

认知无线电网拥塞控制源算法的性能优化研究

撰文 / 重庆工程学院 李成勇

本文以 TCP 拥塞控制机制、原理为切入点,深度剖析了拥塞控制典型算法及传统无线电网环境下的 TCP 性能增强技术。针对目前 5 种典型 TCP 拥塞控制算法在认知无线电网中的应用进行比较和优化,通过对仿真结果进行比较和分析,本文认为 TCPW 在认知无线电网中性能最优。

一、引言

随着物联网技术在各领域的应用,无线通信技术的应用也越来越广,解决频谱资源的有效利用问题也变得越来越突出,尤其是无线局域网技术应用越来越成熟,宽带无线应用面临更高的要求,涉及宽带利用率及信息传输率等多方面问题处理。为了充分有效利用和管理宽带频谱资源,提出了动态频谱分配技术,用于解决频谱资源没有充分利用导致的频谱资源不足。认知无线电技术可以从时间、空间和频率等多维度对宽带频谱资源的重复利用和共享,能很好解决频谱资源不足等问题。

二、TCP 拥塞控制机制原理

拥塞通常是指在一个网络中报文分组过多,导致网络的性能下降,主要表现在发生传输信息变慢等情况。拥塞的网络状态主要是持续过载,此时网络资源的固有容量不能满足使用,或者资源不够分配,导致部分用户需要等待网络资源的分配,这是产生拥塞的根本原因。当网络资源使用量较少时,单位时间内成功地传送数据的数据量随着网络使用的增加而增加,两者是线性关系,而响应时间增长缓慢。当使用量达到网络固有容量时,单位时间内成功地传送数据的数据

则出现缓慢增长的趋势,但响应时间却急剧增加,通常把这个转折点称为 Knee。如果网络使用量继续增加,路由器开始丢包,当网络使用量超过一定量时,单位时间内成功地传送数据的数据急剧下降,而这一点通常称为 Cliff。单位时间内成功地传送数据的数据(吞吐量)和响应时间随网络使用量(负载)的变化曲线仿真结果如图 1 所示。从图 1 可以看出,网络使用量在 Knee 附近时网络的利用效率是最高的。为了避免出现网络拥塞或者对已经发生的拥塞做出应对处理,这就要求网络节点采取拥塞控制措施来解决。

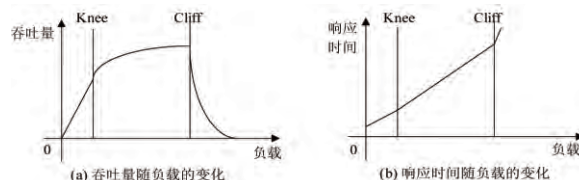


图 1 吞吐量和响应时间随负载的变化曲线

目前拥塞控制主要采用的是 TCP 拥塞控制,这种拥塞控制系统主要特征是具有自适应、分布式处理功能,在一定程度上能够解决网络拥塞,做出及时、有效的响应。TCP 拥塞控制算法有很多形式,有基于确认 ACK 的,还有基于时延 RTT。都是 TCP 拥塞控制算法的延伸和优化,各有所长。

三、拥塞控制源算法

传统的拥塞控制算法主要是依靠 TCP 协议，通过协议控制信息源端流入网络的数据流，这样的拥塞控制功能基本上都是在端系统上实现的。经典的 TCP 拥塞控制源算法 Reno、New-Reno、SACK、Vegas、Westwood 等，这些算法构成了拥塞控制源算法。

1. TCP Reno 和 TCP NewReno

Reno 在 Tahoe 的基础上增加了“快速恢复”算法，这种算法的优化主要是解决当传输网络处于半拥塞状态下，采用传统拥塞控制算法而出现发送通道尺寸突然变小导致信息快速传输后通道过分浪费的现象。所以算法在快速恢复阶段设计规则是，当前一次的所有信息在网络中传输结束后，才能允许下一次传输的信息进入网络。

Reno 中的“快速恢复 / 快速重传”算法在数据拥塞传输过程中有一个比较大的问题就是遇到拥塞可能出现丢弃多个报文的情况，这样会造成部分数据信息的缺失，导致传输信息不完整。为了进一步提高 TCP 的性能，NewReno 优化了 Reno 中的“快速恢复 / 快速重传”算法。其算法设计思路是：对于处理同一窗口中丢失的数据时，让网络连接状态一直处于快速恢复阶段，一直等到丢失数据全部恢复，源端才转换状态，停止快速恢复。如果遇到在数据传输过程中出现同一窗口丢失多个数据包的情况后，网络终端反复通过 ACK 通知网络源端重新发送没有恢复的数据，直到这个 ACK 不再发送通知为止，说明数据传输完成，故称为 Partial ACK。所以 NewReno 算法利用 Partial ACK 主要解决检测同一窗口中丢失的多个数据包的问题。

2. TCP SACK

SACK 算法的优化在于采用数据“发”的时候进行选择机制，“收”的时候进行确认机制，所以 Reno 和 SACK 两种优化算法最大的区别在于对于同一个窗口出现多个报文数据丢失的性能和应对的方式是不一样的。SACK 算法设计思路是：在网络连接进行初次握手时，网络起始端和网络终端互相进行命令提示，确定此次连接是否使用 SACK 命令选项。SACK 网络终端根据命令来选择丢失的数据信息再一次发送到源端，同时告知已经完成传输的数据信息，这样起始端不需要重复传输数据，只需重传丢失的数据。这样的算法思路设计可以达到在一个 RTT 内选择重传多个丢失的报文段，提高了 TCP 的性能。在数据传输丢失率处于中

间且长延迟网络环境中比较适用于 SACK 算法。

3. TCP Vegas

TCP Vegas（以下简称 Vegas）算法的创新点在于通过测量传输数据的 TCP 技术，其算法设计 3 种不同的传输机制有效得提高单位时间里传输的数据量、同时降低分组丢弃率。这 3 种不同的传输机制包括新的重传机制，新的拥塞避免机制和扩展的“慢启动”机制。由于 Vegas 是基于回路响应延时 RTT 估计的，它比基于丢包探测的 Reno 拥塞控制性能优越，比 Reno 算法提高 37% 到 71% 的吞吐量。

Vegas 是通过比较实际吞吐量来调整拥塞窗口大小的。

4. TCP Westwood

TCPW 也是一种基于测量的 TCP 算法。它和 Reno 算法的最大区别是降低盲目性，Reno 算法主要处理运算超时或者判断 ACK 是否重复出现，而算法的处理结果就是减半拥塞窗口。TCPW 算法设计的出发点抛开中间路由采用带宽优异估计算法，具体思路是当出现通信网络拥塞时，首先进行通信带宽大小的估计，主要用链路估算法，通过估算出的带宽智能调整启动门限值，控制窗口来匹配带宽，保证窗口的有效利用。这种控制方法称为 AIAD (Additive Increase Adaptive Decrease) 机制，主要用于无线网路下拥塞处理，算法性能相对较优。这种控制算法的工作特征是 ACK 速率的检测，通信源端通过估计目的端的通信速率，实时估算网络中能够应用的通信带宽数，当出现拥塞时，做出响应，调整可用带宽，适应拥塞窗口。具体算法如图 2 所示。

```
When(3 dupacks are received)
  ssthresh=(BWE*RTTmin)/PacketSize;
  If(cwin > ssthresh) cwin = ssthresh;
  When(coarse timeout expires)
    cwin=1;
    ssthresh=(BWE*RTTmin)/PacketSize;
  If ( ssthresh < 2 ) ssthresh =2;
```

图 2 TCPW 算法

在设计算法过程中，有 3 个参数值的注意，其中， BWE 代表网络数多少， $RTTmin$ 代表带宽估算值， $RTTmin$ 代表网络中检测的最小窗口门限值，其结果在理想情况下处理，以中间队列长度为零时得到的数值为准。

因此，TCPW 是在 Reno 算法上的改进，集合了 Reno 算法的优点，避免了 Reno 算法的拥塞窗口过分浪费的问题，TCPW 算法在出现拥塞时，能准确调整带宽窗口及窗口门限值，对传输的数据快速恢复。而且这种算法的最大优势在

于不减少数据包或中途检测，充分遵循了端到端的语义。

四、仿真设计及分析

为了验证拥塞算法及其改进算法在处理无线电环境下网路拥塞的效果，设计通信网络拓扑结构如图3所示，无线电网路覆盖面在 500×400 米范围内，在这个覆盖面存在3个网络节点，其中发送端有2个，分别是0节点和3节点，而接收端是2节点，1节点进行信息中转，是中继端，通信速率设置为2Mbps，是节点之间信息的传输速率。在15秒的时候，1节点1向正上方以5m/s的速度匀速前行，前行150s的时间。1节点移动范围始终保持在0节点和2节点通信频率覆盖面，且保持信号始终是畅通的。

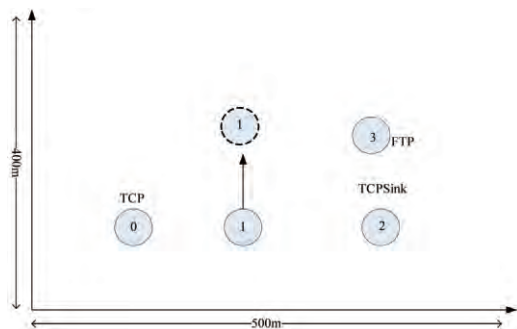


图3 纯无线环境下 TCP 仿真拓扑

仿真实验参数设定：主要比较 Reno、Newreno、Sack、Vegas 和 TCPW 等算法在无线网络和认知无线电网络两种通信环境情况下，各种参数的性能，参数包括拥塞窗口、瞬时吞吐量、阈值和丢包率，仿真结果如图4、图5、图6、图7所示，其中 Reno、Newreno、Sack、Vegas 和

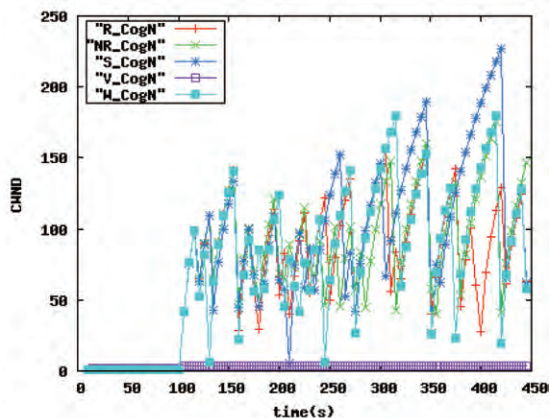


图4 认知无线电环境下 5 种算法拥塞窗口比较

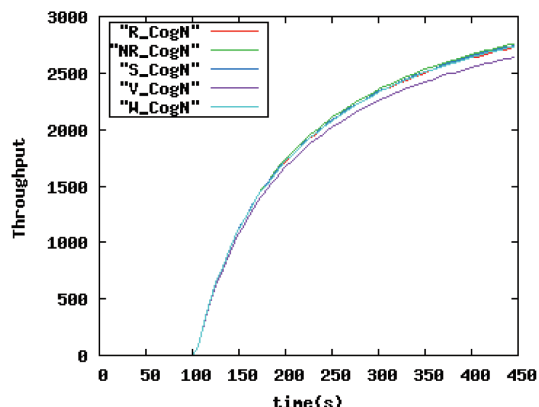


图5 认知环境下 5 种算法瞬时吞吐量比较

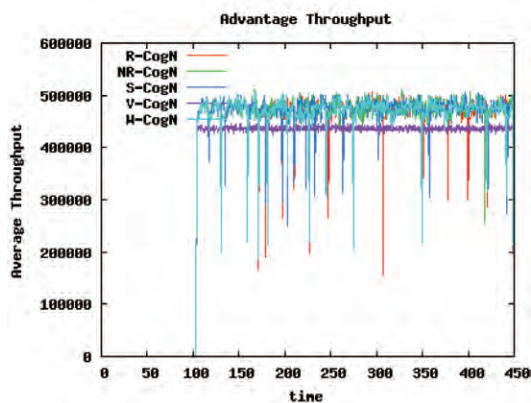


图6 认知环境下 5 种算法平均吞吐量比较

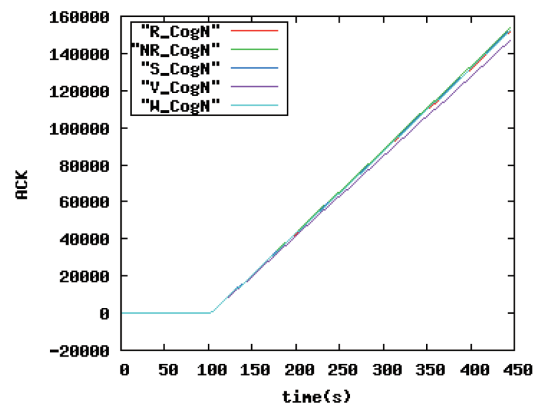


图7 认知环境下 5 种算法 ACK 比较

TCPW 各算法在认知无线电环境下仿真分别用 R_CogN、NR_CogN、S_CogN、V_CogN 和 W_CogN 表示，拥塞窗口用 CWND 表示，吞吐量用 Throughput 表示，门限值用 ssthresh 表示。

从图4至图7仿真结果可以得到各种拥塞算法在认知

无线电通信网络中各种参数（拥塞窗口、吞吐量）的变化趋势。得出如下结论：Reno 算法及其改进算法主要用于带宽较低和有限网络，所以处理认知无线电通信网络中的带宽范围变化较大的动态情况不理想，平均拥塞窗口和吞吐量都高于 Reno 算法或改进算法的是 TCPW 算法。从图 4 中可以看出，在处理网络拥塞时明显性能较好的是 Sack、TCPW 算法，Sack 算法主要采用应答选择方式，只要通信信息包非连续丢失，其处理拥塞时性能比较高，TCPW 算法是通过通信网络带宽使用量来估算当前网络使用效率以此判断是否出现拥塞，而且判别出是哪一种丢包情况，即误码丢包还是拥塞丢包，其处理网络拥塞情况不明时性能更优。Vegas 算法采用反馈 RTT 控制机制，比其它算法最大的差异在于在带宽分配过程中存在不公平现象，导致控制算法在慢启动动作重复性，网络的有效利用被降低。TCPW 算法采用带宽测量方式，根据通信带宽来合理分配，可以有效利用认知网络环境下的带宽、吞吐量，同时保证系统时延降低等性能。

图 6 和图 7 仿真 5 种拥塞算法在认知无线网络中的平均吞吐量和 ACK 变化趋势，可以得到：在平均吞吐量方面，Vegas 算法处理拥塞性能最差，变化趋势没有波动，其他算法处理效果区别不大，TCPW 算法采用 ACK 流控制拥塞窗口的动态变化，效果稍强于其它算法。原因在于在认知无线网络中，TCPW 算法调整慢启动门限值在通信过程中一直未中断，实时进行，门限值调整综合当前网络情况和历史情况智能决定，而 Reno 算法及其改进算法的 TCP 慢启动机制，慢启动时只是探测网络当前使用带宽，在不明网络状态下，把慢启动门限值设置为当前带宽窗口大小，与实时传输信息量不匹配，窗口过大导致重复传输，从而是 RTO 控制超时，降低了系统吞吐量。

为了进一步研究 5 种算法在处理网络拥塞时性能，丢包数、丢包率是另一个重要参数，5 种算法丢包数、丢包率在认知无线网络中的比较如表所示。

表 5 种 TCP 算法丢包率比较

TCP 算法	发送包	丢包数	丢包率
RenocogN	615458	16180	0.026289
NewrenocogN	621926	16836	0.027071
SackcogN	618480	618480	0.027635
VegascogN	595190	10	0.000017
TCPWcogN	616566	15949	0.025867

从表中数据可以比较得出：丢包率最小的是 Vegas 算法，但其在同时间内发送数据是最少的，很多带宽没有充分利用，造成网络资源浪费；其它 4 种算法丢包率差别不明显，Reno 算法稍好于 Newreno 算法，Newreno 算法稍好于 Sack 算法，TCPW 算法好于 Reno 算法、Newreno 算法和 Sack 算法，但其在同时间内发送数据是最多，相对而言丢包率也比较低。从各个参数比较情况综合来看，各种 TCP 拥塞控制算法在处理无线电认知网络中效果已经达不到想要的性能。因此从多参数分析评价，TCPW 算法在处理认知无线网络动态环境效果更理想。

五、结语

本文从 TCP 拥塞控制机制原理介绍出发，研究现阶段常用的 5 种典型的拥塞控制算法，提出针对传统无线网络环境下的 TCP 性能增强方案，从衡量拥塞控制机制的性能参数进行分析对比，得出可行性处理方案。由于无线网络的逐渐普及，网络结构及数据传输变得复杂且要求增高，处理过程呈现动态性、多态性，这使充分利用拥塞控制机制变得尤为重要。由于在无线电认知网络中，每一个通信节点有感知周围环境和重构能力，可以通过感知周围环境对频谱的使用情况进行实时监控，对认知无线电网络的频谱资源进行有效选择，以当前网络拥塞量的平均值作为下一次拥塞时的窗口门限值，调整变化大。在认知无线网络环境下，比较了 5 种典型 TCP 拥塞控制算法性能，通过对丢包、平均拥塞窗口、平均吞吐量、瞬时吞吐量和 ACK 等参数的仿真分析，结果表明 TCPW 在认知无线网络下性能最佳。IM