

内容中心网络状态感知路由设计

蔡岳平, 刘军

(重庆大学通信工程学院, 重庆 400030)

摘要: 为了提高内容中心网络的内容分发效率及降低网络开销, 提出了网络状态感知的路由机制 NSAR(network status aware routing)。NSAR 利用从内容服务节点返回的数据分组收集当前网络状态信息, 并在回传过程中对路径上各节点上匹配端口的转发概率进行更新, 在对后续的兴趣分组进行转发决策时引入转发概率, 从而提高内容分发效率。仿真实验表明, 与传统内容中心网络路由算法相比, NSAR 可以有效地降低内容请求平均时延, 减少网络流通分组数以及降低网络带宽开销。

关键词: 内容中心网络; 路由机制; 网络状态; 状态感知; 转发概率

中图分类号: TP393

文献标识码: A

Network status aware routing in content-centric network

CAI Yue-ping, LIU Jun

(College of Communication Engineering, Chongqing University, Chongqing 400030, China)

Abstract: To improve the efficiency of content delivery and to reduce the network overhead of content centric network (CCN), a network status aware routing (NSAR) mechanism was proposed. NSAR utilized data packets from the content server nodes to collect network statuses. The forwarding probability of the matching ports of the nodes on the path would be updated according to the network statuses. The subsequent interest packet forwarding would be based on the updated forwarding probability. Therefore the content delivery efficiency would be improved. Simulation results show that NSAR can effectively reduce the average delay of content requests and the number of packets in network as well as the network bandwidth overhead compared with the traditional routing algorithm in CCN.

Key words: content-centric network, routing mechanism, network status, status aware, forwarding probability

1 引言

互联网实现端到端的互联互通和资源共享, 使人类通信方式得到前所未有的革命。然而随着网络流量的指数式增长以及用户需求的不断提高^[1,2], 以主机为中心的传统网络越来越难以满足未来网络的需求。为了从根本上解决这一问题, 研究者们提出了以信息为中心的未来网络架构, 典型的有 DONA^[3]、PURSUIT^[4]、PSIRP^[5]、SAIL^[6]、4WARD^[7]、

COMET^[8]、CONVERGENCE^[9]、NDN^[10]、CCN^[11]和 MobileFirst^[12]。其中, CCN (content centric network) 正逐渐被认为是最有前途的方案之一, 它是由 PARC^[13]的 Van Jacobson 于 2009 年提出的, 其核心是通过对内容资源的直接命名以及基于内容名称的路由来进行内容的分发和获取, 其网络节点除了具有传统网络节点所具有的路由和转发能力外, 还具备存储内容资源及服务内容请求的功能, 节点性能较高。

收稿日期: 2015-08-07; 修回日期: 2015-12-23

基金项目: 国家自然科学基金资助项目 (No.61301119); 教育部—中国移动科研基金资助项目 (No.MCM20150102); 教育部高等学校博士学科点专项科研基金资助项目 (No.20120191120025); 教育部留学归国人员启动基金资助项目 (No.1020607820140002)

Foundation Items: The National Natural Science Foundation of China (No.61301119), Joint Research Fund of Ministry of Education and China Mobile (No.MCM20150102), Research Fund of Young Scholars for the Doctoral Program of Ministry of Education (No.20120191120025), Research Fund for Returned Overseas Chinese Scholars of Education of Ministry (No.1020607820140002)

CCN 有 2 种基本的分组格式,即兴趣分组(interest packet)和数据分组(data packet)。兴趣分组是请求者发出的内容请求分组;数据分组是内容服务节点(内容发布者或网络缓存)将内容传输给请求者的内容分组。每个路由节点都需要维护 3 类信息表,即转发信息表(FIB, forwarding information base)、待定兴趣表(PIT, pending interest table)和内容存储表(CS, content store)。FIB 保存了内容名称前缀和到达此前缀代表的内容的下一跳端口;PIT 记录了兴趣分组的输入端口,该信息表为数据分组提供回传路径;CS 缓存流经该节点的内容资源并为后续内容请求提供服务。内容发布者以洪泛的方式向网络发布内容资源的注册信息,路由节点依据接收到的注册信息中的内容名称前缀以及注册信息的到达端口建立 FIB,路由节点通过查询 FIB 来决定兴趣分组的转发端口。

由于在 CCN 网络中,每个内容资源可能存在多个提供者,而且内容注册信息是通过洪泛方式发布,所以在 CCN 的 FIB 中的每一个内容名称可能对应多个转发端口。如何选取合适的转发端口转发兴趣分组是 CCN 研究中的一项重要课题。常用的转发机制有全转发机制^[14]、随机转发机制^[15]和最短路径转发机制^[16]。全转发机制的路由节点会将兴趣分组转发至 FIB 中匹配条目所对应的所有转发端口,虽然这可以利用多径路由特点从最优的内容服务节点处获得内容资源,但是过多的转发会导致网络中产生大量的冗余流量。随机转发机制的路由节点会在 FIB 对应条目中随机选择转发端口进行兴趣分组的转发,这种方法虽然能有效地减少网络中的冗余流量,降低网络开销,但无法保证用户能稳定、快速地从最优的路径上获取资源,服务质量很难保

证。最短路径转发机制是文献[16]中提出的邻居缓存探测方法(NCE),它将兴趣分组转发至距离当前节点路径最短的内容服务节点,虽然 NCE 方法可以获得路径最短的链路作为路由路径,但并不能保证路径最优,因其没有考虑链路状态和节点负载等其他网络状态信息,只选择内容获取跳数作为路径选择的依据,无法反映全面的网络状态,也就难以获得网络状态最优的路径。本文提出了一种网络状态感知路由机制(NSAR, network status aware routing),通过对网络中的链路时延、节点负载以及内容获取跳数等状态进行感知,综合考虑各网络状态并优化选择兴趣分组的转发端口,从而实现在网络开销较低的情况下达到路径最优的目的。

本文的贡献总结如下。

1) 设计网络状态感知机制,实现对链路延时、节点负载及内容获取跳数等网络状态感知功能,并对感知获得的状态信息进行综合处理,得到网络状态综合参数。

2) 针对 FIB 中是否有新增转发端口 2 种情况设计 2 种不同的端口转发概率更新算法,使用网络状态综合参数对各端口的转发概率进行更新,并依据更新后的转发概率设计兴趣分组的转发方案。

2 网络状态感知路由

本节将对网络状态感知路由机制 NSAR 的设计进行介绍。

图 1 是对 NSAR 整体结构的简要描述。NSAR 分成 3 个模块:第 1 部分是网络状态感知模块,NSAR 通过对 3 个网络状态信息,即链路时延、节点负载及内容获取跳数进行感知,并把感知获得的状态信息交给网络状态信息库;第 2 个模块是 FIB

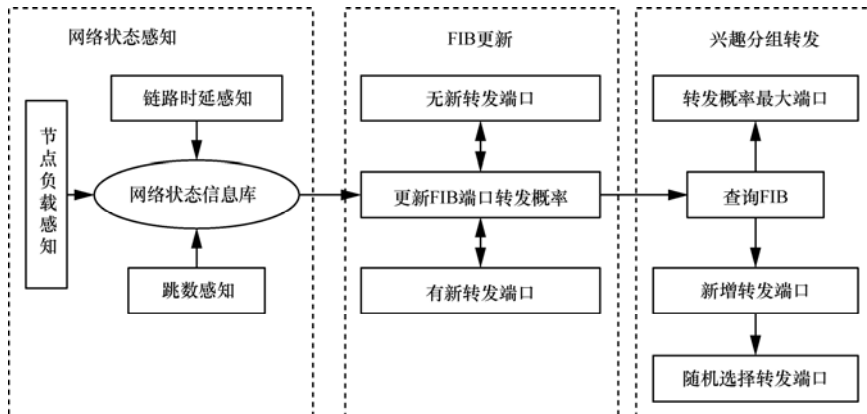


图 1 NSAR 整体结构

更新模块, NSAR 根据网络状态感知模块感知获得的网络状态信息库对 FIB 中的端口转发概率进行更新, 此时分 2 种情况, 一是 FIB 中无新增的转发端口, 二是 FIB 中有新增的转发端口, 两者的更新算法不同, 后面将进行详细介绍; 第 3 个模块是兴趣分组转发模块, 在这个模块中, NSAR 根据 FIB 更新模块更新后的 FIB 转发兴趣分组, 首先将兴趣分组转发至转发概率最大的端口, 然后查询 FIB 中是否有新增的转发端口, 若有, 则将兴趣分组转发至该端口, 随后从除转发概率最大及新增的端口外随机选择若干个端口转发兴趣分组, 随机选择端口数量由探测深度确定, 用于探索其余端口状态。

2.1 网络状态感知

为获取最优的兴趣分组转发路径, 需要对网络状态进行感知。传统的路由机制把链路长度、链路时延、网络拥塞、可用带宽以及节点负载作为主要的网络状态性能指标。本文选取链路时延、节点负载以及内容获取跳数作为网络性能的评价指标。

1) 链路时延

传统的路由机制将路径长度作为路由选择的影响因素。但由于网络拥塞的存在, 路径长度并不能很好地反映网络的状态, 当路径较短, 但网络拥塞较大时, 链路时延可能较大, 对网络性能的影响也较大。链路时延能更好地反应网络状态, 所以把链路时延作为网络状态的评价标准之一。

在本文的路由设计中, 链路时延指当前节点与内容服务节点之间的时延, 由于数据分组的大小远远大于兴趣分组, 所以本文将数据分组的在回传路径上的传播时延作为链路时延参考值。在服务节点对数据分组进行封装时, 会在数据分组内设置链路时延计时器并置零, 当数据分组发出时, 计时器开始计时, 当数据分组到达网络中的路由节点后, 链路时延计时器不清零, 继续计时, 并将当前的链路时延值交给路由节点的网络状态信息库, 以供后续的 FIB 更新模块更新端口转发概率提供数据来源。当数据分组从端口转发出去后, 链路时延计时器继续计时, 直至到达请求者后被删除。

2) 节点负载

节点负载反映网络节点当前处理情况的状态信息。节点的负载情况对数据分组在节点处的处理时延呈正相关。好的路由机制应尽可能将用户请求(即兴趣分组)路由至节点负载较低的网络节点中, 避免因节点过载而导致处理时延增加甚至数据

丢失。

在本文的路由设计中, 主要考虑两方面的节点负载。第一是普通路由节点的负载状况, 数据分组在回传过程中会对路径上路由节点的负载信息进行采集, 并将该信息交给数据分组到达的下一跳节点。第二是内容服务节点的负载状况, 在内容服务节点处封装数据分组时, 当前内容服务节点的负载值会被封装到数据分组中。在数据分组回传过程中, 该状态值会依次交给沿途路由节点的网络状态信息库, 直至到达请求者后被删除。

3) 内容获取跳数

内容获取跳数是指数据分组从路由节点到内容服务节点所需的转发次数。跳数越少, 数据分组被存储转发的次数也就越少, 在节点处的排队时延也就越小。好的路由机制应尽可能使数据分组的转发次数越少, 降低排队时延, 提高内容获取的速度。

在本文的路由设计中, 内容获取跳数也是指当前路由节点到内容服务节点的转发次数。该状态值是在数据分组回传过程中进行统计获得。当数据分组在内容服务节点处被封装时, 会额外加入跳数计数器, 数据分组每被转发一次, 计数值加 1, 并将当前计数值交给路由节点的网络状态信息库, 直至到达请求者后将该计数器删除。

上述各网络状态参数无法单独对网络整体状态进行描述, 为更好地反映网络的整体状态, 本文综合考虑以上各状态参数的影响, 并得到网络状态综合参数, 以实现在网络状态感知下对 CCN 路由进行优化。

2.2 网络状态感知路由机制

网络状态感知路由 NSAR 可以通过数据分组感知传输路径及服务节点的状态信息, 感知的状态信息主要包含 3 个方面: 链路时延、节点负载以及内容获取跳数。NSAR 可以利用感知获得的网络状态信息来更新 FIB 中匹配端口的转发概率, 用于区分各转发端口对应的转发路径的优劣, 端口转发概率越大, 说明其所对应的转发路径的网络状态越好。后续的兴趣分组依据端口转发概率选择转发端口, 实现在网络开销较低的情况下达到路径最优的目的。NSAR 的路由机制如图 2 所示。

网络初始时, 内容服务节点 S_1 以洪泛的方式向网络发布内容对象 O 的注册信息, 由此在各路由节点上建立的 FIB 如图 2(a)所示。FIB 第 1 列表示内容对象名称前缀, 为叙述方便, 本文将其设为 O ;

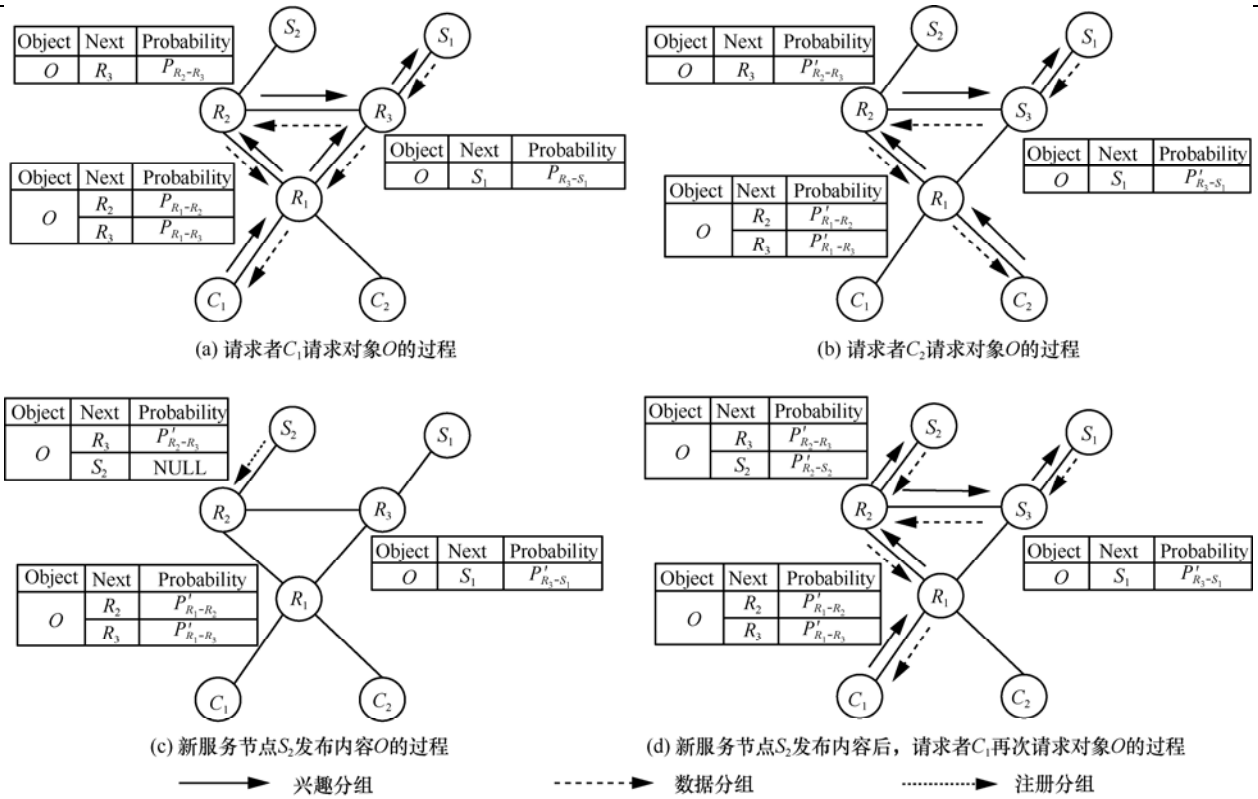


图 2 NSAR 的路由机制

第 2 列表示获取该内容对象的转发端口，图中以转发端口所对应的下一跳节点名称作为转发端口的名称；第 3 列表示端口的转发概率，表示当前路由节点将匹配该内容对象名称的兴趣分组从该端口转发出去的概率。对于在某一路由节点上同一内容对象名称的所有转发端口，其转发概率和为 1，即

$$\sum_{i=0}^N P_i = 1, 0 \leq P_i \leq 1 \quad (1)$$

其中， P_i 表示转发端口 i 的转发概率， N 为在某路由节点上某内容对象名称对应的转发端口数量。

当请求者 C_1 发出请求对象 O 的兴趣分组后，如图 2 (a) 所示，路由节点 R_1 接收该兴趣分组，首先在 FIB 中进行匹配，并获得下一跳转发端口 R_2 和 R_3 。此时，路由节点 R_1 并不会立即将兴趣分组从端口 R_2 和 R_3 转发出去，而是对比这 2 个转发端口的转发概率，即 $P_{R_1-R_2}$ 和 $P_{R_1-R_3}$ 。由于网络初始时，同一内容对象的转发端口的转发概率相等，为转发端口数的倒数 $\frac{1}{N}$ ，所以此时 R_1 会同时将兴趣分组

从这 2 个端口转发出去。兴趣分组在路由节点 R_2 和 R_3 以同样的方式进行选择性转发，最后兴趣分组被转发至内容服务节点 S_1 。在内容服务节点获得数

据对象 O 后，封装数据分组，并开启链路时延计时器和内容获取跳数计数器，获取当前内容服务节点负载值。数据分组在回传过程中会将链路时延值、内容获取跳数值、上一跳节点负载值以及服务节点负载值交给当前节点的网络状态信息库。

当节点 n 在端口 i 收到封装了内容对象 O 的数据分组后，会从中提取链路时延值 T_i 、内容获取跳数值 H_i 、上一跳节点负载值 L_i 以及服务节点负载值 L_s 作为最新的网络状态信息样本值。为了获得各个网络状态值对网络的综合影响，采用加权法获得网络状态综合参数

$$\Delta S_i^n = \frac{1}{\alpha T_i + \beta H_i + \gamma L_i + \varphi L_s} \quad (2)$$

其中， α 、 β 、 γ 和 φ 分别表示链路时延、内容获取跳数、路由节点负载和服务节点负载的影响因子，其值可以依据不同的业务需求进行调整。因本文路由机制只考虑这 4 个影响因素，且对各状态值均作归一化处理，所以 4 个状态影响因子和为 1，即

$$\alpha + \beta + \gamma + \varphi = 1 \quad (3)$$

由式(2)可以看出，随着链路时延、内容获取

跳数、路由节点负载及服务节点负载的增加，网络状态综合参数值减小，易知， ΔS_i^n 恒大于 1，所以对于没有数据分组到达的端口，即没有被探测的端口，将 ΔS_i^n 定为 1。确定各个端口的网络状态综合参数后，需要确定各端口的概率变化值，为综合考虑各个端口的状态值对当前路由节点的端口转发概率值更新的影响，将各端口的转发概率增值定为

$$\Delta P_n(i) = \frac{\Delta S_i^n}{\sum_{i=1}^N \Delta S_i^n} \quad (4)$$

其中， N 为路由节点 n 上某内容对象名称对应的转发端口数量。由式 (2) 和式 (4) 可以看出，当某端口所对应的路径的链路时延、内容获取跳数、路由节点负载及内容服务节点负载值较大时，其状态综合参数较小，导致其端口转发概率增值也较小。

确定各端口的转发概率增值后，进一步对各端口的转发概率进行更新。考虑网络的突变性，本文使用加权平均转发概率，即每得到一个新的端口转发概率增值，就用下式对端口转发概率进行更新

$$P_{n\text{-new}}(i) = (1 - \rho)P_{n\text{-old}}(i) + \rho\Delta P_n(i) \quad (5)$$

其中， $P_{n\text{-new}}(i)$ 表示路由节点 n 的端口 i 更新后的端口转发概率， $P_{n\text{-old}}(i)$ 表示更新前的端口转发概率， ρ 是网络状态影响因子。 $0 < \rho < 1$ ，若 ρ 越接近于零，表示新的端口转发概率值和旧的端口转发概率值相比变化不大，而受新的端口概率增值影响较小。若 ρ 越接近于 1，则表示新的端口转发概率值受新的端口转发概率增值影响较大。

当数据分组被回传到请求者处后，路径上的所有路由节点对应的端口的转发概率均被更新。当下一个请求者 C_2 发出请求内容对象 O 的兴趣分组后，如图 2(b) 所示，路由节点 R_1 接收该兴趣分组，并在 FIB 中匹配获得下一跳转发端口 R_2 和 R_3 ，随后对比 R_2 和 R_3 的端口转发概率，假设 $P'_{R_1-R_2} > P'_{R_1-R_3}$ ，则路由节点 R_1 会将兴趣分组从端口 R_2 转发出去，在其他路由节点的经过相同的处理后，兴趣分组将被转发至内容服务节点 S_1 ，数据分组在回传过程中也会对路径上的所有路由节点的端口转发概率进行更新。

当有除内容服务节点 S_1 外的其他内容服务节点发布内容对象 O 时，如图 2(c) 所示，内容服务节点 S_2 对外发布内容对象 O ，其发布信息会以洪泛

的方式在网络中传播，直至到达原有的该内容对象的传播路径的节点上后停止传播（图 2(c) 中 R_2 节点）。此时，NSAR 会依据注册信息的到达端口在原有的 FIB 中增加内容对象 O 的下一跳转发端口，并将该端口的转发概率初始化为 NULL。

当有请求对象 O 的兴趣分组到达该路由节点后，如图 2(d) 所示，路由节点 R_2 除了将兴趣分组转发至端口转发概率最大 ($P_{R_2-R_3}$) 的端口外，还会将兴趣分组转发至端口转发概率为 NULL 的转发端口。当数据分组从该路径返回时，也会携带该路径的网络状态信息，用于端口转发概率的更新。当路由节点收集完所有转发端口回收的状态信息后，由于在原有转发端口的基础上增加了一个或多个转发端口，所以对原有的端口转发概率进行重新分配。其分配方案如下。

提取各转发端口的网络状态值，并依据式(2)求得各端口的状态综合参数， m 为 NULL 个数。

$$\Delta S^n = \{\Delta S_1^n, \Delta S_2^n, \dots, \Delta S_i^n, \dots, \Delta S_N^n, \Delta S_{N+m}^n\}$$

取得各端口的状态综合参数后，对于新增的转发端口，由于其在此之前没有转发概率值，所以其转发概率值更新为

$$P_{n\text{-new}}(N + j) = \rho \frac{\Delta S_{N+j}^n}{\sum_{i=1}^{N+m} \Delta S_i^n}, j \in [1, m] \quad (6)$$

而对于其他转发端口，其转发概率更新为

$$P_{n\text{-new}}(i) = (1 - \rho)P_{n\text{-old}}(i) + \rho \frac{\Delta S_i^n}{\sum_{i=1}^{N+m} \Delta S_i^n}, i \in [1, N] \quad (7)$$

为了防止所有的兴趣分组都往端口转发概率最大的端口转发而使该端口的转发概率值持续为最大值，使其他端口不被探索。当网络状态发生改变时，最优路径可能发生改变，需要对转发概率最大外的其他端口进行探索。所以，当兴趣分组到达路由节点后，执行如下转发规则。

- 1) 将兴趣分组往转发概率最大的端口转发。
- 2) 若有端口转发概率为 NULL 的端口，将兴趣分组往该端口转发。
- 3) 设定探测深度 n_{deep} ， n_{deep} 表示除转发概率最大及转发概率为 NULL 的端口外，NSAR 随机选择的转发兴趣分组的端口数量。当 $n_{\text{deep}}=n$ 时，表示 NSAR 选择的转发端口有转发概率最大的端口 $Port_{\text{max}}$ 、转发概率为 NULL 的端口 $Port_{\text{NULL}}$ 及 n 个

随机选择的端口 $P_1 \sim P_n$ 。易知, 探测深度 n_{deep} 越大, 表示 NSAR 选择转发的端口越多, 对网络状态信息的探索越详细, 但相应的网络开销越大。NSAR 会将兴趣分组转发至随机选择的 n 个探索端口用于探索网络状态信息。

2.3 算法实现

下面对 NSAR 的算法进行分析。

算法 1 是对 NSAR 的路由转发算法描述。为了方便, 不妨设路由节点 R 为 NSAR 的处理节点。 G 表示内容中心网络拓扑, I 表示到达节点 R 的兴趣分组, FIB 表示节点 R 上的转发信息表。当一个新的兴趣分组到达路由节点 R 时, 算法被触发。算法需要为每个目标兴趣分组提供可用路径以及对该分组的处理办法 (丢弃或路由)。首先, 当新的兴趣分组到达后, 节点 R 会提取分组内的内容名称, 并以最大前缀匹配原则在 FIB 中对提取的内容名称进行匹配。获得匹配项后, 再对各匹配项后的转发概率值进行分析比较。若转发概率值中有值为 NULL 的选项, 则节点将会把兴趣分组转发至转发概率最大的端口, 转发概率为 NULL 的端口, 以及随机选择的 n_{deep} 个随机端口; 若转发概率值中没有值为 NULL 的选项, 则节点除了将把兴趣分组转发至转发概率最大的端口以及随机选择的 n_{deep} 个随机端口。算法 1 的伪代码如下。

算法 1 NSPR 路由转发算法

输入 G : 内容中心网络拓扑;

I : 到达节点 R 的兴趣分组;

FIB : 路由节点的转发信息表

输出 $\{ \langle e.method, e.port \rangle, \forall e \in I \}$; // 兴趣分组的处理方法和转发端口

1) If 一个新的兴趣分组 I 到达 then

2) $ContentName \leftarrow GetContentName(I)$; // 获得内容名称

3) $PORT \leftarrow GetMatchPort(ContentName, FIB)$; // 获得内容名称的匹配端口

4) $P \leftarrow GetPortProbability(PORT, FIB)$; // 获得匹配端口的端口转发概率

5) If ($PORT \neq \Phi$) then

6) $MaxProbabilityPort \leftarrow GetMaxProbabilityPort(P)$; // 获得转发概率最大的端口

7) $NullPort \leftarrow GetProbabilityEqualsNullPort(PORT, P)$; // 获得转发概率为 NULL 的端口

8) $RandomPORT \leftarrow GetRandomPort(PORT, n_{\text{deep}})$; // 随机获取 n_{deep} 个转发端口

9) If ($NullPort \neq \Phi$) then

10) $ForwardingPort \leftarrow \{MaxProbabilityPort, NullPort, RandomPORT\}$; // 转发端口为最大转发概率端口, 转发概率为 NULL 的端口及随机选择的端口

11) Else

12) $ForwardingPort \leftarrow \{MaxProbabilityPort, RandomPORT\}$; // 转发端口为最大转发概率端口和随机选择的端口

13) End

14) Return $\langle Forwarding, ForwardingPort \rangle$; // 返回转发方法与转发端口

15) Else

16) Return $\langle Drop, NULL \rangle$; // 返回丢弃方法

算法 2 是对 NSAR 的端口转发概率更新算法描述。本文的转发概率更新是通过在路由节点上实现的。对于每个到达端口 i 的包含对象 O 的数据分组, 它除了携带传统 CCN 的数据分组所携带的信息外, 还额外携带了内容服务节点到达节点 R 的链路时延值, 内容服务节点负载值, 上一跳的节点负载值以及内容服务节点到达节点 R 的跳数。当一个新的数据分组到达路由节点 R 时, 算法被触发。对于每个目标数据分组, 节点 R 首先提取内容对象的名称、内容服务节点到达节点 R 的链路时延值 T_i 、内容服务节点负载 L_s 、上一跳的节点负载值 L_i 以及内容服务节点到达路由节点 R 的跳数 H_i 。随后在 FIB 中匹配内容名称, 获得匹配项后, 分析各匹配项后的转发概率值。若转发概率值中不含 NULL 选项, 则触发子算法 $ProbabilityUpdate(\cdot)$ 来对各端口的转发概率进行更新; 若转发概率值中含有 NULL 选项, 则触发 $ProbabilityUpdateWithNull(\cdot)$ 来对各端口的转发概率进行更新。算法 2 的伪代码如下。

算法 2 NSPR 端口转发概率更新算法

输入 G : 内容中心网络拓扑;

$D(i)$: 从端口 i 到达节点 R 的数据分组;

FIB : 路由节点的转发信息表

输出 $\{ \langle p.probability \rangle, \forall p \in (\text{对象 } O \text{ 的匹配端口}) \}$; // 匹配端口的更新后转发概率

1) If 一个新的数据分组 D 到达 then

2) $ContentName \leftarrow GetContentName(D(i))$; // 获得内容名称

3) $LinkDelay(i) \leftarrow GetLinkDelay(D(i))$; // 获得服

务节点到达节点 R 链路时延

4) $L_s \leftarrow GetServerLoad(D(i));$ //获得服务节点负载

5) $L(i) \leftarrow GetLastHopLoad(D(i));$ //获得节点 R 上一跳节点的负载

6) $H(i) \leftarrow GetHopNumber(D(i));$ //获得服务节点到达节点 R 跳数

7) $PORT \leftarrow GetMatchPort(ContentName, FIB);$ //获得内容名称的匹配端口

8) $P_{old} \leftarrow GetPortProbability(PORT, FIB);$ //获得匹配端都是旧的端口转发概率

9) $NullPORT \leftarrow GetProbabilityEqualsNullPort(PORT, P_{old});$ //获得转发概率为 NULL 的端口

10) If ($NullPORT \neq \Phi$) then

11) $ProbabilityUpdateWithNull(LinkDelay(i), L(i), H(i), L_s, P_{old});$ //执行有转发概率为 NULL 的更新函数

12) Else

13) $ProbabilityUpdateLinkDelay(i), L(i), H(i), L_s, P_{old});$ // 执行无转发概率为 NULL 的更新函数

14) End

15) Return $\langle P_{new} \rangle;$ //返回更新后的端口转发概率值

16) End

算法 3 是在匹配项的转发概率值中不含 NULL 值时的端口转发概率更新算法。在此算法中，当路由节点获得网络状态值后，首先根据式 (2) 将各网络状态综合获得状态综合参数；然后根据各端口的网络状态综合参数依据式 (4) 计算得各端口的转发概率增值；最后依据式 (5) 对各端口的转发概率进行更新，返回更新后的端口转发概率值。算法 3 的伪代码如下。

算法 3 不含 NULL 项端口转发概率更新算法

输入 $LinkDelay(i)$: 从端口 i 的数据分组中提取的链路延时;

$L(i)$: 从端口 i 的数据分组中提取的上一跳节点的负载;

$H(i)$: 从内容服务节点到达节点 R 的跳数;

$L(s)$: 内容服务节点的负载;

P_{old} : 更新前的端口转发概率

输出 $\{ \langle p.probability \rangle, \forall p \in (\text{对象 } O \text{ 的匹配端口}) \}$ //匹配端口的更新后转发概率

1) if(所有转发了兴趣分组的端口均收到数据分组) then

2) for($i=1; i \leq N; i++$)//对于对象 O 所有匹配端口

3) If(端口 i 有数据分组返回)

4) $VS_i \leftarrow \frac{1}{\alpha T(i) + \beta H(i) + \gamma L(i) + \varphi L(s)}$;

//利用返回的网络状态求得综合影响因子

5) Else

6) $VS_i \leftarrow 1;$ //将没有数据分组返回的端口的综合影响因子置 1

7) End

8) End for

9) for($i=1; i \leq N; i++$)//对于对象 O 所有匹配端口

10) $VP(i) = \frac{VS_i}{\sum_{i=1}^N VS_i}$; //求得各端口的转发概率增值

11) $P_{new}(i) = (1 - \rho)P_{old}(i) + \rho VP(i);$ //求得各端口新的转发概率

12) End for

13) Return $\langle P_{new} \rangle;$ 返回更新后的端口转发概率值

14) End

算法 4 是在匹配项的转发概率值中含 NULL 值时的端口转发概率更新算法。在此算法中，当路由节点获得网络状态值后，首先根据式(2)将各网络状态综合获得网络状态综合参数；然后对内容对象 O 的匹配端口的转发概率作分类处理，对于端口的转发概率值为 NULL 的端口，其转发概率直接使用此次获得的网络状态影响因子进行计算，即按式(6)进行更新；而对于其他转发端口，需要使用先前的转发概率和此次获得的网络状态因子，即按式(7)进行更新，最后返回更新后的端口转发概率值。算法 4 的伪代码如下。

算法 4 含 NULL 项端口转发概率更新算法

输入 $LinkDelay(i)$: 从端口 i 的数据分组中提取的链路延时;

$L(i)$: 从端口 i 的数据分组中提取的上一跳节点的负载;

$H(i)$: 从内容服务节点到达节点 R 的跳数;

$L(s)$: 内容服务节点的负载;

P_{old} : 更新前的端口转发概率

输出 $\{ \langle p.probability \rangle, \forall p \in (\text{对象 } O \text{ 的匹配端口}) \}$ // 匹配端口的更新后转发概率

1) if(所有转发了兴趣分组的端口均收到数据分组) then

2) for($i=1; i \leq N+m; i++$) // m 个转发概率为 NULL 的端口

3) If(端口 i 有数据分组返回)

4) $VS_i \leftarrow \frac{1}{\alpha T(i) + \beta H(i) + \gamma L(i) + \phi L(s)}$; //

利用返回的网络状态求得综合影响因子

5) Else

6) $VS_i \leftarrow 1$; // 将没有数据分组返回的端

口的综合影响因子置 1

7) End

8) End for

9) for($i=1; i \leq N+m; i++$) // 对于对象 O 所有匹配端口

10) if(端口转发概率为 NULL)

11) $P_{\text{new}}(i) = \rho \frac{VS_i}{\sum_{i=1}^{N+m} VS_i}$; // 更新转发概率为

NULL 的端口的转发概率

12) Else

13) $P_{\text{new}}(i) = (1 - \rho)P_{\text{old}}(i) + \rho \frac{VS_i}{\sum_{i=1}^{N+m} VS_i}$; // 更新

其他端口转发概率增值

14) End

15) End for

16) Return $\langle P_{\text{new}} \rangle$; 返回更新后的端口转发概率值

17) End

网络状态感知路由机制 NSAR 为各内容条目匹配端口设置转发概率, 并在数据分组回传过程中对其进行更新, 这将给网络节点带来额外的处理开销。对于数据分组传输路径上的每个路由节点, 假设其内容条目的匹配端口数为 n , 在通过端口转发概率更新算法对各匹配端口的转发概率进行更新时, 对于匹配端口的转发概率中不含 NULL 项的节点, 即无新增转发端口的节点, 首先利用式(2)获得每个匹配端口的网络状态综合参数, 然后依据此参数根据式(4)求得各端口的转发概率增值, 最后通过式(5)加权平均求得更新后的转发概率, 分析此

算法可知其时间复杂度为 $O(n)$, 空间复杂度为 $O(n)$; 对于匹配端口的转发概率中含 NULL 项的节点, 即有新增转发端口的节点, 首先利用式(2)获得每个匹配端口的网络状态综合参数, 然后依据此参数根据式(6)求得新增端口的转发概率, 最后通过式(7)加权平均求得其他匹配端口更新后的转发概率, 分析此算法可知其时间复杂度为 $O(n)$, 空间复杂度为 $O(n)$ 。由此看出, 2 种端口转发概率更新算法的时间复杂度及空间复杂度均较低, 其给网络节点带来的额外开销较小。

3 网络状态感知路由机制性能评价

本节将对 NSAR 进行仿真实验分析。通过分析可以知道, NSAR 属于分布式路由机制, 即各节点只在当前节点对信息进行处理并获得下一跳信息, 且路由转发算法及端口转发概率更新算法的复杂度较低, 所以 NSAR 具有良好的可扩展性。NSFNET^[17]是美国国家科学基金会(NSF, National Science Foundation)为了满足各大学及政府机构研究工作的迫切要求, 在全美国建立的连接各大超级计算中心的网络, 该网络已被广泛作为网络研究者实验所用的网络拓扑, 所以本文将 NSFNET 作为仿真所用的网络拓扑。

下面将对对比 CCN 中常用的 3 种路由机制: 全转发机制(FF, full forwarding), 路由节点将内容请求转发至所有的内容条目的匹配端口; 随机转发机制(RF, random forwarding), 路由节点将内容请求转发至随机选择的一个内容条目的匹配端口; 最短路径转发策略(SPF, shortest path forwarding), 路由节点将内容请求转发至内容获取跳数最少的内容服务节点。以内容请求平均时延、网络流通分组数以及内容传输的总带宽开销作为性能指标进行对比分析。最后, 分析探测深度对网络流通分组数及内容传输总带宽开销的影响。

3.1 仿真设置

实验采用美国自然科学基金组建的 NSFNET 作为实验的网络拓扑, 并使用 Matlab 工具进行仿真实验。仿真参数描述: NSFNET 网络包含 14 个网络节点和 21 条链路, 各节点的连接关系如图 3 所示, 链路带宽为 100 Mbit/s。实验时将节点 College PK MD 作为请求者的连接节点, 请求者发出的兴趣分组将会首先到达该节点; 将节点 Salt Lake City 作为服务者的连接节点, 当服务节点发布内容

时，内容注册信息将会首先到达该节点；其余节点为普通的路由节点。

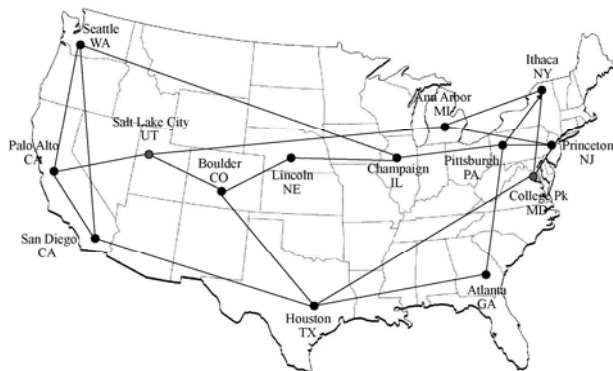


图 3 NSFNET

各仿真参数值如表 1 所示。

表 1 仿真参数值设置

仿真参数	设置值
仿真网络	NSFNET
节点数	14
链路数	21
链路带宽	100 Mbit/s
请求者数量	30
用户带宽需求	均值为 4 Mbit/s 的泊松分布
用户请求频率	均值范围 1~9 个/秒的泊松分布
链路延时因子	0.4
跳数因子	0.2
路由节点负载因子	0.2
服务节点负载因子	0.2
网络状态影响因子	0.5
探测深度	1

3.2 仿真结果

图 4 是对各路由机制的内容请求平均时延进行对比。仿真过程中，对各路径的时延值采用归一化处理，迭代次数为 20 次，并对这 20 次的请求时延求平均得到请求平均时延。从图 4 中可以看出，随着网络负载的增大，4 种路由机制的内容请求平均时延逐渐增加。这是由于随着网络负载的增加，网络发生拥塞的可能性增加，导致网络的时延增加。对比分析可以看出，4 种路由机制中，平均时延最小的是网络状态感知路由机制 NSAR，其次是最短路径转发机制 SPF，而随机转发机制 RF 与全转发

机制 FF 呈交替的状态。NSAR 在选择路由路径时是优先选择转发概率最大的端口进行转发，而转发概率的大小反映的是该路径的网络状态的优劣，所以其内容请求平均时延最小；SPF 选择内容请求跳数最少的路径进行转发，跳数越少，时延叠加越少；FF 将内容请求转发至所有匹配端口，容易造成网络拥塞，导致内容请求平均时延增大。RF 从匹配端口中随机选择转发端口，所以其内容请求平均时延存在波动，但总体比 NSAR 和 SPF 要大。由此看出 NSAR 可以降低内容请求平均时延。

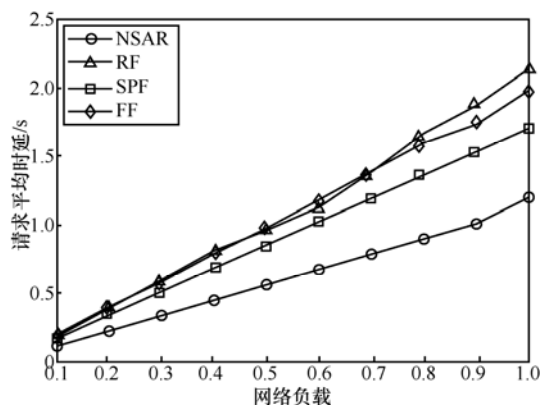


图 4 请求平均时延对比

图 5 是对各路由机制的网络中的流通分组数进行对比。从图中可以看出，随着平均分组到达速率的增加，4 种路由机制的网络流通分组数量均逐渐增大。这是因为随着平均分组到达速率的增加，单位时间内到达网络兴趣分组数增多，导致网络中流通分组数增加。对比分析可以看出，4 种路由机制中，网络流通分组数量从小到大依次为最短路径转发机制 SPF、网络状态感知机制 NSAR、随机转发机制 RF、全转发机制 FF。SPF 选择跳数最少的路径进行转发，数据分组需要被转发的次数最少，网络中流通的分组数最少；NSAR 选择网络状态最优的路径进行转发，此路径可能不是跳数最少的，而且 NSAR 还需要往转发概率为 NULL 的端口及随机探测端口转发数据分组，故其网络中流通的分组数比 SPF 大；RF 选择的路径具有随机性，但其选择最短路由的概率还是较 SPF 和 NSAR 要小；FF 需向所有匹配端口转发兴趣分组，故其网络中流通的数据分组数最多。由此看出 NSAR 的网络流通分组数相对较少。

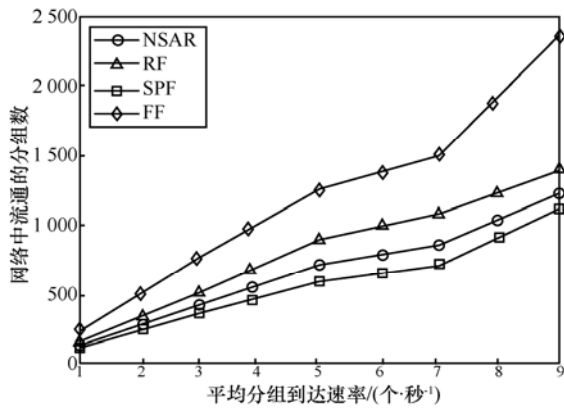


图 5 网络中流通的分组数对比

图 6 是对各路由机制的带宽开销进行对比。从图中可以看出，随着用户数量的增加，4 种路由机制的网络带宽总开销均逐渐增大，而当用户数量达到一定时，带宽开销趋于平稳。这是因为网络的链路带宽是一定的，当用户的总带宽需求达到链路带宽阈值时，带宽开销将趋于最大值。对比分析，4 种路由策略的带宽开销从小到大依次为最短路径转发机制 SPF、网络感知路由机制 NSAR、随机转发机制 RF、全转发机制 FF。SPF 选择跳数最少的路径进行转发，占用的链路资源最少，带宽开销也最小；NSAR 选择网络状态最优的路径进行转发，此路径可能不是跳数最少的，而且 NSAR 还需要往转发概率为 NULL 的端口及随机探测端口转发数据分组，故其所占用的链路资源比 SPF 多，带宽开销比 SPF 大；RF 选择路径具有随机性，但占用的链路还是较 SPF 和 NSAR 要多；FF 向所有匹配端口转发兴趣分组，占用链路资源最多，故其带宽开销最大。由此看出 NSAR 的带宽开销相对较小。

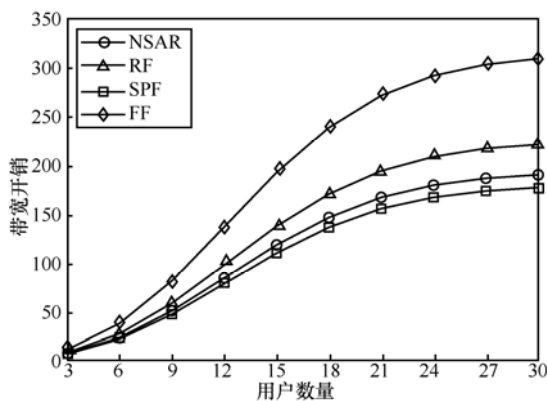


图 6 带宽开销对比

图 7 是分析探测深度对网络流通分组数及内容传输的总带宽开销的影响。实验时，将用户数设为

15，平均分组到达速率为 5 个/秒。从图中可以看出，随着探测深度的增大，网络中流通的分组数及网络带宽开销均增加。这是由于探测深度的增大，每个路由节点需要选择的转发端口数增加，使网络中流通的分组数增加，占用带宽资源也增加，导致网络带宽开销增大。但是探测深度的增加可以使路由机制 NSAR 对网络状态感知越仔细，对端口转发概率的更新越准确，对路由的指导也越精确。所以可以根据路由机制的性能要求动态地调整探测深度以获得符合服务要求的路由机制。

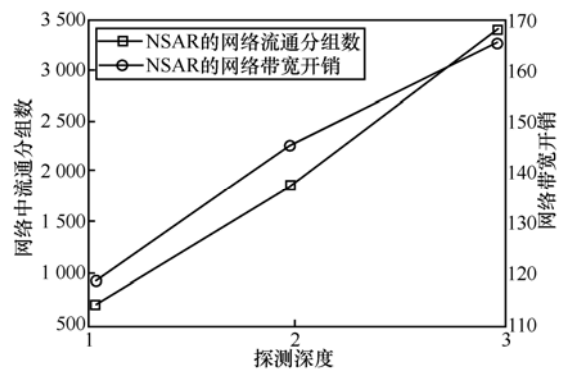


图 7 探测深度对流通分组数和网络带宽开销的影响

4 结束语

本文主要研究了 CCN 的路由机制，提出了一种基于网络状态感知的路由机制 NSAR。NSAR 利用从内容服务节点返回的数据分组收集网络状态信息，并在回传过程中对路由节点上匹配端口的转发概率进行更新。在端口转发概率更新算法上，本文针对有无新增端口的 FIB 情况提出 2 种更新方案，当无新添加的转发端口时，路由节点将按无新添加端口的更新算法对端口转发概率进行更新，当有新添加的转发端口时，节点将按有新添加端口的更新算法对端口转发概率进行更新。当后续的兴趣分组到达路由节点后，节点首先会将兴趣分组往转发概率最大的端口进行转发，然后判断是否有新添加的转发端口，若有，则往该端口转发兴趣分组，随后依据探测深度从剩余端口中随机选择若干个端口转发兴趣分组。仿真结果表明，与其他路由机制相比，NSAR 可以在较低的网络开销下提供最优的转发路径。

由于探测深度对 NSAR 路由机制的性能有较大的影响，接下来的工作是探究探测深度与路由机制 NSAR 性能的相互关系，得到探测深度与路由性能

的最优配比。同时 NSAR 未对网络缓存^[18,19]利用作过多考虑,所以接下的工作是将网络缓存利用加入到路由机制中,并对该路由算法下的网络缓存命中率及内容获取代价作相关研究分析。

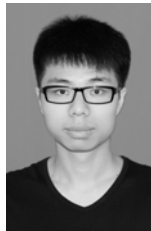
参考文献:

- [1] Cisco. Visual networking index: forecast and methodology, 2010-2015, white paper[EB/OL]. <http://www.cisco.com/go/vni>.
- [2] Google. We knew the Web was big[EB/OL]. <http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html>.
- [3] KOPONEN T, CHAWLA M, CHUN B, et al. A data-oriented (and beyond) network architecture[C]//ACM SIGCOMM. c2007:181-192.
- [4] FP7 PURSUIT project[EB/OL]. <http://www.fp7-pursuit.eu/PursuitWeb/>.
- [5] FP7 PSIRP project[EB/OL]. <http://www.psirp.org/>.
- [6] FP7 SAIL project[EB/OL]. <http://www.sail-project.eu/>.
- [7] FP7 4WARD project[EB/OL]. <http://www.4ward-project.eu/>.
- [8] FP7 COMET project[EB/OL]. <http://www.comet-project.org/>.
- [9] FP7 CONVERGENCE project[EB/OL]. <http://www.ictconvergence.eu/>.
- [10] NSF named data networking project[EB/OL]. <http://www.named-data.net/>.
- [11] Content centric networking project[EB/OL]. <http://www.ccnx.org/>.
- [12] NSF mobility first project[EB/OL]. <http://mobilityfirst.winlab.rutgers.edu/>.
- [13] PARC[EB/OL]. <https://en.wikipedia.org/wiki/PARC>.
- [14] ZHANG L, ESTRIN D, BURKE J, et al. Named data networking project[R]. Tech.Report ndn-0001, PARC, 2010.
- [15] EUM S, NAKAUCHI K, MURATA M, et al. CATT: potential based routing with content caching for ICN[C]//ACM SIGCOMM Workshop on Information-Centric Networking. c2012: 49-54.
- [16] 叶润生, 徐明伟. 命名数据网络中的邻居缓存路由策略[J]. 计算机科学与探索, 2012, 6(7): 593-601.
YE R S, XU M W. Neighbor cache explore routing strategy in named data network[J]. Journal of Frontiers of Computer Science and Technology, 2012, 6(7): 593-601.
- [17] National science foundation network[EB/OL]. https://en.wikipedia.org/wiki/National_Science_Foundation_Network.
- [18] CHAI W K, HE D, PSARAS I, PAVLOU G. Cache “less formore” in information-centric networks[C]//IFIP-TC6 Networking Conference. c2012: 758-770.
- [19] PSARAS I, CHAI W K, PAVLOU G. Probabilistic in-network caching for information-centric networks[C]//ACM Workshop on Information-Centric Networking. c2012:55-60.

作者简介:



蔡岳平 (1980-), 男, 江苏丹阳人, 重庆大学副教授、硕士生导师, 主要研究方向为数据中心网络、光通信网络、未来互联网等。



刘军 (1990-), 男, 江西遂川人, 重庆大学硕士生, 主要研究方向为内容中心网络、软件定义网络等。