计算机网络大作业报告

项目名称：基于 ns3 的无线交付模拟

实验人员：TIN TUN AUNG

报告作者：TIN TUN AUNG

完成时间：2021/12/28

1．项目目的

1）学习 NS3 的基本操作，设计场景进行模拟，并提取数据

2）学习无线网络标准，及组网方式

2．项目内容

1）在 Linux 平台上安装 NS 比较新的版本，能够运行 NS 自带的测试脚本

2）选取无线网络中一种（如：Bluetooth、WLAN），设计场景，运行一定的 TCP 或 UDP 业务源，用 Nam 进行演示。

3）在 2 的基础上，分析 trace 文件，测定网络的性能（如：吞吐量、时延、丢包率等，任选择一个），用图标的方式（可以用 Linux 下的 Xgraph 或 Gnuplot，或者 Matlab）说明测定结果

3.实验环境

使用 VirtualBox 的 Ubuntu20.04 虚拟机，Ns3 的版本是 3.30 。

3．实验步骤及测试结果

1）正确编译并运行 Ns3

在网上下载好 ns3.30，然后按照官网说明文件进行编译。

运行 test.py 脚本，得到结果如图，没有发生错误。

```
PASS: TestSuite routing-aodv-regression
PASS: TestSuite routing-aodv-loopback
258 of 261 tests passed (258 passed, 3 skipped, 0 failed, 0 crashed, 0 valgrind errors)
List of SKIPped tests:
    ns3-tcp-cwnd (requires NSC)
    ns3-tcp-interoperability (requires NSC)
    nsc-tcp-loss (requires NSC)

***  Note: ns-3 examples are currently disabled. Enable them by adding
***  "--enable-examples" to ./waf configure or modifying your .ns3rc file.
```

图 1 test.py 运行结果

在“ns3.30/example/tutorials”目录下有一些教程脚本，我们可以

运行 first.py 作为样例，测试程序安装正确性。运行截图如下

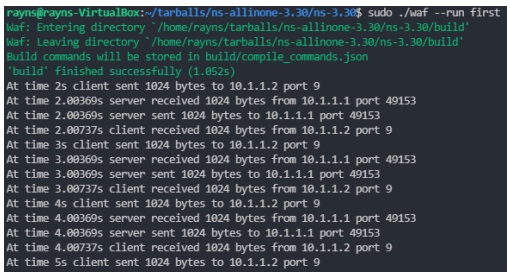

图 2 first.cc 运行结果

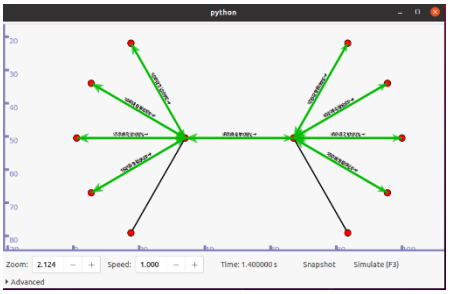

图 3 dumbbell.cc 带有--vis 参数运行结果

，同时我们也可以带有“—vis”参数，用 python 自动生成图像。

2）设计无线网络的场景，并使用 Nam 进行演示

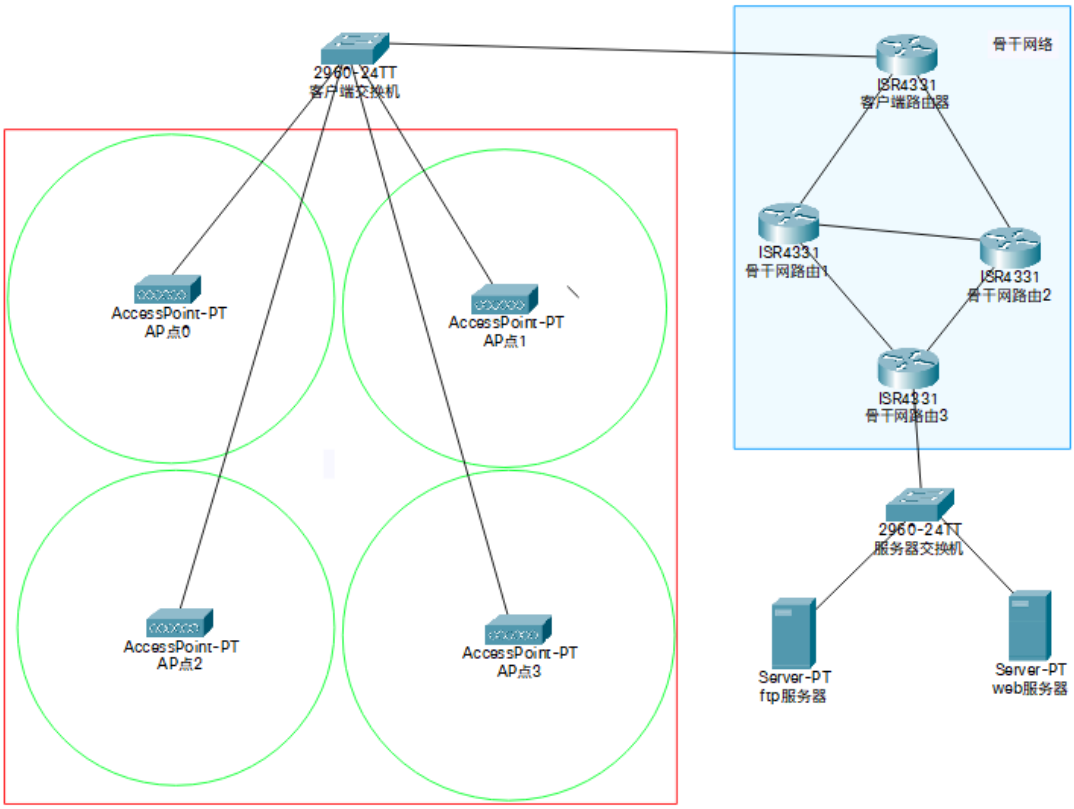

图 4 无线交付场景图

设计场景如图，是校园网的简易实现版本，从左往右介绍。左边的红框，是生成的移动终端的活动范围，生成后会在红框的范围内随机移动。绿色的圆框是 AP 点的覆盖范围，可以看到，在红框中，有些位置是没有无线网络覆盖的，在这些地方移动终端无法接收到网络，还有一个值得注意的地方是，无线信号使用的是同一信道，因此故意让

各个 AP 点的信号之间没有重叠。

所有 AP 点的流量经过交换机去到路由器，到达骨干网络。在骨干网络里面有 4 个路由器，使用 RIP 协议来自动生成路由表。服务器分两种类型一个是 Ftp 服务器，负责大流量的传输，在模拟中会一直发送尽可能多的数据来占满带宽。另一个则是 http 服务器，模拟用户访问网页的行为（间歇性产生流量）。

在整个网络的连接中，使用了不同的速率，在骨干路由以及路由器和交换机之间（客户端及服务端）都是 1000Mbps 传输速率,2ms 时延的 CSMA 信道，而服务器和交换机之间是使用 400Mbps，5ms 的 CSMA 信道，而客户端交换机和 AP 点之间是 100Mbps，10ms 时延的 CSMA 信道，因此在 FTP 传输数据时，由于其会尝试占满带宽，当数据到达交换机和 AP 点之间时会出现需要进行速率匹配的问题。

我们可以通过 NetAnim 来观看整个模拟过程，其中节点 0-3 是骨干路由器，4 是服务器交换机，5 是客户端交换机，6 是 Ftp 服务器，7 是 Web 服务器，8-11 是 AP 点，而序号 12 之后的节点都是移动终端。

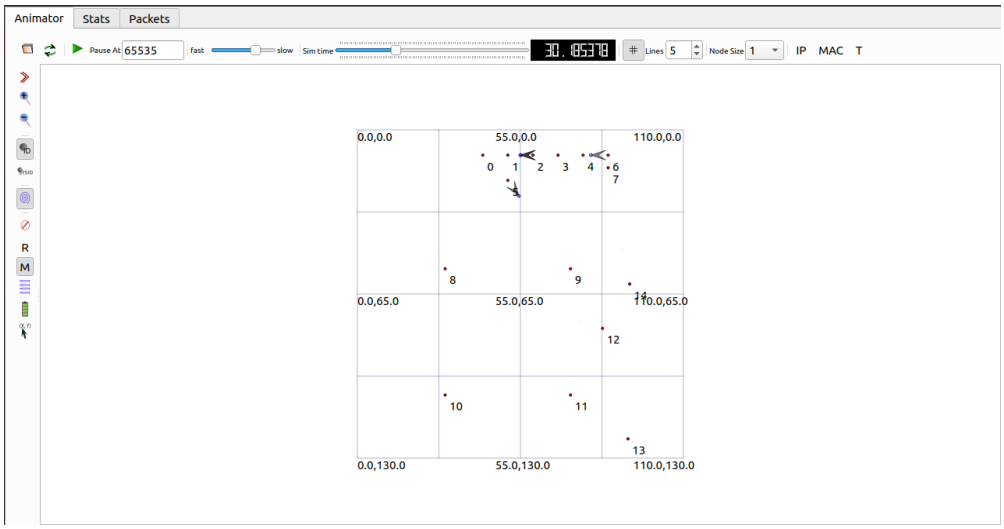

图 5 Ftp 流量可视化 Net Anim

3）借助 Trace 文件分析 Ftp 服务器吞吐量

我们可以通过 NS3 自带的 API 得到 Ftp 服务器 CSMA 信道的 PCAP 或 tr 文件，用来分析其吞吐量。使用 Linux 下的 Gawk 来处理数据，由于每个包的大小都是 524Bytes，可以通过返回的 ACK 包的 ACK 序列来计算累计发送成功的包：

设当前累计发送包数量为 AckCnt,第一个数据包发送的时间为 start

$$AckCnt = \left(Ack序列号 - 1\right) \Big/ 524Bytes$$

$$吞吐量 = (AckCnt \times 524 \times 8 \div 1000) / \left(ACK到达时间 - start\right)$$

使用 gawk 处理之后，再使用 Gnuplot 画图，得到结果如下所示，

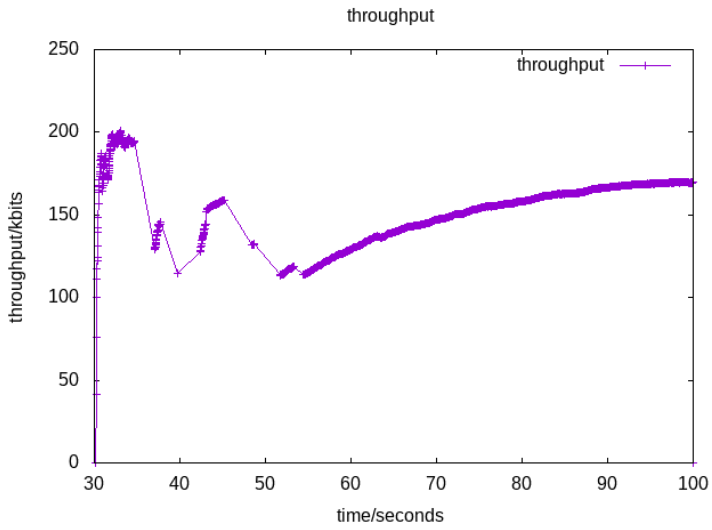

图 6 Ftp 服务器吞吐量

注意这里 Y 轴的单位是 kbit，X 轴是秒。可以看到一开始 Ftp 发送了大量的数据，然后在 35 秒左右开始下降，然后多次变化，最终在 55 秒后趋于稳定，在 160kbps 左右。

结合移动终端和客户端路由器和交换机的 PCAP 文件，可以得出一个结论，从客户端路由器到交换机的这一段线路中，所有来自于 Ftp 的

流量都被转发了，然而在 AP 点向移动终端发送的信息中，却缺少了很多信息，而移动端对于得到的数据包都返回了 ACK，因此可以断定是客户端交换机到 AP 点的这一段出现了链路拥堵。一些 Ftp 流量被丢弃，因此我们可以看到重复的 Ack 以及快重传机制的出现。

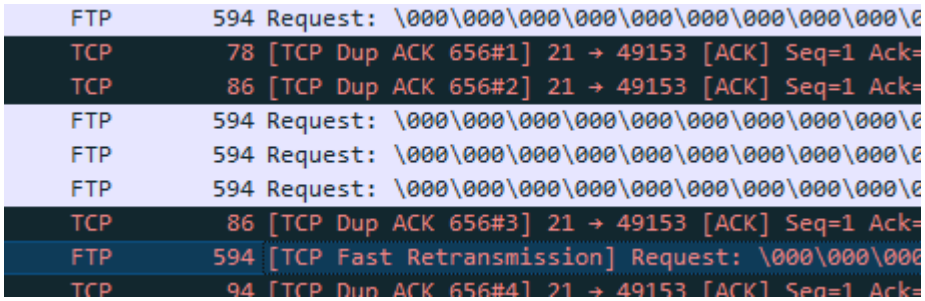

图 7 三次重复 Ack 后的快重传

由于是链路拥堵导致的问题，客户端并没有减小接收窗口的大小，因此是拥塞窗口在起到流量控制的作用。

我们就来详细分析一下在该场景下的链路层流量控制技术，及它影响到的传输层相关拥塞控制技术。由图 8 可以看到链路层维护了

```
//链路层流量控制
std::string qdiscTypeId = "ns3::FifoQueueDisc";
TrafficControlHelper tch;
tch.SetRootQueueDisc(qdiscTypeId);
QueueDiscContainer qd;
//主要是在交换机和Ap之间，有信道速率的不匹配，需要进行控制
for (int i = 2; i < 5; i++)
{
    //tch.Uninstall(clientSwitchNode.Get(0)->GetDevice(i));
    qd.Add(tch.Install(clientSwitchNode.Get(0)->GetDevice(i)));
}
tch.SetQueueLimits("ns3::DynamicQueueLimits");
```

图 8 链路控制相关代码

先入先出的队列，且是动态调整队列大小的，仔细想想好像相当于没有链路控制算法，实际上这个流量控制只是控制从客户端交换机这个设备发出去的速率，有一个先入先出的限制，除此之外似乎就没有了（这是我在设计场景时没有想到的，后面重读了书本相关内容才理

解）。

```
//Tcp设置，依次为重传机制，使用的tcp版本（此处为new reno），发送缓冲区，接收缓冲区
Config::SetDefault("ns3::TcpL4Protocol::RecoveryType", TypeIdValue(TypeId::LookupByName(recovery)));
Config::SetDefault("ns3::TcpL4Protocol::SocketType", TypeIdValue(TypeId::LookupByName(tcpTypeId)));
Config::SetDefault("ns3::TcpSocket::SndBufSize", UintegerValue(1 << 20));
Config::SetDefault("ns3::TcpSocket::RcvBufSize", UintegerValue(1 << 20));
// 设置默认的初始拥塞窗口大小为10
Config::SetDefault("ns3::TcpSocket::InitialCwnd", UintegerValue(10));
//设置默认延迟ack数量为1
Config::SetDefault("ns3::TcpSocket::DelAckCount", UintegerValue(delAckCount));
//设置默认tcp段大小为524Bytes（设置较小的tcp段，以防止IP分片）
Config::SetDefault("ns3::TcpSocket::SegmentSize", UintegerValue(segmentSize));
//开启选择性重传
Config::SetDefault("ns3::TcpSocketBase::Sack", BooleanValue(isSack));
```

图 9 传输层流量控制代码

那整个网络的组建及吞吐量变化过程就很明显了：一开始在骨干网中相邻的两个节点通过 RIP 协议交换路由信息，产生路由表，然后 FTP 服务器在与客户端建立 TCP 连接后，开始发送大量 FTP 数据，由于发送速率大于客户端路由器存储转发的速率，有大量的包丢失，以及可能产生了失序（前面的数据包丢失了，后面的数据包成功传输），开始出现大量的重复 ACK 帧，前三次 ACK 帧触发了 FTP 服务器的快重传，一直到 FTP 服务器对一些包触发了超时机制，于是重新从断点处重发，调整拥塞窗口，在经过一段时间的调整后，最终稳定下来。

4．实验遇到的问题

1）为什么不让 AP 点的信号范围大到足够覆盖整个红框？

值得注意的一点是，这些 AP 点使用的都是同一信道，这意味着，如果有一个 Sta 在两个 AP 的信号重叠区域，而两个 AP 正好都是同时发送数据包，那么 Sta 接收到的数据是碰撞后的数据，是错误的，会被丢弃。这导致的一个严重后果就是假设客户端路由器尝试发送 ARP 广播来得到一个 Sta 的 MAC 地址，由于两个 AP 同时发送信息，会导致碰撞，则目标 Sta 无法正确接收信号。

对于上面的问题，有两个解决方案，一个，是给交换机到 AP 点的信道设置不同的时延，这样的话，两个 AP 就不会同时发送从上面来的 ARP 数据包,且由于 NS3 实现了只与最近的 AP 点关联的功能[1]，因此 Sta 可以正常通信。但是，这样的话，在重叠信号区的 Sta 还是有可能受到信号冲突影响，最终影响吞吐量。

另一个解决方案就是不同的 AP 使用不同的信道，对此方法也有两种实现方式，一种是 Sta 只安装一个 NetDevice，让该 NetDevice 能够适应不同的信道。在这个模拟中，我们使用的是 802.11a 协议，默认使用的是 36 号信道，信道宽度为 20MHz，如果两个 AP 分别使用 36，40 号信道，则 Sta 应该设置为 38 号信道，信道宽度为 40MHz，信道宽度要覆盖 AP 的信道。但是我在实际测试时，按如上进行配置，却无法让 AP 和 Sta 进行通信，在 NS3 论坛上找了许久，也没有解决方案，看到其他人大多是使用 LTE 来做交付的模拟。

另一种实现方式，是给 Sta 安装多个 NetDevice，每一个 NetDevice 对应不同的信道，这样的话，任何 AP 都能给 Sta 发信息，但是，当 Sta 要发消息的时候，却会因为无法自动选择所使用的 Interface，而导致无法正确发送数据出去。

因此权衡之下，只能使用同一信道，并且让其不重叠以防止冲突。在无线信号覆盖范围外的 Sta 就无法接收/发送数据。

2）为什么不在客户端路由器使用 DHCP 服务器？

其实一开始是使用的，但由于 DHCP 服务器的 IP 地址是动态分配的，在测试的时候，不方便获得客户端的 IP 地址，因此就把他关闭

了。另一个原因就是，测试的时候，出现了无线信号重叠区域无法通信的问题（即问题 1），误以为是 DHCP 的问题，现在想想，应该是可行的。

3）如何模拟交换机？

在 NS3 没有 SwitchHelper，因此我们使用 Bridge 即桥接的方式来模拟交换机，将节点上的所有 NetDevice 桥接成一个设备，这样所有端口就在同一个广播域了。在解决问题 1 的过程中，尝试将 Sta 的多个 NetDevice 桥接成一个设备，但是失败了，似乎 NS3 不支持将 Sta 设备进行桥接。

5．实验思考

1）业界主流的无边界无线交付（漫游）是什么原理，如何避免冲突？

在翻阅了一些论文[2]，结合 NS3 的 staWifiMAc 类的实现源码，可以了解到，交换的场景如下，一个 AP 点对应一个 BSS，移动终端（MS）在这些 BSS 之间移动，就会出现需要重新连接的情况。

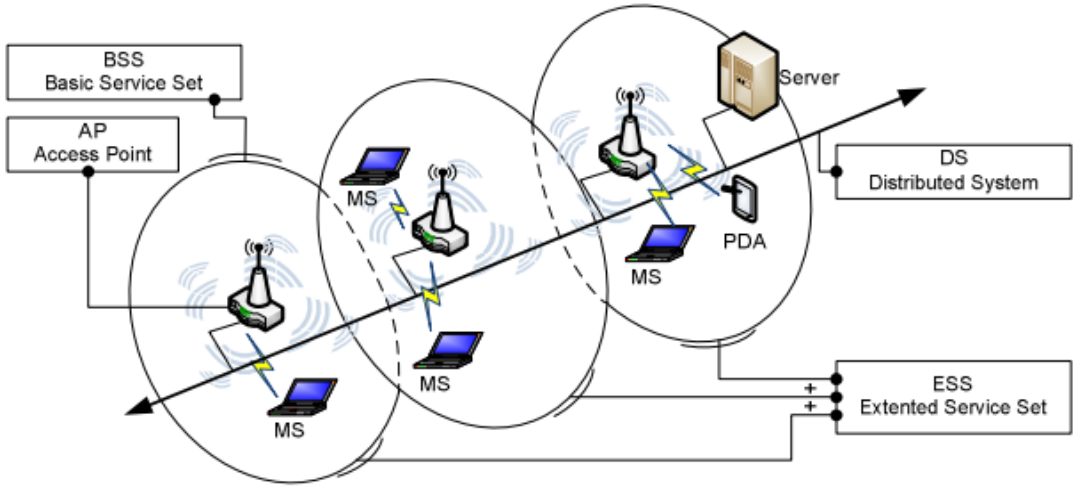

图 10 IEEE802.11 基础设施模式

整个移交过程分为三个阶段，检测，扫描，正式移交，当 MS 和当前 AP 的信号强度低于设定的阈值时，就会触发重新扫描的过程。当然，

引用的论文所描述的实现方式侧重点在“快”，而多信道的支持是显而易见的。

当前有几个新的 wifi 标准用来更好支持漫游功能，如 802.11k、802.11r，旨在提高移交的速度，减小延迟。

2）日常生活中一些无线接入点，是使用什么协议来通信？

手机热点: 802.11ac(wifi 5)

家用路由器: 802.11n(wifi 4)2.4GHz

802.11ac (wifi 5) 5GHz

商场公共 wifi: 802.11ac (wifi 5) 5GHz

运营商公共 wifi： 802.11ac (wifi 5) 5GHz

一些地方的公共 wifi 也实现了最新的 802.11ax （wifi 6）标准，可以同时工作在 2.4GHz 和 5GHz。wifi 的最长可用范围在直径 100 米以内，因此，可想而知，在商场等公共场所的 wifi 都是布置在室内的，也意味着商场里能接收到运营商的公共 wifi 一定是因为两者之间有合作才可能。

6．实验总结

关于无线网络组网的探究，就到这里告一段落了，在搜索关于 802.11wifi 标准的内容时，发现大多论文都是近十几年才开始增多的，而 NS3 作为网络模拟器，有大量的研究人员使用它来测试所设计的网络的性能。虽然到最后，我还是没能让 Sta 在两个 AP 点之间成功通信，但是能够认识到 NS3，能够学习到有关于无线网络组网，这些比较靠近行业前沿的知识，还是特别让人兴奋的。我们的生活已经

越来越离不开无线网络，能够研究与自己的生活切身相关的东西，怎么能让人不激动呢，努力学习，还有许多自己不懂的知识，有时间定要再继续探索无线网络组网的应用。

引用：

[1] https://www.nsnam.org/bugzilla/show_bug.cgi?id=2399#a0

[2] https://arxiv.org/ftp/arxiv/papers/1102/1102.5189.pdf