INTRODUCTION

With the rapid development of Internet and the fast increase of terminal devices, the problem of network congestion has become critical. Congestion includes packet losses and the increase of transmission delay, and both of them affect the performance of upper layer, as well as the stability and robust of network. That is the reason that effective congestion control is critical to the stability of Internet and it is also the foundation of management control mechanisms and applications.

To control Internet congestion effectively and incessantly, at one hand, the existing TCP congestion control algorithm should be improved continuously, and the compatible end to end congestion control should be adopted for each network terminal. At another hand, proper scheme algorithm or queue management strategy, such as First-In-First-Out (FIFO) packet scheme combined with Drop-tail or Active queue Management (AQM) strategy, need to be added to IP layer to avoid congestion. As an extremely complex system, Internet has a complex, huge scale, and time-varying network structure, so that modelling it becomes much more complicated.

During the process of congestion control, we will use the Explicit Control Protocol (XCP), an experimental congestion control protocol.

XCP is designed to deliver the highest possible end-to-end throughput over a broad range of network infrastructure, including links with very large bandwidth-delay products, which are not well served by the current control algorithms. XCP is potentially applicable to any transport protocol, although initial testing has applied it to TCP in particular. XCP routers are required to perform a small calculation on congestion state carried in each data packet.

XCP routers also periodically recalculate the local parameters required to provide fairness. On the other hand, there is no per-flow congestion state in XCP routers.

1.2 IMPORTANCE OF STUDY
For the internet to continue to thrive, its congestion control mechanism must remain effective as the network evolves. Technology trends indicate the future internet will have a very large number of very high-bandwidth links. Less ubiquitous but still commonplace will be satellite and wireless links with high latency. These trends are problematic because TCP reacts adversely to increases in bandwidth or delay. Mathematical analysis of current congestion control algorithms reveals that, regardless of the queuing scheme, as the delay bandwidth product increases, TCP becomes oscillatory and prone to instability.

By casting the problem into a control theory framework, show that as capacity or delay increases, Random Early Discard (RED), Random Early Marketing (REM), Proportional Integral Controller and Virtual Queue all eventually become oscillatory and prone to instability. They further argue that it is unlikely that any Active queue Management scheme (AQM) can maintain stability over very high-capacity or large delay links.

Adaptive Virtual Queue (AVQ) also becomes prone to instability when the link capacity is large enough (e.g. Gigabit links).

Inefficiency is another problem facing TCP in the future Internet. As the delay-bandwidth product increases, performance degrades. TCP’s additive increase policy limits its ability to acquire spare bandwidth to one packet per RTT. Since the bandwidth-delay product of a single flow over very high bandwidth links may be many thousands of packets, TCP might waste thousands of RTTs ramping up to full utilization following a burst of congestion. Further, the increase in link capacity does not improve the transfer delay of short flows (the majority of the flows in the internet). Short TCP flows can not acquire the spare bandwidth faster than “slow start” and will waste valuable RTTs ramping up even when bandwidth is available.

Additionally, since TCP’s throughput is inversely proportional to the RTT, fairness too might become an issue as more flows in the current Internet traverse satellite links or wireless WANs. As users with substantially different RTTs compete for the same bottleneck capacity, considerable unfairness will result. Although the full impact of large delay-bandwidth products is yet to come, we can see the seeds of these problems in the current Internet. For example, TCP over satellite links has revealed network utilization issues and TCP’s undesirable bias against long RTT flows. Currently, these problems are mitigated using ad-hoc mechanisms such as ack
spacing, split connection, or performance enhancing proxies. This paper compare a novel protocol for congestion control that outperforms TCP in conventional environments, and further remains efficient, fair, and stable as the link bandwidth or the round trip delay increases. This Explicit Control Protocol, XCP, generalizes the Explicit Congestion Notification proposal (ECN). Instead of the one bit congestion indication used by ECN, our routers inform the senders about the degree of congestion at the bottleneck. Another concept is the decoupling of utilization control from fairness control. To control utilization, this protocol (XCP) adjusts its aggressiveness according to the spare bandwidth in the network and the feedback delay. This prevents oscillations, provides stability in face of high bandwidth or large delay, and ensures efficient utilization of network resources. To control fairness, the protocol reclaims bandwidth from flows whose rate is above their fair share and reallocates it to other flows.

With the rapid development of Internet and the fast increase of terminal devices, the problem of network congestion has become critical. Congestion includes packet losses and the increase of transmission delay, and both of them affect the performance of upper layer, as well as the stability and robust of network. That is the reason that effective congestion control is critical to the stability of Internet and it is also the foundation of management control mechanisms and applications [10].

The following are some of the techniques commonly used to avoid congestion for high bandwidth delay network.

**TCP**

Technology trends indicate the future internet will have a very large number of very high-bandwidth links. Less ubiquitous but still commonplace will be satellite and wireless links with high latency. These trends are problematic because TCP reacts adversely to increases in bandwidth or delay. Mathematical analysis of current congestion control algorithms reveals that, regardless of the queuing scheme, as the delay bandwidth product increases, TCP becomes oscillatory and prone to instability.

One of TCP’s primary functions is to properly match the transmission rate of the sender to that of the receiver and the network. It is important for the transmission to be at a high enough rate to ensure good performance, but also to protect against overwhelming the network or receiving
TCP’s 16-bit window field is used by the receiver to tell the sender how many bytes of data the receiver is willing to accept. Since the window field is limited to a maximum of 16 bits, this provides for a maximum window size of 65,535 bytes. The window size advertised by the receiver tells the sender how much data, starting from the current position in the TCP data byte stream can be sent without waiting for further acknowledgements. As data is sent by the sender and then acknowledged by the receiver, the window slides forward to cover more data in the byte stream. This concept is known as a “sliding window”.

**TCP Reno:**

TCP Reno is the most popular version of TCP used on the current Internet. It increases the congestion window size on the reception of an ACK packet and decreases it when packet loss occurs. Data within the window boundary is eligible to be sent by the sender. Those bytes in the stream prior to the window have already been sent and acknowledged. Bytes ahead of the window have not been sent and must wait for the window to “slide” forward before they can be transmitted by the sender. A receiver can adjust the window size each time it sends acknowledgements to the sender. The maximum transmission rate is ultimately bound by the receiver’s ability to accept and process data. However, this technique implies an implicit trust arrangement between the TCP sender and receiver. It has been shown that aggressive or unfriendly TCP software implementations can take advantage of this trust relationship to unfairly increase the transmission rate or even to intentionally cause network overload situations.
Active Congestion Control (ACC) uses Active Networking (AN) technology to make feedback congestion control more responsive to network congestion. Current end-to-end feedback congestion control systems detect and relieve congestion only at endpoints.

ACC includes programs in each data packet that tell routers how to react to congestion without incurring the round trip delay that reduces feedback’s effectiveness in wide area networks. The congested router also sends the new state of the congestion control algorithm to the endpoints to ensure that the distributed state becomes consistent. We discuss a model for extending feedback congestion control into an Active Network, apply that model to TCP congestion control, and present simulations that show that the resulting system exhibits up to 18% better throughput than TCP under busy traffic. In simulations without busy traffic, the systems behaved comparably.

**XCP**

XCP represents a significant advance in Internet congestion control: it extracts congestion information directly from routers, without any per-flow state. XCP should be able to deliver the highest possible application performance over a broad range of network infrastructure, including extremely high speed and very high delay links that are not well served by the current control algorithms. XCP achieves fairness, maximum link utilization, and efficient use of bandwidth. XCP is novel in separating the efficiency and fairness policies of congestion control, enabling routers to put available capacity to work quickly while conservatively managing the allocation of capacity to flows. XCP is potentially applicable to any transport protocol, although initial testing has applied it to TCP in particular.

XCP’s scalability is built upon the principle of carrying per-flow congestion state in packets. XCP packets carry a congestion header through which the sender requests a desired throughput. Routers make a fair per-flow bandwidth allocation without maintaining any per-flow state. This enables the sender to learn the bottleneck router’s allocation to a particular flow in a single round trip.