

## 面向 SON 的 QoS 重路由机制

王珑飞, 王兴伟, 黄敏

(东北大学 信息科学与工程学院, 辽宁 沈阳 110819)

**摘要:** 随着计算机网络的发展, 自组织网络因其具有灵活性、自适应性、顽健性等优点而受到广泛的关注。在 OSPFv3 协议的基础上, 添加了 QoS state 交互报文, 用于建立邻居信息表, 进而完成链路上服务质量 QoS 的信息存储和重路由触发的判断, 并引进了节点的休眠、苏醒等生物特性行为。为保证网络状态变化下应用的 QoS, 采用了混合式重路由机制, 针对不同网络失效状况作出相应的处理。仿真实验表明, 该路由机制具有良好的性能。

**关键词:** 自组织网络; OSPFv3; QoS; 重路由

中图分类号: TP393

文献标识码: A

文章编号: 1000-436X(2012)Z2-0280-10

## QoS rerouting mechanism for SON

WANG Long-fei, WANG Xing-wei, HUANG Min

(College of Information Science and Engineering, Northeastern University, Shenyang 110819, China)

**Abstract:** Along with the development of computer network, SON (self-organizing network) has attracted wide attention with its flexibility, adaptability, robustness. On the basis of OSPFv3, the interactive message of QoS state was defined to establish the information table of neighbors, store the link QoS information, judge the triggering of rerouting, and then the biological behaviors, such as hibernation and awakening, were introduced to the routing nodes. A hybrid rerouting mechanism was adopted to make the corresponding treatment under various network failure situations, so that the application QoS support could be provided when the network state was changing. Simulation results show that the proposed mechanism has better performance.

**Key words:** self-organizing network; OSPFv3; QoS; rerouting

### 1 引言

随着计算机网络规模的日益膨胀和网络应用需求的不断增长, 异构性、动态性、不稳定性对于大型网络服务质量 QoS (quality of service) 需求提出

了更高的挑战<sup>[1]</sup>, 新的网络处理机制需要考虑网络状态发生变化时如何快速有效地保证网络性能。为满足更灵活、更多样、更高服务质量 QoS 的网络服务需求, 需要使网络中的参与者脱离中心控制, 并能够实现个体间直接交流, 进而对局部间的环境变

收稿日期: 2012-10-23

**基金项目:** 国家杰出青年科学基金资助项目 (61225012); 国家自然科学基金资助项目 (61070162, 71071028, 70931001); 高等学校博士学科点专项科研基金资助项目 (20100042110025, 20110042110024); 工信部物联网发展专项资金资助项目; 中央高校基本科研业务费专项基金资助项目 (N110204003)

**Foundation Items:** The National Science Foundation for Distinguished Young Scholars of China (61225012); The National Natural Science Foundation of China (61070162, 71071028, 70931001); The Specialized Research Fund for The Doctoral Program of Higher Education (20100042110025, 20110042110024); The Specialized Development Fund for The Internet of Things from The Ministry of Industry and Information Technology of China; The Fundamental Research Funds for The Central Universities (N110204003)

化作出连续反映。在这样的背景下,受自然界中普遍存在的生物自组织现象启发,提出了自组织网络的概念<sup>[2,3]</sup>。基于生物灵感的自组织网络技术已经研究了十余载<sup>[4,5]</sup>,其希望网络中无需全局的协调通信,个体组织成的群体通过本地的局部环境变化和局部环境交互,动态地调整全局属性和行为,使网络具有可伸缩性、自适应性以及简单自治性来保障网络性能的稳定。当网络因某个或某些节点出现故障而导致相关链路异常时,自组织网络能通过某种方式发现其他可行路径替代当前路径,从而保证整个网络的性能不受影响<sup>[6,7]</sup>。

当前互联网所部署的 OSPFv3 路由协议中,重路由机制通常由路由器检测到链路失效或节点失效后,将相关链路状态信息的更新扩散到其他所有路由器,重新计算生成路由表。在检测、扩散和重计算过程中存在的处理延迟<sup>[8]</sup>,将会导致花去几秒甚至几分钟的时间去恢复链路或节点失效后的数据分组转发,这种数据分组转发中的不连续性对于网络中的业务需求来说是一个弊端。现有研究提出了很多解决方案,文献[9]提出了一个接口特定转发(interface specific forwarding)的概念,它在决定为达到网络负载均衡的业务重路由方面更具灵活性。文献[10]提出了一种混合式端到端重路由策略,称之为共享顽健式重路由(shared robust rerouting),它可看作是一些修复策略的结合,包括路径的多样性、端到端的重路由以及全局重路由。文献[11]提出了一种失效不敏感(failure insensitive)的局部重路由方法<sup>[12]</sup>,当链路失效发生时,邻近节点会触发全局更新,初始局部重路由,其他节点会根据数据分组的流入接口推断链路失效,提前计算特定接口失效下的转发路由表,进而快速完成重路由。文献[13]提出了一种基于负载均衡的重路由算法,有效地降低阻塞率并提高带宽利用率。光路级的重路由策略在文献[14]中提出,它包含基于最少资源利用和负载均衡 2 个启发式算法,并指出根据不同的情况分别触发。文献[15]则认为通过及时的全局重路由和随机的重新尝试也会取得较好的性能。文献[16]提出了一种对于链路状态协议的局部恢复算法,该算法规定当路径上的链路权重降到 0 时,重新计算路由并更新路由表,这减少了对链路失效的响应时间,但增加了节点的负载。文献[17]和文献[18]提出了利用原路径上信息计算新路径的动态算法,其优点是时间复杂度是线性的,但每次链路失效时,都实时

运行是不现实的。

本文提出了一种面向 SON 的 QoS 重路由机制,对 OSPFv3 协议加以改动,以备份路由为基础采用了混合式重路由机制,针对不同网络失效状况做出相应的处理。仿真实验表明,该路由机制具有良好的性能。

## 2 重路由机制设计

### 2.1 网络模型

将网络拓扑抽象成一个连通图  $G=(V,E)$ 。其中,  $V$  是具有转发和处理能力的节点集合,  $E$  是边的集合。对每条边考虑的参数主要有:带宽、延迟、延迟抖动、分组丢失率。本文自组织网络考虑的节点行为有:新生,即产生并初始化新节点;感知,即节点感知相连边和相邻节点信息;复制,即将本节点的信息复制到邻近节点(当被复制节点失效时,邻近节点可代替该节点完成相应工作);休眠,即节点因长时间无工作负载而进入休眠状态;苏醒,即休眠节点恢复到正常工作状态;假死,即因链路或节点故障而导致节点接口失效状态;死亡,即节点因硬件或软件故障而永久性地无法工作。

本文设计的重路由机制包括重路由触发感知、路由信息更新、网络失效类型判断、局部或完全重路由、休眠及苏醒触发和新生及死亡触发。路由器之间通过交换链路参数感知周围环境建立邻居关系,并按规定时间通过与邻居节点交互更新路由信息。当网络中出现链路或节点失效时,针对不同的网络失效类型,源节点将进行局部或完全重路由来处理。重路由有时会因网络资源稀缺而无法找到到达目的地址的路径,故本文设计休眠和苏醒报文用于解决该失效情况。网络中的新生节点通过与周围节点交互,建立邻接关系后成为新成员,而当节点死亡后,其邻居节点将取消与其的邻居关系,并更新转发路由表和备份路由表。

### 2.2 重路由触发感知

自组织网络中每个节点要与其邻近的节点进行交互,通过感知其周围环境和局部交互来获得全局属性。在自组织网络中不仅要考虑网络拓扑的动态性变化,还要考虑链路上 QoS 参数的动态变化,因此本文借鉴了 OSPFv3 中的 HELLO 报文并设计了 QoSstate 和 QoSstateack 报文用于重路由触发的探测情况。

在 OSPFv3 协议<sup>[19]</sup>中,每个路由器利用 HELLO

数据分组通告其路由器 ID，动态地发现相邻的路由器节点并与其邻近的路由器建立和保持邻居关系。因此，可以利用 HELLO 报文来维护网络中节点的邻接情况。HELLO 报文构成邻居间通信的基础，但为了在路由协议的基础上满足 QoS 需求，除了知道每条链路的连接状态外，还需要获得每条链路的 QoS 信息，包括带宽、延迟、延迟抖动及分组丢失率。

为此，本文在 OSPFv3 协议中添加 QoSstate 和 QoSstateack 报文来获取计算链路 QoS 信息并定义其协议报文头类型值分别为 6 和 7。两者之间的交互过程如下（假设路由器 A 和 B 是邻居节点）。

1) A 向 B 发送 QoSstate 报文，通告本地链路带宽和分组丢失率并记录时间 *send\_Statetime*，其中，*send\_Statetime* 指 QoSstate 报文的发送时间。

2) 当 B 收到了 QoSstate 报文之后需要记录 *recv\_Statetime*，读取并保存其中的带宽和分组丢失率，还要记录 *send\_Acktime*，同时也需要计算  $p\_time = send\_Acktime - recv\_Statetime$ ，读取对应接口上的带宽和丢失率并填写至 QoSstateack 报文中的相关字段，发送 QoSstateack 报文给 A。其中，*recv\_Statetime* 指收到 QoSstate 报文的时间，*send\_Acktime* 指发送 QoSstateack 报文的时间，*p\_time* 指处理时间。

3) A 收到 QoSstateack 后，从中读取 *p\_time*，需要记录当前的 *recv\_Acktime*。计算  $delaycurrent = (recv\_Acktime - send\_Statetime - p\_time) / 2$ ，计算  $jitter = delaycurrent - delaylast$ ，其中，*recv\_Acktime* 指收到 QoSstateack 报文的时间，*delaycurrent* 指 A 与 B 之间的延迟，*jitter* 指延迟抖动，*delaylast* 是从数据库中直接读出来的。将延迟与抖动连同报文中的带宽和丢失率保存至链路状态数据库中。

若路由器不支持 QoSstate 报文，则由于报文类型未知而忽略该报文，不进行任何处理。由于网络资源的动态性，为了尽可能地准确描述链路上的资源情况，从而满足不同业务的 QoS 请求，QoSstate 报文的发送频率应该是动态变化的。

QoSstate 报文格式如图 1 所示，其报文中每个字段定义如下。

1) 组序号：标识 QoSstate 报文组，具有相同组序号的报文需要由路由器统一处理，不同组序号间的报文分别处理。通过处理同一组中的报文获得 QoS 信息的平均值。

0		15		23		31	
组序号		报文序号		最大报文序号			
报文发送时间							
可用带宽				分组丢失率			
报文间隔时间		长度		检验和			

图 1 QoSstate 报文

2) 报文序号：标识同一组中的报文序号，从 1 计数起。

3) 最大报文序号：标识本组报文中的最大报文序号。

4) 报文发送时间：报文发送时的路由器时间，以格林威治时间 1970 年 1 月 1 日 0 时 0 分 0 秒计为基准，以 ms 为单位。

5) 可用带宽：报文发送接口可用带宽，由发送路由器根据接口记录值填写。

6) 分组丢失率：报文发送接口的分组丢失率，由发送路由器检测发送队列后填写。

7) 报文时间间隔：本组下一报文的发送时间，接收路由器根据该字段值设置定时器。若定时器超时，则认为报文丢失。

8) 长度：报文长度。

9) 检验和：报文检验和，对整个 QoSstate 报文校验。

QoSstateack 报文格式如图 2 所示。

0		15		23		31	
组序号		报文序号		最大报文序号			
报文处理时间							
可用带宽				分组丢失率			
长度		检验和					

图 2 QoSstateack 报文

QoSstateack 报文中除报文处理时间字段外其余字段定义与 QoSstate 字段相同。而报文处理时间字段的填写参看本节 QoSstate 与 QoSstateack 交互过程的步骤 2)。

节点发送固定频率的 HELLO 报文，当发现新的邻居时，发送 QoSstate 报文来完成 QoS 资源的初始化。由于不同类型的应用服务有各自的 QoS 需求，当某节点发现链路上的资源无法满足某类型服务时就会向对应的邻居节点发送 QoSstate 报文通告当前 QoS 资源情况并通知其进行相应节点行为，以便及时地发现网络上的资源情况，从而更好地进行

路由及数据分组转发。

### 2.3 重路由信息更新

在获取链路上的 QoS 信息后, 需将其扩散至网络中的其他节点上。为满足指定应用的 QoS 需求, 需将标准路由协议中的报文进行扩展, 这里将主要考虑 QoS 信息更新。标准的 OSPFv3 路由协议通过以下操作进行链路状态更新, 进而获得最新的自治域拓扑。

1) 当路由器初始化时, 网络拓扑结构发生变化(例如增减路由器或链路状态发生变化等)或链路资源发生较大变化影响了特定业务的 QoS 需求时, 路由器会产生 LSA(link state advertisement), 该报文里包含路由器上所有相连链路, 即所有端口的状态信息。

2) 路由器将其 LSA 报文发送给所有与其相邻的路由器, 相邻路由器根据其接收到的链路状态信息更新自己的数据库, 并将该链路状态信息转发给与其相邻的路由器, 直至稳定。

为简化协议, 本文只对路由器 LSA 进行扩展, 为保证与标准 OSPFv3 协议的兼容性, 在扩展报文中填入标识字段, 表明是否包含扩展的 QoS 信息。其中, 路由器 LSA 扩展用于发布链路的 QoS 信息。

支持扩展协议的路由器接收路由器 LSA 报文后通过查看附加标志位来确定该报文是否携带链路 QoS 信息。若不携带链路 QoS 信息, 则按照标准路由器 LSA 报文处理; 否则, 在获取标准链路信息后读取附加检验和字段和附加长度字段, 对附加 QoS 信息进行检验。若检验通过, 则顺序读出每条链路的带宽、延迟、延迟抖动和分组丢失率等 QoS 信息。这样, 路由器在接收路由器 LSA 报文后可根据报文内容更新链路状态数据库。若路由器不支持扩展协议, 则在收到扩展 LSA 报文后忽略附加标志字段, 按照标准 LSA 报文处理, 只读取检验和字段与长度字段, 并根据字段内容检验附加检验和字段内容。由于此时读取到的内容与标准路由器 LSA 报文相同, 标准路由器可正常处理扩展 LSA 报文并更新标准链路状态数据库。

更新流程如下。

1) 路由器 A 向其邻居路由器发送 QoSstate 报文, 通告链路的带宽和分组丢失率并标识报文发送时间。

2) 邻居路由器收到一组 QoSstate 报文后, 发

送 QoSstateack 报文通告自身计算链路的带宽、分组丢失率和该接收报文的处理时间。

3) A 收到邻居路由器返回的 QoSstateack 报文后, 计算延迟和延迟抖动并连同所接收的带宽、分组丢失率存储到邻居信息表中。

4) 若 A 的链路状态数据库中存储信息发生变化或探测到链路资源稀缺无法满足任一业务需求时, 则发送路由器 LSA 报文, 向其他路由器通告其链路及 QoS 信息。

5) 当路由器 LSA 报文收敛后, 所有路由器将链路状态数据库中存储的信息映射为网络拓扑结构。

6) 使用 OSPFv3 协议本身的单播路由算法或其他路由算法为每种业务计算 QoS 路由, 并使用标准 OSPFv3 路由协议计算标准路由, 更新 QoS 单播路由表。

### 2.4 网络失效类型判断

网络失效类型可以分为链路失效和节点失效 2 种情况。链路失效指连接 2 个节点的链路间发生物理连接上的失效, 导致两节点间通信异常; 而节点失效指节点由于硬件或软件上的问题导致与其相连的邻居节点通信异常。节点失效可以看作链路失效的特殊情况, 即多条链路失效, 但两者对应的重路由处理方法却有所不同。链路失效可通过局部重路由算法<sup>[20,21]</sup>找到从自身节点出发到达由该失效接口所连接节点的局部路径, 用于替代失效链路, 数据分组通过局部路径后再沿原剩余路径转发即可; 而当发生节点失效时, 采用局部重路由算法找到的局部路径转发会导致数据分组丢失不可达, 故需要采用完全重路由算法<sup>[22]</sup>重新计算出一条从自身节点出发到达目的地址且满足对应业务 QoS 需求的完整路径, 对路径上的节点进行沿途路由表更新后即可保障分组转发正常。

因此, 需要对 2 种不同网络失效情况进行区分判断。然而对于单一节点而言, 2 种失效情况下的失效表象是相同的: 皆为在一段连续时间内无法收到邻居发送的 HELLO 报文, 现有机制不能对此作出区分, 故本文中设计了假死判断报文, 用于判断网络失效类型。

由于标准 OSPFv3 协议中, 当节点连续 4 个周期内未收到 HELLO 报文, 则会认为节点死亡, 为与此相区别, 这里将接口失效状态称为假死状态。为了减少网络失效对分组转发的影响, 当连续 2 个周期内未收到 HELLO 报文时, 则会触发判断机制。

假死判断请求格式如图 3 所示。

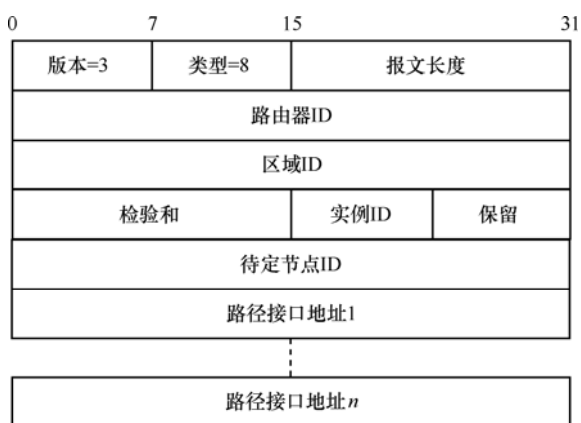


图 3 假死判断请求报文

各字段说明如下。

1) 类型值设置为 8，用于标明该报文为假死判断请求报文。

2) 待定节点 ID，指不可达下一跳的节点 ID。

3) 路径节点地址 1-n，指从当前节点到达待定节点的路径，用于假死判断请求报文的沿途发送。为保证该报文路径的可达性，由全局重路由算法完成计算，由于只需保证路径可达性，QoS 约束可以不予考虑。若算法未找到对应路径则说明节点失效，无需发送假死判断请求报文。

当路径最后一个节点收到假死判断请求报文后，需发送假死应答报文，其格式如图 4 所示。

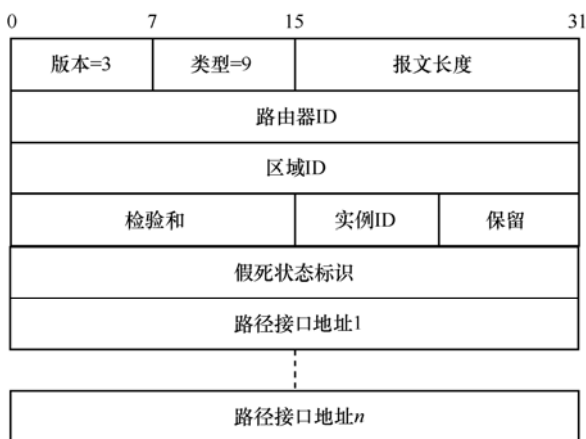


图 4 假死应答报文

各字段说明如下。

1) 类型值设置为 9，用于标明该报文为假死应答报文。

2) 假死状态标识若为 1，则表明为节点失效，若其为 0，则表明接口失效。

3) 路径节点地址 1-n：指从待定节点到达发起节点的路径，根据假死判断请求报文进行装填，用于沿原路径返回更新链路信息。

### 2.5 局部重路由触发

当网络中发生链路失效时，则会触发局部重路由。这里将采用节点复制行为进行处理，即根据节点中所保存的失效接口所对应的备份路由由表更新转发路由表。

以图 5 为例：当图中节点 2 发现转发路由表中到达节点 3 的链路失效或链路(2-3)QoS 资源无法满足分组转发业务的要求时，则要向其邻居节点(1,5)求助，通过邻居节点完成分组转发。这里采用文献[21]中局部重路由算法进行预计算，当网络节点完成汇聚后计算备份路由表，将节点间的备份关系存入备份路由表，当链路失效发生时，查找备份路由表并更新转发路由表即可。需要注意的是，应考虑备份链路状态数据库的开销问题，采用预计算的方法完成重路由会用到多个备份路由表，这不仅会造成节点空间上的开销，也会给路由表的查找、更新以及维护带来一定麻烦，故应存储尽可能少的备份路由表项。当备份链路数据库中的节点发生失效后，应及时将该节点从中删除以避免产生不必要的开销。

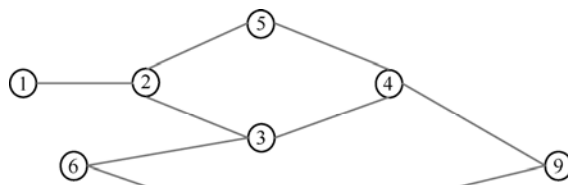


图 5 局部重路由触发

### 2.6 完全重路由触发

当网络中发生节点失效或者转发路由表和备份路由表中的下一跳都处于不满足状态时，则会触发完全重路由，将无法满足其转发需求的数据分组交由其他节点完成转发。本文称该报文为迁移报文，如图 6 所示。迁移报文中添加修改后的各字段说明如下。

1) 类型值设置为 10，用于标明该报文为路径更新报文。

2) 负载目的节点地址连同 IP 头部中 ToS(type of service)字段用于共同标识所要修改的转发路由表项。

3) 路径节点地址 1-n：节点地址序列，共同标识该新路径。



图 6 迁移报文

路径更新的动作序列如下。

1) 当节点检测到网络中发生节点失效等多链路失效状况，或节点中的转发路由表和备份路由表项中的下一跳都无法满足数据分组指定的应用需求时，该节点将以自身为源节点使用完全重路由算法动态实时地计算一条到达负载目的地址且满足指定 QoS 应用需求的新路径，并按图 6 中格式封装成迁移报文。

2) 根据计算出的迁移路径更新本地转发路由表中的路由表表项，并以此向下一跳节点转发该迁移报文。

3) 节点在收到迁移报文后，根据 IP 头部中的 ToS 字段和负载目的地址确定所要修改的转发路由表项，并将下一路径节点作为该表项中的下一跳添加到其中，将本身节点地址从报文中删除，更新迁移报文，将修改后的迁移报文发送到路径上的下一节点，该节点完成迁移操作。

4) 路径上每个节点重复 3) 中的更新修改操作，当迁移报文中路径上所有节点都完成后，迁移行为结束。

### 2.7 休眠及苏醒触发

由于网络中存在资源较为稀缺的情况，此时重路由算法将无法找到到达目的地址且满足 QoS 约束的路径。故本文设计了休眠和苏醒报文，通过两者的交互用于完全重路由无法进行时的失效处理。休眠报文如图 7 所示。

新增及修改字段说明。

1) 类型值设定为 11，表明是休眠报文。

2) 目的地址：128bit，IPv6 地址描述对应的目的地址与 IP 头部中的 ToS 字段一起标识负载。

苏醒报文格式如图 8 所示，新增及修改字段说明如下：

1) 类型值设定为 12，表明是苏醒报文；

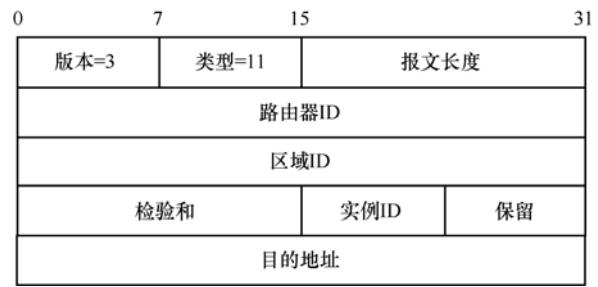


图 7 休眠报文

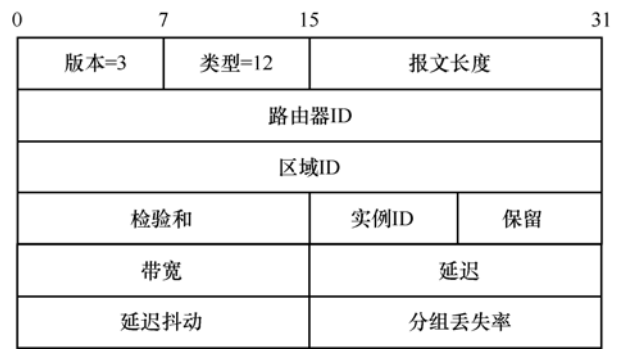


图 8 苏醒报文

2) 带宽、延迟、延迟抖动、分组丢失率分别表示苏醒后的节点与相应的邻居节点的 QoS 资源。

休眠及苏醒的行为序列有以下 5 条。

1) 当节点的转发路由表和备份路由表的下一跳转发都无法满足指定应用类型的转发需求（处于过载状态）且找不到满足 QoS 需求的新路径节点时，该节点要进入休眠状态。

2) 处于休眠状态的节点查询转发路由表中下一跳为过载状态的路由表项，将与触发休眠负载所对应的应用类型和目的地址封装成休眠报文并向其所有邻居节点发送，通知其邻居节点，自身将进入休眠状态，禁止邻居节点向其发送到指定目的地址的分组转发，但可保持其他数据分组转发的正常。

3) 邻居节点收到休眠报文后，仍保留与该节点的邻接关系，若休眠报文中指定的负载转发由邻居节点完成，则根据备份路由表修改转发路由表，否则跳过，从而邻居节点的负载不会增加。（标识转发路由表中的表项为不可达即可）

4) 当休眠节点由于负载应用结束或迁移成功导致相关链路资源得到释放后，节点就要进行苏醒行为，向邻居节点发送苏醒报文，通知相连链路上恢复后的 QoS 资源情况（含有苏醒后的链路资源）。

5) 收到苏醒报文后，恢复转发路由表中不可达

表项即可。

## 2.8 新生及死亡触发

对于网络拓扑中新生的路由节点可以按照 OSPFv3 中新节点加入机制处理。新生节点通过发送 HELLO 报文，与其周围的路由节点进行交互，建立邻接关系后，成为网络拓扑中的新成员。

新生节点在与其邻居节点建立邻接关系后，将进行 LSA 的收集扩散。当获得网络中其他所有节点的链路状态描述报文后（达成完全邻接状态），按单播路由算法计算到达其他节点的路径，映射成转发路由表，并根据由此建立的汇聚树按局部重路由算法中提及的算法计算出相应的备份路由表。其他节点在获得新生节点的相关链路状态描述报文后，按单播路由算法计算并更新相关的转发路由表项和备份路由表项。

如果某路由节点的邻接路由节点在路由失效间隔（router dead interval）过后仍未收到该邻居路由器发送的 HELLO 报文，则认为该邻居路由器已经失效，到达该路由的路径也同样被认定为失效，即可确认其已经死亡，取消与其建立的邻接关系。

由于节点死亡会对网络中的数据分组转发造成一定的影响，故需要考虑节点死亡时所触发的重路由。与死亡节点邻接的节点发现节点死亡后，发送对应链路状态描述更新的报文，所有节点在达到完全状态后计算新路径并更新转发路由表并备份路由表。在此计算期间，与死亡节点邻接的节点数据分组转发由备份路由表完成。

## 2.9 重路由机制描述

重路由机制（S 机制）流程如图 9 所示。重路由机制流程描述如下。

1) 当所有节点达到完全邻接状态后，计算出到达其他节点的最短路径并映射成转发路由表。

2) 根据节点中存储的路径信息，调用局部重路由算法建立备份路由表。

3) 若链路 QoS 资源不足，则跳至 6)。

4) 若连续 2 个 HELLO 周期内未收到 HELLO 报文，则发送假死判定请求报文，否则跳至 9)。

5) 若为节点失效则跳至 7)。

6) 查询备份路由表更新转发路由表，若备份路由表不可用，则转 7)，否则跳至 9)。

7) 调用全局重路由算法计算满足 QoS 业务需求的路径，并装填迁移报文，沿途发送。

8) 若无满足需求的路径，则向邻居节点发送休眠报文，并保持休眠状态直到相关链路资源恢复后，跳至 9)。

9) 结束。

## 3 性能评价

本文采用 Linux 下 GNU C 扩展的 Quagga 软件，分别参考 CERNET2、NSFNET、UNICOM 网络拓扑结构进行环境搭建，为了测试本文中所设计的面向 SON 的 QoS 重路由机制的可行性、实用性以及机制在运行时节点的自组织行为表现情况，拓扑中使用 C++ 语言进行了仿真，仿真中随机生成一些失效节点或链路进行测试，并对其性能（主要包括建立重路由路径的报文开销和时间开销）进行分析评价。

本文中所设计的重路由机制（简称为机制 S）由 2 部分组成，即局部重路由和完全重路由部分，两者分别针对链路失效和节点失效 2 种情况。由于当前广泛采用的汇聚重路由机制（简称为机制 F）未对此加以区分，故此将 2 种机制进行比较。

无论哪种失效情况，汇聚重路由机制的应对措施皆为节点等待邻居发现并确定失效异常，生成 LSAupdate 报文，将其洪泛至其他全部节点，其中，报文内容为失效链路的描述信息（若是链路失效则报文包含一条失效链路信息；若是节点失效则报文包含多条失效链路的描述信息）。在本文机制中，若是链路失效则待节点发现确定后仅根据备份路由表进行本地更新即可；若是节点失效则采用完全重路由算法进行计算，将可行路径沿路分发即可。下面为 2 种机制在不同拓扑下报文开销和时间开销的比较情况。

### 3.1 建立路径的报文开销

建立路径的报文开销是指节点发现网络失效后用于建立新路径，恢复数据转发所花费的字节数，不同机制在不同拓扑下的报文开销比较如图 10 和图 11 所示。

由于机制 F 建立重路由路径时，需要对失效链路的 LSA 进行更新并扩散至全局网络中的所有其他节点，网络中链路越多，相应的报文开销越大。

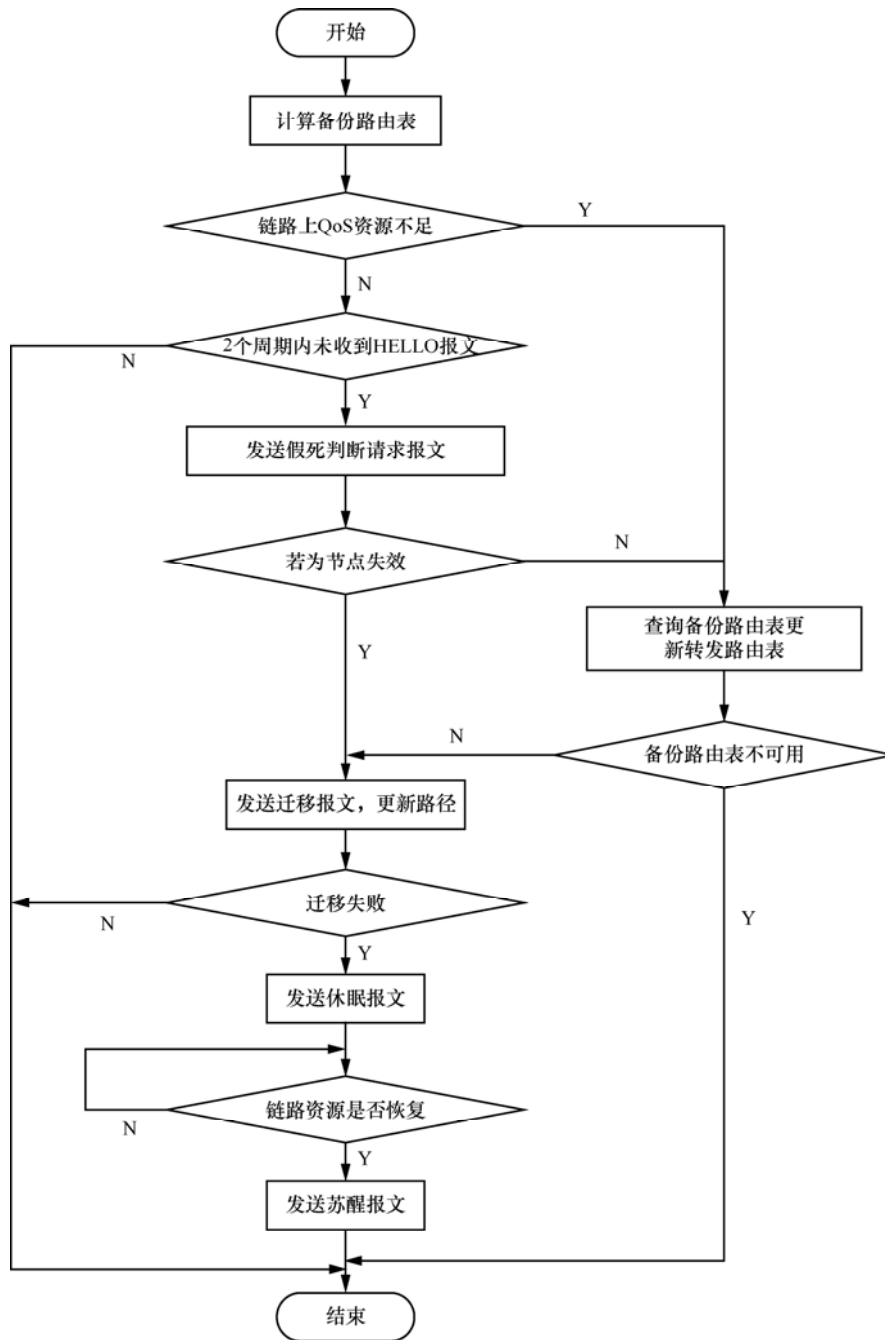


图 9 重路由机制流程

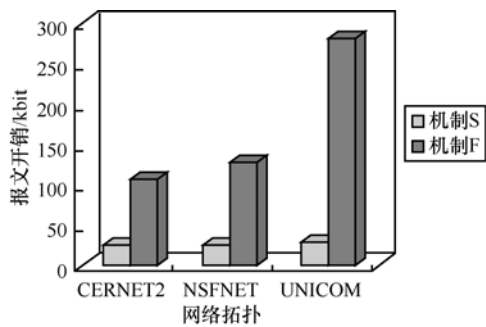


图 10 链路失效状态下报文开销比较

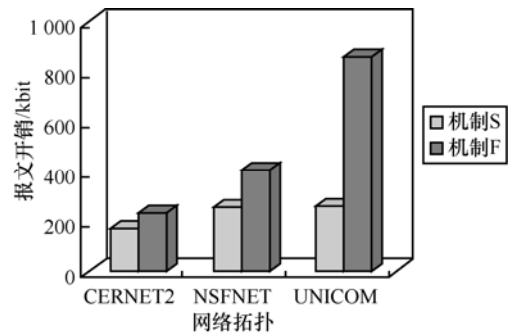


图 11 节点失效状态下报文开销比较

当发生节点失效时, 由于涉及到与失效节点相关联的多条链路更新, 故报文开销又与节点的度数相关, 报文开销将会随之增加。而机制 S 无论在何种失效情况下, 都将发送假死报文进行失效类型的判断, 假死报文的发送仅发生在局部链路上, 故开销较小, 尽管节点失效时涉及到多个邻居节点的判断和路径更新, 但比起机制 F 仍开销小很多, 从图 10 和图 11 中可以看出: 机制 S 随网络中节点链路的增加, 报文开销呈缓慢增长, 而机制 F 则增长较为剧烈。

### 3.2 建立路径的时间开销

建立路径的时间开销是指从节点发现网络失效后建立重路由路径开始, 到完成数据恢复转发所花费的时间, 不同机制在不同拓扑下建立路径的时间开销比较如图 12 和图 13 所示。

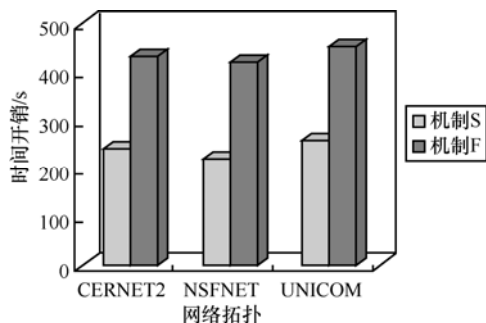


图 12 链路失效状态下时间开销比较

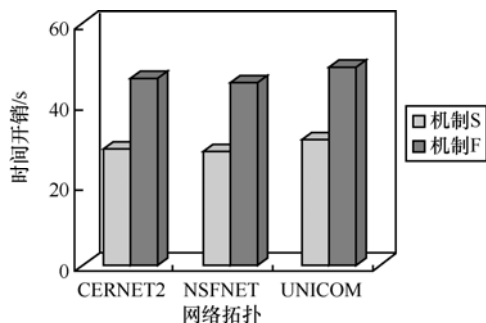


图 13 节点失效状态下时间开销比较

机制 S 的时间由于采用了假死报文的判断执行, 故仅需  $20+S$  的时间就能完成路径的重新建立, 而机制 F 由于采用 HELLO 交互故需  $40+S$  的时间才能完成; 而 2 种机制的路径建立时间都会随着网络中节点数的增多而增加; 由于节点失效会导致多条链路失效, 使得节点失效时路径建立的时间略长于链路失效情况。

## 4 结束语

本文设计实现了一种面向自组织网络的 QoS

重路由机制。在 OSPFv3 协议的基础上, 添加了 QoSstate 报文, 用于建立邻居信息表、搜集链路 QoS 信息及重路由触发条件的判断。局部重路由通过节点的复制行为来实现, 完全重路由则通过迁移报文实现。若因网络资源稀缺而重路由失败时, 则由添加的节点休眠苏醒机制处理该类失效情况。最后, 将该机制与当前广泛使用的机制在建立路径报文开销和时间开销上进行比较, 结果表明, 该机制实用性强、部署方便, 具有良好的性能。

### 参考文献:

- [1] FRIKHA A, LAHOUD S. Performance evaluation of pre-computation algorithms for inter-domain QoS routing[A]. Proceedings of the 2011 18th International Conference on Telecommunications[C]. Ayia Napa, Cyprus, 2011. 327-332.
- [2] DIXIT S, SARMA A. Advances in self-organizing networks[J]. IEEE Communications Magazine, 2005, 43(7):76-77.
- [3] PREHOFER C, BETTSTETTER C. Self-organization in communication networks: principles and design paradigms[J]. IEEE Communications Magazine, 2005, 43(7):78-84.
- [4] DRESSLER F, AKAN O B. Bio-inspired networking: from theory to practice[J]. Communications Magazine IEEE, 2010, 48(11):176-183.
- [5] TENAZINHA N, VINGA S. A survey on methods for modeling and analyzing integrated biological networks[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2011, 8(4):943-958.
- [6] ALDERSON D, WILLINGER W. A contrasting look at self-organization in the Internet and next-generation communication networks[J]. IEEE Communications Magazine, 2005, 43(7):94-100.
- [7] QU Z T, HE P. Self-organization theory and surveillance of network anomalous behaviors[A]. Proceedings of the 2010 2nd International Conference on Advanced Computer Control[C]. Shenyang, China, 2010. 174-178.
- [8] LEE S S W, TSENG P K, LI K Y, *et al.* Interface specific fast failure rerouting for load balanced IP networks[A]. Proceedings of the 2011 IEEE Symposium on Computers and Communications[C]. Kerkyra, Greece, 2011. 316-319.
- [9] BASHLLARI A, FUNDO A, NACE D, *et al.* A note on a hybrid rerouting scheme[A]. Proceedings of the 2011 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops[C]. Budapest, Hungary, 2011. 1-5.
- [10] LI Y, WANG J P, XU Y. Adaptive intentional connection rerouting for traffic grooming in WDM mesh networks with biological attractor selection[A]. Proceedings of the 2010 16th Asia-Pacific Communications[C]. Auckland, New Zealand, 2011. 83-88.
- [11] SANGHWAN S N, NELAKUDITI S, SANGHWAN L, *et al.* Failure insensitive routing for ensuring service availability[J]. Lecture Notes

- in Computer Science, 2003. (2707):153-154.
- [12] BANERJEA A. Fault recovery for guaranteed performance communications connections[J]. IEEE/ACM Transactions on Networking, 1999, 11(5):653-668.
- [13] WANG S W. Light path-level active rerouting algorithms in all-optical WDM networks with alternate routing and traffic grooming[A]. Proceedings of the International Conference on Information Networking (ICOIN)[C]. Bali, Indonesia, 2012. 42-46.
- [14] XIE C, XU F, GHANI N, *et al.* Load-balancing for advance reservation connection rerouting[J]. IEEE Communications Letters, 2010, 14(6): 578-580.
- [15] LAKSHMAN K M. Restorable dynamic quality of service routing[J]. IEEE Communications Magazine, 2002, 40(6):72-81.
- [16] NARVAEZ P, SIU K Y, TZENG H Y. Local restoration algorithm for link-state routing protocol[A]. Proceedings of the 1999 IEEE International Conference on Computer Communications and Networks[C]. Boston, USA, 1999. 352-357.
- [17] FRIGIONI D, MARCHETTI-SPACCAMELA A, NANNI U. Fully dynamic shortest paths in digraphs with arbitrary arc weights[J]. Journal of Algorithms, 2003, 49(1):86-113.
- [18] NARDELLI, PROIETTI, WIDMAYER. Swapping a failing edge of a single source shortest path tree is good and fast[J]. Algorithmica, 2003, 35(1):57-74.
- [19] FERGUSON D, MOY J. OSPF for IPv6[S]. 1999.
- [20] LIU Y, NARASIMHA REDDY A L. A fast rerouting scheme for OSPF/IS-IS networks[A]. Proceedings of the 13th International Conference on Computer Communications and Networks[C]. Chicago, USA, 2004. 47-52.
- [21] ITO H, IWAMA K, OKABE Y, *et al.* Single backup table schemes for shortest-path routing[J]. Theoretical Computer Science, 2005, 333(3): 347-353.
- [22] KORKMAZ T, KRUNZ M. Multi-constrained optimal path selection[A]. Proceedings of the INFOCOM 2001[C]. Anchorage, USA, 2001. 834-843.

#### 作者简介:



王珑飞 (1988-), 男, 吉林延吉人, 东北大学硕士生, 主要研究方向为绿色互联网。



王兴伟 (1968-), 男, 辽宁盖州人, 博士, 东北大学教授、博士生导师, 主要研究方向为未来互联网、云计算、网络安全与信息安全。



黄敏 (1968-), 女, 福建长乐人, 博士, 东北大学教授、博士生导师, 主要研究方向为优化方法及其应用。