Application of Network Coding in TCP

Sohil Mahajan (sohil@), Ankit Singh Rawat (ankitsr@), Kartik Venkat (kartikv@iitk.ac.in)

Abstract—In this project we have implemented a model for application of network coding into the TCP stack. Further, we have presented a description of the network coding layer, and the effects of network coding on parameters such as link throughput in a lossy network. In this modified scheme, the source transmits a linear combination of \( n \) packets, within a novel interpretation of ACK’s - which provide for a smooth reaction to packet drops. Implementation has been performed on the ns-2 platform. The algorithmic basis for our approach is provided in [1]

I. MOTIVATION

Over the past few decades, TCP (Transmission Control Protocol) has established itself as a robust framework for internetwork communication. This protocol provides for variable packet sizes, transmission failures, sequencing, flow control, end to end error checking, and creation of logical process connections. The flexibility and power offered by TCP is the reason why it has been adopted in the functioning of large scale distributed networks such as the internet.

Network Coding has recently emerged as an important field of study in communication systems. The basic idea of network coding is to facilitate efficient sharing of available network resources to maximize robustness and throughput in the network system. Conventional techniques involve store and forward paradigms which do not exploit the structural topology and temporal information available in the network. Network Coding therefore introduces an intelligence factor in the operation of the system. The packets sent by nodes are mixed across times and across flows. The basic aim of network coding is to allow individual links to achieve optimal capacities even in large scale distributed networks.

Network Coding simultaneously provides throughput gain in a static environment while drastically improving robustness and adaptability in a distributed network. In [1] Sundararajan et. al. have proposed a novel scheme to incorporate the network coding with the existing TCP with minimal changes in the protocol stack. We want to simulate the results of this proposed scheme, so as to gain insight into this new and fast growing area

II. OBJECTIVES

In this project we seek to explore the applicability of network coding in TCP which is the protocol of choice in several distributed networks. Particularly we seek to exploit the robustness of network coding to provide congestion control in a lossy link scenario (such as a wireless scenario).

Our aim is to integrate network coding in a manner that requires minimal changes to the protocol stack. We detail below, the methodology adopted and the technical specifications in the implementation of this project.

III. SPECIFICATIONS

The protocol stack which incorporates the end-node coding layer is as depicted below. We have introduced a new layer referred to as the Network Coding Layer, between the TCP and the IP Layers. In this layer we perform operations upon packets from the TCP source end. Random Linear Combinations are made of \( n \) packets using coefficients chosen over a finite field \( F_q \) of size \( q \). The packets themselves are treated as vectors over the same field. We now proceed to define a concept of "seeing a packet" which exploits the fact that reliable information transmission does not depend upon reception of exact packets, but on receiving sufficient information. A node is said to have seen a packet \( p_k \) if it has enough information to compute a linear combination of the form \( (p_k + q) \) where \( q = \sum_{l>k} \alpha_l p_l \).

We now present below, an example of such a linear combination and the associated ACK deployment upon seeing a new packet. In this manner we avoid Duplicate ACKs’ which is the bottleneck as far as throughput is concerned in a lossy link running on TCP. The packet loss will be viewed as a queuing delay at TCP sender. We have used TCP Vegas for the purposes of this project because of certain desirable properties that congestion avoidance algorithms in Vegas adopt. Particularly, it is different in its approach of incorporating congestion avoidance algorithms based on estimated variance of packet Round Trip Time.

We shall now look at the implementation aspects and platform details which we have employed
IV. IMPLEMENTATION

The simulation of the network coding protocol in TCP has been performed on ns-2 in the Linux environment. Network Simulator is a discrete event simulator targeted at networking research. It provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) network. The following figure provides an overview of the functionality in the network simulator -

- **Packet Structure in Network Simulator**
- **Creation of a new protocol**
  We have implemented a new Agent in the TCL class of ns-2 which is a Sub Agent of TCP Vegas. Here we have defined the attributes of a packet generated by this agent. Particularly, we store the coefficients of the linear combination process in a newly defined header. These coefficients are generated in a random fashion using RAND() Function in C.

Corresponding the newly created TCP-NC Agent, we have had to implement the novel acknowledgment procedure at the receiver end. Based on the RTT of each packet determined, TCP Vegas will automatically perform congestion control.

V. SIMULATIONS AND DEMONSTRATION

**Network Topology**

**Specifications:** All the links have a bandwidth of 1 Mbps, and a propagation delay of 100 ms. The buffer size on the links is set at 200. The TCP receive window size is set at 100 packets, and the packet size is 1000 bytes.

The following plots represent a comparative analysis of conventional TCP Vegas with the newly proposed TCP Vegas with Network Coding. The inferences drawn from the analysis is detailed in the Summary section.
VI. CONCLUSION AND FUTURE WORK

In this project we have explored and implemented a novel congestion control technique which integrates the concept of network coding with an existing TCP protocol. We have further developed a new protocol in network simulator incorporating coding in the control algorithm. We have also presented a comparative analysis of TCP Vegas with and without network coding. The results are very promising with a marked increase in throughput of the network when network coding is employed in a lossy network. The exponential decay in throughput is transformed to a linear decrease with % increase in loss probability. In the future we hope to implement this strategy while allowing intermediate nodes to code as well. Theoretically, a lot more can be gained by allowing this, though it may have problems with integration into the TCP stack as we are familiar with. We can also explore applications in a more distributed and general network setting involving multicast transmission.

REFERENCES
[2] NS By Example
[4] Network Coding Homepage