

ndnSIM 2:更新的NS-3 NDN模拟器

Spyridon Mastorakis, Alexander Afanasyev, Ilya Moiseenko 和 Lixia Zhang

来自加州大学洛杉矶分校

{mastorakis, aa, iliama, lixia}@cs.ucla.edu

f

命名数据网络(NDN)通信范式从IP原则的根本背离需要通过实验进行广泛的评估,而模拟是实现大规模实验的必要工具。我们在2012年6月发布了第一个版本的ndnSIM,这是一个基于ns-3的开源NDN模拟器。从那时起,ndnSIM经历了大量的开发,产生了ndnSIM 2,其第一个版本于2015年1月发布。本文报道了更新后的模拟器的设计和功能。更新模拟器的目标是使仿真平台与NDN研究的最新进展相匹配。因此,它使用NDN-cxx库(带实验扩展的NDN c++库)和NDN转发守护进程(NFD)在仿真环境中使用真实代码进行实验。

1 简介

命名数据网络(NDN)[1,2,3,4]代表了对今天的互联网架构的根本背离,该架构命名通信端点,并渴望成为TCP/IP协议堆栈的继承者。因此,NDN架构中的各种设计选项需要在大规模实验中进行评估。然而,用现实世界的基础设施进行这样的实验是不可行的,基于模拟的评估变得必要。ndnSIM的主要目标一直是为NDN社区提供一个基于NS-3模拟器框架[5]的通用、用户友好和开源的仿真平台。

ndnSIM的第一个公开发布是在2012年6月发布的。从那时起,ndnSIM已成为全球许多研究人员使用的流行工具。在撰写本文时,ndnSIM邮件列表已经有超过400个订阅者,并且基于使用ndnSIM所做的研究已经发表了290多篇论文。自第一个ndnSIM发布以来,NDN团队已经发布了协议的更新版本,部分特点是新的模块化NDN转发守护进程(NFD)[7,8]和NDN-cxx库(NDN c++库与实验扩展)[9],支持NFD实现,并用于其他基于c++的应用程序。

这个新的ndnSIM版本的目标是使模拟平台与NDN研究的最新进展相匹配,并巩固代码开发工作:ndnSIM的核心现在完全由NFD和NDN-cxx代码提供支持,并进行了一些小的更新。有了这个变化,ndnSIM 2提供了最高的仿真结果保真度,以及允许简单的模拟实验转移到现实世界,反之亦然。总而言之,ndnSIM 2有

与第一个版本相比,有以下主要的增强和特性:

- 所有NDN转发和管理直接使用NFD和NDN-cxx的源代码实现。
- 报文格式更改为最新的NDN报文格式[9]。
- 针对ndn-cxx库编写的真实应用程序可以适应在模拟环境中运行。

这个版本的模拟器,就像前一个一样,以模块化的方式实现,使用不同的c++类来模拟每个NDN实体的行为:与本地应用程序和其他模拟节点通信的面孔,NFD的转发信息表(FIB),未决兴趣表(PIT)和竞赛商店(CS)等。这种模块化结构允许轻松修改或替换任何组件,而不会对其他组件产生影响或影响最小。此外,新版本提供了更广泛的接口和帮助程序集合,以执行每个组件以及NDN流量的详细跟踪。

为了进一步改善用户体验,我们鼓励社区通过提交bug报告向我们提供有价值的反馈。我们也欢迎新功能开发的请求。¹关于模拟器、基本示例和教程的更多信息可在ndnSIM网站上获得:<http://www.ndnsim.net/>。

2 design

ndnSIM 2的设计目标是实现与命名数据网络转发器(NFD)的完全集成[7,8]。在本节中,我们将介绍ndnSIM的总体设计,并演示其主要结构组件及其相互交互的方式。

2.1 设计总结

ndnSIM 2提供了与NFD的集成,同时它还允许模拟针对c++库编写的应用程序。通过这种方式,ndnSIM为研究人员和开发人员提供了一个集成的模拟环境,以大规模部署和评估他们的实际应用程序。

1. Bug reports and feature recommendations can be submitted on the NDN project issue tracking system website: <http://redmine.named-data.net/projects/ndnsim>

表1:ndnSIM 2和ndnSIM 1.0的组件比较

nnsim组件2	组件存在于从nnm10引入的功能/设计更改中继承的功能/设计原则中	ndnSIM的核心组件	ndnSIM2
ndn: l3协议	是的	ndnSIM的核心组件	NFD集成
nfd:货代	以ndn::ForwardingStrategy形式存在	-	由于与NFD集成，将报文转发分为转发管道和转发策略决策两部分
nfd:face:LinkService	以ndn:Face的形式存在	ndn的基类:AppLinkService和ndn::NetDeviceLinkService	由NFT实现的nfd::Facel的链接服务抽象
rdn::AppLinkService	作为ndn::AppFace存在	支持与应用程序通信	nfd:face: linkservice抽象的实现
ndn: netdevicelinkserviceexisting as	ndn:NetDeviceFace	允许与其他模拟对象通信	nfd:face:LinkService抽象的实现
ndn:脸	不	节点	支持针对ndn-cxx库编写的真实应用程序的模拟
ndn::cs	是的	相同的设计	NFD集成
nfd:	以ndn::cs形式存在	-	1)利益选择者处理
nfdPi	以ndn:pit的形式存在	-	2)缓存策略不灵活，由NFD实现抽象由
nfd:Fib	以ndn:fib形式存在	-	NFD实现抽象
nfd:fw::Strategy	以ndn::ForwardingStrategy形式存在	-	1)每个命名空间策略
应用	是的	等价功能错误功能	2)不同的内置策略使用ndn-cxx
跟踪帮助程序	是的	-	库直接从NED跟踪事件

作为NFD集成的结果，在模拟节点上安装NDN协议栈(NDN::L3Protocol)会创建NFD实例并将其与节点关联。用于任何NDN转发实验的代码(例如，自定义转发策略)可以直接在ndnSIM中使用，反之亦然。此外，为不同的命名空间选择不同的转发策略的能力，对兴趣选择器和加密操作的全功能支持，以及全功能NDN数据包格式的使用确保了模拟的最大真实性。与ndnSIM 1.0相比，ndnSIM 2增加了内存和CPU需求，但仍然允许在通用硬件上进行大规模模拟。每个CS和PIT条目的内存开销如表2所示。ndnSIM 1.0每时钟秒可以处理大约53K个数据包(兴趣和数据)，而ndnSIM 2每时钟秒可以处理大约17K个数据包(兴趣和数据)。

ndnSIM的NDN栈是作为网络层协议模型实现的，可以运行在任何可用的链路层协议模型(点对点、CSMA、无线等)之上。除此之外，模拟器还提供了大量的帮助程序(NDN堆栈、应用程序、FIB、全局路由、链路控制和策略选择帮助程序)和跟踪程序(包级、应用程序级、内容存储)，以执行每个组件的详细跟踪行为，以及NDN流量。

这个ndnSIM版本的基本组件如图1所示，表1给出了模拟器新版本和前一个版本的设计原则和特性的比较。这些组件也列在下面：

- **ndn::L3Protocol:** NDN栈实现的NS-3抽象。它的主要任务是初始化参与仿真场景的每个节点的NFD实例，并提供跟踪源来测量NDN性能(发送/接收兴趣和数据，满足/未满足兴趣)。
- **NFD:**命名数据网络转发守护进程的实现，包括：

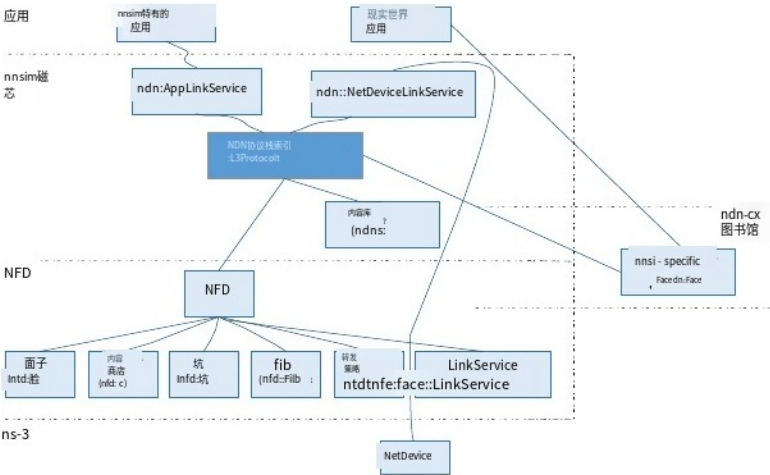


图1:ndnSIM设计组件结构图

- nfd::Forwarder:** NFD的主要类，拥有NDN路由器节点的所有面和表，实现NDN转发管道。
- nfd::Face:** NFD Face抽象，实现所需的通信原语，以实际发送和接收兴趣和数据包。
- nfd::face::LinkService:** NFD LinkService抽象的基类。LinkService在网络层报文(interest、Data、nack)和链路层报文(TLV块)之间进行转换。
- nfd::face::Transport:** NFD Transport抽象的基类。Transport为一个面的链路服务提供尽最大努力的包交付服务。
- nfd::Cs:** NFD使用的数据包缓存。当前版本的ndnSIM还包括从上一个版本移植过来的旧的**ndn::ContentStore**抽象(见下文)，以便为模拟内容存储操作提供更丰富的选项(nfd::Cs在缓存替换策略方面还没有那么灵活)。
- nfd::Pit:** NFD的未决利息表(Pending Interest Table)

TABLE 2: Average memory overhead of ndnSIM 2 and ndnSIM 1.0

输入类型	ndnSIM 1.0	ndnSIM 2
每个CS条目	1.3 Kb	2.8 Kb
每个PIT条目	0.8 Kb	5.1 Kb

跟踪向上游一个(或多个)内容源转发的兴趣包。通过这种方式,数据可以向下游发送给一个(或多个)请求者。

-ndf::Fib: FIB(转发信息库)用于向一个(或多个)潜在源转发兴趣报文。

-ndf::fw::Strategy: NFD中的转发策略决定是否、何时、何地转发Interest报文。ndf::fw::Strategy是一个抽象类,需要由所有内置或自定义的转发策略来实现。

-ndf::measurements::measurements: 表, NFD在其中存储有关名称前缀的测量信息。转发策略使用该表。

-ndf::strategy choice::StrategyChoice: 该表包含为每个命名空间选择的转发策略。

· ndn::AppLinkService: 实现ndf::face::LinkService抽象,以便与应用程序通信。

· ndn::NetDeviceLinkService: ndf::face::LinkService 抽象的实现,以实现与其他模拟节点的通信。

· ndn::cs::*: ndnSIM 1.0中实现的CS结构。它包括许多替代策略,总体而言,比当前NFD的CS实现更加灵活。

· 基本NDN应用: 实现内置的NDN消费者和生产者应用,可以生成和接收NDN流量。这些应用包括用户可以在模拟场景中配置的参数,从而根据用户自定义的模式生成NDN流量。

· 跟踪助手: 一组跟踪助手,用于简化关于仿真的各种必要统计信息的收集和聚合,并将这些信息写入文本文件中。

· ndnSIM助手: 一个助手的集合,它简化了将在模拟执行期间使用的各种参数的配置。

我们使用LinkService抽象来实现基本的通信原则,而不是实现我们自己的Transport。这是一个优化数据包处理和避免不必要的内存操作的解决方案。通过这种方法,我们还设法维持了模拟节点发送/接收的每个数据包所附加的NS3数据包标签,否则这些标签将会丢失。

2.2 核心NDN协议

ndnSIM体系结构的核心组件是ndn::I3协议。该组件作为NDN协议栈的整合器,可以以类似于其他网络协议

栈(如IPv4和IPv6)的方式安装在每个模拟节点中。当它安装在NS-3节点上时,它执行NFD实例的初始化,创建必要的NFD管理器(FibManager, facm - manager, StrategyChoiceManager),表(PIT, FIB, StrategyChoice, 测量)和特殊面(即Null Face, Internal Face)。除此之外,ndn::I3协议类定义了一个API来处理使用AddFace方法向NFD注册新的ndf::Face实例,并提供ndn级数据包跟踪的NS-3 TraceSource入口点。

2.3 命名数据网络转发守护进程(NFD)

ndnSIM集成了NFD代码库,并对模拟环境进行了一些小的调整,²在本节的其余部分,我们将描述NFD结构,并列出我们在集成过程中面临的主要挑战。

2.3.1 NFD内部结构

NFD的基本模块如下:

- **ndn-cxx库、核心和工具**, 提供不同NFD模块之间共享的各种公共服务。
- 在各种低级传输机制之上实现通信抽象的面孔。
- **表**, 实现内容存储(CS)、未决兴趣表(PIT)、转发信息库(FIB)、策略选择、测量和其他数据结构,以支持NDN数据和兴趣包的转发。
- **转发管道**, 在数据包或PIT入口上操作的一系列步骤。单个管道由特定事件触发,例如接收到Interest,检测到接收到的Interest已被循环,当Interest准备好从Face转发出去时,等等。
- **转发策略**, 一个关于兴趣转发的决策者,它附加在管道的末尾或开始。
- **管理**, 允许应用程序配置NFD和设置/查询NFD的内部状态。

2.3.2 NFD整合的挑战

为了实现这种整合,我们必须解决以下挑战:

- 我们必须在NFD中启用模拟时间。因此,我们利用ndn-cxx库提供的CustomClock类,将nnsim时间转换为系统时钟::时间点和稳定时钟::时间点。
- NFD的调度器被重定向到ns3::Simulator,以便NFD可以调度将由模拟器执行的事件。

2. Because of these required adaptations, the version of NFD that ndnSIM is based on tends to be slightly behind the latest version of NFD/ndn-cxx

为了优化NFD通过其管理协议与其管理器交互时使用的签名过程，我们设计了一个自定义密钥链，在模拟期间提供高性能(即较小的加密开销)。然而，对于需要真实加密操作的模拟，可以在模拟场景中选择使用全功能的钥匙链结构。

NFD的转发管道必须使用beforeSatisfyInterest和beforeExpirePendingInterest信号进行扩展，以便在模拟器中启用对SatisfiedInterests和timedoutinterests事件的跟踪。

我们使用专门设计的配置文件在内部启用NFD参数的可配置性，以避免解析原始外部文件的开销，从而优化模拟过程。

2.3.3 人脸抽象

它类似于之前的nnsim版本的相应抽象。这个抽象(nfd::Face)包含了处理Interest和Data数据包所需的底层通信原语。正如[7]中提到的，这些原语包括发送Interest/Data数据包和终止Face上的通信的函数。

2.3.4 ndnSIM specific ndn::Face

与ndnSIM绑定的ndn-cxx库版本包含了ndn::Face的修改版本，用于直接向NFD的模拟实例发送和接收Interest和Data数据包。通过此修改，ndnSIM支持模拟针对NDN-cxx库编写的真实NDN应用程序。关于如何模拟此类应用程序的详细指南可在ndnSIM网站上获得：<http://ndnsim.net/2.1/guide-to-simulate-real-apps.html>。

2.3.5 LinkService抽象

NFD的链路业务需要在网络层报文(兴趣、数据和nack)和链路层报文(TLV块)之间进行转换。在ndnSIM中，我们实现了一个特殊的链路服务基类(ndn::TracingLinkService)，它被ndnSIM应用程序(ndn::AppLinkService)和第二层设备(ndn::NetDeviceLinkService)使用。

2.3.6 NFD的内容库

在NDN通信模型中，内容库为数据包提供了网络内缓存。到达的数据包将尽可能长时间地放在缓存中，以便满足将来可能请求相同数据的兴趣。这样可以增强协议性能，使NDN对丢包和错误具有鲁棒性，并具有固有的多播能力。

与许多其他转发组件一样，这个版本的ndnSIM使用来自NFD代码库的内容存储实现。这个实现充分考虑了兴趣选择器，但是在缓存替换策略方面还不够灵活。扩展CS灵活性的特性目前正在积极开发中，目前，我们还将旧的ndnSIM 1.0内容库移植到新的代码库中，这将在2.6中讨论。

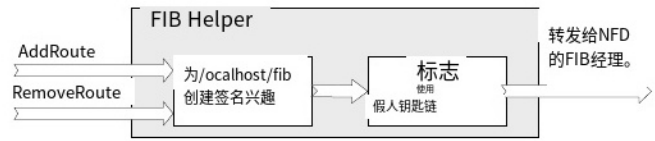


图2:FIB helper的操作

2.3.7 未决利息表(PIT)

在我们的实现中，NFD的nfd::Pit类被用作PIT抽象。PIT维护已向上游的一个(或多个)匹配数据的潜在数据源转发的兴趣包的状态。它提供了向数据消费者反向转发数据包的方向。除此之外，PIT还包含最近满足的兴趣包，用于防止环路。

有关PIT结构和对其执行的操作的更多信息，可以参考[7]。

2.3.8 转发信息库(FIB)

使用NFD的nfd::Fib类作为FIB抽象。转发策略使用这个抽象来对潜在的内容源进行兴趣转发。对于每个需要转发的兴趣，在FIB上执行最长前缀匹配查找。

FIB仅通过FIB管理协议进行更新，该协议由FIB管理器在NDN转发端进行操作。为了简化常见操作，我们创建了一个FIB助手，它为高级FIB操作准备特殊签名的兴趣命令，并将它们发送给FIB管理器。目前，FIB助手实现了两个高级操作(图2):

AddRoute: 创建新的FIB表项，向FIB表项添加路由，或者更新FIB表项中已有记录的开销。

RemoveRoute: 从FIB表项中删除一条路由记录(NextHop记录为空的FIB表项将自动删除)。

发送到FIB管理器的Interest命令使用前一节提到的自定义密钥链进行签名，这是专门设计用于消除模拟目的的签名加密开销的。如果有必要，可以通过切换到ndn-cxx库提供的标准KeyChain来重新启用全功能的加密支持。

2.3.9 转发策略抽象

如前所述，NFD的转发策略抽象对兴趣转发进行决策。也就是说，一个Interest是否会被转发，上游面，将被转发到哪里，以及何时转发到所选的上游面。ndnSIM/NFD提供了一个抽象接口(策略API)，它提供了转发策略的基本实现，而不需要重新实现完整的兴趣处理管道。转发管道的概述如图3所示，并在本节的其余部分进行详细描述。

实现的转发管道允许对每个命名空间选择特定的转发策略。这

每个名称空间的转发策略在策略选择表中注册和维护。策略选择表通过管理协议更新，由策略选择管理器操作。与FIB操作类似，我们创建了一个策略选择助手，当在模拟场景中请求策略选择时，它准备并向管理员发送特殊签名的兴趣命令。

- 广播:将每个兴趣转发到所有上游面。
- 客户端**控制策略**:允许本地消费者应用程序选择每个发送的兴趣包的传出面。
- 最佳**路由**:以最低的路由成本将兴趣包转发到上游面。
- NCC:重新实现CCNx 0.7.2默认策略。

如果策略需要存储信息，则需要确定该信息是否与名称空间或兴趣相关。与名称空间相关但不特定于某个兴趣的信息应存储在Measurements条目中;与利益相关的信息应存储在PIT项、PIT下游记录或PIT上游记录中。在做出此决定之后，需要声明从StrategyInfo类派生的数据结构。在现有的实现中，这样的数据结构被声明为嵌套类，因为它为特定于策略的实体提供了自然的分组和范围保护，但不需要遵循相

最后一步是至少实现“After Receive Interest”触发器和下面列出的其他三个触发器中的任何一个(或一个都没有):

- **收到利息后:**当收到利息后, 通过必要的检查, 需要转发时, 入息管道调用此触发器, 并提供PIT表项、入息包和FIB表项。
- **在满足兴趣之前:**当一个PIT项被满足时, 在数据被发送到下游面(如果有的话)之前, 传入数据管道用PIT项、数据包及其传入面调用此触发器。
- **Before Expire Interest:**当一个PIT项由于在所有记录过期之前未被满足而到期时, 在它被删除之前, 未满足的兴趣管道将使用PIT项调用此触发器。

· **拒绝未决利息:**当进入利息拒绝管道时触发。

2.4 应用链接服务

该类支持模拟应用程序与NDN网络的通信。具体来说, 这个抽象提供了发送兴趣数据包的功能

3 当前版本和未来计划的限制

尽管我们为开发人员提供了一个API，以便使用nnsim来模拟真实的应用程序，但这些应用程序必须满足某些要求。这些要求主要是由真实和模拟应用程序之间的不同逻辑提出的(因为它们需要在NS3和nnsim中实现)。这些要求的详细列表可以在我们的网站上找到:<http://ndnsim.net/guide-to-simulate-real-apps.html#requirements>。

NFD 0.4 及以上版本处理NACKs，因此支持NDNLPv2。在这个版本中，nnsim不支持NDNLPv2，因此，还不能用于模拟跨模拟节点的网络级NACKs。这将在nnsim的下一个版本中解决。

4 相关工作

近年来，对NDN研究的兴趣有所增长。因此，开发通用和方便的方法来评估拟议的NDN研究方法是绝对必要的。

实现这一目标的第一个也是最流行的方法之一是以以前版本的nnsim[6]。这个版本与当前版本完全相同，以模块化的方式实现，并针对仿真目的进行了优化。但是，之前的版本包含了NDN数据包转发的独立实现，并使用了已弃用的NDN数据包格式。此外，nnsim 1.0从未实现NDN选择器的全功能处理，因此在模拟结果的准确性方面存在局限性。

Chiocchetti等人提出了另一个现有的研究成果[10,11]。ccnSim是一个可扩展的块级模拟器，适用于分析NDN网络的缓存性能。它是在c++中使用OMNeT++框架开发的。但是，它主要针对NDN路由器的各种缓存替换策略的实验进行优化，并没有提供任何转发过程的灵活性。因此，ccnSim不能用于NDN体系结构的重要核心组件，即转发策略层的实验。

CCN-lite[12]是CCNx-NDNx协议的轻量级实现。它提供了一种使用OMNeT++仿真平台的仿真模式。CCN-lite支持块级和包级的调度，以及包分段。它还支持没有任何IP层的可能的本机部署。然而，这一努力主要是为了在资源受限的设备上运行，并没有优化以提供高性能，因为它的数据结构依赖于链表。

内容中心网络包级模拟器(CCNPL-Sim)[13]是Orange实验室开发的另一个NDN模拟器。CCNPL-Sim在SSim模拟器中使用组合广播和基于内容的路由方案(CBCB)[14]实现来处理事件管理和基于名称的转发和路由。尽管SSim仿真调度程序的有效性，强制使用CCNB使得其他路由协议(如OSPFN[15]和NLSR[16])

的评估不可能，从而限制了该模拟器的实验范围。

最近推出的另一项工作是Mini-CCNx[17]。Mini-CCNx是mininet - hifi的一个分支，专门为支持CCNx-NDNx节点的仿真而定制。它的主要目标是为执行的测试添加一个真实的行为。由于Mininet的基于容器的仿真特性和简单的配置GUI界面，Mini-CCNx提供了灵活性。然而，它基于NDNx的数据包格式，这是NDN通信模型的过时版本。它也主要侧重于模拟节点硬件，而不是通信模型本身。

跨现有研究网络试验台/基础设施(例如，GENI[18]，开放网络实验室(ONL) [19]，Emulab[20]等)的计算资源的丰富也为进行实时研究实验提供了有价值的选择。这些试验台既提供了研究人员评估其设计所需的硬件系统，也提供了软件系统。然而，为了配置和管理所有委托资源而引入的复杂性以及有限的实验规模是导致研究人员求助于模拟的两个关键原因。

5 摘要

在这个更新的模拟器中，我们通过将命名数据网络转发守护进程(NFD)与ndnSIM集成，并直接使用NDN -cxx库和最新的NDN数据包格式，集中精力提供更真实的模拟行为。ndnSIM为大规模实验提供了框架，而其模块化设计为研究人员提供了灵活性，可以在对其实现的其他部分进行最小(如果有的话)更改的情况下修改其组件。有关当前版本和其他文档的详细信息可在ndnSIM网站上获得:<http://ndnsim.net>。

我们真的希望NDN社区会发现ndnSIM是一个有价值的工具，我们期待着收到社区无价的反馈，以便进一步改进模拟器。

6 附录

在本节中，我们介绍了本文档的修订历史。

6.1 修订历史

· 修订版2(2016年11月11日):ndnSIM 2.1和2.2的更新:

- 增加了针对ndn-cxx库和ndnSIM特定的ndn::Face编写的应用程序的仿真描述。
- 将ndnSIM的AppFace和NetDeviceFace类重命名为AppLinkService和netdevicelinkservice - service，并更新了各自的描述。
- 增加对NFD各参数可配置性支持的说明。
- 更新表1和图1，增加nfd::face::LinkService, ndn::AppLinkService和ndn::NetDeviceLinkService类。

-修订整篇论文，将ndnSIM 2.0改为ndnSIM 2.0。

·修订版1(2015年1月27日):首次发布

参考文献

- [1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of ACM CoNEXT*, 2009.
- [2] L. Zhang et al., "Named data networking (NDN) project 2010 - 2011 progress summary," PARC, <http://www.named-data.net/ndn-ar2011.html>, Tech. Rep., November 2011.
- [3] L. Zhang et al., "Named data networking (NDN) project," PARC, Tech. Rep. NDN-0001, October 2010.
- [4] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *ACM SIGCOMM Computer Communication Review*, July 2014.
- [5] (2012, May) ns-3. [Online]. Available: <http://www.nsnam.org/>
- [6] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN, Technical Report NDN-0005, October 2012. [Online]. Available: <http://named-data.net/techreports.html>
- [7] A. Afanasyev, J. Shi, B. Zhang, L. Zhang, I. Moiseenko, Y. Yu, W. Shang, Y. Huang, J. P. Abraham, S. DiBenedetto, C. Fan, C. Papadopoulos, D. Pesavento, G. Grassi, G. Pau, H. Zhang, T. Song, H. Yuan, H. B. Abraham, P. Crowley, S. O. Amin, V. Lehman, , and L. Wang, "NFD developers guide," NDN Project, Tech. Rep. NDN-0021, July 2014.
- [8] NDN Project, "NFD - named data networking for-warding daemon," Online: <http://named-data.net/doc/NFD/0.2.0/>, 2014.
- [9] —, "NDN Packet Format Specification," Online: <http://named-data.net/doc/ndn-tlv/>, 2014.
- [10] D. Rossi, G. Rossini, "Caching performance of content centric networks under multi-path routing (and more)," Telecom ParisTech, Tech. Rep., 2011.
- [11] G. Rossini and D. Rossi, "ccnSim: an highly scalable CCN simulator," in *IEEE ICC*, 2013.
- [12] C. Scherb, M. Sifalakis, and C. Tschudin, "CCN-lite," Available: <http://www.ccn-lite.net>, 2013.
- [13] L. Muscariello. (2011) Content centric net-working packet level simulator. Orange Labs. [Online]. Available: <http://perso.rd.francetelecom.fr/muscariello/sim.html>
- [14] A. Carzaniga, M.J. Rutherford, and A.L. Wolf, Ed., *A Routing Scheme for Content-Based Networking*. IEEE INFOCOM, March 2004.
- [15] L. Wang, A. Hoque, C. Yi, A. Alyyan, and B. Zhang, "OSPFN: An OSPF based routing protocol for Named Data Networking," NDN, Tech. Rep NDN-0003, 2012.
- [16] NDN Project, "NLSR - Named Data Link State Routing Protocol," Online: <http://named-data.net/doc/NLSR/0.1.0/>, 2014.
- [17] C. Cabral, C. E. Rothenberg, and M. F. Magalhães, "Re-producing real NDN experiments using mini-CCNx," in *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, 2013.
- [18] (2015, January) GENI (Global Environment for Network Innovations). [Online]. Available: <http://www.geni.net>
- [19] (2015, January) Open Networking Lab. [Online]. Available: <http://onlab.us>
- [20] (2015, January) Emulab - Network Emulation Testbed. [Online]. Available: <http://www.emulab.net>