

考虑负载均衡的多下一跳路由自愈方法

陈天平¹, 孟相如¹, 崔文岩¹, 许 媛²

(1 空军工程大学 信息与导航学院, 陕西 西安 710077; 2 中电集团 第 39 研究所, 陕西 西安 710065)

摘 要: 针对现有 IP 网络路由自愈方法存在的不足, 提出一种考虑负载均衡的多下一跳路由自愈方法。首先设计基于距离矢量路由协议的多下一跳路由生成算法, 改进了节点可用下一跳的生成策略; 然后考虑到故障恢复后的负载均衡问题, 采用 Dinic 算法对故障恢复路径进行动态规划, 设计了一种限制洪泛与多路径并行传输相结合的路由自愈方法; 最后, 对多下一跳路由生成算法的有效性进行了理论证明, 并通过仿真实验从可用下一跳数目、故障恢复时间两个方面验证了所提方法的相对优势。

关键词: 距离矢量路由; 多下一跳路由; 负载均衡; 路由自愈

中图分类号: TP309.1

文献标识码: A

文章编号: 1000-7180(2016)04-0016-05

A Multi Next Hop Routing Self-Healing Scheme With Load Balancing Taken Into Account

CHEN Tian-ping¹, MENG Xiang-ru¹, CUI Wen-yan¹, XU Yuan²

(1 Telecommunication Engineering Institute, Air Force Engineering University, Xi'an 710077, China;
2 No. 39 Research Institute, CETC, Xi'an 710065, China)

Abstract: Aiming at the shortages of the existing routing self-healing schemes in the IP networks, a multi next hop routing self-healing scheme considering load balancing is proposed. Firstly, the multi next hop routing generation algorithm based on distance vector routing protocol was designed, and the generation strategy of available next hop nodes was improved. Then, failure recovery paths were programmed dynamically using Dinic algorithm, considering the problem of load balancing after failure recovery, and a routing self-healing scheme combined constrained flooding with concurrent multipath transfer was designed. Lastly, the validity of multi next hop generation scheme was proved theoretically. The simulation results show that the proposed scheme has advantages in the numbers of available next hop nodes and failure recovery time.

Key words: distance vector routing; multi next hop routing; load balancing; routing self-healing

1 引言

随着互联网的迅速发展, VoIP、在线视频等实时业务大量涌现, 各类网络故障日益频发, 对传统路由协议的故障恢复能力提出了严峻挑战。在网络突发故障情况下, 网关协议 IGP、RIP 的路由收敛时间为 100 秒的数量级, 且存在“无穷计算”问题; OSPF、ISIS 的路由收敛时间在几秒到几十秒, 且存在“路由黑洞”问题, 极易造成数据包丢失^[1]。因此, 如何

提高路由自愈能力已成为当前的一个研究热点。

路由自愈是指网络在发生故障的情况下, 无需人工干预, 就能快速、自发地恢复受故障影响的业务。目前路由自愈实现方法主要有以下几类: (1) 调整权值设置^[2], 此类方法常用于链路状态协议, 仅能处理可预见的故障, 无法处理突发故障; (2) 限制洪泛^[3], 此类方法能有效节约网络资源、增强网络稳定性, 但容易形成网络环路; (3) 快速重路由^[4], 此类方法的故障恢复速度较快, 但容易造成网络局部拥塞;

(4)本地重路由^[5],此类方法的故障恢复速度极快,但一般未考虑故障恢复后的负载均衡问题,从而易导致网络震荡;(5)多下一跳^[6],是指在故障发生前预先准备好多个下一跳地址,采用多路径并行转发受故障影响的数据分组.此类方法通常会使整个网络路由的收敛时间变长,但可以缩短网络中断的时间.

目前多下一跳路由算法大致可分为两类^[7]:距离矢量类的多下一跳路由算法(如 MDVA、MN-HRP 等)和链路状态类的多下一跳路由算法(如 MPDA、QMPDA 等),文献[8]对以上两类算法进行了仿真实验,结果表明前一类算法比后一类算法具有较好的故障恢复能力和灵活性,但是前一类算法仍存在计算到无穷的缺陷.本文对现有距离矢量类的多下一跳路由算法进行改进,提出一种考虑负载均衡的多下一跳路由自愈方法.

2 多下一跳路由生成算法

在距离矢量路由协议中,根据所有节点相对于某一目的节点的距离,构建网络层次图,使网络中每一节点获知自己与其所有邻居对应于此目的节点的网络层次,然后再依据网络层次定义出邻居节点是否可为下一跳,从而节点可从这些邻居中选择可用的下一跳进行报文转发.相关定义如下所示.

距离:在网络中,两节点之间的路由跳数;

网络层次图:针对某一目的节点,网络中所有其他节点相对于它的距离层次分布图;

路由器 ID:将节点 i 的所有接口中配置的最大 IP 地址定义为节点 i 的路由器 ID;

可用下一跳集合:表示从节点 i 往目的节点 j 发送报文时的可用下一跳组成的集合.

2.1 构建网络层次图

构建网络层次图是由每个目的节点 j 发起的,其他节点收到关于目的节点 j 的网络层次图构建触发报文后,将按照下式计算自己的层次值:

$$\begin{cases} L(j, j) = 0 \\ L(i, j) = \min[L(k, j)] + 1, (j \in N, k \in K_i, k \neq i) \end{cases} \quad (1)$$

$L(i, j)$ 表示节点 i 相对于目的节点 j 所对应的层次值; N 为网络中的节点集合; K_i 为节点 i 的邻居节点集合.针对网络中任意节点 i ,其网络层次图的构建算法实现如图 1 所示.

以图 2 所示的网络拓扑为例,构建相对于目的节点 D 的网络层次如图 3 所示.

图 3 中节点之间按位置关系可分为两类:(1)同

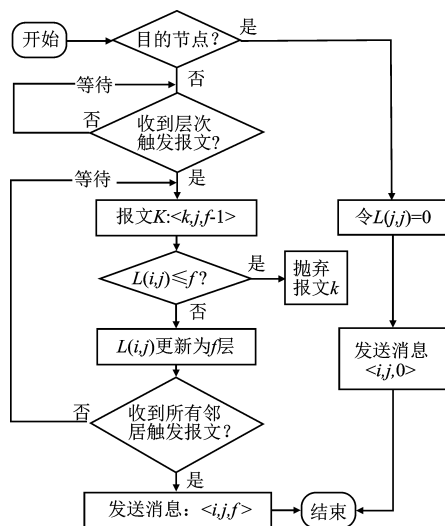


图 1 网络层次图构建流程

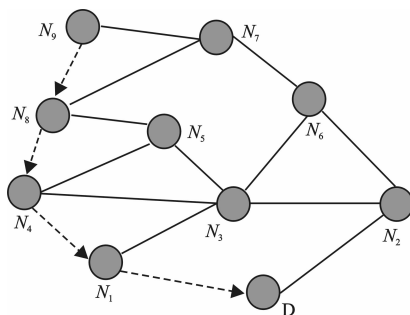


图 2 网络拓扑图

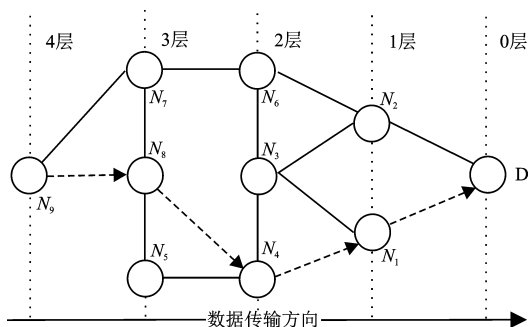


图 3 网络层次图

层关系,指层次值相同的节点,如第 2 层节点 N_3 、 N_4 、 N_6 ;(2)上、下游关系,如 N_8 为 N_4 的上游节点, N_3 为 N_5 的下游节点.

2.2 生成可用下一跳集合

当网络中所有节点都知道自己及其邻居相对于目的节点在层次图中所处的层次后,接下来每一节点将生成自己的可用下一跳集合,步骤如下所示.

Step1:目的节点定义其可用下一跳集合为空;

Step2:其余节点选择所有下游邻居节点以及 ID 号小于自己的同层邻居节点作为其可用下一跳,

并存入可用下一跳集合. 集合中的排序规则为: (1) 按节点的层次值从小到大进行排序; (2) 同层次的节点按 ID 号从大到小进行排序;

Step3: 由上一步初步生成可用下一跳集合后, 对于可用下一跳数目少于两跳的节点(目的节点除外), 为其选择与下游节点至少有两跳以上路径的最优上游邻居节点作为可用下一跳. 一旦节点的唯一主用下一跳故障时, 节点将受故障影响的流量经过这一上游邻居节点直接转发给自己的同层节点, 以避免形成环路. 整个过程如图 4 所示.

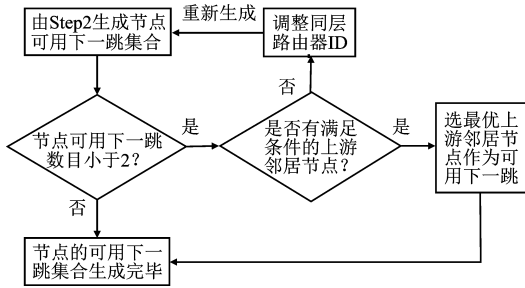


图4 可用下一跳集合更新流程图

以图 3 为例, 通过上述步骤可生成每一节点的可用下一跳集合为: $D: \{\varphi\}, N_1: \{D, [N_3]\}, N_2: \{D, [N_3]\}, N_3: \{N_1, N_2\}, N_4: \{N_1, N_3\}, N_5: \{N_3, N_4\}, N_6: \{N_2, N_3\}, N_7: \{N_6, [N_9]\}, N_8: \{N_4, N_5, N_7\}, N_9: \{N_7, N_8\}$, 其中 N_1, N_2 和 N_7 称为单跳节点, 即运行 Step2 后只能生成唯一下一跳, 因此继续运行 Step3, 为这 3 个节点选择可用的上游邻居节点作为可用下一跳(“[]”内).

2.3 可用下一跳的切换策略

可用下一跳集合中的元素按其位置关系可分为 3 类, A 类为下游可用下一跳, B 类为同层可用下一跳, C 类为上游可用下一跳. 无论网络正常或故障情况下, 节点 N 的可用下一跳切换策略如下:

① 如果节点 N 至少存在一个 A 类可用下一跳 N_a , 则将受故障影响的流量切换至 N_a 进行数据转发, 否则跳转执行 2 步;

② 如果节点 N 至少存在一个 B 类可用下一跳 N_b , 则将受故障影响的流量切换至 N_b 进行数据转发, 否则跳转执行 3 步;

③ 如果节点 N 至少存在一个 C 类可用下一跳 N_c , 则将受故障影响的流量切换至 N_c 进行数据转发, 当 N_c 收到来自节点 N 发往目的节点 D 的异常数据后, 就判定节点 N 的唯一主用下一跳发生了故障, 于是再将数据转发给除 N 以外其他下游邻居节点, 从而起到跟 1、2 步相同的效果.

3 考虑负载均衡的多下一跳路由自愈方法

现有多下一跳路由算法较少考虑到故障恢复后的流量分配问题, 从而导致部分链路过载, 严重影响网络稳定性. 为此, 提出一种新型的路由自愈模型, 采用图论中的 Dinic 算法^[9]对故障恢复路径进行动态规划, 实现网络负载均衡. 相关定义如下所示.

容量网络: 设 $G(V, E)$ 是一个有向网络, 在 V 中指定一个顶点, 称为源点(记为 V_s), 以及另一个顶点, 称为汇点(记为 V_t); 对于每一条弧 $\langle u, v \rangle \in E$, 对应一个权值 $c(u, v) > 0$, 称为弧的容量.

网络流: 在网络 G 中, 所有弧上实时流量的集合 $F_t = \{f_t(u, v)\}$, 称为容量网络 G 的一个网络流.

增广路: 从 V_s 到 V_t 的一条可达路径 P , 若 P 满足如下条件: 即在 P 的所有链路 $\langle u, v \rangle$ 上, 存在 $0 \leq f(u, v) < c(u, v)$, 则称 P 为从 V_s 到 V_t 的一条增广路.

3.1 路由自愈模型

考虑负载均衡的路由自愈模型如图 5 所示.

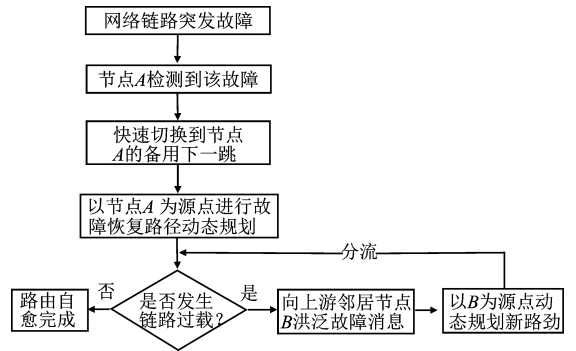


图5 路由自愈模型

在图 3 中, 假设网络正常情况下从源节点 N_9 向目的节点 D 发送数据, 其原始传输路径为 $N_9 \rightarrow N_8 \rightarrow N_4 \rightarrow N \rightarrow D$. 如果链路 L_{41} (即 $N_4 \rightarrow N_1$) 突发故障, 则节点 N_4 最先检测到该故障, 并立即将受故障影响的流量快速切换至备用下一跳 N_3 , 同时动态规划从 N_4 出发, 经过 N_3 , 最终到达 D 的故障恢复路径. 通过比较受故障影响流量与恢复路径的最大可行流量来判断链路是否发生了过载, 如果前者流量小于后者, 那么路由自愈完成, 否则网络将发生拥塞, 此时 N_4 向上游邻居节点 N_8 洪泛此故障消息. 当 N_8 收到此消息后, 将以 N_8 为源点动态规划到达目的节点 D 的新路径, 利用这些路径分流受故障影响的流量, 以减少通过 N_4 的总流量, 从而尽可能保证故障恢复后的网络负载均衡.

3.2 基于 Dinic 算法的恢复路径动态规划

首先将 IP 网络抽象为一个容量网络 $G(V, E)$, 为每条链路 $l \in E$ 给出一组数 $(c(l), f_t(l))$, 其中 $c(l)$ 代表链路 l 的最大可用带宽, $f_t(l)$ 代表 t 时刻链路 l 上的实际流量; 其次将故障链路从原拓扑中删掉以生成新的子拓扑; 最后确定增广路的搜索源点 N_i . 在子拓扑中, 采用标号法进行增广路搜索的步骤如下:

Step1: 查找 N_i 的可用下一跳集合, 按照下一跳路由 (去除故障下一跳) 排序依次进行遍历, 并对遍历到的每个下一跳路由进行标号. 假设遍历到 N_i 的可用下一跳 N_k , 则对 N_k 的标号为 $(N_i, e_t(l_{ik}))$, 其中 $e_t(l_{ik}) = c(l_{ik}) - f_t(l_{ik})$ 为链路的当前可行流量, 若当 $e_t(l_{ik}) = 0$ 时, 放弃对 N_k 对进行标号;

Step2: 将 N_k 作为新的搜索源点, 对其所有可用下一跳进行遍历和标号, 如此不断循环, 直到遍历到目的节点为止;

Step3: 根据目的节点的标号, 采用“倒向追踪”的方法, 一直找到源点 N_i , 使得所有增广路.

以图 3 中链路 l_{41} 发生故障为例, 以 N_4 为源点进行故障恢复路径动态规划可得增广路 P_1 和 P_2 , 如图 6、图 7 所示.

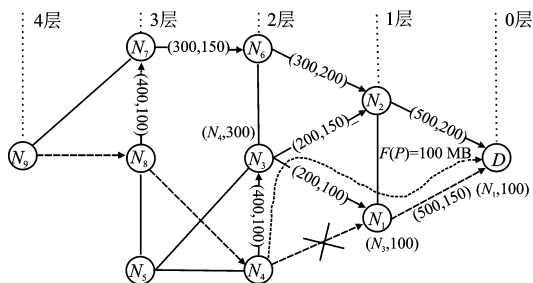


图6 第1次标号

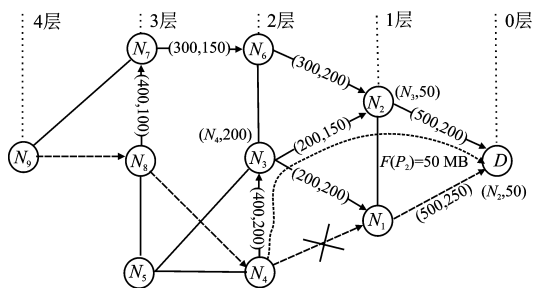


图7 第2次标号

当受故障影响的流量 $F(f)$ 小于或等于这两条增广路的可行流量之和 $(F(P_1) + F(P_2) = 150 \text{ MB})$ 时, 路由自愈过程结束. 而当 $F(f) > 150 \text{ MB}$ 时, 可能会导致某些链路过载, 因此 N_4 向原路径上游邻居 N_8 洪泛此故障消息, N_8 接收到该消息后, 以 N_8 为源点进行故

障恢复路径动态规划可得增广路 P_3 , 其可行流量 $F(P_3)$ 为 100 MB, 即表明受故障影响的流量可以经过恢复路径 P_3 至多分流 100 MB, 如图 8 所示.

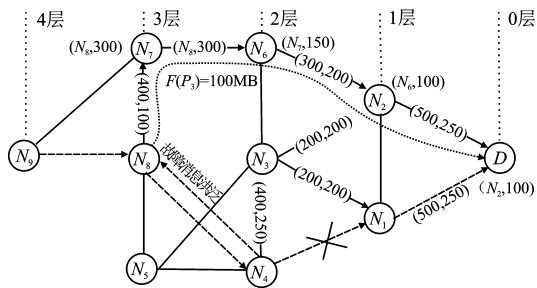


图8 第3次标号

3.3 Dinic 算法的复杂度分析

假设所得增广路的平均长度为 m 跳, 节点的可用下一跳数目平均为 n 个, 那么针对某一个源点进行增广路搜索的时间复杂度为 $O(n^m)$. 另外, 该算法的时间复杂度还与各条链路的容量相关, 一旦增广路的最大可行流量不够时, 就会触发故障消息洪泛, 并重启该算法. 假设恢复某次故障需要向上洪泛 p 步, 则该算法的时间复杂度至少可表示为 $O(p \cdot n^m)$.

4 算法理论证明与仿真实验

4.1 算法理论证明

论证问题: 网络正常情况下, 采用本文所提的多下一跳路由生成算法建立的路由信息是正确、有效的.

证明: 现假设数据从源节点 S 发往目的节点 D , 节点转发数据所使用的主用下一跳是依据可用下一跳切换策略选出的. 在此需要证明两点: (1) 网络正常情况下, 该算法能找到一条从 S 到 D 的可达路径 $S \rightarrow N_1 \rightarrow N_2 \rightarrow \dots \rightarrow N_n \rightarrow D$; (2) 在路径 $S \rightarrow N_1 \rightarrow N_2 \rightarrow \dots \rightarrow N_n \rightarrow D$ 上没有环路, 即 $\{N_1, N_2, \dots, N_n\}$ 互不相同.

证明(1): 在数据转发中, 节点选取可用下一跳会存在三种可能情况: ① 选用 A 类下一跳; ② 选用 B 类下一跳; ③ 选用 C 类下一跳. 如果路径上所有节点都选 A 类下一跳, 那么数据是沿距离差变小的方向传播的, 所以一定能找到一条路径到达节点 D . 如果节点选 B 类下一跳 N_b , 即数据在同层之间进行传输, 那么 N_b 将继续转发数据, 其选取可用下一跳也存在以上三种可能情况: ④ N_b 选用 A 类下一跳; ⑤ N_b 选用 B 类下一跳; ⑥ N_b 选用 C 类下一跳. 如果是情况 4, 那么数据一定能找到一条路径到达节点 D . 如果是情况 5, 因为同层节点数目有限, 且满足 B 类下一跳 ID 号小于 N_b 的 ID 号, 因此总能找到一个

同层节点将数据转发给 A 类下一跳。如果是情况 6, 数据经过 C 类下一跳中转后, 再转发给自己的同层非邻居节点, 与情况 5 类同。如果节点选 C 类下一跳 N_c , 那么 N_c 将继续转发数据, 此时 N_c 只能选取 A 类下一跳, 因此与情况 5 类同。综上所述, 本文所提算法总会找到一条路径将数据送达目的节点。

使用反证法证明(2): 假设 $N_i = N_k$, 且不妨假设 $i < k$, 即有环路 $N_i \rightarrow N_{i+1} \rightarrow \dots \rightarrow N_{k-1} \rightarrow N_{k(i)}$, 且 $\{N_i, N_{i+1}, \dots, N_{k-1}\}$ 互不相同。由以上假设可知, N_{k-1} 选择 N_i 作为自己的下一跳, 所以有

$$L(N_i, D) \leq L(N_{k-1}, D) \quad (2)$$

由可用下一跳生成规则可得

$$L(N_i, D) \geq L(N_{i+1}, D) \geq \dots \geq L(N_{k-1}, D) \quad (3)$$

联合式(2)和式(3)可得

$$L(N_i, D) = L(N_{i+1}, D) = \dots = L(N_{k-1}, D) \quad (4)$$

由式(4)可知节点 $\{N_i, N_{i+1}, \dots, N_{k-1}\}$ 均是同层节点, 联合式(2)可得

$$ID(N_i) \leq ID(N_{k-1}) \quad (5)$$

再由节点选择同层节点作为可用下一跳的条件可得如下关系式:

$$ID(N_i) \geq ID(N_{i+1}) \geq \dots \geq ID(N_{k-1}) \quad (6)$$

由式(5)、(6)可得 $ID(N_i) = ID(N_{i+1}) = \dots = ID(N_{k-1})$, 即 $\{N_i, N_{i+1}, \dots, N_{k-1}\}$ 据均为同一节点, 这与假设条件相矛盾, 所以假设不成立, 即路径不存在环路。

当网络中发生单故障或并发多故障时, 只要网络中存在到达目的节点的路径, 采用该算法建立的多下一跳路由信息就一定能够找到一条路径到达目的节点。其证明方法与上类同, 不再赘述。

4.2 仿真实验及分析

MDVA^[10]是一种经典的多下一跳距离矢量算法, 通过采用分布式贝尔曼福特算法 DBF 来计算到目的地的最优距离, 并通过无环不等式 LFI 来保证无环路由。为了验证本文所提算法的优势, 使用 SS-FNet 仿真工具对本文所提算法和 MDVA 进行仿真实验, 拓扑是 BRITE 生成的 20 个节点的随机拓扑, 接口是 10 Mbit/s 的以太网口, 设链路延迟为 0.001 s, 接口输出延迟 0.001 s, 路由器故障检测时间设为 0.02 s。

(1)可用下一跳数目

在网络中分别配置本文所提的多下一跳路由生成算法和 MDVA 算法, 统计全网到各个目的节点

的可用下一跳数目, 网络拓扑由 BRITE 随机产生的 20 个节点的网络拓扑, 实验结果表明。

采用本文所提算法生成可用下一跳总数目大于或等于 40 个(即表明每一节点的可用下一跳平均数目均大于两跳), 远大于采用 MDVA 算法生成可用下一跳总数目。

(2)故障恢复时间

故障恢复时间取检测故障时刻与路由表重新计算并稳定时刻之差, 使用网络拓扑与(1)中相同, 对本文所提算法和 MDVA 算法的故障恢复时间进行仿真比较。

分别在网络中随机断开 1、2、3、4、5 条链路, 仿真对比结果表明。

虽然本文所提算法采用了局部收敛策略, 但随着网络中故障链路数量的增多, 其故障恢复时间增长速度变快, 而 MDVA 在 1 条链路故障时, 其故障恢复时间与本文所提算法相差不大, 随着故障链路数量的增加, 其故障恢复时间也不断增大; 在相同失效链路数目情况下, MDVA 的故障恢复时间明显大于本文所提算法。

5 结束语

本文设计基于距离矢量路由协议的多下一跳路由生成算法, 通过理论证明表明无论在网络正常或故障情况下, 该算法所建立的路由信息是正确的、有效的; 另外, 本文提出了一种限制洪泛与多路径并行传输相结合的路由自愈方法, 通过仿真实验从可用下一跳数目、故障恢复时间两个方面对本文所提方法进行了验证, 仿真结果表明本文所提方法比 MDVA 算法具有更好的故障处理能力。

参考文献:

- [1] 徐明伟, 杨莞, 李琦. 域内自愈路由研究综述 [J]. 电子学报, 2009, 37(12): 2753-2761.
- [2] Pierre Francois, Mike Shand, Olivier Bonaventure. Disruption free topology reconfiguration in OSPF networks [C]// Proceedings of INFOCOM 2007. Anchorage, AK: IEEE Press, 2007: 89-97.
- [3] Karthik Lakshminarayanan, Matthew Caesar, Murali Rangan, et al. Achieving convergence-free routing using failure carrying packets [C]// Proceedings of ACM SIGCOMM 2007. Kyoto, Japan: ACM Press, 2007, 37(4): 241-252.

(下转第 26 页)

在不显著增加资源的情况下,有效地提高了时钟频率;与 CLA 电路结构相比,时钟频率略有降低,但资源开销却大幅减小.

5 结束语

本文完成了高速 SM3 算法 ASIC 实现,并且在硬件设计中关键路径进行优化,有效地减少了时钟延时,极大地提高了密码处理性能,使之能够应用于密码处理速度要求较高的环境. 同时本文设计的 ASIC 兼具很好的通用性,既可以作为单独的密码处理模块挂接在总线上,也可以与其他算法模块配合,构成嵌入式密码系统.

参考文献:

[1] 刘宗斌,马原,荆继武,等. SM3 哈希算法的硬件实现与研究 [C]//第 26 次全国计算机安全学术交流会论文集. 武夷山:国计算机学会计算机安全专业委员会, 2011:191-193.

[2] 王晓燕,杨先文. 基于 FPGA 的 SM3 算法优化设计与实现 [J]. 计算机工程, 2012,38(6):244-246.

[3] 张倩,李树国. SM3 杂凑算法的 ASIC 设计和实现[J]. 微电子学与计算机,2014,31(9):143-146.

[4] Bai Ling , Li Shuguo. VLSI implementation of high speed SHA-256 [C]// IEEE 8th International Confer-

ence on ASICON. Changsha, Hunan: IEEE, 2009: 131-134.

[5] Ao Tianyong, He Zhangqing, Rao Jinli, et al. A compact hardware implementation of SM3 hash function [C] // 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications. Beijing: IEEE, 2014:846-850.

[6] Henzen L, Aumasson J, Meier W, et al. VLSI characterization of the cryptographic hash function BLAKE [J]. Very large scale integrationsystems, IEEE transactions on, 2011, 19(10):1746-1754.

[7] Krad H, Al-Taie A Y. Performance analysis of a 32-bit multiplier with a carry-look-ahead adder and a 32-bit multiplier with a ripple adder using VHDL [J]. Journal of Computer Science, 2008, 4(4):305-308.

[8] 国家密码管理局. SM3 密码杂凑算法[S]. 北京:国家密码管理局,2010.

作者简介:

于永鹏 男,(1991-),硕士研究生. 研究方向为安全芯片设计与防护. E-mail:1228964300@qq. com.

严迎建 男,(1973-),博士,教授. 研究方向为密码芯片安全防护.

李 伟 男,(1983-),博士研究生,讲师. 研究方向为专用集成电路设计.

(上接第 20 页)

[4] Pierre Francois, Clarence Filsfils, John Evans, et al. Achieving subsecond IGP convergence in large IP networks [J]. ACM SIGCOMM Computer Communication Review, 2005, 35(2):35-44.

[5] Csiko L, Tapolcai J, Retvari G. Optimizing IGP link costs for improving IP-level resilience with Loop-Free Alternates [J]. Computer Communications, 2013, 36(6): 645-655.

[6] 杨华卫,王洪波,程时端,等. 最小割多路径路由算法 [J]. 软件学报, 2012, 23 (8): 2115-2129.

[7] 王滨,杨强,吴春明. IP 网络可生存性技术[M]. 北京:人民邮电出版社,2013.

[8] Vutukury S. Multipath routing mechanisms for traffic engineering and quality of service in the Internet[D]. Santa Cruz: University of California, 2001.

[9] 高随祥. 图论与网络流理论[M]. 北京:高等教育出

版社, 2009.

[10] Vutukury S, Garcia-Luna-Aceves J J. MDVA: A distance-vector multipath routing protocol[C]// Proceeding of the INFOCOM. Anachorage: IEEE, 2001, 1: 557-564.

作者简介:

陈天平 男,(1979-),博士后. 研究方向为信息安全及网络可生存性. E-mail:690004398@qq. com.

孟相如 男,(1963-),教授,博士生导师. 研究方向为宽带网络通信系统.

崔文岩 男,(1987-),博士研究生. 研究方向为信息系统容灾抗毁.

许 媛 女,(1983-),工程师. 研究方向为微波天线设计及测试.