

# 基于强化学习的TCP拥塞控制

钟嘉伦 孙昊海

2019.1.23

## 1 问题概述

近来，计算机网络方向的研究逐渐增多，大多都是采用传统的网络仿真方法进行研究与分析。随着人工智能的兴起，深度学习与强化学习逐渐在计算机网络仿真中扮演重要的角色。考虑到不同网络模拟器的输出格式不一样导致难以确定一个统一的评判标准，本次实验采用NS3作为仿真模拟的软件，OpenAI作为仿真算法库，来研究强化学习能在计算机网络领域（特别是对TCP拥塞控制）产生什么影响。

## 2 研究动机

工欲善其事，必先利其器。在确定了选题之后，我们小组首先对本次课设的工具进行了分析，首先是不同网络仿真模拟器之间的区别。在研究中比较流行的几类仿真模拟器之中，NS3的编程语言支持Python，这对深度学习乃至强化学习的研究比较友好。

表 1: 不同模拟器之间的比较

	可获得模型	网络层次模拟	仿真模型	编程语言	GUI 支持	是否开源
OPNET	TCP/IP,ATM,Ethenet	支持	FSM	C++	是	否
OMNeT++	TCP/IP, SCSI ,FDDI	支持	FSM/Thread	C++	是	是
NS2	围绕 TCP/IP	不支持	FSM	C++/Tcl	否	是
NS3			FSM	C++/Python		是

同时为了快速上手强化学习在计算机网络仿真之中的应用，我们小组将以自适应视频流传输作为本次课设的切入点，所以我们小组的课程设计将会由这两个方面构成。

## 3 设计内容

### 3.1 概要设计

系统环境：

1. Ubuntu 16.04
2. python 3.6
3. tensorflow 1.12.0

### 3.2 具体设计

tcp\_newreno.py 是 newreno 算法的 Python 移植，而 tcp\_base.py 表示着 Deep Q-learning 算法模型，我们小组先进行算法的转换，将 C++ 转化为 Python，与论文作对比，然后在 TcpNewReno.py 中实现强化学习的模型。首先分析数据接收的对比图：

Tcp version	bottleneck_bandwidth	bottleneck_delay	access_bandwidth	access_delay	nLeaf	mtu_bytes	duration	RxPkts
TcpNewReno	2Mbps	0.01ms	10Mbps	20ms	1	400	10	5367
TcpHybla	2Mbps	0.01ms	10Mbps	20ms	1	400	10	5367
TcpHighSpeed	2Mbps	0.01ms	10Mbps	20ms	1	400	10	5327
TcpHtcp	2Mbps	0.01ms	10Mbps	20ms	1	400	10	5064
TcpVegas	2Mbps	0.01ms	10Mbps	20ms	1	400	10	1404
TcpScalable	2Mbps	0.01ms	10Mbps	20ms	1	400	10	3951
TcpVeno	2Mbps	0.01ms	10Mbps	20ms	1	400	10	5367
TcpBic	2Mbps	0.01ms	10Mbps	20ms	1	400	10	4671
TcpYeah	2Mbps	0.01ms	10Mbps	20ms	1	400	10	5805
TcpIllinois	2Mbps	0.01ms	10Mbps	20ms	1	400	10	5357
TcpWestwood	2Mbps	0.01ms	10Mbps	20ms	1	400	10	2688
TcpWestwoodPlus	2Mbps	0.01ms	10Mbps	20ms	1	400	10	4485
TcpLedbat	2Mbps	0.01ms	10Mbps	20ms	1	400	10	5367
TcpLp	2Mbps	0.01ms	10Mbps	20ms	1	400	10	267

图 1: 传统方法数据接收对比图

由于要选择一个数据包接收效果尽可能好的协议，同时仿真智能体应基于时间，我们选择了 TcpNewReno 作为参照的仿真协议。通过这个协议可以得到传统方法的 CWND 图。

对应着强化学习的方法，可以从 QTCP 论文之中获取吞吐量以及 RTT 的示意图。接下来第一步，我们小组尝试改进论文中的 DEMO，也就是自适应视频流传输的算法，尝试加快模型的收敛速度。第二步，尝试传统方法从 C++ 到 Python 的移植。第三步，采用 Deep Q-Learning 算法进行训练，算法模型如下图。第四步，改进 reward 和 action。

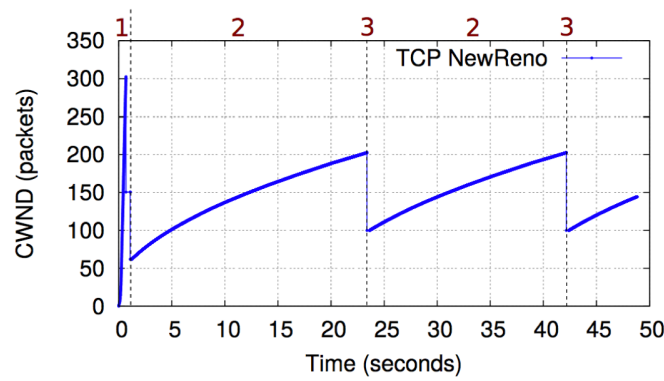


图 2: 传统方法 CWND 曲线图

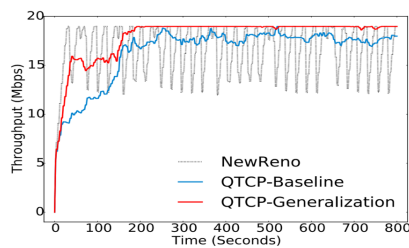


图 3: 吞吐量

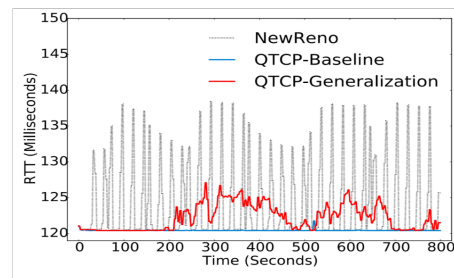


图 4: RTT

```
self.model = keras.Sequential()
self.model.add(keras.layers.Dense(obsSpace.shape[0], input_shape=(obsSpace.shape[0],), activation='relu'))
self.model.add(keras.layers.Dense(50, activation='relu'))
self.model.add(keras.layers.Dense(3, activation='softmax'))
self.model.compile(optimizer=tf.train.AdamOptimizer(0.001),
                    loss='categorical_crossentropy',
                    metrics=['accuracy'])
```

图 5: 强化学习模型

## 4 评测结果

### 4.1 程序清单

1. sim.cc: 传统方法仿真脚本
2. tcp\_base.py: Deep Q-learning 算法模型
3. tcp\_newreno.py: newreno 算法实现
4. tcp-rl-env.cc: 网络环境在强化学习当中的环境构建
5. tcp-rl.cc: 强化学习环境构建
6. test\_tcp.py: 强化学习运行脚本

## 4.2 实验结果

```
if utilites(next_obs[15], next_obs[11]) > utilites(obs[15], obs[11]):  
    reward = 10  
else:  
    reward = -20
```

图 6: 改进 reward

```
new_cWnd = cWnd  
if actioan_cwnd == 0:  
    new_cWnd += segmentSize  
elif actioan_cwnd == 1:  
    new_cWnd -= segmentSize
```

图 7: 改进 action

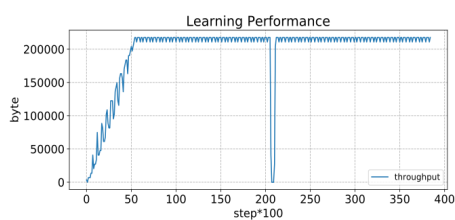


图 8: throughput

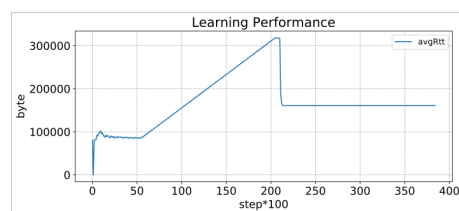


图 9: avgRTT

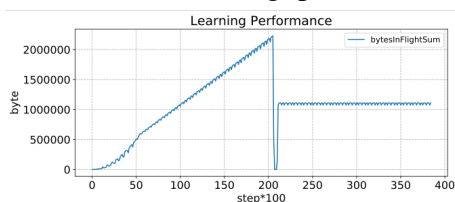


图 10: bytesInFlightSum

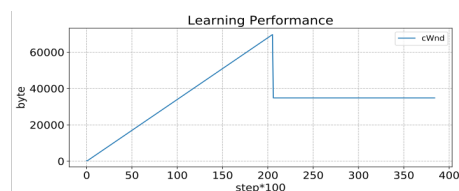


图 11: CWND

改进后，1 个 step 的平均运行速度为 0.007s，收敛速度也超过了论文水平。由此可见，强化学习方法在计算机网络领域作用是能起到作用的。

## 5 讨论

### 5.1 问题讨论

1. NS3 的使用不熟悉，导致不知道该如何搭建相应的环境，花了很长的时间查看别人的代码，最终解决问题。
2. 强化学习的内容不清楚，在开始设计相关算法之前查阅相关资料。
3. 设计出来的 action 不合适，导致需要很多 step 才能看出相应的结果，这里通过调整 action 的步长快速看出效果，快速调节相关参数。
4. 基于事件的仿真智能体始终不能正常运行，后面改成基于时间的仿真智能体。

### 5.2 老师评论

### 5.3 反思与建议

在课设的最开始，最好能先理顺整个文件目录的结构，以及整个项目的运行逻辑，比如使用传统方法还是强化学习方法，基于时间还是基于事件。这样能够比较有针对性的快速上手，从而节省时间。

## 6 相关工作

基于 OpenAI 有许多 NS3 仿真研究，主要包括：

1. TCP 拥塞控制、自适应视频流传输等网络应用
2. OpenAI Gym 扩展开发
3. 基于强化学习的网络问题解决方案

我们的工作是在网络应用这一部分中实现了模型的优化，加快了模型的收敛速度与准确率。

## 7 总结

### 7.1 完成工作

钟嘉伦：自适应视频流传输调研，模型移植调研，各种文档撰写。

孙昊海：RL 方法调研，强化学习模型搭建，强化学习模型调优。

### 7.2 个人小结

钟嘉伦：这次的计算机网络课设对我还是有很大帮助的。首先是了解了计算机网络仿真的相关知识，比如学会了 NS3 仿真的基本操作以及计算机网络应用的系统结构。其次是了

解了强化学习的基本过程，特别是强化学习在计算机网络领域的具体应用，从算法研究走向实际的工程之中。此外，课程答辩的准备以及文档方面的撰写也与以往不同，我写了两个版本的报告，由于对 **Latex** 不太熟练且图比较多，**Latex** 版直到最后一刻还在调图片的显示。所以十分感谢有这样一次机会能让我发现自己的不足，从而提升自己。

孙昊海：这次实验对我的启发非常大，之前一直是做的深度学习的内容，涉及到的领域都大多是自然语言处理以及图像处理等方面，这是我第一次接触强化学习的内容，强化学习在计算机网络上的应用也是非常的广。**ns3** 的使用之前也不太熟悉，有很多不懂的地方，花了非常多的时间去翻看别人的代码。**tcp** 拥塞控制是一个很棒的选题，强化学习在这个应用上会发挥非常大的作用，在后续过程当中，我想把现有的模型继续调优，达到一个比较不错的效果。

## A 附录

参考文献如下：[1, 2]。

## 参考文献

- [1] “ns3-gym extending openai gym for networking research.” [Online]. Available: <https://github.com/tkn-tub/ns3-gym>.
- [2] “Qtcp: Adaptive congestion control with reinforcement learning.” [Online]. Available: <http://www.ece.neu.edu/personal/meleis/tnse-18.pdf>