本文介绍了数据中心网络中的TCP吞吐量崩溃问题， “Incast”一词指的是在数据中心网络中的多对一通讯模式中出现的传输拥塞问题。文章建立了一种分析模型，分别从观测Incast症状、确认症结、探索可行的解决方案。

拥塞控制是当代计算机网络体系结构中一个无法回避的问题，TCP协议也是如此。这是一个普遍存在的问题，文章开头对可能造成Incast的原因做出了分析：RTO计时器的初始值不够小、交换器和路由的缓存过小、应用层解决方案不够等。

受到如MapReduce这种分布式存储应用和批块处理的启发，作者团队从工作流、测试平台和工具方面入手制定了一系列度量标准和测试规程。本文作者使用了很多前人的相关研究成果和思路，也借鉴了很多业界成熟的模型来为自己的测试作支撑，我认为这是做研究的一种好方法。

曾经在本科毕设阶段对Linux Kernel有多一些研究，为了定制出一个精简的内核来承载容器服务，我曾经想过来优化Linux TCP栈，最显而易见提升性能的方法就是禁用或者删除一些没有用的模块（如SELinux模块）来提升内核的启动速度。当时在网络栈的优化过程中遇到的困难是最多的，源于其结构的复杂和混乱，测试的难度也是相当高。文章作者也是从优化网络栈开始，减小、随机化TCP RTO计时器的初值，或者设置更小的乘法器等。大部分都是无用功。最有可能有成效的是减小RTO计时器的最小值。

作者团队接着从此点入手做了很升入的测试和分析，从不同大小的RTO计时器、时延ACK、高解析度计时器、工作流和测试平台方面进行的量化的测试，很直观的展示了结果。进而作者提出了量化模型：针对大的RTO计时器值，减小计时器值，针对较小的计时器则需要控制内包的等待时间。针对各种优化所带来的问题都在第六节做出了解释。

读完全文，我受益匪浅，相对于作者团队针对TCP Incast提出的分析模型和优化方案，我学到更多的是针对一个复杂的网络问题如何寻找切入口去分析，从一个普遍存在的问题入手，不断的寻找下一步的方向，自己制造新的问题并去用自己写的测试工具进行测试。文章最后的展望部分，作者也提到了使用机器学习来减少人工劳动，思维的发散性很重要。时刻将工业界和学术相结合，方法和学术可以相互借鉴。

（备注：附件PDF为阅读完整理的提纲）