

机会网络模拟器 ONE 及其扩展研究 *

王 朕¹, 王新华^{1,2}, 隋敬麒¹

(1. 山东师范大学 信息科学与工程学院, 济南 250014; 2. 山东省分布式计算机软件新技术重点实验室, 济南 250014)

摘 要: 网络仿真是 DTN 网络研究的重要方法和手段。通过对 ONE 模拟器的开发背景、结构、功能及其应用的介绍, 探讨与分析了基于 Java 环境的 DTN 网络模拟器 ONE 的扩展方法, 进一步对 ONE 的不足之处进行了完善和改进, 并通过仿真实验证实了改进的 ONE 对路由性能评价的影响。

关键词: 网络模拟; 机会网络; ONE

中图分类号: TP393 **文献标志码:** A **文章编号:** 1001-3695(2012)01-0272-06

doi:10.3969/j.issn.1001-3695.2012.01.076

Extending research for ONE simulator of opportunistic network

WANG Zhen¹, WANG Xin-hua^{1,2}, SUI Jing-qi¹

(1. School of Information Science & Engineering, Shandong Normal University, Jinan 250014, China; 2. Shandong Provincial Key Laboratory for Distributed Computer Software Novel Technology, Jinan 250014, China)

Abstract: Network simulation is an important method in the area of delay tolerance network research. This paper introduced the development background, composition, function and application of opportunistic network environment (ONE). It analyzed the extending method of ONE which was Java-based simulator for DTN. Finally, improved some deficiencies of ONE and presented simulation results to prove the improvement of ONE in the area of routing performance evaluation.

Key words: network simulation; opportunistic network; opportunistic network environment(ONE)

0 引言

随着个人手持通信设备技术的发展,特别是具备足够的处理能力、存储空间和电池容量的先进移动设备的出现,使得多个用户能够轻易地组成 Ad hoc 网络进行通信。但是当节点稀疏时,Ad hoc 网络就会出现分割的现象,即节点之间不具备完整的端到端通信的路径。为了实现这些网络之间的互连,近年来研究人员提出了在端到端连接和节点资源都受限时的一种新型网络体系结构和应用接口,称为延迟容忍网络(delay tolerant networks,DTN)^[1]。DTN 网络体系是由多个 DTN 域组成的,其中每个 DTN 域的底层都运行独立的通信协议。域间网关则是利用“存储—转发”的工作模式,即消息只在去往目标 DTN 域的链路存在时转发;否则,将消息存储在本地持久存储器中,直到产生可用链路。

进行网络研究有许多种方法,包括分析建模、实验测试和网络仿真等。网络仿真以周期小、经济有效、准确性高、可操作性强等众多优点被网络研究学术界所广泛使用。在 DTN 网络环境的路由协议和应用的研究分析上,网络模拟仿真具有重要的作用,特别是在节点分布较稀疏时,DTN 仿真能够更好地描述无线链路的细节特征(如两个节点在对方的通信范围内时可以进行通信)。基于以上原因,本文不再对底层环境进行建模,而是把重点放在 DTN 协议的分析上。对 DTN 协议进行评估分析需要用到大量的人为移动模型或真实世界的移动数据,

其中人为的节点移动往往只包含了有限的移动特性。另外,真实世界的移动记录数据只是粗略短时的数据或对用户群的抽样结果。由于协议经常在不同的配置环境下进行比较分析,本文通过对这两种方法的综合使用,来对 DTN 协议性能分析的结果进行补充。

本文主要对 DTN 网络模拟器 ONE(opportunistic network environment)进行介绍,它是基于 Java 开发的适用于 DTN 网络环境的路由协议和应用研究分析的模拟器,具有面向对象、离散事件驱动、可以模拟真实网络环境的特点。ONE 在单个框架中具有模拟分析多种 DTN 协议的能力,并且不像其他 DTN 模拟器那样只是关注于路由的模拟,它将移动模型、DTN 路由和可视化的图形界面整合为一体。这样 ONE 就非常容易进行扩展,并可以提供大量的结果报告和分析模型。

1 背景及其相关工作

缺少强大的专用 DTN 网络模拟器是阻碍 DTN 网络研究进展的一个重要原因,而现存的许多模拟器都是为 MANETs 设计的,如 NS2^[2]和 OMNET++^[3],这类模拟器缺少对 DTN 网络环境很好的支持。一些专门为 DTN 网络环境设计的模拟器,如 dtnsim^[4]和 dtnsim2^[5],只是单纯地关注于路由的仿真,其发布的第一个版本对现在大多数的经典 DTN 路由算法都不能实现,以后的版本针对 DTN 路由的仿真技术的改进也基本上停止了。

收稿日期: 2011-05-24; 修回日期: 2011-06-26 基金项目: 山东省优秀中青年科学家科研奖励基金项目(2010BSE14022)

作者简介: 王朕(1987-),男,山东菏泽人,硕士研究生,主要研究方向为 DTN、机会网络;王新华(1970-),男,教授,博士,主要研究方向为机会网络、社会网络(wangxhe@sdnu.edu.cn);隋敬麒(1988-),男,硕士研究生,主要研究方向为社会网络。

对于一款模拟器来说,只有路由算法是远远不够的,数据的输入和相关事件的产生(如把节点间的通信连接建立或中断的时间告诉路由协议)也至关重要。目前有许多的工程项目已经通过蓝牙设备^[6]或 WiFi^[7]收集了大量的通信记录数据(节点、持续时间、位置等),其中 CRAWAD 工程^[8]在这方面表现得较为突出,提供了相对完整的真实世界的移动记录数据。这些记录数据可以从真实世界的角度来观察移动用户之间的交互通信,为网络仿真的实际应用提供了真实依据。

虽然真实世界的记录数据能展示移动用户的真实特征,但也有其局限性,主要表现在:

- a) 进行分析的群体数量往往非常有限,实验结果可能会有一定的偏见性。
- b) 由于移动设备电池容量的限制,时间间隔的长短得不到保证。例如 Haggie iMotes^[9]工程每 5 min 发送一个 Hello 包用来发现邻居节点,如此长的时间间隔可能发现不了许多潜在的邻居节点。

因此,真实数据记录对于连通的数据问题并不能提供一个完善的解决方案。

由于节点的数量和行为很容易根据环境进行更改,人为的移动模型在仿真可行性和可扩展性方面表现良好。即使在最简单的移动模型当中,具有某些连通特性模块的模型也要比随机分布的模型更具有真实性,而有时还要考虑节点感兴趣的地点或车辆只能在街道上行驶等这样的情况^[10]。因此开发一个性能良好的移动模型至关重要。现存的移动模型有简单移动模型如 Random Waypoint^[11]、复杂移动模型如 Random Trip^[12]、群组移动模型和社区模型等。

由于在非可视化的情况下,良好的路由算法和移动模型对于人们来说十分抽象,同时对其进行编程的难度也大大增加,因此图形用户界面(GUI)的开发研究就非常必要。NS2 在模拟时可以利用 iNSpect^[13]来解决可视化的问题。但是一些 DTN 网络模拟器并没有相似的工具可用,所以具有 GUI 的 ONE 模拟器就应运而生。

2 ONE 模拟器

ONE 模拟器是由 SINDTN 和 CATDTN 工程项目开发的,由芬兰的诺基亚研究中心提供支持。ONE 是一种基于离散事件模拟引擎的,在每一步模拟时引擎都会更新许多的模块,从而实现整个模拟的功能。它能够通过使用不同的路由协议来模拟 DTN 消息的收发,并生成移动轨迹的记录。ONE 模拟器的主要功能包括节点移动的建模、节点之间的通信、路由和消息的处理、通过 GUI 观察实时的节点移动和消息传输、可视化的数据结果的收集和分析等,并可以借助如 gnuplot^[14]等工具把数据图示化。图 1 展示了 ONE 的结构^[15]。

2.1 软件结构

ONE 是一款基于 Java 的开源的 DTN 网络模拟器,它的主页^[16]提供了完整的源代码下载。模拟器中的各个部分由不同的包组成^[17],如图 2 所示。其中模拟器的核心部件(如定义 DTN 的主机及其主机间连接的类)都放在了 core 包里。GUI 相关的类放在了 GUI 包里。GUI 包里还包含一个 playfield 的子包,里面定义了一些在 playfield 的视图中显示的类。基本的用户接口类和基于文本的输出类都放在了 UI 包中。各个包的功能都是独立的,如 GUI 类从 core 包里实例化了模拟器环境

和 world 类,在 routing 包里能构造路由模块,在 movement 包里可以构造移动模块,在 report 包里可以构造报告模块等。在整个模拟过程中,路由模块和移动模块为报告模块提供了数据源。Test 包并不直接与模拟器相关,但是它包含了一系列的单个的系统测试,这些可以测试系统是否运行良好。

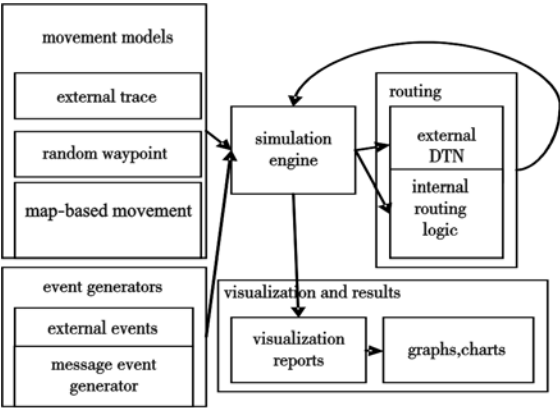


图1 ONE模拟器的整体结构

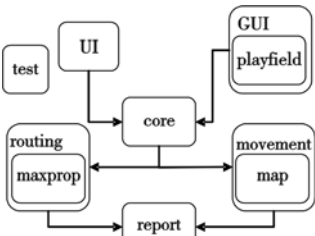


图2 ONE软件包结构

2.2 移动模型

移动模型提供了节点在仿真过程中的移动方式。它们定义了坐标、速度、停留时间等参数。最初 ONE 提供了五种移动模型: a) RWP; b) MapBasedMovement (MBM); c) ShortestPath-MapBasedMovement; d) MapRouteMovement; e) ExternalMove-ment。以上所有的移动模型都继承了 MovementModel 这个父类,而 MovemenModel 类提供了一个节点请求新路径以及询问新路径的可行性的接口,它的子类则是利用不同的模型来实现这些方法。例如, RWP 模型给出了简单的 zig-zag 路径,而 MapBasedMovement 模型则限制对象只能在拓扑图的道路上运动。在基于地图的最短路径的移动模型中, DijkstraPathFinder 这个类可以使用 SimMap 类描述的地图数据,从而来寻找拓扑图中两个节点之间的最短路径。其中地图数据也可以包含兴趣点(POIs),它是由 PointsOfInterest 类实现的。POIs 的作用在于取代节点随机选择的下一个目的地,以一定的概率到达感兴趣的地点。POIs 可以为住宅区、办公区、商店、旅游景点等建立模型。最新的 working day movement (WDM) model^[18]可以尝试表现出人类在一周工作中的移动模式,这种移动模型更具有真实性。移动模型中一些重要的类的结构如图 3 所示。

1) 基于地图的移动模型(MBM)

基于地图的移动模型是把节点限制在预先定义好的路径上移动。可以为某个节点定义不同类型的路径,也可以为一组节点定义一种路径。例如,车辆禁止在室内和人行道上运动等。其中最基本的基于地图的移动模型是 MapBasedMove-ment,这种模型初始化时把节点分布在任意两个地图点的连接路径上,然后节点就可以在路径上进行移动。当节点到达下一个地图点时,节点就会随机选择下一个地图点。如果不是仅有

返回的道路可以选择,那么节点就不会选择返回的道路,这样就避免了无谓的重复移动。一旦节点通过了 10 ~ 100 个地图点,那么节点就会暂停运动一段时间,然后重新开始运动。更加复杂的基于地图的移动模型需要使用 Dijkstra 最短路径算法在地图区域找出源点到目的点的最短路径。一旦节点到达目的地,节点就会暂停一段时间,然后重新进行运动。MapRouteMovement 模型可以使节点遵循一定的路由进行移动,如公交线路、地铁路线等。所有的基于地图的移动模型都是从 wkt^[19] 格式的文件中获取的输入数据,OpenStreetMap^[20] 允许导出真实世界的道路数据,可以很方便地进行道路拓扑图的设计工作。

2) 工作日移动模型(WDM)^[18]

RWP、MBM 等移动模型很容易理解,并且在仿真当中具有高效性。但是它们并不能生成符合真实世界记录的那种相互联系的时间分布,特别是在仿真过程中节点数量很少的应用背景下。为了使仿真更加具有真实世界的特征,为 ONE 模拟器开发了一种 WorkingDayMovement (WDM) 模型。WDM 模型通过真实人类活动的规律(如在家睡觉、在办公室工作、晚上出去聚餐等)来表现出节点移动的真实性。另外 WDM 模型包含了三种不同的移动方式,当节点移动时可以选择步行、开车或坐公交车,这样以不同的速率进行移动对路由的性能研究具有很大的影响。最后,WDM 模型还表现出了社区和社会关系模型。这里的社区划分是基于地理位置划分的。

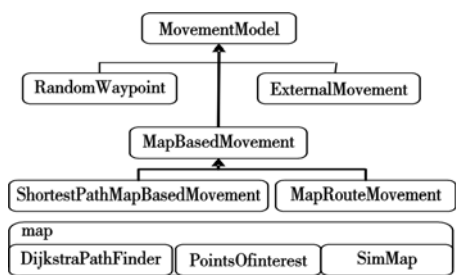


图3 movement包

2.3 路由模型

Routing 包的结构与 2.2 节提到的 movement 包的结构很相似,它能够提供路由模块。路由模块定义了在整个仿真过程中如何处理消息。其中包括了六个不同的主动路由协议模块:a) Direct Delivery (DD); b) First Contact (FC); c) Spray-and-Wait; d) PRoPHET; e) Max-Prop; f) Epidemic。这些提供了大多数重要的 DTN 路由协议的原型。其中有单消息拷贝的、n 消息拷贝的和消息无限拷贝的协议。另外还有基于统计的改进路由协议模型。routing 包的结构如图 4 所示。

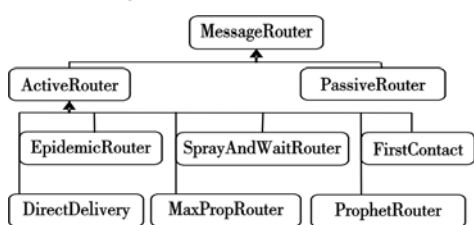


图4 routing包

为了在 ONE 模拟器当中评价新的路由协议,需要创建新的路由模块。所有的路由模块都继承 MessageRouter 这个模块,它定义了大量不同的消息相关事件和资源管理的回调函数,这些回调函数可以被模拟引擎调用。例如,一个新消息创建成功或一个消息投递到了其他节点上时,一个路由模块就需

要处理这些事件。ActiveRouter 是 MessageRouter 的子类,它提供了邻居节点传送消息以及处理成功传输和中断传输等情况的功能,同时还提供了 FIFO 和随机的缓存管理方式。开发人员如果建立自己的路由模块,需要重写、覆盖或扩展 MessageRouter 和 ActiveRouter 这两个模块的方法函数。

2.4 外部事件和报告框架

前面已经提到的路由模型可以实现新消息的分发,但是如何产生新的消息、什么时间产生等问题就需要一种导入外部事件的方法。ONE 提供了两种不同的导入事件的方法:a) 导入外部事件数据;b) 事件产生模块。其中,一个外部记录文件可以是 txt 格式的文件,文件中需要记录时间戳事件,如在某时产生新消息等。事件产生模块是一个普通的 Java 类,它可以动态地产生事件,ONE 能够同时支持这两种不同的事件产生方式。

外部事件为仿真提供了数据的输入,而数据的输出则是由报告模块完成。报告模块可以注册到节点的连通、消息转发、运动等相关事件上,这样当节点的相关事件发生时,注册后的报告模块就可以对发生的相关事件产生相对应的数据。其中 ONE 提供了大量的报告模块,这些报告模块几乎包含了全部的 DTN 网络环境性能的评测指标。

2.5 配置仿真

所有的仿真环境参数都是由配置文件给出的。配置文件都是 txt 格式的文件,大部分变量的语法为:Namespace. key = value。配置文件中也可以包含注释,注释必须以“#”开头。不管运行什么配置文件,其中 default. txt 总是运行的,它包括了一些基本的配置内容。用户自定义的配置文件只是定义了更多的设置或是重写覆盖了 default. txt 文件中的一些或者全部的设置。如果配置环境与 default. txt 中的完全不同,可以把 default. txt 置为空文档,使用自己的配置文件。这里介绍几个重要的参数。

Scenario. nrofHostGroups:在仿真过程中显示的主机群组数量。

transmitRange:节点的通信范围(meters)。

transmitSpeed:传输速率(bps)。

nrofHosts:每组的主机数量。

bufferSize:每个节点的消息缓存大小。当缓存满时,节点在丢弃旧的消息之前,不会再接收任何消息。

router:主机群组所使用的路由协议。

msgTtl:消息的生存时间。如果没有定义生存时间,则 TTL 默认为无限。

MapBasedMovement. mapFileN:地图文件的路径,可以使用不同的组合方式地图。

3 扩展模拟器

正如前面所提到的,ONE 模拟器程序是基于 Java 的且是开源的,对 ONE 模拟器进行扩展时非常方便,用户可以依照自己的思路对 ONE 模拟器进行修改和扩展。为了方便用户扩展 ONE 模拟器功能,开发者设计了几种接口,从而使用户不必修改源代码就可以对 ONE 进行扩展。换句话说,就是开发者把多种模块(如路由模块、移动模块等)作为模拟器核心的插件来使用,这几个模块都是在仿真开始时通过 Java 的 API 接口动态地加载进去的,用户仅仅需要在相应的包中创建一个属于

自己的新类,就可以对相关模块进行扩展。

如果用户想实现自己的路由算法,需要在 routing 包内创建一个新类,并要求新类必须继承于 MessageRouter 这个父类,但是为了更好地实现路由算法,创建的路由类通常继承于 ActiveRouter 这个类。其中创建自己新类的最简单方法就是覆写父类的 update 方法,其他如消息传输和交换等方法可以全部继承于 ActiveRouter 类。

新的移动模型可以继承 MovementModel 这个父类,并且至少要覆写 getInitialLocation 和 getPath 这两个方法。getInitialLocation 方法定义了仿真开始阶段节点所处的位置,getPath 方法则定义了节点请求移动的路径。另外 MovementModel 类也提供了移动速度、停留时间等移动的相关参数设定的方法。

事件产生器需要创建在 input 包内,并且它们需要实现 EventQueue 这个接口。使用这个接口,模拟核心就可以查询下一个事件发生的时间等相关参数。最后可以在 Report 包里进行定义报告模块来实现数据的输出。Report 包提供了大量自定义的关于仿真的数据统计模型,包括从消息交互率到资源消耗率等多个统计指标。用户也可以创建自己的新的报告模块,新的报告类需要继承 Report 这个父类,并实现相关的接口,不同的接口可以提供不同的数据输出。

4 ONE 的局限性

尽管 ONE 模拟器是一款很优秀的 DTN 模拟器,但它也有其缺点和局限性。为了使仿真可行,有时需要对现实世界的某些方面进行抽象或完全舍弃,这几乎是仿真软件的通病。例如,基于地图的移动模型无法体现出交通阻塞等现象。

ONE 仿真的环境缺少物理层和链路层的支持,当两个节点在彼此的通信范围内时,它们的通信速率是不变的,但是在真实世界中,由于距离或干扰等情况的发生,传输速率往往达不到预设的最大值。另外,在仿真过程中,移动设备都是假设为总是开启的,但是现实中为了节省电能,有些设备往往会切换成空闲状态,也有一些设备会周期性地对邻居节点进行探测。综上分析,ONE 仿真环境中的通信时间过于乐观。

本文可以从 GUI 看出所有节点都是显性地显示出来的,ONE 虽然能够表现出节点良好的动态性,对路由模块的评估具有高效性和客观性,但是并不能动态地发现邻居节点,缺少一种邻居节点从隐性到显性的发现过程。

5 改进 ONE

在前面的描述中,了解了 DTN 网络模拟器 ONE 的缺点和不足,针对此,本文在核心模块中添加了物理层配置接口,这样可以根据手动的环境设定,或随机地产生一些干扰因素,从而使得传输速率不能达到一个期望值,这样就更趋于真实。

如图 5 所示,本文定义的物理层配置接口实现了 ConnectionListener 这个接口中的 hostsConnected 方法,当两个节点进入到彼此的通信范围时,节点就会根据配置的 setting 文件中的 scenario.disturbpro = value1 来设置环境干扰产生的概率,通过 scenario.disturbtype = value2 来配置干扰的类型。其中本文定义了三种类型:high、medium、low,它们分别对应于不同的干扰强度。另外,物理层配置接口还可以根据距离的长短自动调节传输速率。其中分别定义了 lowwaterlevel 和 highwaterlevel 这两个参数用来定义距离的门限。如图 6 所示,这里定义的距离

低于 lowwaterlevel 的传输速率保持不变;在 lowwaterlevel 和 highwaterlevel 的传输速率缓慢地降低;高于 highwaterlevel 的随着距离的不断增大,传输速率快速地降低。这样长中短距离的不同传输速率就大大影响到了路由算法的性能评价,使得仿真更加具有真实性。

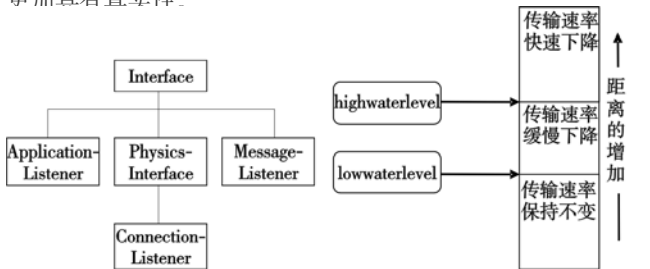


图5 物理层接口定义

图6 距离对传输速率的影响

ONE 原本的节点假设条件过于理想化,认为节点总是处于工作状态的,但是真实情况却不是如此,往往有些节点因为种种原因会处于休眠、死机等状态,而无法进行消息的发送、传输和中转。为了更好地体现这些真实情况,在核心包当中定义节点的 DTNhost 类中加入了休眠机制。其中需要在 setting 配置文件中定义 host.sleep = value3,value4 参数。当节点运行一段时间,并且系统开销超过了一定量的时候,节点就会随机选择 value3 ~ value4 之间的值进行休眠;然后在经过一段时间后,节点会恢复运行状态,可以重新发送和接收消息数据。

如图 7 所示,此时节点 p30、w107 和 w80 都处于工作状态,这时 p30 和 w80 在彼此的通信范围内可以进行通信(黑线表示连通)。经过一段时间,节点 w107 处于休眠状态,当 w80 与 w107 相遇时则不会进行通信,如图 8 所示。

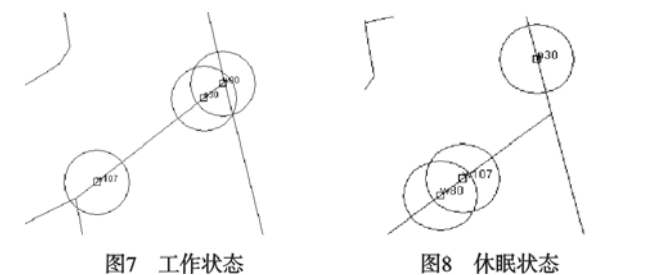


图7 工作状态

图8 休眠状态

6 仿真评价

ONE 可以在 Windows、Linux、MacOS 等多种平台上进行操作。这里假设运行的环境是 Windows 7,并已经配置好了 Java 6 JDK。下载的是 zip 格式的源代码文件。然后进行解压,解压的目录为 F:\one。打开 cmd 命令行控制台,输入命令,进入到 ONE 解压的所在目录。首先运行 compile.bat 这个批处理文件,对源代码进行编译处理;然后输入命令 one.bat mytest.txt,这样就打开了 GUI,如图 9 所示。其中 one.bat 后的可用选项有:a)在批处理模式下运行模拟,不开启 GUI,在 UI 终端输出结果;b)configfilename.txt,模拟器运行所需的配置文件(如果不加此参数的话默认执行 default.txt 文件);c)run-index(count),可以使用不同的环境配置参数运行多次,这里使用的分别是名称为 mytest1.txt 和 mytest2.txt 的仿真环境配置文件。

Mytest1.txt 和 mytest2.txt 配置文件中都使用的是济南市区的城市道路拓扑图,并设置了 1 000 个移动节点分布在拓扑图中,其中移动节点的缓存大小设置为 100 MB,传输速率为 100 kbps。把移动节点分为三组:第一组 ID 为 Bus,节点数量为 100 个,通信接口设定为 WiFi,通信范围为 100 m,并且采用

的是 BusMovement 移动模型,这种模型是通过预先定义的移动轨迹进行移动的;第二组 ID 为 Person,节点数量为 500 个,采用的是 WDM 移动模型,其中定义的办公室数量为 200 个,工作时间设定为 10 ~ 100 min,每个节点拥有私家车的概率为 0.3;第三组 ID 为 Car,节点数量为 400,采用的是 ShortestPath-MapBasedMovement 移动模型,节点移动到目的地的停留时间设定为 10 ~ 100 s。主机群组所使用路由协议是 EpidemicRouter。其中 mytest2.txt 中加入了物理层配置接口和节点模式切换等参数。这里设置的干扰强度为 high,通过实现环境的干扰与原有的环境配置来进行比较。本文给出了一些配置文件中的重要参数,如表 1 所示。

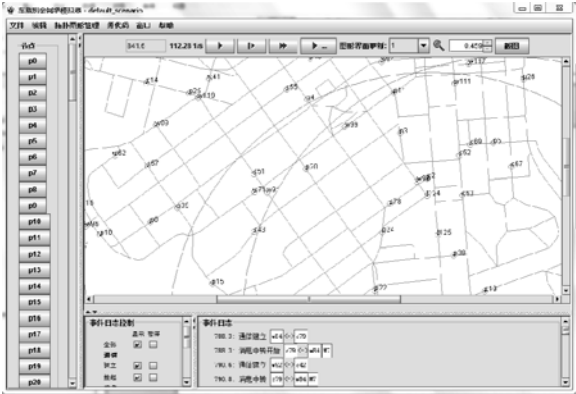


图9 GUI

表 1 重要仿真配置参数

配置参数	数值	配置参数	数值
Group. msgTtl	1433	Group. router	EpidemicRouter
Group. transmitSpeed	100KBps	Group1. groupID	Bus
Group. speed	0.5 ~ 1.5	Group1. movementModel	BusMovement
Group. nrOfOffices	200	Group1. interface	WiFi
Group. workDayLength	28800	Group2. groupID	Person
Group. probGoShopping-AfterWork	0.5	Group2. nrofHosts	500
Group. nrOf-MeetingSpots	24	Group2. movement-Model	WorkingDay-Movement
Group. nrofHosts	1000	Group2. busControlSystemNr	8
Group. bufferSize	100 MB	Group2. nrOfOffices	200
Group. movement-Model	MapBased-Movement		

仿真过程结束后,结果报告会放在 ONE 目录下的 reports 子目录中,然后可以利用 gnuplot 实现数据结果的图形化。

在运行了 20 000 s 的仿真中,得到了 EPI 路由算法在无环境干扰和干扰很强的环境下的消息交互率对比情况,如图 10 所示。在无干扰的仿真环境中,EPI 算法在 16 000 s 前,消息交互率随着时间的增长而不断提高,而配置的缓存容量并不是无限的,所以在 16 000 s 后,由于空闲缓存的减少,没有缓存管理机制的 EPI 路由算法的消息交互率呈现下降的趋势。而在具有干扰的环境中,即使是采用洪泛机制的 EPI 路由算法,消息交互率也会有所降低。这主要是因为外界的干扰,传输环境可能会变得异常恶劣,丢包率会大大增加。这与笔者从剑桥大学的 Hagggle project^[21] 中训练出的真实数据具有很相似的结果。

图 11 为这两种环境的消息交互累计时延对比。总体来看,随着仿真的运行,无干扰环境下的交互时延低于有干扰的

环境,并且保持了较高的稳定性。EPI 算法在无干扰环境中能很好地进行消息的投递,但是在干扰很强的环境中消息则会很少投递成功。根据统计,EPI 在无干扰环境中的平均时延在 6 000 s 左右;而在干扰很强的环境中则表现不好,平均时延在 8 000 s 左右。

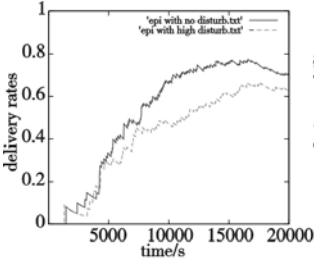


图10 消息交互率

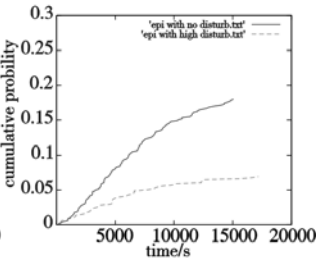


图11 消息时间延迟

图 12 为 EPI 路由算法在这两种不同的环境中,随着距离的增长,消息交互率的变化情况。从图中可以看出,在无干扰的环境中,EPI 路由算法消息交互率相对稳定;在强干扰的环境中,由于这里设置的 lowwaterlevel 和 highwaterlevel 分别为 3 km 和 6 km,随着距离的增长,传输速率降低,相对较大的消息出现丢包的概率就会增大,这样消息交互率就会降低。

图 13 为 EPI 路由算法的消息距离延迟,即消息交互时所经历过的跳数。从图中可以看出,EPI 路由算法在具有休眠模式环境的仿真过程中相对于没有多模式切换的环境具有较低的距离延迟,平均跳数只有 3 跳左右,这与 Cityware project 中的平均跳数一样,而总处于工作状态的 EPI 平均跳数则是在 6 跳左右。在节点具有休眠状态的环境中,由于很多节点会处于休眠状态,这样节点就不会进行消息的发送和接收,节点的密集程度就会降低。

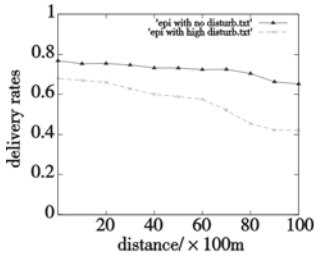


图12 距离对交互率的影响

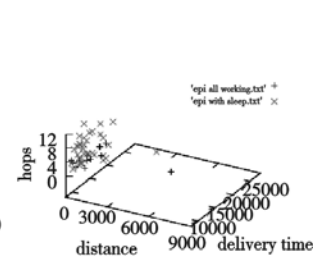


图13 消息距离延迟

从上述得出的结果分析来看,环境的影响还是很大的,为了更好地表现出真实性,加入物理层配置接口是必要的。另外,从仿真过程中发现,经典的 EPI 算法即使在很恶劣的环境中还是具有相当优势的,为了更好地发挥 EPI 算法的优势,在原来算法的基础上加上智能缓存管理机制就成为了必然。下一步工作就是研究在 EPI 的路由方式上加上智能缓存管理机制,在有限的系统资源条件下更好地发挥其性能。

7 结束语

本文介绍了 DTN 网络模拟器 ONE 的研究开发背景,描述了 ONE 的结构、特征、用途以及扩展 ONE 的方法。实例演示结果表明了 ONE 对路由模块性能分析的可行性。最后指出了 ONE 的不足之处,并对这些不足之处进行了扩展改进,使得 ONE 仿真更具真实性。作为未来的工作重点,笔者将探索一种开发动态导航系统的方法。另外将继续在此基础上扩展路由模块,提高路由模块的实现效率。

参考文献:

- [1] FALL K. A delay-tolerant network architecture for challenged Inter-nets[C]//Proc of Conference on Applications Technologies, Architectures, and Protocols for Computer Communications. New York: ACM Press, 2003: 27-36.
- [2] The network simulator NS-2[EB/OL]. (2010-05). <http://www.isi.edu/nsnam/ns/>.
- [3] VARGA A. The OMNET ++ discrete event simulation system[C]//Proc of the European Simulation Multiconference. 2001: 319-324.
- [4] JAIN S, FALL K, PATRA R. Routing in a delay tolerant network[J]. ACM SIGCOMM Computer Communication Review, 2004, 34(4): 145-158.
- [5] University of Waterloo. DTNSim2 delay-tolerant network simulator[EB/OL]. (2010-05). <http://watwire.uwaterloo.ca/DTN/sim/>.
- [6] HUI Pan, CHAINTREAU A, SCOTT J, *et al.* Pocket switched networks and human mobility in conference environments[C]//Proc of ACM SIGCOMM Workshop on Delay-Tolerant Networking. New York: ACM Press, 2005.
- [7] EAGLE N, PENTLAND A. Reality mining: sensing complex social systems[J]. Personal Ubiquitous Computing, 2006, 10(4): 255-268.
- [8] Community resource for archiving wireless data at dartmouth[EB/OL]. (2010-03). <http://crawdad.cs.dartmouth.edu/>.
- [9] Diot. Hagggle project[DB/OL]. (2004). <http://www.hagggleproject.org>.
- [10] CHOFFNES D R, BUSTAMANTE F E. An integrated mobility and traffic model for vehicular wireless networks[C]//Proc of the 2nd ACM International Workshop on Vehicular Ad hoc Networks. New York: ACM Press, 2005.
- [11] JOHNSON D B, MALTZ D A. Dynamic source routing in Ad hoc wireless networks[M]. [S. l.]: Kluwer Academic Press, 1996.
- [12] Le BOUDEDEC J Y L, VOJNOVIC M. Perfect simulation and stationarity of a class of mobility models[C]//Proc of the 24th annual Joint Conference on IEEE Computer and Communications Societies. 2005: 2743-2754.
- [13] KURKOWSKI S, CAMP T, MUSHELL N, *et al.* A visualization and analysis tool for NS-2 wireless simulations[C]//Proc of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems. 2005: 503-506.
- [14] JANERT P K. Gnuplot in action-understanding data with graphs[M]. [S. l.]: Manning Press, 2009.
- [15] KERÄNEN A, OTT J, KÄRKKÄINEN T. The ONE simulator for DTN protocol evaluation[C]//Proc of the 2nd International Conference on Simulation Tools and Techniques. 2009.
- [16] TKK/COMNET. Project page of the ONE simulator[EB/OL]. (2008). <http://www.netlab.tkk.fi/tutkimus/dtn/theone>.
- [17] TKK/COMNET. Project page of the ONE simulator[EB/OL]. (2008). http://www.netlab.tkk.fi/tutkimus/dtn/theone/javadoc_v141.
- [18] EKMAN F, KERÄNEN A, KARVO J, *et al.* Working day movement model[C]//Proc of the 1st ACM Press/SIGMOBILE Workshop on Mobility Models. New York: ACM, 2008.
- [19] Well known text[EB/OL]. (2011-03). http://en.wikipedia.org/wiki/well-known_text.
- [20] Wiki. openstreetmap[EB/OL]. (2011-03). <http://www.openstreetmap.org>.
- [21] LEGUAY J, LINDGREN A, SCOTT J, *et al.* Opportunistic content distribution in an urban setting[C]//Proc of SIGCOMM Workshop on Challenged Networks. New York: ACM Press, 2006: 137-145.

(上接第 252 页)

5 结束语

本文提出的 TR-SLM 级联算法, 较好地将经过优化的 TR 算法与分段扰码插入的 SLM 算法结合起来, 充分利用改进的简化 TR 算法在降低计算复杂度的同时保持其他性能上的优势, 以及改进的 SLM 算法在实现过程中不需要传输边带信息的优势, 既克服了传统 SLM 算法的缺点, 又继承了 TR 算法和 SLM 方案的优点。从仿真结果可知, TR-SLM 级联算法不仅能有效降低系统传输的冗余信息, 而且能有效降低峰均比。将 TR-SLM 级联算法用于多天线的 MIMO-OFDM 系统, 不仅能达到改善系统峰均比性能的目的, 还大大降低了系统复杂度, 提高了系统频带利用率, 因此算法具有较大的实用价值。

参考文献:

- [1] SEUNG H H, JAE H L. An overview of peak-to-average power ratio reduction techniques for multicarrier transmission[J]. IEEE Wireless Communications, 2005, 12(2): 56-65.
- [2] YANG L, CHEN R S, SIU Y M, *et al.* PAPR reduction of an OFDM signal by use of PTS with low computation complexity[J]. IEEE Trans on Broadcasting, 2006, 52(1): 83-86.
- [3] WANG L, WANG Y. MIMO-OFDM peak-to-average power ratio reduction by two-dimensional permutation[J]. Electronics Letters, 2007, 43(10): 579-580.
- [4] FOSCHINI G J. Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas[J]. Bell Labs Technical Journal, 1996, 1(2): 41-59.
- [5] FISCHER R F H, HOCH M. Peak-to-average power ratio reduction in MIMO OFDM[C]//Proc of IEEE International Conference on Communications. Piscataway: IEEE Press, 2007: 762-767.
- [6] YUNG L L, YOUNG H Y, WON G J. Peak-to-average power ratio in MIMO-OFDM systems using selective mapping[J]. IEEE Communications Letters, 2003, 7(12): 575-577.
- [7] HEO S J, NOH H S, NO J S, *et al.* A modified SLM scheme with low complexity for PAPR reduction of OFDM systems[J]. IEEE Trans on Broadcasting, 2007, 53(4): 804-808.
- [8] 王文博, 郑侃. 宽带无线通信 OFDM 技术[M]. 2 版. 北京: 人民邮电出版社, 2007: 1-251.
- [9] 周恩, 张兴. 下一代宽带无线通信 OFDM 与 MIMO 技术[M]. 北京: 人民邮电出版社, 2008: 1-177.
- [10] HE Xiao-fan, JIANG Tao, ZHU Guang-xi. A novel simplified tone reservation combined with cross antenna rotation and inversion to reduce PAPR for MIMO-OFDM system[J]. Wireless Communication and Mobile Computer, 2010, 11(10): 1406-1414.
- [11] MARUYAMA T, MARUTA K, MASHINO J, *et al.* Control information transmission without resource consumption for PAPR reduction of OFDM signal[J]. Electronics Letters, 2009, 45(8): 406-408.
- [12] ALSUSA E, YANG L. Selective post-FFT amplitude randomising for peak-to-average power ratio reduction in orthogonal frequency-division multiplexing-based systems[J]. Communications Letters, 2008, 12(4): 553-556.
- [13] IRUKULAPATI N V, CHAKKA V K, JAIN A. SLM based PAPR reduction of OFDM signal using new phase sequence[J]. Electronics Letters, 2009, 45(24): 1231-1232.