Large File Distribution Using Efficient Generation-based Network Coding

Ye Li
Steven Blostein
Wai-Yip Chan

Department of Electrical and Computer Engineering

Queen's University

Outline

Outline

System Model and Background

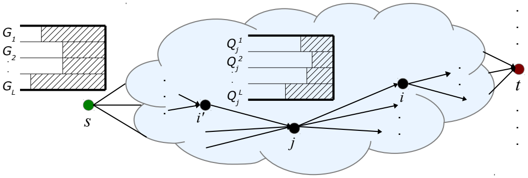
Proposed Scheme: Encoding, Decoding and Scheduling

Analysis and Design of the Proposed Scheme

Simulation Results

Summary

Targeted System Model



- Distribute a large file via a complex network (e.g. cloud system).
- Duplicate big data archives to different places for easier access.
- Links are lossy or unreliable: can be modeled as erasure channels.
- Big data means complex protocol design and coordination messages between nodes (e.g. for ARQ).
- Motivation: a reliable scheme with less node coordination needed.
- Network coding could be a potential solution.

Network Coding for Large File Transmission

Network coding: allows intermediate nodes to combine incoming packets before forwarding downstream.

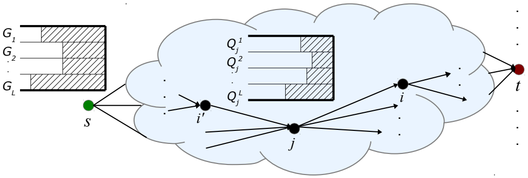
Benefits:

- Achieves max-flow capacity of the network.
- Can be implemented in a distributed way using random linear network coding (RLNC).
- It is rateless, meaning that upstream can keep sending packets while the destinations keep collecting coded packets.
- No protocol messages are exchanged before sinks finish decoding.

Challenges

- Decoding of NC needs to solve a linear system of equations.
- Using Gaussian elimination, the decoding has cubic complexity.
- Cost is too high for large files.

Generation-based Network Coding (GNC)



- Purpose: reduce decoding complexity of NC.
- Group a large number of packets into multiple smaller subsets, called generations.
- NC is performed within each generation.
- Scheduling is needed in transmission of generations.
- Only local scheduling is allowed in order to maintain ratelessness.
- *Local scheduling* is not network-wide optimal.
- Transmission overhead ε: N source packets need $(1+\varepsilon)N$ coded packets to recover the source.
- Overlapping generations can be used to combat the issue.

Outline of Contributions

We propose to use overlapping GNC to achieve rateless transmission of large files in a network.

- A maximum local potential innovativeness strategy is proposed to perform local scheduling.
- And-or tree analysis technique is used to analyze the performance of overlapping GNC.
- The optimal amount of overlap among generations can be determined through an optimization problem that based on the and-or tree analysis result.

Outline

System Model and Background

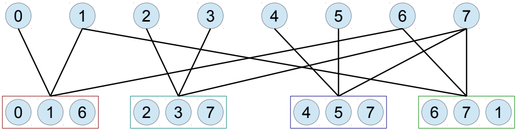
Proposed Scheme: Encoding, Decoding and Scheduling

Analysis and Design of the Proposed Scheme

Simulation Results

Summary

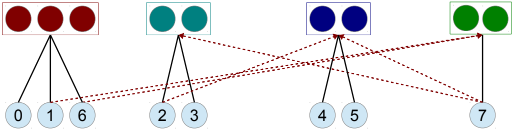
Overlapping GNC: Grouping and Encoding



- $N = 8$ packets are randomly grouped into $L = 4$ generations with replacement, each with size $a_R = 3$.
- *Overlapping* because some packets are in multiple generations.
- Make sure that each packet is contained in at least one generation.
- Amount of overlap is determined by a_R.
- Synchronized discrete time system: each node, including the source, sends a packet on each of its outgoing link in each time slot.
- Coded packets from generation l are in the form of

$$\mathbf{p}^{(l)} = \sum_{j=1}^{a_R} g_j^{(l)} \mathbf{s}_j^{(l)}.$$

Overlapping GNC Scheme: Decoding



- Received packets are buffered according to generation index.
- Generations are decoded by GE in the finite field.

$$\begin{bmatrix} g^l_{1,1} & \cdots & g^l_{1,a_R} \\ \vdots & \ddots & \vdots \\ g^l_{N_l,1} & \cdots & g_{N_l,a_R} \end{bmatrix} \begin{bmatrix} \mathbf{s}^l_1 \\ \vdots \\ \mathbf{s}^l_{a_R} \end{bmatrix} = \begin{bmatrix} \mathbf{r}^l_1 \\ \vdots \\ \mathbf{r}^l_{N_l} \end{bmatrix}$$

- Decoded packets are substituted to not-yet-decoded generations.
- One step of GE plus back-substitutions is referred to as an iteration.
- Decodable: "# of received innovative packets" $\geq$ "# of unknowns".

Queen's

Overlapping GNC Scheme: Scheduling

- Scheduling: determine from which generation to form a coded packet.
- On each outgoing link of the source node, generations are scheduled in a round-robin way.
- Intermediate nodes choose a generation from those who have packets been buffered.
- *Maximum local potential innovativeness* (MaLPI) scheduling at intermediate node.

Maximum Local Potential Innovativeness Scheduling

- Chooses the generation that is most likely to generate a useful packet for downstream.
- Intermediate node j chooses the generation of the index

$$l_{ji}^{*}(n) = \arg\max_{l}(|Q_j^l(n)| - S_{ji}^l(n))$$

  on its outgoing link (j, i).
- $Q_j^l(n)$ - the number of buffered packets of generation l at j.
- $S_{ji}^l(n)$ - the number of scheduled times of l on link (j, i).
- Local buffer status is utilized, but no node-coordination.
- Previous works use random scheduling, which randomly picks up a generation regardless of buffer status.
- The strategy is shown to have the lowest probability to generate a useless packet when the scheduling is performed locally.

Outline

System Model and Background

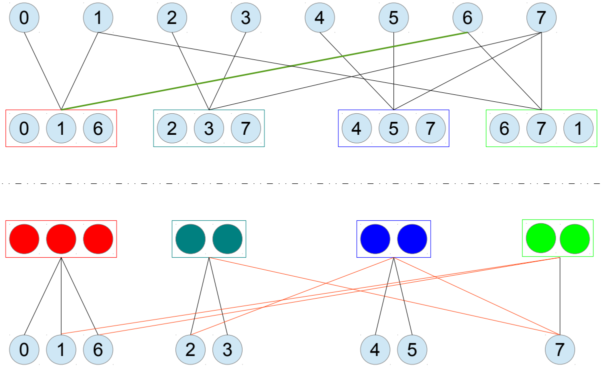
Proposed Scheme: Encoding, Decoding and Scheduling

Analysis and Design of the Proposed Scheme

Simulation Results

Summary

Revisit the Encoding and Decoding Example



- Some numbers of coded packets for each generation are received.
- Decoding is a combination process of Gaussian elimination and back-substitution between generations.
- A kind of message-passing process on the bipartite graph.

Decoding Analysis: And-Or Tree Analysis

- The message-passing modeling makes analysis possible.
- Task: Determine/predict the decodability of generation construction for given numbers of received packets of each generation.
- Treat the generation construction as a random bipartite graph connecting packets and generations on two sides.
- Packet-side degrees: $\Psi(x) = \sum_{i=1}^{L} \Psi_i x^i$, where Ψ_i denotes the probability that a packet is contained in i generations.
- Generation-side degree is a constant: generation size.
- Each generation is associated with an *unknown degree* during the decoding, which equals to the number of undecoded packets it connects and evolves as the decoding proceeds.
- Decoded packets will decrease unknown degrees of generations who also contain them, resembling a message-passing mechanism.

Decodability in Iterations

Given the number of sent packets from each generation, n_g, the end-to-end erasure probability, ϵ, generation size a_R, packet-node degree distribution $\lambda(x) = \Psi'(x)$, we can write the probability that one generation is not decodable in one iteration as an iterative function of itself in the previous iteration, y:

$$
\begin{aligned}
& f(\lambda(y), \tau_{n_g,\epsilon,\mu}, a_R) \\
= & \sum_{\mu=a_R}^{n_g} \tau_{n_g,\epsilon,\mu} \left(1 - p_{a_R,\mu}\right) \\
& + \sum_{\mu=0}^{a_R-1} \tau_{n_g,\epsilon,\mu} \sum_{k=0}^{\mu-1} \binom{a_R - 1}{k} (\lambda(y))^k (1-\lambda(y))^{a_R-1-k} \left(1 - p_{k+1,\mu}\right) \\
& + \sum_{\mu=0}^{a_R-1} \tau_{n_g,\epsilon,\mu} \sum_{k=\mu}^{a_R-1} \binom{a_R - 1}{k} (\lambda(y))^k (1-\lambda(y))^{a_R-1-k},
\end{aligned}
$$

Iteration begins from $y = 1$, i.e., when no generations are decoded.

Decodability in Iterations (Cont'd)

- Iteration begins from $y = 1$.
- Iteration ends at $y = \delta = \inf\{f(\lambda(y), \tau_{n_g,\epsilon,\mu}, a_R)\}$.
- In reality, δ corresponds to not-decodable generation fraction in the end.
- Precode can be used to recover this last fraction, δ is related to the precode rate.
- To successfully decode a $(1-\delta)$ fraction of generations, we need

$$f(\lambda(y), \tau_{n_g,\epsilon,\mu}, a_R) < y, \quad y \in [\delta, 1]$$

- For randomly grouped generations, $\lambda(x)$ is shown to be

$$\lambda(x) \approx \left(\frac{d_R}{a_R} + \left(1 - \frac{d_R}{a_R}\right)x\right) \mathrm{e}^{-(a_R/d_R-1)(1-x)}$$

- Analysis relates overhead (n_g) with generation construction parameters, a_R and $\lambda(x)$.

Estimate Optimal a_R Using the Analysis Result

Task: Finding the optimal amount of overlap, i.e., a_R such that the fewest transmissions are wasted, or equivalently the highest rate is achieved.

$$\begin{aligned} \underset{a_R}{\text{minimize}} \quad & n_g \\ \text{subject to} \quad & f(\lambda(y), \tau_{n_g,\epsilon,\mu}, a_R) < y, \quad y \in [\delta, 1], \end{aligned}$$

- Minimize the number of packets needed to be sent from the source.
- Equivalently minimizing the number of wasted transmissions from an end-to-end viewpoint.

Queen's

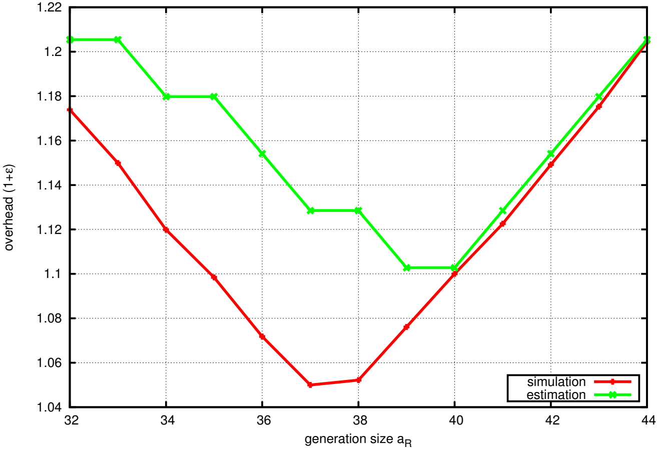
Estimate Optimal a_R for Multiple Users

$$\begin{aligned} \underset{a_R}{\text{minimize}} \quad & \frac{1}{T}\sum_{t=1}^{T} n_T(1-\epsilon_t) \\ \text{subject to} \quad & f(\lambda(y), \tau_{n_T,\epsilon_t,\mu}, a_R) < y, \quad \forall t, y \in [\delta, 1], \end{aligned}$$

- n_T packets per generation are sent from the source.
- ϵ_t is the end-to-end erasure probability between source and user t
- Minimize n_T subject to that all users can decode $(1-\delta)$ fraction of generations.

Outline

Why Estimating a_R is Important?



- $N_s = 65536$ source packets, precoded to $N = 67229$ packets and grouped into $L = 2101$ generations.
- End-to-end lossy link: $\epsilon = 0.2$, $\delta = 0.02$.

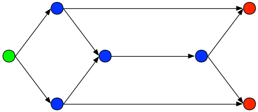
Using Estimated a_R

- Estimated optimal a_R vs. Actual optimal a_R.
- Actual optimal a_R is obtained by extensive simulations.

ϵ	estimated a_R	$1+\hat{\varepsilon}$	optimal a_R	$(1+\bar{\varepsilon})$
0.1	37	1.0587	36	1.0447
0.2	39	1.0752	38	1.0522
0.3	40	1.0754	39	1.0547
0.4	41	1.0788	39	1.0515
0.5	42	1.0842	40	1.0484
0.6	43	1.0916	41	1.0567

- Estimated a_R and the corresponding overhead performance are close to the actual optimal.

Using MaLPI Scheduling and Estimated a_R in Networks



- Simulated in a butterfly network, each link with erasure prob ϵ.
- Max-flow capacity: $2(1-\epsilon)$.
- Compare to random scheduling and when no overlap is introduced.

	Random Scheduling		MaLPI	
	rate	operations	rate	operations
$\epsilon = 0.1,\ a_R = 32$	1.0579	37.10	**1.2452**	34.35
$\epsilon = 0.1,\ \hat{a}_R = 37$	**1.1767**	42.80	1.5148	39.82
$\epsilon = 0.2,\ a_R = 32$	0.9300	37.23	**1.1178**	34.80
$\epsilon = 0.2,\ \hat{a}_R = 39$	**1.0795**	45.54	1.3034	43.14

- Achieves 80% capacity of the lossy butterfly network without node coordination.

Outline

Summary

- Our target: achieve rateless transmission of large files over networks.
- Our solution: use overlapping generation-based network coding.
- MaLPI: generations are scheduled using local information in the middle.
- An and-or-tree analysis technique is used to analyze overlapping GNC.
- Improve generation construction by optimizing.

Questions?

Thank you!