第 37 卷 第 24 期
Vol.37 No.24

计 算 机 工 程
Computer Engineering

2011 年 12 月
December 2011

• 网络与通信 •

文章编号：1000—3428(2011)24—0061—03 文献标识码：A 中图分类号：TP393.09

网络心跳包序列的数据流分簇检测方法

易军凯[1]，陈 利[1]，孙建伟[2]

(北京化工大学信息科学与技术学院，北京 100029)

摘 要：在对网络会话进行时序分析的基础上，提出基于数据流分簇处理的心跳包序列检测方法。对数据流进行时序分簇处理，按周期性特征扩充簇集合，筛除不符合特征的簇对象，根据稳定的簇集合检测心跳包序列。实验结果表明，该方法检测率较高、误检率较低，能够实现实时检测和处理。

关键词：周期性序列；心跳包；分簇；心跳包检测；网络同步

Data Flow Clustering Detection Approach of Network Heartbeat Packet Sequence

YI Jun-kai[1], CHEN Li[1], SUN Jian-wei[2]

(College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China)

【Abstract】Based on timing analysis of network sessions, this paper puts forward a detection approach of heartbeat packet sequence based on clustering processing. It processes the network data stream with time clustering, expands the cluster collection by periodic features, and screens out the clusters which do not meet the characteristics, and detects the heartbeat packet sequence within steady cluster collection. Experimental results show that this approach can achieve higher correct detection rate, real-time detection and processing.

【Key words】periodic sequence; heartbeat packet; clustering; heartbeat packet detection; network synchronous

DOI: 10.3969/j.issn.1000-3428.2011.24.020

1 概述

心跳包是网络数据流中一种自定义协议、固定信息、循环发送的数据包[1]，在各种网络应用中作为在线状态检测、状态汇报方式、网络同步或其他定时机制的应用而普遍存在[2]。一个网络会话是否包含心跳包序列以及该心跳包序列的特征[3]往往可以作为区分一个会话与其他会话的重要特征。在很多网络会话中，心跳包与业务数据包混杂在同一个网络连接里，现有技术一般是通过源端口、目的端口、数据包大小等与预定值特征匹配[4]进行检测的，不具备识别一般性网络数据流心跳包序列的功能。为此，本文提出一种基于数据流分簇处理的检测心跳包序列的方法。

2 网络数据流周期性心跳包序列特征

心跳包的发送通常有 2 种技术：(1)由用户在应用层自定义协议实现的心跳包；(2)由 TCP/IP 协议层内置的 KeepAlive 功能[5]。

心跳包序列是网络会话中周期性心跳行为的数据流反映，主要表现为 3 个特征：(1)周期性，在时序关系的网络数据流中，心跳包序列(包含发送心跳包和响应心跳包)周期性出现，在普通网络应用中，心跳包发送周期一般为 10 s、30 s 或者 60 s。(2)特定性，即周期性出现的心跳包序列具有固定的序列特征，每次出现的数据包序列内容和大小一致。(3)简洁性，由于心跳包序列只是作为连接状态的检测手段，往往简洁精悍，通常只包含一个发送数据包和一个响应数据包，而且数据包比较小，便于节省网络带宽资源。

3 基于分簇处理的心跳包序列检测方法

3.1 相关术语

包序列：以时间为序的数据包的集合。

包序列特征值：由既定的特征向量组成的包序列的属性。

包簇：在时序上关系紧密的数据包的集合。具体判断标准为：当相邻 2 个数据包的时间间隔不大于时间值 T，则认为这 2 个数据包在同一个包簇内；否则，判断这 2 个数据包分属于 2 个包簇。

3.2 参数设置

获取网络数据流，并根据网络状况计算包簇间最小时间间隔值为 T，定义构成最小包簇所需网络包的个数为 $S_{\min}$，$S_{\min}$ 为正整数，T 由网络的往返时延决定，具体计算如下：

$$T = a \times \left(\sum_{i=0}^{n} t(i+1) - t(i) \right) / n$$

其中，函数 $t(i+1)-t(i)$ 表示当前连接中，有交互行为的 2 个邻近数据包的时间差，代表一定时刻的往返时延，一般将最近 20 次不同往返时延取平均值(即 $n=20$)，然后经过放大作为 T 取值，a 为放大倍数，本检测方法默认设置为 10。

3.3 网络数据流分簇处理

首先对网络数据流进行分簇处理，得到包簇列表，步骤如下：

第 1 步 构造一个包簇列表 Qc 和一个临时的数据包列表 Lp。Qc 用于存储包簇对象 c；Lp 用于存储从网络数据流中获取的数据包。Qc 包含但不限于以下 3 个属性：数据包序列 Qc_l，数据包数量 Qc_n，包簇对象特征值 Qc_c；Lp 包含但不限于以下 4 个属性：数据包序列 Lp_l，数据包数量 Lp_n，包簇

基金项目：国家部委基金资助项目

作者简介：易军凯(1972－)，男，教授、博士，主研方向：信息安全，优化调度；陈 利，硕士研究生；孙建伟，副教授、博士

收稿日期：2011-06-03 **E-mail：**bhchenli@sina.com

对象特征值 Lp_c，最后一个数据包的时间 T_{last}；c 包含但不限于以下 6 个属性：序列数量 c_{num}，序列均值 c_{avg}，序列纯度 c_{pure}，时间跨度 c_{tc}，序列最值 $c_{\max}$，起始包时间 c_{tb}。

第 2 步 从网络数据流中读取一个数据包；记数据包的时间为 Tp，并设定 Lp_n 值为 0。

第 3 步 将第 2 步或第 4 步获取的数据包添加至 Lp_l，Lp_n 值增 1，并更新 $T_{\text{last}} = Tp$。

第 4 步 判断网络数据流是否结束；若结束，则以 Lp_l 构造一个包簇对象 c，计算 c 的特征值，并以特征值为索引添加至 Qc 中，结束分簇；否则，继续读取下一个数据包，记录其时间为 Tp。

第 5 步 判断 $Tp - T_{\text{last}} > T$ 是否成立。如果不成立，将数据包添加到 Lp_l，Lp_n 增 1，并更新 $T_{\text{last}} = Tp$，回到第 4 步；如果 $Tp - T_{last} > T$ 成立，且满足 $Lp_n \geqslant S_{\min}$，则以 Lp_l 构造一个包簇对象 c，添加至 Qc 中，然后回到第 4 步；如果 $Tp - T_{\text{last}} > T$ 成立，且满足 $Lp_n < S_{\min}$，则清空 Lp，然后回到第 4 步。处理过程如图 1 所示。

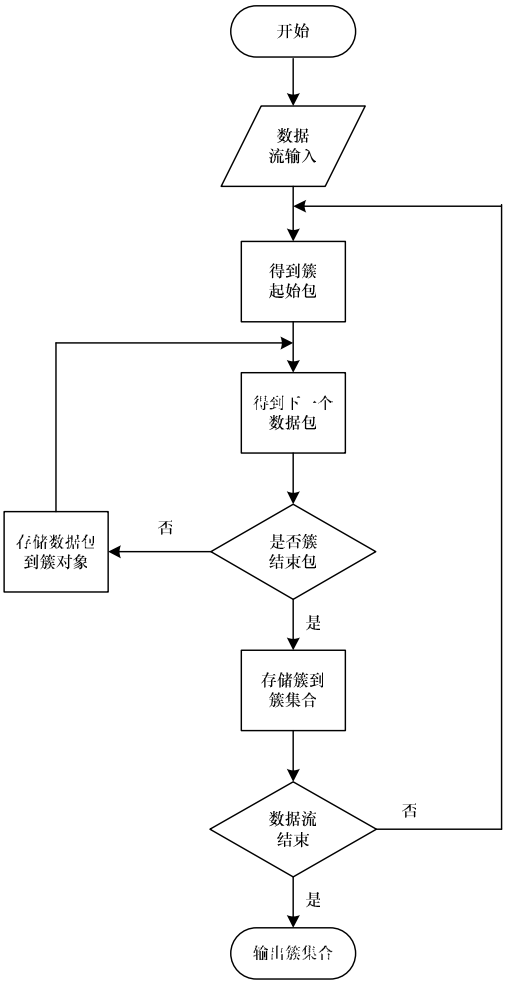

图 1 数据流分簇处理流程

第 4 步中所述计算包簇对象 c 的特征值的方法为：如果 c 属性 c_{num} 满足 $c_{\text{num}} < m$，其中 m 为设定的正整数且 $m > S_{\min}$ 成立，则以 Lp_l 作为 c 的特征值；否则，以 c 的属性集 $\{c_{\text{avg}}, c_{\text{pure}}, c_{\text{tc}}, c_{\max}\}$ 作为 c 的特征值。

3.4 簇列表更新和心跳包序列识别

从得到的 Qc 中挑选出具有相同特征值并且出现的时间间隔差小于周期性最大容忍方差 Ma（Ma 为设定值）的包簇对象集合，记录其数量为 Mum；如果 $Mum \geqslant Mum_1$（Mum_1 为设定正整数），则判断其是否具有周期性特征；否则从 Qc 删除该包簇对象 c 集合。操作步骤为：

第 1 步 对 Qc 按照特征值进行索引，此时特征值相同的包簇对象位于同一个集合中，记录每个特征值对应集合中的 c 数量 Mum。

第 2 步 依次判断每个特征值对应集合中的包簇对象 c 数量是否满足 $Mum \geqslant Mum_1$，如果不满足，则将该集合从 Qc 中删除。

第 3 步 依次对 c 数量 Mum 不小于 Mum_1 的特征值集合进行操作，判断是否具有周期性特征。具体操作为：

(1)对该集合中每 2 个相邻的包簇对象 c 的起始时间 c_{tb} 做差值运算，得到相邻包簇对象之间的时间间隔。

(2)得到相邻包簇对象 c 之间的时间间隔的平均方差。

(3)如果该平均方差小于周期性最大容忍方差 Ma，则认为该集合中的包簇序列具有周期性特征；否则，执行第(4)步。

(4)依次使用(1)中得到的每一个时间间隔做如下操作：针对原始网络数据流，以该时间间隔的 2 个相邻包簇对象，以在前的包簇对象的起始时间为起点，该时间间隔为步长，分别向前和向后查询与该集合的特征值相同的数据包序列；若存在这样的数据包序列，则将其构造包簇对象 c 并按特征值索引添加到 Qc 中，然后返回到(1)对该集合进行操作；否则，结束对该时间间隔的操作。

第 4 步 从稳定的 Qc 里的具有周期性特征的包簇序列集合中，选取数量比例大于 $Rate$（设定值）的包簇序列作为周期性心跳包序列。

至此，检测过程结束。在算法参数中，包簇间最小时间间隔值 T 由网络环境决定，最小簇包个数 $S_{\min}$、时间间隔差最大容忍方差 Ma 及参数 Mum_1、$Rate$ 由心跳包特征判定严格度决定。通过大量实验证明，$S_{\min} = 2$、$Ma = 0.2$、$Mum_1 = 5$、$Rate = 0.8$ 的设置对于所有实验具有最佳检测效果。

4 实验及结果

本文实验选取大量含有周期性心跳行为与不含周期性心跳行为的网络应用程序，并分别在实验室、计算机楼、校园网 3 个网络环境运行，使用抓包程序获取应用程序产生的网络数据流，并对数据流做心跳包序列检测。实验环境为：CPU Pentium 2.49 GHz，内存 2 GB，开发平台 Python2.6.4，操作系统 Ubuntu8.04 Desktop。

本文以一个含周期性心跳行为的网络应用为例，绘制其网络数据流原始时序关系图，如图 2 所示。

图 2 反映了主机 A 与主机 B 会话过程中网络数据流的时序关系。图中横轴代表时间，纵轴上柱形代表网络数据包，柱形长度代表数据包的大小，柱形顶端标注具体数值，正负代表方向，正数代表该柱形是 A 发向 B 的数据包，反之，负数代表 B 发向 A 的数据包。等长虚线代表时间分隔线，一条虚线代表 1 s 间隔，两分隔线之间的数据包在时间上属于同一秒。

在检测过程中，计算出包簇间最小时间间隔 T 为 1 s，定义最小包簇数据包个数 $S_{\min}$ 为 2，时间间隔差最大容忍方差 Ma=0.2，通过检测最终得到心跳包序列。

图 3 为该序列所绘制时序图，证明该序列确实符合网络心跳包序列特征。

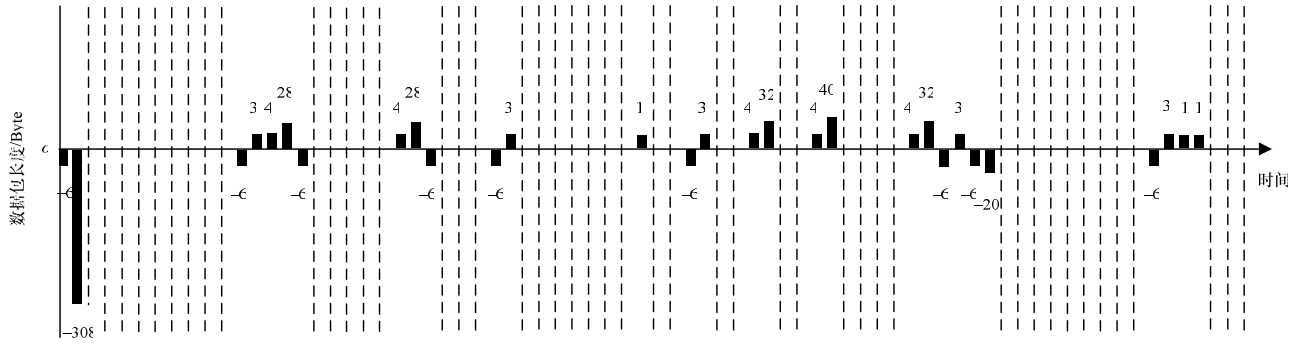

图 2　网络数据流原始时序图

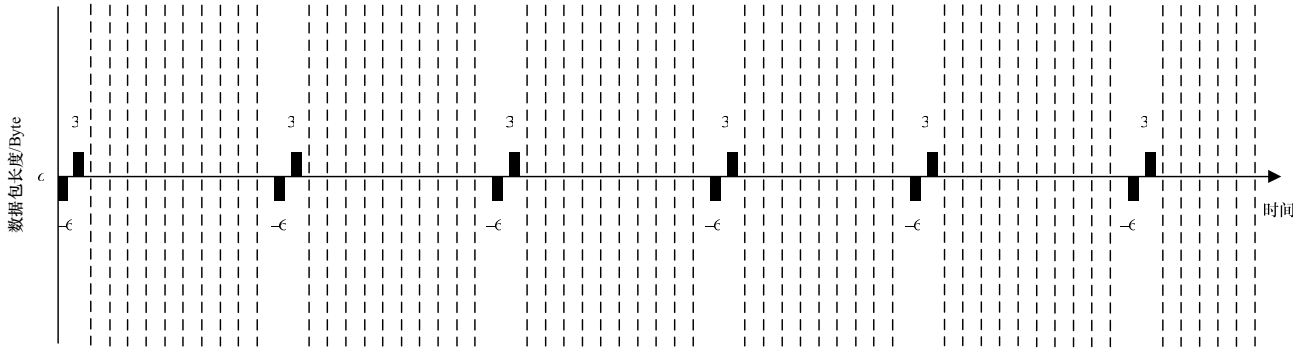

图 3　被检测出的周期性心跳包序列时序图

本文选取了 500 个不同的网络应用，对每个网络应用在不同的网络环境下所产生的网络数据流进行心跳包序列检测，并绘制时序关系图。

实验结果如表 1 所示。

表 1　检测结果统计

实验环境	网络状况	324 个含有心跳行为的网络应用				176 个不含心跳包行为的网络应用		
		检测数	正确检测数	正确检测率/(%)	平均检测时间/s	检测数	错误检测率/(%)	平均检测时间/s
实验室	良好	324	324	100.00	0.98	0	0.00	0.78
计算机楼	一般	321	319	99.38	1.02	0	0.00	0.97
校园网	拥塞	320	319	98.46	2.32	1	0.56	2.13

经过多次实验，从正确检测率、错误检测率、平均检测时间 3 个方面衡量该检测方法的可用性。由表 1 中统计数据可以看出，该检测方法达到了比较高的正确检测率，将错误检测率降低在 1%以下，同时耗费了很少的 CPU 运算时间，证明了该检测方法的可用性，对网络数据流特征的分析具有重要的意义。

5　结束语

本文在对网络会话数据流进行时序分析的基础上，提出基于数据流分簇处理的检测心跳包序列的方法。实验表明，该检测方法能够有效地识别出一般网络应用数据中的心跳序列包；算法参数值能较好地自适应环境，适用于不同网络状况的应用环境；同时该方法高效地利用了处理器资源，在拥塞的网络环境下也能做到快速处理。此外，该方法已经被实际应用于网络入侵检测和网络状态监控等实时网络数据流分析当中，并取得了较好的效果，表明该方法具有很强的实用性。但是该方法仍存在一些不足，在拥塞恶劣的网络环境下，正确检测率略有降低，如何在拥塞网络环境下保证较高的正确检测率还有待提高。

参考文献

[1]　Li Feifei, Yu Xiangzhan, Wu Gang. Design and Implementation of High Availability Distributed System Based on Multi-level Heartbeat Protocol[C]//Proc. of the International Conference on Control, Automation and Systems Engineering. [S. l.]: IEEE Press, 2009: 83-87.

[2]　Zhao Haijun, Ma Yan, Huang Xiaohong, et al. Forecasting Heartbeat Delay for Failure Detection over Internet Using Nonlinear System[C]//Proc. of the World Congress on Computer Science and Information Engineering. [S. l.]: IEEE Press, 2009: 589-593.

[3]　Hou Zonghao, Huang Yongxiang, Zheng Shouqi, et al. Design and Implementation of Heartbeat in Multi-machine Environment[C]//Proc. of the 17th International Conference on Advanced Information Networking and Applications. Xi'an, China: [s. n.], 2003: 583-586.

[4]　Babaoglu P O, Binci T, Jelasity P M, et al. Firefly-inspired Heartbeat Synchronization in Overlay Networks[C]//Proc. of the 1st International Conference on Self-adaptive and Self-organizing Systems. [S. l.]: IEEE Press, 2009: 77-86.

[5]　Li Huaming, Tan Jindong. Heartbeat-driven Medium-access Control for Body Sensor Networks[J]. IEEE Transactions on Information Technology in Biomedicine, 2010, 14(1): 44-51.

编辑　顾逸斐