主动网络的研究与进展*

任三源，任 勇，山秀明

(清华大学 电子工程系,北京 100084)

E-mail: renfy@sdp.ee.tsinghua.edu.cn

http://www.tsinghua.edu.cn

摘要：目前的Internet体系结构缺乏一定的灵活性,制约了新协议、新应用的开发与应用.主动网络试图通过为节点赋予计算处理能力来改变这一现状.分析和阐述了主动网络产生的背景与动机,并详细介绍了这一新的网络体系结构,讨论了它对网络体系结构研究带来的影响,总结了主动网络的研究现状,最后结合主动网络的应用和在网络整体行为学研究中取得的阶段性成果,给出了一种有新意的应用实例.

关键词：主动网络;协议体系结构;移动代码;"胶囊";端到端观点

中图法分类号：TP393 **文献标识码**：A

快速的发展与变化是目前网络领域的一个显著特点.新的技术和新的应用如雨后春笋般地涌现,并且迄今为止这一趋势仍然没有丝毫减弱的迹象.从技术角度分析,任何新的网络应用模式都得益于功能有所增强、性能有所提高的网络新业务.这是Multicast支持的Web会议、RSVP和InterServ/DiffServ增强的实时应用等诸多实例所充分证明了的,然而,对于网络主流应用模式的预测却是极赋挑战性的.这一点同样可以由人们起初并没有充分认识到E-mail和Web会成为目前Internet上主导应用[1]的事实得到证实.虽然可以肯定地说,Internet的应用必将是丰富多彩的,但没有一个人能够具体描述未来若干年内到底会出现什么样的新应用.因此,将需求各异的用户限制在所谓"最小公分母"业务范围内的设想是不可行的.网络体系结构对新业务应该有良好的适应能力,也就是说,网络所提供的业务应该是灵活的,这一点已经被包括NGI在内的众多相关组织越来越深刻地认识到了[2].

Internet虽然是先进而通用的计算机网络的典型成功范例,但依旧存在不少缺陷,其中最为突出的问题之一就是其体系结构所依赖的沙漏模型(如图1所示)使其缺乏灵活性.瓶颈处的传输层和网络层是各种链路层技术和高层应用程序实现互操作的基础.保持它们的相对稳定性及向后兼容性是必须的,但同时却不可避免地给完善和修改它的工作增添了诸多不便.在RED网关[3]和TCP拥塞控制改进[4]工作中为保证透明性而付出了不少努力.O'Malley早在1991年时就曾经质疑TCP的有效性[5],并提出了一种适宜改进的机制.这里暂且不讨论机制本身的价值,但它能说明人们早已认识到了TCP在适应不断发展变化的网络应用时所处的尴尬境地.网络业务的标准化工作却能保持核心协议的相对稳定,然而考虑多方建议,寻求一致意见,使得标准化工作进程缓慢,在一定程度上制约了网络新业务的及时推广和应用.一种新的业务从原型系统的设计到大规模的应

* **收稿日期**：2001-04-06；**修改日期**：2001-08-20

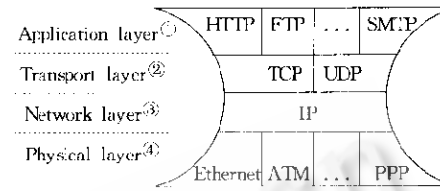
基金项目：国家自然科学基金资助项目(69972040)

作者简介：**任丰源**(1970－),男,甘肃临洮人,博士,主要研究领域为网络流量控制,网络管理,主动网络;**任勇**(1964－),男,黑龙江哈尔滨人,博士,副教授,主要研究领域为人工生命理论及应用,极化雷达,业务量工程;**山秀明**(1945－),男,黑龙江哈尔滨人,教授,博士生导师,主要研究领域为复杂自适应系统,鲁棒控制,Multi-Agent系统.

© 中国科学院软件研究所 http://www.jos.org.cn

用,一般而言,往往需要大约10年的时间[6].资源预留(RSVP)、IPv6、移动IP等就是因此而不能大范围地进入实际应用的最好佐证.

鉴于以上原因,在1994～1995年美国国防部高级研究计划署(DARPA)有关未来网络发展方向的研讨会上,经过多方酝酿,提出了主动网络这一新的网络体系结构.其主要目的是解决现有体系结构在集成新技术、扩展新应用时存在的诸多不便.主动网络继续沿用了存储/转发的结构,一个显著的特点是网络节点上的路由器或交换机可对经过它的分组流进行定制处理.这种定制处理并非传统分组交换网络中为了将分组转发至目的地而对分组头进行的简单处理(如路由器对分组头的修改或标记交换机对标记的修改等).主动网络中的分组除携带净荷数据以外,还有可能携带程序代码,网络内的节点可以在自己的运行环境内执行分组中携带的程序,从而改变网络的行为和状态.这相对于"被动"的转发分组而言要"主动"许多,这也就是之所以称其为"主动网络"的原因.主动网络通过节点的计算能力抽象高层协议,省去了为新应用制定标准的过程,新协议的应用等价于网络分组中新应用程序的开发,通过此加速网络的发展是主动网络赋予网络可编程能力的主要动机.



①应用层,②传输层,③网络层,④物理层.
Fig.1 Sandglass model of network protocol
图1 协议栈的沙漏模型

1 主动网络的体系结构

1.1 主动节点的体系结构

相关组织和机构经过几年的研究与讨论,已经基本上清晰地勾勒出了一个有关主动网络体系结构的框架[7,8].在这个框架中定义了主动网络的组成构件及接口,并对有可能阻碍主动网络发展的安全和性能问题作了特别的考虑.主动网络由节点组成,节点间通过各种低层网络技术(如Ethernet,ATM,SDH和DWDM等)连接在一起,不是要求所有节点都"主动",传统的网络节点依旧可以工作在主动网络环境中.主动节点上运行着一个节点操作系统(NodeOS)[9]、一个或多个执行环境(execution environment,简称EE).用户通过主动应用(active application,简称AA)获得业务.AA是通过EE提供端到端业务的.NodeOS负责分配、调度和管理节点的资源,通过EE向AA提供抽象应用.对于EE而言,NodeOS屏蔽了资源管理的细节和不同EE间行为的相互影响;对于NodeOS而言,EE屏蔽了许多与用户交互的细节.每个EE由一个独立的虚拟机实现,负责解释到达节点的主动分组,不同的EE定义不同的虚拟机.当EE向NodeOS请求业务时,必须附带请求生成者的标识.它可以是EE本身,也可以是用户.NodeOS将请求信息送至安全执行引擎(security enforcement engine),通过检查节点的策略数据库校验请求的真实性,然后视情况来授权请求者接受请求的业务或执行请求的操作.每个节点还有一个管理执行环境(management execution environment).借助它来控制本地节点的相应配置和策略主要的管理和控制功能包括3个方面:维护节点的安全策略数据库;载入新的EE,更新或配置已有的EE;支持远程触发的网络管理业务的实例化等.图2描述了组成

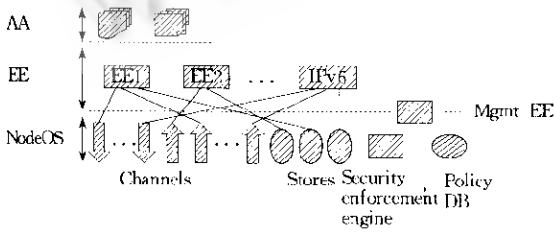


Fig.2 Architecture of active node
图2 主动节点的体系结构

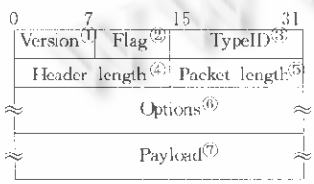
© 中国科学院软件研究所 http://www.jos.org.cn

主动网络节点的主要构件及其相互关系.

节点上的 EE 和 NodeOS 之间存在着一个自然边界,这一接口的通用性是保证主动分组中的移动代码能够正常运行于具有不同实现的 NodeOS 上的前提.通道(channel)、线程、内存和流(flow 被用来定义该接口,它们分别对应于系统的通信、计算和存储资源.通道一般被绑定于 EE 上实现与传输网络的分组交换,道被进一步分为入通道(inChannel)和出通道(outChannel).为了简化没有携带主动应用的分组通过节点时的操作,还定义了直通通道(cut-through Channel),流经它的分组无须 EE 处理.这种模型方便了复杂业务的开发.定义具有层次结构流主要是审计、访问控制和系统调度的需要.典型的流由一对收发消息的通道、线程和内存等系统资源构成.

1.2 主动网络封装协议(ANEP)

为了保证一定的互操性和对现有网络的有效利用,IETF 的主动网络工作组制定了主动网络的帧封装格式[10].如图 3 所示,主动分组是由 ANEP 信头及净荷组成的,信头中 8 比特的版本域用于定义信头格式.8 比特的标记域中现在只有一位有效,用以指示主动节点在不认识 TypeID 域中的信息时是转发分组还是丢弃分组,0 转发,1 则丢弃.TypeID 域的信息表明了处理分组的 EE.TypeID 后面是信头长度和分组长度,以 32 比特为单位.可选项域中每一条纪录由 2 比特的标记位和 14 比特的可选项类型域组成.对于 ANEP 而言,新协议和新 EE 的实现仅仅意味着一个新的 TypeID 或可选项类型,从而使得协议体系结构具有一定的鲁棒性和可扩展性.



①版本,②标志,③类型号,④报头长,⑤分组长,⑥选项,⑦净荷.

Fig. 3 ANEP format
图3 ANEP的格式

Wetherall 和 Tennenhouse 提出了一种称为 ActiveIP 的方案[11],在目前的有些实验系统中得到了应用.该方案是将主动应用程序作为可选项嵌入于 IP 分组的可选项域中.传统的路由器将其视为一般的净荷,而主动路由器却能识别并加以执行.这一机制虽然以较小的开销为网络提供了主动处理的能力,但其扩展性差,很难进一步实施实际应用中所必须的安全策略.而对于 ANEP 而言,扩展它的 Options 域即可解决这个问题.

1.3 安全体系结构

主动网络的安全体系结构应该是可配置且可评估的.这一方面是总结现有 Internet 安全体系结构的经验所得,另一方面是因为主动网络的移动代码机制给安全问题带来了更大的挑战.主动网络安全工作组在安全体系规范草案中充分强调了将安全性结合到主动网络体系结构设计中的重要性[12].有关主动网络安全性的研究大致可以分为 3 类:第 1 类主要研究网络安全中较一般和较普遍的问题,包括加密、签名、认证、授权、密匙的管理和分发、访问控制和安全策略等;第 2 类主要涉及与代码移动属性相关的安全问题,具体讲就是防止主动节点或主动分组受到恶意的主动节点或移动代码的攻击;第 3 类是动态安全问题的研究.它是主动网络实现可扩展性和体现灵活性的必要保证,因为不同的用户、不同的组织、甚至不同的任务都可能需要不同的安全策略和机制,再者,主动网络的大范围应用使得主动分组频繁穿越域边界,这就要求主动网络的不同域之间能够很好地实现互操作,自然主动网络的安全系统也应该支持动态的可互操作的安全策略,从而保证认证、授权和访问控制等一定安全措施的实施.这将成为主动网络安全研究中的一个难点.

在主动网络安全工作组于 1998 年 7 月最后一次给出的安全体系结构中着重考虑了前两类问题,作为节点的安全模型是充分的,但对第 3 类问题的考虑较少.有关这方面的研究工作正在进行之中[13].

© 中国科学院软件研究所 http://www.jos.org.cn

2 主动网络对网络体系结构带来的影响

主动网络为网络赋予了用户可控制的计算能力,使网络节点从比特流的"搬运工"变成了有处理能力的计算引擎[14],使网络应用软件的开发摆脱了现有体系结构下固有模式的制约,加速了新业务、新应用的开发和使用,同时,使得网络硬件设备的升级如同软件的升级一样经济、方便[15].当然也有一种观点认为,主动网络的设计思想违背了协议的分层设计中具有普适性的"端到端观点"(end-to-end argument)[16,17].作为在分布式系统中安放功能模块实现位置的理论基础,"端到端观点"的确是设计许多现代分层协议的指导原则.它主张在分层系统中与应用相关的功能应该提升到最接近应用层的位置来实现,位于较低层的功能实现是冗余的,甚至可能是无价值的,最多有利于性能的增强[18],网络作为分布式系统低层的一部分应该避免最好是在终端系统上实现的功能.但Bhattacharjee 等人认为,"端到端观点"的主旨是考虑应用程序的要求,主动网络将用户可控的计算置于网络之中,及时利用网络本身才可能有的信息,为应用程序提供期望的功能实现.这种依据用户需求剪裁网络业务的能力一方面遵从"端到端观点"的指导思想,另一方面为网络系统的设计者在确定功能模块的实现位置时提供了更多的选择[19].Partridge 和 Strayer 等人则认为主动网络对除了网络层之外的所有层都或多或少地具有潜在的优越性[20].20 年前提出"端到端观点"的Saltzer 和 Reed 认为,主动网络赋予网络的可编程能力一方面与"端到端观点"相违背,因为分层系统的某层实现一种业务或功能的必要条件是该层的所有用户都需要它,并且它应该完全在该层上得以实现;另一方面,可编程能力又可以使网络恰如其分地实现客户所需要的业务,而这一点是与"端到端观点"相一致的.产生这一矛盾的原因在于,主动网络提供的可编程能力是相当广泛的,要想用"端到端观点"对此给出一个通用且明确的结论几乎是不可能的[20].我们认为,"端到端观点"所给出的仅仅是一个分层系统设计的指导原则,它为实际设计中平衡性能和成本、评价设计方案提供了一个依据.用它来判定主动网络体系结构的合理与否并不恰当,但用它去指导和评价每一个具体的主动网络设计细节却是很有意义的.

3 主动网络的研究现状

从 1996 年 DARPA 正式启动主动网络的研究计划到现在仅仅有 4 年的时间.随着近 30 个研究子项目的进行,出现了一些初见端倪的原型系统,取得了不少有价值的研究结论,产生了许多富有创意的应用实例,加之主动网络的思想与目前正处于研究热点的可编程网络(programmable networks)和移动代理(mobile agent)技术存在着某些天然的联系,使得主动网络的研究得到了广泛的关注.从 1999 年主动网络国际工作会议(IWAN)的情况来看,目前有关主动网络的研究工作主要集中在以下几个方面:

(1) 主动网络体系结构和节点体系结构的研究;

(2) 主动网络节点操作系统的设计与实现;

(3) 主动网络平台的实现与评价;

(4) 主动网络中安全问题的研究;

(5) 主动网络环境下的新概念、新应用.

3.1 典型的主动网络平台

3.1.1 基于"胶囊"结构的实现

Tennenhouse 提出了"胶囊"(capsules)[6]这个非常形象的概念来描述分组中封装的小段应用

© 中国科学院软件研究所 http://www.jos.org.cn

程序在网络节点虚机上运行的机制."胶囊"有点类似于移动代理,是通过定制的转发例程来指引它穿越网络的.相应的转发例程是用移动代码技术送至"胶囊"可能经过的网络节点,终端用户在向网络释放"胶囊"时,在分组的相应字节中填写转发例程的类型,当"胶囊"到达主动节点时执行相应的例程加以转发,在传统节点上,仅依据IP分组的头信息转发.MIT的研究小组利用"胶囊"机制开发了ANTS(active node transfer system)[21],并用Java语言给出了一个最基本的实现.因为JVM(Java虚拟机)依赖于主机操作系统,Java API无法直接管理底层的系统资源,所以ANTS Toolkit也就无法实现后来主动节点体系结构中定义的诸如流、通道等概念.其安全性是建立在Java早期的"沙盒"(sandbox)基础上的,所提出的基于MD5算法的签名认证最终也没有实现.再者,Java的执行效率也影响了节点处理"胶囊"的速度.为此,该研究小组进一步开发了基于二进制机器代码的PAN[22],效率有了提高,但却是以牺牲安全性为代价的.纵然如此,ANTS从本质上仍然体现了主动网络的设计思想,以此平台实现的诸多应用实例能够很好地说明这一点.

BBN公司的Smart Packets[23]也是用"胶囊"方法实现的.它引入了一种称为Sprocket的编程语言,目的是为了生成足够紧凑的代码在一个以太帧中完成传输.节点虚机解释运行的代码之前是经过认证的,执行过程中的运行时间也有限制.Smart Packets用于网络管理收到了显著的效果.

宾夕法尼亚大学开发的SwitchWare[24]是另一个主动网络的原型系统.它有3个主要组件:主动分组、交换应用程序(Switchlets)和安全主动网络环境(secure active network environment,简称SANE).主动分组与ANTS中的"胶囊"相似,Switchlets是为节点提供业务的可动态装载的应用程序,SANE在最低层通过鉴别系统组件的最小集合来保证整个环境的完整性.主动分组中的程序代码是用主动网络编程语言(programming language for active network,简称PLAN)编写的.PLAN简单实用,是一种强类型语言,PLAN程序在被注入网络之前要经过静态类型检查,而且限制某些敏感操作,这些都是出于严密的安全考虑.当然,对于PLAN程序的限制可以通过调用Switchlets来弥补.Switchlets由CAML语言写成,与Java相似,CAML的字节代码可动态加载且与机器无关,但比Java的效率高,更重要的一点是,在动态加载代码时提供了有限但却充分的访问控制.SwitchWare的显著特点在于安全方面的考虑与设计,主要问题在于,在处理复杂应用时,PLAN的编程能力显得有些不足,只能通过事先在节点上附加Switchlets的方式来解决,这样就降低了系统的灵活性.

3.1.2 基于硬件的实现

现今路由器技术发展的趋势之一是尽可能地使用ASICS芯片以提高处理标准分组的速度,主动网络节点对分组的处理与具体的应用程序有关.这在客观上会降低节点转发分组的效率,但这却又是ASICS技术无能为力的.华盛顿大学的ARL(应用研究实验室)对他们基于ATM交换结构的Gigabit路由器进行了适当的优化改进,形成了高性能的主动网络节点(active network node,简称ANN)[25].每个ANN由多个可以根据需要扩展的主动网络处理元素(active network processing elements,简称ANEP)组成,每个ANEP上由一个CPU、10万门的FPGA和缓存组成.ANEP通过ATM端口将控制器(ATM port interconnect controller,简称APIC)连接在交换背板上,CPU在线编程FPGA,实现与性能密切相关的处理,FPGA对分组进行常规处理,然后依据是否需要进行附加处理而选择直接送至链路还是交予CPU作进一步处理.负载均衡算法能负责将分组分散到不同的ANEP上进行处理.ANN的结构证明了设计高性能的主动路由器是可行的.

3.1.3 基于规范的实现

乔治亚理工学院和肯塔基大学联合进行的CANE(composable active network elements)[26]项

© 中国科学院软件研究所 http://www.jos.org.cn

目基本上遵从了同期发布的主动网络体系结构规范,设计并实现了自己的 NdoeOS 和 EE. CANE 的 EE 由固定和可变两部分组成. 固定部分为驻留在节点上的低层支持程序,对每个分组进行一致处理. 可变部分是通过分组注入网络中,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.

此外,哥伦比亚大学的 Netscript[27]、DePaule 大学的 SafetyNet[28]、亚利桑那大学的 Liquid Software[29]、U. C. Berkeley 的 Ninja[30]、卡耐基·梅隆的 Darwin[31]等诸多项目都从不同侧面对主动网络的研究与发展起到了推动作用.

3.2 NodeOS 的实现

主动网络早期的研究结果表明,安全和性能等方面的问题使得主动网络无法用现有的标准主机操作系统很好地实现,有关 NodeOS 的研究也就变得较为重要. 目前已有的典型平台有 3 个,分别是 Bowman,Scout 和 Moad.

Bowman[32]是 CANE 的 NodeOS,实现了通道、流和状态存储等基本功能单元,具有模块化及可扩展的体系结构,通信协议、路由协议、队列调度、流处理等模块可以在运行时动态加载. 直通通道、多线程、多处理器的支持有利于提高性能. 它支持多种 EE 和主机操作系统.

Scout[33]是面向通信的操作系统,内核由以模块实现的低层通信原语组成,而模块本身与功能独立,组合在一起可在 I/O 数据流之上建立逻辑通道. 运行在 Scout 之上的 Joust 是一个 Java 虚机(JVM),由一个运行系统和及时编译器组成. JVM 的 API 与 Scout 交互使得应用程序可以访问低层资源. 实验表明,用 Scout/Joust 组合实现的 ANTS 比用 Sun 的 JDK 同现成主机操作系统(如 Linux)组合实现的 ANTS 快两三倍,但它似乎不太适于处理大容量大带宽的业务量.

Moad[34]是 Janos(Java-oriented active network operating system)的 NodeOS. Janos 为运行 Java 代码的主动网络节点提供了一个管理和控制本地资源的平台. Moda 建立在直接与硬件设备交互的 OSKit 之上,实现了 Janos 的安全框架和主动网络工作组定义的 NodeOS 接口规范中低层的业务. Moda 和 OSKit 组合可以为非 Java 实现的 EE 提供一个良好的 NodeOS. 结合上层的 JavaVM 和 Java NodeOS 可以作为 Java 实现的 EE 的 NodeOS.

也有不少主动网络原型系统的 NodeOS 是在已有的操作系统上改进形成的. 比如,ANN 的节点操作系统就是对优化和改进的 NetBSD,国内清华大学计算机科学与技术系在 SED-08 路由器操作系统的基础上改进形成了 TH-AOSR 主动式路由器操作系统[35].

3.3 主动网络的应用

主动网络的产生与发展是应用驱动的结果,分析已提出或已实现的应用,可以大致将它们分为两类,一类是针对现有网络体系结构下无法圆满解决的技术难题,另一类则是新颖别致的网络新应用. 属于前者的主要应用包括利用主动网络实现可靠的多播(multicast)业务[36]、主动拥塞控制[37]、主动的时延抖动控制[38]、用主动业务抵御拒绝性服务攻击(denial of service attack)[39]、主动桥[40]、主动差错恢复技术[41]等. 属于后者的实例有 Web 缓存(Web caching)[21]、Nomadic 路由器[6]、主动虚拟网络管理[42]、实时多媒体编码中的主动业务[43]、主动存储网络[44]、分布式仿真中的主动网络[45]等.

从技术角度分析,主动网络在以下 3 个方面占有得天独厚的优势,好的应用实例都是充分挖掘和利用了它在这些方面的潜力.

(1) 灵活的体系结构简化和加快了新协议、新应用的开发.

© 中国科学院软件研究所 http://www.jos.org.cn

(2) 主动网络中间节点的可处理能力简化了具有内在分布式特性的网络应用协议的设计.

(3) 只有网络中间节点才可准确获取的信息与节点处理能力的结合为网络管理、流量控制、拥塞控制、QoS 的控制与管理提供了更为有效的手段.

虽然第 1 个方面是研究和设计主动网络的初衷与主旨,但分析目前已有的应用实例,后两方面却更受青睐.

4 结束语

当前,基于 TCP/IP 的 Internet 是一种缺乏灵活性的网络体系结构,开发新应用、新业务所必需的漫长标准化工作拖延了它们进入实际应用的进程.主动网络通过为网络节点赋予计算处理能力而使得网络体系结构发生了深刻变化.这种体系结构上的变革实现了最初加快新协议开发进程的初衷,同时也为网络应用的开发提供了最初不曾预想到的便利,但也有人疑虑它违背了分层协议设计中"端到端观点"这一具有普适性的指导原则,进而引起了人们对这一问题的审视.纵然如此,不少主动网络原型系统的研究结果和许多富有创意的应用实例表明,具有灵活性的主动网络体系结构在开发某些网络应用时比目前的网络体系结构具有明显的优越性,并且它潜在的优势也是不容忽视的.目前,有关主动网络的研究主要包括体系结构和应用两个领域.就前者而言,三层框架结构得到了较广泛的认可,研究的热点主要集中在性能和安全两个方面.而有关应用的研究却相当广泛,具有吸引力的往往充分挖掘和展现了主动网络独特的技术优势.比如,在网络拥塞避免和恢复技术的研究中,过去的经验证明,拥塞往往发生在网络内部,因为无法实时控制网络内部节点上的行为,大多数拥塞控制技术都只好在终端系统上寻求解决问题的方法.终端系统和网络内部拥塞节点由于空间距离而引入的 RTT 时间是这类拥塞控制技术客观上无法逾越的障碍,因此,终端系统探知拥塞的发生和控制机制的作用就都存在一定的滞后效应.对于复杂网络系统中瞬息万变的网络状态,这种滞后效应很难使拥塞控制机制做到"有的放矢".如果能利用主动网络的节点处理能力根据节点负载的历史信息对未来的网络状态进行预测,并提前显式反馈于终端,则应该会有比目前的隐式反馈更好的控制效果.再者,对于局部的轻度拥塞,采取类似于 ATM 流量控制中的"后压"也将是一种有效的手段.组播流量控制中因为众多分布的异质异构节点的存在,使得 NAK 消息的融合和差错帧的恢复对于提高资源利用率显得尤为重要,主动网络结点的处理能力应该会在这两个技术上有很好的发挥.我们最近的研究结果表明,计算机网络具有自组织临界现象,系统中大量的组元(网络节点)相互作用,自然地朝着一种临界状态演化,这种临界状态应该是动态情况下网络最有效的状态[46].对于不取决于微观机制的自组织临界状态,是不能期望通过分别处理系统中各个独立子系统来试图优化整个系统性能的,需要对相互作用的网络节点实施协同控制与管理才有可能使网络保持在有效的亚临界状态,而这种协同工作机制是需要网络节点的分析处理能力的.主动网络节点正好具备了这种能力.基于此,我们提出并正在进行主动分布式协同网络管理的研究,期望新的网络管理模式能够克服现有集中式网络管理的缺陷,为网络性能管理增添智能、主动的技术内涵,通过嵌入主动分组中的 Mobile Agent 对网络状态的感知、预测以及它们之间的协商与协作,动态实施流量工程(traffic engineering),从直观上讲,应该比目前流行且被广泛接受的基于 MPLS(multi-protocol label switching)的准静态的流量工程要灵活和有效得多.

References:

[1] Leiner, B., Cerf, V., Clark, D., *et al*. The past and future of the Internet. Communications of the ACM, 1997,40(2):

© 中国科学院软件研究所 http://www.jos.org.cn

13～20.

[2] CRA. Research challenge for the next generation Internet. http://www.whitehouse.gov/WH/EOP/OSTP/NSTC/html/97ann-rpt.html.

[3] Floyd, S., Jacobson, V. Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking, 1993,1(4):397～413.

[4] Stevens, W. TCP slow start, congestion avoidance, fast retransmit, fast recovery algorithm. Request for Comments 2001, 1997.

[5] O'Malley, S., Peterson, L. TCP extensions considered harmful. Request for Comments 1263, 1991.

[6] Tennenhouse, D.L., Smith, J.M., Sincoskie, W.D., *et al*. A survey of active network research. IEEE Communication Magazine, 1997,35(1):80～86.

[7] Calvert, K.L. Architecture framework for active network. 1999. http://www.dcs.uky.edu/～calvert/arch-docs.html.

[8] Calvert, K.L., Bhattacharjee, S., Zegura, E.W., *et al*. Directions in active networks. IEEE Communication Magazine, 1998,36(1):72～78.

[9] Calvert, K.L. NodeOS interface specification. 2000. http://www.dcs.uky.edu/～calvert/nodeos.ps.

[10] Alexander, D.S., Braden, B., Gunter, C.A., *et al*. Active network encapsulation protocol (ANEP). http://www.cis.upenn.edu/～switchware/ANEP/docs/ANEP.txt.

[11] Wetherall, D, Tennenhouse, D.L. The active IP option. In: Proceedings of the 7th ACM SIGOPS European Workshop. 1996. http://www.tns.lcs.mit.edu/publications/sigops96ws.html.

[12] Murphy, S. Security architecture for active nets. 1998. http://www.dcs.uky.edu/～calvert/active-security.ps.

[13] Liu Zhao-yu, Campbell, R.H., Varadarajan, S.K., *et al*. Seraphim: dynamic interoperable security architecture for active network. In: Proceedings of the IEEE OPENARCH2000. 2000.

[14] Wetherall, D., Legedza, U., Guttag, J. Introducing new Internet services: why and how. IEEE Network, 1998,12(3): 12～19.

[15] Tennenhouse, D.L., Wetherall, D. Towards an active network architecture. Computer Communication Review, 1996,26(2):21～28.

[16] Stalling, W. High Speed Network Design and Analysis. New York: Prentice-Hall, Inc., 1997.

[17] Saltzer, H., Reed, D.P., Clark, D. End-to-End argument in system design. ACM Transactions on Computing System, 1984,2(4):277～288.

[18] Bhattachrjee, S., Calvert, K.L., Zegura, E.W. Active networking and end-to-end argument. In: Proceedings of the International Conference on Networking Protocol. 1997.

[19] Bhattacharjee, S., Calvert, K.L., Zegura, E.W., *et al*. Commentaries on 'Active Networking and End-to-End Argument'. IEEE Network, 1998,12(3):66～71.

[20] Wetherall, D. Active network vision and reality: lesson from a capsules-based system. Operating System Review, 1999,34(5):64～79.

[21] Wetherall, D. Service introduce in an active networky [Ph.D. Thesis]. Massachusetts Institute of Technology, 1999.

[22] Nygren, E. PAN: a high performance active network node supporting multiple code system. In: Proceedings of the 2nd Conference on Open Architecture and Network programming. 1999.

[23] Schwartz, B., Jackson, A.W., Strayer, W.T., *et al*. Smart packets for active network. 1998. http://www.net-tech.bbn.com/smtpkts/smart.ps.gz.

[24] Alexander, D.S., Arbaugh, W.A., Michael, W.H., *et al*. The SwitchWare active network architecture. IEEE Network, 1998,12(3):29～36.

[25] Decasper, D.S., Plattner, B., Zürich, E.T.H., *et al*. A scalable high performance active network node. IEEE Network, 1999,12(3):8～19.

[26] Merugu, S., Bhattacharjee, S., Chue, Y., *et al*. Bowman and CANEs: implementation of an active network. In: Proceedings of the 37th Annual Allerton Conference. 1999.

[27] Yemini, Y., Silva, S. Towards programmable networks. In: Proceedings of the IFIP/IEEE International Workshop on Distributed Systems: Operations and Management. 1996.

[28] Jeffrey, A. SafetyNet: designing an object-oriented language for network programming. http://klee.cs.depaul.edu/an/.

© 中国科学院软件研究所 http://www.jos.org.cn

[29] Liquid software. http://www.cs.arizona.edu/liquid/.
[30] Gribble, S. D., Welsh, M. von Behren, R., *et al.* The Ninja architecture for robust Internet-scale systems and services. http://ninja.cs.berkeley.edu/.
[31] Gao, Jun, Steenkiste, P., Takahashi, E., *et al.* A programmable router architecture supporting control plane extensibility. http://www.cs.cmu.edu/~darwin/.
[32] Merugu, S., Bhattacharjee, S., Zegura, E. W., *et al.* Bowman: a node OS for active networks. In: Proceedings of the IEEE INFOCOM2000. 2000.
[33] Joust: a Java OS in scout. http://www.cs.princeton.edu/nsg/joust.html.
[34] Janos, architecture. http://www.cs.utah.edu/flux/janos/arch.html.
[35] Ma, Hong-jun, Zhang, Yao-xue, Chen, Hua. Design and implementation of Tsinghua active operating system for router. Journal of Software, 1999,10(6):580~583 (in Chinese).
[36] Li, Wei, Lehman, H., Garland, S. J., *et al.* Active reliable multicast. In: Proceedings of the IEEE INFOCOM'98. 1998.
[37] Faber, T. ACC: using active networking to enhance feedback congestion control mechanisms. IEEE Network, 1998,12(3):61~65.
[38] Bush, S. F. Active jitter control. http://www.crd.ge.com/people/bush.
[39] Van, C. A defense against address spoofing using active network [MS. Thesis]. MIT, 1999.
[40] Alexander, D. S. Active bridge. In: Proceedings of the SIGCOMM'97. 1997.
[41] Active error recovery. http://www.tascnets.com/panama/AER/index.html.
[42] Bush, S. F. Active virtual network management protocol. http://www.crd.ge.com/people/bush.
[43] Amir, E. McCanne, S., Katz, R. An active service framework and its application to real-time transcoding. In: Proceedings of the ACM SIGCOMM'98. 1998.
[44] Active storage network. http://www.ece.cmu.edu/~asn/research.html.
[45] SANDS: specialized active network technologies for distributed simulations. http://www.tascnets.com/sands/.
[46] Yuan, Jian, Ren, Yong, Shan, Xiu-ming. Self-Organized criticality in a computer network model. Physics Review (E), 2000,61(2):1067~1071.

附中文参考文献:

[35] 马洪军,张尧学,陈桦.主动式路由器操作系统 TH-AOSR 的设计与实现.软件学报,1999,10(6):580~583.

Research and Development of Active Network*

REN Feng-yuan, REN Yong, SHAN Xiu-ming

(*Department of Electronics Engineering, Tsinghua University, Beijing* 100084, *China*)
E-mail: renfy@sdp.ee.tsinghua.edu.cn
http://www.tsinghua.edu.cn

Abstract: The current Internet architecture lacks flexibility. The new network services are always postponed by tedious standardization process. The active network makes change through adding computing to network node. In this paper, the background and the motivation of designing active network are described, the new architecture is presented in detail, its effect on network architecture is discussed, and the current researches about active network are summarized. Finally, combining with the conclusion of network dynamics, a novel application based on active network is put forward.

Key words: active network; protocol architecture; mobile code; capsule; end-to-end argument

* Received April 6, 2001; accepted August 20, 2001
Supported by the National Natural Science Foundation of China under Grant No. 69972040

© 中国科学院软件研究所 http://www.jos.org.cn