INVESTIGATING THE EFFECTS OF ACTIVE QUEUE MANAGEMENT ON THE PERFORMANCE OF TCP APPLICATIONS

AUTHOR

OMOSHOLA AKINBADE (PHD) DEPARTMENT OF COMPUTER SCIENCE UNIVERSITY OF LAGOS NIGERIA

AIM

Congestion occurs in the Internet when queues at routers fill to capacity and arriving packets are dropped ("lost"). Today, congestion is controlled by an adaptive mechanism built into TCP that regulates the transmission rate of a TCP connection. This mechanism dictates that each connection should detect instances of packet loss, interpret such instances as an indication of congestion, and respond to loss by reducing its transmission rate. After the rate has been reduced, the connection probes for additional bandwidth by slowly increasing its transmission rate. This adaptive behavior, applied independently on each end system, has been one of the keys to the operational success of the Internet. Nevertheless, as the Internet has grown, networking researchers and the Internet Engineering Task Force (IETF) have expressed concern about the scalability of pure end systems' congestion control. For example, pure end systems' congestion control mechanism only detects and reacts to a congestion event after a router queue has overflowed.

BACKGROUNG

In response to these concerns, active queue management (AQM) has been proposed as a router-based mechanism for early detection of congestion inside the network. AQM algorithms execute on network routers and detect incipient congestion by monitoring some function of the instantaneous or average queue size in the router. When an AQM algorithm detects congestion on a link, the router signals end systems and provide an "early warning" of congestion. This signaling is performed either explicitly, by setting a specific bit in the header of a packet, or implicitly by dropping some number of arriving packets. Many AQM algorithms have been proposed in recent years but none of them have been thoroughly investigated under comparable (or realistic) conditions in a real network. Moreover, existing performance studies have concentrated on network-centric measures of performance and have not considered application-centric performance measures such as response time.

RESULT

I investigated the effects of a large collection of existing AQM algorithms on the performance of TCP applications under realistic conditions in a real network. At a high-level, the primary results are that many existing AQM algorithms do not perform as well as expected when they are used with packet dropping. Moreover, a detailed investigation of the classical random early detection, or RED algorithm, has uncovered a number of design flaws in the algorithm. I have proposed and investigated a number of modifications to RED and other algorithms and have shown that my variants significantly outperform existing algorithms.

CONCLUSSION

this Research had reflected a promising results for AQM. When combined with packet marking, AQM algorithms significantly improve network and application performance over conventional drop-tail queues. Moreover, AQM enables network operators to run their networks near saturation levels with only modest increases in average response times. If packet marking is not possible, the research also shows how a form of differential treatment of flows that I invented can achieve a similar positive performance improvement. I have been able develop a new AQM algorithm that can balance between loss rate and queuing delay to improve the overall system performance.