

分布式紧耦合虚拟专用网可靠性计算方法

戴伏生 刘功亮

(哈尔滨工业大学威海校区 威海 264209)

摘要: 为了评估构建在互联网基础上,具有多个服务质量约束要求的分布式紧耦合虚拟专用网可靠性能,该文提出一套可靠性指标计算方法。首先,在赋有权值参数的互联网络中,计算出所有满足约束条件的分布式紧耦合节点之间路由。然后,根据这些路由建立满足服务质量要求的网络邻接矩阵,对邻接矩阵进行分解与整合等变换,求得满足约束条件的不交化网络状态集。最后,根据不交化网络状态集得到可靠性指标的计算式,进而完成可靠性指标计算。由于所有过程都实现了代数化或逻辑代数化计算,所以易于计算机编程实现。论述了算法关键环节的正确性。通过算例、实验以及与传统算法结果比较等方式,说明新算法能够非常贴切地反映出分布式紧耦合虚拟专用网的可靠性。

关键词: 通信网络; 可靠性; 性能评估; 虚拟网络; 服务质量路由; 路由算法

中图分类号: TN915.2

文献标识码: A

文章编号: 1009-5896(2009)07-1582-06

Reliability Evaluation Algorithm for Distributed Tightly-Coupled Virtual Private Networks

Dai Fu-sheng Liu Gong-liang

(Weihai Campus, Harbin Institute of Technology, Weihai 264209, China)

Abstract: An algorithm of reliability indexes is proposed in this paper for evaluating the reliability of the distributed tightly-coupled virtual private networks with multi-QoS requirements, which is constructed in the environment of interconnection network. Firstly, according to the requirements of QoS, all routes satisfying the QoS requirements are calculated between the distributed tightly-coupled nodes in the weighted networks. Secondly, the network adjacency matrix is constructed according to these routes, and the disjoint network state set satisfying the QoS requirements is obtained by decomposition transformation and conformity transformation of the adjacency matrix. Finally, the calculation expression of the reliability indexes is obtained according to the disjoint network state set, and then the reliability indexes are accomplished. The whole procedure is realized through algebra or logical algebra calculation, and it is easy to be programmed by computer. The correctness of each key step of the new method is discussed. It is proved by numerical example, experiment and comparison with the results of the traditional algorithm, that the new algorithm reflects relevantly the reliability of the distributed tightly-coupled virtual private networks.

Key words: Communication networks; Reliability; Performance evaluation; Virtual Private Network(VPN); QoS routing; Route algorithm

1 引言

构建在互联网上的分布式紧耦合虚拟专用网 VPN(Virtual Private Network)^[1,2],不仅对核心节点与任务端节点之间有很高的信息安全传输要求,而且对通信的服务质量 QoS(Quality of Service)有多约束要求,相关设备失效或者 QoS 达不到约束要求,任务就不能按计划完成。虽然,学者们在传统网络的可靠性计算与评价方法方面取得了许多相对成熟的研究成果^[3-8]。然而,这些方法并不适合分析具有多约束 QoS 要求的分布式紧耦合 VPN 可靠

性能。学者们也曾经提出过分布式网络可靠性评估方法^[9-14],且用于分析无 QoS 约束要求的专用网络也是有效的。但是不考虑 QoS 约束要求的可靠性评估,不能贴切地反映出构架在互联网基础上的分布式紧耦合 VPN 实际情况。如何评估互联网络能给分布式紧耦合 VPN 提供多大的通信可靠性保证,目前尚未见到报道。因此需要研究这类网络的可靠性评估方法。

2 网络模型及相关定义

(1)把互联网络抽象为一个赋权值的图 $G(V, E, W)$ 。其中, $V = \{v_1, v_2, \dots, v_n\}$ 代表所有交换节点

组成的集合, $E = \{e_1, e_2, \dots, e_m\}$ 代表所有链路的集合, $W = \{(w_1^1, w_2^1, \dots, w_{n+m}^1), (w_1^2, w_2^2, \dots, w_{n+m}^2), \dots, (w_1^t, w_2^t, \dots, w_{n+m}^t)\}$ 代表所有节点和链路 t 种权值参数的集合, W 将影响到网络的 QoS。图中存在一个任务的核心节点, 称为汇聚节点; 若干个协同完成特定任务的节点, 称为协作节点; 汇聚节点和各协作节点以互联网络为基础建立路由构建出分布式紧耦合 IPSec VPN。其中需要经过的中继接力节点, 统称为辅助节点。

(2) 每个节点和每条链路的权值参数, 通过统计和预测是可知的。包括带宽、时延、时延抖动和数据包丢失率等几种, 不同的业务 QoS 要求也不同, 但至少包含其中一种参数。

(3) 每个节点和每条链路有统计独立的正常或失效两种状态, 以可用性和不可用性概率来度量。

(4) 用 $v_{ij}(r)$ 表示从节点 v_i 到 v_j 串联 r ($r \geq 1$) 条链路的路径或路由。当 $r = 1$ 时路由退化为节点 v_i 到 v_j 的直达链路, 写为 $v_{ij}(1)$ 或 v_{ij} 。用 $G[v_{ij}(r)]^{\text{bit}} = [v_1, v_2, \dots, v_n]$ 表示路由节点利用标记分组, 当节点与路由相连时元素取 1; 否则取 0。用 $L[v_{ij}(r)]^{\text{bit}} = [e_1, e_2, \dots, e_m]$ 表示路由的链路利用分组, 当某链路 $e_x \in$ 路由时元素取 1; 否则取 0。用 $X[v_{ij}(r)]^{\text{bit}} + X[v_{kh}(r)]^{\text{bit}}$ 表示两个等长分组对应位(bit)进行“逻辑或”运算。

(5) 若两个网络状态表达式“逻辑与”的结果为 0, 则为不相交网络状态。若网络状态集合中任意两式均不相交, 则为网络状态不交化集合。

3 多约束 QoS 路由集计算方法

步骤 1 初始化后, 赋予 $r = 1$, 建立邻接分组和关联分组, 确定组中元素初始表达式, 方法如下。

(1) 建立邻接分组。设网络有 n 个节点 $v_1, v_2, \dots, v_x, \dots, v_y, \dots, v_z, \dots, v_n$ 。不妨假设 v_1 为汇聚节点, v_x, v_y, v_z 为协作节点。建立 n 个节点的邻接分组为: $V_1 = [v_{12}, v_{13}, \dots, v_{1x}, \dots, v_{1y}, \dots, v_{1z}, \dots, v_{1n}] \dots, V_x = [v_{2x}, v_{3x}, \dots, v_{xx}, \dots, v_{yx}, \dots, v_{zx}, \dots, v_{nx}] \dots, V_y = [v_{2y}, v_{3y}, \dots, v_{xy}, \dots, v_{yy}, \dots, v_{zy}, \dots, v_{ny}] \dots, V_z = [v_{2z}, v_{3z}, \dots, v_{xz}, \dots, v_{yz}, \dots, v_{zz}, \dots, v_{nz}] \dots, V_n = [v_{2n}, v_{3n}, \dots, v_{xn}, \dots, v_{yn}, \dots, v_{zn}, \dots, v_{nn}]$ 。元素的初始表达式为: 当 $i = j$ 表示 v_i 节点本身时, $v_{ii} = 0$; 当节点 v_i 与 v_j 无直接连接链路时, $v_{ij} = 0$; 当节点 v_i 与 v_j 直接连接链路为 e_a 时, $v_{ij} = e_a$ 。

(2) 建立关联分组。设网络有 m 条链路 e_1, e_2, \dots, e_m , 建立 m 个关联分组形式为: $G[e_a]^{\text{bit}} = G[v_{ij}]^{\text{bit}} = [v_1, v_2, \dots, v_x, \dots, v_y, \dots, v_z, \dots, v_n]$ 。其中, $a = 1, 2, \dots, m$, $i, j = 1, 2, \dots, n$, $i \neq j$ 且 $v_{ij} \neq 0$ 。元素的初始表达式只有两个元素为 1, 当节点 v_i 和 v_j 与链路 e_a 有连接时 v_i 和 v_j 用 1 表示, 其余元素用 0 表示。

步骤 2 判断 v_1 节点与其它所有节点之间可达路由集是否全部为空, 如果全部为空, 则执行步骤 5。

步骤 3 依据权值参数及约束条件计算与筛选满足条件的路由 $v_{1x}(r), v_{1y}(r)$ 和 $v_{1z}(r)$, 并生成路由节点利用标记分组 $G[v_{1x}(r)]^{\text{bit}}, G[v_{1y}(r)]^{\text{bit}}$ 和 $G[v_{1z}(r)]^{\text{bit}}$, 及链路利用标记分组 $L[v_{1x}(r)]^{\text{bit}}, L[v_{1y}(r)]^{\text{bit}}$ 和 $L[v_{1z}(r)]^{\text{bit}}$ 。

步骤 4 令 $r = r + 1$, 判断 $r \leq n - 1$? 成立则计算 v_1 节点到其它所有节点的可达路由集, 伴随进行关联分组迭加运算, 判断并删除闭环路由, 同时标注非闭环路由的端节点, 然后跳转到步骤 2; 当 $r > n - 1$, 则表明所有可达路由由全部搜索完毕, 顺序执行步骤 5。其中汇聚节点与协作节点的直达路由(即链路), 在建立邻接分组时就已经得到, 串联链路两个以上的路由需要计算, 方法如下。

(1) 用邻接分组的迭代运算获得路由表达式。用 $V_1(r)$ 表示汇聚节点串联链路数为 r 才能到达协作节点的路由分组。组中 $v_{ik}(r)$ 表示从节点 i 出发, 通过其它节点串联 r 条链路才能到达协作节点 k 的一组路由。求汇聚节点到各协作节点之间串联 $r + 1$ 条链路 ($1 \leq r \leq n - 1$) 的路由集, 按照以下迭代式计算:

$$\begin{aligned} V_1(r+1) = & [v_{12}(r+1), v_{13}(r+1), \dots, v_{1x}(r+1), \dots, \\ & v_{1y}(r+1), \dots, v_{1z}(r+1), \dots, v_{1n}(r+1)] \\ = & [(v_{12}(r) \cdot v_{22} + v_{13}(r) \cdot v_{32} + \dots + v_{1x}(r) \cdot v_{x2} + \dots + v_{1y}(r) \cdot v_{y2} + \dots + v_{1z}(r) \cdot v_{z2} \\ & + \dots + v_{1n}(r) \cdot v_{n2}), (v_{12}(r) \cdot v_{23} + v_{13}(r) \cdot v_{33} \\ & + \dots + v_{1x}(r) \cdot v_{x3} + \dots + v_{1y}(r) \cdot v_{y3} + \dots \\ & + v_{1z}(r) \cdot v_{z3} + \dots + v_{1n}(r) \cdot v_{n3}), \dots, (v_{12}(r) \cdot v_{2x} + v_{13}(r) \cdot v_{3x} + \dots + v_{1x}(r) \cdot v_{xx} + \dots \\ & + v_{1y}(r) \cdot v_{yx} + \dots + v_{1z}(r) \cdot v_{zx} + \dots + v_{1n}(r) \cdot v_{nx}), \dots, (v_{12}(r) \cdot v_{2y} + v_{13}(r) \cdot v_{3y} + \dots \\ & + v_{1x}(r) \cdot v_{xy} + \dots + v_{1y}(r) \cdot v_{yy} + \dots + v_{1z}(r) \cdot v_{zy} + \dots + v_{1n}(r) \cdot v_{ny}), \dots, (v_{12}(r) \cdot v_{2z} + v_{13}(r) \cdot v_{3z} + \dots + v_{1x}(r) \cdot v_{xz} + \dots + v_{1y}(r) \cdot v_{yz} + \dots \\ & + v_{1z}(r) \cdot v_{zz} + \dots + v_{1n}(r) \cdot v_{nz}), \dots, (v_{12}(r) \cdot v_{2n} \\ & + v_{13}(r) \cdot v_{3n} + \dots + v_{1x}(r) \cdot v_{xn} + \dots + v_{1y}(r) \cdot v_{yn} \\ & + \dots + v_{1z}(r) \cdot v_{zn} + \dots + v_{1n}(r) \cdot v_{nn})] \quad (1) \end{aligned}$$

式中 $v_{1x}(r+1), v_{1y}(r+1)$ 和 $v_{1z}(r+1)$ 表示的是汇聚节点到协作节点串联 $r + 1$ 条链路的路由集, 其它 $v_{1j}(r+1)$ 表达式, 其中 $j \neq x, y, z$, 表示的是汇聚节点到辅助节点串联 $r + 1$ 条链路的路由集。

(2) 删除闭环路由并添加节点。按照式(1)迭代计

算得到路由结果，当 $v_{1i}(r+1) = v_{1i}(r) \cdot v_{ij} \neq 0$ ， $i, j = 2, 3, \dots, n$ ，需要伴随进行关联分组迭加运算，计算公式如下：

$$G[v_{1i}(r+1)]^{\text{bit}} = G[v_{1i}(r)]^{\text{bit}} + G[(v_{ij})]^{\text{bit}} \quad (2)$$

若出现 $G[v_{1i}(r+1)]^{\text{bit}} = G[v_{1i}(r)]^{\text{bit}}$ ，表明路由出现闭环，则令 $v_{1i}(r+1) = 0$ ，删除它。当 $G[v_{1i}(r+1)]^{\text{bit}} \neq G[v_{1i}(r)]^{\text{bit}}$ 时，分组中新出现 1 所对应的节点号就是 $v_{1i}(r+1)$ 路由末端节点，添加即可。

步骤 5 输出路由集和对应的路由节点利用标记分组，以及各路由链路利用标记分组。

4 可靠性计算方法

4.1 可靠性指标计算公式

设需要计算汇聚节点 v_1 到协作节点 $\{v_x, v_y, v_z\}$ 之间的可靠性，计算公式为

$$\text{PI} = \sum_{s_{1(xy z)} \in S_{1(xy z)}} P_{s_{1(xy z)}} \quad (3)$$

式中 $S_{1(xy z)}$ 表示汇聚节点 v_1 到协作节点 $\{v_x, v_y, v_z\}$ 之间，存在满足 QoS 连接的有效不变化网络状态集； $s_{1(xy z)}$ 为某有效不变化网络状态； $P_{s_{1(xy z)}}$ 表示 $s_{1(xy z)}$ 出现的概率，按照以下规则和公式计算。

$$\left. \begin{aligned} X_i &\leftrightarrow p_i, \bar{X}_i &\leftrightarrow q_i \\ X_c &\leftrightarrow P_c, \bar{X}_c &\leftrightarrow Q_c \\ \text{布尔代数积} &\leftrightarrow \text{算术积} \\ P_{s_{1(xy z)}} &= \prod_{e_i \in X_i} p_i \prod_{e_i \in \bar{X}_i} q_i \prod_{v_c \in X_c} P_c \prod_{v_c \in \bar{X}_c} Q_c \end{aligned} \right\} \quad (4)$$

式中 p_i 和 q_i 分别表示链路 e_i 正常 (X_i) 和失效 (\bar{X}_i) 的概率。 P_c 和 Q_c 分别表示节点 v_c 正常 (X_c) 和失效

		v_2	v_3	\dots	v_{i-1}	v_i	v_{i+1}	\dots	v_w
v_1	v_{12}	v_{13}	\dots	$v_{1(i-1)}$	v_{1i}	$v_{1(i+1)}$	\dots	v_{1w}	$v_{1(w-1)}$
v_2	v_{22}	v_{23}	\dots	$v_{2(i-1)}$	v_{2i}	$v_{2(i+1)}$	\dots	v_{2w}	$v_{2(w-1)}$
\vdots	\vdots	\vdots	\dots	\vdots	\vdots	\vdots	\dots	\vdots	\vdots
v_{i-1}	$v_{(i-1)2}$	$v_{(i-1)3}$	\dots	$v_{(i-1)(i-1)}$	$v_{(i-1)i}$	$v_{(i-1)(i+1)}$	\dots	$v_{(i-1)w}$	$v_{(i-1)(w-1)}$
v_i	v_{i2}	v_{i3}	\dots	$v_{i(i-1)}$	v_{ii}	$v_{i(i+1)}$	\dots	v_{iw}	$v_{i(w-1)}$
v_{i+1}	$v_{(i+1)2}$	$v_{(i+1)3}$	\dots	$v_{(i+1)(i-1)}$	$v_{(i+1)i}$	$v_{(i+1)(i+1)}$	\dots	$v_{(i+1)w}$	$v_{(i+1)(w-1)}$
\vdots	\vdots	\vdots	\dots	\vdots	\vdots	\vdots	\dots	\vdots	\vdots
v_w	v_{w2}	v_{w3}	\dots	$v_{w(i-1)}$	v_{wi}	$v_{w(i+1)}$	\dots	v_{ww}	$v_{w(w-1)}$

(5) 确定汇聚节点与协作节点之间联络分组。联络分组只有 1 个，形式为 $H^{\text{bit}} = [v_1, v_2, \dots, v_x, \dots, v_y, \dots, v_z, \dots, v_w]$ ，其中元素取值的规则是对应汇聚节点与协作节点的元素取 1，按照假设只有 $v_1 = v_x = v_y = v_z = 1$ ，其余取无关符号“*”。

步骤 2 计算考虑辅助节点可能失效的节点各种不变化组合形态和相应邻接矩阵，计算方法如下。

(\bar{X}_c) 的概率。

4.2 可靠性指标计算方法

步骤 1 按照上节方法计算得到各链路对应的关联分组，汇聚节点和协作节点之间满足约束条件的路由集，各路由对应的路由节点利用标记分组 $G[v_{1x}(r)]^{\text{bit}}$, $G[v_{1y}(r)]^{\text{bit}}$ 和 $G[v_{1z}(r)]^{\text{bit}}$ ，以及路由链路利用标记分组 $L[v_{1i}(r)]^{\text{bit}}$ 。根据这些条件得到满足 QoS 的各种分组和建立邻接矩阵，方法如下：

(1) 分别计算 $\sum G[v_{1x}(r)]^{\text{bit}}$ ， $\sum G[v_{1y}(r)]^{\text{bit}}$ 和 $\sum G[v_{1z}(r)]^{\text{bit}}$ ，并判断结果是有为全 0 否，有则说明汇聚节点和各协作节点之间无法全部满足约束条件，输出 0 后退出可靠性指标计算。

(2) 计算 $\sum L[v_{1x}(r)]^{\text{bit}} + \sum L[v_{1y}(r)]^{\text{bit}} + \sum L[v_{1z}(r)]^{\text{bit}}$ 。裁减掉计算结果中 0 对应的链路，重新整理得到满足 QoS 要求网络中路由链路利用标记分组。

(3) 计算 $\sum G[v_{1x}(r)]^{\text{bit}} + \sum G[v_{1y}(r)]^{\text{bit}} + \sum G[v_{1z}(r)]^{\text{bit}}$ ，删除计算结果中 0 对应的节点，重新整理得到满足 QoS 要求网络中各链路的关联分组。

(4) 建立邻接矩阵。设满足 QoS 要求的分布式网络有 w 个节点 $v_1, v_2, \dots, v_x, \dots, v_y, \dots, v_z, \dots, v_w$ ， l 条链路 e_1, e_2, \dots, e_l 。汇聚节点 v_1 到协作节点 v_x, v_y 和 v_z 之间邻接矩阵如下。邻接矩阵元素初值取值方法与上节邻接分组初值取值方法相同。矩阵上侧和左侧为节点标记符号不参加运算。

首先，按照逻辑极小项分解法对辅助节点不变化分解，得到节点可能出现的各种不变化组合形态。然后，对每种形态进行分别处理，根据辅助节点失效即表达式中的反变量，在邻接矩阵 V 中删除失效节点所对应行和列，结果就是节点不变化组合形态对应的邻接矩阵。

步骤 3 对节点各种不变化组合形态的邻接矩

阵进行变换, 得到满足 QoS 的链路不变化网络状态集, 计算方法为: 依据邻接矩阵第 1 行不为零的元素, 不断进行矩阵分解与合并运算, 同时伴随着 QoS 路由匹配和节点匹配运算得到。比如对不为零元素 v_{1i} 完成 1 次矩阵分解与整合运算的规律性运算公式如下。

$$F([\mathbf{V}]_{w(w-1)}) = f_{1i} \cdot F([\mathbf{V}']_{(w-1)(w-2)}) + \bar{f}_{1i} \cdot F([\mathbf{V}'']_{w(w-1)}) \quad (5)$$

式中 \mathbf{V}' 和 \mathbf{V}'' 是变换后的矩阵表达式。 \mathbf{V}' 是在矩阵 \mathbf{V} 的基础上, 把 v_i 行元素“逻辑或”到第 1 行对应元素之后, 删除 v_i 行和 v_i 列的结果; \mathbf{V}'' 则是在矩阵 \mathbf{V} 的基础上, 仅令矩阵元素 $v_{1i} = 0$ 便得到结果。 f_{1i} 和 \bar{f}_{1i} 称为分解函数变量, 表达式确定方式为: 当 $v_{1i} = e_i$ 时, $f_{1i} = e_i$, $\bar{f}_{1i} = \bar{e}_i$; 当 $v_{1i} = e_i + e_j + e_k + \dots + e_{m-1} + e_m$ 时, $f_{1i} = e_1 + \bar{e}_1 e_2 + \dots + \bar{e}_1 e_2 e_3 \dots \bar{e}_{m-1} e_m$, $\bar{f}_{1i} = \bar{e}_1 \bar{e}_2 \bar{e}_3 \dots \bar{e}_{m-1} \bar{e}_m$ 。

说明 每进行一次分解, 每个分解函数将在原基础上增加一新元素后裂变成两个。要依据分解函数中以原变量形式出现的链路, 生成网络状态的链路利用分组, 对每个分解函数进行 QoS 路由和节点匹配运算, 筛选出分解函数所包含所有 QoS 路由。再把这些路由对应的关联分组进行逻辑迭加, 当运算结果同联络分组所有 1 元素对应匹配时, 分解函数就是一个有效的不变化网络状态, 可停止对该函数继续分解。当所有分解函数与联络分组相匹配时, 便得到有效不变化状态集。再有, 如出现第 1 行元素全为 0, 且分解函数的关联分组迭加运算结果与联络分组不匹配, 则分解函数为无效表达式, 应予以删除。

步骤 4 依据满足 QoS 的有效不变化网络状态集, 按照式(4)和式(3)计算出可靠性指标。

5 正确性论述及算例与实验

5.1 新算法的正确性论述

5.1.1 多约束 QoS 路由算法正确性阐述 式(1)中 $v_{1j}(r) \cdot v_{jm}$ 运算, $j, m = 2, 3, \dots, n$, 所得到的结果是从节点 v_1 出发通过 v_j 接力后达到目的节点 v_m 串联 $r+1$ 条链路的路径。式(1)中的“或逻辑”运算体现的是 v_1 经过不同中间节点形成多条到达其它节点的路径。它涵盖了节点 v_1 通过全部 $n-1$ 个中间节点到目的节点的连接关系, 确保了路径集的全面性。在每次增加链路构成新路径时, 由于链路对应的两端节点, 一个端节点与路径端节点是重合的, 所以按照式(2)计算, 节点利用分组只会有一个元素由 0 变为 1, 新变为 1 的节点就是新增链路后路径的新端节点。如果节点利用分组计算前后没有改变, 显然

增加链路后路径必然存在闭环。所以用式(2)计算, 不仅可以正确标注路径的节点顺序, 而且完全能够正确的判断出路径是否存在闭环问题。可见新算法完全可以获得满足要求的全部路由。

5.1.2 获得不变化网络状态集算法正确性阐述 在计算不变化网络状态集过程中, 既考虑了节点失效对网络的影响, 而且在用链路进行邻接矩阵分解时, 又考虑了获得网络状态的不变化性、全面性和有效性, 体现为: (1)考虑节点 v_i 失效影响是, 不仅在网络拓扑中减少了 v_i 节点, 同时也减少了与 v_i 相连的链路。显然在邻接矩阵中删除相应 v_i 行和 v_i 列后, 就能够正确地表示出节点 v_i 失效后网络拓扑的改变。(2)用链路对邻接矩阵分解和变换运算, 实质是网络拓扑变换的逻辑代数化表现形式。链路只能有 $f_{1i} = e_i$ 和 $\bar{f}_{1i} = \bar{e}_i$ 两个不变化状态, 分解为 e_i 时认为该链路正常, 对应网络拓扑收缩变换, 把 e_i 和它两端节点整体收缩成一个节点, 两端节点到其它节点的链路并归到合并节点上。显然式(5)第 1 部分就是链路 e_i 正常时网络拓扑收缩的结果。分解为 \bar{e}_i 时认为该链路失效, 式(5)第 2 部分令矩阵元素 $v_{1i} = 0$ 就是网络拓扑改变的状态。可见用式(5)对邻接矩阵变换所得到的两种状态, 完全能够正确表述链路状态对网络状态的影响。(3)用式(5)不断分解就能够获得所有不变化网络状态, 确保了全面性。但汇聚节点与每个协作节点间至少存在一条满足 QoS 约束路由才是有效的网络状态。由于用分解函数生成的链路利用分组和路由关联分组, 进行 QoS 路由和节点匹配运算, 只有满足 QoS 要求的分解函数才能最终被保留, 可见新算法得到的不变化网络状态集是全面且有效的。

5.2 算例、实验及结果分析

5.2.1 算例 抽象表示的网络拓扑如图 1 所示。虚线表示以 v_1 为汇聚节点, $\{v_3, v_4, v_5\}$ 为协作节点的分布式紧耦合 VPN。设各节点正常工作时具有代表性的时延抖动统计值均为 1。各链路两个方向的数据包丢失率、带宽和时延等状态参数假设均相同, 统计值分别为: $e_1 = (10^{-7}, 105, 2)$, $e_2 = (10^{-7}, 90, 2)$, $e_3 = (10^{-6}, 90, 5)$, $e_4 = (10^{-6}, 100, 4)$, $e_5 = (10^{-6}, 95, 5)$,

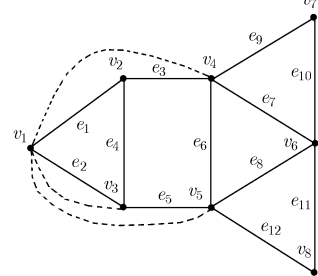


图 1 抽象表示的网络拓扑图

$e_6 = (10^{-6}, 80, 3)$, $e_7 = (10^{-6}, 110, 3)$, $e_8 = (10^{-6}, 100, 5)$, $e_9 = (10^{-6}, 85, 6)$, $e_{10} = (10^{-6}, 60, 7)$, $e_{11} = (10^{-6}, 70, 4)$, $e_{12} = (10^{-6}, 65, 5)$ 。假设各链路的可靠性参数都为 $p_i=0.9$, 各节点可靠性参数都为 $P_c=0.9999$, 求满足时延 ≤ 15 , 时延抖动 ≤ 6 , 丢失率 $\leq 10^{-5}$, 带宽 ≥ 70 等 QoS 参数下分布式紧耦合 VPN 可靠性指标。

步骤 1 计算网络中满足 QoS 要求的路由, 结果列于表 1。

步骤 2~步骤 3 得到各种情况下满足 QoS 的有效不变化网络状态集, 结果列于表 2。

步骤 4 计算可靠性指标。由于各节点可靠性参数以及各链路的可靠性参数相同, 所以得到可靠性指标为

$$PI = (p^4 + 4p^4q + 5p^4q^2 + 2p^5q^2 + 2p^5q^3 + p^3q^2 + p^4q^3)P^6 + (p^3 + 5p^4q + p^4 + 5p^4q^2 + p^3q^2)P^5Q + (p^3)P^4Q^2 = 70.9591\%$$

5.2.2 实验简述和结果分析 如果按照不考虑 QoS 要求的传统分布式网络可靠性方法计算^[9-14], 则得到指标值为 97.6846%。从计算结果上看传统算法远高于新算法, 两者相差 26.7225%, 为此通过实验检验准确性。按照图 1 拓扑用计算机组成物理网络, 为了模拟分布式紧耦合 IPsec VPN 通信过程, 各节点和链路的工作状态、参数等由软件设定。网络的选路原则是满足约束条件的最短路径优先选择, 以反映出实际网络的工作过程。按照算例中的参数统

一由 v_1 节点随机生成节点和链路的失效概率, 并以命令数据包的形式通知其它节点。由 v_1 节点发出命令数据包。其它节点收到命令数据包后, 根据命令动作。由 v_1 节点接收 v_3, v_4, v_5 节点反馈信息进行判断与统计。多次实验得到的统计结果是, 满足 QoS 要求的成功概率为 75.87589%。

通过实验结果可以看出, 新算法计算结果与实验结果非常贴切, 虽然计算值略低于实验值, 但是两者相差不到 5%。而传统算法的计算结果与实验结果相差近 22%。原因是新计算方法不会出现不符合 QoS 约束要求的网络状态, 且为最苛刻情况下的可靠性指标值, 所以计算值略低于实际值。而传统的计算方法, 没考虑 QoS 约束要求, 不加以选择地包含网络拓扑中所有节点和链路, 产生大量冗余不变化网络状态, 所以计算值要比实际值高很多。说明传统的评估方法夸大了分布式紧耦合 VPN 的可靠性。

6 结束语

本文提出的分布式紧耦合 VPN 可靠性计算方法, 不仅可以根据网络的拓扑结构完成一般分布式网络的可靠性计算, 而且由于充分考虑了分布式紧耦合业务提出的 QoS 约束要求, 因此能够更加贴切实际地体现出互连网络承载这些业务所提供的通信保障。再有, 可靠性指标计算方法所有运算过程, 全部实现了代数化或逻辑代数化运算, 因此易于编程实现, 可借助计算机进行分析, 解决了大型网络

表 1 路由及相关计算结果汇总

计算顺序	满足 QoS 路由	路由节点利用标记分组	
		$G[\cdot]^{bit} = [v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8]$	$L[\cdot]^{bit} = [e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}]$
$V_1(1)$	$v_1e_2v_3$	$G[v_1e_2v_3]^{bit} = [1, 0, 1, 0, 0, 0, 0, 0]$	$L[v_1e_2v_3]^{bit} = [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$
	$v_1e_1v_2e_4v_3$	$G[v_1e_1v_2e_4v_3]^{bit} = [1, 1, 1, 0, 0, 0, 0, 0]$	$L[v_1e_1v_2e_4v_3]^{bit} = [1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]$
$V_1(2)$	$v_1e_1v_2e_3v_4$	$G[v_1e_1v_2e_3v_4]^{bit} = [1, 1, 0, 1, 0, 0, 0, 0]$	$L[v_1e_1v_2e_3v_4]^{bit} = [1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]$
	$v_1e_2v_3e_5v_5$	$G[v_1e_2v_3e_5v_5]^{bit} = [1, 0, 1, 0, 1, 0, 0, 0]$	$L[v_1e_2v_3e_5v_5]^{bit} = [0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]$
$V_1(3)$	$v_1e_2v_3e_4v_2e_3v_4$	$G[v_1e_2v_3e_4v_2e_3v_4]^{bit} = [1, 1, 1, 1, 0, 0, 0, 0]$	$L[v_1e_2v_3e_4v_2e_3v_4]^{bit} = [0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0]$
	$v_1e_2v_3e_5v_5e_6v_4$	$G[v_1e_2v_3e_5v_5e_6v_4]^{bit} = [1, 0, 1, 1, 1, 0, 0, 0]$	$L[v_1e_2v_3e_5v_5e_6v_4]^{bit} = [0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0]$
	$v_1e_1v_2e_4v_3e_5v_5$	$G[v_1e_1v_2e_4v_3e_5v_5]^{bit} = [1, 1, 1, 0, 1, 0, 0, 0]$	$L[v_1e_1v_2e_4v_3e_5v_5]^{bit} = [1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0]$
	$v_1e_1v_2e_3v_4e_6v_5$	$G[v_1e_1v_2e_3v_4e_6v_5]^{bit} = [1, 1, 0, 1, 1, 0, 0, 0]$	$L[v_1e_1v_2e_3v_4e_6v_5]^{bit} = [1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0]$
$V_1(4)$	$v_1e_1v_2e_3v_4e_6v_5v_3$	$G[v_1e_1v_2e_3v_4e_6v_5v_3]^{bit} = [1, 1, 1, 1, 1, 0, 0, 0]$	$L[v_1e_1v_2e_3v_4e_6v_5v_3]^{bit} = [1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0]$
	$v_1e_1v_2e_4v_3e_5v_5e_6v_4$	$G[v_1e_1v_2e_4v_3e_5v_5e_6v_4]^{bit} = [1, 1, 1, 1, 1, 0, 0, 0]$	$L[v_1e_1v_2e_4v_3e_5v_5e_6v_4]^{bit} = [1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0]$
	$v_1e_2v_3e_5v_5e_8v_6v_7v_4$	$G[v_1e_2v_3e_5v_5e_8v_6v_7v_4]^{bit} = [1, 0, 1, 1, 1, 1, 0, 0]$	$L[v_1e_2v_3e_5v_5e_8v_6v_7v_4]^{bit} = [0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0]$
	$v_1e_2v_3e_4v_2e_3v_4e_6v_5$	$G[v_1e_2v_3e_4v_2e_3v_4e_6v_5]^{bit} = [1, 1, 1, 1, 1, 0, 0, 0]$	$L[v_1e_2v_3e_4v_2e_3v_4e_6v_5]^{bit} = [0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0]$
$v_1e_1v_2e_3v_4e_7v_6e_8v_5$	$G[v_1e_1v_2e_3v_4e_7v_6e_8v_5]^{bit} = [1, 1, 0, 1, 1, 1, 0, 0]$	$L[v_1e_1v_2e_3v_4e_7v_6e_8v_5]^{bit} = [1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0]$	
$V_1(5), V_1(6), V_1(7)$	0	0	0

表 2 满足 QoS 要求的不变化网络状态集

节点组合	由链路分解的不变化网络状态
$v_1 v_2 v_3 v_4 v_5 v_6$	$e_1 e_2 e_3 e_5, e_1 \bar{e}_2 e_4 e_3 e_5, e_1 e_2 e_3 \bar{e}_5 e_6, e_1 \bar{e}_2 e_4 e_3 \bar{e}_5 e_6, e_1 e_2 e_3 \bar{e}_5 \bar{e}_6 e_7 e_8, e_1 \bar{e}_2 e_4 e_3 \bar{e}_5 \bar{e}_6 e_7 e_8, e_1 e_2 \bar{e}_3 e_5 e_6, e_1 \bar{e}_2 e_4 \bar{e}_3 e_5 e_6, e_1 e_2 \bar{e}_3 \bar{e}_5 e_6 e_7, e_1 \bar{e}_2 e_4 \bar{e}_3 \bar{e}_5 e_6 e_7, e_1 e_2 \bar{e}_3 \bar{e}_5 e_6 e_7 e_8, e_1 \bar{e}_2 e_4 \bar{e}_3 \bar{e}_5 e_6 e_7 e_8,$ $e_1 \bar{e}_2 e_4 e_3 e_6 e_5, \bar{e}_1 e_2 e_4 e_3 e_5, \bar{e}_1 e_2 e_4 e_3 \bar{e}_5 e_6, \bar{e}_1 e_2 e_4 \bar{e}_3 e_5 e_6, \bar{e}_1 e_2 e_4 \bar{e}_3 e_5 \bar{e}_6 e_7, \bar{e}_1 e_2 e_4 e_3 e_5, \bar{e}_1 e_2 e_4 e_3 \bar{e}_5 e_6 e_7$
$v_1 \bar{v}_2 v_3 v_4 v_5 v_6$	$e_2 e_5 e_6, e_2 e_5 \bar{e}_6 e_8 e_7$
$v_1 v_2 v_3 v_4 v_5 \bar{v}_6$	$e_1 e_2 e_3 e_5, e_1 e_2 e_3 \bar{e}_5 e_6, e_1 \bar{e}_2 e_4 e_3 e_5, e_1 \bar{e}_2 e_4 e_3 \bar{e}_5 e_6, e_1 e_2 \bar{e}_3 e_5 e_6, e_1 \bar{e}_2 e_4 \bar{e}_3 e_5 e_6,$ $e_1 \bar{e}_2 \bar{e}_4 e_3 e_6 e_5, \bar{e}_1 e_2 e_4 e_3 e_5, \bar{e}_1 e_2 e_4 e_3 \bar{e}_5 e_6, \bar{e}_1 e_2 e_4 \bar{e}_3 e_5 e_6, \bar{e}_1 e_2 e_4 \bar{e}_3 e_5 e_6$
$v_1 \bar{v}_2 v_3 v_4 v_5 \bar{v}_6$	$e_2 e_5 e_6$

人工计算的繁琐及复杂性问题。不仅从原理上阐述了新算法的正确性，而且举例详细计算了构建在互联网之上，具有 QoS 约束要求紧耦合分布式 VPN 网络可靠性指标，通过实验数据，并且与传统分布式网络可靠性指标计算结果进行分析和对比，说明利用新计算方法可以从克服传统计算方法的不足，得到的结果更加合理。为评估构建在互联网平台上的分布式紧耦合虚拟专用网可靠性，提供了理论和方法上的支撑。

参 考 文 献

[1] Ivancic W D and Paulsen E P. Use of virtual mission operations center technology to Achieve JPDO's virtual tower vision[C]. 2006 IEEE Integrated Communication, Navigation and Surveillance Conference and Workshop, Baltimore, May 1-3, 2006: 273-280.

[2] 李频, 唐家益, 陈丹伟, 等. 虚拟专用网分类和比较研究[J]. 计算机工程, 2006, 32(22): 133-135.

Li P, Tang J Y, and Chen D W, et al. Study on classification and comparison of virtual private network[J]. *Computer Engineering*, 2006, 32(22): 133-135.

[3] Gebre B A and Ramirez-Marquez J E. Element substitution algorithm for general two-terminal network reliability analyses[J]. *IIE Transactions*, 2007, 39(3): 265-275.

[4] Satitsatian S and Kapur K C. An algorithm for lower reliability bounds of multistate two-terminal networks[J]. *IEEE Transactions on Reliability*, 2006, 55(2): 199-206.

[5] Hardy G, Lucet C, and Limnios N. K-terminal network reliability measures with binary decision diagrams[J]. *IEEE Transactions on Reliability*, 2007, 56(3): 506-515.

[6] Younes A and Girgis M R. A tool for computing computer network reliability[J]. *International Journal of Computer Mathematics*, 2005, 82(12): 1455-1465.

[7] Hui K P, Bean N, and Kraetzl M, et al. The cross-entropy method for network reliability estimation. *Annals of Operations Research*[J]. 2005, 134(1): 101-118.

[8] 戴伏生. 通信网络可靠性指标的新定义及计算方法[J]. 系统工程与电子技术, 2006, 28(11): 1641-1647.

Dai F S. New definition of reliability index and algorithm for a communication network[J]. *Systems Engineering and Electronics*, 2006, 28(11): 1641-1647.

[9] Chen D J and Huang T H. Reliability analysis of distributed systems based on a fast reliability algorithm[J]. *IEEE Transactions on Parallel and Distributed Systems*, 1992, 3(2): 139-154.

[10] Chen D J, Chen R S, and Huang T H. A heuristic approach to generating file spanning trees for reliability analysis of distributed computing systems[J]. *Computers and Mathematics with Applications*, 1997, 34(10): 115-131.

[11] Ke W J and Wang S D. Reliability evaluation for distributed computing networks with imperfect nodes[J]. *IEEE Transactions on Reliability*, 1997, 46(3): 342-349.

[12] Lin M S and Chen D J. The computational complexity of the reliability problem on distributed systems[J]. *Information Processing Letters*, 1997, 64(3): 143-147.

[13] 胡华平, 金士尧. 分布式系统可靠模型[J]. 计算机工程与应用, 1999, 35(8): 1-3.

Hu H P and Jin S Y. Reliability model of distributed systems[J]. *Computer Engineering and Applications*, 1999, 35(8): 1-3.

[14] 孙艳蕊, 张祥德. 一种计算具有不可靠结点分布式计算网络可靠性的算法[J]. 通信学报, 2002, 23(9): 22-28.

Sun Y R and Zhang X D. An algorithm for reliability evaluating of distributed computing networks[J]. *Journal of China Institute of Communications*, 2002, 23(9): 22-28.

戴伏生: 男, 1963 年生, 教授, 研究方向为通信网与通信电子系统.

刘功亮: 男, 1979 年生, 讲师, 研究方向为通信网与卫星通信系统.