基于免疫原理的多拓扑路由生成算法

陈铎龙[1,2] 孟相如[1] 梁 霄[1] 袁荣坤[1]
(空军工程大学信息与导航学院 西安 710077)[1] (93968部队 乌鲁木齐 830075)[2]

摘 要 针对现有多拓扑路由生成算法存在存储资源利用不合理、多故障恢复能力无法有效适应网络环境等问题,提出了一种结合生物免疫原理的多拓扑路由生成算法。该算法以适应网络环境中常见的多故障为目标,根据免疫原理中抗原与抗体决定基无须完全吻合而只需关键部位相匹配的机制,将子拓扑生成看作免疫原理中的抗体产生,增强了多故障情况下的网络抗毁性,同时引入人工免疫算法来解决算法中的寻优问题。实验结果表明,该方法提高了网络在多故障环境下的抗毁性。

关键词 网络抗毁性,多故障恢复,免疫机制,多拓扑路由
中图法分类号 TP393 **文献标识码** A

Multi-topology Routing Generation Algorithm Based on Immune Mechanism

CHEN Duo-long[1,2] MENG Xiang-ru[1] LIANG Xiao[1] YUAN Rong-kun[1]
(School of Information and Navigation, Air Force Engineering University, Xi'an 710077, China)[1]
(93968 Army, Urumqi 830075, China)[2]

Abstract An multi-topology routing generation algorithm based on immune mechanism was proposed due to the fact that the algorithm in existence causes unreasonable usage of storage resource and maladjustment of multi-faults recovery. Aiming at adapting for fault of high probability, according to the principle of the key part matching instead of corresponding completely between epitope and paratope in the immune mechanism, sub-topology generation is treated as antibody generation in immune system in the proposed algorithm, thus the network invulnerability under complexion of multi-faults is enhanced. And the question of optimization is solved by using artificial immune algorithm. The experimental result shows the network invulnerability under multi-faults is improved.

Keywords Network invulnerability, Multi-faults recovery, Immune mechanism, Multi-topology routing

1 引言

近年来,随着信息技术的发展,网络结构越来越复杂,规模越来越大,传统的路由收敛方式已经无法保证网络通信业务的高效运行,因此如何快速地完成路由收敛正成为网络可生存性研究的热点。

常见的域内路由协议如OSPF的收敛过程是,当检测发现故障后,路由器向全网络洪泛故障信息,其余路由器接收到故障信息后,重新计算到达各个节点的路径,并根据计算的路径更新路由表。OSPF协议可以保证在网络故障发生后都可以找到一条新的路径,但收敛速度达到了几秒甚至几十秒,在收敛期间会导致分组的丢失,不适合承载实时性较强的业务。因此快速路由恢复的方式被一些学者提出。

文献[1]直观地针对收敛时间过长问题,缩短故障检测周期,加快Hello包发送频率,提高了故障检测速度,减少了收敛时间,但频繁地发送Hello包会导致网络的振荡与带宽资源的浪费;多路径方法[2,3]是在故障发生前准备多条备份路径,当故障发生时,将流量切换至备份路径,同时进行正常的路由收敛过程,完成后再将流量切换回完成收敛的路径,该方法适合持续时间较长的严重故障,但无法处理频发的瞬时故障;基于IP备用路径[4,5]是当故障发生后,不发送故障信息,而是使用备份路由直至故障自行恢复,这类方法的优点是思路简单,适合处理频发瞬时故障,但存在可能无法找到备份路径的缺陷;文献[6]中提出故障非敏感路由(failure insensitive routing, FIR),其核心思想是当故障发生后,不等待故障通告,直接通过备份转发表将流量返回相应节点,节点接收到流量后推断出相应的下游链路发生故障,随即将流量切换至不经过故障链路的备份路径,FIR的优点是适合频发的单链路故障,但会造成远距离通信时下游链路利用率低下,不利于负载均衡。

文献[7-9]中提出了多拓扑路由(multi-topology routing, MTR)的方法,在IP层中抽象出若干子拓扑,当故障发生后,将流量切换至备份拓扑,隔离故障链路,保证了通信业务的运行,同时可以处理多故障情况,由于子拓扑是在全网生成,还

到稿日期:2013-03-15 返修日期:2013-07-23 本文受国家自然科学基金(61201209),全军军事学研究生课题(2011JY002-524)资助。
陈铎龙(1988-),男,硕士生,主要研究方向为网络抗毁性,E-mail:cdl120220@126.com;**孟相如**(1963-),男,教授,博士生导师,主要研究方向为宽带通信网络技术;**梁 霄**(1988-),女,硕士生,主要研究方向为网络安全态势感知;**袁荣坤**(1986-),男,硕士生,主要研究方向为网络可生存性。

可以利用权值调整进行负载均衡。MTR 虽然具有以上优势，但是仍存在占用存储资源多、多故障恢复效果不佳等问题，因此如何使 MTR 生成的子层能够更适应网络环境需求的研究正逐渐受到学者的重视。

本文第 2 节着重介绍 MTR 的原理；第 3 节提出一种结合免疫原理的多拓扑路由子层生成算法；第 4 节针对多拓扑进行计算机仿真与性能分析；最后给出了结论和后续改进方向。

2 多拓扑路由

多拓扑路由是一种主动式的路由恢复方案，它在网络路由初始化阶段，就在 IP 层中抽象出若干子层，其中每个子层都能够对应地保护网络中的某些结构（链路或节点），且单个子层可以同时保护多条链路（或节点），故 MTR 能够有效处理多故障的发生；由于子层是在全网络范围内生成的，因此可以通过对子层链路权值的设置实现链路的负载均衡。

为清楚地描述多拓扑路由的工作原理，先对拓扑结构做出规定：网络拓扑在数学上可以描述成一个图 $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E})$，其中 $\boldsymbol{V}=\{v_1,v_2,\cdots,v_n\}$ 为图中的顶点集合，每一个顶点对应一个拓扑中的节点，$\boldsymbol{E}=\{e_1,e_2,\cdots,e_m\}\subseteq\boldsymbol{V}\times\boldsymbol{V}$ 表示图中边的集合，每条边对应拓扑结构中的链路。$n=|\boldsymbol{V}|$ 表示节点数量，$m=|\boldsymbol{E}|$ 表示边数量。

在 MTR 中，由原网络拓扑中链路的子集与 $\boldsymbol{G}$ 中所有节点构成的拓扑结构称为子层 $\boldsymbol{L}$。每一个子层 $\boldsymbol{L}$ 都从 $\boldsymbol{G}$ 中删除一部分链路，剩余的链路与节点构成了子层的拓扑结构，被删除的链路称为被保护链路。为了使子层能够有效地保障被保护链路的通信业务，每一个子层 $\boldsymbol{L}$ 都应当满足下述条件：

(1)$\boldsymbol{L}$ 中每个节点对之间应当至少有一条连通路径；

(2)$\boldsymbol{G}$ 中的所有链路都应当至少被一个子层 $\boldsymbol{L}$ 所保护；

(3)所有经过被保护链路的路径在相应保护子层上都至少存在一条备份路径避开被保护链路。

因为每个子层中都包含了 $\boldsymbol{G}$ 中所有节点，各子层最终将所有链路均纳入保护范围，且总存在一条路径将被保护链路替换，所以 MTR 可以在链路故障情况下通过切换子层来保护链路承载的流量，同时由于一个子层删除了多条原始拓扑中的链路，MTR 还具有处理多链路故障的能力。

文献[7]提出了 Minimum、Rich、Sparse 3 种子层生成算法，3 种算法分别以生成最小数目子层、保证子层可用性（连通性）、最大限度处理多故障为目标，生成子层拓扑。

Minimum 算法的目标是生成最小数目的子层，其核心思想是每个子层通过尽量移除子层中割边以外的链路来实现生成的子层的数目最小化这一目标。同时，为了避免子层承担业务不均衡，Minimum 算法还对生成子层中的链路进行均衡，在完成初步子层生成后，将含链路最多的子层中一部分链路转移到含链路最少的子层中去，以防止一些子层中包含链路很少而一些子层中包含的链路很多的情况。

Rich 算法的思路大体与 Minimum 算法相同，不同之处在于，Rich 算法在生成子层之前就规定了子层的个数 k，而每个子层中移除的链路数为 $n-n/k$，即每个子层所保护的链路数相同，同时通过增加子层的数目，使得每个子层的连通性得到了提高，缩短了恢复时延，提升了子层可用性。

多个链路故障分别发生在不同子层时，一般的 MTR 就无法有效进行保护，针对这种情况学者提出了 Sparse 算法。Sparse 算法是通过生成链路稀疏的子层，在同一子层内尽可能保护多的链路，能够更好地适应多故障情况下流量的保护。Sparse 算法首先固定了子层的数目，并对原始拓扑中的链路进行分组，处在不同分组中的链路将被同一个子层所保护，同时使子层拓扑结构尽可能地稀疏，最终形成树状结构，这样将最大限度地在多链路故障情况下形成有效保护。

当网络发生故障后，MTR 通过将业务流量切换至备份子层的方式来实现对流量的保护，图 1 是 MTR 故障恢复流程的示意图，当网络正常运行时，网络使用原始拓扑即 0 号拓扑进行正常的路由计算，转发路径为 6-4-5-3。网络发现链路 4-5 损坏后，立即将报文返回至报文源节点，源节点接收到未送达报文后，得知链路 4-5 发生故障，随即使用保护 4-5 链路的 1 号拓扑，1 号拓扑中预先计算的转发路径为 6-7-8-3，将报文通过该路径转发，至此，业务流量恢复。

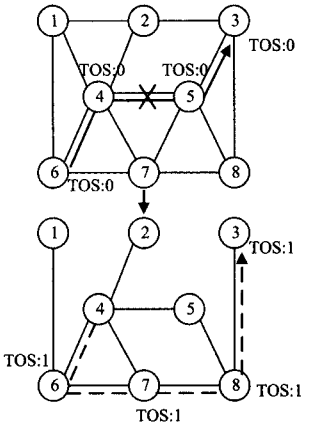


图 1 MTR 故障恢复示意图

与传统路由收敛方式及其它快速路由故障恢复方法相比，MTR 显然克服了诸如收敛速度缓慢、频繁收敛产生的网络振荡、无法找到恢复路径、链路负载不均衡、只能处理单一故障等缺陷。

MTR 虽然克服了上述缺陷，但也存在一些问题，比如存储子层信息会占用大量存储资源，同时传统子层生成方式未必能够适应网络中的各种故障。

因此，本文接下来将结合免疫原理，通过引入人工智能算法完成子层的生成，使得生成的子层适应性更强。

3 结合免疫原理的子层生成算法

在免疫系统中，B 细胞在识别抗原后可以进行克隆扩增分化为浆细胞，产生一种能够与抗原相结合并破坏这些抗原的蛋白质分子，这种蛋白质分子称为抗体。抗体与抗原都具有特异的决定基，二者决定基在匹配过程中，匹配程度越好，抗体与抗原的亲合力就越大，但不像锁与钥匙需要完全一致，而只需要大致匹配互补就可以，决定基的结合导致 B 细胞进入活化状态开始克隆，这就是克隆选择，只有匹配程度高的抗体才能进行扩增，减少了系统资源的浪费。

将人工免疫系统应用于网络故障恢复时，MTR 中每一个子层相当于一个抗体，被移除的链路类似于抗体中的决定基，发生故障的拓扑结构相当于抗原，故障链路可以看作抗原决定基。在子层生成过程中，所有可能形成的子层与故障拓扑进行匹配，匹配程度最好的子层就是需要被克隆选择的子层，

这样网络中只保留能够最大程度匹配故障的子层，节省了存储资源，并使得生成子层对网络故障的抵抗能力更强。

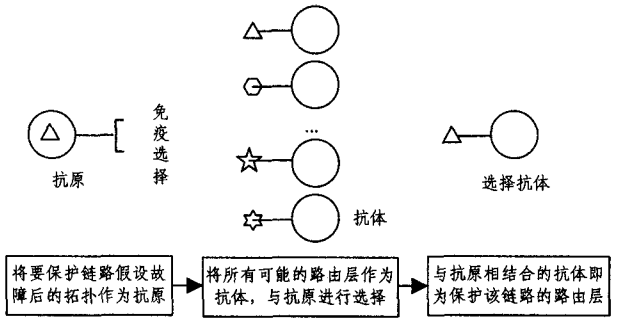


图 2　结合免疫机制的多拓扑生成示意图

3.1　克隆选择原理

生物免疫系统[10]理论中的一个重要学说是克隆选择，即通过细胞连续分裂传代形成群体，在分裂过程当中，细胞的基因会产生变异，从而产生出新的抗体，而免疫系统依据细胞产生的抗体与抗原的亲和度的高低来选择保留亲和度高的抗体，去除亲和度低的抗体。

这一概念已经被广泛运用到计算机领域，其学习、记忆等特性也引起了人工免疫系统研究者的兴趣，以下将介绍克隆选择算法的基本原理。

研究以 $\boldsymbol{A}=\{a_1,a_2,\cdots,a_m\}$ 为变量的优化问题 (P)：$\max\{f(\boldsymbol{A}):\boldsymbol{A}\in I\}$，其中 $a_1,a_2,\cdots,a_m$ 为抗体 $\boldsymbol{A}$ 的遗传基因，共 m 段；$\boldsymbol{I}$ 称为抗体空间，包含系统内所有抗体，f 为 $\boldsymbol{I}$ 上正实值函数，称为抗体-抗原亲合度函数。

抗体空间可以描述为：

$$\boldsymbol{I}^n=\{\boldsymbol{A}:\boldsymbol{A}=[\boldsymbol{A}_1,\boldsymbol{A}_2,\cdots,\boldsymbol{A}_n]\} \tag{1}$$

正整数 n 为抗体种群规模，$\boldsymbol{A}=[\boldsymbol{A}_1,\boldsymbol{A}_2,\cdots,\boldsymbol{A}_n]$ 中每个元素代表一个抗体。定义问题 (P) 的全局最优解集为：

$$\boldsymbol{B}^*\equiv\{\boldsymbol{A}\in\boldsymbol{I}:F(\boldsymbol{A})=\max(f(\boldsymbol{A}'):\boldsymbol{A}'\in\boldsymbol{I})\} \tag{2}$$

抗体种群 $\boldsymbol{A}(k)$ 在克隆选择算子的作用下，其演化流程可以表示为：

$$\boldsymbol{A}(k)\xrightarrow{T_c^C}\boldsymbol{Y}(k)\xrightarrow{T_g^C}\boldsymbol{Z}(k)\tilde{\cup}\boldsymbol{A}(k)\xrightarrow{T_s^C}\boldsymbol{A}(k+1) \tag{3}$$

式中，k 表示迭代次数，T_c^C、T_g^C、T_s^C 分别表示克隆操作、免疫基因操作、克隆选择操作。

对于 $\boldsymbol{X}=\{x_1,x_2,\cdots,x_n\}$ 与 $\boldsymbol{Y}=\{y_1,y_2,\cdots,y_n\}$，$\tilde{\cup}$ 运算定义为：

$$\boldsymbol{X}\tilde{\cup}\boldsymbol{Y}\equiv\bigcup_{i=1}^{n}\{x_i\cup y_i\} \tag{4}$$

克隆操作定义为：

$$\boldsymbol{Y}_i(k)=T_c^C(\boldsymbol{A}_i(k))=\boldsymbol{I}_i\times\boldsymbol{A}_i(k),\ i=1,2,\cdots,n \tag{5}$$

式中，$\boldsymbol{I}_i$ 为 $q_i(k)$ 维元素全为 1 的行向量，$q_i(k)$ 定义如下：

$$q_i(k)=\mathrm{INT}\left[n_c\cdot\frac{f(\boldsymbol{A}_i(k))}{\sum_{j=1}^{n}f(\boldsymbol{A}_j(k))}\cdot\Theta_i\right] \tag{6}$$

该式反映了克隆选择的种群规模不仅受个体竞争（抗体-抗原亲合度）影响，也受抗体间亲合度 Θ_i 影响，二者相互制约，式中具体参数的设定参见文献[11]。克隆操作实现了抗体个体规模的扩张，为后续产生新抗体的变异功能的实现提供了条件。

免疫基因操作定义为：

$$\boldsymbol{Z}(k)=T_g^C(\boldsymbol{Y}(k)) \tag{7}$$

免疫基因操作可使 $Y(k)$ 中第 i 个抗体的第 m 段基因依概率 p_m^i 发生变异，产生新的抗体，属于一种进化过程。其中：

$$p_g(\boldsymbol{Y}_{ij}(k)\to\boldsymbol{Z}_{ij}(k))=(p_m^i)^{d(\boldsymbol{Z}_{ij}(k),\boldsymbol{Y}_{ij}(k))}(1-p_m^i)^{l-d(\boldsymbol{Z}_{ij}(k),\boldsymbol{Y}_{ij}(k))} \tag{8}$$

式中，$d(\boldsymbol{A},\boldsymbol{B})$ 表示 $\boldsymbol{A}$ 与 $\boldsymbol{B}$ 的汉明距离。

克隆选择操作定义为：

$$\boldsymbol{A}(k+1)=\max\{f(\boldsymbol{Z}(k)\tilde{\cup}\boldsymbol{A}(K))\} \tag{9}$$

该操作通过局部最优选择，实现了种群的压缩，完成了对新抗体空间的构建，使得新的抗体空间更加适用于当前免疫环境。

3.2　子层生成算法

MTR 子层生成的过程可以用免疫原理来描述，抗体初始化、变异与克隆选择的过程类似于子层的生成过程。

在生成子层之初，需要有初始化的子层模型作为智能算法的初始子层，这一生成初始子层的过程类似于克隆选择算法中抗体初始化的过程；网络发生故障的情形，类似于抗原对生物系统进行入侵的情形；子层与故障情形进行匹配，从而调整自身子层结构的过程，类似于生物免疫系统中抗体根据抗原入侵状态而对自身结构进行进化的过程，故将网络故障情形对应于克隆选择算法中的抗原，而将生成的子层结构对应为克隆选择算法中的抗体。由上可见，多拓扑路由子层生成的过程与免疫原理克隆选择算法相类似，可以从中抽象出数学模型。

这里把子层集合 $\boldsymbol{L}=\{\boldsymbol{L}_1,\boldsymbol{L}_2,\cdots,\boldsymbol{L}_k\}$ 视为抗体空间，其中每个子层就是抗体空间中的抗体，即：

$$\boldsymbol{L}_i=[a_{1,2},\cdots,a_{i,j},\cdots] \tag{10}$$

而子层中包含的所有链路，分别相当于每个抗体中的基因，式中 $a_{i,j}$ 为布尔型变量，为 1 表示链路 i,j 在该层中存在，为 0 表示不存在。

对于原始拓扑，我们首先统计每条链路所承载的流量 $S_{i,j}$，抗原集合为 $\boldsymbol{R}=[\boldsymbol{R}_1,\boldsymbol{R}_2,\cdots,\boldsymbol{R}_l]$，定义初始抗原为：

$$\boldsymbol{R}_j=[r_{1,2},\cdots,r_{i,j},\cdots] \tag{11}$$

式中，$\boldsymbol{R}$ 为所有双故障情形的集合，$l=C_m^2$，$r_{i,j}$ 同样为布尔型变量，当值为 1 表示链路正常，为 0 则链路故障，其中每条链路以概率 $p_{i,j}=\frac{S_{i,j}}{\sum S_{i,j}}$ 发生故障。

为方便描述抗体-抗原亲合力函数，定义如下算子：

定义 1　对于两个布尔型变量 $\boldsymbol{X}=[x_1,x_2,\cdots,x_n]$ 与 $\boldsymbol{Y}=[y_1,y_2,\cdots,y_n]$，combin 算子运算为：

$$Z=\boldsymbol{X}\mathrm{combin}\boldsymbol{Y} \tag{12}$$

该算式表示，初始时 $Z=0$，依次比较 $\boldsymbol{X},\boldsymbol{Y}$ 中各元素，若 $x_i=0$ 且 $y_i=0$，则 $Z=Z+1$，其余情况下 $Z=Z$，比较下一组元素直至 n 个元素均比较完毕，返回 Z 值。

定义 2　抗体-抗原亲合力函数为：

$$f(\boldsymbol{R},\boldsymbol{L}_i)=\sum_{j=1}^{l}(P_kP_h\prod_{n=1}^{m\backslash k,h}(1-P_n))\boldsymbol{R}_j\mathrm{combin}\boldsymbol{L}_i \tag{13}$$

式中，k,h 为发生故障的链路编号，P_n 为链路故障概率，等同于 $p_{i,j}$。

由抗原-抗体亲合力函数可以看出，如果子层能够保护抗原中出现故障的链路，则抗体-抗原亲合度就会提高，反之亲合度不会提高。

根据上述定义，下面提出基于免疫原理的 MTR 多拓扑生成算法，步骤如下：

Step1　产生初始抗原 **R**；

Step2　随机产生子层集合 $\mathbf{L}=\{\mathbf{L}_1,\mathbf{L}_2,\cdots,\mathbf{L}_k\}$，其中 k 为预先规定的子层数目，第 i 个子层必须包含全网络中故障概率由大至小排在第 i 位的链路，同时生成网络所有链路集合 **E**，检验每个子层是否连通，若联通则执行 Step3，若不连通，则重新产生子层直至联通；

Step3　$\mathbf{E}=\Phi$ 或若迭代次数 k 满足要求，跳至 Step7；否则对每个抗体执行克隆操作：$\mathbf{Y}_i(k)=T_c^c(\mathbf{L}_i(k))$，将 **E** 初始化；

Step4　对每个扩展种群执行免疫基因操作：$\mathbf{Z}_i(k)=T_g^c(\mathbf{Y}_i(k))$，检查变异个体是否连通，若不连通删除该个体；

Step5　执行克隆选择操作：$\mathbf{L}_i(k+1)=T_s^c(\mathbf{Z}_i(k)\cup\mathbf{L}_i(k))$，若 **L** 中子层各不相同，保存 **L**，否则重新执行 Step5；

Step6　检查 **E**，若 **E** 中包含有 **L** 所保护的链路，则从 **E** 中删除该链路后跳至 Step3；

Step7　若 **E** 中仍有链路，则将该链路从保护链路最少且不保护该链路的子层中移除，直至 $\mathbf{E}=\Phi$，产生最终的子层空间 **L**，算法结束。

为了使最终生成的子层能够最大限度地处理各种故障，算法的迭代次数应当大于网络中的链路数，且当迭代次数足够大时，会大大减少克隆选择算法中错误发生的次数。

以上给出了基于免疫原理的多拓扑子层生成算法，网络系统在运行之前即使用该算法生成子层，当网络发生故障时，网络就可以使用相应的子层使流量避开故障链路，对网络进行故障恢复。接下来将从网络负载流量、鲁棒性等方面对该算法进行性能分析。

4　性能分析

实验拓扑、业务量矩阵与物理链路长度采用 14 个节点、21 条双向链路构成的 US 网络拓扑[12]，如图 3 所示。

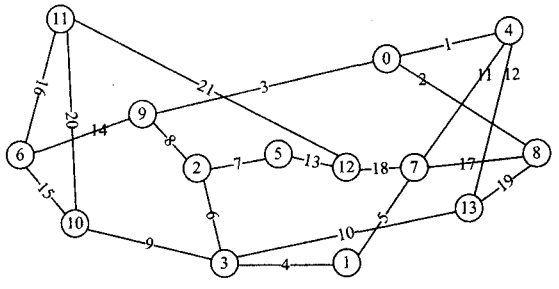


图 3　US 网络物理拓扑图

全网络承载总业务量为 13253.364Gbit/s，链路容量为 5Tbit/s，拓扑所承担的负载大小会影响到网络业务的正常运行链路的可靠性，备份拓扑网络的鲁棒性决定了网络保护业务能力的大小，而存储资源利用合理性也是评价生成拓扑是否适应网络环境的标准。本节就从备份拓扑负载流量、子层故障恢复能力和存储资源利用合理性 3 个方面对该方法进行性能分析。

4.1　备份拓扑负载流量

实验中统计在双链路故障下，各备份拓扑链路所承载的最大业务量，即链路业务流量峰值，单位为 Gbit/s。

实验分别统计无故障情况下，使用 Minimum 算法和使用免疫原理算法时 US 拓扑的链路承载业务流量情况。免疫原理算法中，使用 3 个备份拓扑，并迭代 50 次。

图 4 为上述 3 种情形的链路负载流量峰值统计图，由左至右 3 簇条状图依次为无故障、Minimum 算法和免疫原理算法。从图中可以看到无故障情况下流量由网络中所有链路承担，因此负载流量较均衡，因此峰值最低；与 Minimum 算法相比较，免疫原理算法重点保护业务量较大的链路，生成过程中可能不会平衡每个子层的链路数，因此个别子层相较 Minimum 算法生成子层稀疏，负载流量峰值略高，符合实际情况。

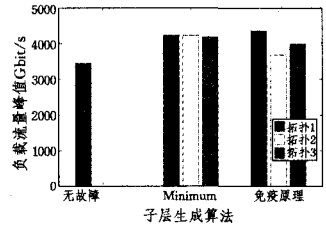
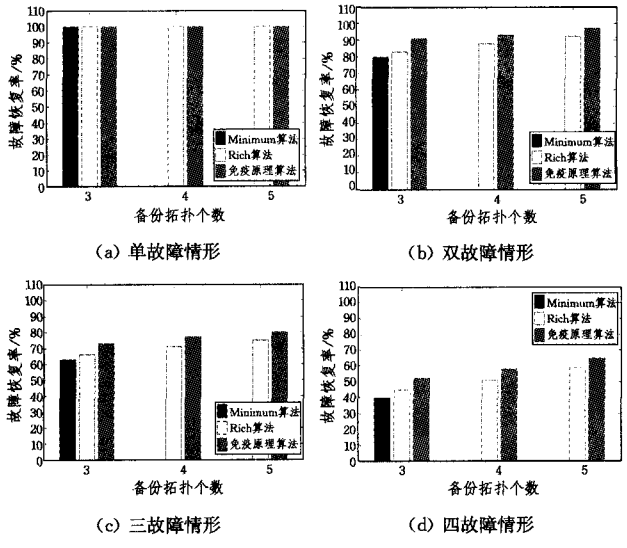


图 4　备份拓扑负载流量

4.2　子层故障恢复能力

本次实验使用 Minimum 算法、Rich 算法和免疫原理算法生成多拓扑子层分别在单故障、双故障、三故障和四故障 4 种条件下对网络业务进行保护，统计备份拓扑能够恢复的故障的几率，实验中 Rich 算法与免疫原理算法分别使用 3、4、5 个备份拓扑。

由图 5 可以看出，当备份拓扑数目相同时，使用免疫原理算法生成的子层能够恢复更多的故障，这是因为免疫原理算法在生成拓扑层过程中以多故障情形为抗原进行匹配，因此比其余两种算法更加适应多故障情形；而使用相同算法，备份拓扑越多则能够抵抗的故障越多，这是因为子层越多能够使每个子层移除更多种组合的链路，能够适应更多的故障条件。



(a) 单故障情形　(b) 双故障情形

(c) 三故障情形　(d) 四故障情形

图 5　备份拓扑鲁棒性

4.3　存储资源利用

对 MTR 子层生成算法而言，能否以较少的子层实现较高的故障恢复率，能够表明该算法对存储资源的利用程度。接下来分别使用 Minimum、Rich 和免疫原理 3 种算法，在全网络中随机产生 100 次故障，即每次从网络中随机删除一部分链路，每次删除链路的数目服从分布 $X\sim N(2,0.5)$。统计 3 种算法在达到规定故障保护率时所使用备份拓扑的个数。

由图 6 可以看出，在达到相同故障恢复率时，免疫原理算法使用的备份拓扑数目最少，Rich 算法次之，Minimum 算法使用的备份拓扑最多，这是由于免疫原理算法生成的备份拓

扑是依据链路发生故障概率大小而进行链路保护的。故免疫原理算法能够对存储资源的利用进行优化。

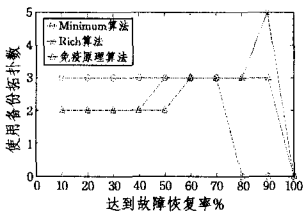


图6 存储资源利用合理性

结束语 本文提出了一种基于免疫原理的多拓扑子层生成算法，它能够有效地保障网络在多故障情况下的业务运行。由于子层中的拓扑是依据链路故障发生概率生成的，因此该算法较其它多拓扑生成算法抵抗多故障的能力更强，且生成过程使用了智能算法，使得生成子拓扑能最大限度满足目标。在后续研究中，将综合考虑网络的链路保护与负载大小，通过调整链路权值实现生成拓扑的负载均衡，进一步提高该方法的网络可生存性。

参 考 文 献

[1] Alaellinoglu C, Jacobson V, Yu H. Towards millisecond IGP convergence [EB/OL]. http://www. nanog. org/meetings/nanog20/abstracts. php? pt = MTA3MiZuYW5vZzlw&nm = nanog20, 2000

[2] 徐明伟，杨芫，李琦. 域内自愈路由研究综述[J]. 电子学报，2009，37(12)：2753-2761

[3] Zhang Xin, Perrig A. Correlation-resilient path selection in multi-path routing[C]//Proceedings of IEEE Globecom. 2010

[4] Bryant S, Shand M, Previdi S. IP fast reroute using notvia addresses[EB/OL]. http://tools. ietf. org/html/draft-ietf-rt-gwg-ipfrr-notvia-addresses-03, 2008

[5] Xu Ming-wei, Yang Yuan, Li Qi. Selecting shorter alternate paths for tunnel-based IP fast reroute[J]. Computer networks, 2012, 56(2): 845-857

[6] Lee S, Yu Yin-zhe, Nelakuditi S, et al. Proactive vs reactive approaches to failure resilient routing[C]//Proceedings of INFOCOM 2004. Hong Kong, IEEE Press, 2004: 176-186

[7] Kvalbein A, Hansen A, Cicic T, et al. Fast recovery from link failures using resilient routing layer[C]//Proceedings of the 10th IEEE symposium on computers and communications. Barcelona, Spain: ISCC Press, 2005: 554-560

[8] Scheffel M C, Gruber C G, Schwabe T, et al. Optimal multi-topology routing for IP resilience[J]. International journal of electronics and communications, 2006(60): 35-39

[9] Kvalbein A, Hansen A, Cicic T, et al. Multiple routing configurations for fast IP network recovery[J]. IEEE/ACM transactions on networking, 2009, 17(2): 473-486

[10] Luan Lin-lin, Wang Zhi-jie, Liu San-ming. Quantum Immune Algorithm for 0/1 Knapsacks Problem[J]. Intelligent Information Managent Systems and Technologies, 2012, 8(1): 117-122

[11] 焦李成，杜海峰，刘芳，等. 免疫优化计算、学习与识别[M]. 北京：科学出版社，2006：92-94

[12] Betker A, Gerlach C, Hulsermann R, et al. Reference transport network scenarios[R]. MultiTeraNet Project, 2004

[13] 黄赫，王晟. 多拓扑路由实现 IP 网络区分服务的优化算法[J]. 计算机应用研究，2010，27(12)：4735-4737

[14] 包学才，戴伏生，韩卫占. 基于拓扑的不相交路径抗毁性评估方法[J]. 系统工程与电子技术，2012，34(1)：168-174

[15] Sterbenz J, Hutchison D, Cetinkaya E, et al. Resilience and survivability in communication networks: strategies, principles, and survey of disciplines[J]. Computer networks, 2010, 54(3): 1245-1265

(上接第 207 页)

参 考 文 献

[1] Zhan Ying, Sun Yong. Cloud storage management technology [C]//2009 Second International Conference on Information and Computing Seience. 2009

[2] Storage networking Industry Association. Cloud storage for cloud computing[EB/OL]. http://www. snia. org

[3] Larry D. Cloud computing hasn't gone fortune 500 yet, but it's coming[EB/OL]. http://blogs. zdnet. corn/BTL/? p=8199

[4] Atenises G, Burns R, Curtmola R, et al. Provable data possession at untrusted stores[C]//CCS'07: Proceedings f the 14th ACM Conference on Computer and Communications Security. New York: ACM Press, 2007: 598-609

[5] Atenises G, Kamara S, Katz J. Proofs of storage from homomorphic identification protocols[C]//ASIACRYPT'09: Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology. Berlin: Springer-Verlag, 2009: 319-333

[6] Shah M A, Baker M, Mogul J C, et al. Auditing to keep online storage services honest[C]//HOTOS'07: Proceedings of the 11th USENIX Workshop on Hot Topics in Operating Systems. Berkeley, CA: USENIX Association, 2007: 1-6

[7] 曹夕，许力，陈兰香. 云存储系统中数据完整性验证协议[J]. 计算机应用，2012，32(1)：8-12

[8] 颜湘涛，李益发. 基于哈希树的云存储完整性检测算法[J]. 计算机科学，2012，39(12)：94-97

[9] Sahai A, Wates B. Fuzzy identify-based encryption [C]// Advances in Cryptology-EUROCRYPT. Berlin: Springer-Verlag, 2005: 457-473

[10] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryptio[C]//Proc of IEEE Sympoisum on Security and privacy. Washington DC: IEEE Computer Society, 2007: 321-334

[11] 刘帆，杨明. 一种用于云存储的密文策略属性基加密方案[J]. 计算机应用研究，2012，29(4)：1452-1456

[12] 刘鹏等. 云计算[M]. 北京：电子工业出版社，2010

[13] Goldberg I, Wagner D, Thomas R, et al. A Secure Environment for Untrusted Helper Applications (Confining the Wily Hacker) [C]//Proceedings of the Sixth USENIX UNIX Security Symposium. 1996

[14] Pierce B C, Vouillon J. What is in Unison[R]. MS-CIS-03-06. Philadelphia, Pennsylvania: Department of Computer and Information Science, University of Pennsylvania, 2004