基于自动机的TCP流识别算法

张孝国[1,2],丁 伟[1]
(1.东南大学计算机科学与工程学院,江苏南京 211189;2.河南科技大学信息工程学院,河南洛阳 471023)

摘 要: 为提升网络流识别性能,本文提出了一种TCP流识别算法.该算法基于传输控制协议(Transmission Control Protocol,TCP)下网络通信双方的交互过程构建双向流自动机,由该自动机根据TCP协议规则和网络流当前状态判断TCP流终止,同时以基于规则的过滤机制和超时策略为辅助措施,快速识别单包流和异常中断流.该算法内存开销、计算和内存总开销均低于经典算法固定超时策略(Fixed Timeout strategy,FT)和同类代表性算法两层自适应超时策略(Two-level Self-Adaptive Timeout,TSAT),同时该算法精度高于TSAT,且仅比默认精度标准略有下降.该算法基于协议规则识别TCP流,既保证了流的准确性,又节省了流的超时等待时间,而且算法尤其适合中流、小流和不规则TCP流比重较大的情况,使得识别系统在面临DDoS攻击、蠕虫爆发等网络异常时仍能正常运行.

关键词: 流识别;TCP;自动机;属性识别度;流超时
中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2017)06-1396-07
电子学报URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2017.06.017

TCP Flow Identifying Algorithm Based on Finite State Automaton

ZHANG Xiao-guo[1,2], DING Wei[1]
(1. *School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu* 211189, *China*;
2. *Information Engineering School, Henan University of Science and Technology, Luoyang, Henan* 471023, *China*)

Abstract: In order to improve flow-identifying performance, a flow-identifying algorithm for TCP (Transmission Control Protocol) traffic was proposed. This algorithm constructs bidirectional-flow finite state automaton based on TCP communication process and judges flow-termination according to TCP protocol rules and flow states by this automaton. Meanwhile, the algorithm adds filtering mechanism and timeout strategy to identify single-packet flows and abnormal interrupt flows. This algorithm is lower in memory overhead, the total overhead of memory and computing resources than the classic algorithm FT (Fixed Timeout strategy) and the similar representative algorithm TSAT (Two-level Self-Adaptive Timeout). Furthermore, this algorithm is higher than TSAT in accuracy and only loses little accuracy compared to the default accuracy standard. Our algorithm identifies TCP flows based on protocol rules, so it can obtain high identifying accuracy and can save extra flow keeping-time. And our algorithm is especially suitable for situations when the proportion of small flows, medium flows or irregular flows is larger, so it can ensure flow-identifying system to work normally when network anomalies occur, such as worm infection, DDoS attack, and so on.

Key words: flow identifying; TCP; finite state automaton; attribute recognition degree; flow timeout

1 引言

随着网络流技术在网络安全管理、性能管理、计费管理、流量分类、拓扑结构分析等众多领域的广泛应用,准确而高效的网络流识别算法显得尤为重要,在保证高精度的前提下,提高网络流识别效率一直是网络流技术研究的重要问题之一.流识别算法的核心是流终止策略,流终止策略按适用范围可分为两类:一类是通用型的网络流终止策略,即对所有的网络流量都适用的流终止策略;另一类是专用型的网络流终止策略,即策略仅适用于某种类型的网络流量.

通用型流终止策略主要有超时策略和强制终止策略,超时策略是目前研究最多的流终止策略,其基本思想是当流的不活跃时间超过一定阈值时认为流终止,

收稿日期:2015-10-12;修回日期:2016-04-07;责任编辑:覃怀银
基金项目:国家重点基础研究发展规划(973计划)项目(No.2009CB320505);国家科技攻关计划基金资助项目(No.2008BAH37B04)

其代表性算法有固定超时、自适应超时、概率保证的自适应超时、两层自适应超时以及动态超时策略等[1~5],强制终止策略是一种辅助性策略,其通常基于时间或空间强制终止流[6],这些策略在识别精度和效率上各有千秋,但至今仍未有达到两者之间理想平衡的公认算法[1~7],这些策略过分依赖于流速特性,忽略了流的其他特征,而相关研究表明不同的协议和应用类型对流超时阈值具有较大影响[4,5,7],因此,合理的网络流终止策略应具有协议或应用类型方面的针对性. 目前,只有一些针对 UDP (User Datagram Protocol) 流量的专用型终止策略,如多分类支持向量机策略[8]、区分更新超时策略[9]等. 然而,对于 TCP 流而言,目前仅有基于特殊 TCP 标志的流终止辅助策略,即当遇到 FIN 或 RST 标识的数据包时立即终止流[1~7],该辅助策略提高了 TCP 流识别效率,但存在以下问题:(1) TCP 流量中存在 20% 甚至更高比例的单包流[1,2,4,11],该策略通常无法识别. (2) FIN 包并不是 TCP 流的最后一个数据包,该策略会造成流截断.

在传输层 98% 左右的网络流量使用 TCP 和 UDP 协议,其中 TCP 流量在字节数和数据包数上均处于主导地位[10,11],同时,TCP 通信具有严格的规范,通信双方必须严格遵守这些规范才能进行有效交互,对于 TCP 流而言,TCP 连接释放时刻正是 TCP 流终止时刻,因此基于 TCP 协议规范判断 TCP 流终止不仅能够节省 TCP 流的额外保持时间,而且能够获得高的识别精度. 目前,对于 TCP 流而言除了最初提出的基于特殊 TCP 标志的流终止辅助策略之外,尚未出现新的专用型终止策略,因此探索新的针对 TCP 流量的流识别策略,进一步提升 TCP 流识别效率和精度,对网络流技术及其应用具有重要意义,本文在深入分析 TCP 流特性和现有流终止策略的基础上,提出了一种基于自动机的 TCP 流识别算法(TCP Flow Identifying Algorithm based on finite state automaton, TFIA),旨在保证高识别精度的前提下提升 TCP 流识别效率.

2　TCP 流识别属性选择

现有流识别策略大都基于数据包到达间隔识别流,然而这些策略并没有对该属性的识别能力进行分析,更未对其他流属性(如数据包的状态、大小、生存时间等)的识别能力进行分析比较,而流属性的识别能力对流识别算法的设计具有重要参考价值,因此本节将基于信息论的相关原理量化流属性的识别能力并寻找更加有效的 TCP 流识别属性.

定义 1　TCP 连接中一个数据包的 flag 字段值,称为该数据包的 TCP 状态.

定义 2　拥有相同五元组(源地址、目的地址、源端口、目的端口、协议)且满足一定终止约束条件的一系列数据包,称为一个流,当五元组中的协议为 TCP 时,该流为 TCP 流.

2.1　属性识别度

对于任意随机变量 X 和 Y,信息熵 $H(X)$ 表示获得 Y 取值之前 X 的不确定性,条件熵 $H(X|Y)$ 表示已知 Y 取值之后 X 的不确定性,则 $H(X)-H(X|Y)$ 必然表示由随机变量 Y 所提供的关于 X 的信息量,即 X 和 Y 之间的互信息 $I(X;Y)$:

$$I(X;Y)=\sum_{i=1}^{n}\sum_{j=1}^{m}P(x_i y_j)\log\frac{P(x_i|y_j)}{P(x_i)}=H(X)-H(X|Y) \tag{1}$$

$I(X;Y)$ 表示了随机变量 X 与 Y 之间的统计约束程度,若变量 X 与 Y 不相关,则 $I(X;Y)=0$,否则 $I(X;Y)>0$,且 $I(X;Y)$ 值越大,表明 X 与 Y 的相关性越强. 由于互信息只能反映不确定性的减少量,其大小受变量熵值影响,因此通常采用对称不确定性 (Symmetrical Uncertainty, SU) 进行归一化:

$$\mathrm{SU}(X;Y)=\mathrm{SU}(Y;X)=\frac{2I(X;Y)}{H(X)+H(Y)} \tag{2}$$

SU 将互信息标准化到 0~1 之间,值为 1 时表示根据一个变量的值完全可以预言另外一个变量的值,即完全相关,取值为 0,表示两个变量完全独立,归一化互信息的值越大,变量相关度越大,反之越小,当归一化互信息小于 0.1 时,通常认为两个变量没有必然联系.

定义 3　一个流属性所能提供的关于网络流终止状态的归一化互信息,称为该属性的识别度.

如果随机变量 X 表示某个流属性,如数据包的 TCP 状态、到达间隔等流属性,X 的取值为 $\{x_1,x_2,\cdots,x_n\}$,对应概率为 $\{P(x_1),P(x_2),\cdots,P(x_n)\}$,$Y$ 表示流终止状态(Y 有两个取值,分别为 1 和 0,1 表示流终止,0 表示流未终止),Y 的取值为 $\{y_1=1,y_2=0\}$,对应概率为 $\{P(y_1=1),P(y_2=0)\}$,那么 $I(Y;X)$ 归一化之后的值 $SU(Y;X)$ 即为属性 X 的识别度.

2.2　属性选择

本小节采用 64s 固定超时策略,基于实测 Trace 数据分别选择数据包的源端口、宿端口、大小、生存时间、服务类型、到达间隔和 TCP 状态为属性 X,根据 2.1 小节中的方法计算其识别度.

本文使用的 Trace 数据如表 1 所示,其中所有 Trace 均来源于 IPTAS[12],皆为以 1/4 流抽样采集于 CERNET 江苏省边界主干信道的实测 IP Trace 数据,该信道带宽 10Gbps,覆盖范围超过 100 个以上高等院校为主体的接入单位. 本文选择最近 5 年来的 8 条 IP Trace 分别计算各属性识别度,表 2 为基于 Trace 1~8 的属性识别度测量统计值,可见数据包到达间隔的识别度最高,TCP 状

态的识别度仅次之,此外数据包大小虽然对 TCP 流的识别有一定的贡献,但其识别度则很低,而生存时间、源端口、目的端口、服务类型等属性的识别度极低,可视为与流识别无关的属性.

表 1 Trace 数据集

编号	日期	开始时间	持续时间	TCP 流数	编号	日期	开始时间	持续时间	TCP 流数
1	09/02/2015	23:55:05	1 小时	135,434,567	9	03/12/2011	00:00:02	1 小时	33,922,537
2	11/08/2014	23:54:37	1 小时	58,127,755	10	01/16/2011	09:55:17	1 小时	23,188,424
3	05/08/2014	23:55:05	1 小时	118,917,859	11	11/14/2010	09:55:16	1 小时	36,631,177
4	01/21/2013	23:55:05	1 小时	22,579,919	12	09/11/2010	09:55:17	1 小时	17,478,313
5	09/20/2012	23:55:05	1 小时	33,434,012	13	05/18/2010	09:55:16	1 小时	19,759,581
6	04/25/2012	23:55:05	1 小时	32,500,885	14	03/28/2010	13:55:16	1 小时	14,926,245
7	03/19/2012	23:55:05	1 小时	44,742,398	15	02/23/2010	13:55:17	1 小时	6,797,344
8	04/17/2011	23:55:04	1 小时	19,177,374	16	12/17/2009	13:55:16	1 小时	14,745,839

目前,流识别策略基本上都是以数据包到达间隔为主进行流终止判断的,尚未出现以其他属性为主的流终止判断策略. 由于 TCP 交互需要遵守严格的协议规则,而这些协议规则实际上就是 TCP 状态转换规则,所以以 TCP 状态属性为主识别 TCP 流是合理的,但对于一些特殊情况,如通信意外终止时因无法接收到下一个数据包则无法基于 TCP 状态判断流终止,此时使用数据包到达间隔辅助识别是一种较好的选择,因此本文选择 TCP 状态属性为主,到达间隔属性为辅共同识别 TCP 流.

表 2 基于 Trace 1 ~8 的属性识别度

流属性	到达间隔	TCP 状态	大小	生存时间	源端口	宿端口	服务类型
识别度均值	0.3610	0.3150	0.1067	0.0216	0.0074	0.0062	0.0026
识别度标准差	0.0083	0.0214	0.0091	0.0101	0.0079	0.0058	0.0016

3 TFIA 识别算法

TCP 通信双方都必须遵守 TCP 协议定义的状态变迁规则,通信过程中任何一个状态变迁非法时,通信双方对应的网络流应该立刻终止,而正常情况下 TCP 交互双方中任何一方的结束时刻正是其对应 TCP 流的终止时刻,可见基于 TCP 状态变迁规则识别 TCP 流,不但能够在最佳时刻结束 TCP 流,而且能够保证 TCP 流的完整性. 因此,本文采用 TCP 状态识别 TCP 流,核心思想是基于 TCP 双向流自动机在 TCP 交互过程中的所有状态变迁时刻都进行 TCP 流终止判断,不局限于个别特殊状态或个别状态变迁,是一种以 TCP 状态识别为主的完整的 TCP 流识别策略.

3.1 TCP 双向流自动机

自动机是表示有限个状态以及这些状态之间的转移和动作等行为的数学模型,自动机的当前状态概括了过去输入处理的结果,系统只需根据当前所处的状态和面临的输入即可决定系统后续的行为. 本节将根据 TCP 双向流的特征构造出一个完整的 TCP 双向流自动机.

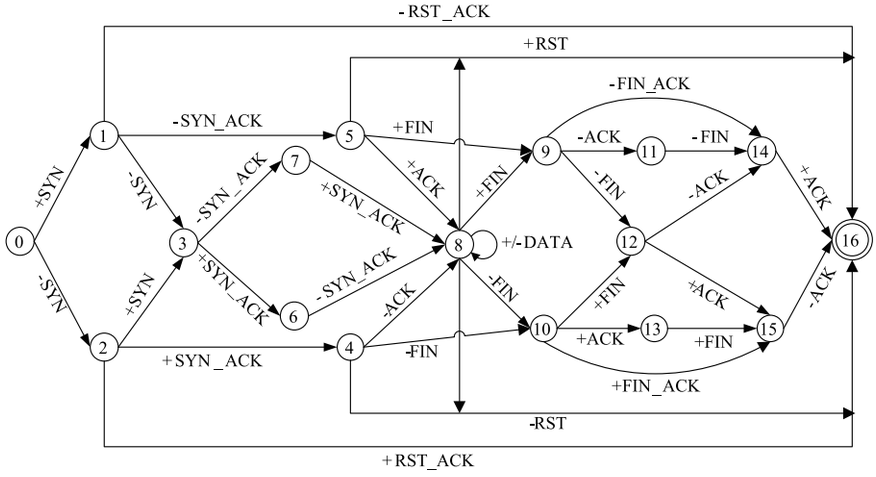


图1 完整TCP连接状态转换图

定义 4　如果一个 TCP 流的源 IP 地址小于目的 IP 地址,称该流为正向流,流中所有数据包的方向为正向,用“+”标识.

定义 5　如果一个 TCP 流的源 IP 地址大于目的 IP 地址,称该流为负向流,流中所有数据包的方向为负向,用“-”标识.

定义 6　一对终端 A、B 之间一个 TCP 连接过程中产生的一个正向流与一个负向流的集合,称为一个 TCP 双向流.

一个正常 TCP 连接的生命周期可分为连接建立、数据传输和连接终止三个阶段,当网络故障、主机不可达、软件异常等事件发生时,连接将不再经历所有阶段. 最常见的连接异常终止有以下 3 种:① 发起连接的 SYN 直接引发 RST_ACK 回应;② 接收端收到 SYN 后回应 SYN_ACK,接着又收到 RST 响应;③ 正在传输数据的连接突然被重置. 考虑到连接异常所增加的状态转换,则一个完整的 TCP 双向流自动机可构造为 $M=(Q,\Sigma,\delta,s_0,F)$,其中 $Q=\{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16\}$,$\Sigma=\{+SYN,-SYN,+RST\_ACK,-RST\_ACK,+SYN\_ACK,-SYN\_ACK,+ACK,-ACK,+RST,-RST,+FIN,-FIN,+FIN\_ACK,-FIN\_ACK,+DATA,-DATA\}$,$s_0=0$,$F=\{16\}$,$\delta$ 为状态转换函数,其对应的状态转换如图 1 所示.

3.2　TFIA 算法

定义 7　某一时刻一个 TCP 双向流对应的 TCP 双向流自动机的状态,称为此时该 TCP 双向流的状态.

公理 1　某一时刻一个 TCP 双向流接收到一个数据包之后,如果能够按照双向流自动机由当前状态转换到下一个状态,那么该状态转换合法,否则该状态转换非法.

TCP 交互双方都必须遵守 TCP 协议规定,任何一方违背,都将导致整个交互无法进行下去,这时交互进入非法状态,对应的流应该立刻终止,当 TCP 连接正常终止时,其对应的流也应立刻终止,本文 TFIA 算法正是基于该思路进行设计的.

TFIA 算法基于 TCP 双向流自动机识别 TCP 流,自动机中的每一条边都为一个输入,除 DATA 之外,所有输入都是数据包的 TCP 状态值,这里的 TCP 状态值是指当前新接收数据包的 TCP flag 字段的 SYN、FIN、RST 和 ACK 标志位的值,而输入 DATA 是指当前接收的数据包是携带载荷数据的数据包,另外 SYN、FIN、RST 为关键 TCP 标志,具有高优先级,当收到数据包时首先检查关键 TCP 标志位,按关键 TCP 标志进行状态转换,若关键 TCP 标志位未置位,再查看数据包是否携带载荷数据. 正常情况下,TFIA 生成的流都是双向流,与其他流识别算法相比,本文算法在流识别过程中需要为每条双向流增加一个状态,状态取值范围为 0 ~ 16,即双向流自动机的 17 种状态,状态值可保存在流记录某些字段的闲置位中,从而不增加额外内存开销. 考虑到不完整 TCP 连接的存在,需要设置流状态保持时间阈值,超过阈值仍未合法状态转换则认为双向流终止,由于在连接建立阶段流状态保持时间一般很短,而连接建立成功之后,流状态保持时间则一般较大,因此 TFIA 在连接建立阶段和连接成功建立之后分别使用不同的超时阈值 T_1 和 T_2,$T_1<T_2$,本文基于 Trace 1 ~ 8 及 64s 固定超时策略给出 T_1 的两种经验值 $T_1=T_{SYN40}=2s$ 和 $T_1=T_{SYN}=16s$,其中 T_{SYN40} 用于大小为 40 字节的 SYN 数据包开始的连接,T_{SYN} 用于其他 SYN 数据包开始的连接,连接成功建立之后则使用正常的流超时阈值 $T_2=64s$. 同时,为尽可能节省存储空间,在连接建立阶段,设立筛选区,用于暂存 SYN 包所对应的单向流,凡在时间阈值 T_1 内未能合法状态转换者,则识别为单包流,立即终止并从内存中导出,合法转换者则移入双向流空间. TFIA 工作流程如下:

(1)接收 TCP 数据包,获取相关信息,判断该数据包是否为 SYN 包,如果是 SYN 包,在筛选区创建其对应的单向流,进入步骤(1);否则查询该数据包对应双向流是否存在,若对应双向流存在,进入步骤(2),否则进入步骤(3).

(2)根据双向流状态和该数据包对应的状态输入值,按照双向流自动机进行状态转换. 如果状态转换合法,则更新双向流信息,进入步骤(4);否则进行双向流终止处理:添加数据包信息到双向流,然后导出双向流对应的正向流和负向流,进入步骤(1).

(3)查询该数据包对应单向流是否存在,若对应单向流存在,进入步骤(5);否则由数据包信息创建单向流并导出,进入步骤(1).

(4)判断双向流状态是否为终态,如果不是,进入步骤(1);否则导出该双向流对应的正向流和负向流,进入步骤(1).

(5)根据单向流状态和该数据包对应的状态输入值,按照双向流自动机进行状态转换. 如果状态转换合法,修改单向流为双向流,插入双向流空间,进入步骤(4);否则进行单向流终止处理:导出该数据包对应单向流,同时由数据包信息创建单向流并导出,进入步骤(1).

与上述步骤同时,需定时扫描筛选区和双向流空间,导出超过时间阈值仍未合法状态转换的 TCP 流.

4　TFIA 性能评价

衡量一个流识别算法精度时,默认以经典算法 64s 固定超时策略的流识别结果为标准,而且评价一个算

法时空效率时通常首先与经典算法 FT 对比,获得相对于基准的效率提升值,然后再与同类代表性算法比较,目前兼顾效率与精度的代表性流识别算法中 TSAT 的 TCP 流识别效率最高[4,5],因此,本文选择 FT 和 TSAT 作为算法性能比较的 2 个基础算法.

4.1 时空代价分析与对比

TFIA 算法的代价指它创建并维护一个流的平均计算资源和内存资源,本文基于 TFIA 计算资源和内存资源开销的上限 C_{MAX}、M_{MAX}和下限 C_{MIN}、M_{MIN}构建量化模型,TFIA 创建并维护一个流的平均计算资源(标记为 C_{TFIA})和平均内存资源(标记为 M_{TFIA})分别表示为式(3)和(4),同时 FT 算法创建并维护一个流的平均计算资源(标记为 C_{FT})和平均内存资源(标记为 M_{FT})分别表示为式(7)和(8),TSAT 创建并维护一个流的平均计算资源(标记为 C_{TSAT})和平均内存资源(标记为 M_{TSAT})分别表示为式(9)和(10),其中主要参数含义见表 3 所示. 流识别算法分析表明 $C_{FC}=3.625C_{FS}$, $C_{SC}\approx C_{FC}$, $C_{SS}\approx C_{FS}$, $M_S\approx M_F$,选择 $T_{SYN}=16\text{s}$, $T_{SYN40}=2\text{s}$, $T_F=64\text{s}$, $\alpha=1\text{Hz}$,则基于 Trace 1 ~ 16 可获得 TFIA 相对于 FT 与 TSAT 的计算资源和内存资源减少幅度,计算结果分别如图 2 和图 3 所示.

$$C_{TFIA}=\mu[P_{SYN}C_{SYN}+(1-P_{SYN})C_{FC}]+(1-\mu)[C_{FC}+\alpha C_{FS}(T_{DR}+\beta T_F)+C_{DFA}] \quad (3)$$

$$M_{TFIA}=\mu P_{SYN}M_{SYN}+(1-\mu)M_F(T_{DR}+\beta T_F) \quad (4)$$

$$C_{SYN}=P_{SYN40}(C_{FC}+\alpha C_{FS}T_{SYN40})+(1-P_{SYN40})(C_{FC}+\alpha C_{FS}T_{SYN}) \quad (5)$$

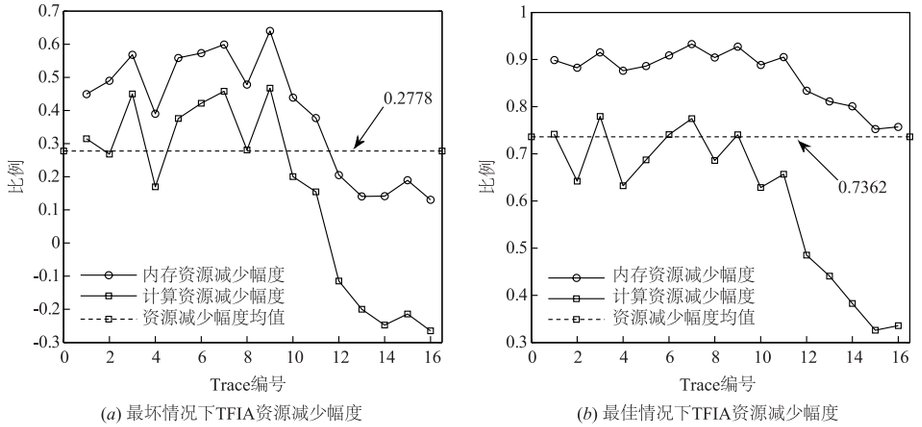
$$M_{SYN}=P_{SYN40}M_FT_{SYN40}+(1-P_{SYN40})M_FT_{SYN} \quad (6)$$

$$C_{FT}=\mu(C_{FC}+\alpha C_{FS}T_F)+(1-\mu)[C_{FC}+\alpha C_{FS}(T_{DR}+T_F)] \quad (7)$$

$$M_{FT}=\mu M_FT_F+(1-\mu)M_F(T_{DR}+T_F) \quad (8)$$

$$C_{TSAT}=\mu(C_{SC}+16\alpha C_{SS})+(1-\mu)[C_{SC}+C_{FC}+\alpha C_{FS}(T_{DR}+\chi T_F)] \quad (9)$$

$$M_{TSAT}=16\mu M_S+(1-\mu)M_F(T_{DR}+\chi T_F) \quad (10)$$

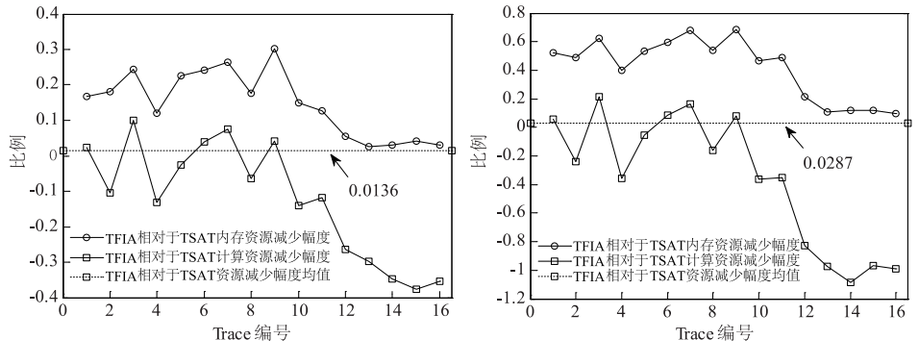


(*a*) 最坏情况下TFIA资源减少幅度 (*b*) 最佳情况下TFIA资源减少幅度

图2 Trace 1~16 TFIA与FT效率对比

与 FT 相比,自动机的引入增加了 TFIA 的计算负担,主要是因为自动机做了对流终止判断无贡献的状态变迁合理性判断,而且这种无贡献判断会随着流大小的增加而增加,因为一个流的数据包越多,进行的判断就越多,而只有最后一次判断是有贡献判断,例如 Trace 15 的平均流长最大,它的无贡献判断则最多;然而同样存在大量对流终止判断有贡献的状态变迁合理性判断,这些有贡献判断能节省计算资源,同时 TFIA 的单包流识别机制能够快速识别单包流,也能节省计算资源. 从实验结果可以发现,虽然 TFIA 相对于 FT 而言,最坏情况下计算资源开销有增加的可能,但总体而言计算开销仍有较大程度的减少,在最佳情况下,计算资源开销则大幅度减少;而在内存资源开销上,TFIA 在任何情况下都比 FT 有大幅度的减少,因此 TFIA 算法在效率上远优于 FT 算法.

与 TSAT 算法相比,无论在最坏情况下还是在最佳情况下 TFIA 算法的平均计算代价都高于 TSAT,这是因为自动机进行的无贡献判断增加了 TFIA 的计算资源开销. 然而,在部分 Trace 上 TFIA 的计算代价低于 TSAT,说明在自动机开销不太大的情况下,TFIA 的计算代价低于 TSAT,而自动机开销随着流大小的增加而增加,流的数据包越少,自动机进行的判断越少,资源开销就越少,通常情况下中流和小流的数据包数较少,因此 TFIA 算法更适合中流和小流比重较大的情况. 而在内存资源开销方面,TFIA 算法在任何情况下都大幅低于 TSAT 算法. 就综合效率而言,最坏情况下 TFIA 比 TSAT 节省了 1.36% 的资源开销,最佳情况下则节省了 2.87% 的资源开销,而且对于整个流识别系统而言,内存代价的减少比计算代价的减少更有意义,因此 TFIA 算法的综合效率优于 TSAT 算法.



(*a*) 最坏情况下资源减少幅度　　(*b*) 最佳情况下资源减少幅度

图3　Trace 1~16 TFIA与TSAT效率对比

表 3　代价模型参数

编号	参数	参数意义	编号	参数	参数意义
1	C_{FC}	创建流记录所需计算资源	9	T_{SYN}	SYN 状态保持时间阈值
2	C_{FS}	扫描流记录所需计算资源	10	T_{SYN40}	大小为 40 字节的 SYN 数据包的状态保持时间阈值
3	C_{DFA}	自动机消耗的平均计算资源(折合为 C_{FS}的倍数)	11	T_F	固定超时策略超时阈值
4	α	扫描频率	12	T_{DR}	非单包流的平均持续时间
5	M_F	流记录所需内存资源	13	C_{SC}	TSAT 算法在单包空间创建流所消耗的计算资源
6	μ	TCP 流中单包流比例	14	C_{SS}	TSAT 算法在单包空间扫描流所消耗的计算资源
7	P_{SYN}	TCP 单包流中 SYN 单包流比例	15	M_S	TSAT 算法中流在单包空间占用的内存资源
8	P_{SYN40}	SYN 单包流中 SYN40 的比例	16	β,χ	区分资源开销上限($\beta=1,\chi=1$)和下限 ($\beta=0,\chi=0$)

4.2　精度分析与对比

在连接建立阶段设置较小的流状态保持时间阈值,势必提升 TFIA 的识别效率,然而却牺牲了一定的识别精度. 在流识别领域,通常以 64s 固定超时策略的识别结果为默认精度标准,本文将基于该默认标准使用流截断率量化 TFIA 识别精度并与其他流识别算法进行精度对比.

定义 8　对于一个 Trace,当使用某种流识别策略之后 Trace 中真实流被截断的比例,称为流截断率,标记为 R_{SHOR}.

$$R_{SHOR}=\frac{N_{TRUE}-N_{SAME}}{N_{TRUE}} \tag{11}$$

其中,N_{TRUE}为 Trace 中的真实流数,N_{SAME}为使用某种策略之后产生的流中与 Trace 中真实流相同的流数.

基于 FT 策略的精度分析,仅是对流识别算法自身精度的一种量化,为进一步评价 TFIA 算法精度,仍需与 TCP 流识别方面的代表性算法 TSAT 进行对比,本文以 64s 固定超时策略的识别结果作为标准答案,选择 $T_{SYN}=16s$,$T_{SYN40}=2s$,基于 Trace 1 ~ 16 分别计算 TFIA 和 TSAT 的流截断率,结果如表 4 与表 5 所示.

表 4　基于 Trace 1 ~ 8 的流识别精度值

Trace 编号	1	2	3	4	5	6	7	8
TFIA 流截断率	1.42%	0.72%	1.53%	1.39%	1.05%	1.13%	1.67%	1.78%
TSAT 流截断率	3.63%	2.93%	3.76%	3.50%	3.30%	3.58%	4.07%	3.96%

表 5　基于 Trace 9 ~ 16 的流识别精度值

Trace 编号	9	10	11	12	13	14	15	16
TFIA 流截断率	1.28%	1.02%	2.06%	1.24%	1.29%	1.32%	1.36%	1.84%
TSAT 流截断率	3.52%	7.47%	6.58%	7.30%	7.06%	6.91%	5.93%	6.02%

由实验结果可得,TFIA 基于 Trace 1 ~ 16 的平均流截断率为 1.38%,优于 TSAT 的平均流截断率 4.97%. TFIA 之所以比 TSAT 精度高,主要原因是 TSAT 使用 FIN 标志定义流终止,而 FIN 数据包不一定是流的最后一个数据包,通常 FIN 数据包之后还跟随一个 ACK 数据包,因此这部分流将被截断;而 TFIA 基于 TCP 协议规则进行流终止判断,避免了 TCP 流的不合理截断.

5　结论

提高网络流识别效率一直是基于流粒度网络行为

研究面临的重要问题之一. 本文基于对现有网络流终止策略的分析,提出了网络流属性识别度,分析和比较了不同流属性的识别能力;通过对 TCP 交互过程的研究,构造了 TCP 双向流自动机,设计了基于该自动机的 TCP 流识别算法 TFIA. 相关分析和大量实验表明 TFIA 与流识别领域默认基准算法 FT 相比,识别效率平均提升了 50.70%,而识别精度仅损失了 1.38%;同时与目前流识别算法的代表 TSAT 相比,虽然计算代价有一定程度的增加,但内存开销则有更大幅度的减少,所以综合效率仍有 2.11% 的提高,而且流识别精度也提高了 3.59%. 因此,就综合性能而言,TFIA 优于经典算法和现有同类代表性算法.

然而,自动机的引入增加了 TFIA 的计算开销,而且这种开销随着流大小的增加而增加,因此 TFIA 算法比较适合中流、小流和不规则 TCP 流比重较大的情况. TFIA 的这个特点使得该算法在面临 DDoS 攻击、蠕虫爆发等网络异常时,能够很好地避免系统因小流数量急剧增加导致的资源耗尽,保障了流识别系统的正常运行,另外还可以根据 TFIA 算法与其他算法的效率对比结果感知网络流量状态.

参考文献

[1] CLAFFY K C, BRAUN H W, et al. A parameterizable methodology for internet traffic flow profiling [J]. IEEE Journal on Selected Areas in Communications, 1995, 13(8):1481 – 1494.

[2] RYU B, CHENEY D, et al. Internet flow characterization: adaptive timeout strategy and statistical modeling [A]. Proceedings of PAM 2001 Workshop [C]. Amsterdam, Netherlands: RIPE NCC, 2001. 95 – 105.

[3] WANG J F, LI L, et al. A probability-guaranteed adaptive timeout algorithm for high-speed network flow detection [J]. Computer Networks, 2005, 48(2):215 – 233.

[4] 周明中,龚俭,等. 网络流超时策略研究[J]. 通信学报, 2005,26(4):88 – 93.
ZHOU Ming-zhong, GONG Jian, et al. Study of network flow timeout strategy [J]. Journal on Communications, 2005, 26(4):88 – 93. (in Chinese)

[5] 周明中,龚俭,等. 高速网络中基于流速测度的动态超时策略 [J]. 软件学报,2006,17(10):2141 – 2151.
ZHOU Ming-zhong, GONG Jian, et al. High-speed network flows′ dynamical timeout strategy based on flow rate metrics [J]. Journal of Software, 2006, 17(10):2141 – 2151. (in Chinese)

[6] Cisco. NetFlow Services Solutions Guide [M/OL]. http://www.cisco.com/en/US/docs/ios/solutions_docs/netflow/nfwhite.pdf, 2007 – 01 – 22.

[7] RODRIGUEZ J M, ESPANOL V C, et al. Empirical analysis of traffic to establish a profiled flow termination timeout [A]. Proceedings of International Wireless Communications and Mobile Computing Conference [C]. Cagliari, Italy: IEEE Computer Society, 2013. 1156 – 1161.

[8] CAI J, ZHANG Z B, et al. An adaptive timeout strategy for UDP flows using SVMs [A]. Proceedings of Parallel and Distributed Computing, Applications and Technologies [C]. Wuhan, China: IEEE Computer Society, 2010. 118 – 127.

[9] 赵小欢,夏靖波,等. 高速网络 UDP 流超时策略研究 [J]. 合肥工业大学学报(自然科学版),2013,36(2):176 – 180.
ZHAO Xiao-huan, XIA Jing-bo, et al. Research on UDP flow timeout strategy in high-speed network [J]. Journal of Hefei University of Technology (Natural Science), 2013, 36(2):176 – 180. (in Chinese)

[10] LEE D, CARPENTER B E, et al. Observations of UDP to TCP ratio and port numbers [A]. Proceedings of International Conference on Internet Monitoring and Protection [C]. Barcelona, Spain: IEEE Computer Society, 2010. 99 – 104.

[11] ZHANG X G, DING W. Comparative research on internet flows characteristics [A]. Proceedings of International Conference on Networking and Distributed Computing [C]. Hangzhou, China: IEEE Computer Society, 2012. 114 – 118.

[12] Jiangsu Key Laboratory of Computer Networking Technology. IP Trace Distribution System [DB/OL]. http://iptas.edu.cn/src/system.php, 2011 – 10 – 01.

作者简介

张孝国 男,1980 年 3 月出生于河南省平顶山市,讲师,博士研究生,主要研究方向:网络测量,网络行为学等.
E-mail:xgzhang@njnet.edu.cn

丁 伟 女,1962 年 5 月出生于江苏省南京市,工学博士. 现为东南大学教授,博士生导师. 主要研究方向:网络系统结构,网络测量,网络安全,网络行为学等.
E-mail:wding@njnet.edu.cn