

文章编号:1001-9081(2005)02-0252-04

FARA:一种重组地址的体系结构

张丽^{1,2},余镇危¹,张英³,皇甫鑫霖¹

(1. 中国矿业大学研究生院,北京 100083;2. 河南理工大学应用数学与信息科学系,河南焦作 454000;

3. 中国科学院计算技术研究所,北京 100080)

(Zhangli_lily@sina.com)

摘要:提出了一种新的网络体系结构的抽象模型“FARA”,它基于终端命名与网络地址分离的机制,避免引入新的全局域名空间,具有一定的灵活性、通用性和安全性。这一新的命名和绑定的概念组织形式为新一代网络体系结构的研究提供了非常有价值的方向。

关键词:体系结构;实体;关联;约会;移动性

中图分类号:TP393.02 **文献标识码:**A

FARA: an architecture of reorganizing the address

ZHANG Li^{1,2}, YU Zhen-wei¹, ZHANG Ying³, HUANGFU Xin-lin¹

(1. Graduate Student College, China University of Mining and Technology, Beijing 100083, China;

2. Department of Applied Mathematics and Informatics Science, Henan Polytechnic University, Jiaozuo Henan 454000, China;

3. Institute of Computing Technology, Chinese Academy of Science, Beijing 100080, China)

Abstract: Based upon the decoupling of end-system names from network addresses, FARA provides considerable generality, flexibility and security with a range of assurance levels by avoiding the introduction of a new global namespace. The new concept of naming and binding presents an all-important direction to new generation network architecture.

Key words: architecture; entity; association; rendezvous; mobility

0 引言

FARA (Forwarding directive, Association and Rendezvous Architecture)模型是The NewArch project项目中的一项目(是由DARPA资助,USC/ISI, MIT LCS和ICSI合作进行的)。FARA模型定义的是一种体系结构类,通过添加特别的前提条件和机制就能得到各种特定的体系结构,又称为FARADS architecture (Forwarding directive, Association, Rendezvous and Directory service architecture)。

众所周知,IP地址不但表示网络的位置而且表示节点标识^[1]。这使得IP地址既有优点也有缺点。简单地说,网络地址可以提供一些(最低限度的)安全,但是使得灵活性不大。相反,网络位置和实体标识分开能使体系结构更加灵活,但却给安全性带来了困难。FARA模型的一个主要思想就是把应用实体和网络层转发机制相分离,避免引入新的全局域名空间,从而保证一定的安全,避免了IP地址的两难局面。

1 概念介绍

FARA中host-to-host的通信是由称为“关联”(association)的逻辑连接的成对的实体(entity),通过通信底层进行分组交换,相互通信。

1.1 实体

实体是应用的通称,即网络通信的终端,可以是一个进程,进程中的一个线程,一组进程,一台计算机,甚至计算机群等。实体包括应用程序状态和连接状态,是移动的单元。实体移动时携带应用状态和通信状态。

FARA没有对实体定义全局名称。FARA潜在的思想就是与远程实体的通信,只需要能向它送过去分组就行了,没有必要知道它的名字。所以要想和一个实体通信,可通过某种方法如通过给此实体传送数据包的FD来确定实体的位置。

1.2 关联

关联是实体间的通信状态和在实体间传输的分组不间断序列的组合。实体间通过关联这种逻辑连接相互通信。通信状态在关联的生存期内更新,和关联同步。每个分组仅属于一个关联,一个实体可以有多个当前的关联。

FARA中每个分组携带一个关联ID(Aid),利用关联ID,接收实体可以多路分解信息到它的多个关联。FARA对于关联没有全局域名。在目标实体中一个关联ID(Aid)必须是唯一的。对于每个实体,Aid是本地的,在每个终端端点都有一个不同的Aid,所以每个分组携带一对Aid(目的Aid,源Aid)。关联和当前体系结构中的传输层的连接大致相似(目前FARA模型仅限于成对实体之间端到端的通信)。

收稿日期:2004-07-27;修订日期:2004-12-12 基金项目:国家博士点基金资助项目(20030290003)

作者简介:张丽(1973-),女(回族),河南信阳人,硕士研究生,主要研究方向:新一代网络体系结构、OVERLAY网络和流媒体;余镇危(1942-),男,上海人,教授,博士生导师,主要研究方向:新一代网络体系结构、OVERLAY网络;张英(1951-),女,河北献县人,研究员,主要研究方向:信息系统网络设计;皇甫鑫霖(1980-),男,江苏徐州人,硕士研究生,主要研究方向:新一代网络体系结构、OVERLAY网络。

1.3 通信底层

实体由底层系统支持,包括操作系统和网络。通信底层代表关联传输数据。通信底层只提供无连接的服务,所以实体负责端到端的可靠性。可选用的机制包括传统的逐跳转发(hop/hop delivery)、显式(源)路由或者标签交换等。

当一个实体由关联发送分组时,它在分组上加一个头域,称为目标转发指令(FD),再把分组转交到通信底层。目标FD包含分组最终转发到目标实体所需的信息。除了包括目标FD,分组也可包括一个应答FD,用来向源端实体发送一个返回分组。FARA中的分组并不是传送到节点或者目的主机协议栈,而是传送到关联着的实体。传送受网络以及目标实体运行的操作系统环境的影响。一旦分组被传送到目标实体,便解释目标AId,找到相应的关联状态。

FARA模型把通信底层的转发机制和实体执行的端到端的通信功能分离开。比如用一条界线来划分,通信底层在“界线以下”进行运作,而实体以及其关联是在“界线以上”运作。对应“界线”,FARA模型定义一个接口规范(API),保证转发功能和实体的模块化分离。分离的好处,就是可以自由地改变某个特殊关联的分组转发路径,即改变实体的逻辑“连接点”。这种自由就是移动性,而移动性是当前体系结构所缺少的^[2]。FARA有相应的机制来保证每个实体都维持最新的FDS,从而定义它关联的转发路径。

2 完整的FARA模型

2.1 创建关联

两个实体A和B建立关联包括几次握手:A首先发送给B一个初始化信息,A必须有一个FD到达B。从A到B的后序分组携带一个关联AId(第一个分组不携带)。创建关联需要两个特别的FARA组件:约会机制和FARA目录系统。

2.1.1 约会机制

A发送到B的第一个分组不包含目标AId,而包含一个约会信息(RI)字符串,B用这个字符串来创建关联,并且分配一个AId。而且,如果这是客户机和服务器之间建立的第一个关联,第一个分组就发送一个特定的FD到调度后台程序。RI字符串接着就提供给后台程序所需要的所有参数(例如,服务名称),用来启动实体,创建关联。以上假定约会发生在目标系统(服务器)上。若约会点是某些中心位置的代理^[3],则代理收到初始分组后,重写初始FD,指向正确的目标实体,或另一个约会代理。

约会机制有查找和初始化两个部分。查找将返回一个FD和一个RI字符串。初始化部分即:一旦(FD,RI字符串)被找到后,用握手建立关联。

2.1.2 FARA目录服务

FARA模型把查找过程归入“目录服务”。查找部分通常可由各种高层机制完成,如类似域名服务器(DNS)的目录系统,Web站点和其他的程序。约会过程要最终辨别所有供选择的(FD,RI字符串)。查找过程返回的一对(FD,RI字符串)能够直接被初始化实体所使用。复杂情况下,初始系统将这一对数据进行变换,得到完整的FD和最终RI字符串。M-FARA就是使用这种机制。

2.2 通信底层机制

2.2.1 分组转发

底层的基本功能就是无连接(尽力)的分组转发。因此,它对应当前体系机构中的网络层。FARA模型允许各种转发机制,在网络的不同部分,甚至同一个节点上共存。不同的机制在移动性、身份标识以及匿名登录上有不同的权衡。

SLOT用来表示FD把分组传到哪里,以及实体位置所在地。实体当前正占有目标系统中一个特殊的SLOT。因为FD传输到SLOT,所以一个SLOT就是实体和网络拓扑间的一个逻辑连结点。

2.2.2 转发指令FD(Forwarding Directive)及其管理

FD告诉网络如何传递信息到实体。FD源于网络拓扑结构,可以由一系列子FD构成总的源路径,每一个子FD在一定作用域内起作用。特殊情况下,一个FD有一个网络部分和一个本地转发(SLOT ID)部分。网络部分控制把数据包传送到包含实体的节点协议堆栈中,而局部转发部分完成传送到指定的SLOT。这与当前体系结构中的(IP地址,端口号)相似。

通信底层“FD管理”机制提供FD管理,控制通信。当实体移动时,能改变逻辑连结点,并更新FD,跟踪并记录这些改变。实体移动时必须能够通知FD管理,FD管理也需要通知实体新的FD。

2.2.3 资源控制

分组消耗资源,这一点体现在拥塞、拥塞控制以及QoS要求上。FARA模型中的拥塞/QoS报头在分组对网络不透明的部分中(例如,和FD在同样的报头部分中),有一个接口允许实体得到或释放这个报头。

2.2.4 网络层安全

通信底层易受资源攻击:窃取服务或者拒绝服务等。需要“界线以下”的安全措施如:隧道模式IPsec、QoS的认证准入控制等。

2.3 协议堆栈

一般的分组报头:“界线以下”的基本转发功能由一个网络层报头控制,携带目标FD和源FD。它还可包括和IP相似的数据报分段存储机制和循环控制机制。“界线以上”有一个或几个协议层,一起称为关联层。关联层必须携带目标AId和源AId,还可携带关联状态信息,和为了源端检验或/和确认所需的安全信息。

2.4 系统模型

图1表示了FARA终端的SLOT中的一个实体。图中底部的虚线框内表示在终端系统中的底层操作系统,其功能是用FD把数据包传递到SLOT。

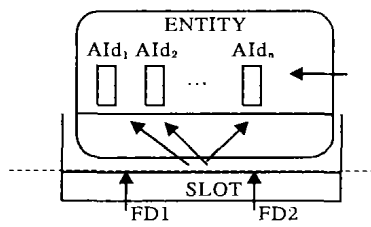


图1 FARA终端系统

一个实体开始服务后,其执行步骤如下:实体编码将请求分配一个SLOT,接着查询本地路由子系统(通过FD管理器)得到自己的FD(或者是FD片断,由此计算出完整的FD),其

中包含了它自己的 SLOT ID。如果实体提供了一个 FDS 中分类的服务,实体将注册一个(服务名和 FD 相映射)与 FDS 对应。这样,实体必须向内核请求,根据 FDS 的 FD,与 FDS 创建关联,并且发送注册请求。

3 实例:M-FARA

M-FARA 是在 FARA 体系结构基础上,定义具体机制,专门研究 FARA 在移动性和路由方面的功能。

3.1 M-FARA 中的网络寻址

M-FARA 不采用全局唯一的地址空间。M-FARA 假定有许多域,每一个域都是一个独立的地址空间(与 NAT Boxes 使用的私有地址空间相似),M-FARA 的 FD 包含子 FD 的总的源路由。

当实体移动到一个新的位置时,它必须为每一个存在关联的另一端计算一个新的 FD。这就意味着,路由子系统或者其他的子系统必须能够把原位置有意义的多个 FD 转换成在新的位置有意义的 FD。如果网络是由一层平铺网络组成的私有域空间,那么 FD 转换就会相当的复杂。为了避免这种情况,M-FARA 采用一个两层地址域的等级结构,如图 2 所示。上层是一个唯一的全球地址域,称为核心域。端到端的 FD 由两个 FD 片断组成:(FDup,FDdown),FDup 把分组传输进入核心域,而 FDdown 从核心域中把分组发送到正确的私有路由域。在数据包转发到达目的实体过程中,FDdown 始终保持不变,又称为规范(canonical)路由。在发送端实体向目的实体时,已知 FDdown,路由子系统在本地计算出 FDup,FDup 和 FDdown 一起构成了端到端的 FD。DNS 就执行匹配的功能。这种方法避免了转换的复杂性,使 M-FARA 结构更加灵活。

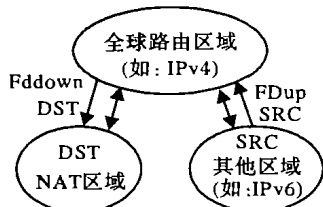


图2 M-FARA 的地址域等级结构

3.2 M-FARA 中的 FD 维护

M-FARA 使用 M-agent(移动代理)来进行实体移动时的 FD 管理。M-agent 作为移动实体和其他实体通信的约会点,也是更新 FD 的第三方。每个移动服务器实体都有一个相关联的 M-agent(它是不移动的特殊实体,且有一个可全球路由的 FD)。服务器在开始就向 M-agent 进行注册,M-agent 的 FD 反过来也会在目标实体的 FD 中的 fDS 被注册,使得客户机能够找到移动的实体。当实体移动时(改变了 FD),就会通知它的 M-agent 跟踪实体的位置。实体将带有更新过的应答 FD 的分组发送到和其有关联的远端实体,通知远端实体它的位置,从而确保 M-FARA 很好的支持动态移动性。

M-agent 同时具有“线上”和“线下”功能。当收到一个分组想要访问它管理的一个实体时,M-agent 重写分组目标 FD 以指向实体,并且重新发送分组,或者给源实体发送一个重定向消息,指定新的 FD。

3.3 M-FARA 中的关联

M-FARA 支持四种关联:简单的,连结的,移动的和可靠

的关联,四种关联在协议层上的复杂性逐渐递增,如图 3 所示。简单的关联在功能上类似于 UDP,是无序的、不可靠的、没有验证的信息。有连结的关联也是不可靠的、无序的,但是它包括了一次用来建立和销毁关联的握手,而且支持验证。移动的关联是在有连结的关联基础上建立的,目的是提供透明传输,在每次移动后或者在远端实体的状态不确定的时候进行验证,并保持同步。可靠的关联由 FARADS Transport (fTP) 提供,是建立在移动的关联基础之上的,它包括可靠性和有序性,类似于 TCP。fTP 头在语法和语义上与标准的 TCP 头一致。

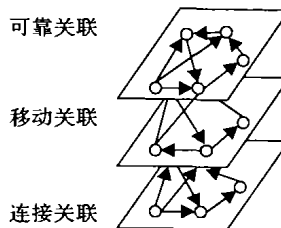


图3 M-FARA 协议层

3.4 M-FARA 中的源端验证

位置和标识的分离引入了安全问题,同时为了降低计算复杂度,M-FARA 采用了相对简单的安全机制,它在创建关联的初始分组交换时进行验证,并且在移动之后,或者其他的一些事件关系到远程的状态,而远程的状态又不会验证每个分组的时候,它就进行再验证。它使用 DCCP - nonce^[4] 作为验证(Reauthentication)协议。

协议规定了要验证的资格和为此需要在端点建立的状态。用一个 challenge 向远处端点提出验证资格的要求,对端用 challenge response 来响应。若一个端点的端 FD 或目的 FD 发生变化(如端点移动),或者上一组数据交换后过了阈值时间,都会触发 challenge 要求。图 4 中有一个固定实体、一个移动实体和一个代理,其中移动实体进行了移动,代理执行转发模式。移动实体首先更新代理(agent),并把相应的资格发送给固定实体。图 4(a)、(b) 分别是移动实体和固定实体发起的验证要求示意图。DCCP-nonce 协议保证了 M-FARA 必要的安全性。

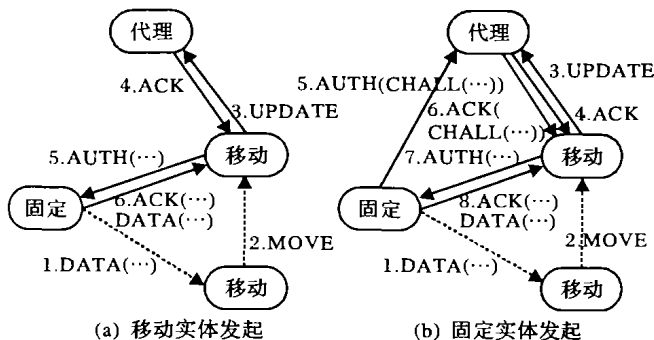


图4 验证要求示意图

3.5 M-FARA 的原型

M-FARA 原型实现目的是测试 M-FARA 在网络上确实会提高其灵活性以及可扩展性。原型在用户层实现,采用 kernel-based 的实现方法,用 FreeBSD 作为实验平台,采用 IP 连接。

本原型使用 C++ 语言写成的,用 Unix 进程作为实体,多个模拟的 M-FARA 主机相互连接起来构成了 OVERLAY 网

络,并通过 UDP 封装进行通信,如图 5 所示。为了实现方便,像给 SLOT 传递信息这样的低层终端系统功能在用户空间由 Unix 进程实现,

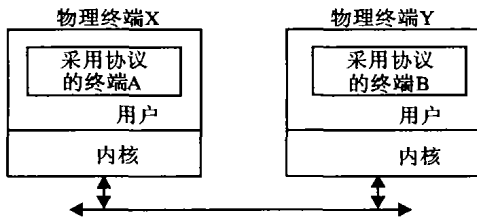


图5 M-FARA网络

这就是 FARA 内核或者 fKernel。实体/fKernel 的“系统呼叫”接口(slot 接口)是通过 Unix IPC 来实现的。这样,一个模拟的 M_FARA 终端系统就由一个 fKernel 进程和实体进程(零个或多个)表示出来了。

IDS 没有实现,但是它也可以由一个 Unix 进程来表示。原型所支持的 M-FARA 关联类型在 4.4 节已经明确了。它支持传统的逐跳转发,而且源路由使用 IPv4 和 IPv6 地址。特别指出,FD 的形式如下: (HopFD1, HopFD2, ..., SlotID), 这里每一个 HopFD 包含一个 IPv4 和 IPv6 地址,而 SlotID 是在目标系统中被 Kernel 使用的来发送给目标实体。

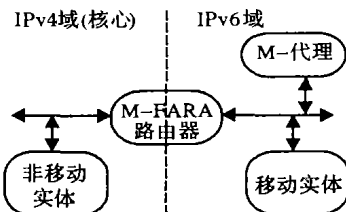


图6 架构的 M-FARA 原型

图 6 是对原型的实验化的架构。这里是一个固定的实体和一个移动的实体进行通信。有一个 M-FARA 路由器连接 IPv4 和 IPv6 区域,实体间建立一个可靠的关联,当移动实体从一个域移动到另一个域时,这个关联仍旧存在而且进行重新验证。M-agent 可以在其中任何一个域之内(尽管它是不可移动的)。实际上,已经通过域中存在的多个移动实体,和相应的向上、向下的逻辑接口,模拟了移动性。下面以简单关联(simple_association)为例,给出实现模块,具体细节见文献[5]。

```
class simple_association : public association_base
{
protected:
enum { BUFLen = 8192 }; //定义 message 长度
static message_queue q; //所有简单关联共享队列
//下面从 slot 读出信息,创建一个 message,并排入队列,若 slot
//报错,就丢弃信息.
virtual void process_message() throw(slot_exception)
{
ip_forwarding_directive fd;
char pkt[BUFLen];
int len;
if((len = s.recv(pkt, BUFLen, fd)) != -1) {
message * m = new message(pkt, len, fd);
q.push(m);
}
else throw slot_exception("Slot receive failed");
}
virtual int send_message(const void *, int)
throw(farads_exception)
{
throw("Cannot send without forwarding descriptor");
}
```

```
} //因为没有目的 FD
//下面将接收数据
virtual int recv_message(void * b, int len) throw(farads_exception)
{
ip_forwarding_directive fd;
return recv(b, len, fd);
}
public:
simple_association(const slot& sl, const string &rv) :
association_base(sl, rv) { }
//下面向关联发送 message, 包括目的 FD
int send(const void * b, int l, const forwarding_directive&fd)
throw(farads_exception) {
while (s.message_waiting()) process_message();
return s.send(b, l, fd);
}
//若队列中已无待处理的信息,则从队列中接受此信息.
int recv(void * b, int l, forwarding_directive&fd)
throw(farads_exception) {
while (q.empty())
process_message();
message * m = q.front();
if (l >= m->len) q.pop();
else throw association_exception("buffer too small");
fd = m->fd;
copy(m->buf, m->buf + m->len, reinterpret_cast<char *
>(b));
delete m;
return l;
}
};
```

4 结语

FARA 模型对体系结构进行模块化:上层是抽象的实体和关联,下层是无连接分组转发的通信低层。FARA 设想在两层之间有一条 API“界线”。其目的是把位置和标识清楚的区分开,既为一般的移动提供支持,也支持两层相互独立的进化机制。M-FARA 在 FARA 体系结构基础上定义了具体机制,专门研究 FARA 在移动性和路由方面的功能,不仅为 FARA 体系结构建构了一个实现平台,而且表明了存在一个 FARA 的实例能够提供显然超越当前互联网体系结构的功能。但是,由于可行性选择和时间限制,还需要更深一步的研究 M-FARA(还有许多功能需要完善),继续开发其他的可能根据 FARA 派生的有用的实例。

参考文献:

- [1] POSTEL J. Internet Protocol[S]. RFC 791, September 1981.
- [2] CLARK D, BRADEN R, FALK A, *et al.* FARA: Reorganizing the Addressing Architecture. MIT Laboratory for Computer Science 200 Technology Square Cambridge [EB/OL]. <http://www.isi.edu/newarch, 2004-07>.
- [3] STOICA I, ADKINS D, ZHUANG S, *et al.* Internet Indirection Infrastructure[A]. Pro ACM SIGCOMM 2002[C], 2002. 73-86.
- [4] KOHLER E, HANDLEY M, SHENKER S. Datagram Congestion Control Protocol (DCCP). Datagram Congestion Control Protocol (DCCP) [EB/OL]. <http://www.icir.org/kohler/dccp/draft-ietf-dccp-spec-09.txt, 2004-11-14>.
- [5] PINGALI VK, FALK A, FABER A, *et al.* FARADS Prototype Design Document (M-FARA prototype) [EB/OL]. <http://www.isi.edu/newarch/NewArchProjectFuture-GenerationInternetArchitecture.htm, 2004-04-15>.
- [6] SALTZER J. On the Naming and Binding of Network Destinations [A]. In Local Computer Networks, North-Holland Publishing Company[C], 1993. 311-317.