



一种 SDN 架构下业务属性相关的多径路由算法

余翔, 易明敏, 杨路

(重庆邮电大学信息与通信工程学院, 重庆 400065)

摘要: 面对当前网络中流量的增长、业务种类的增多, SDN 中多数的路由算法只支持一种 QoS 参数, 没有兼顾对系统调度服务公平性的考虑, 然而多参数限制的 QoS 明显是 NP 难问题, 该问题用普通的路由算法难以解决, 引进蚁群算法, 在蚁群算法的基础上, 将链路的时延、分组丢失率引入蚁群算法中, 作为算法选择路径的依据, 提出一种新的路由算法。该算法在对不同业务属性的数据流分类的基础上, 根据网络的实时状况, 为不同业务属性的数据流选择合适的路径, 对网络中的数据流进行多路径传输。仿真实验表明, 该算法能有效地降低数据流的时延、分组丢失率。

关键词: SDN; 业务属性; 蚁群算法; 多径路由

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2016282

A business attribute correlation multi-path routing algorithm based on SDN architecture

YU Xiang, YI Mingmin, YANG Lu

School of Information and Communication Engineering, Chongqing University
of Posts and Telecommunications, Chongqing 400065, China

Abstract: Currently, most of the routing algorithms support only one QoS parameter, and do not consider the fairness of system scheduling service. Obviously, multi-parameters constrains problem is NP hard problem, it is impossible to solve this kind of problem by traditional routing algorithms. The routing thought from ant colony algorithm (ACO) was introduced. On the basis of the ACO, and combined with the business attributes of traffic flow, a new algorithm was put forward which introduced the delay and packet loss into the ACO, as a dependence of routing. In new algorithm, which according to the real-time status of network, the suitable routing was chosen for different types traffic and forward by multi-path base on the classification of traffic flows. The experiment shows that this algorithm can decrease the delay and packet loss effectively.

Key words: SDN, business attribute, ACO, multi-path routing

1 引言

随着大数据、云计算技术的兴起, 互联网进入了发展的快车道, 互联网服务越来越多元化, 各种新兴业务层出

不穷, 人们对服务的需求也与日俱增, 各种服务产生的数据爆发式地增长, 这使得对网络资源调度的灵活性要求越来越高。实现特定流量调度, 诸如 QoS、接入控制、流量工程, 变得越来越力不从心。然而, 当前网络中存在着大量互

收稿日期: 2016-08-17; 修回日期: 2016-11-10

不相干的协议,这些协议或许只在某一域内的网络设备中实现,网络设备之间只是简单地交互信息,很难做到对网络资源的集中监控,再加上传统网络设备底层的封闭性,使得策略部署实施变得困难,在这种情况下美国斯坦福大学的 MacKneow 教授提出了软件定义网络 (software defined networking, SDN)^[1] 的全新架构。

SDN 是一种新型的网络架构模型,它的主要特性是控制与转发分离,将传统网络设备中的控制功能解耦,形成一个集中式的控制平面,该控制平面能够对网络设备进行实时的监控,对网络中的数据流进行细粒度处理,根据网络的状况来动态地实现流量调度。SDN 对网络设备的集中控制,使得网络策略的部署变得便捷。本文的目的是解决当前软件定义网络中流量传输过程中出现的高时延、高分组丢失率以及网络的资源利用率低等问题,提出了一种业务属性相关的多路径传输机制。

2 研究现状

网络中的流量传输是学术界研究的热点,怎样设计出更好的流量传输模型,保障数据流的时延、分组丢失率等参数,提高用户的服务质量,传统网络中有一系列的方法来解决这种问题,如综合服务模型 (integrated service model),使用资源预留协议,对网络中优先级高的数据流通过预留带宽资源,保障数据流的 QoS 属性,这种机制使得每个路由器都要对网络状态进行监控、存储数据流的信息、运行相同的协议,造成了策略实现复杂、可扩展性差等缺点。区分服务模型 (differentiated service model),通过使用数据分组的 ToS 字段来标记流的类型,对等级高的数据流优先处理,这种虽然能够保障优先级高的数据流,但是无法对数据流进行细粒度处理。

在 SDN 中许多研究者针对这一问题提出了一系列的方法,如利用 SDN 架构集中控制,对网络资源实时监控,对数据流进行细粒度处理。Hilmi E E 等人^[2]提出了一种端到端的流量传输模型,该模型将网络中的数据流分为高优先级数据流和背景流,以链路的时延、带宽作为选路的标准,将多约束问题中的 CSP 问题引入 SDN 中,根据链路的时延、带宽动态地选择路径,将优先级高的数据流优先进行路由,保障其 QoS 属性。但是这种传输模型只是保障一种数据流的 QoS 属性,导致其他数据流的 QoS 属性不可控,同时单路径传输出现了链路负载不均衡、网络资源利用率低的问题。Cui H Y^[3]等人提出通过控制器获取每条

链路的时延参数和可用带宽,选取路径时,在所有可达路径中选取可用带宽最大且时延小于最大值的路径。该策略以可用带宽和时延为基准,能够保证数据分组的快速传输,但由于没有考虑不同链路的负载占用程度,没有考虑数据流的业务属性,无法根据不同业务属性数据流的特性来选择路径,导致数据流的时延、分组丢失率在一个较高的水平。Yu T F 等人^[4]提出了一种针对视频流的动态路由机制,该机制也是采用分层的思想,将视频流分为基础层和增强层,该机制优先将基础层的数据分组进行转发,保障最基础的服务质量。YAN J Y 等人^[5]提出了一种自适应路由算法,根据链路的开销、带宽和节点之间的物理距离选择多条路径,在多条路径里面选择链路利用率最低的路由作为当前最优路径,并在一定时间间隔内对网络状态进行更新,当网络发生故障时能够对数据流采取自适应的重路由策略。但是该算法并没有考虑数据流的业务属性,单路径传输使得网络的资源不能充分利用。

Yang Y^[6]等人提出了一种多路径动态路由算法,通过最大流流经节点间的数目、链路的容量来定义链路的关键度,结合链路的剩余带宽将数据流优先分配到链路关键度低的路径。Zhang H L^[7]等人提出将 ECMP 算法应用到数据中心网络中,结合 SDN 对控制平面的监控,根据链路的带宽实现动态路由,提高网络中链路的负载均衡度,但是并没有考虑到数据流的业务属性,选择链路的剩余带宽作为选路标准,不能较精确地反映链路的时延、分组丢失情况。

本文在现有研究的基础上提出了一种基于业务属性相关的多路径传输机制,将网络中的数据流进行分类,对于不同业务类型的数据流结合网络中链路的时延、分组丢失率,选择合适的路径进行路由,降低了数据流的时延、分组丢失率。

3 业务属性相关的多路径传输机制的设计与实现

本文提出的多路径传输机制的思想为:为不同业务属性的数据流,根据其对时延和分组丢失率要求不一样,结合网络链路的实时状况,在蚁群算法的基础上动态地选择适合该数据流的路径,充分地利用了 SDN 对网络资源的实时监控能力,对数据流的细粒度处理能力,降低数据流的时延、分组丢失率,提高网络的资源利用率。

3.1 业务的划分

随着互联网的发展,网络业务种类越来越丰富,3GPP



对网络中的数据流分为 4 类^[7]:会话类、流媒体类、交互类、后台类。

- 会话类业务:这类业务是对实时性要求非常强的业务,如视频会议、VoIP,对时延非常敏感,而对分组丢失率的要求不是那么高。
- 流媒体类业务:这类业务也属于实时性业务,但是由于它是单向传输,不需要进行交互,所以实时性要求没有会话类业务那么严格。
- 交互类业务:这类业务属性非实时性业务,包括网页请求、网络管理数据、控制信令,对时延和分组丢失率都有一些要求。
- 后台类业务:后台类业务包括一些自动的后台 E-mail 接收、SMS 或者接收一些文件和数据库下载。这类业务的特点是用户对传输时间没有特别的要求,但对分组丢失率的要求很高。

对时延要求: 会话类业务 > 流媒体类业务 > 交互类业务 > 后台类业务。

对分组丢失率要求: 后台类业务 > 交互类业务 > 流媒体类业务 > 会话类业务。

在 OpenFlow 协议的头字段中定义了 ToS 域,通过 ToS 域来标记数据流的类别,当 packet_out 消息到达控制器时,控制器解析 ToS 字段就可以知道该数据流的类别。OpenFlow 协议的头字段如图 1 所示。

3.2 网络资源监测模块

本模块主要是用来监测当前网络中链路状态信息,首先定义一个网络拓扑 $G(N, A)$, N 表示网络中节点集合, A 表示网络链路集合,定义 $i, j \in N$, (i, j) 表示节点 i 到节点 j 的链路,为了表示方便,用 l_{ij} 表示链路 (i, j) , d_{ij} 表示链路 (i, j) 的时延, p_{ij} 表示链路 (i, j) 的分组丢失率。OpenFlow^[8] 协议的流表项中定义了 4 种计数器用,统计已处理数据分组的数量等信息。本文使用 OpenFlow 中 per port 计数器来计算链路的分组丢失率,首先由控制器向所测链路两端的交换机发送 ofp_port_status_request 消息,交换机收到消息后,读取所有端口中的计数器中的数据,将读取到的数据

封装在 ofp_port_status 消息中并将该消息反馈给控制器,通过读取上一跳交换机中所有端口的 tx_packets 之和与下一跳交换机中所有端口的 rx_packets,所以链路 (i, j) 的分组丢失率为:

$$p_{ij} = \frac{\sum tx_packets - \sum rx_packets}{\sum tx_packets} \quad (1)$$

在控制器监测链路的分组丢失率之后,为了满足路由选择需求,还需要监测链路的时延,监测方法如下。

(1) 首先,控制器向交换机 i 发送一个 packet_out 报文,该报文携带了一个下发时的时间戳,并且指示将其转发到与链路 l_{ij} 相连的下一跳交换机 j 。

(2) 当交换机 j 收到该数据报文时,因为没有能与之相匹配的流表项,所以将其封装在 packet_in 消息中发送给控制器,当控制器收到消息时与当前的时间相减得到一个时间差 T_1 。

(3) 用 (1)(2) 所述的方法,控制器给 j 发送一个 packet_out 报文,同理控制器会收到从 i 发过来的 packet_in 报文,得到一个时间差 T_2 ,所以 $T_1 + T_2 =$ 控制器与交换机 i 的 RTT(往返时延) + 控制器与交换机 j 的 RTT + 链路 l_{ij} 的 RTT。

(4) 控制器向交换机发送一个带有时间戳的 Echo request,交换机会回复一个 Echo reply,按照这种方法就能够得到控制器到交换机 i 的 RTT 为 T_a ,到交换机 j 的 RTT 为 T_b 。那么链路 l_{ij} 的 RTT $= (T_1 + T_2) - (T_a + T_b)$,设链路的往返时延相等,那么 $d_{ij} = \frac{(T_1 + T_2) - (T_a + T_b)}{2}$ 。

3.3 更新周期

由于 SDN 架构集中控制的特点,网络中所有的交换机要连接到控制器,通过控制器的计算得到相应的策略,以下发流表的方式来控制网络中所有的交换机,在这种情况下,对控制器的计算处理能力是一个很大的挑战,本文对网络资源进行监测,在控制器中周期性地更新网络中的链路信息,并根据计算机中 CPU 的负载情况来设置网络拓扑、链路状态的更新周期,研究表明^[1],普通的计算机中

入端口	源 MAC 地址	目的 MAC 地址	以太网类型	WLAN ID	WLAN 优先级	源 IP 地址	目的 IP 地址	IP	IP ToS 域	TCP/UDP 源端口	TCP/UDP 目的端口
Igresss port	Ether source	Ether des	Ether type	WLAN ID	WLAN priority	IP src	IP des	IP proto	IP ToS bit	TCP/UDP Sreport	TCP/UDP des port

图 1 OpenFlow 协议头字段

的控制器能够每秒处理大约 10 万条消息,当 CPU 的利用率小于 75% 时,控制器能够对到达的数据流进行实时处理,当 CPU 的利用率小于 50% 时,说明控制器没有得到充分的利用。因此本文的链路状态信息根据 CPU 的利用率进行动态设置,具体函数:

$$T_n = \begin{cases} T_0 - (50\% - p) \times 0.5, & p \leq 50\% \\ T_0 - (p - 50\%) \times 0.1, & 50\% < p \leq 75\% \\ T_0 + (p - 50\%) \times 0.5, & p > 75\% \end{cases} \quad (2)$$

其中, T_0 为当前更新周期, 设为 5 s。通过对控制器更新链路状态信息周期的设定, 提高了控制器对数据平面的管理粒度, 在一定程度上起到了充分利用网络资源的目的。

3.4 路由计算模块

(1) 蚁群算法介绍

蚁群算法是意大利学者 Marco D 根据蚂蚁觅食行为提出的一种启发式算法, 蚂蚁在寻找食物的途中会留下信息素, 下一只蚂蚁会根据信息素浓度的大小和启发能力来决定自己的路径, 这样就会形成一种正反馈的机制, 最终蚂蚁会选择一条信息素浓度最高的路径, 那么这条路径就是最优的路径。

(2) 路由选择规则

每只蚂蚁选择下一跳节点的概率计算式如下:

$$p_{ij}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{k \in L} [\tau_{ik}(t)]^\alpha \cdot [\eta_{ik}(t)]^\beta}, & j \in L \\ 0, & \text{其他} \end{cases} \quad (3)$$

其中, $p_{ij}(t)$ 表示蚂蚁选择下一跳节点的概率, L 是节点集 N 的子集, 与节点 i 相邻的节点集合, $\tau_{ij}(t)$ 、 $\eta_{ij}(t)$ 分别表示信息素因子、探索新路径的启发函数, α 表示链路上的信息素对蚂蚁选择路径的影响因子, α 值越大表示链路上的信息素对蚂蚁选择路径的影响越大, β 表示蚂蚁探索新路径的启发函数对选择链路的影响因子, β 值越大表示探索新路径的启发函数对蚂蚁选择路径的影响越大。根据算法的原理蚂蚁最终选择了一条信息素浓度最高的路径。当数据流为会话类业务时令 $\tau_{ij}(t) = \frac{1}{d_{ij}(t)}$, $\eta_{ij}(t) = \frac{1}{p_{ij}(t)}$ 且 $\alpha \gg \beta$, 当数据流为流媒体业务时, $\alpha > \beta$ 。当数据流为交互类业务时, $\tau_{ij}(t) = \frac{1}{p_{ij}(t)}$, $\eta_{ij}(t) = \frac{1}{d_{ij}(t)}$ 且 $\alpha > \beta$, 当数据流为后台类业务时 $\alpha \gg \beta$ 。

(3) 信息素更新规则

当蚂蚁经过一条链路时会留下信息素, 设当 t 时刻时, 该路径上蚂蚁留下的信息素总量为 $\tau_{ij}(t)$, 那么经过 n 时间段之后链路 (i, j) 的信息素总量为:

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (4)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (5)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{蚂蚁经过链路}(i, j) \\ 0, & \text{其他} \end{cases} \quad (6)$$

其中, Q 表示经过一条路径留下的信息素总量, L_k 表示蚂蚁 k 经过路径的总长度, ρ 表示信息素挥发因子, m 表示蚂蚁的数量, k 表示蚂蚁的编号, 其中 $k \leq m$, $\Delta\tau_{ij}(t)$ 表示 n 时间段内有 m 只蚂蚁经过链路 (i, j) , 并留下了 $\Delta\tau_{ij}(t)$ 信息素, 此时链路 (i, j) 上的信息素为 $\tau_{ij}(t+n)$ 。

(4) 算法的流程

输入: 网络的拓扑结构 $G(N, A)$, 源节点 s 、目的节点 d , 节点的邻接矩阵, 节点间链路的时延矩阵、分组丢失率矩阵。算法中的参数 α 、 β 、 ρ , 确定的迭代次数 $countMax$, 每次迭代中蚂蚁的数量 m 。

输出: 源节点到目的节点的最优路径 $path$ 。

初始化: 循环计数器 $count=1$; $\Delta\tau_{ij}(t)=0$; 迭代计数器 $count=0$; $path=s$ 。

For($count=0$; $count \leq countMax$; $count++$)

$tabu_k = \phi / * tabu_k$ 为第 k 只蚂蚁的禁忌表, 每当蚂蚁选择一个节点后将该节点加入禁忌表中, 且每只蚂蚁开始选路时禁忌表设为空集 $*$

For($k=1$; $k \leq m$; $k++$)

$r_k = s$ /* 蚂蚁 k 的当前节点

While($r_k \neq d$)

确定蚂蚁 k 的下一跳节点集合 $allow_k$

If $allow_k = \phi$

跳出本次循环

Else

依据式(3)计算蚂蚁选择下一跳各节点的概率将概率最大的节点 v 作为下一跳节点

$Path = [path, v] / *$ 将节点 v 加入路径中 $*$

$r_k = v$

更新 $tabu_k, allow_k$



End If

End While

End For

根据式(4)~式(6)进行信息素更新。

初始化 $\text{tabu}_k, \text{allow}_k$ 。

选择信息素最高的路径。

如有多条路径的信息素同时最高,则选择瓶颈带宽最大的路径。

输出最优路径。

按照上述算法为每一种业务的数据流计算出一条最优路由,通过控制器下发表给路径上的交换机,当有数据流到达交换机时,通过匹配相应的流表将数据流进行转发。综合分析上述的步骤可以得出算法的时间复杂度为 $O(\text{countMax} \times m)$ 。

4 实验与结果分析

在理论分析之后,将对前面论述的算法进行仿真。仿真主要分为3个步骤:仿真平台的搭建、算法的参数设计、仿真结果分析。

4.1 仿真平台搭建

仿真环境的搭建分为数据平面和控制平面。其中数据平面采用 Mininet 仿真平台,自定义具有 Open vSwitch2.3.1^[9]虚拟交换机的网络拓扑,在熟悉了 Mininet 的基础之上,创建胖树(fat-tree)拓扑结构作为验证本文所提算法的数据平面。

控制平面采用 OpenDaylight 控制器,将本文所论述的算法的实现代码作为插件导入控制器,该插件主要实现两个功能:链路状态检测和路由计算。其中链路状态检测功能为周期性地向数据平面发送带有 LLDP 数据分组的 packet_out 消息,收到该消息之后数据平面的交换机就会向控制器 packet_in 消息,控制器通过解析该消息记录的数据来计算链路的时延和分组丢失率。路由计算的功能为:通过链路状态检测得到链路的时延、分组丢失率,从而计算相应数据分组的路由。

4.2 算法的参数设置

经过多次的仿真比较,在算法中设定每次迭代的蚂蚁个数为 15,信息素挥发因子 $\rho=0.3$,蚂蚁经过一条路径留下的信息素总量 $Q=1$,迭代次数 $\text{Count}=40$,当数据分组为流媒体和交互类业务时 $\alpha=0.5, \beta=0.2$ 。拓扑中每条链路的带宽设置为 100 Mbit/s。

4.3 仿真结果分析

本文比较分析了 ECMP 算法与本文所提出的算法在时延和分组丢失率上的不同,在图 2 中 data2、data1 表示在使用 ECMP 算法时流媒体类业务和交互类业务的时延, data3 表示交互类业务时延, data4 表示流媒体类业务的时延,从图 2 中可以看出,流媒体类的时延明显低于交互类的时延,能够满足业务的需求。在图 3 中 data1、data2 分别表示使用 ECMP 算法时,网络中的流媒体业务数据流、交互类业务数据流的分组丢失率,从图 3 可以看出,该算法不能区分出数据流的种类,不能保障不同业务属性数据流的分组丢失率。data3、data4 分别表示使用本文所提出的算法时,交互类业务数据流、流媒体类业务的数据流,从图 3 可以看出该算法能够为不同业务属性的数据流保障其分组丢失率,而且分组丢失率相对较低。

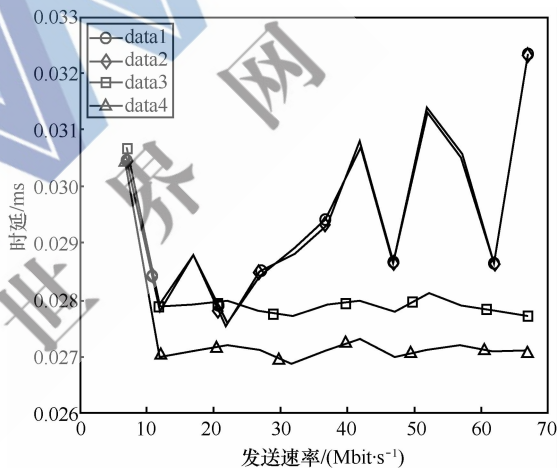


图 2 时延比较

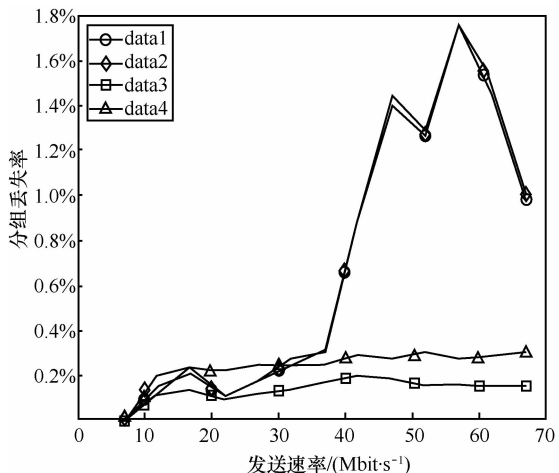


图 3 分组丢失率比较

5 结束语

提出了一种 SDN 架构下业务属性相关的多径路由算法, 该算法利用蚁群算法的正反馈机制, 通过将链路的时延、分组丢失率引入蚁群算法中, 结合网络中不同业务属性的数据流对时延、分组丢失率的要求不一样, 实行通过多次迭代, 得出源/目的节点之间适合不同业务属性的多条路径。下一步将研究根据网络的状态来进行多径路由选择。

参考文献:

- [1] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow: enabling innovation in campus networks[J]. ACM Sigcomm Computer Communication Review, 2008, 38 (2): 69-74.
- [2] HILMI E E, MURAT A T. Distributed QoS architectures for multimedia streaming over software defined networks [J]. IEEE Transactions on Multimedia, 2014, 16(8): 1597-1609.
- [3] CUI H Y, ZHU Y, YAO Y, et al. Design of intelligent capabilities in SDN [C]//IEEE International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems, May 11-15, 2014. New York, USA. New Jersey: IEEE Press, 2014: 1-5.
- [4] YU T F, WANG K C, HSU Y H. Adaptive routing for video streaming with QoS support over SDN networks [C]//2015 International Conference on Information Networking, March 12-15, 2015, New York, USA. New Jersey: IEEE Press, 2015: 318-323.
- [5] YAN J Y, ZHANG H L, SHUAI Q J, et al. HiQoS: an SDN-based multipath QoS solution [J]. Wireless Communication over Zigbee for Automotive Inclination Measurement China Communications, 2015, 12(5): 123-133.
- [6] YANG Y, YANG J H, QIN D H. Multipath routing algorithm for data center networks [J]. Journal of Tsinghua University(Science and Technology), 2016, 56(3): 262-268.
- [7] ZHANG H, GUO X, YAN J, et al. SDN-based ECMP algorithm for data center networks [C]//IEEE Computing, Communications and it Applications Conference, October 22-25, 2014, London, UK. New Jersey: IEEE Press, 2014.
- [8] Quality of service (QoS) concept and architecture [EB/OL]. [2016-08-01]. <http://www.3gpp.org/ftp/Specs/html-info/23107.htm>.
- [9] Openflow switch specifications version 1.3.2 implemented[EB/OL]. [2016-08-01]. <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.pdf>, 2013.

[作者简介]



余翔(1969-),男,重庆邮电大学通信与信息工程学院副教授,主要研究方向为通信网及交换技术、计算机网络及信息安全、下一代网络技术。



易明敏(1992-),男,重庆邮电大学通信与信息工程学院硕士生,主要研究方向为软件定义网络。



杨路(1969-),女,重庆邮电大学通信与信息工程学院高级工程师,主要研究方向为通信网及交换技术、计算机网络及信息安全、下一代网络技术。