检测的几率方面，起始时就注定是不均衡的。

2.3　优化策略

针对上述问题，为充分利用系统分配的存储空间，引入窗口-储备函数改善 LEAST 淘汰规则。

定义 1　时间顺延窗口是指将测量时间 t 按顺序划分为 n 个等间距的测量时间段。每一个窗口的长度为 $l = t/n$。

定义 2　对第 i 个不同的时间窗口累积设置不同的 LEAST 表的最大表项 $L(i)$，累积因子 $r(i)$ 表示前一窗口的 $L(i-1)$ 比当前窗口的 $L(i)$ 多储备出的表项，通过依次给每个窗口储备一定 LEAST 表项的方法降低对后到大流的不均衡性，其中的累积因子 $r(i)$ 称为储备函数，具有以下特征：

1) $\sum_{i=1}^{n} r(i) = L$，L 为常数，即给定的存储空间配置 LEAST 的最大表项数；

2) $r(i-1) > r(i)$，其中 $2 \leqslant i \leqslant n$，假设 $r(i-1) = a^i r(i)$，其中，$a > 1$；

3) $L(i) = \sum_{j=1}^{i} r(i)$，$L(i)$ 为表的最大长度。

根据以上特征计算 $L(i)$，$r(i)$：

由 $r(i-1) = a^i r(i)$，知 $r(i) = a^{-i} r(i-1)$，故 $r(i) = a^{-(i+2)(i-1)/2} r(1)$，$i \geqslant 2$。$\sum_{i=1}^{n} r(i) = L$ 得：

$$r(1) + \sum_{i=2}^{n} r(i) = r(1) + \sum_{i=2}^{n} a^{-(i+2)(i-1)/2} r(1) = L$$

即：$r(1)=L/(1+\sum_{i=2}^{n}a^{-(i+2)(i-1)/2})$，

$$r(i)=a^{-(i+2)(i-1)/2}r(1)=\frac{a^{-(i+2)(i-1)/2}L}{1+\sum_{i=2}^{n}a^{-(i+2)(i-1)/2}},$$

令 $A=a^{-(i+2)(i-1)/2}$，$B=\sum_{i=2}^{n}a^{-(i+2)(i-1)/2}$，简化为：

$$r(i)=AL/1+B \tag{1}$$

由特征 3)得 LEAST 表的最大长度 $L(i)$：

$$L(i)=\sum_{j=1}^{i}r(i)=\sum_{j=1}^{i}\frac{AL}{1+B} \tag{2}$$

综合式(1)、(2)可以得出不同内存空间下的储备 LEAST 表项数分布，见图 2。

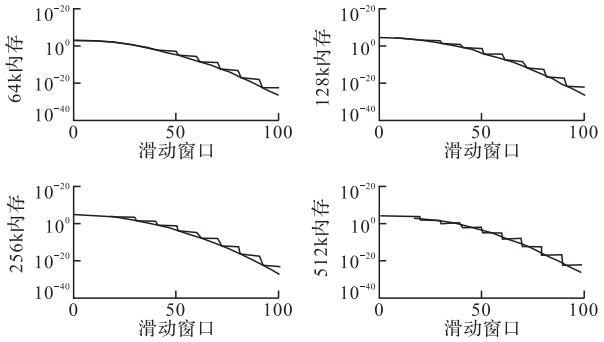

图 2 不同内存空间下的储备 LEAST 表项数分布

Fig.2 List distribution in different memory space

图 2 是每隔 10 s 设置 1 个窗口，不同窗口储备不同数量的表项数，可以看出不同内存空间下的储备 LEAST 表项数分布趋势是相同的，实验部分需要的储备 LEAST 表项数就按此分配。

2.4 算法复杂度分析

LEAST 的表项实质是记录流的 2 个部分：存储流关键字需要 104 bit(采用五元组)，存储流字节数需要 32 bit，以及为了加快访问速度，设置双向链路指针 64 bit，因此，一个表项所占空间 I 为 200 bit，即 25 Byte。

时间复杂度：当 1 个数据分组进入 LEAST 列表缓存时，首先查询是否存在相应的流记录，本文算法采用一维 Hash 映射其计算复杂度为 $O(1)$。为了加快访问速度，设置双向链路指针，流记录在此双向链路是有限指针操作，其计算复杂度为 $O(1)$。取两者最大的时间复杂度即为一个数据分组的时间复杂度 $O(1)$。

空间复杂度：假设系统空间存储容量为 C，因为所有的空间都是用来存储 LEAST 表项数，故 $C=LI$，假设每个大流所占链路总流量的比率为 Q，所以本文的空间复杂度 $O(1/Q)$。

内存访问次数：将算法应用到具体实现方面，内存访问次数是衡量这一类算法性能的关键指标。具体的访问存储器的次数情况如表 1。当一个数据分组进入 LEAST 列表缓存时，首先查询是否存在相应的流记录：情况 1，如果不存在相应的流记录，重建新表项；情况 2，否则，对相应的流记录进行修改。

表 1 不同情况下的内存访问次数

Tab.1 Number of memory access

情况类型	查找记录	重建流记录	修改流记录	修改第一个指针	修改指针	总计访问次数
情况 1	1	1	*	2	*	4
情况 2	1	*	4	*	1	6

注 * 表示情况类型中不存在的访问次数。

目前 SRAM 的访问速度为 2～5 ns，若选择 5 ns 的 SRAM，算法处理 1 个数据分组最多 30 ns，实际 OC-192(10 GB)链路处理时隙为 32 ns。算法处理速度完全满足 OC-192 链路的要求。

3 实验与分析

将总测量时间 100 s 分为 H 段，每段 10 s 为 1 组数据，OC-12 链路与 OC-48 链路上分别采 10 组用 Matlab 进行仿真。此次分别选用一级结构的 LRU、文献[5]提出的两级结构 LLR^{+} 以及文献[10]提出的两级改进结构 LEAST-CBF 检测大流算法与本文提出的 $LEAST^{*}$ 算法作比较。

用 Matlab 对实际流量进行仿真，通过评价指标平均大流漏检率 δ、大流流量平均误差 ε 衡量算法的准确性：

平均大流漏检率 δ：

$$\delta=\frac{1}{H}\sum_{h=1}^{H}\frac{Q_R-Q_M}{Q_R} \tag{3}$$

式中：Q_R 为在第 h 组数据内真实大流的个数；Q_M 为在第 h 组数据内使用该算法检测出大流的个数。

第 h 组数据内相对误差率 ε_h：

$$\varepsilon_h=\sum_{i=1}^{Q_h}\frac{|B_i-B_i'|}{Q_h\times B_i} \tag{4}$$

式中：B_i 为第 h 组数据内每个真实大流对应的字节数；B_i' 为第 h 组数据内使用该算法检测每个大流对应的字节数。则整个测量时间平均相对误差 ε：

$$\varepsilon=\frac{1}{H}\sum_{h=1}^{H}\varepsilon_h \tag{5}$$

为保证实验结果可靠性，每组缓存容量各算法均测试 5 次，取平均值作为实验结果，见表 2。

首先与一级结构 LRU 比较，当有大量小流突发到来时，LRU 算法会导致原本为大流流量被替换出去，实验结果也表明相同的情况和配置下 LRU 存在较大误差，漏检率高。

给定如参数配置：

1)LLR^{+} 算法中 LRU 的表项数、LEAST 的表项数进行最优配置：起始时刻 LRU 的表项数最大，在算法执行过程中 LEAST 的表项数逐渐增加，直

至 2 个表项数达到平衡为最佳。

2）LEAST-CBF 算法中 LEAST 的表项数、CBF 的表项数进行最优配置：CBF 的表项数 m 是与哈希函数 k 的个数，当前已经被映射入 CBF 的流的个数 n，以及所要求达到 CBF 的误正率有关即 $P=(1-(1-1/m)^{kn})^k$，确定最优 m 值，LEAST 的表项数 L 再由 $C=L\times 25\ \text{Byte}+m\times 8\ \text{Byte}$ 确定。

比较算法 LLR$^+$ 与 LEAST-CBF 算法，2 种算法都具有 LEAST 机制，从表 2 中可以看出，后者效果明显要优于前者，这说明根据持续时间长短和字节数来检测大流具有统一性，因为采用 CBF 淘汰小流较采用 LRU 的所消耗内存少，从而有更多的空间可用于下一级过滤。纵向比较实验结果在增加缓存容量虽然有一定的效果，但无限增加时趋于饱和。当所用缓存容量都一样时，可以发现，本文所提的算法从大流的漏检率和大流流量平均误差要明显优于 LEAST-CBF 算法，这是因为单纯地用第二层来淘汰小流是不必要的，而且占据了本该用来检测和存储大流的空间，造成了误差，本文提算法采用几乎整个空间都用来存储 LEAST 表项数的策略，较 LEAST-CBF 有更多的空间分配给检测机制，高效地降低大流的漏检率，提高了准确性和实用性。

表 2　各算法平均大流漏检率平均相对误差比较

Tab.2　The comparison of all kind algorithm　　%

实验数据	缓存容量	LRU		LLR$^+$		LEAST-CBF		LEAST*	
		δ	ε	δ	ε	δ	ε	δ	ε
OC-12	64 kB	73.17	22.89	4.85	0.69	0.83	9.55×10^{-2}	0.13	4.65×10^{-2}
	128 kB	55.60	11.11	3.78	0.13	0.88	9.48×10^{-2}	0.00	3.49×10^{-2}
OC-48	256 kB	45.81	5.66	6.08	0.64	0.24	7.00×10^{-2}	0.08	1.33×10^{-2}
	512 kB	7.54	0.46	2.36	0.21	0.24	6.85×10^{-2}	0.00	4.44×10^{-3}

4　结语

提高大流检测精度是高速网络流量检测和控制的基础，可以更好地了解网络运行情况。本文提出了基于 LEAST 改进型算法，利用窗口-储备机制解决前后到达的大流检测不公平的问题，结合 LEAST 将最少字节数淘汰机制的优点，使得该算法具有实现简单，消耗内存空间少，漏检率低，准确性高等特点。与同类算法相比，不仅参数设置方式比较简单，而且各项评价指标也明显更优。在高速网络环境中，尤其对于流量计费应用、网络攻击识别等应用而言，及时、准确地提取大流具有重要意义。

参考文献(References)：

[1] REN Wuyue, LI Ruiying, LI Meinan. The Applicability of Traditional Sampling Techniques in the Measurement of LAN Availability[C]//2012 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering(ICQR2MSE). Chengdu: IEEE, 2012: 83-88.

[2] Smitha A, Kim I, Reddy A L N. Identifying Long Term High Band Width Flows at A Router [C] //Proc of HiPC2001. Berlin: Springer, 2001: 361-371.

[3] RASPALL F. Efficient Packet Sampling for Accurate Traffic Measurements[J]. Computer Networks, 2012, 56(6): 1667-1684.

[4] 张震，汪斌强，陈庶樵.基于多维计数型布鲁姆过滤器的大流检测机制[J].电子与信息学报，2010，32(7)：1608-1613.
ZHANG Zhen, WANG Binqiang, CHEN Shuqiao. A Mechanism of Identifying Heavy Hitters Based on Multi-dimensional Counting Bloom Filter. Journal of Electronics & Information Technology, 2010, 32(7): 1608-1613. (in Chinese)

[5] 王风宇，云晓春，王晓峰.高速网络监控中大流量对象的提取[J].软件学报，2007，18(12)：3060-3070.
WANG Fengyu, YUN Xiaochun, WANG Xiaofeng. Identifying Heavy Hitters in High-Speed Network Monitoring[J]. Journal of Software, 2007, 18(12): 3060-3070. (in Chinese)

[6] 裴育杰，王洪波，程时端.基于两级 LRU 机制的大流检测算法[J].电子学报，2009，37(4)：685-691.
PEI Yujie, WANG Hongbo, CHENG Shiduan. A Dual-LRU Based Algorithm for Identifying and Mea-suring Large Flows [J]. Acta Electronica Sinica, 2009, 37(4): 685-691.(in Chinese)

[7] 赵小欢，夏靖波，付凯.基于散列和计数方法的网络流频繁项挖掘算法[J].华中科技大学学报：自然科学版，2013，41(9)：57-62.
ZHAO Xiaohuan, XIA Jingbo, FU Kai. Frequent Items Mining Algorithm over Network Flows Based on the Combination of Hash Method and Counting Method[J]. Journal of Hua Zhong University of Science and Technology: Natural Science Edition, 2013, 41(9): 57-62.(in Chinese)

[8] 张震，汪斌强，张风雨. 基于 LRU-BF 策略的网络流量测量算法[J].通信学报，2013，31(1)：111-120.
ZHANG Zhen, WANG Binqiang, ZHANG Fengyu. Traffic Measurement Algorithm Based on Least Recent Usd and Bloom Filter [J].Journal on Communications, 2013, 31(1): 111-120.(in Chinese)

[9] 孙昱，夏靖波，赵小欢.基于 LEAST 和 CBF 两级结构的大流检测算法[J]. 华中科技大学学报：自然科学版，2014，42(4)：40-44.
SUN Yu, XIA Jingbo, ZHAO Xiaohuan. A LEAST and CBF Two-Level Architecture Based Algorithm for Identifying and Measuring Large Flows[J].Journal of Huazhong University of Science and Technology: Natural Science Edition, 2014, 42(4): 40-44.(in Chinese)

（编辑：姚树峰）