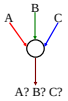
Secure Network Coding for a P2P System

Hun J. Kang[1], Aaram Yun[1], Eugene Y. Vasserman[1], Hyung-Tae Lee[2], Jung Hee Cheon[2], Yongdae Kim[1]

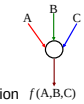
University of Minnesota[1] Seoul National University[2]

Introduction

What is Network Coding?



- Traditional Coding/Routing
  - Coding is done on *end-to-end basis* (if necessary)
  - Intermediate nodes forward received data
- Network Coding
  - Allowing intermediate nodes to manipulate information (to perform coding)
    - Information flowing out of a node is a function of information coming into the node



Network Coding is Useful for P2P

File Sharing

BitTorrent, GNUTELLA, Live Messenger FolderShare, Groove

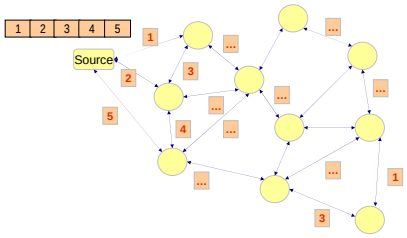
Multimedia Broadcasting/Streaming

PPLive, UUSee.com, Windows Media, real

Distributed Storage Systems, …

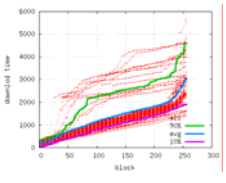
BitTorrent-like P2P Systems

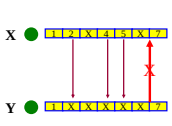
Peers exchange blocks



Problems of BitTorrent-like Systems

- Last (rare) block problem
  - Long download time of last missing blocks
- Inefficient Tit-for-Tat
  - Peers with many blocks: lower chance to get missing blocks
  - Peers with few bocks: smaller chance to be "unchoked"





Benefits of Network Coding

- Solves the last (rare) block problem
- Makes efficient tit-for-tat possible
- Robust to churn and attacks

Problems with Network Coding

Pollution Attack

- What if a node sends out an incorrect linear combination, or just a random packet?
- It will be quickly mixed with and corrupt other packets
- Potentially prevents all nodes from properly decoding

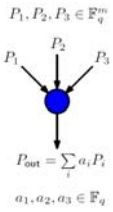
Traditional Solutions Do Not Work

- Traditional digital signatures cannot be used for preventing pollution attack:
  - Intermediate nodes can't sign 'new' packets
  - Only after decoding, a receiver may realize that the signature doesn't match

Our Solution

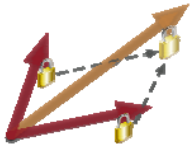
Linear Network Coding

- Information
  - Packets are vectors over finite field with a fixed dimension
- Encoding
  - Outgoing packets are linear combination of incoming packets
- Decoding
  - By solving a system of linear equations



Homomorphic Signature

- Source can sign individual packets
- Everyone can verify
- Everyone can *combine signatures* to produce a signature for linear combination of packets
  - Only way to produce new signatures



Our Homomorphic Signature

- Private key = $(s_1, \ldots, \ldots, s_{m+n})$
- Public key = $p, q, g, (g^{s(1)}, g^{s(2)}, \ldots, \ldots, g^{s(m+n)})$

- Signature generation at the source
  - network-coded block $w_i = (w_{i,1}, \ldots, w_{i,m+n})$
  - *signature* $\sigma_i = s_1 w_{i,1} + s_2 w_{i,2} + \ldots + s_{m+n} w_{i,m+n}$
- Signature aggregation at intermediate nodes
  - new block $w' = \alpha_1 w_1 + \ldots + \alpha_k w_k$
  - new signature $\sigma = \alpha_1 \sigma_1 + \ldots + \alpha_k \sigma_k$
- Signature σ can be verified by checking
  - $g^{\sigma} = (g^{s1})^{v1} \ldots (g^{sm+n})^{vm+n}$?
  - $v = (v_1, v_2, \ldots, \ldots, v_{m+n})$ is the corresponding coded block

Analysis and Results

Comparison

Pros

- Very small initial cost, signing cost, and aggregating cost, compared with other schemes
  - Initial cost (source, before sending file):
    - Close to single verification cost
    - For some schemes, it is close to total verification cost
  - Signing and aggregation cost is negligible
- Based on simple modulus arithmetic
  - No elliptic curves, pairings, …

Cons

- Key pair can be used only once
  - New key pair should be used to send a new file
  - To prevent two files mixed and corrupt each other
  - Not a big problem for our purpose: the size of public key about 1% of the whole file
  - Many other schemes share the same characteristic

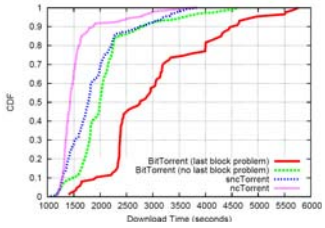
Implementation and Evaluation

Implementation and Tests

- Implemented to Mainline BitTorrent client 5.2
- Tested in PlanetLab

Preliminary Results

Performance improvement compensates for overhead



Work in Progress

System Issues in Practice

- Finding "good" system parameter values
- Reducing encoding/decoding overhead
- Solving "initial credit problem" and "free-riding"
- Optimizing implementation for better performance

Architecture for P2P Content Distribution with Network Coding

- Semi-Structured Overlay Multicast
  - Peer neighbor selection: DHT topology
  - Data delivery: random paths
  - Robust and efficient

Semi-structured overlay multicast reduces delivery time