

基于移动代码的网管策略性能的分析

李 捷¹, 刘先省¹, 皇甫伟²

(1. 河南大学计算机与信息工程学院, 河南开封 475001; 2. 中国科学院软件所, 北京 100080)

摘要: 针对移动代码在网管中的三种应用机制, 以网管操作的响应时延和产生的数据流量作为指标分别给出了定量模型. 从中归纳出了影响性能的关键要素: 网络规模、网管站带宽资源以及分布式计算平台. 仿真试验结果表明, 在网络规模较大的情况下, 采用移动代理的网管系统表现出较其它两种机制较小的数据流量和操作延时, 可以很好地应用于大规模网络的管理.

关键词: 分布式计算; 网络管理; 性能

中图分类号: TN913.24 **文献标识码:** A **文章编号:** 0372-2112(2005)07-1323-04

Towards Evaluating the Performance of Network Management Strategies Based on Mobile Code

LI Jie¹, LIU Xianning¹, HUANGFU Wei²

(1. College of Computer & Information Engineering, Henan University, Kaifeng, Henan 475001, China;

2. Institute of Software, Chinese Academy of Science, Beijing 100080, China)

Abstract: This paper is committed to compare the performances of three kinds of mobile code based network management system. The quantitative models are presented to analyze the major factors that affect the response time and traffic, such as the number of managed elements, the bandwidth of network management station. These factors are evaluated with a variety of experiments. The simulation results shows that the mobile agent based network management approach can response more quickly and produce less traffic in large scale network and thus can be more scalable than other two mobile code based methods.

Key words: distributed computation; network management; performance

1 引言

随着网络自身的大型化、异构化和分布化的发展趋势, 作为管理网络的工具——网管系统应该适应这一发展. 而传统的集中式网管模式由于其固有的、本质的缺陷而无法适应网络的分布式发展要求, 解决问题的关键在于实现网管系统的分布化^[1]. 我们认为这种分布化不仅表现在直观上的地理分布性, 更应体现于内在的计算分布性, 基于移动代码机制的网管策略正是在此背景下应运而生. 移动代码的计算思想可以溯源到远程脚本程序发送和远端批处理^[2]. 文[3]给出了通用意义上的移动代码语义, 它包含基于 PUSH 模式的远端执行 (RPC, Remote Process Call)、基于 POP 模式的按需代码 (COD, Code on Demand) 和移动代理^[4] (MA, Mobile Agent). 文[9]根据计算代码的获取机制, 将网管环境下应用的分布式计算策略同样划分为 RPC、CoD 和 MA 三种模式. 其中前两种技术已在 CORBA 中得到了成功的大规模工业应用^[4]. 而对于网管环境下的 MA 应用, 目前尚处于研究阶段^[5]. 显而易见, 作为管理网络的工具, 网管系统的性能, 包括网管操作产生的流量, 以及网管操作的响应时间等是至关重要的. 目前针对基于移动代码的分布式网管系统性能的研究多限于与集中式网管系统的纵向比较^[6], 在基于各种分布式技术构造的网管系统之间的性能横向比较方面所做工作尚未发现. 文[7]仅针对移动

代理的网管系统响应时间进行了分析. 基于分布式计算的网管策略研究的深入和工业标准的建立^[8], 迫切需要对已有的分布式计算模式在网管环境的性能进行全面的比较分析. 本文致力于此, 通过对现有的几种应用于网管环境的分布式计算机制进行数学建模和仿真试验, 给出了一个较为全面的比较分析结果.

下文结构组织如下: 第二节中对现有的三种应用于网管环境的分布式计算策略进行了概括总结; 第三节中对其数据交换模式分别建立定量模型; 在第四节中构建了原型系统, 并在其上进行了定量分析; 最后在第五节给出了一个全面客观的结论和下一步的工作展望.

2 相关工作

网络管理环境下, 分布式计算的引入是通过代码迁移来实现的. 具体策略为: 网管站把管理代码和控制逻辑封装后迁移至被管理网元处执行并将结果返还给网管站. 由此可见, 分布式计算是通过将代码引入数据端来实现的. 本节将对 RPC、CoD 和 MA 三种模式在网管应用的流程进行简单总结. 假设运行环境为管理站节点 L1 和被管理网元节点 L2, 则在此环境下三种分布式计算策略的数据交互流程分别概括如下.

2.1 RPC 模式下数据交互流程

在 RPC 模式中, 一个完整的数据交互过程依次需要如下

收稿日期: 2003-11-28; 修回日期: 2005-02-26

基金项目: 国家自然科学基金项目 (No. 60272024); 河南省自然科学基金项目 (No. 0411014100); 河南省高校杰出科研人才创新工程 (No. 2003KYCX003); 河南省高校创新人才培养工程

几个步骤: ①L1 绑定服务器 L2; ②L1 将执行代码和相关参数定制生成协议数据包; ③数据包经网络传输至 L2; ④L2 接收数据包并还原参数; ⑤L2 运行本地过程; ⑥L2 将运行结果定制生成协议数据包; ⑦数据包经网络传输至 L1; ⑧L1 接收数据包并还原运行结果。

2.2 CoD 模式下数据交互流程

在 CoD 模式中, L2 处预先配置部分网管功能代码(例如使用频率较高的网管功能), 当 L1 发送的网管操作命中, 则在 L2 端执行相应代码, 否则 L1 需重新发送代码至 L2。因此一个完整的 CoD 数据交互过程依次需要如下几个步骤: ①L1 绑定服务器 L2; ②L1 将执行参数定制生成协议数据包, 跳转到 ④; ③L1 将网管代码和参数定制生成协议数据包; ④数据包经网络传输至 L2; ⑤L2 接收数据包判断操作代码是否命中: 不命中则跳转至 ③, 否则 L2 调用相应功能代码; ⑥L2 将运行结果定制生成协议数据包; ⑦数据包经网络传输至 L1; ⑧L1 接收数据包并还原运行结果。

2.3 MA 模式下数据交互流程

MA 是一段独立于操作平台和操作系统的软件实体, 它可以在计算机网络中漫游, 代表用户在不同网络节点上进行交互工作。MA 与前二者相比, 它不仅携带了代码和参数, 甚至可以携带执行状态, 因此其最大的特点在于其移动性和智能性。MA 模式中, 一个完整的数据交互过程依次需要如下几个步骤: ①L1 生成一个移动代理 MA 并定制迁移策略; ②L1 将该 MA 按照移动代理平台规范定制生成数据包; ③MA 迁移至 L2; ④L2 接收 MA 并解析还原其原始信息; ⑤L2 本地运行 MA 代码; ⑥L2 将运行结果定制生成协议数据包; ⑦数据包经网络传输至 L1; ⑧L1 接收数据包并还原运行结果。值得注意的是, 由于当前环境中只有 L1 和 L2 两个节点, MA 的移动性并没有很好的体现出来, 后文将对此情况具体分析。

3 定量模型

本节对 RPC、CoD 和 MA 三种移动代码方式在网管应用中的性能建立数学模型。事实上, 三种分布式计算机制在网管环境下的引入都是基于代码委派管理 MbD^[9] (Management by Delegation) 策略, 因此它们在体系结构和具体实现中具有如下相似之处: (1) Java 语言凭借其优秀的平台无关性和良好的面向对象特征在三种模式下得到了充分的应用; (2) 三种模式下管理站节点和网元节点都应有相应的支撑平台, 特别是对于 MA 而言, 其支撑平台除了具备接收移动代码功能外, 还需要有迁移机制的支持。基于上述分析, 我们对三种分布式计算模式在网管中的应用建立数学模型。

现有多节点网络 N

$= \{L_0, L_1, L_2, \dots, L_{n-1}\}$, 其中 L_0 为管理站节点, $L_i (1 \leq i \leq n-1)$

为被管理节点序列。不

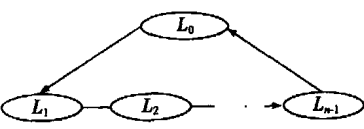


图 1 被管理网络拓扑结构图

失一般性有如下假设: ①被管理网络为环形网络拓扑结构如图 1 所示; ②每个节点具有相同的运算资源; ③每个链路 $L_{i,j} (0 \leq i, j \leq n-1, i \neq j)$ 有相同的网络带宽和网络延时; ④网管

运算的响应报文与采用何种分布式计算模式无关且数据报定制与解析报文速度相同。

3.1 基于 RPC 网管性能定量模型

RPC 模式下 L_0 进行一次网管操作产生的数据流量为:

$$T_{RPC} = T_{Req} + T_{Res} \quad (1)$$

其中 T_{Req} 为 L_0 请求报文数据量; T_{Res} 为 $L_i (1 \leq i \leq n-1)$ 响应报文(运算结果)的数据量。

RPC 模式下 L_0 进行一次网管操作产生的系统延时为:

$$D_{RPC} = 2\varepsilon + \left(\frac{1}{\gamma} + 2\theta_{RPC} \right) T_{RPC} \quad (2)$$

其中 ε 为链路延时, γ 为链路带宽, θ_{RPC} 为 RPC 模式下定制数据包时间与数据量的线性因子。

在 RPC 模式下 L_0 对整个网络进行操作是通过每一个被管理节点发送代码来实现的, 因此 L_0 对全网进行一次网管操作产生的数据流量为:

$$T'_{RPC} = \sum_{i=1}^{n-1} T_{RPC} = (n-1)(T_{Req} + T_{Res}) \quad (3)$$

由式(3)得 L_0 对整个网络进行一次网管操作的响应时间为:

$$\begin{aligned} D'_{RPC} &= \sum_{i=1}^{n-1} \left[2\varepsilon + \left(\frac{1}{\gamma} + 2\theta_{RPC} \right) T'_{RPC} \right] \\ &= (n-1) \left[n\varepsilon + \left(\frac{1}{\gamma} + 2\theta_{RPC} \right) T'_{RPC} \right] \end{aligned} \quad (4)$$

3.2 基于 CoD 网管性能定量模型

在 CoD 模式下 L_0 进行一次网管操作产生的数据流量为:

$$T_{CoD} = T_{Req} + T_{Res} + p(\text{CoD}_{Req} + T_{Code}) \quad (5)$$

其中 p 为所需代码命中失败的概率。 T_{Req} 、 CoD_{Req} 为 $L_i (1 \leq i \leq n-1)$ 向 L_0 发送的请求传送代码的报文数据量; T_{Code} 为 L_0 向 $L_i (1 \leq i \leq n-1)$ 发送的代码报文数据量;

CoD 模式下 L_0 进行一次网管操作产生的系统延时为:

$$D_{CoD} = 2\varepsilon(1+p) + \left(\frac{1}{\gamma} + 2\theta_{CoD} \right) T_{CoD} \quad (6)$$

其中 θ_{CoD} 为 CoD 模式下定制数据包的时间与数据量的线性因子。

在 CoD 模式下 L_0 对全网进行一次网管操作的方法与 RPC 方式完全一样。因此对全网进行一次网管操作所产生的数据流量为:

$$T'_{CoD} = \sum_{i=1}^{n-1} T_{CoD} = (n-1)[T_{Req} + T_{Res} + p(\text{CoD}_{Req} + T_{Code})] \quad (7)$$

由式(7)可推导出 L_0 对整个网络进行一次网管操作产生的相应时间为:

$$\begin{aligned} D'_{CoD} &= \sum_{i=1}^{n-1} \left[2\varepsilon(1+p) + \left(\frac{1}{\gamma} + 2\theta_{CoD} \right) T'_{CoD} \right] \\ &= (n-1) \left[n\varepsilon(1+p) + \left(\frac{1}{\gamma} + 2\theta_{CoD} \right) T'_{CoD} \right] \end{aligned} \quad (8)$$

观察式(7)和式(8), 当 $p=0$ 即代码命中时, 其形式分别近似于式 1 和式 2, 说明了后者与前者本质上的一致性。一般情况下, 考虑到被管理节点节电上的资源是有限的, 因此 CoD 的这种“缓存命中”模式更具有通用性。

3.3 基于 MA 网管性能定量模型

在 MA 模式下 L_0 对一个节点进行一次网管操作产生的

数据流量为:

$$T_{MA} = MA_{data} + MA_{code} + MA_{state} + T_{Res} \quad (9)$$

其中 MA_{data} , MA_{code} 和 MA_{state} 分别为 MA 中的数据部分(运算参数)、执行代码部分和运算状态部分(运算结果)的数据量。

由式(9)可得 MA 模式下 L_0 对一个节点进行一次网管操作的响应时间为:

$$D_{MA} = 2\varepsilon + \left(\frac{1}{\gamma} + 2\theta_{MA} \right) T_{MA} \quad (10)$$

其中 θ_{MA} 为移动代理模式下数据定制时间与数据量的线性因子。

在 MA 模式下 L_0 对整个网络进行一次网管操作时, 采取了与前面两种移动代码机制完全不同的方法, 由于可以对 MA 定制迁移策略, 以及 MA 可在移动过程中携带数据和运算状态, 因此 L_0 只需对一个节点发送一次 MA 就可以实现对全网的管理操作。由于 MA 在一节点完成网管操作后, 携带着本次操作的结果数据按照定制的迁移策略向下一个节点移动, 当 MA 结束全网操作之后将携带所有节点的结果数据返回网管站, 因此产生的数据流量应为:

$$T'_{MA} = T_{MA} + \sum_{i=1}^{n-1} T_{Res} = T_{MA} + (n-1)T_{Res} \quad (11)$$

由式(11)可得, L_0 对整个网络进行一次网管操作产生的系统时延为:

$$D'_{MA} = n\varepsilon + \left(\frac{1}{\gamma} + 2\theta_{MA} \right) T'_{MA} \quad (12)$$

4 仿真试验及分析

4.1 仿真环境

我们利用 NS2^[11] 实现了本文的仿真工作。文献[12]设计并实现了一种 NS2 移动代理网络功能扩展方法。我们在此基础上对 NS 进行功能扩展实现了三种移动代码机制网络功能的仿真。根据移动代码的功能需求以及 NS2 的现有功能和结构特点, 我们从如下两个方面对 NS2 进行了扩展: 1. 对 NS2 原有的数据报(Packet)结构进行了扩展, 使其能够装载具体的移动代码报文; 2. 修改原有的节点结构, 当判断收到一个移动代码类型的数据报时, 应将其传递到该节点的执行环境中运行。我们按照图 1 生成了仿真网络的拓扑结构。下面分别对前文建立的定量模型进行理论分析, 并通过仿真试验进行验证。

4.2 RPC 与 CoD 的性能比较

对式(1)~(8)进行分析可知, 当 p 为 0 时, CoD 模式具有与 RPC 模式及其相近的性能定量模型, 即在命中效率较低的情况下, CoD 模式可近似转换为 RPC 模式。我们认为, CoD 模式可被视作是一种牺牲部分性能以换取通用性的 RPC 的特定模式。一方面, 在网管功能较少的情况下, 利用 RPC 模式, 将所有网管功能代码赋予被管理网元; 另一方面, 利用 CoD 模式可以提供较为灵活的网管手段。此外, 通过扩大被管理网元缓存, 将使用较频繁的网管功能代码赋予被管理网元来提等手段来降低非命中概率 p , 以提高基于 CoD 模式的网管系统的性能。

我们对此进行了仿真实验, 具体方法为, 管理站节点的出

口带宽和被管理网元之间的带宽 γ 都被设置为 2Mbps, T_{Req} , T_{Res} 被设置为 1.5kbyte, 网络延迟 ε 为 10^{-3} s, T_{Code} 为 0.5kbyte, θ_{RPC} 和 θ_{CoD} 被设置为区间上 $[0.2, 0.4]$ 的平均分布。在管理站节点定制网管策略分别对不同数量的网元进行管理操作。在管理站节点测量出口带宽处的流量以及每次操作从发起到结束的响应时间。每次操作重复进行 100 次, 对试验采样结果取平均值。其结果如图 2 所示。

图 2(a) 和 2(b) 中, 无论是响应时间还是流量都随着被管理节点个数的增加而增大, 这与前面建立的定量模型是吻合的。另外, 响应时间的曲线几乎是相同的, 特别的当 p 的值较小, 此时命中率较高, CoD 模式表现出较 RPC 模式更好的性能, 即更小的响应时间和更小的流量。随着 p 的增大, 命中率降低, 其曲线与 RPC 的曲线趋紧。鉴于 CoD 的优势就在于预先配置部分网管功能代码从而表现出比 RPC 较高的网管性能, 因此后文中的性能分析比较中, 我们以 $p = 0.5$ 的 CoD 模式代替 RPC 与 MA 模式进行性能的比较分析。

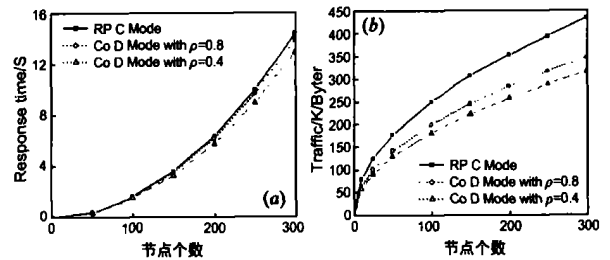


图 2 RPC 和 CoD 网管性能采样图

4.3 网络规模对性能的影响

如前文所述, 本节中我们选择 $p = 0.5$ 的 CoD 模式与 MA 模式进行性能仿真分析

试验验证, 试验参数和试验方法类似节 4.1。由于仿真平台中的所有链路带宽一致, 因此本次试验仅对系统的响应时间进行统计。试验结果如图 3 所示:

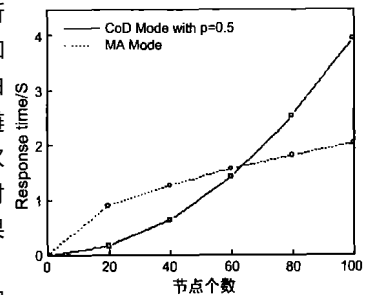


图 3 网络规模对性能的影响

图 3 中, 当被管理节点数目小于 60 的情

况下, CoD 模式比 MA 模式表现出较少的响应时间。当被管理节点数目超过 60 以后, CoD 的响应时间大于 MA, 且增长幅度较明显。而 MA 模式表现了较为平稳的走势。通过比较式 5 和式 9、式 6 和式 10, 可发现 T_{CoD} 和 T_{MA} 是两个重要的指标, 它们不仅代表了在各自模式下的数据流量, 而且在很大程度上也决定了相应模式下的系统延时。由于移动代理的自身要求, 初始状态携带了较 CoD 更多的数据量, 因此在被管理网元节点较少的环境中, 基于 CoD 的分布式计算模式表现出比 MA 模式较高的网管性能。而在被管理网元节点增长的情况下, 再次比较式(7)和式(11)、式(8)和式(12)。可发现, 被管理网元个数 n 已经成为一个不可忽视的重要参数。表现在: 随着网络规模的扩大, T'_{RPC} 整体成倍增长, 而 T'_{MA} 只有部分数据随之

成倍增长;反映在系统延时指标,此情况更为突出, D'_{RPC} 的时间复杂度为 $O(n^2)$,而 D'_{MA} 的时间复杂度保持为 $O(n)$ 。因此我们认为,在网络规模较大的情况下,MA模式逐渐显示出其高性能。相应的,基于RPC模式和CoD模式和C/S架构的数据交互模式比较适合于小规模网络。

4.4 可扩展性测试

网管站的出口带宽一直被认为是网络外管理的瓶颈所在。集中式的网络管理的最大桎梏也在于此。本次试验来验证采用移动代码的集中分布式技术在网管应用中的可扩展性表现。试验方法,被管理节点数最大值设为300, L_0 处的出口带宽分别为2M,4M,8M,其他参数设置与前文一致。试验结果仅对系统响应时间进行测量统计。

图4(a)和4(b)分别对移动代理模式和CoD模式在三种出口带宽下的响应时间进行了统计。总的来说,管理站端的出口带宽越高,则系统的响应时间就越短。从图4(a)可以看出,三条曲线基本是相同的,这说明在MA模式下,出口带宽对响应时间影响不大,不易形成系统的瓶颈。而在图4(b)中,三条曲线较为分散,且相差较大,说明CoD模式对于管理站出口带宽较为敏感,整个系统在管理站端易成为瓶颈。这说明了基于MA的网络管理系统较其他两种技术构建的网管系统具有更好的可扩展性。

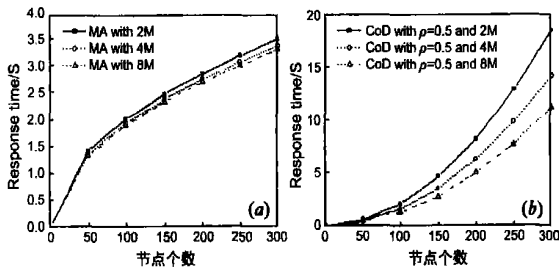


图4 可扩展性测试

5 结论

基于分布式计算模型的网络管理的基础是分布式计算模型,分布式计算模型数据交互的性能直接决定了它在网管中的应用性能。本文从分布式计算的数据交互模式入手,对已有的相关策略进行归纳分类并针对网络管理分别建立其数据交互的数学模型。通过对这些数学模型的定性分析和原型系统试验的定量分析验证,我们认为在特定情况下,RPC模式和CoD模式可视为等同。MA技术较其他两种技术适用于大型网络的管理,这是因为基于MA技术的网管系统具有较好的可扩展性。本文仅考虑了系统响应时间和网络流量等指标。在实际的网络环境中,还有很多性能因素需要加以考虑。只有MA的智能型和安全性得到了充分的体现,MA才能在网管领域真正体现出其较其他移动代码技术无法比拟的功能。

参考文献:

[1] Li jie, WANG Rui chuan. Study of the improvement on network management performance with mobile agent[J]. The Journal of China Universities of Posts and Telecommunications, 2003, 10(3): 11-17.

- [2] Collin G Harrison, David M Chess, Aaron Kershenbaum. Mobile Agents: Are they a good idea? [R]. USA: IBM Research Division, 1995.
- [3] Andrzej Bieszadz, Bernard Pagurek. Network management application oriented taxonomy of mobile code[A]. Proc of the IEEE/IFIP Network Operations and Management Symposium NOMS'98[C]. New Orleans: IEEE, 1998. 659-669.
- [4] 李捷, 郑逢斌, 陈志国. 基于CORBA的网管方案的研究与实现[J]. 计算机应用研究, 21(11):2004.
Li Jie, Zheng Feng bin, Chen Zhi guo. Research and implementation of network management based on CORBA[J]. Application Research of Computer, 21(11): 2004.
- [5] 李治文, 孟洛明, 等. 网络管理环境下移动代理技术应用研究的现状、问题与展望[J]. 电子学报, 30(4): 2002.
LI Yewen, MENG Luoming, QI Feng. The study and perspective of mobile agent applications in network management environment[J]. AGTAE. ELECTRONICA SINICA, 30(4):2002.
- [6] 王汝传, 卞正彪, 张登银. 基于移动代理技术网管模型性能的研究[J]. 通信学报, 2003, 5(24): 49-56.
WANG Rui chuan, BIAN Zheng ai, ZHANG Deng yin, LI Jie. The study of the performance for the mobile agent based network management model[J]. Journal of China Institute of Communications, 2003, 5(24), 49-56.
- [7] 孙普含, 孙玉芳. 基于移动代理的网络管理系统的响应时间分析[J]. 电子学报, 2002, 30(10):
Zhang Pu han, Sun Yu fang. Evaluating response time of a network management system based on mobile agents[J]. ACTAE. ELECTRONICA SINICA, 2002, 30(10)
- [8] Perlikeas M, Chatzipapadopoulos F. An Evaluation Study of Mobile Agent Technologies: Standardization, Implementation and Evolution, Multimedia Computing and Systems[J]. IEEE International Conference, 1999, 2: 287-291.
- [9] Levid D, Schonwaker J. Definitions of Managed Objects for the Delegation of Management Script[S]. IETF RFG 2592, May 1999.
- [10] 何炎祥, 陈华萌. Agent和多Agent系统的设计和应用[M]. 武汉: 武汉大学出版社, 2001.
He Yan Xiang, Chen Sherr Meng. Design and applications of agent and multi agent system[M]. Wuhan: Wuhan University Press, 2001.
- [11] NS2 Manuscript [EB/OL]. The Network Simulator <http://www.isi.edu/nsnan/>
- [12] 李建枫, 糜正琨, 成际镇. NS-2移动代理网络功能扩展设计和实现[J]. 系统仿真学报, 2004, 16(1):
LI Jian feng, MI Zheng kun, CHENG Ji zhen. Design and implementation of NS-2 functional extension to mobile agent network emulation [J]. Journal of System Simulation, 2004, 16(1)

作者简介:

李捷 男, 1975年生, 讲师, 博士生, 主要研究方向为网络管理、分布式计算。

刘先省 男, 1964年生, 教授, 博士, 中国电子学会高级会员, 主要研究方向为信息处理、网络建模。E-mail: liuxianxing@henu.edu.cn.

皇甫伟 男, 1975年生, 讲师, 博士, 主要研究方向为网络及其