

基于网络状态图模型的分簇式路由收敛方法

潘成胜¹, 陆煌杰¹, 石怀峰^{1,2}, 王英植¹

(1. 南京信息工程大学复杂环境智能保障技术教育部重点实验室, 江苏 南京 210044;

2. 国防科技大学第六十三研究所, 江苏 南京 210007)

摘要: 针对强对抗高机动作战条件下战术通信网络中的节点连接关系时变、链路故障频发, 导致路由收敛频繁、业务有效传输时间短的问题, 提出了一种基于网络状态图模型的分簇式路由收敛方法 (OSPF-CSG)。首先, 依据 Hello 报文统计特征获取链路状态, 基于链路状态聚合特征刻画节点状态, 构建网络状态图模型。其次, 改进邻居状态机状态转移触发条件, 构建 2 种新型 OSPF 报文, 并据此设计完全邻接关系主动/被动关联机制。最后, 在此基础上, 设计簇头自举算法、节点入簇算法和簇间连接算法实现基于簇的路由收敛。仿真结果表明, 在 8 种不同规模的网络拓扑中, 针对不同链路损毁程度, OSPF-CSG 相较于传统 OSPF 协议的路由收敛算法, 其路由收敛总次数平均减少 82%, 路由开销平均降低 70%, 分组投递率平均提升 60%。

关键词: 网络状态图; 节点分簇; 路由收敛; OSPF 协议

中图分类号: TN92

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2024267

Cluster-based route convergence method based on network state graph model

PAN Chengsheng¹, LU Huangjie¹, SHI Huaifeng^{1,2}, WANG Yingzhi¹

1. Key Laboratory of Intelligent Support Technology for Complex Environment, Ministry of Education, Nanjing University of Information Science and Technology, Nanjing 210044, China

2. 63rd Research Institute of National University of Defense Technology, Nanjing 210007, China

Abstract: To address the challenges of time-varying node connectivity and frequent link failures in tactical communication networks under conditions of strong adversarial and high-mobility operations, which result in frequent route convergence and short effective transmission times, a cluster-based route convergence method based on network state graph model (OSPF-CSG) was proposed. Firstly, link states were obtained based on the statistical characteristics of Hello messages, and node states were characterized using aggregated link state features to construct the network state graph model. Secondly, the neighbor state machine's state transition triggers were improved, and two new OSPF message types were developed to design an active/passive association mechanism for complete adjacency relationships. Finally, on this basis, a cluster head bootstrap algorithm, node clustering algorithm, and inter-cluster connection algorithm were proposed to achieve cluster-based routing convergence. Simulation results show that, in eight different network topologies and under different link damage conditions, OSPF-CSG achieves an average reduction of 82% in the total number of route convergences, a reduction in routing overhead by 70% on average, and an improvement in packet delivery rate by 60% on average compared to the traditional OSPF protocol's route convergence algorithm.

Keywords: network state graph, node clustering, route convergence, OSPF protocol

收稿日期: 2024-07-10; 修回日期: 2024-12-05

通信作者: 潘成胜, 003150@nuist.edu.cn

基金项目: 国家自然科学基金资助项目 (No.61931004, No.B2202401)

Foundation Items: The National Natural Science Foundation of China (No.61931004, No.B2202401)

0 引言

近年来,美军相继提出了“多域战”“马赛克战”等新型作战方式^[1],促使战争形态从网络中心战向智能化作战转变^[2-3]。智能化作战要求战术通信网络具备广域分布、无缝链接和动态开放的基本能力,以支撑作战要素的跨域自适应组合,从而取得体系对抗优势。在此背景下,作战单元的高机动性以及战场环境的强对抗性,使节点连接关系更加时变,链路故障更加频发,致使路由收敛更为频繁,过度占用网络资源,业务有效传输时间变短。因此,在兼顾路由开销(RO, routing overhead)的同时,如何提高战术通信网络路由收敛效率是一个亟须解决的问题。

开放最短路径优先(OSPF, open shortest path first)协议是一种被广泛应用于战术通信网络路由收敛的基础协议^[4-7],其固定的路由收敛机制(如故障检测机制、链路状态通告(LSA, link status announcement)泛洪机制等)在面对节点连接关系时变时,往往难以实现快速收敛。为提高路由收敛效率,Katz等提出了双向转发检测(BFD, bi-directional forwarding detection)协议,将BFD作为一种OSPF路由收敛过程中的辅助机制,加速其故障检测速度,在BFD加持下可将收敛速度提升到毫秒级^[8-10]。然而,在复杂战场的电磁和气象环境中,链路故障频发极易导致BFD报文的频繁丢失,从而引起虚假报警,严重影响战术通信网络质量。为避免引起虚假报警,有学者提出了OSPF+协议,通过在邻居状态机中增加“Leaving”状态,改进卫星网络下高时延链路中Hello包因时延造成的虚假报警问题^[11-13]。然而,该方案的故障检测时间会进一步增加,进而导致业务有效传输时间变短。为增加业务有效传输时间,大量学者提出利用网络拓扑的中心性特征优化OSPF协议的收敛时间,通过调整参数和权重设置减少故障恢复时间,实现路由快速收敛,增加业务有效传输时间^[14-18]。然而,由于节点连接关系的动态变化以及链路故障的频繁发生,网络拓扑的中心性特征随之波动,进而呈现出无中心性的特征,使这种优化策略难以有效实施,导致在实际应用中,路由收敛时间未能得到实质性改善。

战术通信网络具有网络拓扑动态、全分布式和无中心性等特征,因此可以被视为一种特殊的Ad Hoc

网络^[19-20]。针对此类网络,OSPF协议的扩展协议(OSPF-MPR, OSPF-multipoint relay)、(OSPF-OR, OSPF-overlapping relay)和(OSPF-MDR, OSPF with MANET designated routers)被提出^[21-24]。然而,随着网络规模的不断扩大,这些协议在链路故障发生时,仍未能有效缓解由于LSA泛洪机制引发的路由开销的指数级增长。战场环境中频繁的链路故障将进一步加剧这一问题,导致路由开销激增,严重消耗网络资源。为此,针对Ad Hoc网络拓扑动态、全分布式和无中心性的特征,大量研究提出了分簇算法。文献[25-26]通过结合基于地理位置的分簇算法,优化无线传感器网络的能量效率和网络寿命。文献[27-28]提出了一种基于节点ID的分簇算法,提高移动自组网的能量效率和安全性。文献[29-31]基于节点能量、移动性和度等特性,结合遗传算法实现MANET的节点分簇。Husnain等^[32]引入了鲸优化算法用于选择最佳簇首,并基于智能和能力进行计算和评估,以提升簇优化性能。Asaamoning等^[33]引入了一种新的动态分簇机制,融合政治优化器和Shannon熵函数,旨在实现数据包的高效传播,确保良好的可靠性和可扩展性。Khedr等^[34]介绍了MWCERSF(mobility-based weighted cluster routing scheme for FANET),利用麻雀搜索算法(SSA, sparrow search algorithm)进行初始节点集群建立,并利用多标准决策的天际线运算符优化簇首选择和下一跳路由,提高数据传输效率,保持集群更新。Hosseinzadeh等^[35]提出了QSCR(Q-learning-based smart clustering routing)方法,根据速度相似性确定Hello间隔,采用自适应分簇,基于merit值选择簇头,最后使用贪婪算法实现网络数据的路由。Khedr等^[36]提出了一种新的网络分簇技术HMGOC(hybrid mountain gazelle optimizer to cluster-based data routing),通过条件贝叶斯定理设计了适应网络变化的路由机制,提高负载均衡、降低时延、增强网络吞吐量。上述技术均采用了分簇技术,通过分簇有效降低路由开销,大幅提升业务有效传输时间。在此基础上,上述文献利用网络状态评估与建模、仿生算法、智能算法等手段优化传统网络分簇技术,进一步提升了Ad Hoc网络的综合性能。然而,上述分簇技术通常仅运用于无人机、车联网、传感器等网络环境,这些场景中节点距离较近且系统组成单一。

相比之下, 战术通信网络中异构链路类型达到 10 种以上, 涵盖区宽、卫星、有线等多种传输方式, 且复杂战场环境下, 存在节点连接关系时变、链路故障频发。在此类环境下, 现有的分簇机制面临极大挑战, 难以适用于强对抗和高机动的战场环境。此外, 传统 OSPF 协议在路由收敛过程中无法实现网络中的节点分簇功能, 并且也并不具备网络状态评估与建模、仿生算法、智能算法等分簇优化技术部署的能力。

因此, 为应对节点连接关系时变和链路故障频发的强对抗战场网络环境, 本文提出了一种基于网络状态图模型的分簇式路由收敛方法 (OSPF-CSG)。OSPF-CSG 通过将分簇技术与传统 OSPF 协议有效融合, 并结合网络状态图模型进一步强化分簇技术的优势, 大幅提升战场环境中的路由收敛效率和网络的鲁棒性, 具体如下。

1) 依据 Hello 报文的统计特征获取链路状态, 依据链路状态的聚合特征刻画节点状态, 构建网络状态图模型。该过程能够捕获描述战场环境下节点连接关系的实时变化。

2) 改进邻居状态机状态转移触发条件, 构建 2 种新型 OSPF 报文, 据此设计完全邻接关系主动/被动关联机制。该机制确保了网络状态图模型信息的有效传递, 促进节点间邻接关系的快速建立, 从而支

撑分簇算法在复杂战场环境中的敏捷部署。

3) 提出簇头自举算法、节点入簇算法和簇间连接算法。这些算法可减少强对抗和高机动战场环境中故障链路的收敛次数, 提高战术通信网络中优质链路的收敛效率, 从而减少路由收敛总次数, 降低路由开销, 增强整体的路由收敛性能, 进而延长战场环境下的业务有效传输时间。

1 OSPF-CSG 设计与实现

1.1 OSPF-CSG 概述

OSPF 利用链路状态信息构建网络拓扑, 采用 Dijkstra 算法计算最短路径, 确保数据包正确传输。当网络拓扑发生变化时, 节点利用 Hello 报文感知变化, 生成新 LSA 并泛洪给其他节点, 确保网络中所有节点具备统一的路由表, 该过程称为路由收敛。OSPF-CSG 在继承 OSPF 协议基本功能的基础上, 重构了结构, 如图 1 所示。

具体改进如下。

1) 构建网络状态图模型, 对战术通信网络当前状态进行评估和建模, 为分簇算法提供运行所需的数据支持。

2) 设计完全邻接关系主动/被动关联机制, 该机制通过构造 2 种新型 OSPF 报文并改进邻居状态机状态转移触发条件, 以传递网络状态图模型并引

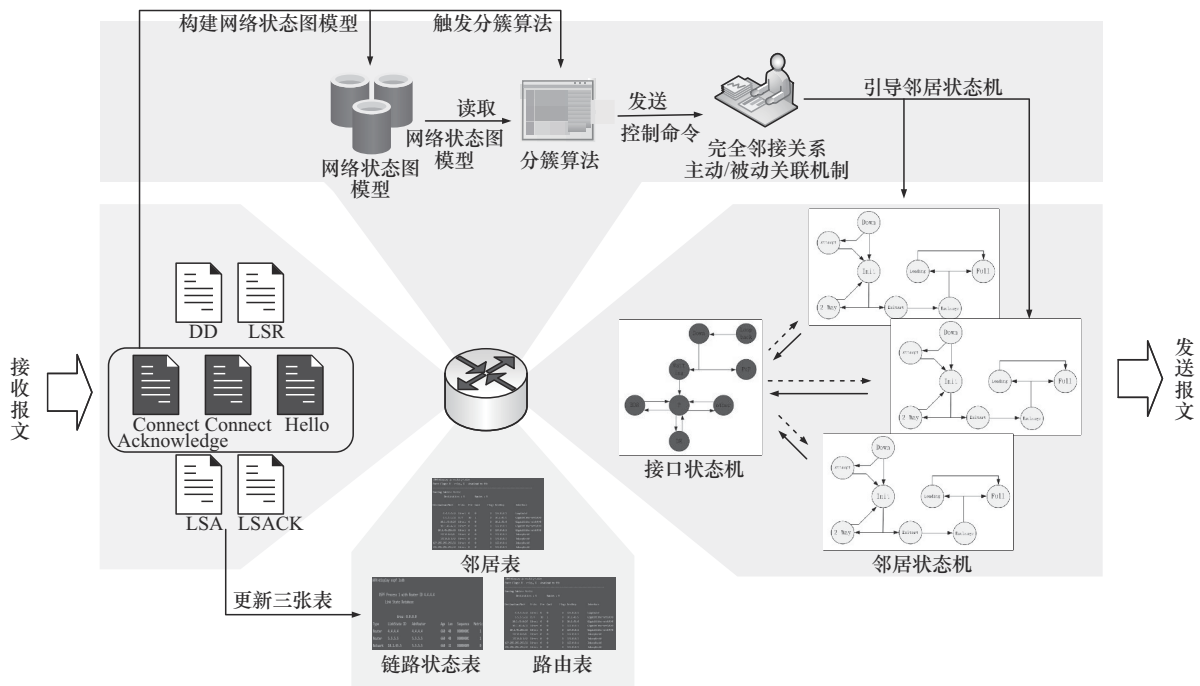


图 1 OSPF-CSG 结构

导邻居状态机建立完全邻接关系。两机制在分簇算法的控制下协同运作、相辅相成，为分簇算法的敏捷部署提供了基础支撑。

3) 提出分簇算法，该算法分为簇头自举、节点入簇和簇间连接 3 个子算法，最终实现只有簇头节点与其簇内节点之间的邻居状态机建立完全邻接关系，以及指定的簇间节点之间的邻居状态机建立完全邻接关系。

1.2 网络状态图模型构建

战术通信网络拓扑可以被视为一个有向图 $G = (N, E)$ ，其中 N 是网络节点的集合， E 是通信链路的集合。假设节点 $u \in N$ ， $v \in N$ ，且 u 和 v 之间存在一条链路 $l (l \in E)$ ，则可将该链路记为 $l = uv$ 。那么，以有向图 G 中任意节点 v 为中心构建的子图，可用如下数学定义进行描述。

定义 1 定义 $D(v)$ 为一阶子图。 N' 为与节点 v 直接相连的邻居节点集合，表示为

$$N' = \{ u \in V | vu \in E \} \quad (1)$$

E' 为与节点 v 直接相连的所有通信链路集合，表示为

$$E' = \{ e = vu | u \in N' \} \quad (2)$$

则 $D(v)$ 可表示为

$$D(v) = (N', E') \quad (3)$$

定义 2 定义 $M(v)$ 为二阶子图。 N_2 为与 $D(v)$ 内部节点直接相连的节点集合，表示为

$$N_2 = \{ u | \exists w \in N', \exists u \in N \setminus N', vw \in E, wu \in E \} \quad (4)$$

E_2 为 $D(v)$ 与 N_2 之间直接相连的通信链路集合，表示为

$$E_2 = \{ e = wu | \exists w \in N', \exists u \in N_2, wu \in E \} \quad (5)$$

则 $M(v)$ 可表示为

$$M(v) = (N' \cup N_2, E' \cup E_2) = (N'', E'') \quad (6)$$

在复杂的战场环境下，战术通信网络拓扑存在高动态、弱连接，因此需要对战术通信网络中的链路状态和节点状态进行评估，进而构建网络状态图模型。首先，依据评估周期内链路 l 中 Hello 包的缺失情况，评估链路状态，如式(7)所示。

$$\omega_1(uv) = \frac{T - t_d}{T} \quad (7)$$

其中， $\omega_1(uv)$ 表示链路状态评估函数， T 表示评估周期， t_d 表示在评估周期内，缺失 Hello 包所构成的时间片段，即在预计下一接收 Hello 包的时间与

实际最后接收时间之间的差值，具体链路状态如图 2 所示。

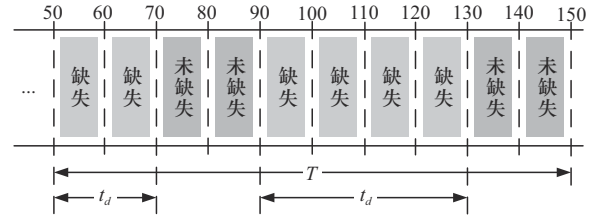


图2 链路状态

其次，依据节点 v 所有接口上链路状态聚合后的平均值，评估节点状态，如式(8)所示。

$$\omega_2(v) = \frac{\sum_{uv \in E'} \omega_1(uv)}{|E'|} \quad (8)$$

其中， $\omega_2(v)$ 表示节点状态评估函数，反映了节点 v 的整体健康状况与可靠性。此函数综合考虑了与节点相关联的链路状态，能够有效反映节点在复杂战场环境中的潜在损毁风险及其高机动性所带来的不确定性。 $|E'|$ 表示 E' 的基数，即与节点 v 相关联的链路条数。

节点状态的值通过线性映射至 Hello 报文的 Priority 字段进行传递^[4-6]。一方面，能够优化基于 Priority 字段的指定路由器 (DR, designated router) 选举过程，使具有更高节点状态的节点优先当选 DR 或备份 DR，从而确保网络中关键角色状态更优的节点担任，提升网络的稳定性与路由收敛效率。另一方面，与引入额外的信令或控制协议相比，该方法充分利用了现有 OSPF 协议的通信机制，不需要额外复杂性或路由开销。

基于 Hello 包周期性广播实现高效链路和节点状态感知的方式，可快速反映战场环境下的网络拓扑变化，其低带宽占用和可控频率的特性能够确保通信开销最小化。同时，该机制凭借短周期快速更新能力，可有效应对强对抗和高机动的复杂战场环境，大幅提升对网络连通性与通信质量的感知能力，为网络状态图模型的构建提供技术支撑。

在此基础上，对 $D(v)$ 和 $M(v)$ 的链路和节点状态进行评估，据此得到的网络状态图模型，定义如下。

定义 3 定义 $S(v)$ 为一阶邻居状态图模型。 $S(v)$ 为对 $D(v)$ 中的各节点 $n'_i (n'_i \in N')$ 通过 ω_2 得到

网络状态图模型,以评估邻居节点的节点状态,进而实现簇头自举算法,如式(9)所示。

$$S(v) = (N', E', \omega_2) \quad (9)$$

定义 4 定义 $U(v)$ 为一阶无权邻域状态图模型。 $U(v)$ 为对 $S(v)$ 中的各链路 $e'_j (e'_j \in E')$ 通过 ω_1 计算链路状态得到的网络状态图模型,如式(10)所示。

$$U(v) = (N', E', \omega_1, \omega_2) \quad (10)$$

定义 5 定义 $X(v)$ 为二阶无权邻域状态图模型。 $X(v)$ 为对二阶子图 $M(v)$ 的各链路 $e''_j (e''_j \in E'')$ 和各节点 $n''_i (n''_i \in N'')$ 通过 ω_1 和 ω_2 计算链路和节点状态得到网络状态图模型,如式(11)所示。

$$X(v) = (N'', E'', \omega_1, \omega_2) \quad (11)$$

在此基础上,对 $U(v)$ 和 $X(v)$ 进行赋权求和,以评估节点 v 到达目的节点所经过路径的状态,据此得到的赋权图模型,定义如下。

定义 6 定义 $U'(v)$ 为一阶赋权邻域状态图模型。 $U'(v)$ 为对 $U(v)$ 中节点 v 到达邻居节点 $u_i \in N'$ 路径上的 ω_1 和 ω_2 赋权求和,形成赋权图模型。该模型将综合评估节点 v 到节点 u_i 路径上的链路和节点状态值,进而依据该模型实现节点入簇算法和簇间连接算法,如式(12)所示。

$$U'(v) = (N', E', \omega) \quad (12)$$

其中, ω 表示赋权值,可表示为 $\omega(v, u_i) = \alpha\omega_1(vu_i) + \beta\omega_2(u_i)$, $vu_i \in E'$, $u_i \in N'$, α 和 β 分别表示对 ω_1 和 ω_2 不同的参考程度,这里预设 $\alpha = 0.5$, $\beta = 0.5$ 。状态图赋权示意如图 3 所示, v 到 u_1 路径上的赋权值 ω 为

$$\begin{aligned} \omega(v, u_1) &= \alpha\omega_1(vu_1) + \beta\omega_2(u_1) = \\ &0.5 \times 0.2 + 0.5 \times 0.4 = 0.3 \end{aligned} \quad (13)$$

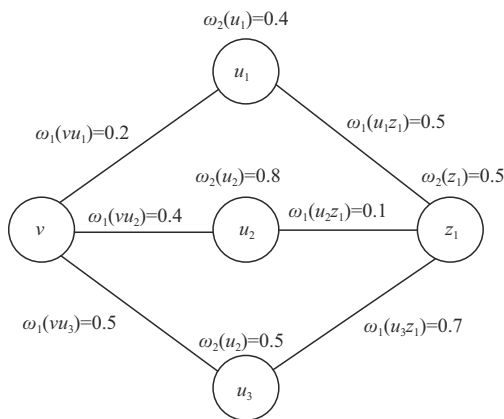


图 3 状态图赋权示意

定义 7 定义 $X'(v)$ 为二阶赋权邻域状态图模型。 $X'(v)$ 为对 $X(v)$ 中节点 v 经过中间节点 $u_i \in N'$ 到达目的节点 $z_i \in N_2$ 路径上所有的 ω_1 和 ω_2 进行赋权求和,构成的赋权图模型。该模型将综合评估节点 v 经过节点 u_i 到达节点 z_i 路径上的链路和节点状态值,进而依据该模型实现簇间连接算法,如式(14)所示。

$$X'(v) = (N'', E'', v) \quad (14)$$

其中, v 表示赋权值,可表示为 $v(v, u_i, z_j)$,即节点 v 经过节点 $u_i \in N'$ 到节点 $z_j \in N_2$ 路径上所有 ω_1 和 ω_2 的赋权求和值,表示为

$$\begin{aligned} v(v, u_i, z_i) &= \theta\omega_1(vu_i) + \lambda\omega_2(u_i) + \\ &\iota\omega_1(u_i z_j) + \kappa\omega_2(z_j) \end{aligned} \quad (15)$$

其中, θ 、 λ 、 ι 和 κ 表示对链路和节点状态的不同参考程度,这里预设 $\theta = 0.5$ 、 $\lambda = 0.5$ 、 $\iota = 0.5$ 和 $\kappa = 0.5$ 。那么,图 3 中节点 v 经过节点 u_1 到达节点 z_1 之间路径的赋权值 v 为

$$\begin{aligned} v(v, u_1, z_1) &= \theta\omega_1(vu_1) + \lambda\omega_2(u_1) + \\ &\iota\omega_1(u_1 z_1) + \kappa\omega_2(z_1) = 0.8 \end{aligned} \quad (16)$$

1.3 完全邻接关系主动/被动关联机制

图 4 为传统 OSPF 协议邻居状态机状态转移过程,包括多种状态和状态转移事件,整体过程由邻居关系建立过程和完全邻接关系建立过程构成。其中,邻居关系建立过程实现邻居发现,完全邻接关系建立过程完成两端节点路由信息的同步和 LSA 泛洪,进而实现路由收敛。二者由 2-Way Reveived 或 AdjOK? 事件实现状态转移,当状态转移到 Ex-Start 时,状态机开始构建完全邻接关系,反之,则退出完全邻接关系的构建。两事件中若本节点为 DR 或 BDR、对端节点为 DR 或 BDR 等条件其中之一时,才能使状态转移至 ExStart,进而触发完全邻接关系的建立^[6]。然而,上述过程无法依据节点分簇算法引导邻居状态机建立完全邻接关系,进而无法实现分簇式路由收敛。因此,本文设计了完全邻接关系主动/被动关联机制,以实现分簇式路由收敛。该机制在 2-Way Reveived 和 AdjOK? 事件中新增判断条件 (NJC, new judgement condition),以支撑分簇算法的有效嵌入,并构建 Connect 和 ConnectAck 这 2 种 OSPF 报文类型,以传递网络状态图模型和连接请求信息。其中,Connect 报文具备入簇请求、一阶赋权邻域状态图模型传输、簇间连接指令和簇间连接请求 4 种功能。ConnectAck 报文用于确认所接收的 Connect 报文。

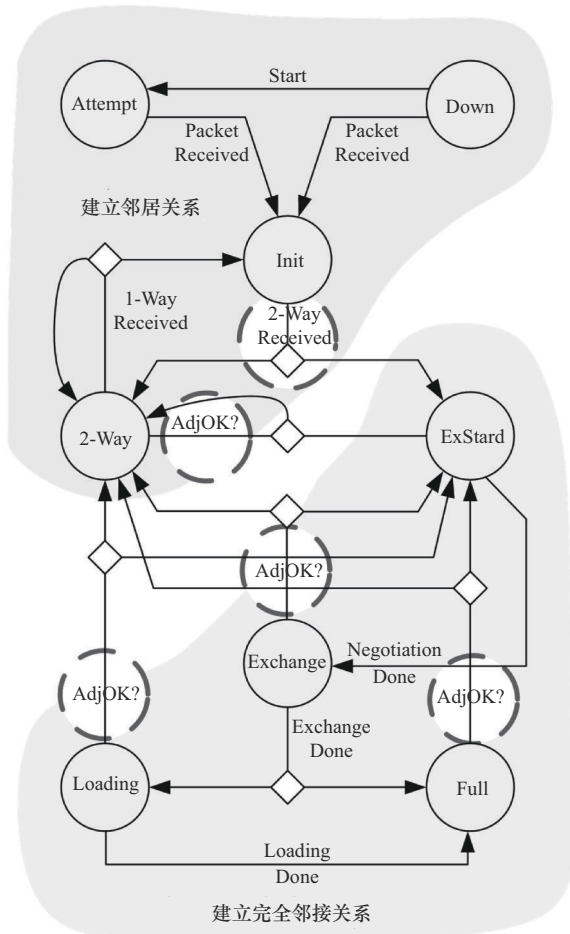


图4 传统OSPF协议邻居状态机状态转移过程

完全邻接关系主动关联机制和完全邻接关系被动关联机制分别应用于节点入簇算法和簇间连接算法。节点入簇算法在确定目标簇后，完全邻接关系激活主动关联机制，并发送入簇请求，以建立簇内的完全邻接关系。在入簇完成后，该机制继续传输一阶赋权邻域状态图模型，使簇头能够实时监测簇内节点及其邻居节点的动态变化。与此同时，簇间连接算法利用簇内节点提供的一

阶赋权邻域状态图模型，识别簇间连接的节点，并启用完全邻接关系被动关联机制，通知簇内节点与相关的簇间节点完成完全邻接关系的建立。两机制在分簇算法的控制下协同运作、相辅相成，最终实现了分簇式路由收敛。具体过程如下。

图5为完全邻接关系主动关联机制实现过程。此时，两端节点由于NJC不成立，邻居状态机状态均为2-Way。簇内节点R1会向簇头节点R2发送Connect报文（入簇请求）。R2接收到Connect报文后其NJC成立，开始建立完全邻接关系并发送ConnectAck报文给R1。R1接收到ConnectAck报文后，其NJC也成立，双方开始建立完全邻接关系，从而实现节点自适应入簇。具体报文内容不为本文重点，不具体描述。

图6为完全邻接关系被动关联机制实现过程。当簇内节点R1完成节点入簇后，即与簇头节点R2建立完全邻接关系后，R1会通过Connect报文（一阶赋权邻域状态图模型传输）传递并不断更新其一阶赋权邻域状态图模型。R2依据该报文，构建二阶赋权邻域状态图模型。在此基础上，R2由簇间连接算法计算要连接的簇间节点，接着发送Connect报文（簇间连接指令）命令簇内节点与相应的簇间节点R3实现连接。此时，簇间节点R1和R3由于NJC不成立，邻居状态机状态均为2-Way。R1接收到指示后，向R3发送Connect报文（簇间连接请求）。R3接收到Connect报文后其NJC成立，开始建立完全邻接关系并发送ConnectAck报文给R1。R1接收到ConnectAck报文后，其NJC也成立，双方开始建立完全邻接关系，从而实现簇间连接算法。具体报文内容不为本文重点，不具体描述。

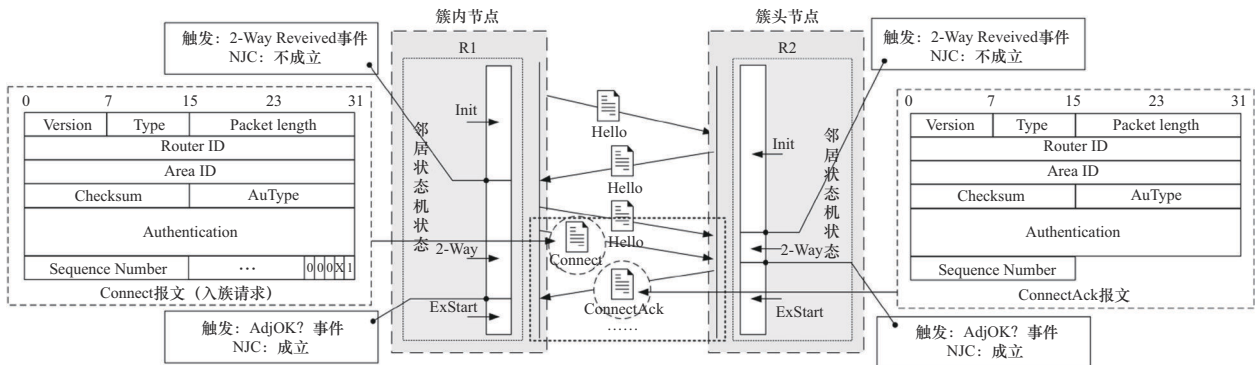


图5 完全邻接关系主动关联机制实现过程

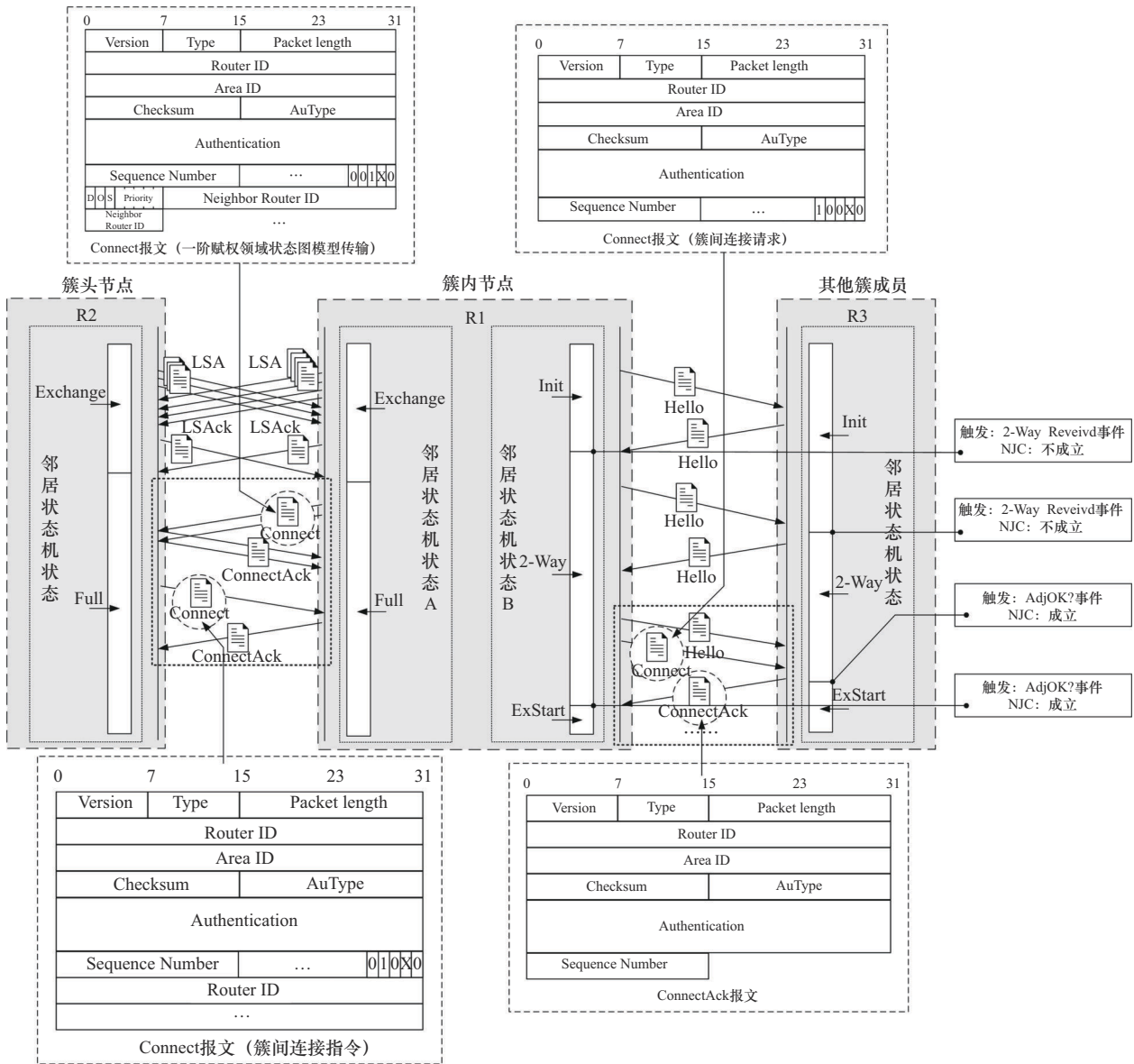


图6 完全邻接关系被动关联机制实现过程

OSPF-CSG 的完全邻接关系主动/被动关联机制通过引入额外的 Connect 报文和 ConnectAck 报文,增加了协议开销,从而实现分簇式路由收敛。此机制虽增加了少量新增报文的开销,但是可有效降低因频繁链路故障引发的路由收敛次数,进而遏制路由由开销因链路故障频发而呈指数级增长的趋势,在总体上大幅减少了路由开销。

1.4 分簇算法设计

本文分簇算法分为簇头自举、节点入簇和簇间连接算法3个子算法,具体步骤如下。

- 1) 节点通过簇头自举算法确定其是否担任簇头。
- 2) 执行节点入簇算法,完成簇内节点的路由

收敛。

3) 簇头节点在簇内节点提供的一阶赋权邻域状态图模型基础上,通过簇间连接算法建立并维护簇间连接,最终实现分簇式路由收敛。

4) 循环执行步骤1)~步骤3),建立并维护网络分簇结构,直至仿真结束。

算法1 簇头自举算法

输入 节点 v 当前担任角色 R (初始状态/簇头/簇内节点), 节点 v 的节点状态 $\omega_2(v)$, 一阶邻居状态图模型 $S(v)$

输出 下一时刻节点担任的角色 role

- 1) 初始化 $G_\omega = (N', E', \omega_2)$

- 2) if R 为初始状态 then
- 3) 构建 $S(v)$ 中节点为簇头的子图, 将其存储到 G_ω
- 4) else if R 为簇头 then
- 5) 构建 $S(v)$ 中节点为簇头的子图, 将其存储到 G_ω
- 6) else
- 7) 构建 $S(v)$ 中节点为簇头和簇内节点子图, 将其存储到 G_ω
- 8) end if
- 9) if G_ω 为空 then
- 10) role 为簇头
- 11) else
- 12) 构建 G_ω 中权值最大的子图 G_{\max}
- 13) if $\forall u \in G_{\max}, \omega_2(v) > \omega_2(u)$ then
- 14) role 为簇头
- 15) else if $\forall u \in G_{\max}, \omega_2(v) < \omega_2(u)$ then
- 16) role 为簇内节点
- 17) else
- 18) if $\forall u \in G_{\max}, v$ 的 Router ID $> u$ 的 Router ID then
- 19) role 为簇头
- 20) else
- 21) role 为簇内节点
- 22) end if
- 23) end if
- 24) end if

算法 1 用于计算下一时刻节点担任的角色 (簇头节点或簇内节点), 具体如下。节点依据担任的角色 R 和一阶邻居状态图模型 $S(v)$ 构建新的一阶邻居状态图模型 G_ω 。在此基础上, 节点根据自身的节点状态 $\omega_2(v)$ 和邻居节点的 Router ID 计算出下一时刻节点所要担任的角色。算法 1 由一系列条件判断模块构成, 其核心执行过程中涉及对既有图模型的完整遍历。因此, 该算法的时间复杂度可表述为 $O(n)$, 其中 n 表示图中节点的总数。

算法 2 节点入簇算法

输入 节点 v 担任角色 R , 一阶无权邻域状态图模型 $U(v)$

输出 要连接簇头的 Router ID: id, 一阶赋权邻域状态图模型 $U'(v)$

- 1) 初始化 $G_\omega = (N', E', \omega_1, \omega_2)$

- 2) if R 为簇内节点 then
- 3) if 节点 v 未与相应簇头节点建立完全邻接关系 then
- 4) 构建 $U(v)$ 中节点为簇头且已与节点 v 建立完全邻接关系的子图, 将其存储到 G_ω
- 5) if G_ω 为空 then
- 6) 构建 $U(v)$ 中节点为簇头的子图, 将其存储到 G_ω
- 7) 通过式(12)由 G_ω 生成一阶赋权邻域状态图模型 G'_ω
- 8) 构建 G'_ω 中权值最大的子图 G_{\max}
- 9) id 为 G_{\max} 中 Router ID 最大节点的 Router ID
- 10) 通过式(12)由 $U(v)$ 生成一阶赋权邻域状态图模型 $U'(v)$
- 11) else
- 12) 通过式(12)由 G_ω 生成一阶赋权邻域状态图模型 G'_ω
- 13) 构建 G'_ω 中权值最大的子图 G_{\max}
- 14) id 为 G_{\max} 中 Router ID 最大节点的 Router ID
- 15) 通过式(12)由 $U(v)$ 生成一阶赋权邻域状态图模型 $U'(v)$
- 16) end if
- 17) 通过完全邻接关系主动关联机制与 id 对应的簇头建立完全邻接关系
- 18) 向簇头传递并不断更新 $U'(v)$ 并监听簇头完全邻接关系被动关联机制发起的消息
- 19) else
- 20) id 为空
- 21) $U'(v)$ 为空
- 22) end if
- 23) else
- 24) id 为空
- 25) $U'(v)$ 为空
- 26) end if

算法 2 用于簇内节点的自适应入簇, 进而实现簇内路由收敛, 具体如下。节点根据一阶无权邻域状态图模型 $U(v)$ 和网络拓扑结构构建相应的子图 G_ω , 进而计算出权值最大的一阶赋权邻域状态图模型 G_{\max} 。在此基础上, 节点依据 Router ID 的大

小确定要入簇的簇头地址id,通过完全邻接关系主动关联机制实现节点入簇,并传递以及不断更新簇间连接算法所需的一阶赋权邻域状态图模型 $U'(v)$,开启完全邻接关系被动关联机制。算法2与算法1类似,由一系列条件判断模块构成,其核心执行过程同样包含对图模型的完整遍历。因此,该算法的时间复杂度亦为 $O(n)$,其中 n 表示图中节点的总数。

算法3 簇间连接算法

输入 二阶赋权邻域状态图模型 $X'(v)$,邻居节点 u_i 的一阶赋权邻域状态图模型 $U'(u_i)$

输出 要连接的二阶节点Router ID链表ids

- 1) 依据 $U'(u_i)$ 和式(14)构建经过邻居节点 u 的二阶赋权邻域状态图模型 $X'(u_i)$
- 2) 依据节点所属簇的不同将 $X'(u_i)$ 划分成多个二阶赋权邻域状态图模型 $\{X'_1(u_i), X'_2(u_i), \dots, X'_n(u_i)\}$
- 3) 依据节点所属簇的不同将 $X'(v)$ 划分成多个二阶赋权邻域状态图模型 $\{X'_1(v), X'_2(v), \dots, X'_m(v)\}$
- 4) for $X'_j(u_i) \in \{X'_1(u_i), X'_2(u_i), \dots, X'_n(u_i)\}$ do
- 5) 得到 $X'_j(u_i)$ 所属簇的Router ID: id_j
- 6) 初始化 $X'_z(v) = (N'', E'', v)$
- 7) for $X'_k(v) \in \{X'_1(v), X'_2(v), \dots, X'_m(v)\}$ do
- 8) if id_j 与 $X'_k(v)$ 所属簇的簇头的Router ID相同 then
- 9) 将 $X'_k(v)$ 存储到 $X'_z(v)$
- 10) end if
- 11) end for
- 12) if $X'_z(v)$ 中已有全连接路径 then
- 13) 将 $X'_j(u_i)$ 与已有全连接的图合并得 $x'(v)$
- 14) 构建 $x'(v)$ 中权值最大的子图 x_{\max}
- 15) id为 x_{\max} 中Router ID最大的二阶节点Router ID
- 16) if 未与id连接 then
- 17) 将id加入ids中
- 18) end if
- 19) else
- 20) 构建 $X'_j(u_i)$ 中权值最大的子图 x_{\max}
- 21) 将 x_{\max} 中Router ID最大的二阶节点的Router ID加入ids中

22) end if

23) end for

24) 激活完全邻接关系被动关联机制实现节点 u_i 与相应ids完成连接

25) 根据式(14)更新 $X'(v)$

算法3用于建立各个簇之间的完全邻接关系,进而实现簇间节点的路由收敛,具体如下。簇头节点 v 根据邻居节点 u_i 传递的一阶赋权邻域状态图模型 $U'(u_i)$ 构建二阶赋权邻域状态图模型 $X'(u_i)$,进而依据从属的簇将 $X'(u_i)$ 和已生成的二阶赋权邻域状态图模型 $X'(v)$ 分成多个子图 $\{X'_1(u_i), X'_2(u_i), \dots, X'_n(u_i)\}$ 和 $\{X'_1(v), X'_2(v), \dots, X'_m(v)\}$ 。在此基础上,簇头节点 v 通过权重和节点Router ID信息,根据上述子图集合 $\{X'_1(u_i), X'_2(u_i), \dots, X'_n(u_i)\}$ 和 $\{X'_1(v), X'_2(v), \dots, X'_m(v)\}$,确定连接的链表ids,进而通过完全邻接关系的被动关联机制实现簇间连接。

具体而言,算法3的第4)~23)行展示了生成连接链表ids的过程。该算法通过一个循环遍历所有子图 $\{X'_1(u_i), X'_2(u_i), \dots, X'_n(u_i)\}$,并在遍历过程中对每个子图 $X'_j(u_i)$,根据其从属簇的簇头Router ID,在集合 $\{X'_1(v), X'_2(v), \dots, X'_m(v)\}$ 中寻找相同从属关系的簇 $X'_z(v)$ 。第5)~11)行为具体的实现过程。若在簇 $X'_z(v)$ 中已存在通往该簇的路径,则依据 $X'_j(u_i)$ 和现有路径计算需要连接的节点。若不存在该路径,则通过 $X'_j(u_i)$ 计算出新的连接节点。第12)~22)行为此过程的具体实现。最终,经过遍历所有子图 $\{X'_1(u_i), X'_2(u_i), \dots, X'_n(u_i)\}$ 得到最终的连接链表ids。算法3在执行过程中主要由2个嵌套的for循环构成,其算法的时间复杂度为 $O(nm)$,其中 n 表示 $X'(u_i)$ 所划分的子图的个数, m 表示 $X'(v)$ 所划分子图的个数。

2 仿真实验

2.1 仿真指标

实验主要通过3个指标来衡量路由收敛性能,分别是路由收敛总次数、路由开销和分组投递率(PDR, packet delivery rate)。指标的定义和公式如下。

1) 路由收敛总次数

路由收敛总次数是5 000 s仿真时间内网络节点因故障链路引起的路由重新收敛总次数。路由重新收敛总次数越少,则表明该分簇式路由收敛

方法规避了更多的故障链路，路由收敛的稳定性越高。

2) 路由开销

路由开销是路由协议控制数据包总比特数 B_r 与传输成功的数据包总比特数 B_s 之比。路由开销越小，则协议路由收敛的成本越低^[37-39]。

$$RO = \frac{B_r}{B_s} \times 100\% \quad (17)$$

3) 分组投递率

分组投递率是节点成功传输数据包 P_r 占节点发送数据包 P_s 的比例。节点的分组投递率越高，则该路由收敛方法的可靠性越高^[40]。

$$PDR = \frac{P_r}{P_s} \times 100\% \quad (18)$$

2.2 仿真实验

本文实验仿真平台为Linux操作系统Ubuntu16.04, 硬件配置为12th Gen Intel(R) Core(TM) i7-127000H, 16.0GB RAM。实验利用NS3网络仿真软件实现仿真。

实验共计两组。实验1在本文提出的分簇式路由收敛架构上融入了基于地理位置的分簇算

法^[25-26]、基于节点ID的分簇算法^[27-28]、基于遗传算法的分簇算法^[29-31]和本文提出的基于状态图模型分簇算法(OSPF-CSG)，并与传统OSPF协议(不分簇)进行性能比较。此实验旨在验证分簇式路由收敛架构的可行性和可扩展性，以及本文算法所具备的优势。实验中随机引入1~15条故障链路，分析不同故障链路条数对路由收敛性能的影响。实验2则对传统OSPF协议、本文算法、基于地理位置的分簇算法、基于节点ID的分簇算法、基于负载中心性(LC, load center)算法^[15]以及OSPF-BFD算法^[9]进行比较，旨在验证OSPF-CSG路由收敛算法的性能优势。实验构建了8种不同规模的网络拓扑，并针对各网络拓扑中的链路设置2种不同的损毁程度，以进一步评估本文算法在不同链路损毁程度和不同网络规模时的有效性。

1) 实验1

实验以文献[41]作为参考，构建战术通信网络拓扑结构，如图7所示。该结构包含39个节点、10种异构链路，异构链路类型如表1所示，流量传输路径为“上级指挥所-本级指挥所-分队”和“分队-本级指挥所-上级指挥所”。

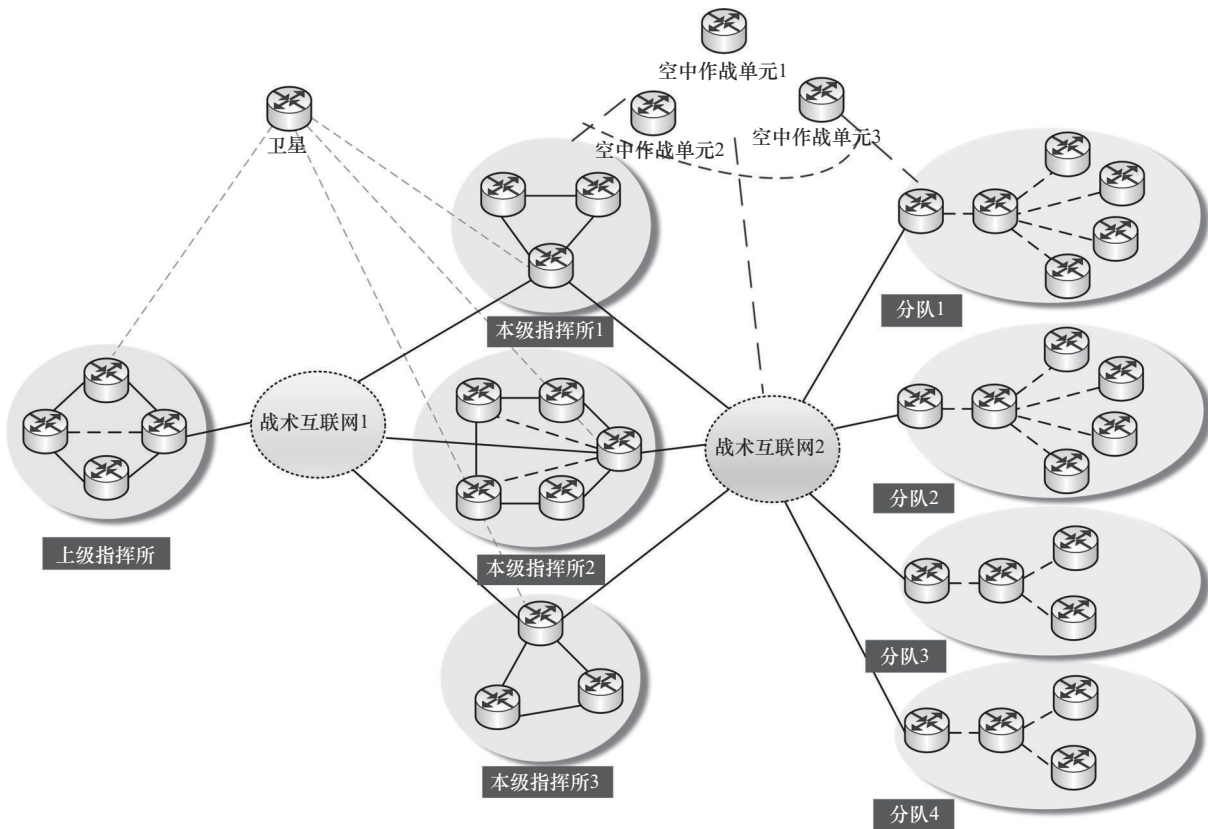


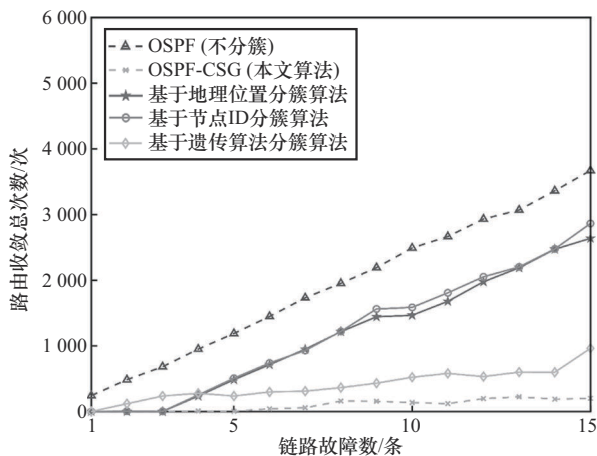
图7 战术通信网络拓扑结构

| 作战要素 | 链路类型 |
|--------|------------|
| 战术互联网 | 微波、散射 |
| 上级指挥所 | 区宽、有线、光纤 |
| 本级指挥所 | 区宽、有线 |
| 分队 | 区宽、UHF、VHF |
| 空中作战单元 | JIDS、综合数据链 |
| 卫星 | 卫星 |

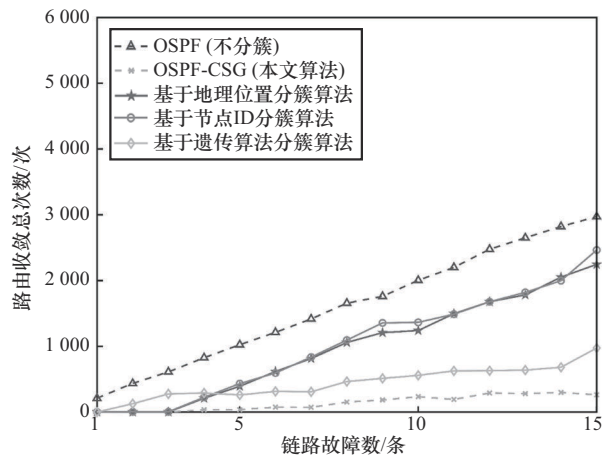
实验分别对路由收敛总次数、路由开销和分组投递率 3 个性能进行仿真, 比较 OSPF-CSG (本文算法)、OSPF (不分簇)、基于地理位置分簇算法、基于节点 ID 分簇算法以及基于遗传算法分簇算法的性能。仿真随机设置 1~15 条故障链路, 总时间为 5 000 s。在 Hello 报文发送间隔分别为 1 s、5 s 和 10 s, 判断链路故障时间分别为 5 s、20 s 和 40 s 的不同检测时间下进行实验, 具体如下。

图 8 为不同检测时间下各算法的路由收敛总次数对比情况。从图 8 可以看出, OSPF-CSG 在不同故障检测时间参数设置下, 路由收敛总次数均低于其他算法, 相较 OSPF (不分簇)、基于地理位置分簇算法、基于节点 ID 分簇算法和基于遗传算法分簇算法路由收敛总次数分别平均下降 80%、48.98%、62.95% 和 66.01%。这表明 OSPF-CSG 在极大程度上规避了链路频繁故障引起的路由收敛, 有效提升了路由收敛的稳定性。但随着链路故障数的增加, 仿真时间内路由收敛总次数的优势明显减弱。原因在于在链路故障数少于 3 条时 OSPF-CSG 直接检测并规避了故障链路, 导致其效果极佳。后随着链路故障数的增多, 为实现节点入簇和簇间连接一些故障链路不可避免地被收敛进来, 导致其路由收敛总次数有所增加, 进而使其性能优势下降。

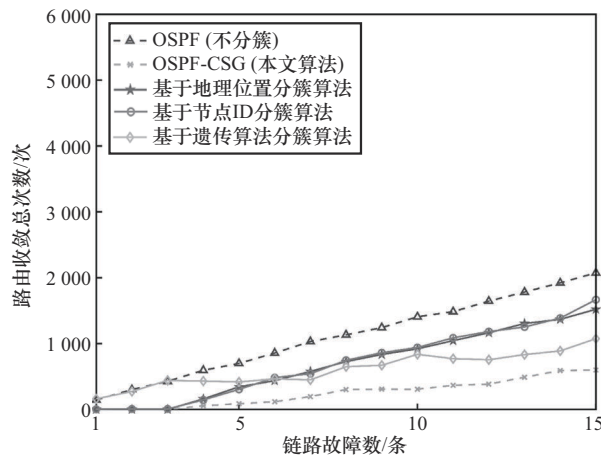
图 9 为不同故障检测时间下各算法的路由开销对比情况。从图 9 可以看出, 在不同故障检测时间



(a) Hello 报文发送间隔为 1 s, 判断链路故障时间为 5 s



(b) Hello 报文发送间隔为 5 s, 判断链路故障时间为 20 s

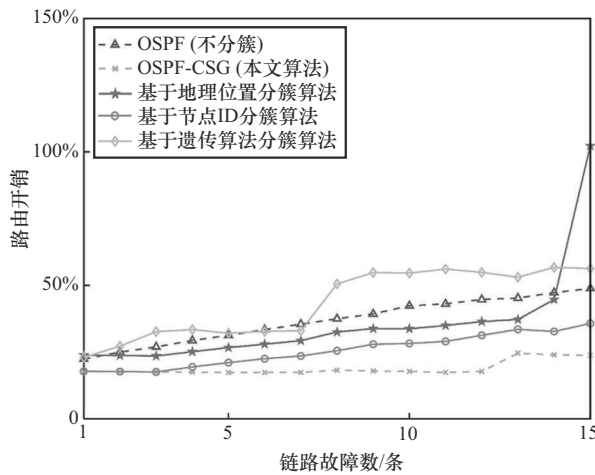


(c) Hello 报文发送间隔为 10 s, 判断链路故障时间为 40 s

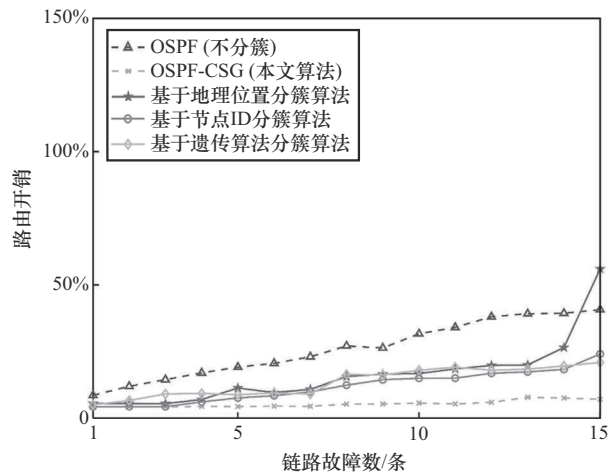
图 8 链路故障数-路由收敛总次数对比情况

参数设置下，OSPF-CSG路由开销均低于其他算法，比OSPF（不分簇）平均降低50%。特别是链路故障数为15条同时故障检测时间为20 s时，OSPF-CSG相较OSPF（不分簇）、基于地理位置分簇算法、基于节点ID分簇算法和基于遗传算法分簇算法路由开销分别下降65.64%、47.01%、28.02%和66.29%，大幅减少无效数据（路由协议功能包）对网络资源的损耗。之所以OSPF-CSG具有如此低路由开销，一方面是由于OSPF-CSG路由收敛时，依据网络状态图模型大幅减少对故障链路的收敛，从而减少对故障链路的反复收敛。另一方面是由于分簇式路由收敛后，邻居状态机建立的完全邻接关系也会随之减少，从而导致LSA的泛洪路径减少，进而减少LSA的泛洪带来的路由开销。两优势共同作用下使得OSPF-CSG具有极低的路由开销。

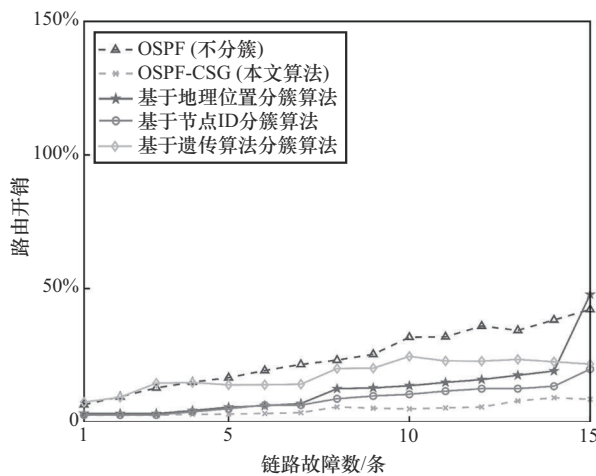
图10为不同故障检测时间下各算法的分组投递率对比情况。从图10可以看出，OSPF-CSG的分组投递率明显高于其他算法，比OSPF（不分簇）在不同检测时间下平均提升7%。特别是在链路故障数为15条同时链路故障检测时间为40 s的情况下，OSPF-CSG相较OSPF（不分簇）、基于地理位置分簇算法、基于节点ID分簇算法和基于遗传算法分簇算法分别提升67.66%、23.42%、9.07%和47.16%，大幅增加业务有效传输时间。上述优势是由于OSPF-CSG基于网络状态图模型进行分簇式路由收敛时，将会收敛更多优质链路并减少故障链路的收敛，致使所形成的路由表的可靠性更强。业务凭借Dijkstra算法计算路径时将会有更高的概率不会经过故障链路，从而带来更高的分组投递率。



(a) Hello报文发送间隔为1 s，判断链路故障时间为5 s

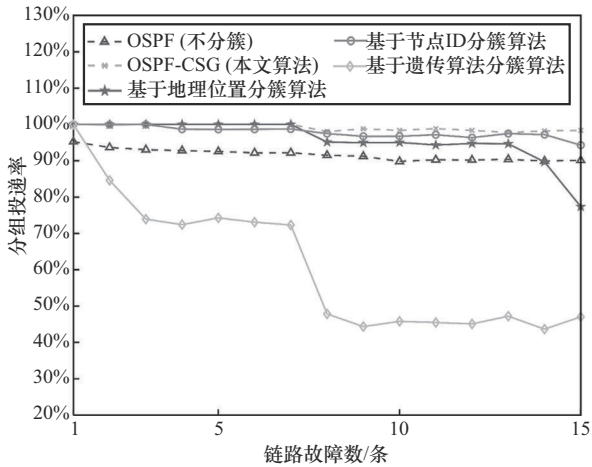


(b) Hello报文发送间隔为5 s，判断链路故障时间为20 s

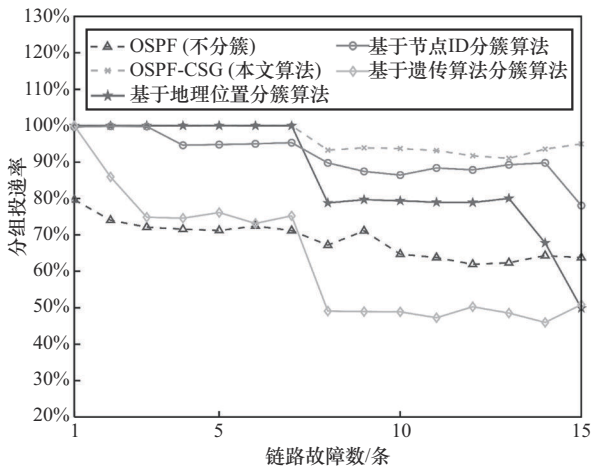


(c) Hello报文发送间隔为10 s，判断链路故障时间为40 s

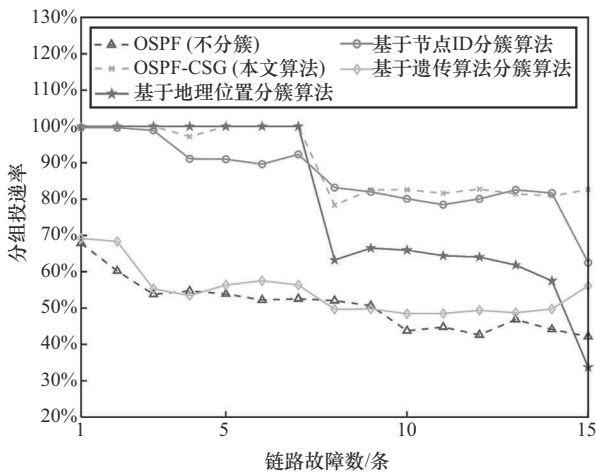
图9 链路故障数-路由开销对比情况



(a) Hello报文发送间隔为1 s, 判断链路故障时间为5 s



(b) Hello报文发送间隔为5 s, 判断链路故障时间为20 s



(c) Hello报文发送间隔为10 s, 判断链路故障时间为40 s

图 10 链路故障数-分组投递率对比情况

2) 实验 2

基于实验 1 的网络拓扑结构, 适度调整了战术通信网络中的分队数量, 从而形成 7 种不同的拓扑结构配置。在此基础上, 进一步扩充具有 7 个分队

的拓扑, 得到网络规模更大的拓扑 8。具体的分队数量、节点数量和链路数量如表 2 所示。

| 拓扑编号 | 分队数量/个 | 节点数量/个 | 链路数量/条 |
|------|--------|--------|--------|
| 1 | 1 | 25 | 45 |
| 2 | 2 | 31 | 54 |
| 3 | 3 | 35 | 61 |
| 4 | 4 | 39 | 68 |
| 5 | 5 | 45 | 77 |
| 6 | 6 | 49 | 84 |
| 7 | 7 | 53 | 91 |
| 8 | 7 | 73 | 111 |

实验通过对路由收敛总次数、路由开销和分组投递率 3 个参数的仿真, 比较传统 OSPF、本文算法、基于地理位置分簇算法、基于节点 ID 分簇算法、负载中心性 (LC) 算法以及 OSPF-BFD 的性能。仿真针对各拓扑的链路分别设置 5% 和 10% 这 2 种不同的链路损毁程度, 仿真总时长为 5 000 s。实验结果具体如下。

图 11 展示了在 8 种不同网络拓扑下, 针对 2 种链路损毁程度的路由收敛总次数对比情况。实验结果显示, 分簇算法的路由收敛总次数明显低于其他算法, 特别是 OSPF-CSG, 其路由收敛次数极为有限。具体而言, OSPF-CSG 相比传统 OSPF、基于地理位置分簇算法、基于节点 ID 分簇算法、LC 算法和 OSPF-BFD, 平均下降幅度分别超过 83%、70%、69%、85% 和 85%。这一优势源于 OSPF-CSG 通过网络状态图模型实现分簇式路由收敛, 专注于优质链路的收敛, 而避免了对劣质链路的非必要收敛。除非劣质链路是连接其他节点的唯一路径, 通常情况下, OSPF-BFD 能够有效规避对该链路的收敛, 从而大幅减少因链路故障导致的总收敛次数。

图 12 展示了在 8 种不同网络拓扑下, 针对 2 种链路损毁程度的路由开销对比情况。结果表明, 分簇算法显著降低了路由开销, 特别是 OSPF-CSG 算法, 其路由开销远低于其他算法。在所有网络拓扑结构和不同链路损毁程度下, OSPF-CSG 相较于传统

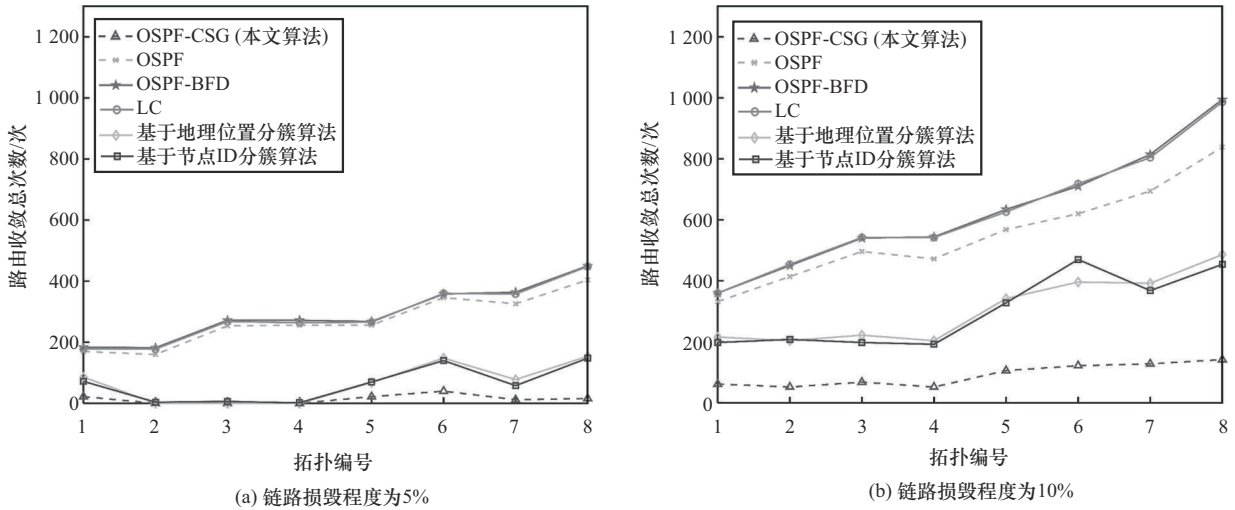


图 11 不同网络拓扑下路由收敛总次数对比情况

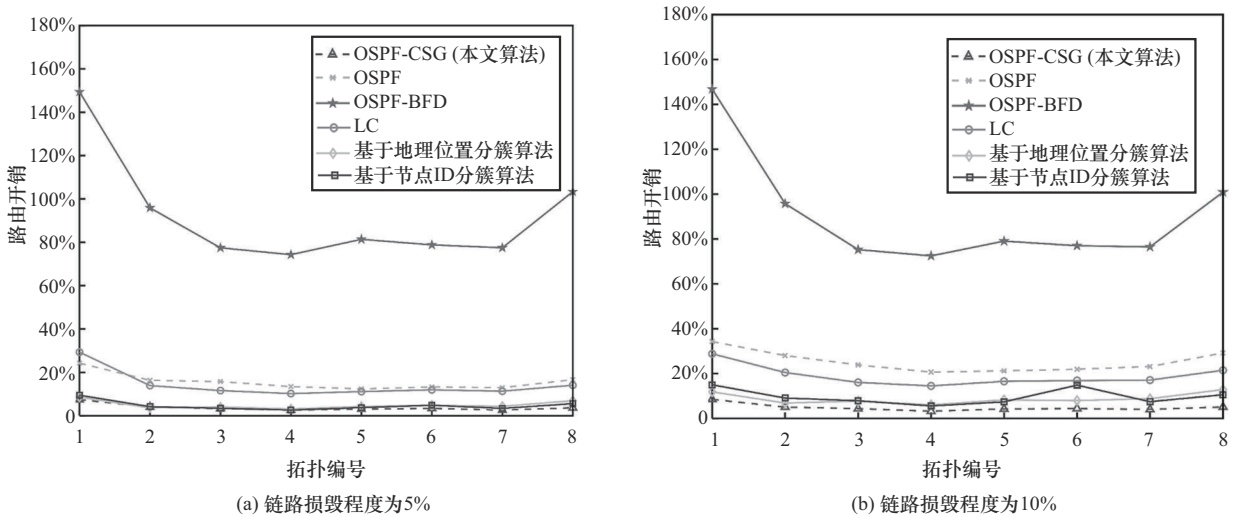


图 12 不同网络拓扑下路由开销对比情况

OSPF、基于地理位置分簇算法、基于节点ID分簇算法、LC算法以及OSPF-BFD，分别平均下降超过76%、20%、16%、73%和91%。OSPF-CSG的低路由开销归因于2个关键因素：一是其采用的分簇算法有效减少了冗余链路的收敛；二是通过构建网络状态图模型，节点能基于当前网络状况进一步优化分簇结构，强化了分簇算法的优势。反观OSPF-BFD，其高路由开销源于毫秒级的控制报文进行链路监测，链路拥塞和抖动导致的虚警触发了大量无效的LSA泛洪，最终造成显著的路由开销增加。

图 13 展示了在 8 种网络拓扑下，链路损毁程度为 5% 和 10% 时的分组投递率对比情况。结果表明，OSPF-CSG 算法的分组投递率基本保持在 98% 以上，甚至与通过毫秒级链路探测实现快速路由收

敛的 OSPF-BFD 算法相当。在 10% 链路损毁程度下，OSPF-CSG 相比传统 OSPF、基于节点 ID 分簇算法及 LC 算法，平均分组投递率分别提升 61%、8% 和 4%。OSPF-CSG 之所以表现出色，主要得益于其实时网络状态图模型的应用，有效规避了大量劣质链路，为业务提供了更优的通信环境。这一特性确保了仿真周期内链路基本不发生断开，大幅增加业务有效传输时间，从而实现了极高的分组投递率。

3 结束语

针对传统 OSPF 协议在战术通信网络中路由收敛效率低、路由开销大的问题，本文提出了一种基于网络状态图模型的分簇式路由收敛方法 OSPF-CSG。OSPF-CSG 基于战术通信网络中的链路和节点状态构

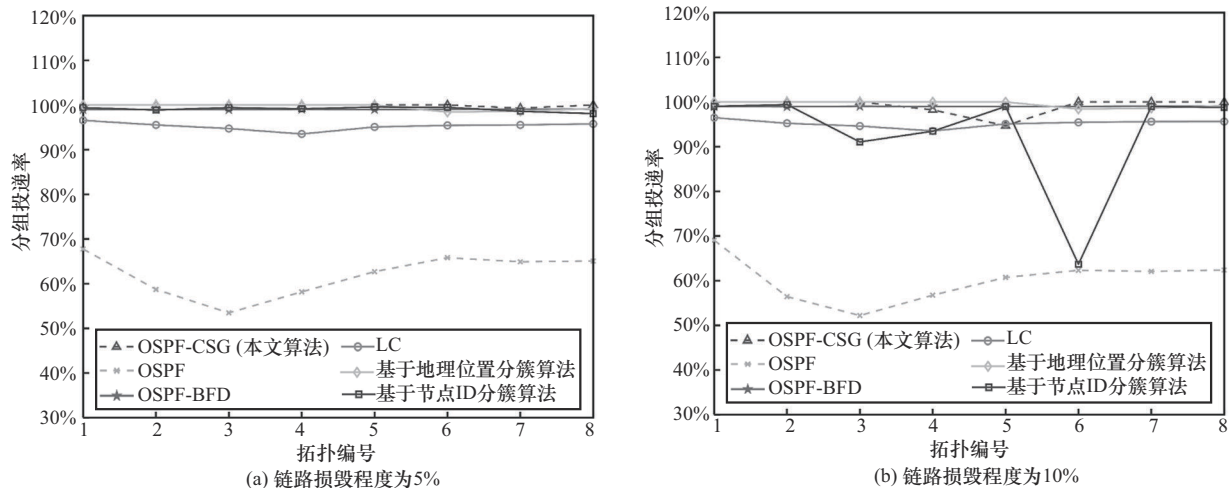


图 13 不同网络拓扑下分组投递率对比情况

建网络状态图模型。引入完全邻接关系主动/被动关联机制传递网络状态图模型并引导邻居状态机完全邻接关系的构建。在此基础上,设计了具备簇头自举、节点自适应入簇和簇间连接功能的分簇算法。仿真结果表明,在战术通信网络节点连接关系时变,链路故障频发情况下,OSPF-CSG在路由收敛总次数、路由开销和分组投递率方面均优于传统OSPF协议,实现了分簇式路由由高效收敛,显著增加了业务有效传输时间。为强对抗和高机动作战条件下,新一代战术通信网络的敏捷构建提供了技术支持。

参考文献:

- [1] 孙盛智, 刘玉, 盛碧琦, 等. “马赛克”战运行机制及制胜机理研究[J]. 指挥控制与仿真, 2023, 45(2): 150-154.
SUN S Z, LIU Y, SHENG B Q, et al. Research on the operation mechanism and winning mechanism of mosaic warfare[J]. Command Control & Simulation, 2023, 45(2): 150-154.
- [2] 胡志强, 刘永普, 姜松, 等. 分布式智能化作战的概念、内涵与未来发展[J]. 指挥信息系统与技术, 2023, 14(5): 62-68.
HU Z Q, LIU Y P, JIANG S, et al. Concept, connotation and future development of distributed intelligent operations[J]. Command Information System and Technology, 2023, 14(5): 62-68.
- [3] 张志华, 王凡. 第五代指挥信息系统总体及其智能化技术设想[J]. 指挥控制与仿真, 2021, 43(5): 1-7.
ZHANG Z H, WANG F. The fifth generation command information system and its intelligent technology[J]. Command Control & Simulation, 2021, 43(5): 1-7.
- [4] COLTUN R, FERGUSON D, MOY J, et al. RFC 5340: OSPF for ipv6[R]. 2008.
- [5] MOY J, SAFARIA O M C. OSPF: anatomy of an Internet routing protocol[M]. Array Addison-Wesley Professional, 1998.
- [6] MOY J. RFC 2328: OSPF version 2[R]. 1998.
- [7] 潘成胜, 曹康宁, 石怀峰, 等. 基于深度强化学习的战术通信网络路径优选算法[J]. 中国电子科学研究院学报, 2024, 19(2): 138-148.
PAN C S, CAO K N, SHI H F, et al. Tactical communication network

- path selection algorithm based on deep reinforcement learning[J]. Journal of China Academy of Electronics and Information Technology, 2024, 19(2): 138-148.
- [8] KATZ D, WARD D. Bidirectional forwarding detection (BFD)[R]. 2010.
- [9] TALAULIKAR K, PSENAK P, FU A, et al. OSPF bidirectional forwarding detection (BFD) strict-mode[J]. RFC, 2023, 9355: 1-10.
- [10] KHAN N, BIN SALLEH R, KOUBAA A, et al. Data plane failure and its recovery techniques in SDN: a systematic literature review[J]. Journal of King Saud University-Computer and Information Sciences, 2023, 35(3): 176-201.
- [11] 俞成康. 面向LEO网络的OSPF路由协议增强技术研究[D]. 北京:北京邮电大学, 2023.
YU C K. Research on OSPF routing protocol enhancement technology for LEO networks[D]. Beijing: Beijing University of Posts and Telecommunications, 2023.
- [12] 邢川, 陈二虎, 韩笑冬. 基于动静结合方法的卫星网络路由方法研究[J]. 空间控制技术与应用, 2020, 46(3): 55-59.
XING C, CHEN E H, HAN X D. Static-dynamic routing algorithm in satellite constellation network[J]. Aerospace Control and Application, 2020, 46(3): 55-59.
- [13] 徐明伟, 夏安青, 杨莞, 等. 天地一体化网络域内路由协议OSPF+ [J]. 清华大学学报(自然科学版), 2017, 57(1): 12-17.
XU M W, XIA A Q, YANG Y, et al. Intra-domain routing protocol OSPF+for integrated terrestrial and space networks[J]. Journal of Tsinghua University (Science and Technology), 2017, 57(1): 12-17.
- [14] WAQAS M, MALIK S U R, AKBAR S, et al. Convergence time analysis of OSPF routing protocol using social network metrics[J]. Future Generation Computer Systems, 2019, 94: 62-71.
- [15] MACCARI L, GHIRO L, GUERRIERI A, et al. Exact distributed load centrality computation: algorithms, convergence, and applications to distance vector routing[J]. IEEE Transactions on Parallel and Distributed Systems, 2020, 31(7): 1693-1706.
- [16] MACCARI L, CIGNO R L. Improving routing convergence with centrality: theory and implementation of pop-routing[J]. IEEE/ACM Transactions on Networking, 2018, 26(5): 2216-2229.
- [17] JAIN N, PAYAL A, JAIN A. Effect of data packet size on the performance of RIP and OSPF routing protocols in hybrid networks[J]. International Journal of Pervasive Computing and Communications, 2021, 17(4): 361-376.
- [18] LIU G, DENG Y, CHEONG K H. Network immunization strategy by eliminating fringe nodes: a percolation perspective[J]. IEEE Transactions

- on Systems, Man, and Cybernetics: Systems, 2023, 53(3): 1862-1871.
- [19] RAMANATHAN R, REDI J. A brief overview of ad hoc networks: challenges and directions[J]. IEEE Communications Magazine, 2002, 40(5): 20-22.
- [20] MACKTOOBIAN M, SHU Z, ZHAO Q. Topology recoverability prediction for ad-hoc robot networks: a data-driven fault-tolerant approach[J]. IEEE Transactions on Signal and Information Processing Over Networks, 2023, 9: 786-799.
- [21] GOYAL M, SOPERI M, BACCELLI E, et al. Improving convergence speed and scalability in OSPF: a survey[J]. IEEE Communications Surveys & Tutorials, 2012, 14(2): 443-463.
- [22] MICHEL O, BIFULCO R, RÉTVÁRI G, et al. The programmable data plane: abstractions, architectures, algorithms, and applications[J]. ACM Computing Surveys, 2021, 54(4): 1-36.
- [23] PARIZOTTO R, COELHO B L, NUNES D C, et al. Offloading machine learning to programmable data planes: a systematic survey[J]. ACM Computing Surveys, 2024, 56(1): 1-34.
- [24] RYBOWSKI N, BONAVENTURE O. Evaluating OSPF convergence with ns-3 DCE[C]//Proceedings of the 2022 Workshop on ns-3. New York: ACM Press, 2022: 120-126.
- [25] JORIO A, FKIHI S E, ELBHIRI B, et al. An energy-efficient clustering routing algorithm based on geographic position and residual energy for wireless sensor network[J]. Journal of Computer Networks and Communications, 2015, 2015: 170138.
- [26] RIZKY R, HAKIM Z, SETIYOWATI S, et al. Development of the multi-channel clustering hierarchy method for increasing performance in wireless sensor network[J]. MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer, 2024, 23(3): 603-614.
- [27] PATHAK S, JAIN S, BORAH S. Clustering algorithms for MANETs: a review on design and development[C]//Advances in Intelligent Systems and Computing. Berlin: Springer, 2020: 563-578.
- [28] VENKATASUBRAMANIAN S, SUHASINI D A, VENNILA D C. An energy efficient clustering algorithm in mobile adhoc network using ticket ID based clustering manager[J]. International Journal of Computer Science & Network Security, 2021.
- [29] HAMZA F, VIGILA S M C. Cluster head selection algorithm for MANETs using hybrid particle swarm optimization-genetic algorithm[J]. International Journal of Computer Networks and Applications, 2021, 8(2): 119-129.
- [30] AHMAD M, HAMEED A, ULLAH F, et al. A bio-inspired clustering in mobile adhoc networks for Internet of things based on honey bee and genetic algorithm[J]. Journal of Ambient Intelligence and Humanized Computing, 2020, 11(11): 4347-4361.
- [31] KALIAPPAN M, MARIAPPAN E, PRAKASH M V, et al. Load balanced clustering technique in MANET using genetic algorithms[J]. Defence Science Journal, 2016, 66(3): 251.
- [32] HUSNAIN G, ANWAR S. An intelligent cluster optimization algorithm based on whale optimization algorithm for VANETs (WOACNET)[J]. PLoS One, 2021, 16(4): e0250271.
- [33] ASAAMONING G, MENDES P, MAGAIA N. A dynamic clustering mechanism with load-balancing for flying ad hoc networks[J]. IEEE Access, 2021, 9: 158574-158586.
- [34] KHEDR A M, SALIM A, RAJ P V P, et al. MWCRSF: mobility-based weighted cluster routing scheme for FANETs[J]. Vehicular Communications, 2023, 41: 100603.
- [35] HOSSEINZADEH M, TANVEER J, RAHMANI A M, et al. A Q-learning-based smart clustering routing method in flying ad hoc networks[J]. Journal of King Saud University-Computer and Information Sciences, 2024, 36(1): 101894.
- [36] KHEDR A M, PRAVIJA R P V. A hybrid MGO-JAYA based clustered routing for FANETs[J]. Vehicular Communications, 2024, 45: 100729.
- [37] SU Y S, XU Y N, PANG Z F, et al. HCAR: a hybrid-coding-aware routing protocol for underwater acoustic sensor networks[J]. IEEE Internet of Things Journal, 2023, 10(12): 10790-10801.
- [38] 尹婷. 面向 5G 端到端的协同定位技术研究[D]. 西安: 中国科学院大学(中国科学院国家授时中心), 2022.
- YIN T. Research on cooperative localization technologies for 5G device to device[D]. Xi'an: University of Chinese Academy of Sciences (National Timing Center, Chinese Academy of Sciences), 2022.
- [39] 齐晓鑫. 低轨卫星网络路由算法与拓扑控制策略研究[D]. 西安: 西安电子科技大学, 2022.
- QI X X. Research on routing algorithm and topology control strategy of LEO satellite network[D]. Xi'an: Xidian University, 2022.
- [40] AMOUR B S, JAEKELA. Data rate selection strategies for periodic transmission of safety messages in VANET[J]. Electronics, 2023, 12(18): 3790.
- [41] 于全. 战术通信理论与技术[M]. 北京: 人民邮电出版社, 2020.
- YU Q. Tactical communications theory and technology[M]. Beijing: Posts and Telecommunications Press, 2020.

[作者简介]



潘成胜 (1962-), 男, 江苏宜兴人, 南京信息工程大学教授、博士生导师, 主要研究方向为智能网络理论与技术。



陆煌杰 (1999-), 男, 浙江绍兴人, 南京信息工程大学硕士生, 主要研究方向为战术通信网络路由快速收敛技术。



石怀峰 (1989-), 男, 江苏徐州人, 南京信息工程大学副教授、硕士生导师, 主要研究方向为智能网络流量控制技术。



王英植 (1994-), 男, 河南邓州人, 南京信息工程大学博士生, 主要研究方向为智能网络理论与技术。