

面向多业务的内容中心网络拥塞控制策略

张成晨,王 雷,吕 威,夏 磊

(中国科学技术大学自动化系,合肥 230027)

摘 要: 现有内容中心网络中的拥塞控制策略在解决网络拥塞时,难以直接获得网络中的拥塞状态信息反馈,在多业务的情况下无法保证各业务的服务质量。为此,提出一种能够在多业务情景下提高服务质量的拥塞控制策略。利用中间路由节点实时监控到达的兴趣包,根据兴趣包队列中的业务情况判定拥塞程度,将链路状态信息反馈给发送端,据此调节发送速率。实验结果表明,相比现有基于发送端的拥塞控制策略,该策略能在不增加业务完成时间的前提下有效解决网络拥塞问题,并保障不同业务的服务质量。

关键词: 内容中心网络;拥塞控制;服务质量;标签;未来网络

中文引用格式:张成晨,王 雷,吕 威,等. 面向多业务的内容中心网络拥塞控制策略[J]. 计算机工程,2016,42(4):79-82,87.

英文引用格式:Zhang Chengchen, Wang Lei, Lü Wei, et al. Content Centric Network Congestion Control Strategy for Multi-service[J]. Computer Engineering,2016,42(4):79-82,87.

Content Centric Network Congestion Control Strategy for Multi-service

ZHANG Chengchen, WANG Lei, LÜ Wei, XIA Lei

(Department of Automation, University of Science and Technology of China, Hefei 230027, China)

[Abstract] It is difficult for the existing congestion control strategies in Content Centric Network (CCN) to obtain the feedback of the network congestion status directly. Besides, it can not guarantee the Quality of Services (QoS) for multi-service of each business. To solve this problem, a congestion control strategy that can improve the QoS in multi-service situations is proposed. This strategy monitors the arrived interest packets by the intermediate routing nodes, judges the congesting degree according to the business conditions of the interest packet queue, and feeds back the link state information to receivers, so as to adjust the sending rate. Experimental results show that this strategy can effectively control congestion in the network without increasing the completion time of services, and ensure QoS of different business.

[Key words] Content Centric Network (CCN); congestion control; Quality of Service (QoS); tag; future network

DOI: 10.3969/j.issn.1000-3428.2016.04.015

1 概述

随着互联网基础设施的逐步完善,以及终端设备和应用软件的不断发展,引发了用户对影音媒体等内容需求的爆发式增长。这种需求决定了网络通信模式必须以高效的内容发布、发现和获取为导向,而传统互联网的端到端的通信模式难以满足这种需求,亟待研究新型的未来网络架构。近年来,信息中心网络 (Information Centric Network, ICN) 等新型网络架构的研究发展迅速^[1]。在众多的信息中心网络的体系结构中 (包括内容中心网络 (Content Centric Network, CCN)^[2]、面向数据的网络架构 (Data-

oriented Network Architecture, DONA)^[3]、PSIRP/PURSUIT, 4WARD, SAIL 等), CCN 在内容分发效率,网络安全和部署方面具有显著的优势。CCN 虽然具有很多优点,但是最初提出的架构还有很多方面有待改进,如缓存机制^[4]、路由转发机制^[5]、移动性、拥塞控制等。

CCN 设计了简单的流控制机制,即一个请求分组对应一个数据包,这种机制不能避免网络拥塞的发生,为此,很多人提出相应的解决方法。兴趣包控制协议 (Interest Control Protocol, ICP)^[6] 和信息中心网络传输协议 (ICN specific Transport Protocol, ICTP)^[7] 基于接收端并采用传输控制协议 (Transport

基金项目:中国科学院战略性先导科技专项基金资助项目(XDA06011202);中央高校基本科研业务费专项基金资助项目(WK2100100022)。

作者简介:张成晨(1990-),男,硕士研究生,主研方向为未来网络;王 雷,副教授;吕 威、夏 磊,硕士研究生。

收稿日期:2015-02-06 **修回日期:**2015-03-26 **E-mail:** zhangcc@mail.ustc.edu.cn

Control Protocol, TCP)的拥塞控制策略,前者利用 AIMD 的方式进行窗口控制,后者将控制过程分为慢启动、拥塞避免、快速重传、快速恢复 4 个阶段,并且将内容进行 2 层分解进行控制。ICP 和 ICTP 采用的定时器是基于前一个 Data 包的 RTT 采样,而不去考虑 Data 包具体来自于何处。文献[8-9]提出的方法是网络中的各节点按照当前获取的信息,包括队列长度、出口宽度、往返时延等对将来的拥塞情况进行动态预测,根据预测结果调整此时接收端发送 Interest 包的速率,从而调整了 Data 包的传送速度,达到改善拥塞的目的,然而,其需要维护大量的状态信息,影响了网络的可扩展性。内容中心网络传输控制协议(CCN Transport Control Protocol, CCTCP)^[10]使用一种创新的预测 Interest 包机制来解决多源情况下 RTT 的不可预测性问题,通过在请求某一个数据块前预测其位置并精确地估计重传间隔,但是这种多计时器的机制会给系统带来很大开销。显式控制算法(Explicit Control Algorithm, ECA)^[11]是一种基于显式反馈的拥塞控制机制,通过主动检测网络中间节点上的传输队列长度来判断网络拥塞水平,并显式的反馈给接收端,接收端据此调整 Interest 包的发送速率,从而达到控制网络拥塞的目的。

实际需求中多种业务同时存在的情况比较常见,但是没有考虑到内容的业务属性,这会影响到服务质量。对于多业务的情况,已有拥塞控制方法会统一控制包的传送速度,而这在很多情况下是不妥的。例如,对于时延要求非常高的会话类业务,如果不加区别地统一控制 Interest 包发送速度,势必会导致其时延变长,从而导致用户体验非常糟糕。本文提出面向多业务的内容中心网络拥塞控制策略。当多种对时延要求不同的业务同时存在时,首先降低对时延要求较低业务的 Interest 包发送速率,从而尽量保护对于时延要求较高的业务。

2 模型设计

本文通过监控经过中间路由器的 Interest 数据包,针对不同业务向接收端反馈。在检测到发生拥塞后,首先降低那些对时延要求较低的业务请求发送速率。另外,为了提供不同的用户体验,在添加业务属性的同时添加优先级标签。

由于 CCN 中是以块来请求数据的,一个请求分组对应一个 Data 包,往往是将一个内容对象分成很多块传输,请求时所用的名字可能不仅包含内容名字,而且还包含一个块标识符,因此在 CCN 中没有 IP 网络中的端到端的流的概念,但是在一定时间内用户请求某个内容对象的不同块时经过同一个路由

器的可能性很大。本文提出的面向多业务的拥塞控制策略基于上述假设。

2.1 拥塞检测

由于 CCN 没有统一的数据包格式要求,这给研究者提供了便利。在本文的设计中,在每一个 Interest 包中,添加上请求内容类型和优先级 2 个字段,请求内容类型用来区分不同业务。根据 3GPP 对网络业务的划分^[12],本文把业务分成 4 类:会话类(conversation class),流媒体类(streaming class),交互类(interactive class),背景类(background class)。会话类包括音视频聊天、视频会议等;流媒体类包括音视频在线播放、体育直播等;交互类包括网页浏览、控制信令的传输等;背景类包括文件传输、电子邮件等。从会话类到背景类,对时延的要求逐渐降低,但对丢包率的要求逐渐提高。优先级用来区别服务等级,同种业务可以包括很多优先级,在进行拥塞控制时,先调节优先级低的业务的服务 Interest 包发送速率,保证优先级高的业务的服务质量。

对于每一条出口链路,每隔一段时间进行检测未发出的 Interest 包队列,一旦发现拥塞,及时反馈给相应的接收端,接收端调整相应的 Interest 包发送速率,从而达到拥塞控制的目的。拥塞模型如图 1 所示。中间路由器收到来自多个请求者的 Interest 包,这些 Interest 包都将发往同一条链路。

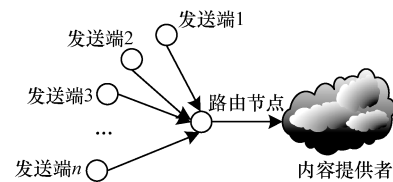


图 1 拥塞模型

拥塞检测的机制是根据当前 Interest 包队列中不同类型的 Interest 包的数量来计算出一个拥塞程度值 V ,当其值达到拥塞阈值时,则认为此时该链路可能已经发生拥塞,需要进行拥塞处理。在进行拥塞程度值的计算时,不同种类的业务会有不同的权值,这主要是根据业务对丢包率的要求来进行赋值的,对丢包率的要求越高,权值会越高,即背景类业务会有最高的权值。下面为拥塞程度值得计算公式:

$$V = w_c \times n_c + w_s \times n_s + w_i \times n_i + w_b \times n_b \quad (1)$$

其中, $n_c, n_s, n_i, n_b, w_c, w_s, w_i, w_b$ 分别是会话类、流媒体类、交互类、背景类的业务数量以及相应的权值,满足:

$$0 < w_c < w_s < w_i < w_b \quad (2)$$

参照文献[11]的做法,将拥塞程度分为 3 类,即

设置 2 个阈值 V_{busy} , $V_{\text{congestion}}$ 。当拥塞程度值小于 V_{busy} 时,链路为空闲状态。当拥塞程度值介于两者之间时,链路为繁忙状态;当拥塞程度值大于 $V_{\text{congestion}}$ 时,链路为拥塞状态。

2.2 拥塞处理

当发现拥塞后,先处理那些对时延要求最低的业务,这样能尽可能地保证多数请求者的服务质量。当同种业务有多个时,先处理优先级较低的业务。如果处理一个业务后,仍然是拥塞状态,则继续使用同样的原则进行处理,直到拥塞程度值低于拥塞状态阈值 $V_{\text{congestion}}$ 。随后每隔一定的时间检测一下拥塞程度,如果链路是空闲状态或繁忙状态,则优先处理对时延要求最高的业务,如果同种业务有多个时,则先处理优先级较高的业务。下面给出拥塞处理的一些算法描述。

算法 1 Select-Interest 算法

输入 Interest 包队列 Interest_list

输出 需要处理的 Interest 包队列 handling_list

1. while link is congestion
2. if there exists Background stream in the pending_list
3. add Interest whose priority is lowest into handling_list
4. feedback information of congestion link to requester
5. delete this Interest from pending_list
6. recalculate V
7. if link is congestion
8. continue
9. else
10. break
11. replay step 2 to step10 in the order of Interactive, Streaming, Conversational

算法 1 从 pending_list 中选出一些 Interest 进行后续处理。Interest_list 是准备发往某个链路的所有 Interest 队列;handling_list 是被选出即将要处理的 Interest 队列,在发生拥塞前是空的;pending_list 是没有被选的 Interest 包队列,在发生拥塞之前 pending_list 与 Interest_list 相同。算法 1 首先检查 pending_list 中是否有背景类业务,如果有则将其中优先级最低的业务添加进 handling_list 中,将链路拥塞的信息反馈给接收者,并且将其从 pending_list 中删除,如果此时链路处于拥塞状态,则继续检查 pending_list,否则此过程结束。如果没有背景类业务,或者背景类业务都已经加入 handling_list 中,则按交互类、流媒体类、会话类业务的顺序检查 pending_list。

算法 2 Response-State 算法

输入 Interest 包队列 Interest_list

输出 请求者的反馈

1. calculate V
2. if $V > V_{\text{congestion}}$
3. goto Select-Interest

4. else if $V < V_{\text{busy}}$
5. for each Interest in handling_list
6. feedback information of free link to requester
7. else
8. for each Interest in handling_list
9. feedback information of busy link to requester

算法 2 根据不同的情况将链路信息反馈给接收者,为了保证公平性,在链路不拥塞的时候,将此信息反馈给 handling_list 中相应的接收者。这种链路状态反馈机制会每隔一段时间执行一次,一旦发现拥塞,则交给 Select-Interest 算法处理。上述 2 个算法都是在路由节点中执行的。当接收者收到相应的反馈信息后将 Interest 包的发送速度进行调整。

算法 3 Regulate-Rate 算法

输入 中间路由器的状态反馈

输出 Interest 包发送速率的调节计划

1. if link is congestion
2. $\text{cwnd} = \text{cwnd} * \alpha$
3. if link is busy
4. $\text{cwnd} = \text{cwnd} + \beta$
5. if link is free
6. $\text{cwnd} = \text{cwnd} * \gamma$

对于速率调节,采用了 ECA^[11] 中的方式。当链路处于拥塞状态时,发送窗口乘以一个小于 1 的系数 α 。当链路处于繁忙状态时,发送窗口加上一个增量 β 。当链路处于空闲状态时,发送窗口乘以一个大于 1 的系数 γ 。

3 实验结果与分析

为了证明上述拥塞控制策略的有效性,本文在 ndnSIM 中模拟了 CCN 的真实环境,并做了一系列仿真实验。ndnSIM 是基于 NS-3 的开源仿真平台,它实现了内容中心网络的通信模型,是内容中心网络的常用仿真软件。由于 ndnSIM 可以实现多业务的拥塞情况,所以本文采用了 ndnSIM 作为仿真工具。实验使用一台 PC 机(2.30 GHz 的 CPU, 4.0 GB 的内存),操作系统为 Ubuntu14.04。实验过程中使用不同种类的业务来分别测试全部业务完成时间以及处理拥塞过程中影响到服务质量的业务数,实验结果与 ECA 和 AIMD 做比较,前者是显式控制算法,后者是已有的传输协议和拥塞控制算法中常用的策略。为了模拟拥塞的情形,采用了图 1 中的拓扑形式,多种请求要去往同一链路。

3.1 业务完成时间

实验中模拟了 5 种情景的拥塞情况,如表 1 所示。表格中列出的每个业务有 4 个参数:(1)业务种类,1~4 分别表示会话类、流媒体类、交互类、背景类;(2)优先级,数字越大表示优先级越高;(3)该请

求的数据量大小;(4)初始时设置的 Interest 发送速率会得到的 Data 包所占用带宽。如情景 1 中的 $S_1(3,1,150,15)$ 表示这是一个交互类业务,优先级为 1,请求的数据大小为 150 MB,初始时 Data 包占用带宽为 15 Mb/s。实验中出口链路带宽设置为 100 Mb/s。

表 1 实验所用情景

情景	情景中的业务
情景 1	$S_1(3,1,150,15), S_2(3,2,300,15), S_3(3,3,225,15), S_4(4,1,200,20), S_5(4,2,300,20), S_6(4,3,200,20)$
情景 2	$S_1(2,1,100,10), S_2(2,2,150,10), S_3(3,2,300,15), S_4(3,1,225,15), S_5(3,2,300,15), S_6(4,1,200,20), S_7(4,2,200,20)$
情景 3	$S_1(1,2,100,5), S_2(2,1,100,10), S_3(2,2,150,10), S_4(2,2,200,10), S_5(3,1,150,15), S_6(3,2,300,15), S_7(4,1,200,20), S_8(4,2,200,20)$
情景 4	$S_1(1,2,75,5), S_2(1,2,100,5), S_3(2,1,150,10), S_4(2,2,150,10), S_5(2,2,200,10), S_6(3,1,150,15), S_7(3,2,150,15), S_8(4,1,200,20), S_9(4,2,200,20)$
情景 5	$S_1(1,1,50,5), S_2(1,2,75,5), S_3(1,2,100,5), S_4(2,1,100,10), S_5(2,2,150,10), S_6(2,2,200,10), S_7(3,1,150,15), S_8(3,2,150,15), S_9(4,1,200,20), S_{10}(4,2,200,20)$

图 2 是得到的实验结果。可以看出,本文提出的面向多业务的拥塞控制策略面向服务的拥塞控制 (Service-oriented Congestion Control, SOCC) 在完成所有业务时所需时间最短,其次是 ECA,性能最差的是 AIMD。之所以取得更好的性能主要是因为初始时,没有像 ECA 那样不分业务地降低所有的 Interest 包发送速率,SOCC 采用的方式是从对时延最不敏感的背景类业务开始进行调节。实验结果证明了面向多业务的拥塞控制策略可以提高业务的总体完成时间。

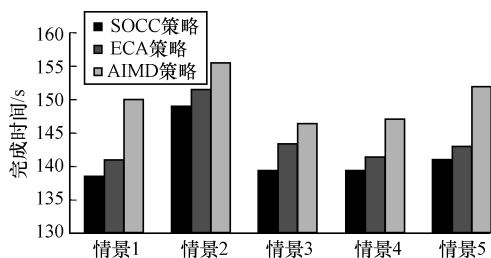


图 2 业务完成时间比较

3.2 受影响业务比例

本文提出的面向多业务的拥塞控制策略不仅能够降低业务的完成时间,而且能够有效提高服务质量。如上文所述,以往的拥塞控制算法都没有考虑业务因素,这样带来的后果有可能严重降低服务质量,影响用户体验。而 SOCC 是在考虑业务的情

况下进行的拥塞控制,基本不会影响对实时性要求很高的业务,同时根据业务的优先级,为用户提供不同等级的服务质量。实验过程中统计了受到网络拥塞影响的用户的比例,实验结果如图 3 所示。由图中可以看出,情景 1~情景 3 中有一个业务受到影响,情景 4 和情景 5 中有 2 个业务受到影响。而以往的拥塞控制算法,如 ECA,AIMD 不区分业务进行拥塞控制,因此,所有的业务都受到了拥塞控制的影响。

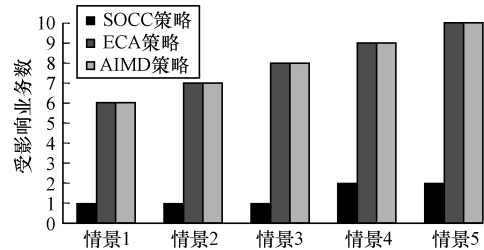


图 3 受影响业务数比较

4 结束语

本文研究多种业务在内容中心网络中的拥塞控制问题。为了尽量不影响业务的服务质量,提出根据不同业务类型和优先级进行拥塞控制的策略。通过在中间路由节点检查兴趣包队列,根据业务类型和优先级判断和处理拥塞。实验结果表明,该策略可以在保障服务质量和不增加业务完成时间的前提下解决拥塞问题。下一步工作内容包括增加业务的动态性,以及内容的分布与热度等因素,使实验结果更符合真实网络环境。

参考文献

- [1] Ahlgren B, Dannewitz C, Imbrenda C, et al. A Survey of Information-centric Networking[J]. IEEE Communications Magazine, 2012, 50(7): 26-36.
- [2] Jacobson V, Smetters D K, Thornton J D, et al. Networking Named Content[C]//Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies. New York, USA: ACM Press, 2009: 1-12.
- [3] Koponen T, Chawla M, Chun B G, et al. A Data-oriented (and Beyond) Network Architecture[J]. ACM SIGCOMM Computer Communication Review, 2007, 37(4): 181-192.
- [4] 曲桦, 王伟萍, 赵季红. 内容中心网络中一种改进型缓存机制[J]. 计算机工程, 2015, 41(3): 41-46.
- [5] 刘涛, 程东年, 田铭. 基于副本通告的内容中心网络快捷路由机制[J]. 计算机工程, 2014, 40(5): 62-67.
- [6] Carofiglio G, Gallo M, Muscariello L. ICP: Design and Evaluation of an Interest Control Protocol for Content-centric Networking[C]//Proceedings of IEEE Conference on Computer Communications Workshops. Washington D. C., USA: IEEE Press, 2012: 304-309.

(下转第 87 页)

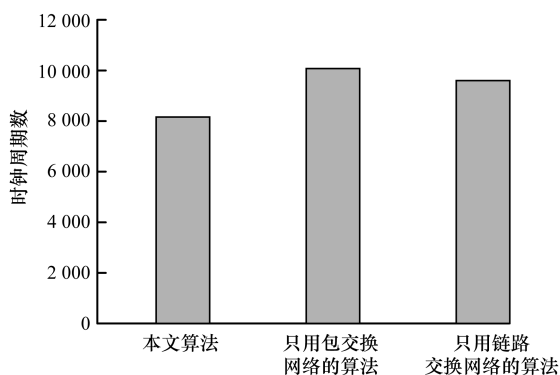


图 8 不同路由算法下 LTE 同步模块的时钟周期数

5 结束语

为了合理利用双层片上网络,提高系统吞吐率和应用程序性能,本文提出一种快速路由算法。通过对核间通信进行分类,为核间通信确定合理的通信方式,并采用基于 FPGA 布线算法的方法为使用链路交换网络进行通信的核间通信选择合适的通信路径。实验结果表明,该算法能够有效提高通信效率与应用程序性能。下一步工作将研究共享存储器在核间通信中的作用,考虑共享存储器与片上网络的结合。

参考文献

- [1] Majed C, Sami Y, Claude T, et al. A Hypercube-based NoC Routing Algorithm for Efficient All-to-all Communications in Embedded Image and Signal Processing Applications [C]//Proceedings of International Conference on High Performance Computing and Simulation. Washington D. C. , USA: IEEE Press, 2010:623-630.
- [2] Yu Zhiyi, You Kaidi, Xiao Ruijin, et al. An 800 MHz 320 mW 16-core Processor with Message-passing and Shared-memory Inter-core Communication Mechanisms[C]//Proceedings of IEEE International Solid-state Circuits Conference. Washington D. C. , USA: IEEE Press, 2012:64-66.
- [3] Mattson T G, Wijngaart R F V D, Riepen M, et al. The 48-core SCC Processor: The Programmer's View[C]//Proceedings of ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis. New York, USA: ACM Press, 2010:1-11.
- [4] Bell S, Edwards B, Amann J, et al. TILE64 Processor: A 64-core SoC with Mesh Interconnect[C]//Proceedings of IEEE International Solid-state Circuits Conference. Washington D. C. , USA: IEEE Press, 2008:88-598.
- [5] 俞剑明. 容偏差百核片上网络设计与多核粒度建模[D]. 上海: 复旦大学, 2014.
- [6] Hu Jingcao, Marculescu R. Energy and Performance-aware Mapping for Regular NoC Architectures[J]. IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, 2005, 24(4): 551-562.
- [7] 秦臻南. 程序并行划分算法及自动并行工具框架实现[D]. 上海: 复旦大学, 2013.
- [8] 欧 鹏. 多核处理器关键技术研究——单核及核间通讯的架构与实现[D]. 上海: 复旦大学, 2013.
- [9] Chang Kuei-Chung, Shen Jih-Sheng, Chen Tien-Fu. Evaluation and Design Trade-offs Between Circuit-switched and Packet-switched NoCs for Application-Specific SoCs[C]//Proceedings of Design Automation Conference. New York, USA: ACM Press, 2006: 143-148.
- [10] Ou Peng, Zhang Jiajie, Quan Heng, et al. A 65nm 39GOPS/W 24-Core Processor with 11Tb/s/W Packet Controlled Circuit-switched Double-layer Network-on-Chip and Heterogeneous Execution Array [C]//Proceedings of IEEE International Solid-state Circuits Conference. Washington D. C. , USA: IEEE Press, 2013: 56-57.
- [11] Swartz J, Betz V, Rose J. A Fast Routability-driven Router for FPGAs [C]//Proceedings of the 6th International Symposium on FPGAs. New York, USA: ACM Press, 1998:140-151.
- [12] Ebeling C, McMurchie L, Hauck S A, et al. Placement and Routing Tools for the Triptych FPGA [J]. IEEE Transactions on Very Large Scale Integration Systems, 1995, 3(4): 473-482.
- [7] Salsano S, Detti A, Cancellieri M, et al. Transport-layer Issues in Information Centric Networks[C]//Proceedings of the 2nd ICN Workshop on Information-centric Networking. New York, USA: ACM Press, 2012:19-24.
- [8] Rozhnova N, Fdida S. An Effective Hop-by-hop Interest Shaping Mechanism for CCN Communications [C]//Proceedings of IEEE Conference on Computer Communications Workshops. Washington D. C. , USA: IEEE Press, 2012: 322-327.
- [9] Carofiglio G, Gallo M, Muscariello L. Joint Hop-by-hop and Receiver-driven Interest Control Protocol for Content-centric Networks [C]//Proceedings of the 2nd ICN Workshop on Information-centric Networking. New York, USA: ACM Press, 2012:37-42.
- [10] Saino L, Cocora C, Pavlou G. CCTCP: A Scalable Receiver-driven Congestion Control Protocol for Content Centric Networking [C]//Proceedings of IEEE International Conference on Communications. Washington D. C., USA: IEEE Press, 2013:3775-3780.
- [11] 唐 潇, 任勇毛, 李 俊, 等. 一种基于显式反馈的内容中心网络 NDN 拥塞控制算法[J]. 科研信息化技术与应用, 2014, 5(3): 68-77.
- [12] 3GPP. Quality of Service (QoS) Concept and Architecture[EB/OL]. [2015-03-03]. <http://www.3gpp.org/Dyna-Report/23107.htm/>.

编辑 刘 冰

(上接第 82 页)

编辑 刘 冰