

# 基于插件的 Bowman & CANEs 服务合成

吴世嘉, 朱学永

(解放军电子工程学院 网络管理中心, 安徽 合肥 230037)

**摘要:** 主动网络允许用户通过节点嵌入代码定制网络服务, 从而实现网络的可编程性, 使灵活加载新业务成为可能。介绍了主动网络的概念, 分析了 Bowman 和 CANEs 主动网络的实现方案。该方案提供了一种与传统操作系统中的可装载模块类似的扩展功能, 通过发送一系列带有配置信息的信令文件, 动态定制用户自己的协议和服务。

**关键词:** 主动网络; Bowman; CANEs; 服务合成

中图分类号: TP393

文献标识码: A

文章编号: 1674-7720(2010)20-0066-04

## Bowman & CANEs service composition based on plug-in

WU Shi Jia, ZHU Xue Yong

(Network Center of PLA Electronic Engineering Institute, Hefei 230037, China)

**Abstract:** Active network allows the user to insert customized programs into the switches of the network to perform customized network services. It can realize the network programmability and improve the network flexibility to introduce new services. The article presents the overview of active network and analyses such an implementation scheme as Bowman and CANEs. It provides an extension mechanism that is analogous to loadable modules in traditional operating systems. With the configuration information by sending a series of signaling, it provides support for additional user protocols and services.

**Key words:** active network; Bowman; CANEs; service composition

主动网络的概念最初由美国国防部高级研究计划署(DARPA)于1995年讨论网络系统未来发展方向时提出。针对当时网络体系结构中存在的一些问题(如较难吸收新业务和新规范、各层协议的冗余操作造成的网络性能下降等), 主动网络提供了一种开发平台, 在此平台上可方便快速地开发和运行新业务, 试验新的网络体系结构<sup>[1]</sup>。

主动网络有两个含义: 一是被称为主动节点的网络中间节点(如路由器、交换机)不仅完成存储转发等网络级的功能, 而且可以对包含数据和代码的所谓主动包和普通包进行计算。具有计算能力的网络节点从网络设备接收数据包后执行相应的程序, 对该数据包进行处理(如路由选择、数据合并、数据解包等), 然后将数据发送给其他网络节点。二是用户根据网络应用和服务的要求可以对网络进行编程以完成这些计算。

## 1 Bowman 和 CANEs

### 1.1 概述

Bowman 构架于主机操作系统的上层。主机操作系

统提供了底层机制, 如内存管理、线程创建和调用、同步原语调用和 I/O 设备访问等<sup>[2]</sup>; Bowman 提供了传递信息的抽象通道(Channels)、流程计算(A-flow Computation)、状态存储(Sate-store Memory)以及为了增加功能而提供的扩充机制。CANEs 执行环境在 Bowman 之上, 通过在特定的插槽上运行代码定制程序, 并通过它为主动服务提供合成框架。

### 1.2 Bowman

Bowman 提供了 3 种基本功能支持主动网络: 通道、流程计算、状态存储。除此之外, Bowman 还提供了一种与传统操作系统中的可装载模块类似的扩展功能, 使 Bowman 节点的其他组成部分能实时动态加载<sup>[3]</sup>。

(1) 通道。Channels 是由 Bowman 提供的通信资源, 通过可扩展协议发送、接收数据包。Bowman 向执行环境 EE 提供一系列函数, 使用户能在通道上创建、终止、查询和通信, 在各种不同的配置下实现传统协议(TCP、UDP、IP 等)。此外, Bowman 的通道还包括其他数据包处理形式, 如压缩、正向纠错法。

## 网络与通信 Network and Communication

(2)流程计算。A-flow 是 Bowman 用于计算的主要概念,它封装了包处理上下文和用户状态。每个流程至少包含一个线程,并由某个用户执行。Bowman 的流程允许用户根据需要同时激活多个线程。

(3)状态存储。State-store 为流程提供存储和检索状态的机制。这些状态由唯一的关键词索引。用隐含的状态存储机制,Bowman 提供注册接口创建存储状态,在流程之间不需要传送程序变量即能实现数据共享。

Bowman 的包输入/输出处理过程如图 1 所示。首先,对物理接口上接收到的数据包进行分类并确定相应的包处理信道,输入数据包分流到特定信道并接收信道定制的处理。一旦完成信道输入数据包,数据包即到达了与抽象链路相连的信道。然后,对数据包进行第二次分类处理:A-flow 处理、直通转发或信道间转发。

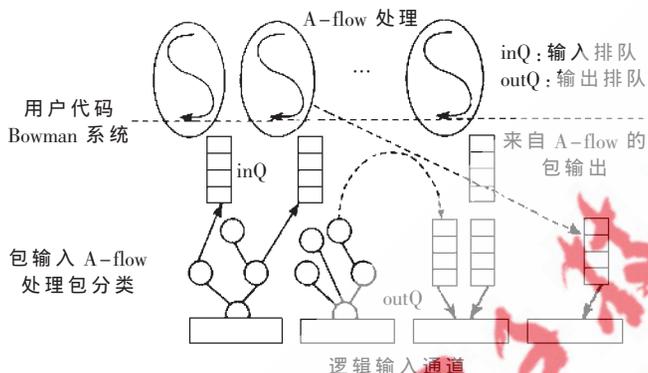


图 1 Bowman 的包处理

### 1.3 CANEs

CANEs 执行环境以 Bowman 节点操作系统为执行平台,其执行环境的插槽处理模型由固定部分和可变部分两部分组成。固定部分为隐含程序,对数据包进行统一处理;可变部分为嵌入式程序,实现数据包上用户定制的功能。隐含程序中执行嵌入程序的特定称为插槽(Slot)。

CANE 中业务的定制分两步完成。首先从主动节点中选取一段提供基本业务的隐含程序,其次选取嵌入式程序定制隐含程序。其中,嵌入式程序可直接从主动节点中获得或从远端服务器上载入。每个定制程序的执行需要一个或多个插槽。

## 2 Bowman 和 CANEs 服务合成的实现

### 2.1 CANEs 的信令文件

CANEs 依次有序地向主动节点发送一系列信令文件,主动节点收到这些文件后,建立 Bowman 下的抽象通道(Channels)和抽象拓扑(Topology),并生成相应的路由表,完成代码调用和相关计算。

#### 2.1.1 抽象链路协议 ALP

相应的 ALP(Abstract Link Protocol)信令文件默认发送到 TCP 端口 9998, Bowman 专门创建一个监听线程,根据收到的 ALP 文件创建或者销毁 Channels,并赋予其一

定的属性。ALP 信息的语法描述如下:

```

([create|destroy] {
    (name LinkName)
    ( proto [ether | udp] //大写字母表示相应的值
    (host (addr HostName) ([dev-name Device Name |
        port PortNumber]))
    (host (addr HostName) ([dev-name Device Name |
        port PortNumber]))
    )
    (status [up | down]) //通道状态
    (dir [bi | uni-incoming | uni-outgoing])
    //通道中数据流的方向
    (type [point-to-point | multi-point | broadcast |
        absolute | relative | raw])
    (access [regular | privileged]) //访问权限
    }

```

#### 2.1.2 抽象拓扑协议 ATP

抽象拓扑在物理拓扑的基础上实现用户定制的网络互联,其类似于虚拟专用网(VPN),叠加于网络之上,每个节点都有一系列特定的属性。ATP(Abstract Topology Protocol)信令文件默认发送到 TCP 端口 9999,它把属于不同抽象拓扑的链路进行分类,其语法描述如下:

```

(name TopologyName)
([add | delete] LinkName1)
([add | delete] LinkName2) //在某一拓扑中添加
// (删除)链路
...

```

#### 2.1.3 路由表 URT

Bowman 提供了一个静态单播路由协议,URT(Unicast Routing Table)配置文件默认发送到 TCP 端口 9990,然后采用最短路径算法在抽象拓扑上的每一个 Bowman 节点上生成一个路由表。URT 文件的语法结构描述如下:

```
(calculate TopologyName RoutingTableName)//计算路由表
```

#### 2.1.4 CANEs 用户接口 CUI

CANEs 信令消息 CUI(CANEs User Interface)由计算和通信两部分组成。其中,通信部分定义一系列包分类器(Packet Classifier),可以匹配数据包头中任意数量的域,以最长匹配原则把收到的不同的数据包进行分类,并送入相应的 flows 中。例如:

```

[ethernet (proto = ETHERTYPE_IP) | ip (proto =
    IPPROTO_UDP) |
    udp (sport = 124) | *] //Filter (过滤规则)

```

表示网络硬件接口将收到的所有源端口为 124 的 UDP 报文送入到该包过滤器所对应的用户 flows 中。

### 2.2 CANEs 的线程实现

在 Bowman 节点操作系统上运行 CANEs 执行环境的主动节点处理过程如图 2 所示。用户 flows 包括一个隐

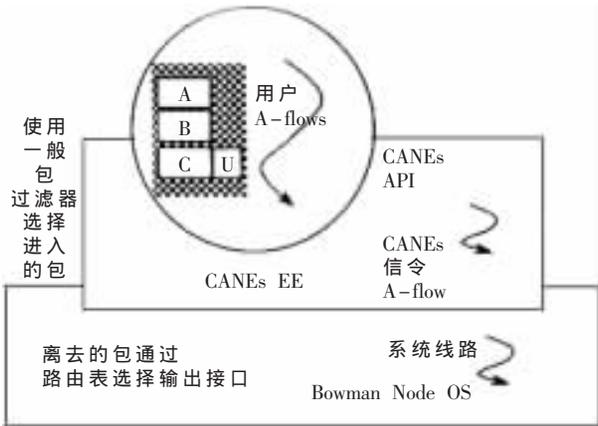


图2 CANEs的线程实现

含程序(U)和一系列定制程序(A、B和C)。在基于Bowman操作系统的CANEs线程实现中,CANEs EE启动控制A-flow,控制A-flow包含处理信令消息的线程。每个用户的计算在各自的A-flow中完成,并可启动多个共享同一地址空间的独立线程。

CANEs信令消息由计算部分和通信两部分组成。其中,计算部分定义了一张直观无环图(DAG)。DAG的根为一隐含程序,每个子节点对应于一个嵌入式程序,该嵌入式程序捆绑在其父节点的插槽上。DAG中的弧定义了子节点所捆绑的插槽,CANEs EE根据节点提供信息,获取执行计算DAC所需要的代码。CUI消息的通信部分为用户定义了一系列路由策略,用户可进一步定义包分类规则,选择A-flow处理的输入数据包。通信部分为计算部分的输入输出信道的建立提供了足够的信息<sup>[4]</sup>。

当隐含程序动态载入时,执行缺省的进入函数“-entry”,进入函数使隐含程序输出的插槽公共化。嵌入式程序在系统解析信令消息的计算部分时绑定插槽,激活插槽类似于激活某一特定事件,嵌入式程序是特定事件的处理程序,当事件被激活时并发执行嵌入式程序。

### 2.3 CANEs的服务合成

服务的合成在一定意义上就是应用程序的部署,而主动应用程序要通过分发机制到达主动节点,目前所采用的方式是带外方法、带内方法和两者相结合的方法。

带外方法是将程序注入网络节点的过程与数据包在网络节点处理的过程割离,即先向网络节点发送将要运行的程序,然后发送数据包,网络节点则运行已经收到的程序从而对该数据包进行处理。

带内方法是把程序与数据包捆绑在一起发送,首先将用户数据和用来处理数据的程序打包,形成胶囊。当网络中的主动节点收到胶囊时,首先对胶囊进行解封,然后用胶囊中的程序在临时执行环境中处理胶囊中的数据<sup>[5]</sup>。

CANEs属于带外处理方式。即先产生由客户向主动节点发送带有配置信息的信令文件,主动节点根据信令

的内容去查找所需要的主动代码,如果本地不存在,就从代码服务器上下载。这一平台所采用的服务合成方式是固定程序与嵌入程序相结合的方式。也就是说执行环境提供一个对于大多数应用都适宜的固定程序段不变,每次相对于不同的应用,可以加入不同的嵌入程序来满足不同的需要。这种服务合成方式可以用图3来表示。

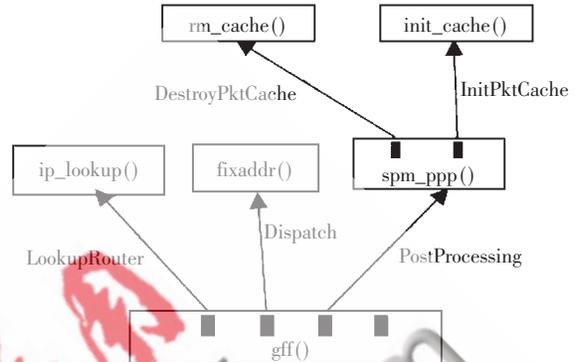


图3 CANEs的服务合成

从图中可以看出,gff是固定部分,而ip\_lookup、fix-addr和spm\_ppp是它的嵌入部分。同时,把spm\_ppp作为固定部分时,rm\_cache和init\_cache又是它的嵌入部分。CANEs将gff提供的4个插槽作为应用接口,而对于图中这一应用,只使用了4个插槽中的3个,它们的组合可以理解为:

- (1)通过数据包查找到目的地址;
- (2)固定这一地址,发送数据包,以免数据包在被处理时遭到破坏;
- (3)对数据分组进行处理。

同时,程序模块也包含插槽,这样就可以进一步扩展来实现其他的特殊功能。

### 2.4 测试结果及分析

以软件包中主动应用Atracroute为例,测试所得的结果如图4所示。大致经历以下步骤:(1)启动Bowman,在相应端口监听信令文件;(2)根据ALP信息创建了2个通道;(3)根据ATP信息构建了一个抽象拓扑;(3)根据URT信令文件建立路由表;(4)由CUI的配置信息,启动了一个flow,并查找所需要的主动代码,最后匹配相应的通道。

```

17:06:03 [ALP-main]: preparing to listen for signals on port[8000]
17:06:03 [ALP-main]: preparing to listen for signals on port[8000]
17:06:03 [CANEs-main]: preparing to listen for signals on port[8000]
17:06:10 [ALP-main]: Created channel [host1_host2] with id [0]
17:06:10 [ALP-main]: Created channel [host2_host3] with id [0]
17:06:22 [ALP-main]: Created CDM-DFP topology
17:06:28 [ALP-main]: Built routing table [DMO-URT] on topology [DMO-DFP]
17:17:08 [CANEs-main]: Starting a new name file
17:17:08 [See CANEs File]: Fetching gff.c from host0
17:17:10 [Atracroute-fed]: ===== Flow 0 [Atracroute-fed] established =====
17:17:20 [Atracroute-fed]: Subscribe filter[0] to chid[0]: Successful subid[34668902]
17:17:20 [Atracroute-fed]: Subscribe filter[1] to chid[0]: Successful subid[34668903]
17:17:20 [Atracroute-fed]: Subscribe filter[0] to chid[0]: Successful subid[34668904]
17:17:20 [Atracroute-fed]: Subscribe filter[1] to chid[0]: Successful subid[34668905]
17:17:20 [Atracroute-fed]: [DMO-DFP.c] matched 2 channels, subscribed 0
17:17:20 [Atracroute-fed]: =====Success Done [3006802]=====
    
```

图4 测试结果

## 网络与通信 Network and Communication

本文介绍了主动网络的一种实现方案——Bowman OS 和 CANEs EE 系统,其中 CANEs EE 基于 Bowman OS,由后者提供执行环境。Bowman OS 为创建虚拟网的抽象拓扑提供了系统支持,并实现了通道、流和状态存储等基本功能。CANEs 的固定部分(驻留在节点上的低层支持程序)对每个分组进行一致处理,而通过分组注入网络的可变部分,是反映用户定义功能的代码,两者之间通过插槽(slot)模型结合在一起构成复合业务<sup>[6]</sup>。Bowman OS 和 CANEs EE 实现了主动网络体系结构中的关键技术,为标准化和实现主动网络体系结构提供了参考。

### 参考文献

[1] 姚晓艳,刘伟.主动网络技术研究综述[J].电脑知识与技术,2009(24):6889-6891.  
 [2] 王珊,倪瑾,唐苑.主动网络中节点操作系统的研究[J].

计算机与现代化,2009(5):99-102.

[3] MERUGU S, BHATTACHARJEE S, CHAE Y. Bowman and CANEs: implementation of an activenetwork[C]. Invited paper at 37th Annual Allerton Conference. 1999.9.  
 [4] ZEGURA E, CALVERT K, MERUGU S. Bowman: a node OS for active networks. Proceedings of IEEE Infocom. 2000.3.  
 [5] 刘雅琴,俞时权.主动网络的应用及具体实现: Bowman 和 CANEs[J]. 计算机工程, 2002, 28(4): 157-158, 231.

(收稿日期: 2010-05-31)

### 作者简介:

吴世嘉,男,1986年生,硕士研究生,主要研究方向:主动网络安全。

朱学永,男,1958年生,教授,硕士生导师,主要研究方向:网络安全。

