SDN 网络环境下的 MPTCP 的移动切换机制

孙茂鑫,钱红燕
(南京航空航天大学 计算机科学与技术学院,江苏 南京 211106)

摘 要:无线网络中,移动设备在不同的接入点(AP)之间切换是不可避免的问题。现有的移动切换机制不能真正地实现无缝切换,并且切换时信令开销大,切换时延长。实现移动设备的快速切换以及在切换中保证 QoS,已经成为无线网络的研究热点。为提高无线切换时的网络性能,文中采用软件定义网络(SDN)架构,提出一种基于多路径传输控制协议(MPTCP)的移动切换机制。目的是使移动设备在不同 AP 间实现无缝切换以及保证在切换过程中网络保持较高的吞吐量。并且为了提高网络的利用率和性能,文中利用信号强度值、丢包率和网络时延共同决定切换网络,可以充分利用各 AP 的网络带宽。实验结果表明,该机制可以有效地实现移动设备的无缝切换,保证了服务质量,并且避免了乒乓效应,提高了网络性能。

关键词:软件定义网络;移动切换;多路径传输控制协议;多指标切换;无缝切换
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2016)06-0011-05
doi:10.3969/j.issn.1673-629X.2016.06.003

Mobile Handover Mechanism Based on MPTCP in SDN Environment

SUN Mao-xin, QIAN Hong-yan
(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

Abstract: Handover of mobile equipment between different access points is an inevitable problem in the wireless network. There are many problems in the existing mobile handvoer mechanism, which cannot ensure the seamless handover, and have serious handover signaling overhead with long handover delay. How to satisfy the user's quality of service and achieve seamless handover becomes a very worth of study in the field of wireless communication network. To improve the performance of the wireless network when roaming, in this paper, a mechanism of handover is proposed based on multipath TCP and software defined networking. It aims to achieve the seamless handover between different AP for mobile equipment and keeps high throughput of network in the handover process. In order to improve utilization and performance, the RSS, loss rate and delay are adopted to make handover decision, whcih can make full use of network bandwidth. Finally, the experiment shows that the proposed handover mechanism can achieve seamless handover and ensure QoS, avoiding the "ping-pong" effect and improving the network performance.

Key words: SDN; mobile handover; MPTCP; multiple parameters handover; seamless handover

1 概 述

随着无线网络技术的发展,特别是无线接入技术的发展,3G、4G 等技术的出现,使得移动设备越来越普及。无线接入已成为人们访问互联网的主要接入方式。在无线网络中,用户的移动性管理是一个很重要的挑战。尤其是移动切换作为移动性管理的关键技术,成为了无线网络中的研究热点。

文中主要研究的是在 SDN 网络下,利用 MPTCP 机制解决移动设备在现有网络架构下信令开销大、三角路由、不支持无缝切换等问题。移动切换主要分为两大类:一类是切换发生在同一种网络技术之间,称为水平切换。例如在 3G 网络内不同 AP 间的切换。另一类是发生在不同网络技术之间的切换,称为垂直切换。例如从 3G 网络中的一个 AP 切换到 4G 网络中的一个 AP。文中主要研究水平切换。

在无线切换领域,学者们已经提出了许多方案来

收稿日期:2015-08-08 **修回日期**:2015-11-17 **网络出版时间**:2016-05-05
基金项目:江苏省科技计划项目(BY2013095-2-10)
作者简介:孙茂鑫(1990-),男,硕士研究生,研究方向为无线网络、SDN 网络;钱红燕,副教授,研究方向为计算机网络、无线通信、信息安全。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20160505.0828.070.html

解决设备在移动中出现的问题。其中,在 TCP/IP 架构下影响最大的是移动 IP 技术[1-2]。移动 IP 定义了三种实现移动功能的实体:

(1)移动主机。当移动设备从一个网络切换到另一个网络时,要保持通信不中断,在切换时保持主机的 IP 地址不变。

(2)家乡代理。有个端口要与设备的家乡链路连接的路由器,在设备进行切换时,设备会通知家乡代理该主机当前的位置。

(3)外地代理。它是移动设备在外地链路上的路由器。

该技术通过本地代理和外地代理使得设备在移动过程中不需要改变 IP 地址,使通信保持连续。但移动 IP 技术存在的最大问题是三角路由问题,增加了网络的时延,导致切换时延过长。同时移动设备作为网络的接收者时,由于移动设备不断改变位置,导致向它提供实时业务变得十分困难,造成了 QoS 没有保障。

在 SDN 网络架构下,也有很多解决方案被提出。文献[3]提出利用 OpenFlow 协议提供路由,可避免移动 IP 技术的三角路由问题。基于异构无线网络垂直切换算法[4]主要解决垂直切换中十分重要的问题:AP 的选择和切换时间的选择。该方案能确保设备在最合适的时间进行切换,并且保证最优的 QoS;但该方案没有真正实现无缝切换,因为 AP 的选择需要花费一定的时间,导致了过长的网络时延。

对 MPTCP 的研究,近年来也取得了一些进展。在文献[5]中,移动设备利用 MPTCP 协议连接多个不同 AP,可以使切换过程保持连接不中断,同时通过大量实验发现,当设备连接的 AP 过多时,会严重降低网络的吞吐量。WiPoMu[6]、FatVAP[7]和 Virtual WiFi[8]都是利用虚拟接口技术使设备连接多个 AP,使设备在不同 AP 间可以快速进行平滑切换,也可以在不同 AP 的不同信道间进行切换,同时提高了网络的聚合带宽。这些方案由于需要修改客户端,很难得到普及,同时这些方案目前都只是在理论上进行仿真验证,很难在实际环境中对网络性能进行验证。Mobility[9]架构也是根据 MPTCP 提出的。该架构通过部署 MPTCP 定量增加代理,同时可以兼容 TCP 协议。该方案能很好地完成平滑切换,同时可以减少能力消耗。

2 相关技术概述

SDN(Software Defined Networking,软件定义网络)[10]是由美国斯坦福大学提出的一种网络架构。SDN 概念和 OpenFlow 的提出,引起了学术界和产业界的广泛关注。SDN 网络将控制平面从嵌入式设备中独立出来,形成控制器。该控制器相当于整个网络的大脑,控制整个网络的行为。而转发设备只负责数据的转发,它相当于网络的四肢,受控制器的控制。控制器通过向转发设备下发流表的方式来控制转发设备。SDN 主要思想是管控分离,SDN 控制器提供了用户编程的接口,用户通过在控制器中编写业务应用程序,控制网络设备按照用户的意愿运行。SDN 网络可以针对不同的业务定制所需要路由、QoS 流量工程、策略、安全等实时下发到网络中,可以很好地实现网络对业务和应用的配置。目前,谷歌已经在全球的数据中心部署 SDN[11],谷歌用 10 Gbit/s 的网络链路链接了全球 12 个数据中心,通过在 SDN 网络中部署 QoS 流量工程和优先级调度,使数据中心间链路的使用率从平均的 30% ~40% 提高到接近 100%。谷歌的成功案例证明了 SDN 网络架构的优越性。

为了提高网络链路的带宽和利用率,同时增强链路的鲁棒性,IETF 提出了 MPTCP 的概念。它是 TCP 协议的扩展。设计 MPTCP 的初衷是在端到端的通信中使用多个链路,将数据流分发到多条传输路径上,通过这种方式来提高网络带宽,并且能平衡拥塞。MPTCP 能避免在拥塞的路径上继续传输数据,这个目标保证了吞吐量和公平性。利用 MPTCP 协议,移动设备可以同时连接多个 AP,根据文中提出的切换机制来保证移动设备在切换时实现无缝切换的目标。MPTCP 的连接过程与 TCP 连接类似,采用“三次握手”,不同的是 MPTCP 在建立连接过程中增加一个 OPT_MPC 选项,用来表示 MPTCP 连接。MPTCP 与传统的 TCP 的不同之处是,在应用层和传输层之间加入了支持多路径传输的 MPTCP 层,如图 1 所示,原 TCP 层只是对子流起作用,这样从端到端的通信双方来看,传输层仍是单路通信。MPTCP 的主要功能是:将数据进行分流,使子流在不同的链路上传输。所有子流到达目的端后,MPTCP 再根据序列号和确认号对子流进行重组发给应用层。

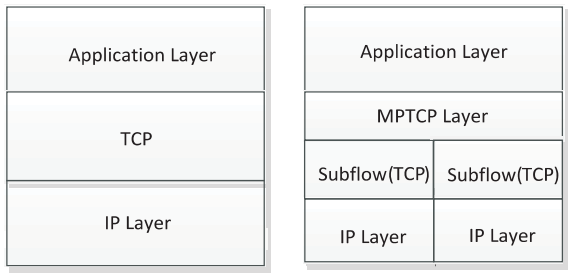

图 1 TCP 与 MPTCP 对比

3 移动切换机制

3.1 系统架构

文中的移动设备和服务器都支持 MPTCP 协议。文献[5]已证明,当设备使用多个接口进行通信时会

降低吞吐量。文中研究的主要目的是如何利用 MPTCP 实现无缝切换,以及在切换时保证链路的吞吐量。所以在文中,移动设备只使用两个接口进行通信。

图 2 是系统架构示意图。控制器控制网络中所有的 OpenFlow 交换机,AP 与 OpenFlow 交换机相连,移动设备通过 AP 接入网络。文中移动设备中的两个接口分别连接一个 AP,并且通信双方都必须支持 MPTCP 协议。

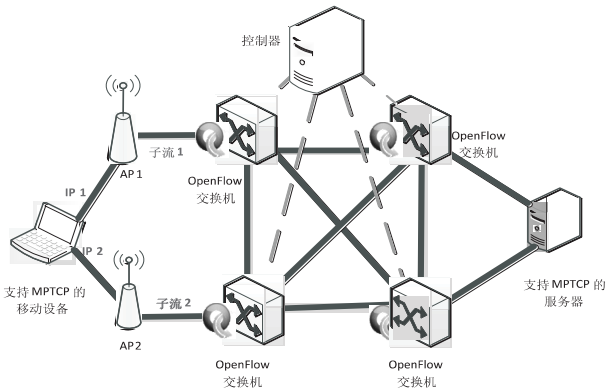

图 2　系统架构图

3.2　切换流程

一个切换过程是由切换触发、切换决策以及切换执行三个过程组成。系统切换流程图如图 3 所示。

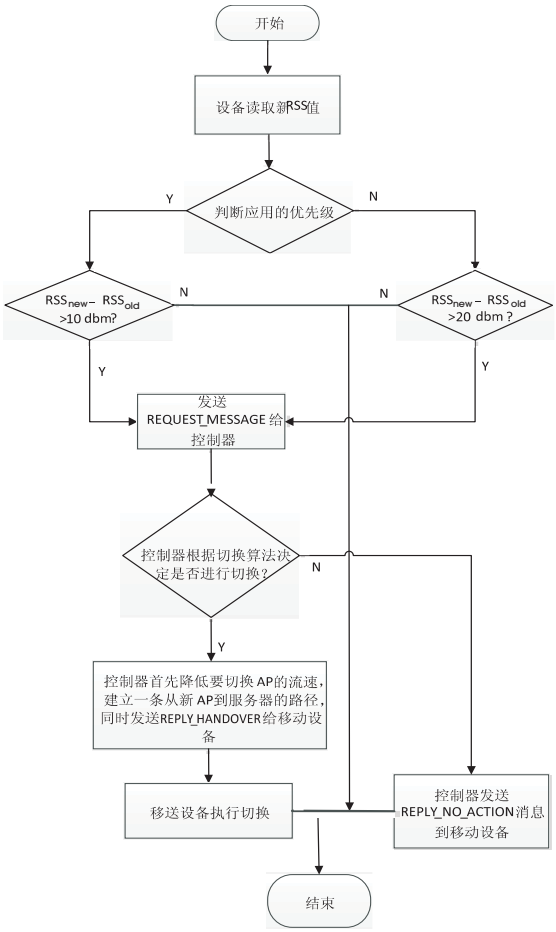

图 3　系统流程图

由移动设备来触发切换过程,当设备监测到周围 AP 的 RSS 值比当前 AP 的 RSS 值大时,首先设备需判断应用程序的优先级,由于实时性应用程序对网络要求较高,当新旧 RSS 值的差相差大于 10 dBm,移动设备发送切换请求信息(REQUEST_MESSAGE)到控制器。该信息包含网络接口的相关信息、IP 地址、MAC 地址以及当前 AP 的信息等。若为其他类型应用,则需要新旧 RSS 值相差 20 dBm 时发送切换请求信息到控制器。由于只使用 RSS 值进行切换决策有局限性,会产生乒乓切换[12],产生负面效应,它只考虑了当前网络的链路状况和目标网络的链路状况,并没有从全局对网络状况进行判定。所以文中所提出的切换机制,在控制器中还需要根据链路的丢包率以及网络时延共同来决策是否进行切换。当设备监测到周围有多个目标 AP 的 RSS 值大于当前 AP 的 RSS 值时,发送到控制器的信息中包含所有 AP 的相关信息。由网络控制器决定选择哪个 AP,以及是否进行切换。由于控制器知道全网拓扑以及网络链路状况,控制器可以基于全局的网络信息来进行决策,可以有效保证网络整体性能,并且可以满足应用程序对网络的要求,能尽可能地利用网络的带宽,充分利用网络资源。

控制器根据多指标切换算法决定移动设备的行为,该算法将在 3.3 节介绍。若控制器做出切换的决定,控制器首先通过设置 OpenFlow 交换机的队列,不同速率的流量发往不同的队列来实现[13],使发往原 AP 的数据进入速率最低的队列,可以降低数据的发送速率,减少丢包率。通过多指标决策算法求出一个新 AP,然后控制器发送切换消息(REPLY_HANDOVER)到移动设备。该消息中包含移动设备要切换到新 AP 的信息,同时控制器向 OpenFlow 交换机下发流表,建立从目标 AP 到服务器的路径。由于此时移动设备的另一个网络接口继续保持与服务器的通信,使得一个接口在进行切换时,应用程序保持连接不中断,可以实现平滑切换。若控制器决策是维持现状,控制器只需要发送无动作消息(REPLY_NO_ACITON)到移动设备。

移动设备收到控制器发来的消息,执行相应的动作。切换过程结束。

3.3　多指标切换算法

多指标切换算法在利用 RSS 值的同时,又增加了其他指标共同完成决策过程。该算法利用多个指标进行切换决策,能有效避免乒乓切换,降低切换次数。

在切换算法中使用了三个评判指标:RSS、网络时延和丢包率。利用 RSS 值触发切换,根据网络时延和丢包率进行切换决策,求出一个目标代价最小[14]的目标 AP 进行切换。RSS 值反映了当前信道的状况,RSS

值越大说明使用该 AP 传输数据时信号损耗越小。网络时延和丢包率反映 AP 所在链路网络的性能,网络时延和丢包率越小,说明链路的性能越好。文中将应用程序分为两大类:一类是实时性应用程序,一类是非实时性应用程序。每一类应用程序的评判指标在决策时有不同的权值,如表 1 所示。

表 1 指标权重

类别	丢包率	时延
实时性应用	0.3	0.7
非实时性应用	0.5	0.5

文中定义 AP 的代价如式(1)所示。

$$c_{(AP)} = \alpha \cdot \text{loss}_{(AP)} + \beta \cdot d_{(AP)} \tag{1}$$

式(2)为 AP 丢包率,表示的含义是从服务器到 AP 所在路径的丢包率。

$$\text{loss}_{(AP)} = 1 - \prod_{l \in p} (1 - \text{loss}_{(l)}) \tag{2}$$

式(3)表示 AP 网络时延,表示的含义是从服务器到 AP 路径上所有链路的时延和。

$$d_{(AP)} = \sum_{l \in p} d_{(l)} \tag{3}$$

其中:p 为从服务器到 AP 的路径;l 为 AP 所在路径的一条链路;$\text{loss}_{(AP)}$ 为 AP 丢包率;$\text{loss}_{(l)}$ 为链路的丢包率;$d_{(AP)}$ 为 AP 的时延;α,β 为表 1 中各个指标的权重值。

该算法利用了一个阈值 Thr,先从所有 AP 中计算代价最小的 AP 记为 $C_{\min}$,再计算原 AP 的代价 C_{old} 与 $C_{\min}$ 的差,再与阈值 Thr 进行比较来决定相应动作。

具体算法如下所示:

```
do
计算相关 AP 的代价,并将其放入集合 S 中;
while 数据都存入集合 S 中
求出 S 中元素的最小值 C_min ;
if C_min ≠ ∅ then
if C_old - C_min <0 then
发送 REPLY_NO_ACTION 到移动设备;
else
if C_old - C_min <=Thr then
发送 REPLY_NO_ACTION 到移动设备;
else then
控制器向就 AP 所在链路的交换机发送流表,降低数据的发送速率,同时发送 REPLY_HANDOVER 消息到移动设备;
end //算法结束
```

4 实验结果及分析

文中搭建了一个 SDN 网络环境测试床,移动设备装有一个内置的无线网卡和一个 Wi-Fi 无线网卡,并且移动设备和服务器安装带有 MPTCP 的 32 位 Ubuntu12.04 操作系统。实验在一个有 13 个 AP 的楼道里进行,并且 AP 间的间距分布不均匀。SDN 控制器选择了 POX,该控制器是模块化编程,方便在控制器中添加自己所需模块,文中在 POX 控制器中添加了多指标切换算法模块。设定的阈值为 Thr=0.05,并且在测试时,使用一些计算机连接在 SDN 网络中互相通信,产生一定的背景流量。

文中比较了在 TCP 协议下以及在 MPTCP 协议下切换时的吞吐量,以及比较了基于 RSS 值切换算法与文中提出算法的切换次数。

第一个实验选取了在一个切换中 30 s 的数据。从图 4 可以看出,使用传统的单 TCP 协议在进行切换时,在切换时会出现长达 5 s 多的连接中断,这对于实时性要求高的应用程序是不能容忍的。而使用 MPTCP 进行切换时,虽然吞吐量有所下降,但始终保持了通信的连续性。这是由于基于 MPTCP 协议的两个接口,在进行切换时一条链路进行切换而另一条链路保持通信,这也是 MPTCP 协议的优势所在。使用该方案能保证高实时性应用程序对网络的要求。

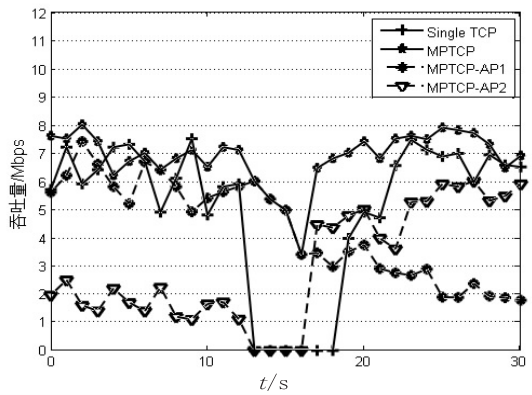

图 4 TCP 与 MPTCP 切换时吞吐量对比

图 5 比较了基于 RSS 阈值切换算法与文中多指标切换算法的切换次数。统计了在 210 s 的时间内,两个算法各进行的切换次数。

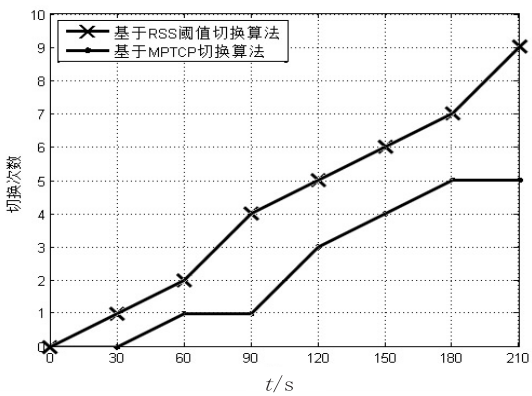

图 5 基于 RSS 阈值的切换算法与多指标切换算法的切换次数对比

从图 5 中可以看出,基于 RSS 阈值的切换算法切换次数较多,这是因为它只考虑了当前的信道状况,而没有考虑到当前网络的状况以及目标网络的状况。而文中的多指标切换算法全面考虑了网络的性能,只有 RSS 值和链路的状况都满足了要求时,才进行切换,从而能显著降低切换次数,减少乒乓效应。

5　结束语

文中提出了在 SDN 网络架构以及 MPTCP 协议下的切换算法,解决了现有切换算法的连接中断、效率低下等问题。利用文中提出的机制,实现了移动主机的无缝切换,并且保证了用户在切换时的服务质量,能有效提高切换时的网络吞吐量。

由于文中的地址是固定 IP 地址,实验环境只是在范围很小的局域网,并且只是在一个 OpenFlow 网络内,没有在更大范围的网络进行实验。未来的工作就是将该机制应用到更大范围的网络中,同时实现在多个 OpenFlow 网络内进行跨网切换。

参考文献:

[1] Perkins C E. Mobile networking through mobile IP[J]. IEEE Internet Computing,1998,2(1):58-69.

[2] Pentikousis K,Wang Y,Hu W. Mobileflow:toward software-defined mobile networks[J]. IEEE Communications Magazine,2013,51(7):44-53.

[3] Pupatwibul P,Banjar A,Sabbagh A A, et al. Developing an application based on OpenFlow to enhance mobile IP networks [C]//Proc of 38th conference on local computer networks workshops. Sydnew,NSW:IEEE,2013:936-940.

[4] Qiang L,Li J,Huang C. A software-defined network based vertical handoff scheme for heterogeneous wireless networks [C]//Proc of global communications conference. [s. l.]: IEEE,2014:4671-4676.

[5] Croitoru A,Niculescu D,Raiciu C. Towards Wifi mobility without fast handover[C]//Proc of USENIX NSDI. Oakland, CA,USA:USENIX,2015:219-234.

[6] Nguyen K,Ji Y,Yamada S. A cross-layer approach for improving WiFi performance[C]//Proc of wireless communications and mobile computing conference. [s. l.]:IEEE,2014:458-463.

[7] Kandula S,Lin K C,Badirkhanli T,et al. Fatvap:aggregating ap backhaul capacity to maximize throughput[C]//Proc of the 5th USENIX NSDI. [s. l.]:USENIX,2008.

[8] Chandra R. Multinet:connecting to multiple IEEE 802.11 networks using a single wireless card[C]//Proc of IEEE INFOCOM. [s. l.]:IEEE,2004.

[9] Raiciu C,Niculescu D,Bagnulo M,et al. Opportunistic mobility with multipath TCP[C]//Proceedings of the sixth international workshop on MobiArch. [s. l.]:ACM,2011:7-12.

[10] 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.

[11] Jain S,Kumar A,Mandal S,et al. B4:experience with a globally-deployed software defined WAN[C]//Proceedings of the ACM SIGCOMM. [s. l.]:ACM,2013:3-14.

[12] 贾宗璞,王红梅,刘淑芬. 乒乓切换在移动 IPv6 扩展协议中的性能分析[J]. 电子学报,2009,37(3):592-597.

[13] Civanlar S,Parlakisik M,Gorkemli B,et al. A QoS-enabled openflow environment for scalable video streaming[C]//Proc of GLOBECOM workshops. [s. l.]:IEEE,2010:351-356.

[14] Juttner A, Szviatovski B, Mecs I, et al. Lagrange relaxation based method for the QoS routing problem[C]//Proc of IEEE INFOCOM. [s. l.]:IEEE,2001:859-868.