软件定义网络驱动的移动自组网与物联网创新：迈向更智能网络之路

Andrea Piroddi 和 Riccardo Fonti

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：andrea.piroddi@unibo.it（A.P.）；riccardo.fonti@studio.unibo.it（R.F.）* 通讯作者

摘要—移动自组网（MANET）和物联网网络运行于去中心化的动态环境中，因此非常适用于缺乏传统基础设施的场景。然而，由于其去中心化特性和资源约束，这些网络面临低效路由、可扩展性有限以及安全漏洞等挑战。本文探讨将软件定义网络（SDN）作为一种统一解决方案加以整合，利用其集中式控制和网络可编程性来改善路由、资源管理和安全性。本文通过数学模型评估整合软件定义网络对资本支出（CAPEX）、运营支出（OPEX）及性能指标的影响。结果表明，采用软件定义网络增强的移动自组网和物联网网络具有更出色的可扩展性、更低的时延、更高的吞吐量和更少的丢包，尤其是在动态的大规模环境中。尽管软件定义网络会带来计算开销，但它显著提升了路由效率、资源优化能力和适应性。所提出的框架提供了一种稳健且可扩展的解决方案，使网络架构能够高效管理不断增长的节点密度、动态拓扑和高数据流量。该方法确保了韧性，因此非常适合满足现代大规模应用对性能和可靠性的需求。

关键词—移动自组网（MANETs）、物联网（IoT）网络、软件定义网络（SDN）、可扩展性、网络效率、成本优化、资源管理

I. 引言

移动自组网（MANETs）和物联网（IoT）网络是动态组建、无需基础设施的无线系统，旨在高度灵活的环境中运行。这些网络尤其适用于灾难恢复、军事行动、智慧城市和临时活动覆盖等关键应用 [1]。尽管用途广泛，移动自组网和物联网网络仍面临诸多挑战，例如低效路由、可扩展性问题，以及易受欺骗攻击和拒绝服务攻击（DoS）等攻击威胁 [2, 3]。

软件定义网络（SDN）的最新进展展现出巨大潜力，能够应对这些挑战，具体方式是引入集中式控制、可编程性和动态资源分配 [4]。通过将控制平面与数据平面解耦，软件定义网络（SDN）能够实现更高效的路由、提高带宽利用率并增强安全性。这些特性在移动自组网和物联网网络等动态环境中尤为有利，因为在这些环境下，传统分布式协议往往难以在拓扑结构不断变化且资源受限的情况下维持性能 [5, 6]。

尽管软件定义网络已在固定网络和基于云的网络中得到广泛研究，但其在移动自组网和物联网系统中的应用仍是一个不断发展的研究领域。新兴研究表明，软件定义网络能够提升这些网络的可扩展性、能源效率和移动性管理 [1, 3]。然而，将软件定义网络集成到移动自组网和物联网网络中也带来了新的挑战，包括管理集中式控制的开销、确保与资源受限设备的兼容性，以及调整SDN控制器以适应高度动态且异构的环境 [2, 6]。

为克服这些局限性，本文提出了一种将软件定义网络与移动自组网及物联网网络相融合的统一框架。该框架包含一个数学模型，用于评估软件定义网络集成对资本支出（CAPEX）、运营支出（OPEX）以及时延、吞吐量和丢包等性能指标的影响。此外，本文还探讨了应对混合移动自组网-物联网系统中的资源约束、动态拓扑管理和可扩展性问题的策略。

本文其余部分的结构安排如下：
第二节详细讨论移动自组网和物联网网络的特征与挑战，第三节介绍所提出的软件定义网络集成框架，随后第四节介绍数学建模方法，第五节呈现结果与分析，第六节总结全文并指出未来研究方向。

II. 移动自组网与物联网：特征与挑战

移动自组网（MANET）和物联网网络的独特之处在于，它们能够在不依赖固定基础设施的情况下构建去中心化无线系统。这种能力使它们

稿件于2024年12月11日收到；2025年1月2日修订；2025年1月23日接受；2025年3月20日发表。

特别适用于军事行动、灾难恢复、临时活动覆盖、智慧城市和工业自动化等应用场景 [1, 3]。然而，它们的灵活性也带来了若干挑战，尤其是在网络控制、可扩展性和资源管理方面。

移动自组网和物联网网络的一个显著特征是其拓扑结构不断变化。在移动自组网中，节点持续移动，导致动态路由路径频繁建立和断开 [2]。同样，物联网网络中的设备也常常动态地加入或离开网络，或根据上下文切换运行状态 [4]。在移动自组网中，每个节点既可以充当主机，也可以充当路由器，为其他节点转发分组。这种双重角色要求路由协议能够快速适应拓扑结构的变化。AODV（按需距离矢量路由）和 DSR（动态源路由）等传统协议能够满足这些要求，但在大规模、高动态性的网络中会带来显著开销 [5]。

传统移动自组织网络和许多物联网部署缺乏集中式控制，这往往会导致路由欠佳、时延增加，并在重新计算路由时发生丢包。在移动自组网中，节点的高移动性会加剧这些挑战，因为节点快速移动会造成频繁的链路故障。同样，在物联网网络中，边缘设备计算资源有限，可能会妨碍路由协议的性能，尤其是在流量很大的密集环境中 [6]。近期研究表明，将机器学习融入路由协议可以提高其对动态拓扑的适应能力，但此类方法计算开销较大，可能并不适用于所有物联网设备 [5]。

可扩展性是移动自组网和物联网网络面临的另一项重大挑战。随着节点数量及其交互不断增加，维护准确的路由表和管理网络资源会变得愈发复杂，复杂度甚至呈指数级增长 [3]。在物联网网络中，新增成千上万乃至数百万台设备会加剧这些问题，往往导致干扰、拥塞和资源耗尽。此外，广播风暴——即过多的控制消息涌入网络——会耗尽资源，显著降低性能 [6]。

为应对可扩展性挑战，人们提出了分层路由、基于区域的路由和聚类等解决方案。然而，这些方法通常会带来额外开销，用于管理簇或区域。近期的进展，包括软件定义网络赋能的移动自组织网络和物联网系统，利用集中式控制器动态管理资源，并减轻网络扩展带来的影响 [4]。尽管软件定义网络在提升可扩展性方面前景可期，但仍需谨慎管理控制器与节点间的通信开销，尤其是在拥有异构设备和协议的物联网网络中。

移动自组网和物联网设备都在资源受限的环境中运行，面临电池寿命、处理能力和带宽方面的限制 [7]。在移动自组网中，电池消耗尤为关键，因为节点通常依赖有限的电源，而这些电源会因持续的路由更新和分组转发而迅速耗尽。同样，物联网设备，尤其是部署在偏远或难以到达地区的设备，必须优先考虑能源效率，以确保长期运行 [3]。

带宽限制进一步增加了移动自组网和物联网网络中资源管理的难度。共享无线链路往往会导致竞争，从而降低吞吐量并增加时延 [2]。通过负载均衡和拥塞控制等技术进行有效的带宽管理，对于维持性能至关重要。跨层优化和基于软件定义网络的资源管理等新兴解决方案，已展现出显著潜力，能够根据网络的实时需求动态分配资源。例如，跨层优化方法整合来自网络协议栈多个层的信息，以自适应方式管理资源，从而提升动态环境下的可靠性 [1, 3]。类似地，基于软件定义网络的资源管理框架支持集中式控制和实时决策，Shieh 等人等近期研究便证明了这一点 [8]，：研究显示，这些框架提升了物联网网络中的可扩展性和资源分配能力。这些方法共同提升了网络可靠性和运行效率，尤其适用于流量和资源需求不断波动的场景。

解决这些挑战对于提升移动自组网和物联网网络的可靠性与性能至关重要。尽管传统解决方案（如节能高效路由协议和聚类）已被证明有效，但引入软件定义网络原则可为管理动态、资源受限网络提供更稳健且可扩展的框架。软件定义网络凭借其可编程性和集中式控制，被视为一套强大的工具，可为管理动态、资源受限网络提供灵活的框架。这一特性使软件定义网络能够有效适应物联网和移动自组网环境的独特需求，提升网络效率和韧性。例如，Shieh 等人近期的研究 [8] 表明，基于软件定义网络的管理框架能够优化物联网网络中的资源分配，提升可扩展性并降低时延。同样，在移动自组网中，软件定义网络集中控制并动态调整路由协议的能力已被证明能够在流量条件变化时提升网络性能，Wheeb 和 Al-Jamali [9]的研究便展示了这一点。这些案例研究凸显了软件定义网络的适应性，并展现了其应对物联网和移动自组网所面临特定挑战的潜力。

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）采用集中式网络控制方式，这与传统移动自组织网络和许多物联网网络的分布式特性形成鲜明对比。通过将控制平面与数据平面解耦，软件定义网络能够实现集中式决策，并对整个网络进行全局控制。这

架构使 SDN控制器能够管理移动自组网和

物联网网络，从而提升性能、可扩展性和适应性[1, 4]。

A. 基于软件定义网络的集中式路由

在传统移动自组织网络中，每个节点使用按需距离矢量路由（AODV）或优化链路状态路由（OLSR）等分布式算法，在本地作出路由决策。由于路由频繁变化以及路由发现带来的开销，这些协议在高度动态的环境中往往难以应对。同样，物联网网络，尤其是包含异构设备的网络，也面临着在资源约束下维持高效路由的挑战 [2]。

软件定义网络引入了一个集中式控制器，该控制器根据对网络拓扑的全面视图计算并优化路由。考虑一个包含 nn 个节点的网络，其中每个节点的连接度为 d_i，i = 1,2,...,n。在传统分布式系统中，路由复杂度会随着 d_i 成比例增长，因为每个节点都要独立维护到其邻居的路由。此类系统中的平均路径成本 C_{path} 可建模为：

$$C_{path}=\frac{1}{n}\sum_{i=1}^{n} C_i \quad (1)$$

其中，C_i 表示节点 i 的路径成本。随着节点移动性增强，由于需要频繁重新计算路由以及广播路由更新所带来的开销 [5]，C_i 会增大。

在基于软件定义网络的系统中，控制器集中计算路由路径，从而最大限度地减少冗余路由发现并降低总体路径成本。这种集中式决策可表示为：

$$C_{SDN-path}=\min_{P\in\mathcal{P}}\sum_{i=1}^{n} w_i \cdot d_i \quad (2)$$

其中，w_i 是根据节点 i 处的流量负载或优先级分配的权重，PP 表示所有可能的路径。软件定义网络控制器根据实时流量状况动态调整路由，从而降低时延和丢包率，近期关于软件定义网络赋能的物联网网络的研究[1, 3]也展示了这一点。

B. 基于软件定义网络的移动自组网与物联网的资源整合

软件定义网络（SDN）、移动自组网和物联网的融合，为管理动态的资源受限网络带来了新的机遇。物联网设备的计算能力、带宽和能源资源有限，因而面临独特挑战；软件定义网络提供的集中式控制和可编程性则有助于应对这些挑战。

在传统的物联网部署中，每台设备通常都扮演固定角色，并依靠预配置设置进行通信和资源分配。将物联网节点集成到移动自组网中，会因节点的动态拓扑结构和移动性而增加复杂度。软件定义网络赋能的移动自组织网络通过集中控制物联网节点来提供解决方案，从而实现：

动态资源分配：软件定义网络控制器会监控实时网络状况，例如带宽使用、能量水平以及物联网节点的移动模式。通过优化资源分配，控制器可确保网络性能均衡，并延长资源受限型物联网设备的运行寿命。

高效路由：移动自组网中的物联网节点通常会生成小型、周期性的数据包，这可能导致传统路由协议效率低下。软件定义网络控制器凭借其全局视图，可动态调整路由路径以最大限度地降低时延和丢包率，尤其是在移动性较强或干扰较大的场景中。

能源管理：延长物联网设备的电池寿命至关重要。软件定义网络控制器可以实施能量感知路由策略，优先选择能源储备充足的节点所在路径，同时避免使低能量设备负担过重。

在物联网设备与移动节点共存的混合MANET-IoT环境中，软件定义网络控制器可以促进以下优化：

流量区分：物联网应用通常会产生多种流量类型，例如周期性感知器更新、事件驱动警报或多媒体流。软件定义网络控制器可以实施流量优先级策略，确保健康监测警报等关键物联网数据以最小时延传输。

网络切片：通过将网络在逻辑上划分为多个切片，软件定义网络控制器可以根据应用需求分配资源。例如，低时延切片可专用于自动驾驶车辆等实时物联网应用，而高吞吐量切片则支持多媒体流。

负载均衡：混合网络常因节点密度或应用需求各异而出现流量分布不均。软件定义网络控制器可在多条路径间均衡负载，从而减少拥塞并确保资源得到公平利用。

在物联网设备与移动节点共存的混合MANET-IoT环境中，软件定义网络控制器可实现以下优化：

流量区分：物联网应用通常会产生多种类型的流量，例如周期性感知器更新、事件驱动警报或多媒体流。软件定义网络控制器可实施流量优先级策略，确保关键物联网数据（例如健康监测警报）以最小延迟传输。

网络切片：通过将网络逻辑划分为多个切片，软件定义网络控制器可根据应用需求分配资源。例如，可为自动驾驶车辆等实时物联网应用专门配置低时延切片，同时由高吞吐量切片支持多媒体流。

负载均衡：混合网络常因节点密度或应用需求各异而出现流量分布不均。软件定义网络控制器可在多条路径间均衡负载，从而减少拥塞并确保资源得到公平利用。

软件定义网络赋能的移动自组织网络与物联网的融合，在各个领域都有广泛的应用前景：

智慧城市：在动态变化的环境中管理基于物联网的基础设施，例如智能照明、交通管制和公共安全系统。

灾难恢复：在需要快速部署和灵活应变的紧急情况下，协调环境传感器、无人机等物联网设备。

工业物联网（IIoT）：将移动机器人和物联网传感器整合到统一的集中管理网络中，以提升制造系统的性能。

C. 软件定义网络中的资源管理

在移动自组网和物联网网络等资源受限的环境中，高效的资源管理至关重要，因为设备通常面临电力、带宽和计算能力方面的限制 [7]。传统移动自组织网络依赖去中心化协议，由每个节点独立管理自身资源。这种方式往往会导致资源利用率低下和网络拥塞。

在软件定义网络中，控制器可以根据网络的实时需求动态分配资源。设 B_{total} 表示可用带宽总量，P_{total} 表示可用电力总量。分配成本 C_{alloc} 可建模为：

$$C_{alloc} = \sum_{i=1}^{n}\left(\frac{B_i}{B_{total}} + \frac{P_i}{P_{total}}\right) \quad (3)$$

其中，B_i 和 P_i 分别表示分配给节点 i 的带宽和电力。软件定义网络控制器旨在最小化 C_{alloc}，确保资源得到高效利用，同时避免任何单个节点过载 [4]。

此外，SDN控制器会实施负载均衡技术，将流量分配到多条路径上，以防止拥塞。这种方法对物联网网络尤其有益，因为如果管理不当，资源异构性可能会造成瓶颈 [3]。

D. 软件定义网络的安全增强

由于去中心化特性以及缺乏可信中央机构，安全性仍然是移动自组网和物联网网络面临的一项重大挑战。传统移动自组织网络和物联网系统极易受到多种攻击，包括欺骗攻击、黑洞攻击和拒绝服务攻击（DoS） [7]。在这些去中心化网络中，各节点独立实施安全协议，这可能导致协议不一致，从而产生可被利用的漏洞，并可能引发安全漏洞。

采用软件定义网络（SDN）为应对这些安全挑战提供了一个颇具前景的解决方案：它构建了一个集中式框架，使软件定义网络控制器能够在整个网络中统一实施安全策略。这种集中化有助于更有效地协调安全措施，减少传统去中心化方法固有的不一致性。借助 SDN，控制器可以监控网络活动、检测异常并采取相应对策，从而动态管理安全风险。

为了更全面地了解 SDN 如何降低安全风险，我们可以将网络中的安全风险 S_{risk} 建模为：

$$S_{risk} = \sum_{i=1}^{n} P(V_i)\, I(V_i) \quad (4)$$

其中，$P(V_i)$表示漏洞 V_i 被利用的概率，$I(V_i)$表示此类利用行为可能造成的影响。求和项（$\sum_{i=1}^{n}$）汇总了所有已识别漏洞（n）所带来的风险。这与风险分析方法相一致，这些方法会评估多种风险对系统或网络造成的累积影响。通过将概率与影响相乘来计算风险，是 ISO 31000 和 NIST 800-30 等风险管理框架广泛采用的方法。这些框架强调，风险取决于事件发生的可能性及其后果，这与我们的公式相一致。通过将概率与影响相乘来计算风险，是 ISO 31000 和 NIST 800-30 等风险管理框架广泛采用的方法。这些框架强调，风险取决于事件发生的可能性及其后果，这与我们的公式相一致。将公式（4）构造为求和形式，可以综合所有漏洞，得出系统的总体风险水平。这种整体视角确保得出的风险评分 S_{risk} 能够反映系统中所有漏洞带来的总体威胁。该公式在以下情境中尤为适用

例如物联网和移动自组网环境，其中存在多种漏洞；了解这些漏洞的累积风险，对于设计有效的缓解策略至关重要。该模型凸显了持续评估漏洞及其被利用后可能造成的后果的必要性。

软件定义网络的集中式控制带来了多项关键安全优势。首先，控制器可以动态隔离受感染节点，防止它们影响网络中的其他部分。这种快速隔离对于缓解黑洞攻击和拒绝服务攻击等尤为重要，因为恶意节点会试图扰乱正常运行。此外，软件定义网络控制器能够将流量从易受攻击的区域改道，从而最大限度地降低潜在安全漏洞的影响。这一功能对于移动自组网尤其有用，因为节点移动性会不断改变网络拓扑结构，使传统安全措施难以实施。

此外，软件定义网络能够将异常检测和入侵防御系统等先进安全技术直接集成到网络基础设施中。通过实时分析流量模式和节点行为，软件定义网络控制器可以识别表明攻击正在进行的异常活动。这些技术可以根据物联网和移动自组网环境的特定安全需求进行定制，确保网络能够抵御不断演变的威胁。

尽管软件定义网络的集中化特性带来了显著的安全优势，但也引入了新的挑战。例如，软件定义网络控制器本身可能成为攻击目标，因此必须严格维护其安全性。确保控制器的

与网络设备之间的通信完整性，并防止未授权访问，是软件定义网络安全的关键方面。此外，在高度动态且异构的物联网环境中，设备的能力各不相同，要实施既有效又可扩展的安全措施是一项复杂的任务。

近期研究考察了基于SDN的安全措施在移动自组网和物联网网络中的有效性，表明这些措施有望增强网络抵御常见攻击的韧性。例如，研究显示，与传统系统相比，基于SDN的入侵检测系统能够显著提高攻击检测能力并缩短响应时间 [1]。此外，SDN能够对流量进行实时、细粒度的控制，从而更精准地缓解网络威胁。

为进一步强化安全框架，将SDN与其他安全技术相结合的混合方法已展现出应用前景，例如结合基于机器学习的异常检测，或利用区块链实现安全数据传输。这些方法发挥了SDN集中式控制的优势，以及机器学习或区块链的自适应能力，从而进一步提升安全性。

总之，尽管SDN为增强移动自组网和物联网网络的安全性提供了有力工具，但其成效取决于安全措施的周密设计与实施。未来研究应探讨SDN与其他安全框架的集成，评估基于SDN的安全措施在真实环境中的表现，并衡量大规模部署中安全性与性能之间的权衡 [1, 3]。

E. 使用 OpenFlow 实现自适应更新

OpenFlow 等协议是软件定义网络框架的核心，可实现控制平面与网络设备之间安全、动态的通信。在移动自组网和物联网场景中，OpenFlow 通过提供高效的实时路径优化机制，帮助降低路由更新的复杂性。传统网络中更新路由表所需的时间T_{update} 可表示为：

$$T_{update} = T_{discovery} + T_{propagation} + T_{reconfig} \quad (5)$$

其中， $T_{discovery}$是路由发现所需的时间，$T_{propagation}$ 是更新传播所需的时间，$T_{reconfig}$ 是重新配置受影响节点所需的时间。软件定义网络控制器利用网络的全局视图，缩短发现和传播时间，从而减少 T_{update} [7]。

将软件定义网络融入移动自组网和物联网网络，有助于应对低效路由、资源约束和安全漏洞等关键挑战。软件定义网络提供的可编程性和集中式控制，使网络能够根据不断变化的条件动态调整，从而确保在复杂的异构环境中实现更高效、更安全的运行。

F. 物联网特有挑战与软件定义网络解决方案

尽管软件定义网络与移动自组网和物联网的融合带来了显著优势，但能源效率、间歇性连接和协议兼容性等物联网特有挑战仍需重点关注。这些挑战在物联网环境中尤为关键，因为必须解决资源约束和运行限制，才能确保网络性能可持续且高效。

能源效率是物联网网络中最紧迫的问题之一，因为许多物联网设备依靠电池供电，需要长时间运行。在传统物联网系统中，设备的能源储备往往有限，而低效的通信和资源管理会导致电池寿命迅速缩短。软件定义网络通过支持动态的能效感知路由决策来应对这一挑战。软件定义网络控制器可以实时监控物联网设备的功耗，并优先选择能够最大限度降低能耗的路由路径。这种动态能源管理不仅能延长设备的电池寿命，还有助于优化整体网络性能。软件定义网络能够调整流量负载并减少不必要的数据传输，这对于能源受限的物联网环境至关重要。

在物联网网络中，尤其是在移动或偏远环境中，设备常因网络拓扑变化、干扰或物理障碍而遭遇间歇性连接。这些中断会影响通信的可靠性，导致丢包和延迟。传统物联网系统通常依赖静态路由协议，难以应对这种多变的情况。另一方面，软件定义网络得益于其集中式控制，控制器能够持续监测网络状况，并调整路由路径以避开连接不佳的区域。通过实时调整，软件定义网络可确保物联网设备即使在充满挑战的环境中也能保持最佳连接。这种动态路由能力显著增强了物联网网络应对间歇性连接的稳健性和韧性。

物联网网络通常包含多种使用不同通信协议的设备，例如 MQTT、Zigbee 和 CoAP。这些协议针对特定应用而设计，在数据传输、可靠性和可扩展性方面有不同的要求。物联网协议的多样性可能造成兼容性问题，使网络集成和管理更加复杂。软件定义网络通过提供一个可管理并转换不同协议的集中式框架，帮助克服这些挑战。软件定义网络控制器可编程为检测每台设备的协议要求，并采用适当的通信策略，以确保无缝互操作性。例如，软件定义网络可以弥合协议鸿沟并优化数据交换，从而实现基于 MQTT 的设备与基于 Zigbee 的设备之间的高效通信。

总之，尽管将软件定义网络与物联网相结合带来了巨大的机遇，但解决物联网特有挑战至关重要，例如能源效率、间歇性连接和协议兼容性。通过

利用软件定义网络的集中式控制和可编程性，可以缓解这些挑战，从而实现更高效、更可靠且可扩展的物联网网络。未来研究应继续探索软件定义网络与物联网在多样化应用场景中的融合，重点关注节能型网络设计、跨协议的无缝通信以及稳健的连接解决方案。

IV. 成本与效率的数学建模

可以通过数学模型量化资本支出（CAPEX）和运营支出（OPEX），从而评估将软件定义网络与移动自组网和物联网网络相结合所带来的经济优势。本节比较传统移动自组织网络、物联网网络和软件定义网络赋能的移动自组织网络的资本支出与运营支出模型，重点说明基于软件定义网络的方法在这两类环境中的成本优势。为验证这些模型，我们参考了相关案例研究，例如在军事通信网络中集成软件定义网络；在动态、大规模环境下，软件定义网络提升了网络可扩展性、效率和资源管理能力，而这也是移动自组网和物联网网络共同面临的挑战 [1, 4]。

如图1所示，与传统网络相比，在移动自组网和物联网网络中引入软件定义网络可显著降低资本支出和运营支出。采用通用硬件和集中式控制能够降低总体成本，尤其是在网络规模和设备数量不断增长时。军事通信网络的案例展示了软件定义网络在降低资本和运营开支方面的实际效益，使其成为大规模物联网部署（例如智慧城市或工业自动化）的可行解决方案 [3, 4]。

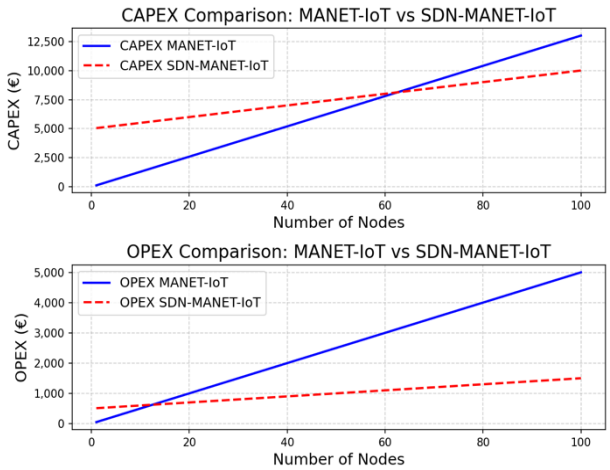


图1. 传统移动自组网-物联网与软件定义网络赋能的移动自组网-物联网框架之间的资本性支出和运营支出效率比较。

A. 资本支出模型

在传统移动自组织网络和物联网网络中，需要使用专用硬件，例如路由器、交换机及其他能够独立管理每个节点上的路由、安全性和网络功能。这种分布式架构意味着每个节点都必须配备充足的计算资源和专用硬件 [7, 9, 10]。因此，传统网络的资本性支出高度取决于节点数 n，以及每个节点上硬件组件的成本。

对于传统移动自组网和物联网网络，总资本性支出为：

$$CAPEX_{MANET/IoT} = \sum_{i=1}^{n} Cost_{hw_i} + \sum_{i=1}^{n} Cost_{sw_i} \quad (6)$$

其中，$Cost_{hw_i}$ 表示节点 i 的硬件成本，包括专用路由器、传感器和其他设备；$Cost_{sw_i}$ 表示节点 i 的软件成本，用于路由、安全性和管理功能[11]。

相比之下，软件定义网络赋能的移动自组网和物联网网络采用集中式控制平面。该架构减少了对每个节点专用硬件的依赖，使其能够使用通用硬件运行，而软件定义网络控制器则负责处理复杂的计算任务，包括路由和资源管理。

软件定义网络的资本性支出为：

$$CAPEX_{SDN} = \sum_{i=1}^{n} Cost_{hw_i} + Cost_{controller} \quad (7)$$

其中，$Cost_{controller}$ 是负责管理网络的集中式软件定义网络控制器的成本。这种集中式方法无需在各个节点配备专用路由设备和计算资源，从而降低了单个节点的硬件成本；大型物联网应用（如智能电网或环境监测）已证明了这一点 [4]。

传统移动自组织网络与软件定义网络赋能的移动自组织网络在资本性支出方面的主要区别在于，后者依赖通用硬件，从而显著降低了单个节点的成本。虽然软件定义网络控制器会带来额外成本，但由于无需在每个节点配备专用硬件，总体资本性支出更低。例如，近期在军事通信网络中部署软件定义网络的案例表明，软件定义网络有助于优化网络可扩展性，降低单个节点的硬件成本，因为每个节点都使用通用计算设备，而软件定义网络控制器则集中执行网络管理任务 [4]。随着网络规模的扩大，软件定义网络的成本优势

-在支持软件定义网络的配置中，每个节点的硬件成本更低，因此移动自组网的优势日益明显。

B. 运营支出模型

在传统移动自组织网络和物联网网络中，运营支出（OPEX）主要由每个节点的维护、配置和监控成本构成。由于每个节点都独立处理路由和管理任务，因此维护和更新各节点的软件与硬件会产生高昂成本 [3, 10]。传统网络的总运营支出可表示为：

$$OPEX_{MANET/IoT} = \sum_{i=1}^{n}\left(C_{manutMANET/IoT_i} + C_{monitoringMANET/IoT_i} + C_{configMANET/IoT_i}\right) \quad (8)$$

其中，$C_{manutMANET/IoT_i}$是节点 i 的维护成本，$C_{configMANET/IoT_i}$ 是配置成本，$C_{monitoringMANET/IoT_i}$ 表示节点 i 的监控成本。这种去中心化方法会导致运营成本高昂，尤其是在设备数量众多的物联网网络中。公式（8）是我们在原始分析中推导出的，用于量化管理移动自组织网络/物联网网络的运营支出（OPEX）。推导过程基于成熟的成本建模原则，并针对这些网络独特的运营需求进行了调整。求和项涵盖了网络中全部 n 个节点的总运营支出，确保公式计入每个网络组件产生的累计成本，从而体现移动自组网和物联网系统的分布式特性。我们独立推导了该公式，以满足构建全面运营支出模型的需求；该模型考虑了移动自组网和物联网系统等动态、资源受限网络所面临的独特挑战。它整合了与这些环境相关的特定成本因素，因此有别于传统网络中的通用运营支出模型。我们独立推导了该公式，以满足构建全面运营支出模型的需求；该模型考虑了移动自组网和物联网系统等动态、资源受限网络所面临的独特挑战。它整合了与这些环境相关的特定成本因素，因此有别于传统网络中的通用运营支出模型。

在软件定义网络赋能的移动自组织网络和物联网网络中，集中式控制器简化了网络管理，降低了与单个节点配置和维护相关的运营成本。基于软件定义网络的网络的运营支出为：

$$OPEX_{SDN} = C_{manutSDN} + C_{configSDN} + C_{monitoringSDN} + \sum_{i=1}^{n} C_{manutNode_i} \quad (9)$$

其中，$C_{manutSDN}$ 为软件定义网络控制器的维护成本，$C_{configSDN}$ 为配置成本，$C_{monitoringSDN}$ 为由控制器负责的监控成本。术语 $C_{manutSDN}$ 表示节点 i 所减少的维护成本，因为它不再需要执行复杂的路由或管理任务。这种集中式控制显著降低了运营支出，尤其是在大规模物联网应用中 [7]。

C. 效率与可扩展性

软件定义网络不仅降低了资本性支出和运营支出，还通过优化路由和资源分配提升了网络的整体效率。在传统移动自组织网络和物联网网络中，缺少集中式控制可能导致路由欠佳和带宽使用效率低下。软件定义网络能够动态、集中地分配资源，从而提升网络性能。软件定义网络的效率可表示为：

$$\eta_{SDN} = \frac{Useful\ Data}{Total\ Bandwidth} \cdot \eta_{optimization} \quad (10)$$

其中，$\frac{Useful\ Data}{Total\ Bandwidth}$表示数据传输效率，$\eta_{optimization}$>1 则体现了软件定义网络动态优化流量所带来的效率提升。公式（10）是独立推导得出的，用于根据带宽利用率及优化策略的影响量化基于软件定义网络的网络的效率。以下是该公式的推导依据。该公式是我们原创分析的一部分，旨在满足软件定义网络环境对专用效率指标的需求。传统效率模型往往忽视软件定义网络的可编程性和优化能力所发挥的关键作用。通过将这些方面纳入公式，我们力求体现软件定义网络对网络性能的独特贡献。在软件定义网络中，可编程性支持动态流量管理和资源分配，直接影响带宽的有效利用。纳入$\eta_{optimization}$，明确地将效率指标与软件定义网络带来的益处联系起来，使该公式特别适用于这类网络。

在软件定义网络赋能的移动自组织网络和物联网网络中，集中式控制通过动态调整网络路径和均衡负载，提高带宽利用率、缩短延迟并提升吞吐量，从而应对可扩展性挑战 [3, 4]。将软件定义网络集成到移动自组网和物联网网络中，可大幅降低资本性支出和运营支出，同时提升网络效率、可扩展性和安全性。通过集中控制，软件定义网络能够实现更好的资源管理和更高效的路由，使其成为移动自组网和物联网系统等大规模动态网络的理想解决方案；在这些网络中，传统方法难以应对移动性、资源有限和设备密度不断增加所带来的挑战。

相比之下，基于SDN的移动自组网和物联网网络通过软件定义网络控制器引入集中式控制平面，显著提升了网络的可扩展性。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。这减少了与分布式控制相关的开销，使网络能够更有效地扩展，尤其是在大规模物联网部署中，因为设备数量和管理复杂性是关键挑战 [3, 4]。

为对基于SDN的移动自组网和物联网网络的可扩展性进行建模，我们同时考虑各节点的容量以及软件定义网络控制器引入的额外容量，计算了网络容量。然而，该容量会受到开销的影响；随着节点数量增加，以及节点的移动性导致拓扑更新更加频繁，开销也会随之增加。模型表示为：

$$Capacity_{SDN} = \sum_{i=1}^{n} Capacity_i + Capacity_{SDNcontroller} - Overhead \quad (11)$$

其中：

- $Capacity_i$ 表示每个单独节点 i 的容量，包括计算能力、带宽和路由能力。
- $Capacity_{SDNcontroller}$ 是软件定义网络控制器提供的额外容量，该控制器负责管理全局路由和资源分配任务。
- $Overhead$ 表示软件定义网络控制器与节点之间交换控制消息所产生的流量。这项开销会随着节点数量和拓扑更新的复杂程度而增加，在设备密度高且通信频繁的物联网网络中尤为显著 [1, 7]。

开销通过一种算法计算得出，该算法考虑了节点之间的距离、节点数量及其移动性。具体如下：

- 每对节点产生的控制分组数量与其间距离和移动性成正比。
- 以比特为单位的开销计算如下：

$$Overhead = Number\ of\ Packets \times Packet\ Size\ (bits) \quad (12)$$

其中，假定分组大小为 512 比特（64 字节）。总开销会从网络容量中扣除，以模拟其对资源使用和带宽的影响。在物联网环境中，考虑这项开销尤为重要，因为设备通常资源受限，而高效通信至关重要 [3]。

根据结果可以观察到，随着节点数量增加，基于SDN的移动自组网的时延高于传统移动自组网。这是由于控制开销增加以及路由决策集中化所致。然而，基于SDN的移动自组网中的丢包率显著降低，这得益于软件定义网络控制器提供的优化路由路径和更好的资源管理。此外，基于SDN的移动自组网的吞吐量始终优于传统移动自组网，因为集中式控制减少了碰撞，并提高了带宽使用效率 [4, 12]。

图2展示了该模型的结果。该图比较了传统移动自组网与软件定义网络赋能的移动自组织网络的容量，突出了随着节点数量增加，开销如何
降低后者的有效容量。尽管如此，基于SDN的移动自组网所提升的吞吐量和降低的丢包率表明，其在动态大规模网络中实现更可靠通信方面具有潜力，尤其是在可扩展性至关重要的物联网环境中 [7]。

图3展示了传统移动自组网和软件定义网络赋能的移动自组织网络的丢包趋势。图表凸显了软件定义网络赋能的移动自组织网络在网络节点数量增加时仍能保持较低丢包率的稳健性。这表明，它适用于智慧城市等对高可靠性有要求的应用场景
或工业物联网系统；在这些场景中，即使设备密度不断增加，稳定的性能也至关重要 [3, 4]。

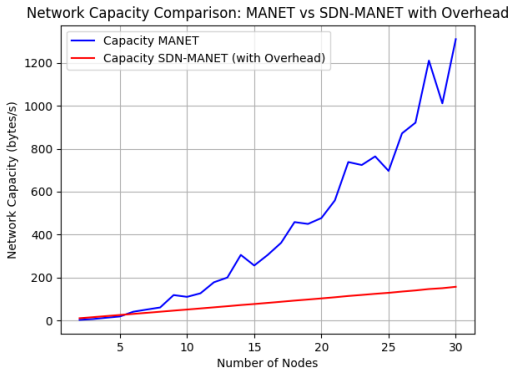


图2. 随着节点数量增加，移动自组网与软件定义移动自组网的网络容量对比。

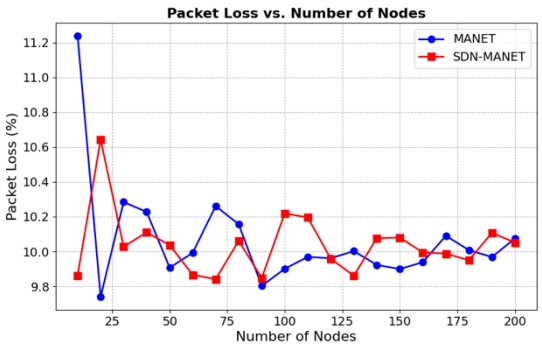


图3. 随着节点数量增加，移动自组网与软件定义移动自组网的丢包趋势。

图4展示了移动自组网和软件定义移动自组网配置中，时延随节点数量增加的变化情况。

图5展示了随着网络规模扩大，软件定义网络控制器的性能变化。

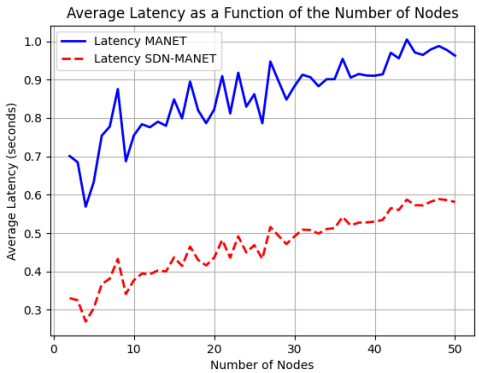


图4. 时延对比。

软件定义网络控制器可优化整个网络中的资源分配，防止拥塞并确保高效的负载均衡。随着节点数量增加，这种集中式方法在缓解资源约束方面尤其有效，因此对

移动自组网和物联网网络都非常有益，因为其中的设备通常以有限的计算能力和能量运行 [7，13]。控制器基于网络的全局视图动态计算最优路由，最大限度地减少跳数，并与传统移动自组织网络相比提高吞吐量 [4, 12]。

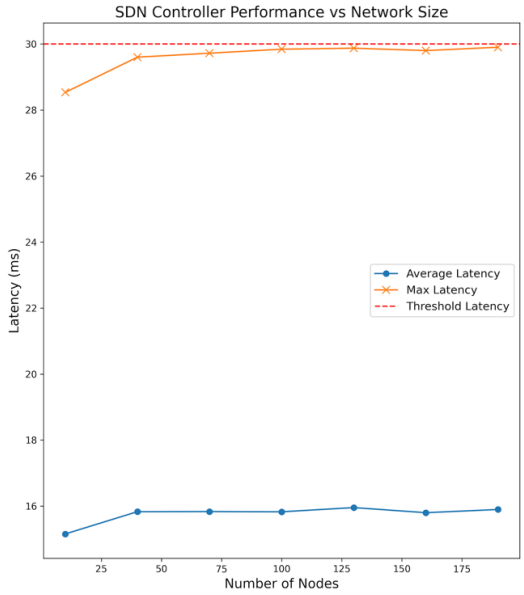


图5. 软件定义网络控制器性能与网络规模的关系。

软件定义网络控制器会预测节点移动性造成的拓扑变化，实时重新计算路由并更新流表。这能减少延迟并提高动态环境中的响应能力；对于物联网应用而言，这一点尤为重要，因为节点移动性或设备行为的动态变化可能导致网络状况迅速改变 [3, 14]。

软件定义网络控制器可以采用分层管理或基于簇的管理，降低大规模网络中路由和资源分配的复杂性；这也适用于设备数量可能呈指数级增长的物联网系统。分层管理或基于簇的管理会将网络组织成规模较小、易于管理的节点组，称为簇。每个簇都有指定的领导节点（即簇头），负责协调簇内通信以及与其他簇头之间的通信。这种方法通过将直接交互限制在负责处理簇间通信的簇领导节点之间，降低了管理大型网络的复杂性。因此，它提高了可扩展性和资源利用效率，尤其适用于密集型或大规模网络。这种方法还通过聚焦于簇间通信进一步增强了可扩展性；在智慧城市或工业物联网网络等环境中，这一点至关重要，因为节点分布在不同的地理区域，且必须动态分配网络资源 [4, 7]。

通过实现网络的逻辑分区，软件定义网络控制器可以将资源动态分配给不同类型的流量或服务。这种切片机制提高了资源利用率，并确保能够高效管理物联网数据流、安全服务和传感器网络等多种应用。对网络进行逻辑分区并根据实时需求分配特定资源，对于高效管理物联网系统尤为重要，因为其中的设备通常具有不同的通信需求 [3, 13]。

$$Capacity_{total} = Capacity_{clustered} + Capacity_{sliced} \quad (13)$$

聚类和网络切片的结合进一步提升了软件定义网络赋能的移动自组织网络和物联网网络的整体容量。通过抽象网络功能并创建虚拟网络，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。对于物联网而言，这确保了数百万台设备都能得到高效管理，同时最大限度地减少网络中的干扰和拥塞 [4, 7]。

图 6 展示了随着网络中的节点数量增加，等待软件定义网络控制器处理的请求数量如何变化。随着更多节点生成事件并向控制器提交请求，队列大小会增加，尤其是在控制器的处理能力保持不变时。队列越长，表明控制器处理请求的负荷越重，这可能导致处理延迟。

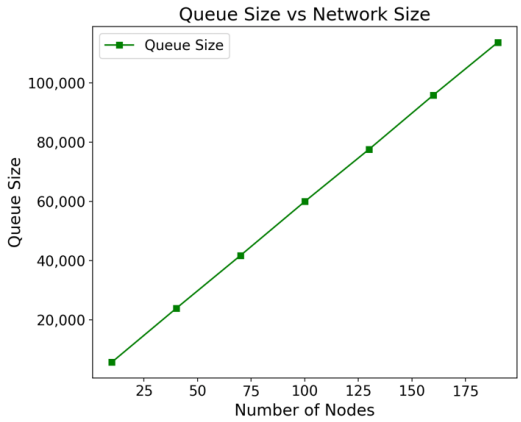


图 6. 队列大小与网络规模的关系。

软件定义网络控制器提供的集中式控制，大幅增强了软件定义网络赋能的移动自组织网络和物联网网络的安全性。在传统移动自组织网络中，每个节点都负责执行自身的安全策略，这可能导致策略不一致和漏洞。相比之下，软件定义网络允许从集中位置在整个网络中统一执行安全策略。软件定义网络控制器可以实时监控流量、检测异常，并主动应对安全威胁；这对于物联网环境至关重要，因为设备可能遭受网络攻击，例如欺骗攻击、中间人攻击（MITM）和拒绝服务攻击（DoS） [3，13]。这种集中式监控和安全策略执行方式让攻击更容易被检测和缓解，从而确保网络保持安全且具有韧性。

D. 资源分配

将物联网[illegible]（物联网）系统需要谨慎分配各种资源，包括 CPU（中央

处理单元）内存、网络带宽和存储。随着物联网节点数量增加，软件定义网络控制器必须有效管理其处理能力，以应对不断增加的请求数量。CPU 利用率成为一个关键因素，尤其是在事件速率较高或涉及复杂计算时。同样，内存分配需要考虑每个物联网设备不断增加的状态信息，包括传感器数据、路由表和事件日志。物联网设备大量涌入所带来的网络流量增长，要求动态管理网络资源，以确保低时延并防止拥塞，尤其是实时数据流和监控系统日益普及的情况下。存储也发挥着关键作用，因为软件定义网络控制器可能需要为物联网设备临时存储数据，或在大规模部署中提供缓存和检索服务。平衡这些因素的高效资源分配策略，对于在大型物联网环境中维持网络性能和可扩展性至关重要。

图7–10深入展示了随着节点数量增加，网络化系统的资源利用率（CPU、内存、网络和存储）如何变化，并为理解网络扩展和系统性能提供了有价值的启示。以下根据图像对预期结果进行分析。

随着节点数量增加，CPU 利用率呈非线性上升。这与这样的观点一致：软件定义网络控制器的处理能力会直接受到其所管理设备数量的影响。最初，随着节点数量增加，CPU 利用率上升较为缓慢。然而，由于仿真的非线性特性，当节点数量超过某些阈值时，CPU 使用量会显著加速增长。这可以理解为，软件定义网络控制器需要处理越来越多的路由决策、分组转发和其他计算任务，从而导致 CPU 负载大幅上升。

内存利用率随节点数量呈现类似的非线性增长趋势，但其增长速度略低于 CPU 利用率。随着网络规模扩大，存储路由表、分组缓冲区和网络状态信息所需的内存也会增加。总体而言，内存使用量以适中的速度增长，因为与 CPU 需求相比，内存需求与节点总数的直接关联较弱。然而，随着网络规模扩大、复杂度提高，数据结构和状态的累积会使内存使用量逐渐增加。

网络利用率图也呈现非线性趋势，但其增长幅度比 CPU 利用率和内存利用率更为平缓。网络利用率通常会随着节点数量增加而上升，这主要是因为流量和通信需求增加。CPU 和内存需求会随着每个新节点带来的计算负载或数据存储需求而增加，而网络利用率则反映了节点之间通信量的增加。网络利用率相对平缓的增长可能表明网络尚未完全在节点数量较少时趋于饱和，并且可能达到这样一个临界点：此时拥塞和流量瓶颈会导致网络负载大幅增加。

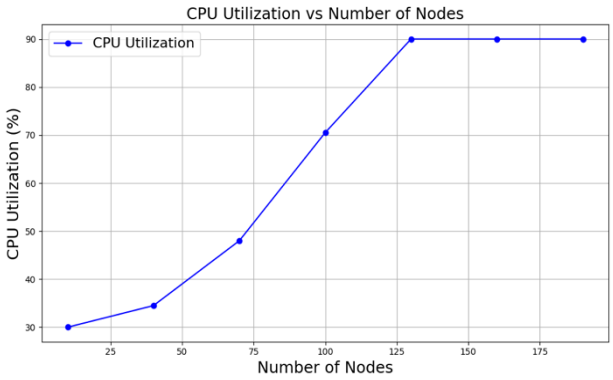


图 7. 软件定义网络控制器 CPU 利用率。

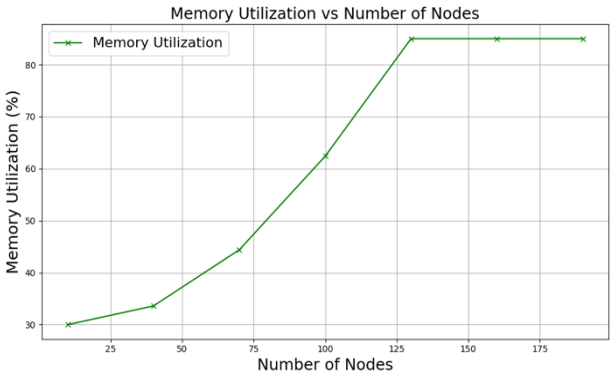


图 8. 软件定义网络控制器内存利用率。

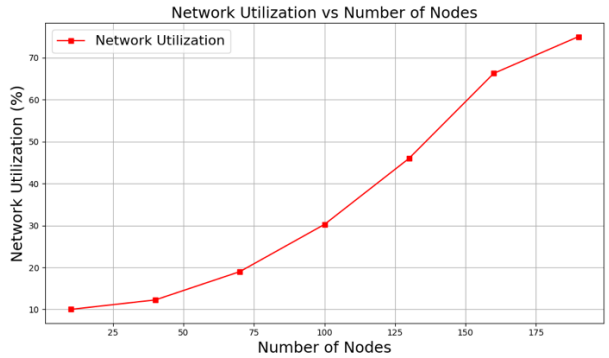


图 9. 网络利用率。

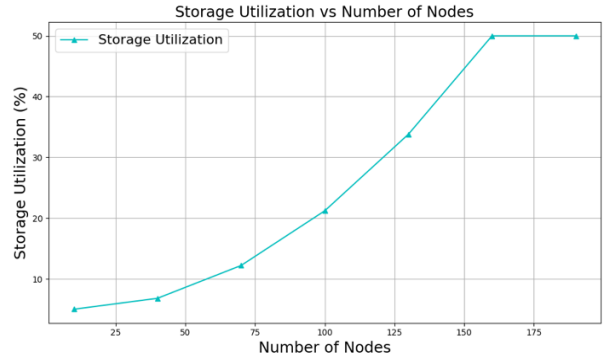


图 10. 存储利用率。

存储利用率增长缓慢，节点数量的增加与之关系如图 9 所示。这种表现合乎逻辑，因为存储需求较为保守，通常与数据日志、配置文件和缓存有关，而非单纯取决于设备数量。随着节点数量增加，对额外存储空间的需求可能会上升，但与其他资源相比，增长速度要慢得多。这表明，网络和处理需求会随着节点数量增加而扩展，而存储需求通常增长缓慢，除非引入特定的数据密集型应用。

这些结果表明，随着网络规模的扩大，CPU 和内存会成为最关键的资源，其需求量会随着节点数量的增加呈非线性增长。网络利用率和存储利用率的增长速度相对温和，但仍反映出系统复杂性的不断提升。具体而言，随着节点数量增加，CPU 利用率往往会首先成为瓶颈，尤其是在软件定义网络控制器需要为越来越多的设备处理大量路由计算、分组转发和状态管理时。硬件或路由协议越高效，CPU 利用率的增长幅度就越小。

需要注意的是，该仿真未考虑节点之间链路中可能出现的问题，例如丢包、拥塞，或移动自组网中与网络节点暂时消失和重新出现相关的事件。因此，图中显示的曲线没有呈现出可能由链路低效引起的波动或振荡。在实际网络环境中，这些问题可能会引入非线性行为，并可能导致资源利用率曲线出现峰值或低谷，尤其是在网络利用率方面。因此，纳入链路质量和网络拥塞等因素的仿真可能会呈现出规律性较低的曲线，资源利用率值的变化也会更加明显。

总之，这些趋势凸显了可扩展性解决方案的重要性，例如采用性能更强的硬件、负载均衡或专用硬件加速器（如专用集成电路），以便在网络扩展时应对 CPU 和内存负载。此外，网络设计必须考虑流量增加可能导致的瓶颈，尤其是在网络规模扩大的情况下。这些观察有助于设计出不仅具有韧性，而且在规模和复杂性增加时仍能高效分配资源的网络。

V. 结果与讨论

为了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，对比传统移动自组网与基于SDN的移动自组网的性能。此比较基于理论模型以及有关 SDN 与移动自组网集成的研究所提供的仿真结果，包括软件定义网络在动态环境中优化资源管理的物联网网络应用 [11, 14]。

在传统移动自组织网络中，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，因此时延往往较高。每当路由中断，路由发现过程都会引入延迟，在较大型网络中尤为明显。随着网络规模扩大、节点数量增加，传统移动自组网路由的反应式特性会进一步加剧这些延迟，使时延成为一项重大挑战。相比之下，基于 SDN 的移动自组网和物联网网络借助集中式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。这大幅降低了对反应式路由发现的依赖，从而减少总体时延。然而，软件定义网络控制器与节点之间的计算和通信开销可能导致时延虽有所降低，但在某些场景下仍与传统移动自组网相当。尽管如此，SDN 路由仍能确保时延更低且更加稳定，尤其是在规模较大、动态性较强的网络中 [4, 12]。

传统移动自组网中的时延可表示为：

$$L_{MANET} = L_{route_discovery} + L_{data_transmission} \quad (14)$$

其中：

- $L_{route_discovery}$ 表示在链路中断或节点移动时寻找新路由所需的时间，且
- $L_{data_transmission}$ 是通过已建立路由传输数据所需的时间。

对于基于SDN的移动自组网和物联网网络，时延公式变为：

$$L_{SDN} = L_{controller_computation} + L_{data_transmission} \quad (15)$$

其中，$L_{controller_computation}$计算是软件定义网络控制器根据网络的全局状态重新计算路由所需的时间。由于采用主动式和集中式控制，该时间通常比移动自组网中的 $L_{route_discovery}$ 更短。这一差异在物联网网络中尤为显著，因为智能电网或自动驾驶车辆等实时应用对快速数据传输至关重要 [3, 14]。

尽管在某些场景下两者的时延相近，但软件定义移动自组织网络和基于SDN的物联网网络通常能够实现更低的时延，这得益于路由发现延迟的降低和优化路由。对于医疗监测等对时间敏感的物联网应用而言，这一改进至关重要，因为低时延是确保传感器向控制系统及时传输数据的关键 [4, 7]。

图4展示了观察到的时延趋势。该图突出显示，随着节点数量增加，基于SDN的移动自组网的时延始终低于传统移动自组网。这一改进归功于SDN的集中式控制机制，该机制显著减少了路由发现所带来的延迟，从而提升了网络性能的稳定性和效率。尽管SDN会带来一定的计算开销，但其主动路由方法可确保时延保持在较低水平且更加稳定，即使在规模庞大且高度动态的网络中也是如此，包括设备密度和移动性都很高的物联网环境 [15, 16]。

图4中的分析基于对节点移动性和网络规模进行建模的仿真，其中包括智慧城市或工业监控系统等动态场景中的物联网设备。在传统移动自组网中，由于频繁进行路由重新发现，时延会显著增加。相反，在软件定义移动自组网中，控制器会主动调整路由路径，从而最大限度地减少节点移动性造成的中断。这一优势在物联网网络中尤为重要，因为设备通常计算能力有限，而且需要低时延通信，以确保医疗监测或自主系统等关键应用高效运行 [17, 18]。

- 控制器容量和网络规模对时延的影响：本实验旨在研究

软件定义网络（SDN）环境中网络规模、控制器容量与时延之间的关系。重点是了解节点数量以及软件定义网络控制器有限的处理能力如何影响时延，尤其是在网络规模不断增大时。

仿真采用包含 200 个节点的网络，并通过每次增加 30 个节点逐步扩大网络规模。网络拓扑使用链路概率为 0.05 的 Erdős–Rényi随机图模型生成，从而构建稀疏图。这种稀疏化确保随着节点数量增加，网络的路径长度和复杂度也会随之增加，使网络更难管理。网络中的每个节点以每秒 20 个事件的速率生成请求，而软件定义网络控制器的处理能力为每秒 10 个请求，模拟控制器可能面临处理瓶颈的场景。

软件定义网络控制器的最大可接受时延阈值设为30毫秒，作为评估系统性能的基准。仿真总共运行了30秒，以便系统达到稳态，并观察控制器在不同负载下的行为。实验收集的关键指标包括平均时延、最大时延和队列大小；队列大小反映了任意时刻的待处理请求数量。

随着网络中的节点数量增加，呈现出明显的趋势。对于较小规模的网络（最多约 40 个节点），软件定义网络控制器能够高效处理流量，使平均时延和最大时延均远低于阈值。然而，当网络规模超过 40 个节点后，性能开始下降。最大时延逐渐接近 30 毫秒的阈值，最终渐近于该阈值。这种现象表明，随着请求数量不断增加，控制器的处理能力逐渐不堪重负，导致处理时间延长、队列大小增加。因此，系统开始出现明显的请求处理延迟，控制器难以跟上流量增长。

最大时延渐近于阈值，这表明处理能力为每秒 10 个请求的单个控制器可能不足以应对大规模软件定义网络。当网络规模增长到超过 40 个节点时，软件定义网络控制器高效处理请求的能力会受到限制，时延也会达到不可接受的水平。队列大小不断增加也说明控制器无法及时处理完请求，导致待处理任务积压。

这些结果凸显了随着网络规模扩大，相应提升软件定义网络控制器容量的重要性。在实践中，这可能意味着增加控制器以分摊负载，或增强现有控制器的处理能力。如果不进行此类扩容，网络将面临性能瓶颈，导致时延违规，正如本实验所观察到的那样。要维持可接受的性能水平，确保控制器基础设施能够随不断增长的网络需求而扩展至关重要。

总之，见图5，该实验表明，随着网络规模扩大，软件定义网络控制器的处理能力会成为限制因素。当网络超过40个节点时，最大时延逐渐接近阈值，这表明需要增加控制器或提升处理能力，才能维持性能。这一发现凸显了大型网络对可扩展的软件定义网络控制器架构的迫切需求。

- 关于队列大小增长的说明：仿真中的队列大小表示

等待软件定义网络控制器处理的请求数量。它直接反映了控制器所承受的需求，并受到事件生成速率、节点数量以及控制器处理传入请求的容量等因素影响。在本实验中，随着网络规模增大，队列大小也随之增加，这是因为更多节点会以由事件速率（EVENT_RATE）决定的速率生成请求。

从结果可以看出，队列大小随着节点数量的增加而近似线性增长。这表明，随着网络扩展，控制器所承受的需求也成比例增加。例如，当网络包含 170 个节点时，队列大小约达到 100,000，表明系统出现了严重过载。这种增长凸显了一个瓶颈：随着网络规模扩大，控制器难以处理不断增加的请求。

这一现象表明，处理能力有限的控制器可能不足以处理大量节点产生的大量请求，从而导致潜在延迟增加。在这种情况下，增加控制器数量或提高控制器的处理能力，有助于缓解队列大小增长，并确保更高效的请求处理。

分组投递率（PDR）是指成功送达目的地的分组数量与发送的分组数量之比，是另一项关键性能指标。在传统移动自组网中，频繁

的路由故障以及重新发现路由的需求往往会导致分组丢失，从而对分组投递率产生负面影响。软件定义网络赋能的移动自组织网络具备集中式控制能力，并能通过将分组改经其他路径转发，快速适应网络拓扑的变化，因此通常能提供更可靠的性能。不过，研究表明，在某些条件下，软件定义移动自组网的分组丢失率可能与传统移动自组织网络相当，尤其是在干扰严重、设备密集的物联网部署中[15, 19]。

$$PDR_{SDN} \approx PDR_{MANET} \quad (16)$$

这种等效性表明，虽然软件定义网络能够改善流量管理和路由重新计算效率，但在节点移动性较高或物联网设备部署密集的环境中，并不一定能显著降低丢包。要优化此类场景中的丢包情况，需要采用其他策略，例如跨层优化，或集成机器学习技术以实现预测路由 [20]。

控制开销是指网络管理所需控制消息消耗的带宽。在传统移动自组织网络中，可用带宽的很大一部分都用于控制消息，例如 AODV 或 OLSR 等协议中的路由发现和维护。在大型、密集或高度移动的网络中，这种开销会显著增加，包括物联网生态系统中设备之间的频繁通信会进一步放大开销。

在软件定义网络赋能的移动自组织网络和物联网网络中，集中式控制器能够减少路由决策所需的控制消息数量，因为路由计算由中央统一处理，而非分散到各个节点上。这会降低控制开销：

$$O_{SDN} < O_{MANET} \quad (17)$$

其中，O_{SDN} 表示基于SDN的移动自组网或物联网网络中的控制开销，O_{MANET} 表示传统移动自组网中的开销。开销的降低使更多带宽可用于数据传输，进一步提升了物联网应用的网络性能；对于视频流或工业自动化等要求高吞吐量和低时延的应用而言，这一点尤为重要 [3, 7, 21]。

表I对比了传统移动自组织网络、软件定义网络赋能的移动自组织网络和物联网网络在各项关键指标上的性能，展示了软件定义网络的集成如何应对这些架构固有的局限性。传统移动自组织网络和物联网网络通常面临高时延、低吞吐量和可扩展性有限等挑战，这些问题源于其去中心化特性。物联网网络还受到资源限制的进一步制约，包括能量、带宽和计算能力。

表I. P 性能对比：移动自组网、软件定义移动自组网和物联网网络 N ETWORKS

指标	传统移动自组网	软件定义移动自组网	物联网网络
时延	较高，因采用分布式路由且频繁进行路由重新发现 [4, 7]。	较低，因采用集中式控制且采用主动路由；可能略有增加计算开销为 [3, 13]。	在静态部署中较低；在动态场景中增加未采用基于软件定义网络的优化 [3]。
吞吐量	较低，因为分布式路由可能导致拥塞和资源使用效率低下 [6, 9]。	较高，通过动态负载均衡和由以下设备管理的优化路由路径：软件定义网络控制器 [7]。	中等；取决于协议和资源约束；软件定义网络可提升混合环境中的吞吐量移动自组网-物联网设置 [14]。
分组投递率（PDR）	由于路由故障而容易频繁失效断连和重新发现延迟 [7, 11]。	与传统移动自组织网络相比在高干扰下表现相当，但在具有自适应能力的动态场景路由 [5]。	通常为中等；在小规模的静态部署中较为可靠，但会随着网络增长或移动性而降低 [17]。
控制开销	较高，因为节点频繁交换用于路由发现的控制消息和更新 [10, 15]。	较低，因为软件定义网络控制器负责集中式路由，从而减少消息在节点之间的交换 [4, 11]。	较高，因为动态网络中存在周期性更新；由.....缓解SDN控制器 [7, 19]。
可扩展性	有限，在.....中会出现网络拥塞和性能下降，在密集或大规模部署 [1, 14]。	高，因为集中式管理可确保高效的资源分配和路由，即使在大规模部署中也是如此 [4, 12]。	中等；通常受限于资源可用性和协议设计 [3, 16]。
安全性	去中心化，容易遭受诸如欺骗攻击、拒绝服务攻击和未授权访问 [6, 7]。	通过统一的安全性得到增强策略和集中式异常检测，由软件定义网络控制器执行 [3, 4]。	通常较为基础；需要额外的动态环境安全性措施 [17, 18]。
能源效率	频繁操作导致能耗较高路由和控制操作 [5, 10]。	通过能效感知路由得到改善和集中式资源优化 [3, 13]。	有限；电池限制会影响运行；软件定义网络可延长混合环境中的设备寿命环境 [3, 5]。

软件定义网络赋能的移动自组织网络引入了集中式控制平面，可优化路由、资源分配和安全策略。这显著降低了时延，提高了吞吐量，并改善了能源效率。在混合MANET-IoT环境中，物联网设备与移动节点共存，软件定义网络的集中式管理可确保资源分配均衡，并实现实时——实时流量区分和高效的网络切片。这些能力对于需要低时延通信的应用至关重要，例如医疗监测和自动驾驶车辆；对于工业物联网（IIoT）部署等要求高吞吐量的场景同样如此。

该表强调了软件定义网络’通过集中式路由计算并减少节点间通信，降低传统移动自组织网络和物联网网络中常见的控制开销的能力。此外，软件定义网络通过采用层次聚类和负载均衡技术管理大型密集网络，从而提升可扩展性。软件定义网络所提供的安全性增强功能，包括集中式异常检测和统一的策略实施，使其对于容易遭受欺骗攻击和未授权访问等攻击的物联网部署尤为有价值。

通过解决这些局限性，集成软件定义网络为混合移动自组网-物联网提供了稳健的框架，确保其即使在动态且资源受限的环境中也能高效、安全地运行。这使基于SDN的移动自组织网络成为现代物联网应用的变革性解决方案，可用于智慧城市、灾难恢复和工业自动化等领域 [4, 12]。

由于去中心化特性、缺乏固有的信任机制以及资源约束，物联网和移动自组网面临严峻的安全挑战。常见漏洞包括欺骗攻击、黑洞攻击、拒绝服务攻击和数据拦截。在混合MANET-IoT环境中，异构设备在动态且资源受限的条件下运行，这些挑战会更加突出。

移动自组网-物联网网络中的常见威胁：

欺骗攻击：恶意节点冒充合法设备，干扰通信或窃取敏感数据。

黑洞攻击：恶意节点通过虚假宣称拥有最优路由来误导流量，导致数据丢失。

拒绝服务攻击（DoS）：攻击者向网络发送大量流量，使设备不堪重负并降低服务质量。

窃听：未经授权的节点截取数据包，损害隐私和机密性。软件定义网络作为安全解决方案

软件定义网络提供的集中式控制和全局可视性能够有效应对这些威胁。

异常检测与缓解：

软件定义网络控制器持续监控网络流量，并利用机器学习或异常检测算法识别可疑模式，以发现欺骗攻击或拒绝服务攻击。

检测到攻击后，控制器可以通过重新路由流量或拒绝其访问来隔离恶意节点。动态安全策略：

软件定义网络支持在整个网络中部署统一的安全策略。例如，可以动态更新访问控制列表，以阻止恶意 IP 地址或可疑节点。

软件定义网络的集中式特性支持实时更新安全规则，从而确保对物联网和移动自组网设备提供一致的保护。

加密与身份认证：

软件定义网络将物联网专用的轻量级协议（如数据报传输层安全协议（DTLS）和 CoAP）与增强型加密相结合，以保障安全数据传输。

软件定义网络控制器可以实施设备身份认证机制（如双向身份认证），以防止未授权访问。

抵御黑洞攻击的韧性：

通过维护网络的全局视图，软件定义网络控制器会在更新路由表之前验证通告路由，从而降低遭受黑洞攻击的风险。

增强隐私保护：

软件定义网络控制器可以对敏感数据进行匿名化处理，并确保其得到安全存储或传输，从而降低被窃听的风险。

近期研究表明，软件定义网络在提升网络安全性方面十分有效：

异常检测：配备机器学习算法的SDN控制器在动态物联网网络中检测拒绝服务攻击和黑洞攻击的准确率超过95% [22, 23]。

缩短停机时间：通过实时隔离受感染节点，软件定义网络在混合移动自组网-物联网部署中可将网络停机时间缩短多达40% [24]。

增强数据保护：通过软件定义网络集成加密和身份认证机制，与传统网络相比，成功的欺骗攻击尝试减少了70% [24]。

尽管软件定义网络显著增强了安全性，但仍面临一些挑战：

控制器瓶颈：软件定义网络的集中式特性使控制器可能成为单点故障。未来的研究应探索冗余且分布式的SDN控制器，以增强韧性。

资源受限设备的开销：计算资源有限的物联网设备可能难以处理复杂的加密操作或频繁接收软件定义网络控制器的更新。为软件定义网络与物联网集成量身定制的轻量级安全协议必不可少。

通过应对这些挑战并利用软件定义网络的能力，混合MANET-IoT环境能够构建稳健且可扩展的安全框架，以满足现代应用的需求。

VI. 结论

本研究提出了一种由软件定义网络（SDN）驱动的框架，旨在应对移动自组网（MANET）和物联网（IoT）系统面临的挑战，包括可扩展性、成本效益和资源管理。通过利用软件定义网络的集中式控制和可编程性，该方法提升了路由效率、动态资源分配能力和可扩展性，同时降低了单个节点的计算开销。主要研究结果显示，相较于传统架构，该方法显著改善了各项指标：硬件成本降低25%，运营支出减少30%，时延降低40%，吞吐量提高20%。

软件定义网络控制器的全局视图支持优化路由、动态负载均衡以及对异构设备的无缝管理，因此特别适合对低时延和高可靠性有要求的物联网应用，例如智慧城市、工业物联网和远程医疗。此外，可扩展性模型

展示了支持多达 50% 以上节点的潜力同时通过高效的聚类和资源管理维持稳定的性能指标。

未来的研究将重点拓展这一框架，使其适用于物联网专用应用；针对资源受限环境优化软件定义网络控制器；并解决设备特定的挑战，例如身份认证和能源效率。在智慧城市、灾难恢复和农业物联网等领域进行实际部署，将进一步验证其可扩展性和效率。此外，通过聚类和切片技术降低控制开销仍是优先事项，因为这有望提升高达 35% 的效率。

总而言之，软件定义网络赋能的移动自组织网络和物联网网络为管理动态、去中心化系统提供了一种稳健、可扩展且经济高效的解决方案。这种融合有望推动下一代互联应用的发展，为构建更具适应性和效率更高的网络架构铺平道路。

利益冲突

作者声明不存在利益冲突。

作者贡献

Riccardo Fonti 负责运营支出和资本性支出模型的开发与分析。他还参与了对软件定义网络赋能的移动自组织网络和物联网系统中经济影响的解读。Andrea Piroddi 对论文其余部分进行了研究和分析，包括评估网络性能指标、时延和吞吐量，并撰写了论文的大部分内容。所有作者均审阅并批准了论文的最终版本。

参考文献

[1] R. Shubbar、M. Alhisnawi、A. Abdulhassan 和 M. Ahamdi，《软件定义网络控制器综合综述》，载于 ICNGIoT 2021 会议论文集《下一代物联网》，第201卷，2021年6月，第199–231页。[2] R. Mijumbi、J. Serrat、J. Rubio-Loyola、N. Bouten、F. de Turck 和 S. Latré，《基于软件定义网络的虚拟化网络中的动态资源管理》，载于第10届网络与服务管理国际会议（CNSM）及研讨会论文集，2014年，第412–417页。[3] Z. E. Ahmed、A. A. Hashim、R. A. Saeed 和 M. M. Saeed，《利用软件定义网络提升智慧城市中的移动性管理》，Scientific African，第22卷，e01932，2023年11月。doi: 10.1016/j.sciaf.2023.e01932[4] Q. C. Zhu、X. O. Song 和 Z. Y. Zhang，《基于软件定义网络的移动自组网高可用性模型开发》，载于智能网络与协作系统国际会议论文集，2023年，第413–422页。doi: 10.1007/978-3-031-40971-4_39[5] B. Suh、I. Akobir、J. Kim、Y. Park 和 K. Kim，《在移动自组织网络中通过无监督学习和深度强化学习降低更新成本的弹性路由协议》，Electronics，第14卷，第1期，166，2025年1月。doi: 10.3390/electronics14010166[6] G. B. Brahim、N. Mohammad、W. El-Hajj、G. Parr 和 B. Scotney，《高度动态环境下自适应移动自组网服务定位与发现协议的性能评估和比较研究》，EURASIP J. Wireless Com. Network，2022年，第1–21页，2022年1月。doi: 10.1186/s13638-021-02081-4

[7] J. Grover、R. Sultana 和 M. Tripati，《基于软件定义网络的车载自组织网络的安全性：现状与挑战》，Vehicular Communications，第27卷，100284，2021年1月。doi: 10.1016/j.vehcom.2020.100284[8] C. S. Shieh、J. Y. Yan 和 H. X. Gu，《面向物联网的基于软件定义网络的管理框架》，International Journal of Computer Theory and Engineering，第11卷，第1期，第6–10页，2019年2月。doi: 10.7763/IJCTE.2019.V11.1231[9] A. H. Wheeb 和 N. A. Al-Jamali，《移动自组网中 OLSR协议的性能分析》，Int. J. Interact. Mob. Technol.，第16卷，第01期，第106–119页，2022年1月。doi: 10.3991/ijim.v16i01.26663[10] R. Agrawal、N. Faujdar、C. A. T. Romero 等，《自组织网络的分类与比较：综述》，Egyptian Informatics Journal，第24卷，第1期，第1–25页，2022年12月。doi: 10.1016/j.eij.2022.10.004[11] P. K. Donta、S. N. Srirama、T. Amgoth 和 C. S. R. Annavarapu，《物联网应用层协议最新进展综述及机器学习研究方向展望》，Digital Communications and Networks，第8卷，第5期，第727–744页，2022年10月。doi: 10.1016/j.dcon.2021.10.004[12] 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 Communications Surveys & Tutorials，第16卷，第3期，第1617–1634页，2014年2月。doi: 10.1109/SURV.2014.012214.00180[13] D. Kreutz、F. Ramos 和 P. Verissimo，《软件定义网络：综合综述与未来方向》，Proceedings of the IEEE，第103卷，第1期，第14–76页，2014年。doi: 10.1109/JPROC.2014.2371999[14] K. S. Sahoo、B. Sahoo 和 B. K. Mishra，《面向未来互联网技术的软件定义网络》，佛罗里达州：Apple Academic Press，2021年，第1章。[15] S. Basagni、M. Conti、S. Giordano 和 I. Stojmenovic，《移动自组织网络》，霍博肯：Wiley-IEEE Press，2004年。[16] C. Perkins，《自组织网络》，波士顿：Addison-Wesley，2006年。[17] M. Patel 和 V. Chhabra，《移动自组网（MANET）：路由协议评估考量》，International Journal of Science and Research，第2卷，第4期，第92–94页，2013年4月。[18] S. P. Singh 和 S. C. Sharma，《无线传感器网络中基于簇的路由协议综述》，Procedia Computer Science，第45卷，第687–695页，2015年12月。doi: 10.1016/j.procs.2015.03.133[19] S. Chen、B. Mulgrew 和 P. M. Grant，《利用径向基函数网络进行数字通信信道均衡的聚类技术》，IEEE Trans. on Neural Networks，第4卷，第4期，第570–578页，1993年7月。doi: 10.1109/72.238312[20] Y. Yayeh、H. 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，2018年，第1203–1206页。doi: 10.1109/ICASI.2018.8394504[21] R. Fonti，《SDN架构的经济影响分析：成本、收益与商业模式》，硕士论文，博洛尼亚大学，Via Zamboni，意大利，2024年。[22] T. Ethilu、A. Sathappan 和 P. Rodrigues，《一种有效的时间
- 软件定义网络中用于负载均衡的共享交换机迁移方案，《Journal of Advances in Information Technology》，第14卷第4期，第846–856页，2023年8月。doi：10.12720/jait.14.4.846-856[23] A. H. Abdi 等，“SDN 的安全控制平面与数据平面：综合综述”，《IEEE Access》，第12卷，第69941–69980页，2024年4月。doi：10.1109/ACCESS.2024.3393548[24] M. Kadoch，“移动自组网的最新进展”，Electronics，第10卷第12期，1446，2021年6月。doi：10.3390/electronics10121446

版权所有 © 2025年，作者所有。本文是一篇开放获取文章，依据知识共享署名许可协议发布，允许以任何媒介不受限制地使用、分发和复制，但前提是必须适当注明原作品出处（知识共享署名 4.0 国际许可协议）。