

8-7-2020

Design and Implementation of Hardware-In-the-Loop Simulation Dynamic Routing Protocol

Xiaohu Liu

Information Engineering University, Zhengzhou 450000, China;

Yuchen Zhang

Information Engineering University, Zhengzhou 450000, China;

Qingpo Fang

Information Engineering University, Zhengzhou 450000, China;

Leigang Sun

Information Engineering University, Zhengzhou 450000, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Design and Implementation of Hardware-In-the-Loop Simulation Dynamic Routing Protocol

Abstract

Abstract: Against the problem of dynamic routing in network Hardware-In-The-Loop Simulation, on the basis of analysis of network HITL simulation principle, *a dynamic routing protocol implementation scheme based on OMNeT++* was proposed. *A method of routing message transform and a algorithm of time slice synchronize* was proposed, overcoming the problem of routing message transform and system real-time. At the same time, using virtual IP technology for multi-host configuration, and OSPF dynamic routing protocol, the *Hardware-In-the-Loop Simulation Dynamic Routing Protocol based on OMNeT++* was realized. The dynamic routing protocol with multi scenarios was verified and tested.

Keywords

HITL, dynamic routing, link-state routing optimization, real-time

Recommended Citation

Liu Xiaohu, Zhang Yuchen, Fang Qingpo, Sun Leigang. Design and Implementation of Hardware-In-the-Loop Simulation Dynamic Routing Protocol[J]. Journal of System Simulation, 2015, 27(9): 2089-2095.

半实物网络仿真动态路由方案设计与实现

刘小虎, 张玉臣, 方青坡, 孙磊刚

(信息工程大学, 郑州 450000)

摘要: 针对半实物网络仿真动态路由难以实现的问题, 在分析半实物网络仿真原理的基础上, 提出了动态路由消息转换方法和基于时间分片的障碍同步算法, 解决了动态路由的消息转换和系统实时性保证两个关键问题, 利用虚拟 IP 技术解决了半实物网络仿真中的多主机配置问题, 采用 OSPF 路由协议, 设计并实现了基于 OMNeT++ 的半实物网络仿真动态路由方案。通过多种虚实结合场景对动态路由协议进行功能验证和可扩展性测试, 验证了方案的正确性。

关键词: 半实物网络仿真; 动态路由; 优化链路状态路由; 实时性

中图分类号: TP391.9

文献标识码: A

文章编号: 1004-731X (2015) 09-2089-07

Design and Implementation of Hardware-In-the-Loop Simulation Dynamic Routing Protocol

Liu Xiaohu, Zhang Yuchen, Fang Qingpo, Sun Leigang

(Information Engineering University, Zhengzhou 450000, China)

Abstract: Against the problem of dynamic routing in network Hardware-In-The-Loop Simulation, on the basis of analysis of network HITL simulation principle, a dynamic routing protocol implementation scheme based on OMNeT++ was proposed. A method of routing message transform and a algorithm of time slice synchronize was proposed, overcoming the problem of routing message transform and system real-time. At the same time, using virtual IP technology for multi-host configuration, and OSPF dynamic routing protocol, the Hardware-In-the-Loop Simulation Dynamic Routing Protocol based on OMNeT++ was realized. The dynamic routing protocol with multi scenarios was verified and tested.

Keywords: HITL; dynamic routing; link-state routing optimization; real-time

引言

半实物网络仿真^[1-2] (Hardware-In-The-Loop: HITL) 具有验证说服力强、能够更直观精确地进行网络软硬件验证与测试等优点, 已成为网络仿真领域的重要研究方向。以往的半实物网络仿真系统, 如 Ersal.T 等人^[3]设计的 ID-HILS 系统、Galli.E 等

人^[4]开发的 HLA-OMNeT++ 平台、唐扬等人^[5]提出的 HITLS 网络安全仿真模型等, 真实节点和虚拟节点均为静态路由, 路由器无法根据实际情况变化对路由表进行动态更新。随着网络仿真规模不断扩大、模块库日趋健全、智能化水平愈来愈高、应用领域越来越广, 网络呈现拓扑动态性、节点不稳定性、无中心性和开放性等特点, 静态路由已不能满足网络仿真的要求。针对该问题, Javaid.N 等人^[6]针对 LEACH, SEP, E-SEP 和 DEEC 等 4 类网络协议, 基于仿真器设计了 ATCEEC 动态路由框架, 但没有扩展到半实物网络仿真领域; Ihbeel.A.S 等人^[7]基于 NS2 对无线传感器网络(WSN)进行了



收稿日期: 2015-05-20 修回日期: 2015-07-06;
作者简介: 刘小虎(1989-), 男, 河南太康, 硕士, 助教, 研究方向为计算机网络仿真; 张玉臣(1977-), 博士, 副教授, 研究方向为信息安全; 方青坡(1979-), 硕士, 讲师, 研究方向为计算机应用。

<http://www.china-simulation.com>

• 2089 •

AODV, DSR 和 DSDV 协议的半实物扩展仿真, 旨在进行 3 种协议数据包传输性能及路由开销的比较, 并未给出路由扩展的具体应用; Nirav.G^[8] 针对移动自组网(MANET)进行基于 OMNeT++ 平台的 OLSR(Optimized Link State Routing)动态路由协议扩展, 该系统仅使用了系统等待函数进行同步, 未采取同步算法, 实时性难以有效保证。

本文基于 OMNeT++ 的 INET^[9] 仿真框架, 研究解决半实物网络仿真中的路由转换和系统实时性保证 2 个关键问题, 利用开放最短路径优先协议 (Open Shortest Path First: OSPF), 使网络内各路由器节点能够动态更新路由表, 实现 OMNeT++ 仿真器对半实物网络仿真的动态路由支持。

1 基于 OMNeT++ 的半实物网络仿真

1.1 基于 OMNeT++ 的半实物数据转换

当前支持 HITL 的仿真器主要有 OPNET、

QualNet, NS2 和 OMNeT++ 等^[10-11], OPNET 和 QualNet 是非开源的商业软件, 内部代码没有公开, 不能进行源代码级开发; NS2 应用于 Linux 系统, 通用性和扩展性相对不强。综合考虑, 本研究选择开源的 OMNeT++ 仿真器作为半实物网络仿真平台。INET 是 OMNeT++ 的扩展模块, 提供了将实物连接至仿真器的 HITL 功能, 其半实物网络仿真架构及数据处理流程如图 1 所示。

INET 提供了连接仿真器与真实网络的网关模型, 可将发送到本地网卡的真实数据包解析为虚拟网络能够识别的类结构, 也可将发送至外部真实节点的虚拟数据包序列化为真实比特流。虚实数据包转换过程对于各节点来说是透明的。

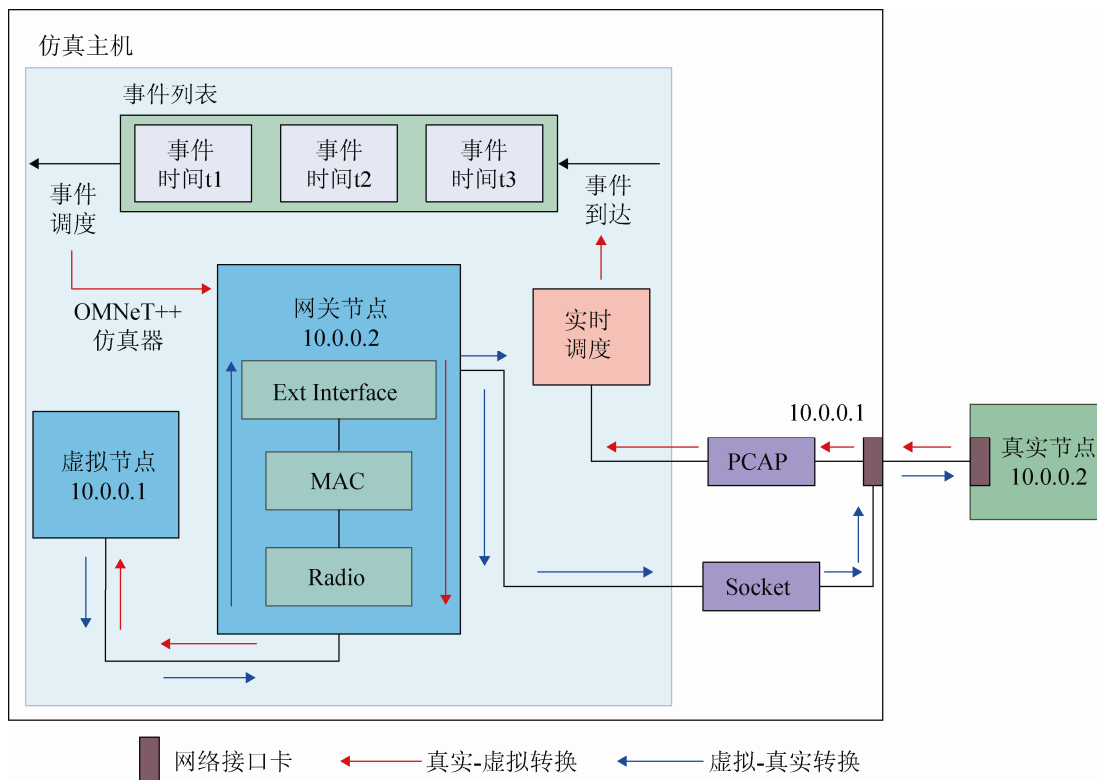


图 1 HITL 虚实数据转换流程

1.2 基于 OMNeT++ 的半实物路由支持

OMNeT++ 仿真器通过配置静态路由实现路由功能。在仿真开始之前, 虚拟节点路由信息可在仿真源文件中进行手工配置, 也可通过写入特定格式的路由文件(mrt 或 irt 文件)来完成; 真实节点通过在操作系统路由表中添加路由。仿真器内部路由协议实现“虚-实”静态路由的协调运行, 但路由信息均为静态, OMNeT++ 不支持对其进行更改, 原理如图 2 所示。

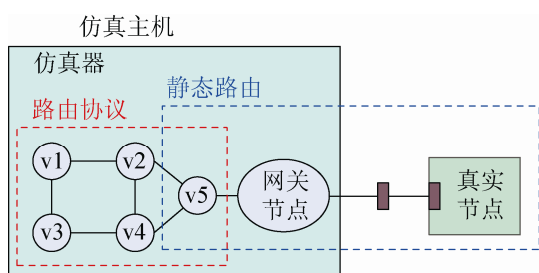


图2 OMNeT++半实物网络仿真路由机制

2 基于 OMNeT++ 的 OSPF 动态路由设计与实现

OSPF 协议的是一种基于链路状态路由算法的动态路由协议, 目前 IPv4 网络中使用的 OSPF 一般都遵循 RFC2328 标准。同一自治系统(AS: Autonomous System)内 OSPF 路由器都维护一个用于跟踪网络状态的链路状态数据库(LSDB: Link State DataBase), 它由承载各种网络状态的链路状态通告(LSA: Link State Advertisement)构成。OSPF 利用 hello 协议、建立近邻关系和“可靠泛洪”机制来实现区域内路由器链路状态数据库的统一, 然后每个路由器利用 Dijkstra 算法, 以自身为树根构建最短路径树, 得出通向每个网络的最佳路由, 完成路由表的构建。路由器根据该路由表进行选路和数据转发。

半实物网络仿真动态路由设计与实现过程中面临 2 个关键问题: 一是路由消息转换; 二是系统实时性保证, 并降低同步开销。

2.1 路由消息转换方法

OSPF 有 5 种类型数据包, 分别为 Hello 包、DD 包(数据库描述包)、LSR 包(连接状态请求包)、LSU 包(连接状态更新包)和 LSAck 包(连接状态确认包)。为了在 OMNeT++ 支持的 HITL 中实现动态路由协议, 首先, 需要在网关模型内实现五类 OSPF 数据包的转换。

半实物网络仿真场景中, 真实节点上运行 OSPF 后台程序, 虚拟节点上运行 OMNeT++ 中虚拟 OSPF 协议, 所有路由节点相互交换路由信息从而动态更新自身路由表。网关模型中 cSocketRTScheduler 是用来进行 HITL 仿真的调度类, 可以实现数据包的转换和系统时钟同步。cSocketRTScheduler 在初始化时进行 PCAP 设备注册, 过滤掉不符合特征的网络数据包。

网关节点中 ExtInterface 用来实现数据包的转换。数据包的每个分组要按字节顺序转换, 真实网络中的节点按照一致协定的顺序进行通信, 称作“网络字节序”(Network Byte Order), 然而, 每个节点中还包含代表多字节数据的预定义字节序, 称作“主机字节序”(Host Byte Order), 当网关节点接收到一个数据包, 都需要将多字节域数据进行 NBO 和 HBO 的转换。

真实节点将数据包发送至仿真机 NIC 卡, PCAP 通过监听 NIC 卡来获得网络数据包, ExtInterface 根据协议定义的数据包类型(如 Hello 包), 将数据包解析为多个大小不同的数据部分, 将多比特域数据转换为 HBO(从 NBO), 而后将这些 HBO 填充到对象类的适当部分, 构成虚拟数据包在虚拟网络中传递, 当发送至虚拟网络的数据到达时, 标记其到达时间戳并存放到优先级事件列表中, 再根据优先级顺序执行各事件。虚拟数据包到真实数据包的转换时, 虚拟数据包的所有类对象多字节域部分将被 ExtInterface 作序列化处理, 单比特流数据不需要处理, 所有 NBO 根据协议定义的数据包类型合并为单比特流发送至真实节点。

OSPF 消息拥有统一的首部格式, 包含数据包长度和分组序号, 一个包可以包含多个相同或不同的消息类型, 每个消息都会根据消息类型独立转换, 为 OSPF 协议预定义的端口是 89, 协议数据包被当作 IP 数据包转换的一部分, 任何通过 89 端口的 IP 数据包都会被认为是 OSPF 协议数据包, 并被解析转换为虚拟数据包。发送至真实节点的虚拟包序列化过程中, 仿真器内部首先进行传输层协议检测, 端口号为 89 的 OSPF 消息将在 ExtInterface 中实现完整的转换并被当作 IP 数据包发送。

2.2 实时性保证机制

实时性是半实物网络仿真的基本条件^[12]。OMNeT++ 是基于消息的离散事件驱动网络模拟器, 仿真时钟由仿真事件推进, 一个仿真事件执行完毕直接跳到下一个仿真事件; 系统内部无仿真事件发生时, 仿真时间停止。因此, 仿真时钟与真实时钟并不一致。故动态路由协议实现过程中需要将仿真器、真实主机时钟进行同步, 解决仿真时钟与真实时钟不一致问题, 如图 3 所示。

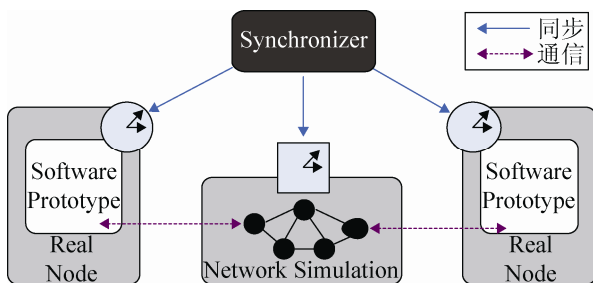


图3 系统时钟同步

本文提出基于时间分片的障碍同步算法。采用类似于 CTW (conservative time windows, 保守时间窗口) 机制来进行系统同步, 流程如图 4 所示。

同步算法中包含仿真器和真实主机(或虚拟机)两大模块, 障碍算法允许每一个同步事件运行在一个确定的时间分片内, 当时间分片内所有仿真事件执行完成, 时间分片到达所设置的障碍时间时, 仿真内核阻断仿真, 采取等待的方式与真实时钟进行

同步。得到此仿真时间点与真实时钟同步后, 障碍被解除, 进入下一个时间分片, 新的障碍被重新设置, 同步事件继续推进。由于仿真时钟和真实时钟都被设置在一个时间分片内, 时间偏差在任何时候都依赖于时间分片的大小, 所以, 同步精度由时间分片的大小直接决定。

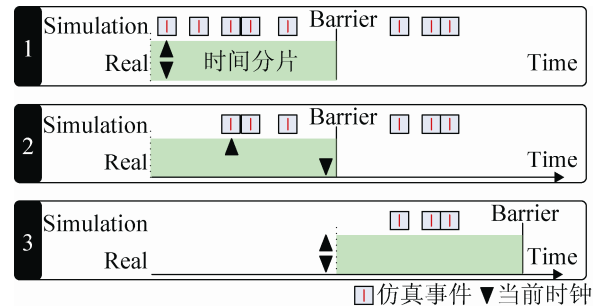


图4 基于时间分片的障碍同步算法

3 验证与测试

本节通过研究多种仿真场景, 确保每个路由节点(虚拟或真实节点)能够成功建立路由。需要说明的是, 仿真主机连接多个真实节点时, 需要配置多个 NIC 卡, 这对于大规模网络来说是不现实的。本文采用基于虚拟 IP 地址的解决方案, 一个 NIC 卡可以拥有多个虚拟 IP 地址分配给多个虚拟机 (VM) 应用程序使用。虚拟机的设置及连接非本文研究重点, 详情可参考文献[8]。

3.1 半实物网络仿真场景设计

3.1.1 单仿真节点连接多接口单真实节点

仿真场景如图 5, 虚拟节点包含两个接口, 10.0.0.1 连接在实物之上, 还有 20.0.0.1 接口, 真实节点有与仿真器相连 10.0.0.2 接口之外的 30.0.0.1 接口。所有节点通过发送 OSPF 数据包建立路由, 图 6 为网关节点动态生成的路由表。

3.1.2 真实-虚拟-真实场景

本场景连接虚拟网络和两个真实网络, 组成真实-虚拟-真实半实物网络进行测试, 连接方式如图

7 所示。通过设置仿真参数并运行仿真, 观察节点路由表信息, 各节点仍可以同其它节点成功建立路由关系。虚拟-真实-虚拟场景同该场景类似, 不再

赘述。与单节点场景相比, 在路由消息发送时间间隔的设置上有所不同, 不正确的设置会导致路由表建立失败。具体原因和解决方法将在下节介绍。

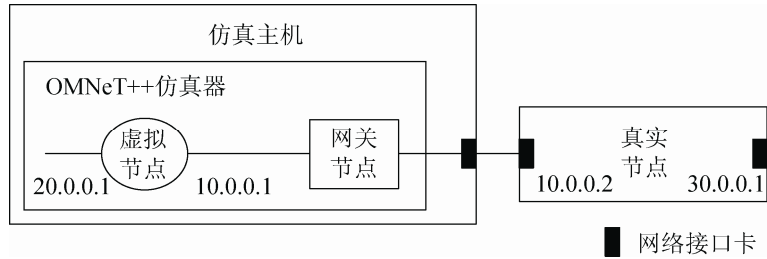


图5 单真实主机连接配置

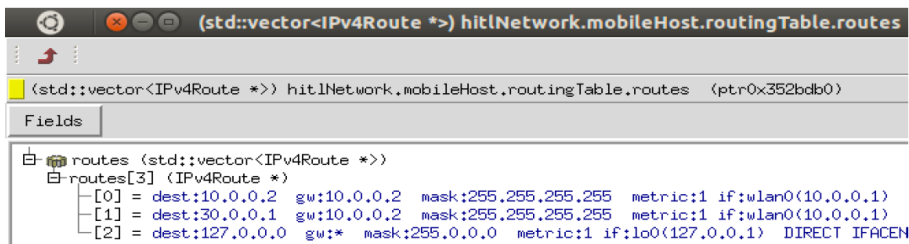


图6 虚拟节点路由表

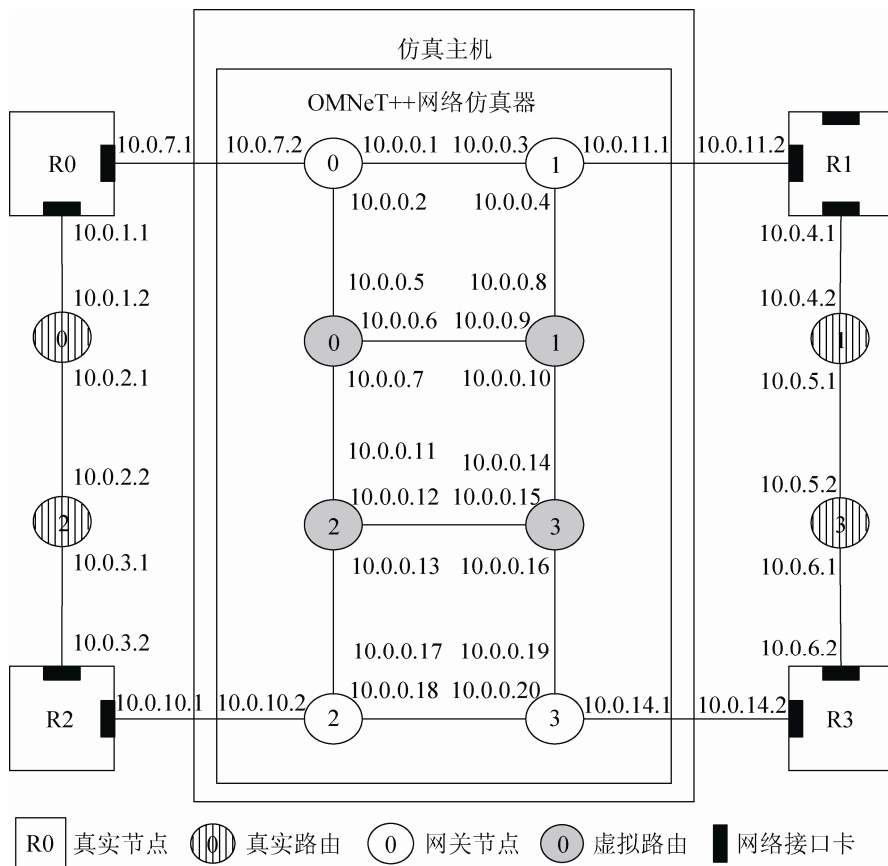


图7 真实-虚拟-真实场景

3.2 可扩展性测试

为了观察在网络规模扩大的过程动态路由协议的有效性, 不断增加仿真节点数目, 节点数扩展至 6×6 网络, 如图 8 所示。扩展过程中 CPU 性能降低, 仿真时钟逐渐变慢, 不同的路由消息产生时间间隔会影响节点路由表的生成, 不适合的参数会导致路由无法生成, 如 4.1 节真实-虚拟-真实场景

中, 在跨越虚拟网络的真实节点之间进行路由消息传递, 一个真实节点接收到来自另一个真实节点的消息, 在这个发送周期内该节点会等待其它来自源节点的消息, 路由消息跨越虚拟网络进行传递会耗费较长时间, 经过多次实验观察, 不同网络拓扑的路由消息发送时间间隔需要进行调整。

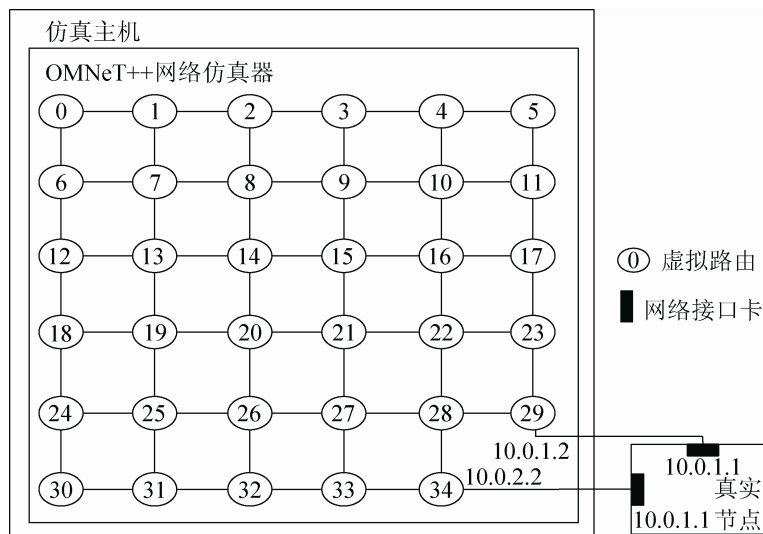


图 8 HITL 可扩展性测试框架

通过设置不同路由消息产生时间间隔, 观察 CPU 的性能指标, 来确定最适合的消息产生时间组合策略, 运用控制变量法观察 HELLO 消息对 CPU 的 coremark 指标影响如图 9 所示。

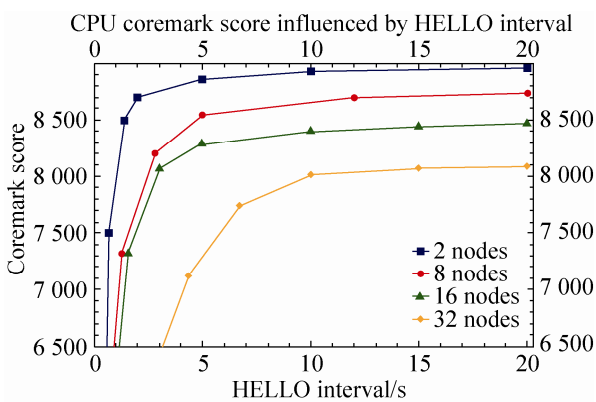


图 9 HELLO 消息发送间隔对 CPU 性能影响

同样对 DD 消息和 LSU 消息产生时间间隔进

行仿真实验, 最终确定各场景能够正确生成动态路由的路由消息时间设置如表 1 所示。

表1 不同仿真场景下路由消息产生速率

仿真场景 V(Virtual), R(Real)	Hello Interval	DD Interval	LSU Interval
1 V&1 R	2s	5s	5s
4 V&4 R	5s	8s	8s
8 V&8 R	5s	8s	8s
31 V&1 R	10s	15s	15s

3.3 同步开销分析

仿真器及真实主机(或 VM)均运行在非实时操作系统(RTOS)下, 引入同步方案, 算法采取不同的时间分片, 系统在实时运行时必产生相应代价。实验采取虚拟机代替真实主机, 测试 120 s 逻辑时间内运行于同一物理机的 VM 同步开销比(逻辑时

间内与所消耗的真实时间比值), 不同个数虚拟机情况下, 测试结果如图 10 所示。

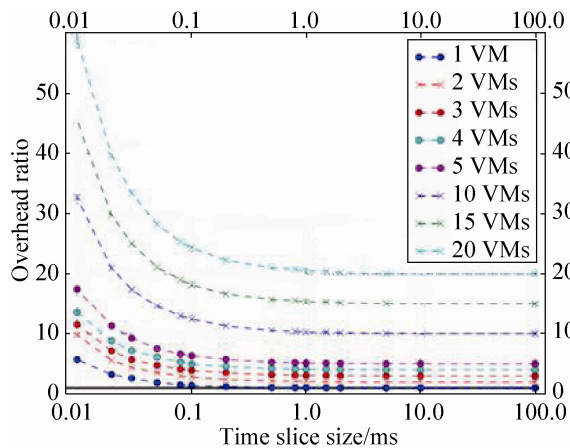


图10 算法时间分片大小对同步开销比的影响

可以看出, 接入少量真实节点, 时间分片较小时, 事件被频繁挂起和暂停, 使得消息开销增大; 时间分片大于 0.5 ms 时, 同步开销保持在 10% 以下, 此时已接近实时行为。所以, 时间分片的大小设置不能低于某一确定值, 这取决于仿真机的时钟分辨率, 只能寻求一个最优设置, 不是越小越好。

4 结论

本文在研究 OMNeT++ 网络仿真软件的 INET 框架所支持的半实物网络仿真环境的基础上, 提出了路由消息转换方法和基于时间分片的障碍同步算法, 解决了半实物网络仿真实现过程中的路由消息转换和系统实时性保证两个关键问题, 利用 OSPF 动态路由协议, 实现了半实物网络仿真过程中动态路由方案。最后, 设计了多种场景对路由算法正确性、系统可扩展性、同步开销进行了验证和测试。实验结果表明, 所设计的方案的正确性。同时, 通过大量实验运行, 得出了三类路由消息产生频率的设置方案。下一步研究方向是大规模网络仿真环境下时间分配策略和事件调度方法深化研究, 和从系统层面综合考虑路由消息配置方案。

参考文献:

[1] 单家元, 孟秀云, 丁艳, 等. 半实物仿真(2版) [M]. 北

京: 国防工业出版社, 2013.

- [2] Yang J, Li Y. Hardware-in-the-loop simulation of communication networks [J]. Journal of Beijing Institute of Technology(S1001-0645), 2012, 21(3): 376-381.
- [3] Ersal T, Brudnak M, Stein J L, *et al.* Statistical Transparency Analysis in Internet-Distributed Hardware-in-the-Loop Simulation [J]. IEEE/ASME Transactions on Mechatronics (S1083-4435), 2012, 17(2): 228-238.
- [4] Galli E, Cavarretta G, Tucci S. HLA-OMNET++: an HLA compliant network simulator [C]// IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications, Vancouver, Canada. USA: IEEE, 2008: 319-321.
- [5] 唐扬. 基于 HILS 的网络安全仿真研究 [D]. 郑州: 信息工程大学, 2009.
- [6] Javaid N, Aslam M, Djouani K, *et al.* ATCEEC: A New Energy Efficient Routing Protocol for Wireless Sensor Networks. [C]// IEEE ICC 2014-Ad-hoc and Sensor Networking Symposium, Sydney, Australia. USA: IEEE, 2014: 263-268.
- [7] Iheeeel A S, Sigiuk H I, Alhsh A A. Simulation Based Evaluation of MANET Routing Protocols for Static WSN [C]// Innovative Computing Technology (INTECH), 2012 Second International Conference on, Casablanca, The Kingdom of Morocco. USA: IEEE, 2012: 63-68.
- [8] Nirav G. Dynamic Routing Framework for OMNET++ Based Hardware-in-the-Loop (HITL) Network Simulation [D]. Tamil Nadu, India: Indian Institute of Technology Madras, 2013.
- [9] 赵永利, 张杰. OMNeT++ 与网络仿真 [M]. 北京: 人民邮电出版社, 2012.
- [10] Mishra V, Jangale S. Analysis and comparison of different network simulators [C]// Special Issue for International Technological Conference-2014. India: Vivekanand Education Society's Institute of Technology, 2014: 5.
- [11] Varga A, Hornig R. An Overview of the OMNeT++ Simulation Environment [C]// Simutools '08 Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops. Marseille, France, Brussels, Belgium: ICST, 2008: 10.
- [12] Matar M, Karimi H, Etemadi A, *et al.* A High Performance Real-Time Simulator for Controllers Hardware-in-the-Loop Testing [J]. Energies (S1996-1073), 2012, 5(6): 1713-1733.