

多约束主动路由协议及其性能分析

刘梦娟 洪佩琳 李津生

(中国科学技术大学电子工程与信息科学系 合肥 230027)

摘要 主动网络是一种新型的网络体系结构,能够快速地把新协议和新应用引入到正在运行的网络中。由于主动节点具有较高的计算能力,所以能够提供比传统网络更为有效的服务质量路由方案。该文尝试把服务质量路由引入到主动网络中,结合两者优点提出一种基于主动网络的多约束主动路由(MAR)协议。仿真结果表明 MAR 比目前其它的 QoS 路由机制更简洁、更灵活、更可控。

关键词 主动网络, QoS 路由, 分布式探测, 显式路由

中图分类号: TP393.02

文献标识码: A

文章编号: 1009-5896(2006)06-1135-06

Analyzing the Performance of Multi-constraints Active Routing Protocol in Active Network

Liu Meng-juan Hong Pei-lin Li Jin-sheng

(Dept. of Electronic Engineering and Information Science, USTC, Hefei 230027, China)

Abstract Active Networks (AN) are a novel approach to network architecture in which the switches of the network perform customized computations on the messages flowing through them. AN seems to be particularly useful in the context of QoS support. In this paper, an active routing protocol is proposed which is called Multi-constraints Active Routing(MAR) by introducing QoS routing into AN. Some simulation and analysis prove this protocol is more concise, more controllable, and more flexible than other QoS routing mechanisms at present.

Key words Active Network (AN), QoS routing, Distributed probing, Explicit routing

1 引言

如何在IP网络上为接入业务提供服务质量(Quality of Service, QoS)保证一直是网络研究的热点。目前IETF关于QoS的研究主要包括:综合服务(IntServ/RSVP)模型、区分服务(DiffServ)模型、流量工程(Traffic Engineering)以及与其相关的服务质量路由(QoS Routing, QoSR)等。其中QoS路由是近期QoS研究中一个非常活跃的方向^[1,2]。

所谓QoS路由,又称为多约束路由(Multi-constrained routing),其主要目的是使网络能够根据给定的QoS约束条件和网络资源,动态地为接入业务选择满足其服务质量要求的传输路径,同时又能保证网络资源的有效利用,避免网络拥塞^[3]。QoS约束条件包括端到端延时、延时抖动、有效带宽、分组丢失率等QoS度量参数。最新的研究是把QoS路由机制与MPLS或GMPLS转发机制结合起来,使用QoS路由来计算支持流量工程的标签交换路径(TE-LSP)^[1,2],以实现均衡负载,提高资源利用率的目的。

目前QoSR的研究已经取得了一定成果,例如Wang-Crowcroft提出的带宽-延迟受限的源路由算法^[4]; Yuan-Liu提出的粒度受限的启发式路由算法^[5]; Shin-Chou提出的延迟受限的分布式算法^[6];以及Chen-Nahrstedt提出的基

于探测的分布式路由算法^[7]等。但是纵观这些算法,还存在以下两方面问题:(1)计算复杂度高,在传统网络中难以实现;(2)扩展性差,大部分算法都只是针对QoSR问题中某些特定情况,难于在不改变原型设计的基础上,增加新的约束条件。

主动网络(Active Network, AN)是一种新型网络体系结构^[8]。它通过标准应用编程接口(Application Programming Interface, API)和移动代码(mobile code)技术,使用户能够对网络中的主动节点进行编程,从而使其自定制的新协议和新应用能够快速引入到正在运行的网络中。主动网络的中间节点不仅能获得准确的状态信息,而且具有较高的计算能力,能对流经它的分组进行语义分析、理解、计算、转发等处理,所以主动网络能够提供比传统网络更为灵活的QoSR方案。本文正是针对这一优点进行研究,提出了一种基于主动网络技术的多约束主动路由协议MAR(Multi-constraints Active Routing)。该协议是在分布式探测路由算法^[6]的基础上结合主动网络的优点改进简化而来的。MAR允许用户根据应用要求和网络状况自行定制QoS约束条件和路由参数,使主动分组在主动节点的路由转发由其携带的路由参数和可执行代码在该节点的执行结果来决定。

作为一种主动网络协议, MAR与传统网络协议存在本质区别:传统网络中协议是一组关于报文、语义和时序的静态规约,具有固定的报文格式和交互流程,需要经过国际标

准化组织颁布,在网络节点上部署才能执行(如RIP, OSPF等);而MAR则是一组用户定制的网络通信程序,可根据需要快速地在局部范围内的主动节点上动态下载和执行。

本文内容安排如下:第2节介绍分布式探测路由算法的基本思想;第3节给出多约束QoS路由问题的形式化描述;第4节详细描述MAR协议的寻路过程;第5节对MAR协议和分布式探测路由算法进行原型仿真和性能测试,对仿真结果进行分析;最后总结全文。

2 分布式探测路由算法

分布式探测路由算法^[6]的基本思想是源节点以泛洪(flooding)的方式向其邻居节点发送携带了探测路径状态信息的路由请求;中间节点在收到路由请求后,首先更新请求中所携带的探测路径状态信息,然后判断节点资源是否满足用户要求的多约束条件,如果满足,则继续向其它邻居节点泛洪该路由请求,否则丢弃该请求;目的节点收到路由请求后,根据请求中携带的探测路径信息,选定一条路径作为传输路径,并沿该路径反向发送确认信息,该路径上每个节点通过收到的确认信息获知该节点在所选路径的下一跳节点。

分布式探测路由算法的特点在于:(1)单个节点不需要维护全局网络状态信息,能够有效避免信息交互带来的通信开销与存储开销;(2)约束判断分布到中间节点进行,所以单节点只需维护本地状态信息,可以避免信息交互延迟带来的信息不准确问题;(3)分布式路由算法同时对所有路径进行探测,所以其路由响应时间短,寻路成功率高。但是分布式探测路由算法采用泛洪方式进行路由请求转发,会给网络带来很大的通信开销,而且还可能出现路由环路问题,从而严重影响了算法的实用性和扩展性。

通过以上分析可以看到,分布式探测路由算法的优越性在于分布式的探测思想,而不在于泛洪的路由方式,因此本文把分布式探测、显式路由(explicit route)和主动网络三者相结合,提出一种更简洁、更灵活、更可控的主动网络服务质量路由协议MAR。

为了方便对MAR协议进行描述,本文设计了一个具体的多约束路由问题:路由跳数、端到端延时和传输带宽受限的QoS路由。注意MAR协议中的约束条件和参数可由用户根据应用需要自行定制并不受本文约束条件的限制。

3 多约束 QoS 路由问题的形式化描述

假设网络可用简单无向图 $G(V, E)$ 表示,其中 V 是网络节点的集合, E 是通信链路的集合。对于每条链路 $e = (u, v) \in E$,分别定义3个度量函数:跳数函数 $\text{hop}(e) = 1$,时延函数 $\text{delay}(e): E \rightarrow R^+$ 以及带宽函数 $\text{bandwidth}(e): E \rightarrow R^+$ 。若用 $P = \{P_1, P_2, \dots, P_n\}$ 来表示源节点 s 到目的节点 d 的所有可达路径集合, H 是跳数限制、 D 是端到端时

延限制、 B 是链路有效带宽限制,则多约束QoS路由问题可表述为从集合 P 中选择一条路径 P_i 满足以下条件:

$$\text{hop}(P_i) = \sum_{e \in P_i} \text{hop}(e) \leq H \quad (1)$$

$$\text{delay}(P_i) = \sum_{e \in P_i} \text{delay}(e) \leq D \quad (2)$$

$$\text{bandwidth}(P_i) = \min_{e \in P_i} (\text{bandwidth}(e)) \geq B \quad (3)$$

4 基于主动网络的多约束主动路由(MAR)协议

MAR协议的基本思想如下:源节点首先计算出源、目的节点对之间最大转发跳数为 H 的所有可达路径,然后根据可达路径信息和设置的路由参数创建 k (由用户选择)个探测消息,同时对 k 条路径进行探测,仅当该路径上所有节点和链路资源都满足业务的QoS约束条件时,这些路径的探测消息才能到达目的节点,目的节点按照某种准则(例如延迟最短、转发跳数最小、剩余资源最大等等)对所收到的路径信息进行优化选择,最后确定其中一条作为最优路径并通知源节点,同时进行反向资源预留。可以看到在MAR协议中,路径计算由源节点独立完成,而路径选择则由多个节点协同完成。

4.1 符号定义

Probe($s, d, \text{sid}, \text{hop_count}, \text{QoS}, \text{path_status}, \text{dlist}$): 探测消息,用于路径探测;

Ack($s, d, \text{sid}, \text{hop_count}, \text{QoS}, \text{opt_dlist}$): 肯定应答消息,用于资源的反向预留;

Nack(s, d, sid): 否定应答消息,用于通知源节点路径探测失败;

Free($s, d, \text{sid}, \text{hop_count}, \text{opt_dlist}$): 释放消息,用于通知节点撤销相应的资源预留;

(s, d): 源、目的节点对, s 表示源节点, d 表示目的节点;

sid: 预留标识,由源端产生的本机唯一的序列号,与源、目的节点一起用于标识一个全局唯一的预留;

hop_count: 消息所经过的节点数,用于指出下一跳节点在dlist中的位置;

QoS{ H, D, B }: 用户制定的QoS约束条件;

path_status{hop, delay, bandwidth}: 探测消息所积累的路径状态信息;

dlist: 探测消息中携带的探测路径地址列表。

4.2 MAR 寻路过程

MAR协议的寻路过程具体可分为4个阶段:(1)路径计算;(2)路径探测;(3)最优路径确定;(4)反向资源预留。

第1阶段 路径计算 当 s 产生一个资源预留请求,首先根据其维护的全局拓扑关系(网络邻接矩阵)计算出(s, d)间所有转发跳数不大于 H 的可达路径;然后根据用户设置的路由

参数选择其中转发跳数最小的 k 条作为探测路径(tentative path), 并生成相应的 k 个探测消息 Probe 向网络中发送。

第 2 阶段 路径探测 当 Probe 消息到达中间节点, 自动下载所携带的可执行代码(executed code)并在标准的执行环境(Execution Environments, EE)中运行: 首先更新 Probe 中所携带的路径状态信息 path_status; 然后判断更新后的 path_status 是否满足 QoS 约束条件; 如果满足, 则从 dlist 中取出下一跳节点的地址作为 Probe 消息的转发地址进行转发, 否则丢弃该 Probe 消息, 同时向 s 发送 Nack 消息, 通知源节点该条路径探测失败, 其执行过程的伪代码描述如图 1 所示。

```

while (Probe arrive at node  $i$ ) do
    hop_count = hop_count + 1;
    if ( $i \neq d$ ) then
        delay = delay + native_delay;
        hop = hop + 1;
        bandwidth = native_bandwidth;
        if (delay <  $D$ ) and (bandwidth >  $B$ ) then
            next_destination = dlist [hop_count + 1];
            forward the Probe to next_destination;
        else
            discard the Probe;
            send a Nack( $s, d, sid$ ) to  $s$ ;
        end if
    end if
end while

```

图 1 路径探测过程

第 3 阶段 最优路径确定 当 Probe 消息到达目的节点, 自动下载所携带的可执行代码在 EE 中执行: 首先更新其携带的路径状态信息, 如果更新后的路径状态仍然满足用户规定的约束条件, 则说明该探测路径为适合路径(feasible path), 目的节点将其携带的 dlist 缓存起来, 否则丢弃该 Probe; 在一个定时周期内, 目的节点对所收到的适合路径进行缓存; 当定时到达, 目的节点对缓存的所有适合路径按用户规定的最优准则(optimal rule)进行选择(包括花费、端到端延迟、转发跳数等等); 然后根据选出的最优路径(optimal path)生成相应的 Ack 消息, 沿所选路径进行反向资源预留。

第 4 阶段 反向资源预留 当 Ack 消息到达中间节点, 自动下载其携带的可执行代码在 EE 中执行: 首先判断该节点是否具有足够的资源, 如果具有则在相应端口进行资源预留, 同时初始化对应的资源管理列表, 然后向所选路径的上一跳节点转发该 Ack; 否则, 向源节点发送 Nack, 通知预留失败, 同时向所选路径的下一跳节点发送 Free 消息, 通知它释放掉已经预留的资源并删除相关的资源管理记录。如果下一跳节点不是目的地址则继续转发 Free 消息, 直到撤销该路径的所有预留资源为止, 其执行过程的伪代码描述如图 2 所示。

```

while (Ack arrive at node  $i$ ) do
    if ( $i \neq s$ ) then
        if (node  $i$  has the required resources) then
            reserve resources for connection sid;
            initialize the resource management table;
            pre_hop = dlist [hop_count - 1];
            forward the Ack to pre_hop;
        else
            send a Nack( $s, d, sid$ ) to  $s$ ;
            send a Free to dlist[hop_count + 1];
        end if
    end if
end while

```

图 2 资源预留过程

当 Ack 消息在定时周期内到达源节点, 则寻路成功; 否则, 源节点认为初始探测失败。源节点根据用户制定的路由参数判断是否进行迭代探测(iterative probing), 如果不采用迭代探测, 则直接通知用户预留失败; 否则, 源节点从剩余可达路径中按路径转发跳数重新选择可达路径作为探测路径, 返回第 1 阶段开始迭代探测, 如果所有允许探测的可达路径都不能满足约束条件, 则通知用户寻路失败。当用户希望撤销该预留时, 由源节点发送 Free 消息, 撤销各节点的资源预留。

注意本文暂不考虑 MAR 协议中涉及到的主动代码生存时间以及代码回收方式等问题。

预留成功建立的过程如图 3 所示, 其中(1), (2), (3)是正向路径探测过程, (4), (5), (6)是反向资源预留过程; 图 4 给出了预留建立失败的过程, 它共分为两种情况, 一种是(1), (2), (3*)所示的正向路径探测失败的情况, 由于探测分组到达中间节点 3 时, 更新后的路径状态信息不满足用户规定的 QoS 约束条件, 所以该路径的探测分组没有到达目的节点就被丢弃, 其中(1), (2)是正向路径探测过程, (3*)是肯定应答过程, 它通知源节点该路径不是适合路径; 另一种是(1), (2), (3), (4), (5), (6), (6*), (7)所示的反向资源预留失败的过程, 它与前者不同的是目的节点已经确立了最优路径, 并且已经在节点 3 上成功地进行了资源预留, 但是在节点 2 进行资源预留时, 发现节点 2 的资源已经不能满足用户规定的 QoS 约束条件, 从而认为预留建立失败, 其中(1), (2), (3)是正向路径探测过程, (4), (5)是肯定应答(反向资源预留)过程, (6*)是肯定应答过程, 通知源节点预留建立失败, (6), (7)则是撤销资源预留的过程, 它通知中间节点 3 和目的节点 4 撤销资源预留。

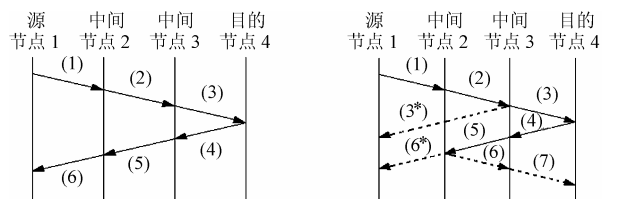


图 3 预留成功建立的过程

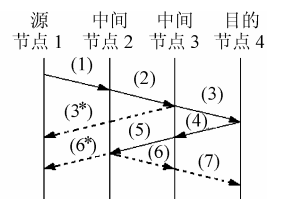


图 4 预留失败的过程

5 仿真实验及性能分析

为了对MAR协议的性能进行评价,本文利用网络仿真软件NS^[9](Network Simulator)分别对分布式探测路由算法(Distributed Routing Algorithm, DRA)和MAR进行了原型仿真和性能测试。仿真中使用的性能指标定义如下:

(1)消息负载(traffic load):指资源预留过程中产生的探测消息数。

(2)预留成功率(success ratio):指规定时间内预留成功建立的概率。

(3)路由响应时间(Route Response Time, RRT):指从发出预留请求到预留成功建立或确认该预留失败所需要的时间。

仿真实验中采用了平均节点度数为3.2的ISP型网络拓扑结构(如图5所示),每条链路均为标准容量为10Mbps的双向对称链路;源、目的节点在所有网络节点中随机选取。为了便于分析,本文把仿真实验分为两组:一组对MAR进行性能测试,讨论路由参数设置对MAR性能的影响;一组把MAR与DRA算法进行比较,讨论各自在性能上的特点。仿真结果如图6~图10所示,其中每个样点值都是1000个独立预留请求的结果。

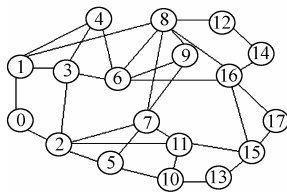


图5 ISP型网络拓扑结构

5.1 路由参数设置对MAR协议性能的影响

因为MAR允许用户根据网络状况和应用要求自定义路由参数,所以有必要测试不同路由参数对MAR协议性能的影响。路由参数设置如下:

(1) MAR1(path = 1, iterate = 0):表示只允许对1条路径进行探测,无迭代;

(2) MAR2(path = 3, iterate = 1):表示最多对3条路径进行探测,初始时沿1条路径探测,1次迭代时沿2条路径探测;

(3) MAR3(path = 6, iterate = 2):表示最多对6条路径进行探测,初始时沿1条路径探测,1次迭代时沿2条路径进行探测,2次迭代时沿3条路径探测。

5.1.1 最大探测路径数对预留成功率的影响 图6给出了3种路由参数下,预留成功的概率。可以看到,MAR3具有最高的预留成功率,MAR2略低于MAR3,MAR1的预留成功率最低。这是因为3种路由参数所设置的最大可探测路径数不同,MAR1只能对1条路径探测,MAR2最多可对3条路径探测,MAR3最多可对6条路径探测,它们的探测范围由小到大,所以探测到适合路径的概率也由小到大。

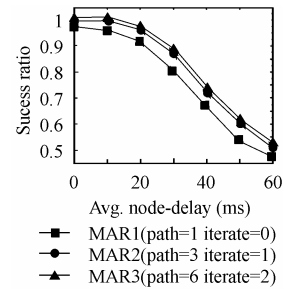


图6 3种路由参数的预留成功率

综上所述可得,预留成功率与用户设定的最大可探测路径数有关,允许探测的路径数越多,预留成功率越高。若对所有可达路径进行探测,则可得到最高的预留成功率。但是在3条成功率曲线中,MAR2的预留成功率比MAR1有较大的改善,而MAR3的预留成功率却与MAR2非常近似。例如平均节点时延为10ms时,MAR2比MAR1提高了2.1%,而MAR3却只比MAR2提高了0.5%。这是因为在MAR中探测路径是按路径转发跳数选择的,而转发跳数越小的可达路径其满足约束条件的可能性也越高,所以虽然可通过增加探测路径数来提高预留成功率,但是其改善程度却随着探测路径数的增加而逐渐减小。

5.1.2 实际探测路径数对平均消息负载的影响 图7给出了3种不同路由参数下,MAR所产生的平均消息负载。可以看到,当节点的平均时延较小时,3种路由参数所产生的平均消息负载相近。这是因为当节点的平均时延较小时,绝大部分预留都可以通过初始探测来建立,例如平均节点时延为10ms时,96.1%的预留都在初始探测阶段建立,只有3.9%的预留会启动迭代探测,所以产生的探测消息数无明显差别。但是随着平均节点时延增加,MAR1产生的消息负载逐渐下降,而MAR2和MAR3产生的平均消息负载却大幅上升,并且MAR3的上升幅度明显大于MAR2。这是因为随着平均节点时延的增加,会有越来越多的探测路径不能满足端到端延迟的约束条件,从而在中间节点就停止探测,所以对于只进行初始探测的MAR1来说,其产生的探测消息数会逐渐减少;而MAR2和MAR3由于设置了迭代探测,当不能通过初始探测建立预留时,会自动转入迭代探测过程,以启动更多的路径探测,所以MAR2和MAR3产生的探测消息数会大幅增加;同时又因为MAR3允许启动2次迭代,其实

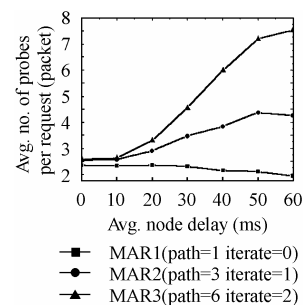


图7 3种路由参数的消息负载

际启动的探测路径数也比 MAR2 多, 所以产生的探测消息数也比 MAR2 多。综合分析可得, 消息负载与预留建立过程中实际启动的探测路径数有关, 实际探测的路径数越多, 产生的消息负载也越多, 而实际探测路径数则与设置的迭代次数和每次探测过程所启动的探测路径数有直接联系。

5.1.3 迭代次数对路由响应时间的影响 图 8 给出了 3 种路由参数下, MAR 的路由响应时间。可以看到, 在相同网络状况和约束条件下, MAR1 的路由响应时间最短, MAR2 次之, MAR3 的路由响应时间最长。这是因为 MAR 的路由响应时间与实际启动的迭代探测次数有关, 迭代次数越多, 其路由响应时间越长。

综合以上分析, MAR3 虽然具有最高的预留成功率, 却是以路由响应时间和消息负载为代价换来的; MAR1 具有最短的路由响应时间和最少的消息负载, 却牺牲了预留成功率。所以 MAR 的性能其实是预留成功率、消息负载、路由响应时间三者的折中, 而折中程度可由用户根据实际情况自行选择, 这也正体现了 MAR 可控、灵活的优点。

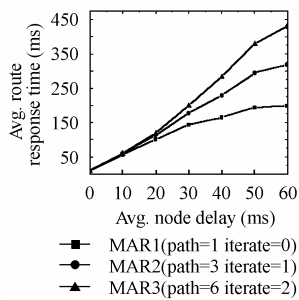


图 8 3 种路由参数的路由响应时间

5.2 两种路由方案的性能比较

第 2 组实验对分布式探测路由算法 DRA 进行了原型仿真, 并把它性能与 MAR 进行比较。仿真中的约束条件仍采用路由跳数、端到端延迟和链路带宽。

图 9 给出了不同链路负载下, 两种路由方案的预留成功率。当网络负载较轻时(负载 $\leq 30\%$), DRA 和 MAR 都能获得很高的预留成功率, $DRA \geq 99.3\%$, $MAR \geq 97.9\%$, 这是因为网络负载轻, 链路带宽资源充足, 因此几乎所有的预留都能成功建立。随着网络负载增加(负载 $40 \sim 70\%$), MAR 和 DRA 仍能获得较高的预留成功率, $DRA \geq 97.2\%$, $MAR \geq 92.9\%$, 这是因为 MAR 采用了迭代探测机制, 当初始探测失败时可启动更多的路径探测, 而 DRA 则是对所有路径

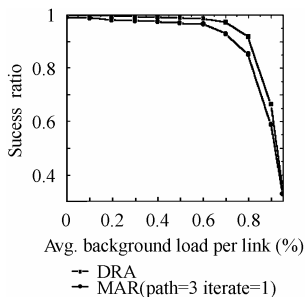


图 9 两种路由方案的预留成功率

进行探测, 所以仍然能维持较高的预留成功率。当负载很重时(负载 $80\% \sim 95\%$), 网络中满足带宽要求的链路急剧减少, 所以 MAR 和 DRA 的成功率都出现了明显下降。

图 10 给出了不同链路负载下, 两种路由方案所产生的消息负载。可以看到, 在轻负载环境下 DRA 产生的探测消息数非常大, 接近 38, 这是因为 DRA 是以一种有限制的泛洪方式对探测消息进行转发, 在负载较轻的情况下, 几乎所有的链路都能满足约束条件, 所以节点会向几乎所有的邻居节点转发探测消息, 同时 DRA 还可能出现同一预留请求的探测消息在相同链路上来回传送的情况。随着网络负载增加, DRA 产生的消息负载逐渐减少, 这是因为随着负载增加, 链路的有效带宽资源减少, 从而不能满足预留请求的带宽约束条件, 所以节点停止向其邻居节点转发该预留请求的探测消息, 所以产生的探测消息数减少。MAR 产生的消息负载远小于 DRA, 且随着链路负载的增加而小幅上升, 直到达到峰值(负载 90%), 再逐渐减小。这是因为链路负载较轻时, 绝大多数预留都能通过初次探测建立, 所以产生的探测消息数少。随着网络负载增加, 通过初次探测建立的预留数逐渐减少, 更多的预留会启动迭代探测, 所以探测的路径数增加, 产生的探测消息数也增加。当网络非常拥塞时($> 90\%$), MAR 产生的消息负载开始下降, 原因与 DRA 类似。

综合以上分析可知, DRA 和 MAR 都能实现为单一业务流建立满足其服务质量要求的传输路径的功能。对于具有相同服务质量要求的用户请求, 虽然 DRA 成功建立传输路径的概率略高于 MAR, 但是其分布式探测带来的通信开销也远远高于 MAR; 而 MAR 则能通过较少的通信开销, 获得与 DRA 相近的路径建立成功率, 所以 MAR 的性能应该优于 DRA。

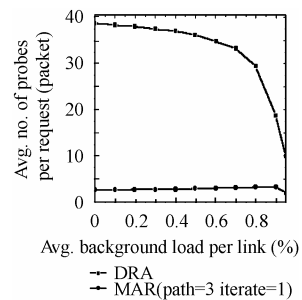


图 10 两种路由方案的消息负载

6 结束语

如何为用户的单一业务流提供确定性的端到端服务质量保证是 QoS 研究的一个重要内容, 而多约束路由正是解决这一问题的一种有效途径。本文提出了一个基于主动网络技术的多约束主动路由协议 MAR, 通过运行 MAR 协议能够为多媒体实时业务流建立一条具有可靠服务质量保证的传输路径。与传统服务质量路由相比, MAR 具有下列优点: (1)

探测路径由源端确定, 每个探测分组都按确定的地址进行转发, 从而不存在分布式路由探测中通信开销过大、路由环路和路由终止等问题; (2)每个节点只需维护简单的全局拓扑关系和本地状态信息, 从而解决了全局链路状态信息交换和维护带来的通信开销和存储开销过大的问题; (3)通过把约束条件判断分布到各中间节点执行, 从而减小了源端计算的复杂度; (4)探测过程中, 探测路径的状态信息保存在主动分组的可执行代码中, 从而减小了各节点的存储开销; (5)用户可根据业务需要自定义约束条件和路由参数, 从而提高协议的可控性和灵活性。

参 考 文 献

- [1] Apostolopoulos G, Williams D, Kamat S. QoS routing mechanisms and OSPF extensions. RFC 2676, August 1999.
 - [2] Path Computation Element WG. <http://www.ietf.org/html.charters/pce-charter.html>.
 - [3] Awduche D, Chiu A, Elwalid A. Overview and principles of Internet traffic engineering. RFC 3272, May 2002.
 - [4] Wang Z, Crowcroft J. QoS routing for supporting resource reservation. *IEEE Journal on Selected Areas in Communications*, 1996, 14(7): 1228–1234.
 - [5] Yuan X, Liu Xingming. Heuristic algorithms for multi-constrained quality of service routing. INFOCOM 2001, Las Vegas, USA, 2001, vol.2: 844–853.
 - [6] Shin K G, Chou Chih-Che, Kweon Seok-Kyu. Distributed route selection for establishing real-time channels. *IEEE Trans. on Parallel Distributed Systems*, 2000, 11(3): 318–335.
 - [7] Chen Shigang, Nahrstedt K. Distributed quality-of-service routing in high-speed network based on selective probing. *Local Computer Networks, LCN '98*, 1998: 80–89.
 - [8] David L T, Smith J M, Sincoskie W D, David J W. A survey of active network research. *IEEE Communications Magazine*, 1997, 35(1): 80–86.
 - [9] McCanne S, Floyd S. Network Simulator. <http://www.isi.edu/nsnam/ns>.
- 刘梦娟: 女, 1979年生, 硕士生, 研究领域为主动网络和服务质量路由。
洪佩琳: 女, 1961年生, 教授, 博士生导师, 主要研究领域为信息通信网络和网络安全。
李津生: 男, 1937年生, 教授, 博士生导师, 主要研究领域为信息通信网络。