

计算机网络大作业个人报告

15331371 姚志立

一、分工情况

本人负责的部分是 HNS 和 Transport 模块的设计与实现。

二、项目过程

在实现 Transport 模块的过程中，由于发送时需要知道物理地址，因此一开始的构想是在 Transport 模块初始化时传入各个主机的标识和主机物理地址的映射表。但这样就无法实现动态增加主机，而且过程及其繁琐，不符合实际。实际情况中，主机应该是可以动态添加，关于主机的标识和物理地址等信息不应该在某些主机还没有出现以前就获取到。为了解决这个问题，刚好在上计网课时讲到 DNS 的时候想到了我们可以 HNS，作为一个独立的服务器，管理所有主机标识和物理地址的映射。于是我们设计并实现了 HNS，任何主机都可以向唯一的 HNS 注册自己的信息，HNS 管理所有主机标识到物理地址的映射信息，并在需要的时候将该映射表分发出去。

关于 Transport 模块的实现，最开始要设计一种新的应用层的报文协议。在这个项目中，我使用了类似于网络层和链路层的报文协议结构，使得发送到下一跳路由所需信息、到达目的地所需信息、分发给不同模块所需信息分离开。

后续的实现中，发现 LS 算法需要广播的机制，考虑到我们的实验环境算是个小型网络，因此使用了书中的第一种解决广播风暴的同时又能成功广播的机制，那就是在广播报文里加入已到达的主机，并不重复向已到达的主机广播。尽管还是存在有主机收到相同报文的情况，而且随着广播的不断进行，记录已广播主机的数据结构将会增大，但对于小型网络来说，足以解决问题。一开始实现广播后，发现初次广播和二次广播有不同的地方。初次广播只需要使用自身标识作为报文的源主机字段，而且一开始的已到达列表只有自己，然后向所有邻居发送该报文即可。但是对于二次广播，必须保留源主机字段和已到达的列表。利用 python 的性质，对于同一个接口函数，可以传入不同的参数，然后在函数里面再作判断属于初次广播还是二次广播，以解决这个问题。

而对于来自其他主机且目的地不是自身的报文，则需要转发，一开始想着直接使用提供给其他模块的发送数据的接口，但是逻辑上和实现上这样做都不好，因此在该项目的后期将发送和转发分离开，使其更加合理。

最后，由于一开始邻居模块要与其他主机建立邻居关系需要发送报文，但此时由于路由模块没有将成为邻居的主机的信息，使用普通的发送逻辑无法获取下一跳主机的信息，因此

在发送的接口增加一个非安全的发送模式，越过查路由表的过程，直接查从主机标识到物理地址的映射表将数据发送出去，以此来解决一开始无法发送的问题。

三、个人总结

这次的大作业再一次让我学习到模块化的思想，如何设计组织各个模块的关系才能使得各个模块能分离开且能正常工作，从队友们身上能学习到很多关于这方面的思想，感谢他们的 carry。

第二，对于网络编程有了更深的理解。从这次的大作业中也能小窥到计算机网络中各层的工作原理大致是怎样的，理解到一个数据报如何通过各层的传递，不断增加相应的首部，才能使得简化发送接收逻辑和成功发送接收数据之间有了很好的折中。

其次，关于如何简化设计逻辑也有了更深的学习。不仅是在队友们设计不同模块时能学习到如何降低模块间的耦合程度，也在 hns 的设计、转发和发送的分离等地方有了一点思考。因为对于这样的一个大的模块分成许多小的模块，小的模块里又有不同的功能要实现的工程，以前没有过这样的经历，因此设计和实现的途中也出现了很多错误，感觉在如何提高程序的稳健性方面还是需要更多地去学习的，很多情况在最初设计时就应该要想到却没有想到。

最后就是关于书本上的知识了，书本上的知识更多的对算法设计这部分能起到很大的作用，对于数据运输层的实现没怎么显式的提到，真正自己实现起来却发现有很多细节的问题需要考虑。在与队友们讨论 DV 算法的 BUG 的时候，再一次体会到毒性逆转并不能解决 3 个以上节点的环路问题。再加上我们的权值是任意的，不像 RIP 算法那样，毒性逆转或者水平分割加上 16 跳阈值就能解决这个问题，但这样的讨论就让我更加对 DV 算法和 RIP 路由协议有了更深的认识。