# 通过软件定义网络（SDN）增强移动自组网（MANETs）：一种平衡的方法

Riccardo Fonti 和 Andrea Piroddi*

博洛尼亚大学工程学院工程与计算机科学系，意大利切塞纳 电子邮箱：Riccardo.fonti@studio.unibo.it (R.F.)；andrea.piroddi@unibo.it (A.P.)*通讯作者 手稿收到日期：2024年12月18日；修订日期：2025年2月2日；录用日期：2025年2月28日；发表日期：2025年4月28日

摘要—移动自组网（MANETs）是去中心化无线网络，其特点是动态拓扑和节点移动性。在尖端技术不断发展的时代，将软件定义网络（SDN）与移动自组网相结合，为更高效地应对这些挑战提供了一种有前景的解决方案。本文对移动自组网和软件定义网络进行了平衡的讨论，阐述了集中式控制和网络虚拟化等软件定义网络原则如何从可扩展性、成本效益和安全性等方面优化移动自组织网络性能。本文还构建了一个数学模型，用于分析资本性支出（CAPEX）、运营性支出（OPEX）和网络效率。

关键词—移动自组网（MANETs）、软件定义网络（SDN）、可扩展性、网络效率、动态拓扑、成本效益分析

## I. 引言

移动自组网（MANETs）是一种无线网络，由移动设备动态构建，无需路由器或接入点等任何现有基础设施 [1]。这类网络具有自配置和去中心化的特点，这意味着网络中的每台设备既可充当主机，也可充当路由器，将数据转发给其他设备。移动自组网能够高度适应网络拓扑的变化，因此适用于网络基础设施难以部署或部署成本过高的环境，例如灾难恢复场景、军事行动和传感器网络 [2]。

然而，移动自组网的去中心化特性带来了重大挑战，尤其是在网络控制、资源管理和安全性方面。缺乏集中式控制往往会导致路由效率低下、可扩展性受限，并容易遭受欺骗攻击或拒绝服务（DoS）等攻击 [3]。此外，由于移动自组网具有动态特性，节点经常改变位置，难以维持稳定连接并有效管理带宽和功耗等资源 [4]。因此，需要高效的机制来应对这些复杂性，同时维持网络性能。

相比之下，软件定义网络（SDN）引入了一种将控制平面与数据平面解耦的范式，从而实现集中式网络控制与管理 [5]。近年来，SDN 凭借其灵活性、可编程性和集中式管理的优势，以及应对传统网络架构诸多挑战的潜力，受到广泛关注。将 SDN 原则融入移动自组网，可缓解网络固有的挑战，例如动态拓扑变化、资源约束和安全漏洞。SDN 的集中式控制平面能够提供网络的全局视图，从而实现更高效的路由、更优的资源分配和更强的安全性 [6]。

软件定义网络与移动自组网的融合，为提升此类网络的可扩展性、灵活性和成本效益带来了有望的方向 [7]。通过软件定义网络实现集中式控制，可进行实时监控和决策，从而提升移动自组织网络在带宽利用率和能效方面的表现。此外，软件定义网络的可编程性支持网络策略的动态重配置，使网络在不可预测的环境中更加稳健、适应性更强 [5]。

本文探讨软件定义网络的原理如何提升移动自组织网络架构，重点解决可扩展性、资源管理和安全性方面的问题。我们通过数学模型量化将软件定义网络融入移动自组网所带来的潜在成本与效率改善。

## II. 移动自组网的特征与挑战 S

移动自组网（MANETs）能够即时建立无线网络，而无需依赖固定基础设施，这一能力使其特别适用于军事行动、灾难恢复和临时活动覆盖等场景 [1]。然而，赋予移动自组网灵活性的这些特征，也带来了若干技术挑战，尤其是在网络控制、可扩展性和资源管理方面。

### A. 动态拓扑

移动自组网的一项显著特征是其动态拓扑。节点的移动会导致网络路由频繁变化，因为连接会不可预测地建立和断开 [3]。移动自组网中的每个节点既充当主机，也充当路由器，为其他节点转发数据包，这意味着路由协议必须持续适应不断变化的拓扑 [2]。为应对这些挑战，人们开发了按需距离矢量路由（AODV）和动态源路由（DSR）等传统路由协议，但这些协议在大规模和高度动态的环境中存在局限性。它们高度依赖路由发现和维护流程，随着网络规模扩大或动态性增强，可能会产生大量开销。

此外，由于缺乏固定基础设施，路由决策必须以分布式方式作出。

这种分布式控制往往会导致路由路径欠佳、时延增加，并在路由重新计算期间造成丢包。此外，节点移动性导致的链路中断会严重降低网络性能，尤其是在节点快速移动、导致路由频繁中断时 [4]。

B. 可扩展性

可扩展性是移动自组网中的另一个关键问题。随着节点数量增加，维护最新路由表和管理网络资源的复杂度会呈指数级增长 [3]。在大型移动自组网中，节点需要存储和处理更多路由信息，从而导致计算开销增加、功耗上升。此外，随着网络规模扩大，广播风暴——即节点向网络泛洪发送控制消息——可能会成为一个严重问题 [8]。

在节点密度较高的场景中，节点之间发生干扰的概率也会升高，这会进一步限制网络的可扩展性。此外，拓扑的动态特性要求路由协议不断更新路由，从而增加信令流量，可能导致网络拥塞，限制其可扩展性。为解决可扩展性问题，人们提出了分层路由、基于区域的路由和聚类等方案，但这些方法也增加了网络管理的复杂度 [9]。

C. 资源约束

移动自组网中的设备通常运行在资源受限的环境中。节点往往配备有限的电池电量、处理能力和带宽 [4]。功耗是一个尤为关键的问题，因为移动设备需要节约能源以延长运行寿命。持续进行路由更新、分组转发以及参与路由过程所产生的开销会迅速耗尽电池电量，尤其是在节点密集或高度移动的网络中。

带宽限制也带来了重大挑战。在移动自组网中，多个节点共享无线链路，导致可用带宽竞争。这种竞争会造成数据吞吐量降低、分组碰撞率升高以及时延增加 [10，11]。此外，动态拓扑会加剧这些问题，因为频繁的路由变化和分组重传会占用额外带宽。有效的带宽管理（例如通过负载均衡和拥塞控制机制）对于确保网络性能达到最佳至关重要。

解决这些挑战对于提升移动自组网的可靠性和性能至关重要。为应对这些问题，人们提出了跨层优化、节能路由协议和自适应资源管理等方案，但这些方案往往适用范围有限，或在复杂性和性能方面需要作出权衡 [9]。

## III. 移动自组网中的软件定义网络：集成框架

软件定义网络（SDN）引入了一种集中式的网络控制方法，这与传统移动自组网的分布式特性形成对比。通过将控制平面与数据平面解耦，SDN 提供了对整个网络的集中式决策和控制。这种分离使 SDN 控制器能够管理移动自组网的路由、资源分配和安全策略，从而提升整体网络性能 [5]。

A. 集中式控制

在传统移动自组网中，每个节点使用按需距离矢量路由（Ad-hoc On-Demand Distance Vector，AODV）或优化链路状态路由（Optimized Link State Routing，OLSR）等分布式算法在本地作出路由决策。由于路由变化频繁且路由发现开销较大，这些协议在高度动态的环境中可能效率低下。软件定义网络引入了一个全局集中式控制器，凭借对拓扑的全局视图来管理整个网络的路由。

设想一个由 n 个节点组成的移动自组网（MANET），每个节点的连接度表示为 $d_i$，其中 i = 1、2、...、n。在传统分布式系统中，路由复杂度会随连接度 $d_i$ 成比例增加，因为每个节点都要独立维护到其邻居节点的路由

[3]。去中心化移动自组网中的平均路径成本 $C_{path}$ 可表示为:

$$C_{\text{path}} = \frac{1}{n}\sum_{i=1}^{n} C_i$$

其中，$C_i$ 表示节点 i 的路径成本。随着节点移动性增加，$C_i$ 也会增加，这是因为需要频繁进行路由重新计算，并且广播路由更新会带来额外开销。

借助软件定义网络（SDN），控制器可以集中计算并优化路由，最大限度地减少冗余的路由发现，并降低整体路径成本。集中式路由决策可建模为:

$$C_{\text{SDN-path}} = \min_{P} \sum_{i=1}^{n} w_i \cdot d_i$$

其中，$w_i$ 是与每个节点的流量负载或优先级相关的权重，P 表示所有可能的路径。软件定义网络控制器可以根据实时流量状况动态调整路由，从而降低时延和丢包，如基于 OpenFlow 的 SDN 实现研究所示 [5]。

B. 动态资源分配

软件定义网络在移动自组网中的一项主要优势是动态资源分配。在移动自组网这类资源受限环境中，节点的功率、带宽和计算能力有限，因此高效的资源管理至关重要。传统的移动自组网协议缺乏动态管理资源的能力，因为每个节点都独立处理自身的功率和带宽分配，这往往会导致资源利用不理想和网络拥塞 [4]。

借助软件定义网络，控制器可以监控网络状态，并根据网络的实时需求动态分配资源。设 $B_{total}$ 为网络中的可用总带宽，$P_{total}$ 为网络中的可用总功率。软件定义网络控制器旨在通过最小化分配成本 $C_{alloc}$来提高资源利用率，成本函数可定义为:

$$C_{\text{alloc}} = \sum_{i=1}^{n} P(V_i) \cdot I(V_i)$$

其中，$B_i$ 和 $P_i$ 分别表示分配给节点 i 的带宽和功率。控制器利用此成本函数优化资源分配，确保没有节点负担过重或资源匮乏 [7]。

此外，软件定义网络可以实施负载均衡技术，将流量分配到多条路径上，以避免拥塞。通过动态调整资源分配，控制器可以降低出现瓶颈的可能性，并确保网络中的资源分配更加公平。

C. 提升安全性

安全性是移动自组网面临的另一项重大挑战，尤其是由于其去中心化特性以及缺乏可信中央机构。传统移动自组网容易受到各种攻击，例如欺骗攻击、黑洞攻击和拒绝服务（DoS）攻击。在去中心化网络中，每个节点都必须独立实施安全协议，这往往会导致执行不一致，并造成网络防御方面的漏洞 [9]。

软件定义网络提供了一种集中式安全框架，控制器可在整个网络中实施统一的安全策略。控制器能够实时监控流量，并检测异常或可疑活动，例如流量突然激增（这可能表明发生了拒绝服务（DoS）攻击），或未经授权的节点试图加入网络。设 $S_{\text{risk}}$ 表示移动自组网中的安全风险，该风险是漏洞数量 V 和这些漏洞影响程度 I 的函数：

$$S_{\text{risk}} = \sum_{i=1}^{n} \left(\frac{B_i}{B_{total}} + \frac{P_i}{P_{total}}\right)$$

其中 P（$V_i$）表示漏洞 $V_i$ 被利用的概率，I（$V_i$）表示此类漏洞利用可能造成的影响。

借助软件定义网络，控制器可以动态调整安全策略，例如将流量重新路由至远离受损节点的位置，或将恶意节点与网络隔离。这既降低了攻击成功的概率，也减小了其潜在影响，从而显著提升网络的整体安全性 [5]。

D. 移动自组网中的 SDN协议

OpenFlow 等协议是软件定义网络框架的核心，因为它们能够在控制平面与网络设备之间实现安全通信 [5]。在移动自组网环境中，集成 OpenFlow 有助于通过提供更高效的路由路径并降低与路由发现过程相关的开销，来管理动态环境中的路由复杂性。

OpenFlow 协议允许 SDN 控制器实时更新路由表，使网络能够快速适应拓扑变化。设 $T_{\text{update}}$ 表示传统移动自组网中更新路由表所需的时间，其可表示为：

$$T_{\text{update}} = T_{\text{discovery}} + T_{\text{propagation}} + T_{\text{reconfiguration}}$$

其中，$T_{\text{discovery}}$ 是发现新路由所需的时间，$T_{\text{propagation}}$ 是路由更新在网络中传播所需的时间，$T_{\text{reconfiguration}}$ 是重新配置受影响节点所需的时间。在基于软件定义网络的移动自组网中，控制器通过集中管理路由过程来缩短更新时间，并将 $T_{\text{discovery}}$ 和 $T_{\text{propagation}}$降至最低，因为控制器拥有网络的全局视图。这使路由重配置更加迅速，从而提升网络的整体响应能力 [7]。

借助SDN协议，移动自组网能够采用更灵活、可编程的网络配置，从而更好地管理资源，并更快地适应不断变化的条件。这种集成为优化移动自组网的性能开辟了新的可能，尤其适用于网络面临快速且不可预测变化的环境。

将OpenFlow集成到移动自组网中，可以实现更高效的路由和网络管理，尤其适用于网络拓扑不断变化的动态环境。

在传统移动自组网中，节点必须依靠按需距离矢量路由或优化链路状态路由等分布式路由协议来发现并维护路由。然而，随着网络规模扩大或节点频繁移动，这些协议的效率可能会降低，因为它们需要不断重新计算路由。OpenFlow通过引入集中式SDN控制器来缓解这些问题，该控制器负责维护

网络拓扑的全局视图，并能够动态更新路由规则。

- 示例：基于OpenFlow的移动自组网路由优化

考虑一个由五个移动节点组成的移动自组网：A、B、C、D 和 E，其中节点 A 和 E 需要通信。在传统移动自组网中，路由依赖于按需距离矢量路由等反应式协议。节点 A 会发起路由发现过程，向邻居广播路由请求（RREQ）。节点 B、C 和 D 会转发该路由请求，直至其到达节点 E。这个过程会产生相当大的开销，尤其是在密集或大型网络中，因为可能有多个节点参与路由发现过程。

在基于软件定义网络的移动自组网中集成 OpenFlow 后，情况将有所不同。首先，部署软件定义网络控制器，以维护网络当前拓扑的全局视图。当节点 A 要向节点 E 发送数据时，它会向软件定义网络控制器发送请求，由控制器根据节点移动性、流量负载和链路质量等网络状况计算最优路径。随后，控制器可以配置支持 OpenFlow 的交换机或节点来建立路由。

- 路由优化的数学表示：

设网络表示为图 G = （V, E），其中 V 是节点（移动设备）的集合，E 是节点之间无线链路的集合。软件定义网络控制器通过求解以下优化问题，计算从节点 A 到节点 E 的最短路径或最优路径 P：

$$P_{\text{A}\to\text{E}} = \arg\min_{P} \sum_{(i,j)\in P} w_{ij}$$

其中，$w_{ij}$ 表示节点 i 和 j 之间链路的权重，该权重可以是链路质量、时延或

带宽可用性的函数。控制器使用网络中的实时数据动态分配权重。

计算出最优路径 $P_{A\to E}$ 后，控制器会向沿途支持 OpenFlow 的设备发送流规则，安装路由规则，使数据包从节点 A 经由中间节点（例如 B 和 C）转发至节点 E。使用流规则可显著降低路由发现和维护所带来的开销，因为路由会根据实时网络状况 [6] 预先建立并进行优化。

- 流表示例：

网络中支持 OpenFlow 的节点会维护一张流表，用于存储不同数据流的路由规则。例如，节点 B 的流表可能如表1所示：

表1. 流表示例

| 匹配项 | 操作 | 优先级 |
|---|---|---|
| 源：A，目的地：E | 转发至 C | High |
| 源：D，目的地：F | 转发至 E | 中等 |

匹配字段指示数据包的源和目的地，而动作字段则指定转发过程中使用的下一跳。通过这种设置，网络可以沿最优路径以最小时延转发数据包，因为节点不再需要各自独立计算路由。

- 动态重配置：.....的一项主要优势是

在软件定义网络支持的移动自组网中使用 OpenFlow，可实现动态重配置。在传统移动自组网中，如果节点移动导致节点 B 与节点 C 之间的链路断开，网络就需要启动新的路由发现过程，从而增加时延和丢包。在基于 OpenFlow 的移动自组网中，软件定义网络控制器可以检测到链路断开，并立即计算备用路由（例如，A → D → E），然后更新受影响节点的流表。

这种动态即时重配置路由的能力缩短了修复时间，并提高了网络的整体可靠性。总重配置时间 $T_{\text{reconfig}}$ 为：

$$T_{\text{reconfig}} = T_{\text{detection}} + T_{\text{computation}} + T_{\text{update}}$$

其中：

- $T_{\text{检测}}$是软件定义网络控制器检测到链路故障所需的时间。
- $T_{\text{计算}}$是计算新路由所需的时间。
- $T_{\text{更新}}$是更新受影响节点中的流表所需的时间。

由于软件定义网络控制器拥有网络的全局视图，$T_{\text{computation}}$ 显著低于传统分布式系统中的用时；在传统分布式系统中，每个节点都需要独立发现新路由。这使得网络能够更快地从链路故障中恢复，并提升网络韧性 [5]。

E. 在移动自组网中使用 OpenFlow 的优势

将 OpenFlow 集成到移动自组网中有几项重要优势：

- 改进的路由效率：通过由控制器全局管理路由，OpenFlow 减少了与分布式路由发现相关的开销。
- 实时网络自适应：软件定义网络控制器持续监控网络状况，从而能够实时调整路由路径。
- 增强安全性：OpenFlow支持集中式安全管理，使软件定义网络控制器能够在整个网络中实施统一的安全策略。
- 降低时延：通过消除对反应式路由发现的需求，OpenFlow可以降低高度动态环境中的时延和丢包。

上面的示例说明了OpenFlow如何优化路由计算，并动态重新配置网络以应对拓扑变化。借助SDN协议，移动自组网能够实现更高效、安全且灵活的网络管理，因此适用于各种应用场景，包括军事行动、灾难恢复和物联网环境。

## IV. 成本与效率的数学建模

通过数学模型量化资本支出（CAPEX）和运营支出（OPEX），可以分析软件定义网络与移动自组网集成所带来的经济效益。本节将比较传统移动自组网与软件定义网络赋能的移动自组网的资本性支出和运营支出模型，展示SDN-移动自组网方案的成本优势。

### A. 资本支出模型

传统移动自组网需要使用专用硬件（如路由器和交换机），以便在每个节点独立处理
路由和管理任务。这种去中心化架构意味着每个节点都必须具备足够的计算能力和专用硬件，以执行路由、安全性及其他网络管理功能 [4]。传统移动自组网的资本性支出很大程度上取决于节点数量 n 以及每个节点硬件组件的成本。

对于传统移动自组网，总资本性支出可建模为：

$$CAPEX_{\text{MANET}} = \sum_{i=1}^{n} \text{Cost}_{hwi} + \sum_{i=1}^{n} \text{Cost}_{swi}$$

其中：

- $\text{Cost}_{hwi}$ 是节点 i 处硬件的成本，包括专用路由器、交换机和其他设备。
- $\text{Cost}_{swi}$ 是节点 i 处软件的成本，这些软件用于路由、安全性和网络管理。

在软件定义网络赋能的移动自组网中，控制平面的架构是集中式的。这意味着路由和网络管理决策由软件定义网络控制器集中做出，从而减少了每个节点对专用硬件的需求。相反，SDN移动自组网中的节点可以依靠通用硬件，由软件定义网络控制器处理复杂的计算任务。

软件定义网络赋能的移动自组网的资本性支出建模如下：

$$CAPEX_{\text{SDN}} = \sum_{i=1}^{n} \text{Cost}_{hwi} + \text{Cost}_{controller}$$

其中：

- $\text{Cost}_{hwi}$ 是节点 i 上通用硬件的成本。
- 成本控制器是集中式SDN控制器的成本。

比较：传统移动自组网与软件定义网络赋能的移动自组网在资本性支出方面的关键区别在于，SDN移动自组网依赖通用硬件，从而降低了每个节点的总体成本。尽管软件定义网络控制器会带来额外成本，但由于每个节点不再需要执行复杂的路由和管理任务，对专用硬件的需求也随之降低，因此资本性支出总额有所减少。随着节点数量增加，SDN移动自组网的成本优势会更加明显，因为SDN移动自组网的单节点硬件成本更低。

B. 运营支出模型

在传统移动自组网中，运营性支出（OPEX）主要来自各节点的维护、配置和监控成本。由于每个节点都要独立处理路由和管理任务，因此维护和更新各节点软件及硬件的成本较高 [3]。传统移动自组网的运营支出总额表示为：

$$\text{OPEX}_{\text{MANET}} = \sum_{i=1}^{n}(C_{\text{manutMANET}_i} + C_{\text{configMANET}_i} + C_{\text{monitoraggioMANET}_i})$$

其中：

- $C_{\text{manutMANET}i}$ 表示节点 i 的维护成本。
- $C_{\text{configMANET}i}$ 表示节点 i 的配置成本。
- $C_{\text{monitoraggioMANET}i}$ 表示节点 i 的监控成本。

在软件定义网络赋能的移动自组网中，集中式控制平面简化了网络管理，降低了配置和维护每个节点所产生的运营成本。软件定义网络控制器负责处理大部分配置和监控任务，从而降低节点层面的运营支出。软件定义网络赋能的移动自组网的运营支出表示为：

$$\text{OPEX}_{SDN} = C_{\text{manutSDN}} + C_{\text{configSDN}} + C_{\text{monitoraggioSDN}} + \sum_{i=1}^{n} C_{\text{manutNode}_i}$$

其中：

- $C_{\text{manutSDN}}$ 是软件定义网络控制器的维护成本。
- $C_{\text{configSDN}}$ 是软件定义网络控制器的配置成本。
- $C_{\text{monitoraggioSDN}}$ 是软件定义网络控制器负责的监控成本。
- $C_{\text{manutNode}i}$ 是节点 i 的维护成本降低值，因为它不再处理复杂的路由任务。

比较：在传统移动自组网中，每个节点都会产生大量的维护、配置和监控成本。相比之下，软件定义网络赋能的移动自组网将这些任务集中交由软件定义网络控制器处理，从而减轻单个节点的运营负担。因此，SDN移动自组网的总体运营支出更低，尤其是在节点数量较多的大型网络中。

C. 效率提升

将 SDN 集成到 MANET 中，不仅可以降低资本性支出和运营支出，还能显著提升网络效率。在传统移动自组网中，路由决策由每个节点在本地做出，这可能导致次优路由并加剧网络拥塞 [5]。SDN 凭借对网络的全局视图，可以优化路由并动态平衡网络负载，从而提高效率。

软件定义网络赋能的移动自组网的效率可表示为：

$$\eta_{\text{SDN}} = \frac{\text{Useful Data}}{Total\ Bandwidth} \times \eta_{\text{optimization}}$$

其中：

- 有效数据$\overline{Total\ Bandwidth}$ 表示数据传输效率。
- $\eta_{\text{优化}} > 1$ 反映了软件定义网络动态优化流量的能力所带来的效率提升。

在传统移动自组网中，缺乏集中式控制可能导致带宽使用效率低下，因为节点必须依靠局部信息进行路由。相比之下，软件定义网络赋能的移动自组网能够优化带宽分配，并确保数据沿最高效的路径传输，从而减少时延并提高吞吐量。

V. SDN移动自组网的可扩展性与安全性

软件定义网络通过动态管理网络资源来增强移动自组网的可扩展性，这对于应对节点数量不断增加至关重要。SDN移动自组网中的集中式控制器能够高效管理网络资源，确保网络在节点数量增加时仍具备可扩展性。安全性也得到提升，因为软件定义网络支持集中式监控并能实时响应威胁，从而降低拒绝服务（DoS）和欺骗攻击等攻击的风险。

A. 可扩展性模型

可扩展性是移动自组网最关键的方面之一，尤其是在节点数量及其移动性都可能快速变化的环境中。

由于网络管理的去中心化特性，传统移动自组网在维持可扩展性方面面临重大挑战。随着网络规模扩大，维持高效的通信路径并确保资源使用达到最优变得越来越困难 [4]。在传统移动自组网中，每个节点都负责路由和管理，这增加了计算能力、带宽消耗和网络拥塞方面的开销。

相比之下，软件定义网络赋能的移动自组网通过软件定义网络控制器引入集中式控制平面，从而显著提升网络的可扩展性。软件定义网络控制器拥有网络的全局视图，能够动态管理资源、优化路由路径，并实时分配带宽。软件定义网络赋能的移动自组网的可扩展性可以通过同时考虑各个节点的容量以及软件定义网络控制器带来的额外可扩展性来建模。

$$\text{Capacity}_{\text{SDN}} = \sum_{i=1}^{n} \text{Capacity}_i + \text{Capacity}_{SDNController}$$

其中：

- 容量$_i$ 表示每个独立节点 i 的容量，包括计算能力、

带宽和路由能力。

• 容量SDN 控制器 是软件定义网络控制器所提供的额外容量，用于管理全局路由和资源分配任务。

B. 软件定义网络赋能的移动自组网的可扩展性优势

• 集中式资源管理：软件定义网络控制器负责管理整个网络的资源，优化带宽和计算能力的分配，从而防止网络拥塞并确保负载均衡 [5]。这种集中式管理可防止密集网络中出现瓶颈，并降低单个节点过载的可能性。

• 高效路由：传统移动自组网通常因缺乏网络全局视图而面临路由低效问题。节点依据局部信息做出路由决策，导致路径次优、丢包增加和时延上升。在软件定义网络赋能的移动自组网中，控制器会动态计算所有节点的最优路由，尽可能减少跳数，并确保路由路径更加高效 [6]。随着节点数量增加，这一优势尤为显著。

• 节点移动性处理：移动自组网的特点是节点具有高度移动性，这常常导致链路中断并需要重新计算路由。在传统移动自组网中，每个节点在本地发起路由发现，从而增加控制消息开销和时延。在软件定义网络赋能的移动自组网中，控制器可以通过快速重新计算路由并更新流表，主动应对链路故障，显著提升网络响应能力，并在高度动态的环境中保持性能 [7]。

• 网络分区与集群化：随着节点数量增加，软件定义网络赋能的移动自组网可以采用分层管理或基于集群的管理。软件定义网络控制器可以将节点划分为多个集群，分别进行管理，并优化集群内通信。这通过降低大规模网络中路由和资源管理的整体复杂度，提升了可扩展性。

• 动态网络切片：软件定义网络控制器可以启用网络切片，对网络进行逻辑划分，为不同类型的流量或服务分配不同资源。这样，网络就能更高效地处理多种应用（例如语音、视频、物联网），在不影响性能的情况下提升可扩展性。

$$容量_{total} = 容量_{clustered} + 容量_{sliced}$$

聚类与网络切片相结合，能够提升整体网络容量，使其超越传统移动自组网的能力上限。通过抽象网络的特定部分并创建虚拟网络，软件定义网络控制器能够有效管理更多节点，而不会使网络资源不堪重负。

## VI. 性能比较：移动自组网与移动自组网+软件定义网络

为评估将软件定义网络集成到移动自组网中的性能优势，我们使用时延、吞吐量、分组投递率（PDR）和控制开销等关键性能指标，对比纯移动自组网与启用软件定义网络的移动自组网的性能。该比较基于理论模型以及有关 SDN-MANET 集成的研究所提供的仿真结果 [7]。

A. 时延

在传统移动自组网中，由于路由具有分布式特性，且节点移动时经常需要进行路由发现，因此时延通常较高。每当路由中断时，路由发现过程都会带来显著时延，尤其是在较大型的网络中。

在启用软件定义网络的移动自组网中，软件定义网络控制器维护网络的全局视图，并主动计算最优路由。这减少了对反应式路由发现的需求，从而降低了整体时延。纯移动自组网中的时延可建模为：

$$\mathrm{L_{MANET}} = \mathrm{L}_{route\ discovery} + \mathrm{L}_{data\ transmission}$$

而在启用软件定义网络的移动自组网中，时延主要取决于控制器重新计算路径的能力：

$$L_{\mathrm{SDN}} = L_{controller\ computation} + L_{data\ transmission}$$

由于 $\mathrm{L_{controller}}$ 计算 通常低于 $\mathrm{L_{route}}$发现，SDN移动自组网能够实现更低的整体时延 [5]。

B. 吞吐量

吞吐量是衡量网络成功传输的数据量的指标。在传统移动自组网中，吞吐量会随着节点数量的增加而下降由于网络拥塞加剧和路由欠佳而增加。

相比之下，SDN 控制器可以动态管理流量，并确保高效利用网络资源。通过优化路由并避开拥塞区域，软件定义网络赋能的移动自组网能够实现更高的吞吐量。总吞吐量可表示为：

$$T_{\mathrm{SDN}} > T_{\mathrm{MANET}}$$

其中，$\mathrm{T_{SDN}}$ 表示软件定义网络赋能的移动自组网中的吞吐量，$\mathrm{T_{MANET}}$ 表示传统移动自组网中的吞吐量。

C. 分组投递率（PDR）

分组投递率（PDR）是成功送达目的地的数据包数量与发送的数据包数量之比。在传统移动自组网中，频繁发生的路由故障以及重新发现路由的需要会导致丢包，从而降低分组投递率（PDR）。

软件定义网络赋能的移动自组网能够通过快速适应网络拓扑变化，并在链路中断时将数据包重新路由至备用路径，从而保持较高的分组投递率。因此：

$$PDR_{\text{SDN}} > PDR_{\text{MANET}}$$

分组投递率的提升得益于软件定义网络控制器能够快速计算备选路由，并更高效地管理流量。

D. 控制开销

控制开销是指网络管理所需控制消息占用的带宽。在传统移动自组网中，可用带宽的很大一部分用于控制消息，例如按需距离矢量路由或优化链路状态路由等协议中的路由发现和维护。在规模较大、密度较高或移动性较强的网络中，控制开销会显著增加。

在软件定义网络赋能的移动自组网中，集中式控制器减少了路由决策所需的控制消息数量，因为路由计算由控制器集中处理，而不是分布在各个节点上。因此，软件定义网络赋能的移动自组网通常具有较低的控制开销，可表示为：

$$O_{\text{SDN}} < O_{\text{MANET}}$$

其中，$O_{\text{SDN}}$ 表示软件定义网络赋能的移动自组网中的控制开销，$O_{\text{MANET}}$ 表示传统移动自组网中的开销。开销的减少使更多带宽可用于数据传输，从而进一步提升网络性能。

E. 性能指标总结

表 2 总结了与传统移动自组网相比，软件定义网络赋能的移动自组网预期实现的性能提升。

表 2. 性能对比：移动自组网与软件定义移动自组网

| 指标 | 移动自组网 | 软件定义移动自组网 |
|---|---|---|
| 时延 | High | 较低 |
| 吞吐量 | 较低 | 较高 |
| 分组投递率（PDR） | 较低 | 较高 |
| 控制开销 | High | 较低 |

从这一比较可以明显看出，将软件定义网络（SDN）集成到移动自组网中，能够显著提升各项关键指标的网络性能。

## VII. 结论

将软件定义网络（SDN）与移动自组网（MANETs）集成，相较于传统移动自组织网络架构具有显著优势，尤其体现在可扩展性、成本效益和安全性方面。SDN 引入的集中式控制平面能够实现更高效的路由、动态资源分配以及对节点移动性的更好管理，而这些正是传统移动自组网面临的关键挑战。

资本支出模型表明，软件定义网络赋能的移动自组网可以依靠单个节点中的通用硬件，同时将计算负担转移至软件定义网络控制器，从而降低总体资本支出。在传统移动自组网中，每个节点都必须独立处理路由和网络管理，导致更高的硬件和软件成本。通过将这些任务卸载至集中式SDN控制器，软件定义网络赋能的移动自组网降低了每个节点的成本，并在节点数量增加时实现了更高的可扩展性。

运营支出模型同样表明，软件定义网络赋能的移动自组网通过集中处理维护、配置和监控任务，能够降低运营成本。传统移动自组网由于网络具有去中心化特性，运营成本较高，因为每个节点都必须独立配置和维护。软件定义网络赋能的移动自组网通过软件定义网络控制器自动执行许多此类任务，减轻了这一负担，并提升了网络的运营效率。

在网络效率方面，软件定义网络控制器维护的全局视图能够实现优化路由和动态负载均衡，从而提高带宽利用率并降低时延。传统移动自组网依赖局部信息进行路由决策，常因路径选择欠佳和路由恢复延迟而效率低下。

可扩展性模型凸显了软件定义网络赋能的移动自组网更有效地处理更大规模网络的潜力。软件定义网络控制器能够集中管理资源，并根据网络拓扑变化进行动态调整，因此即使节点数量增加，网络仍能保持可扩展性。这与传统移动自组网形成对比：由于缺乏集中式控制，传统移动自组网在大型网络中往往会出现网络拥塞和性能下降。

未来工作：本文对软件定义网络赋能的移动自组网与传统移动自组网进行了理论比较，未来的工作应聚焦于实际部署和案例研究，以验证所提出的模型。研究可以探索在各种环境中部署软件定义网络赋能的移动自组网

例如灾难恢复、军事行动和智慧城市等领域，其中网络的可扩展性和灵活性至关重要。此外，还可以进一步优化适用于资源受限环境的软件定义网络控制器，例如针对移动自组网中的此类环境进行优化，以提升这些网络的整体性能。

总之，软件定义网络赋能的移动自组网为下一代移动自组织网络提供了一种极具吸引力的解决方案，在可扩展性、成本效益和网络管理方面均有显著提升。随着软件定义网络技术不断发展，其与移动自组网的集成将日益可行，从而为管理动态、分布式网络提供稳健的框架，适用于广泛的应用领域。

## 利益冲突

作者声明不存在利益冲突。

## 作者贡献

Andrea Piroddi 开展了研究并撰写了论文；Riccardo Fonti 收集并分析了投资数据，并对系统方程进行了建模。所有作者均已批准手稿的最终版本。

## 参考文献

[1] S. Basagni、M. Conti、S. Giordano 和 I. Stojmenovic，《移动自组织网络》，Wiley-IEEE出版社，2004年。[2] C. Perkins，《自组网》，Addison-Wesley出版社，2001年。

[3] X. Li、Y. Zheng、D. Jin 和 L. Zeng，“移动自组网综述”，载于 IEEE 国际网络、架构与存储会议（NAS）论文集，2007年。[4] C. Toh，《移动自组织无线网络：协议与系统》，Prentice Hall出版社，2002年。[5] D. Kreutz、F. M. Ramos、P. Verissimo 等，“软件定义网络：全面综述”，《IEEE学报》，第103卷，第1期，第14–76页，2015年1月。[6] M. Mendonca、B. N. Astuto 等，“软件定义网络综述：可编程网络的过去、现在与未来”，《IEEE通信调查与教程》，第16卷，第3期，第1617–1634页，2014年。[7] J. Qadir、A. Ali 等，“通过软件定义网络管理未来互联网”，《IEEE通信调查与教程》，第18卷，第3期，第2167–2194页，2018年。[8] S. Corson 和 J. Macker，“移动自组网（移动自组网）：路由协议性能问题与评估考量”，IETF RFC 2501，1999年。

[9] N. Abbas、Y. Zhang、A. Taherkordi 和 T. Skeie，“无线传感器网络中的聚类与路由协议：性能比较”，Ad Hoc Networks，第12卷，第45–67页，2015年。[10] Y. Yayeh、H.-P. Lin、G. Berie、A. B. Adege、L. Yen 和 S. -S. Jeng，“利用深度学习预测移动自组网中的移动性”，载于《2018年IEEE应用系统发明国际会议（ICASI）论文集》，日本千叶，2018年，第1203–1206页。doi: 10.1109/ICASI.2018.8394504[11] R. Fonti，“软件定义网络架构的经济影响分析：成本、收益与商业模式”，博洛尼亚大学，2024年。

版权所有 © 2025年，作者所有。本文为开放获取文章，依据知识共享署名许可协议发布。该协议允许在任何媒介中不受限制地使用和传播，但须妥善注明原始作品出处（CC BY 4.0）。