STMS: Improving MPTCP Throughput Under Heterogeneous Networks

Hang Shi and Yong Cui, *Tsinghua University*; Xin Wang, *Stony Brook University*;
Yuming Hu and Minglong Dai, *Tsinghua University*;
Fanzhao Wang and Kai Zheng, *Huawei Technologies*

汇报人：杨向杰 M201873143

background

- 随着网络用户和应用的快速增长，对网络带宽的要求越来越大。

解决策略：Multi-path TCP (MPTCP)，例如无线设备具有两个网络接口，local-area WiFi network 和wide-area cellular network。

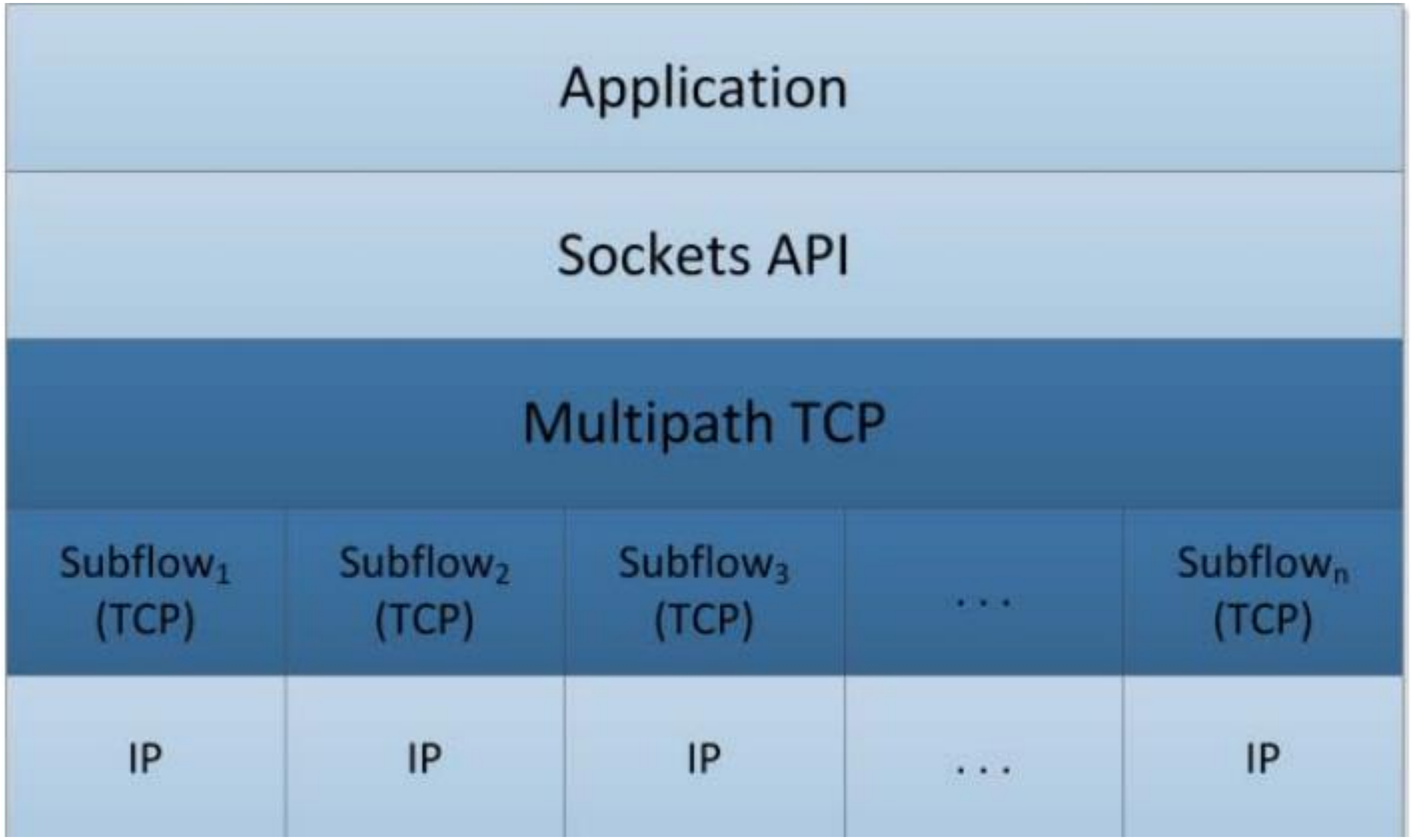
- 网络的类型不同，不同路径的网络传输质量也有很大不同。

例如， WiFi和cellular network的Round-Trip Time (RTT)差异很大。

MPTCP原理

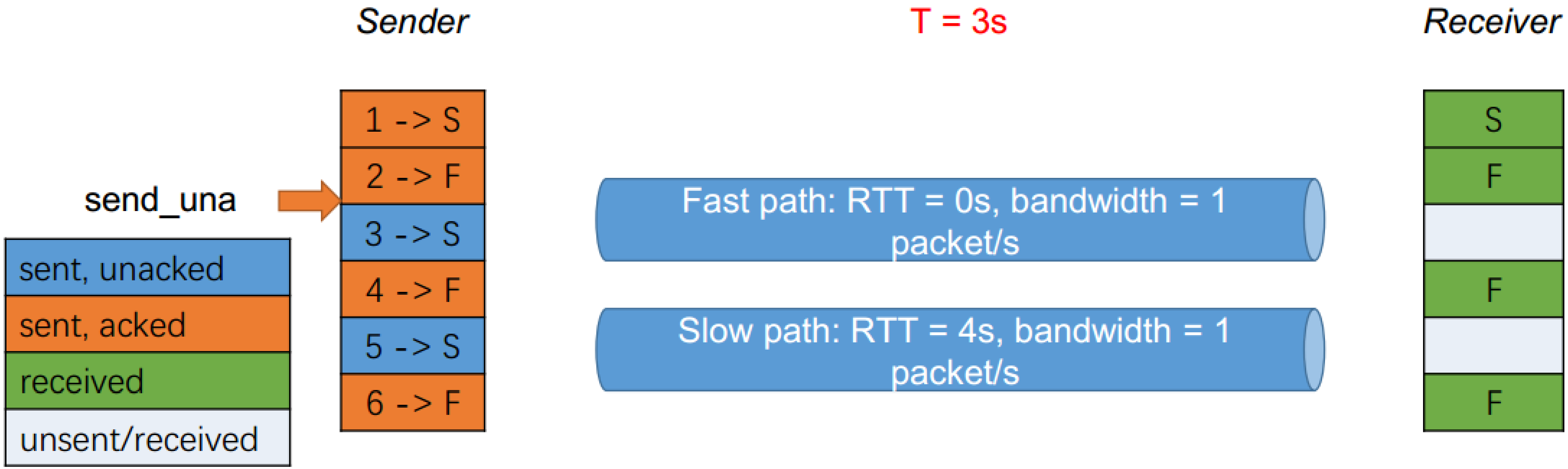
MPTCP默认的调度原理：通过最快的可用路径发送数据包（类似贪心策略）

MPTCP协议栈结构



MPTCP存在问题1

从Slow-Path传输的数据包到达晚，
导致需要更大的host-buffer

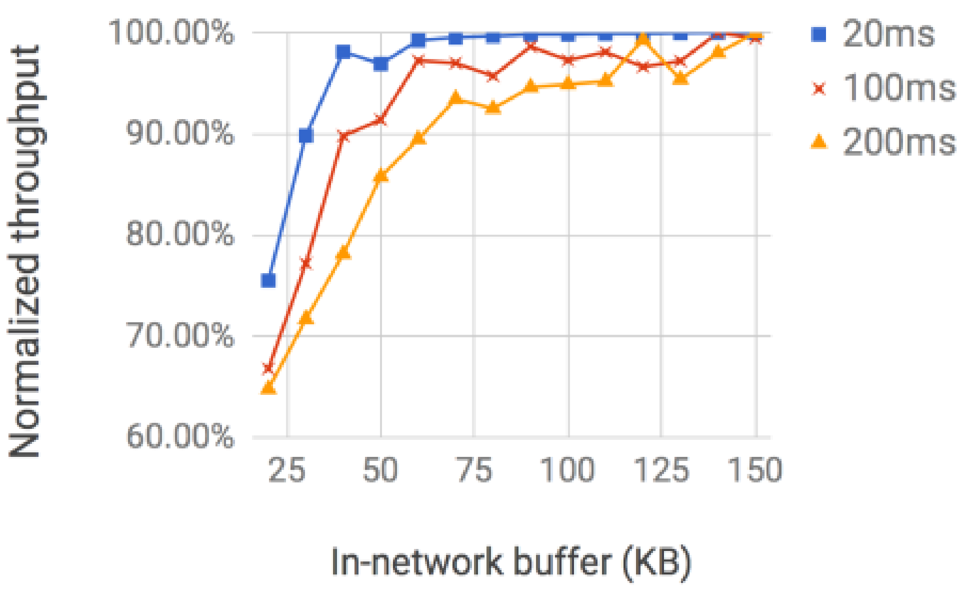


MPTCP存在问题2

在MPTCP中，当接收到有顺序的数据包时，每一个包都会有相应的ACK信号。但是当接收无序的数据包时，直到接收从slow-path传输的数据包之后，Data ACK会同时告知众多数据包从fast-path发送，这就会导致大量突发性的发送行为。

因此，如果network buffer容量不够大的话，就会导致丢包或者堵塞等行为。

MPTCP存在问题2



为了达到与single-TCP相同的吞吐量，双路的MPTCP的缓冲区必须从30KB增加到150KB。

in-network buffer/KB	observed loss rate	Fast path in MPTCP/Mbps	Single TCP/Mbps	Utilization
30	0.05%	12.1	28.4	42.76%
60	0.02%	20.8	28.4	73.50%
90	0.02%	25	28.4	88.34%
150	0.01%	26.5	28.4	93.64%

调度算法

核心思想

预分配数据包，为fast sub-flow缓冲先到达的数据包，并将具有较大序列号的数据包分配给slow-flow，使它们按顺序到达。

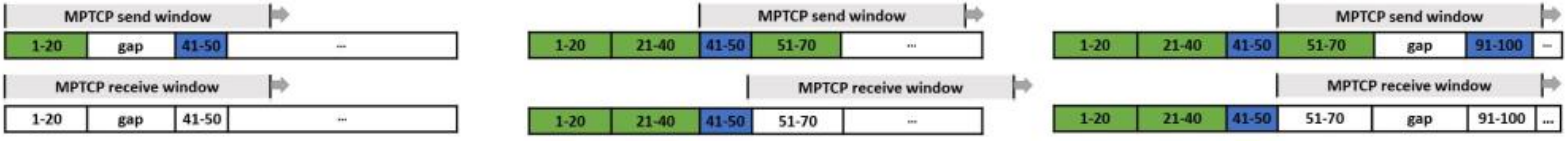
Algorithm 1 Slide Together Scheduler

procedure ST_SCHEDULE(*unsentPackets*) ▷ Scheduler runs when one of sub-flow is available
    if Fast sub-flow has space in send window **then**
        $Fast\ sub-flow \leftarrow$ unsentPackets[0]
    elseSlow sub-flow has space in send window
        $Slow\ sub-flow \leftarrow$ unsentPackets[Gap]
    end if
end procedure

关键点

经slow path到达的时间和经fast-path到达的时间中间会有一个间隙。怎样计算这个间隙是关键。

调度算法



(a) $t = 0$ (b) $t = 2RTT_f$ (c) $t = RTT_s$

蓝色为slow-path，绿色为fast-path

令RTTs = 3RTTf并假设上行链路延迟和下行链路延迟是对称的

计算Gap Value

Bf是fast-path的带宽，OWDs和OWDf是单路延迟，因此Gap可由由下式推导：

$$True\_Gap = B_f \times (OWD_s - OWD_f)$$

存在问题：

（1）one way dalay测量不可能那么精确

（2）当in-network buffer被限制时，路径的带宽测量也不精确

解决办法

在从slow-path传输的数据包到达之后，将发送Data ACK一次性确认无序的数据包，Data ACK的数量也就反映了乱序到达的程度，那么就可以根据Data ACK动态调整Gap Value

Algorithm 2 Gap Adjustment Algorithm

```
procedure GAP_ADJUST(data_acked)    ▷ This function gets called when receiving Data ACK
    if data_acked > 2 then
        send_una ← left edge of MPTCP-level send window
        if send_una was sent from slow path then
            delta_gap = data_acked
        elsesend_una was sent from fast path
            delta_gap = −data_acked
        end if
    end if
    gap+ = EWMA(delta_gap, adjust_interval)
end procedure
```

adjust_interval是可调参数，反映了算法对网络变化的敏捷度。

STMS对host buffer的大小要求

（1）从fast_path发送的未确认数据包：$Bf \ast RTTf$

（2）从slow_path发送的数据包（ACK信号从fast_path返回）：$Bs \ast (OWDs + OWDf)$

（3）fast_path需要缓存的数据包：$Bf \ast (OWDs – OWDf)$

合计：

$$Buf(STMS) = (Bf + Bs) \ast (OWDf + OWDs)$$

$$Buf(default) = (Bf + Bs) \ast RTTs$$

STMS在hostbuffer变化时的表现

(1) Host buffer < Buf(STMS)时，STMS倾向于利用fastpath，并且只有在slowpath不会导致堵塞时才会使用，STMS会使用slowpath时的buffer要求如下:

$$\begin{aligned} Buf(fallback) &= RTT_f \times B_f + Gap \\ &= B_f \times (OWD_s + OWD_f) \end{aligned}$$

(2) Host buffer < Buf(fallback)时，STMS退化为single TCP，只从fast path传输数据

因此，STMS可以在不同host buffer时选择最优方案。

STMS两种变形

不同在如何计算Gap value

STMS-C：

每次从slowpath发包时，STMS-C从sub-flow TCP的算法中得到RTT并估计带宽并计算gap值（假设上行和下行延迟是对称的）。

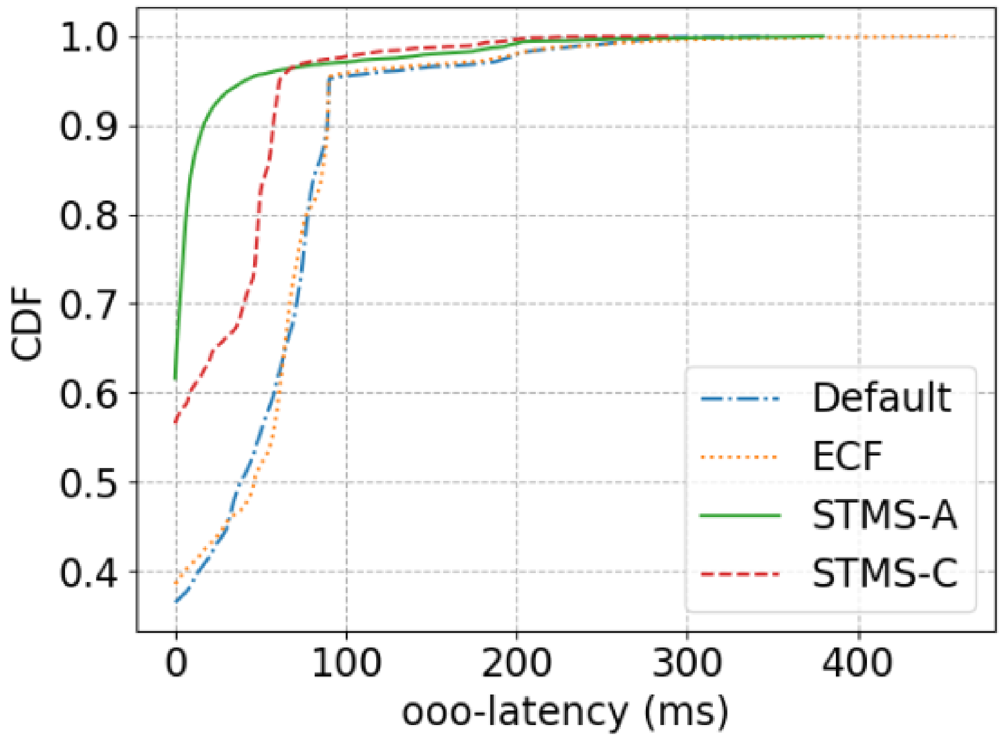
STMS-A：

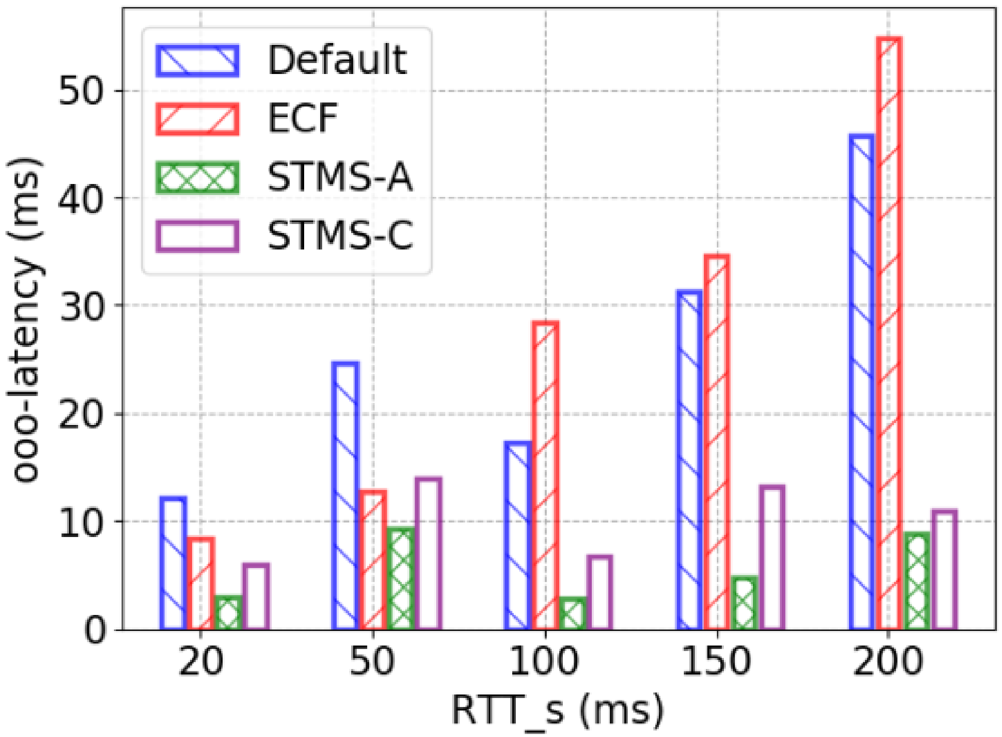
默认算法中，计算delta_gap值。

STMS实验评估

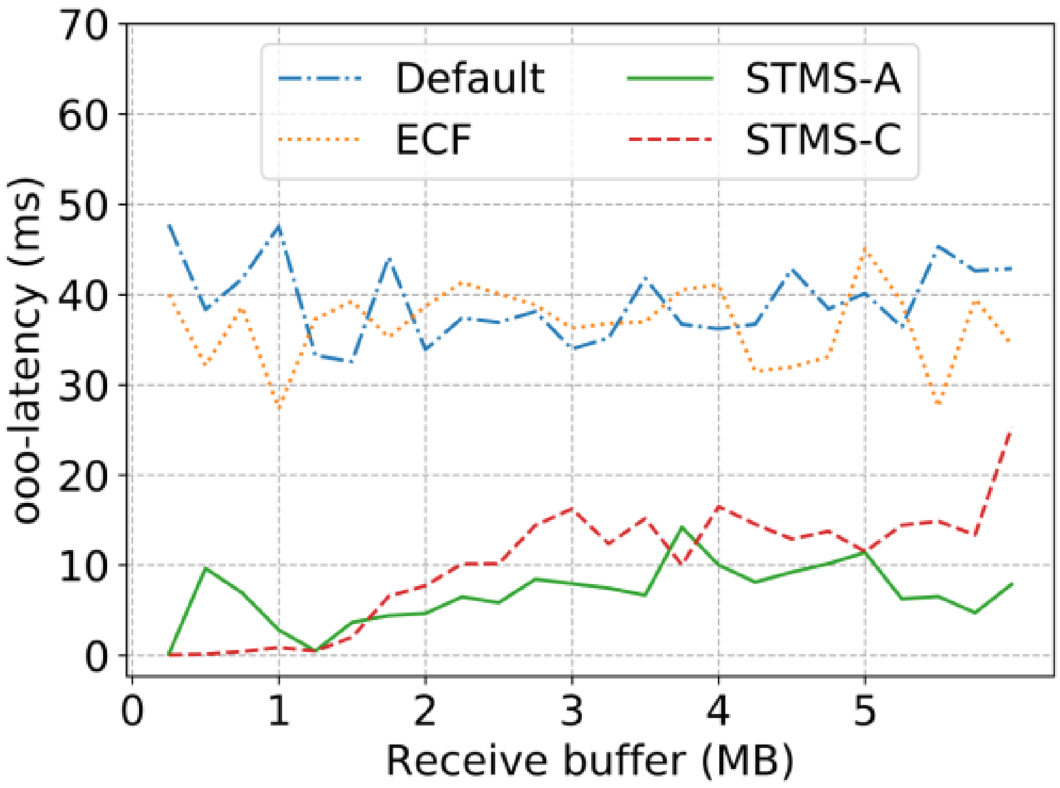
（1）implement STMS in the Linux kernel based on MPTCP version 0.92
（2）和default 、ECF进行对比（Early completion first.Sending tail packets out-of-orderly）
（3）在实验室和真实环境中
（4）不同的静态或者动态网络环境
（5）不同in-network buffer和host buffer

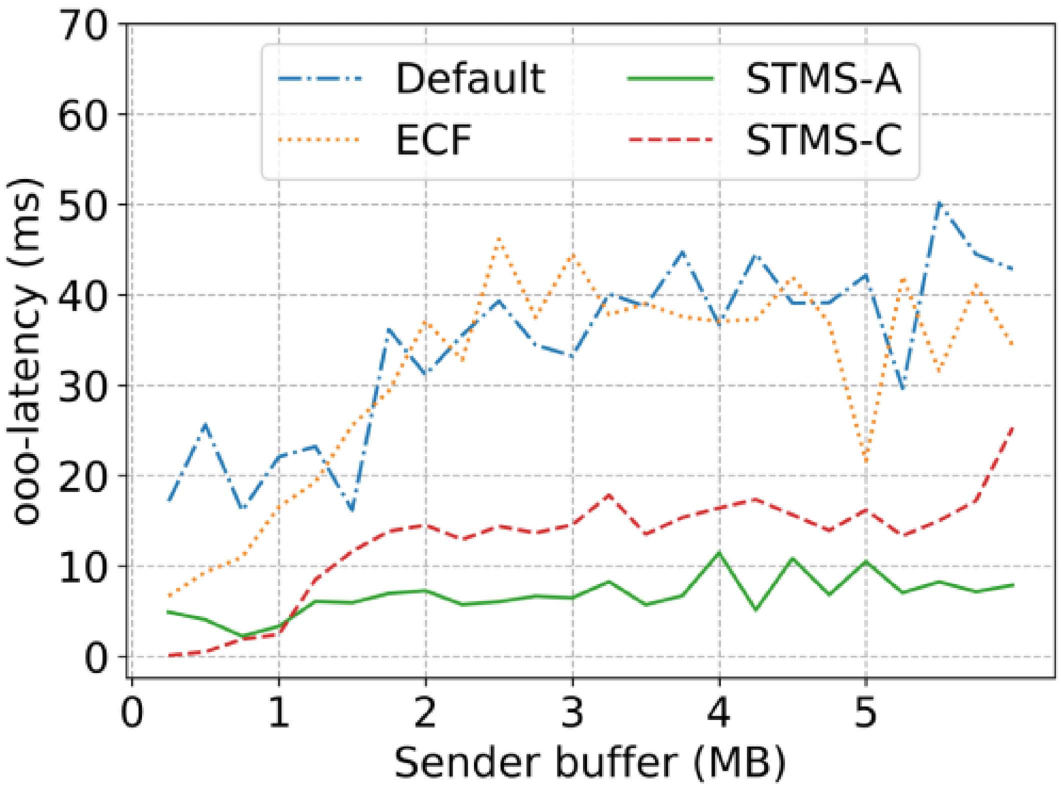
OOO(out of order delay)





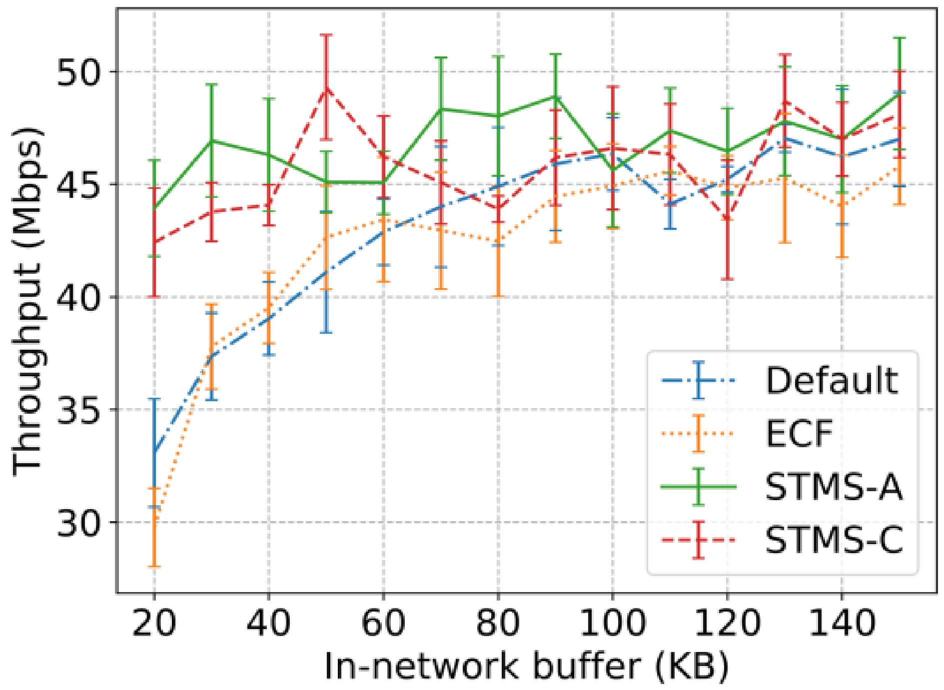
Sender/receive buffer变化时

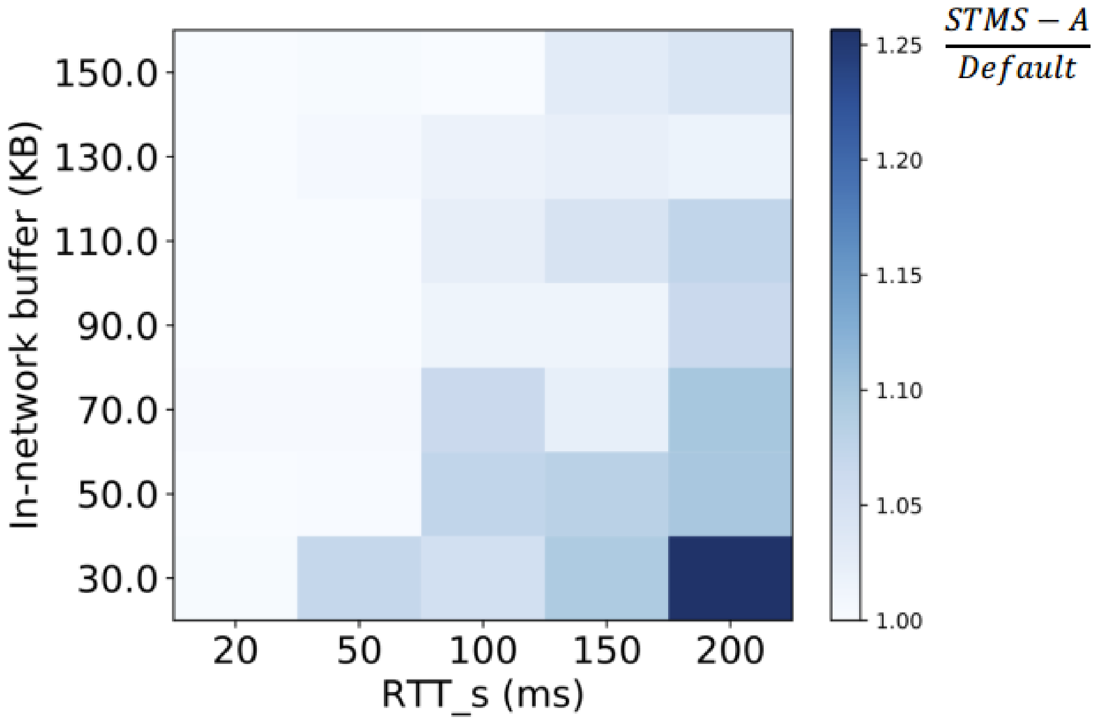




in-network变化时

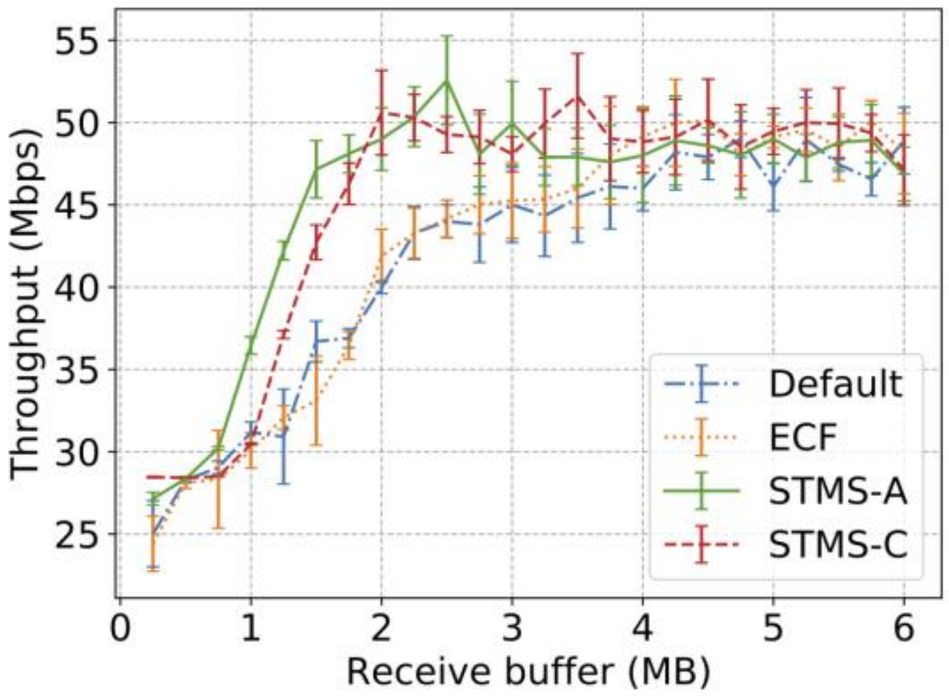
右图，当限制in-network buffer时，大约有25%提高

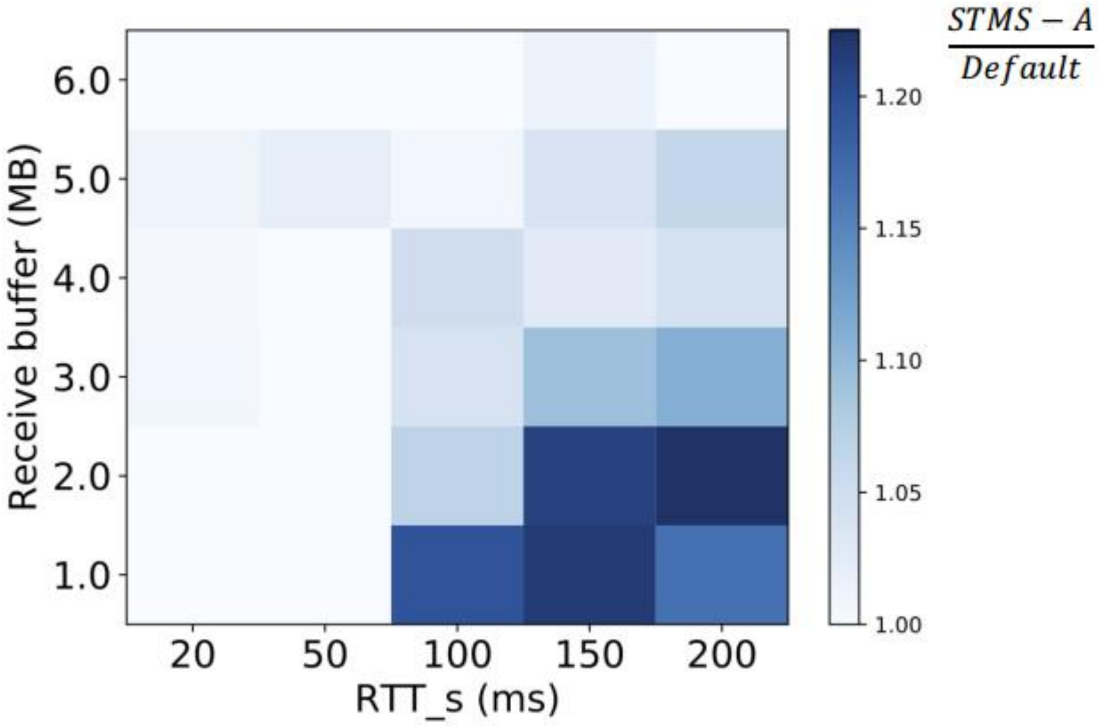




Host buffer变化时

右图，当限制receive/send buffer时，也大约提高了25%





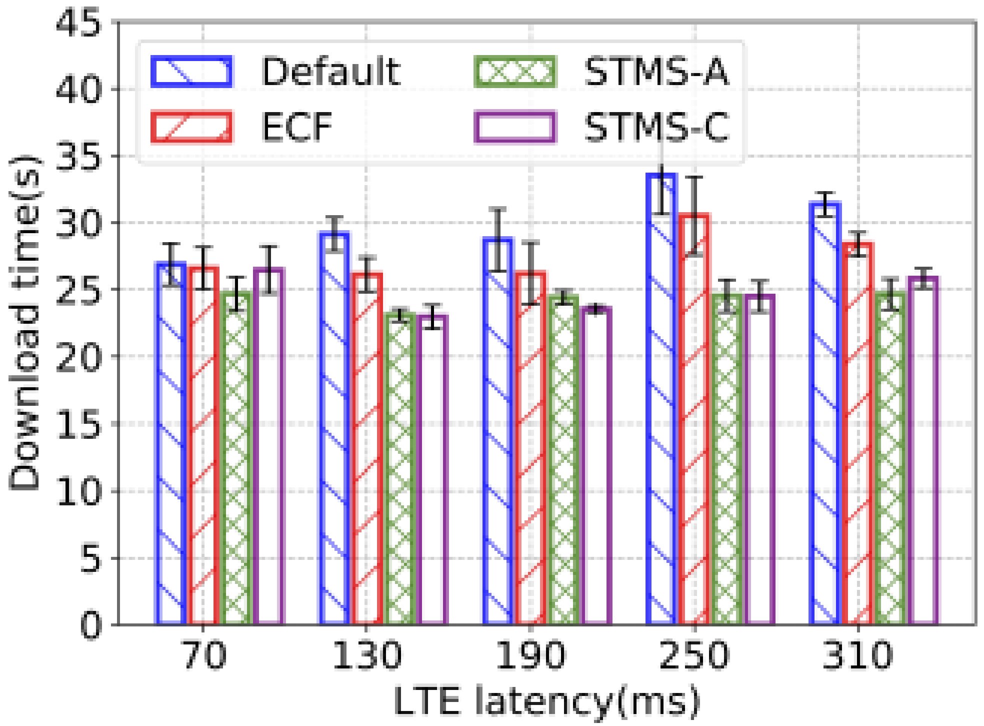
Real world test

（1）Server部署在阿里云上，client在校园中

（2）client通过WIFI和LTE连接server（在现在的情况下，WiFi的延迟小于LTE）

（3）下载的文件为200MB

	Bandwidth(Mbps)	Latency(ms)
WiFi	40	50
LTE	30	70



Conclusion

- 分析了异构的网络路径下MPTCP吞吐量降低的根本原因
- 提出了STMS，缓解host buffer和in-network buffer 大小受限而导致的问题。
- 实验证明了STMS在各种网络情况下均有明显的提高

项目支持方

（1）National KeyR&D Program of China under Grant 2017YFB1010002

（2）National 863 project (no. 2015AA015701)

（3） Protocol Research Lab, Huawei Technologies

谢谢大家