第 37 卷　第 10 期
Vol.37　No.10

计　算　机　工　程
Computer Engineering

2011 年 5 月
May 2011

• 网络与通信 •　　文章编号：1000—3428(2011)10—0102—03　　文献标识码：A　　中图分类号：TP393.08

一种联合 DPI 和 DFI 的网络流量检测方法

叶文晨，汪　敏，陈云寰，张之远

(上海大学特种光纤与光接入网教育部重点实验室，上海　200072)

摘　要：提出一种以深度包检测(DPI)技术为主、深度流检测(DFI)技术为辅的网络流量检测方法。基于 MPC8572 网络处理器的模式匹配引擎模块，利用 DPI 实现细粒度检测，对于 DPI 的误识别情况，通过 DFI 进行鉴别并提示重新检测，以达到纠错目的。实验结果表明，联合方法具有检错和纠错功能，且能提高网络流量检测的准确率。

关键词：深度包检测；深度流检测；网络流量检测；模式匹配引擎

Network Flow Inspection Method of Joint DPI and DFI

YE Wen-chen, WANG Min, CHEN Yun-huan, ZHANG Zhi-yuan

(Key Laboratory of Special Fiber Optics and Optical Access Networks, Ministry of Education, Shanghai University, Shanghai 200072, China)

【Abstract】This paper presents a network flow inspection method which combines Deep Packet Inspection(DPI) with Deep Flow Inspection(DFI). DPI fulfills the target of fine-grained at the use of Pattern Match Engine(PME) of the network processor MPC8572. DFI is to notify the underlying false detection that can be caused by DPI and to request for re-detecting. Experimental results show that joint method has the ability of inspection wrong and error correction, and it can improve the accuracy of inspection.

【Key words】Deep Packet Inspection(DPI); Deep Flow Inspection(DFI); network flow inspection; Pattern Match Engine(PME)

DOI: 10.3969/j.issn.1000-3428.2011.10.034

1　概述

随着互联网的飞速发展，网络流量分布不均，网络安全状况复杂，互联网亟待有效管理和维护。由于对网络流量的检测是实施网络管理的前提，因此产生网络流量检测技术。然而，就目前而言，单独使用任一种技术，都不能满足准确而完整的检测要求。本文给出一种设计方案，主要借助 MPC8572 网络处理器的模式匹配引擎(Pattern Match Engine, PME)模块，实现深度包检测，进而辅以流检测技术，使其具有较好的检测可靠性。

2　现有检测技术

从观测角度而言，现有检测技术大致分为基于数据包的检测、基于数据流的检测和基于节点的检测，其代表技术包括深度包检测(Deep Packet Inspection, DPI)、深度流检测(Deep Flow Inspection, DFI)和盲检测(BLINC)。

文献[1]对现有网络检测技术进行了一个较全面的比较，如表 1 所示。

表 1　各种网络检测技术的比较

技术	名称	优点	缺点
DPI	端口检测	简单高效	准确性差
	深度包检测	细粒度	加密等情况下失效，实现代价大
DFI	Bayesian 检测	不受加密限制	只能分出协议类型
	深度流检测	不受加密限制	只能分出协议类型，受各参数影响大
BLINC	BLINC	全面	只能分出协议类型，需要分析大量协议的行为特征
	连接行为模式	全面	只能分出协议类型

其中，“只能分出协议类型”是指这项检测能分辨出它是 P2P 流量，但不能分辨出是 P2P 中的 BT、eDonkey 或 Napster。

3　联合检测方法的实现

从第 2 节的比较可以看出，流检测和节点检测都不能识别具体协议，属于粗粒度的检测；但是包检测能做到细粒度。从运营商的角度，不仅想知道网络中有 P2P 和流媒体的流量，也希望流量检测技术能精确划分出 BT 或电驴、PPLive 或 PPStream，这更利于流量计费业务和带宽有效分配。

本文方法以 DPI 技术为主，确保细粒度检测。DFI 技术用来弥补 DPI 技术的一些弱点：对于 DPI 可能存在的误识别，DFI 进行鉴别并提示重新检测，达到纠错目的。

3.1　硬件支持

MPC8572 PowerQUICCIII 网络处理器[2]是为高性能网络安全应用设计的。它有 2 个很关键的硬件加速引擎：硬件查找表单元(Table Lookup Unit, TLU)和模式匹配引擎。本文不重点介绍硬件及其工作原理，详细描述可参见文献[2]。总之，MPC8572 使 DPI 技术得到有效的硬件支持。

3.2　DPI 检测的实现

DPI 检测由 TLU 查询表、内存池和 PME 检测模块 3 个部分完成。

TLU 查询表主要用来存放超过 100 万条会话流记录，并且根据五元组迅速查找到这条记录。这条记录包括检测结果和内存地址。TLU 维护会话流是在高效的抓包、拆包前提下实现的。文献[3]详细介绍了网络行为系统数据平面的优化。

内存池存放会话流的详细信息，用于后续流检测。每个内存节点间采取双向链表形式，便于管理。

PME 检测模块检测一个未识别的数据包。

基金项目：上海市科委科研计划开放课题基金资助项目(09511501300)；上海市重点学科建设基金资助项目(S30108)

作者简介：叶文晨(1986－)，男，硕士研究生，主研方向：网络流量检测，嵌入式软件设计；汪　敏，教授；陈云寰、张之远，硕士研究生

收稿日期：2010-11-24　**E-mail：**ywwhy0412@sina.com

本节将介绍 TLU 查询表、内存池和 PME 检测模块之间的关系，以及在检测过程中的流程。

3.2.1　TLU 查询表与内存池

TLU 表和内存池完成会话流的维护和管理。

IP 数据包有五元组(源 IP、目的 IP、源端口、目的端口和协议类型)信息。每一个进入检测系统的 IP 包将按照它的五元组信息访问“TLU 查询表”。如果 TLU 表的“KEY”中没有这个数据包的五元组，说明这是一个新会话流的第 1 个数据包，那么 TLU 表中增加一条记录，并且分配一个内存节点；如果查到相同的五元组，说明当前数据包的会话流已存在，那么只需更新会话流的统计信息。

由于 TLU 表每条记录的字节限制[2]，因此总包数等统计信息不能放在 TLU 表中，而要记录在内存中。本文开辟了一块大小为 120 MB 的内存空间，采用内存池的管理方式。如图 1 所示，TLU 表中的“内存地址”对应每个内存节点在内存池中的偏移地址。

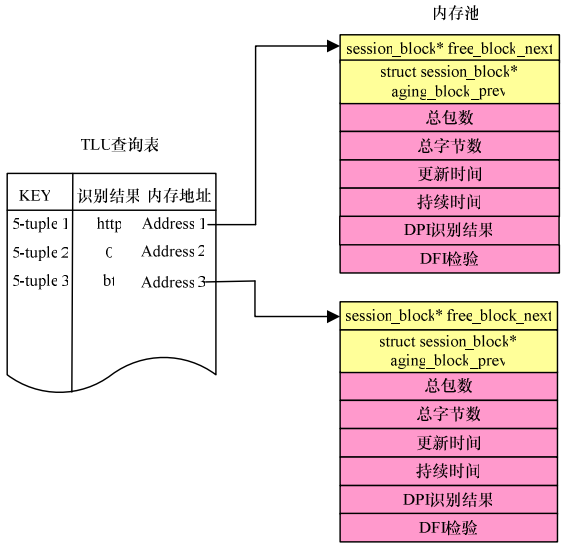

图 1　TLU 表和内存池的对应关系

3.2.2　TLU 查询表和 PME

协议特征一般用正则表达式标识，PME 内部有一个正则表达式的模板，用来匹配特征码。未识别的会话流数据包被送入 PME 进行检测。PME 反馈检测结果给 TLU。

由于一个会话流在一段时间内持续提供同一个业务[4]，因此为保证检测效率，PME 只检测 TLU 表中识别结果为 0 的会话流数据包。对于有识别结果的，不再交给 PME。

3.2.3　内存池和 PME

内存池和 PME 模块在 DPI 检测过程中没有直接交互。内存节点中的识别结果由 TLU 表查询时更新。DPI 检测的实现过程如图 2 所示。

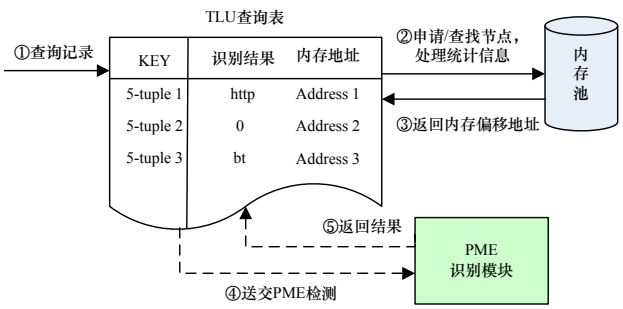

图 2　数据包的 DPI 检测流程

在图 2 中，步骤④、步骤⑤可能因为已识别而跳过；步骤②中“申请/查找”由查询 TLU 表中有无记录决定。

3.3　DFI 对 DPI 的检验

DPI 检测可能存在错检。

错检：如图 2 TLU 表中的第 1 条记录，DPI 分辨出它是 Http 流。但是有可能在 MSN 聊天时，用户输入一个“Get”恰好符合 Http 的特征码。因此，需要确定识别出来的结果是准确可靠的。

本节将介绍如何在前文 DPI 技术基础上引入 DFI 技术，DFI 可以协助解决 DPI 存在的漏检，从而解决错检问题。

DFI 技术可以由软件程序完成。每隔一段时间，查询内存池中统计信息，将该会话流归类。本文统计信息主要有总包长、总字节数、更新时间和持续时间等。通过一些计算可以得到会话流的平均包长、突发性、长包比例和上下行流量比等重要参数，它们是 DFI 划分会话流类别的重要依据。

把 DPI 的检测结果与 DFI 进行比较，判别 DPI 和 DFI 的检测结果是否“吻合”，进而作出下一步决定，如表 2 所示。

表 2　DFI 检验结果和下一次检测策略

编号	DPI 识别结果	DFI 和 DPI 结果是否吻合	下一次检测策略
1	已识别	吻合	认定识别正确，不再检测
2	已识别	不吻合	认为识别错误，重新检测
3	未识别	×	继续检测；超过 100 个还未识别，就放弃检测，并报告统计特性

3.3.1　校验正确

第 1 种情况是指 DFI 和 DPI 检测结果吻合。比如 DPI 识别出一个会话流为 BT 协议，而 DFI 归类到 P2P，那么它们的结果是吻合的。由于 2 种独立的检测技术得到的结果一致，因此认为这个检测结果是可靠的，从而锁定这个会话流的识别结果。

本文从理论上分析结果的可靠性。如文献[5]用准确率衡量可靠性。准确率越高，识别结果在实际应用中的可靠性越强。

假设 DPI 的准确率为 A_P，DFI 的为 $A_F(0<A_P,\ A_F<1)$，它们相互独立。总的准确率 A 可以表示为：

$$A=1-(1-A_P)(1-A_F)$$

相比单独使用 DPI 技术，识别结果的准确率为：

$$\Delta A=A-A_P=A_F(1-A_P)$$

在文献[5]中，A_P 达到 90%以上。实际应用中考虑特征码的影响，A_P 大约在 70%~75%。由于 A_F 受门限影响较大，因此准确性和完整性都能达到 80%才可以在实际中应用[5]。不妨取 A_F=80%。经过 DFI 检验，DPI 的准确率比原来提高了 24%。因此，最终得到的检测结果是准确可靠的。

3.3.2　校验错误

第 2 种情况表示 DFI 检测结果和 DPI 识别结果不吻合。比如 DPI 识别出一个会话流为 BT 协议，而 DFI 归类到 Web，那么它们的结果是不吻合的。对于这种检测结果，认为它是不可靠的，必须通知 TLU 表，清除其识别结果，使它的下一个数据包交给 PME 检测。

3.3.3 未识别

在表 2 中，第 3 种情况表示 DPI 没能识别出此会话流的协议，也不存在吻合与否的问题。本文对此不作细述。但是，它们的统计信息还是有用的，需要提交，并且把它们归入未识别类。

3.4 DPI 与 DFI 的联合检测

根据 3.2 节、3.3 节的描述，联合检测流程如图 3 所示。

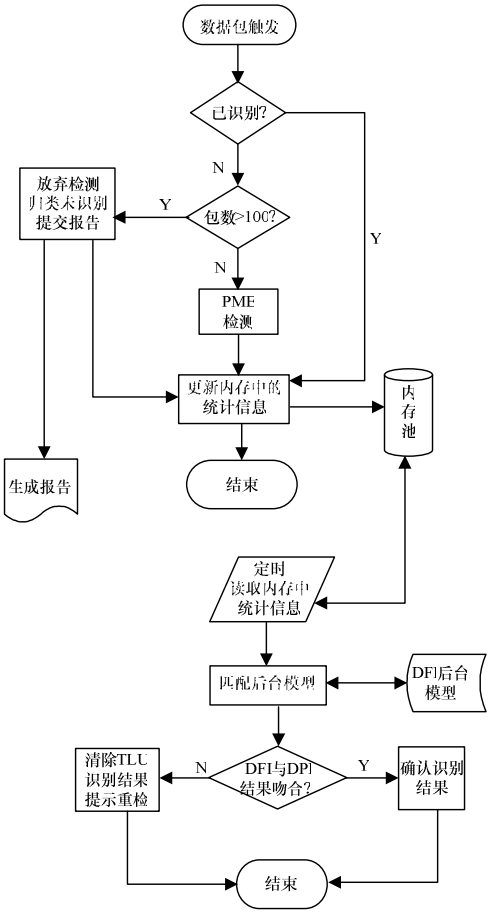

图 3 DPI 和 DFI 的联合检测流程

DPI 和 DFI 的联合检测是围绕内存池进行的。在图 3 中，内存池上方的是 DPI 检测部分，下方的是 DFI 检测部分；DFI 后台模型是按照统计信息划分出的协议类别，由日常流量分析而得。

DPI 对内存池进行写操作，更新会话流的统计信息和识别结果；DFI 读取内存池的统计结果，判断协议类型，进而锁定或清除 TLU 表中的识别结果。

对于一个会话流，它的识别结果只有经过 DFI 校验正确后才能保留下来，检验错误的结果将一次又一次被 DFI 清除。

4 实验分析

针对细粒度检测和错检自纠错功能，本节设计 2 组实验进行验证。

测试工具：Wireshark 抓包软件，SmartBits6000C 网络行为测试仪，Avanlanche 应用层测试软件。

测试环境：基于 MPC8572 的、DPI 与 DFI 联合检测的网络流量识别设备一台。

在测试中，先研究实验室网络流量，用 Wireshark 截取若干 pcap 数据包。由 Avanlanche 软件控制 SmartBits，根据 pcap 数据包产生若干会话流。流量识别设备检测的是这些会话流属于什么协议。

4.1 细粒度检测实验

抓取实验室网络中具有代表性协议的 pcap 数据包，使用 SmartBits 回放模拟真实网络环境。

实验结果给出设备识别出的协议组成以及各自比例。由于 DPI 技术的使用，设备能够识别具体协议，如 Http 占总流量的 49.46%，PPStream 和 BT 分别占 20.08%和 11.28%。

4.2 自检功能实验

实验中使用 Socket 工具通信，并且不断发送含有“Get”和“Socket”的内容，使其与 Http 协议混淆。通过观察最终识别结果是 Http 还是 Socket，可以验证设备的自检功能。

为了便于观察，实验中只发起一个通信，并做如下定义：

(1)Http 协议以“Get”为特征码；定义 Socket 协议，以“Socket”为特征码。

(2)Http 协议属于 Web 类型；定义 Socket 类型，Socket 协议属于 Socket 类型。

本文 DFI 技术利用 TCP 长包比例区分上述 2 个类型。

在不加入 DFI 技术时，检测结果的界面图见图 4。

```
+-----------------+------------+
| AppProtocolName | TotalByte  |
+-----------------+------------+
| http            |      11919 |
+-----------------+------------+
1 row in set (0.00 sec)
```

图 4 只用 DPI 检测 Socket 流量的界面图

图 4 表示识别出的应用协议类型和它的总字节数。由于不做 DFI 检验，会话流被认为是 Http 协议。

在加入 DFI 技术后，再次发送同样内容。检测经过如下步骤：

(1)DPI 检测出流量为 Http 协议；

(2)DFI 校验发现流量是 Socket 类型，标识校验错误，通知重新检测；

(3)DPI 检测出流量为“Socket 协议”；

(4)DFI 校验发现流量是“Socket 类”，标识校验正确，确定检测结果。

步骤(1)、步骤(2)可能由于错检而不断重复，直到两者吻合，这时系统把检测结果锁定，认为检测成功。

图 5 是 DPI 和 DFI 联合检测最终锁定结果的界面图。

```
+-----------------+------------+
| AppProtocolName | TotalByte  |
+-----------------+------------+
| Socket Tool     |      11997 |
+-----------------+------------+
1 row in set (0.00 sec)
```

图 5 联合检测 Socket 流量的界面图

比较图 4、图 5 可以发现，DFI 能对 DPI 做检验，DPI 和 DFI 联合检测设备能正确检测出 Socket 协议。

可见，加入 DFI 技术的检测方法相比单独使用 DPI 技术，不仅能辨别错误的检测，还能在下一次检测中予以纠正。

5 结束语

为了确保识别的细粒度，本文提出一种以 DPI 技术为主，DFI 技术为辅的检测方法，详细介绍 DPI 技术的实现过程和 DFI 对它的校验方法，将校验后的结果分 3 种情况处理，使整个检测过程在细粒度的基础上，又具有检错和纠错的能力，在一定程度上提高识别准确性。

本文提出的检测方法的局限性为：(1)会话流特性差别明显是检纠错的前提，如果遇到具有相似统计特征的协议，则本文方法不适用。(2)由于这种检测方法在加密、无应用层数据等情况下，DPI 技术失效，因此只能由 DFI 做粗粒度检测。(3)对于只有某几个数据包含有特征码的会话流，如果 DFI 校验后提示重检，但是错过了检测时机，则不能纠正为正确的协议。

(下转第 107 页)