Outburst: Efficient Overlay Content Distribution with Rateless Codes

Chuan Wu and Baochun Li

Department of Electrical and Computer Engineering
University of Toronto
{chuanwu, bli}@eecg.toronto.edu

Abstract. The challenges of significant network dynamics and limited bandwidth capacities have to be considered when designing efficient algorithms for distributing large volumes of content in overlay networks. This paper presents *Outburst*, a novel approach for overlay content distribution based on *rateless codes*. In Outburst, we code content bitstreams with rateless codes at the source, and take advantage of the superior properties of rateless codes to provide resilience against network dynamics and node failures. We recode the bitstreams at each receiver node, so that the need for content reconciliation in parallel downloading is eliminated, and the delivery of redundant content is minimized. The effectiveness and efficiency of Outburst are demonstrated with simulations.

Key words: Overlay Network, Rateless Codes, Content Reconciliation

1 Introduction

As compared to traditional solutions using multiple unicasts, content distribution over overlay networks offers more efficient bandwidth usage and server load distribution. There are, however, two key challenges in overlay distribution of large volumes of data.

First, to achieve higher throughput and failure resilience, *parallel downloading* from multiple overlay nodes becomes typical in most recent proposals. Nevertheless, a risk rises that the same content may be unnecessarily supplied by multiple upstream nodes. To maximize bandwidth efficiency, a receiver needs to reconcile the differences among a set of upstream nodes before the actual downloading, a problem referred to as *content reconciliation*. In large-scale overlay networks, such a reconciliation process constitutes a complicated and bandwidth-intensive task [1].

Second, overlay content distribution sessions may be routinely disturbed by dynamics in overlay networks, such as node departures and failures. Throughput for bulk data downloading may be significantly affected in case of such dynamics.

This paper proposes *Outburst*, a novel approach which utilizes *rateless codes* to address both challenges. *Rateless codes*, such as LT codes [2], Raptor codes [3] and online codes [4], possess the important characteristic of being extremely loss resilient. In *Outburst*, we take advantage of such loss resilience to achieve the desirable resilience against losses and node dynamics. In addition, we discuss possible solutions towards solving the content reconciliation problem, and propose an approach based on rateless

recoding at each participating overlay node. Our rateless recoding proposal can completely eliminate the need for content reconciliation in parallel downloading, based on other salient properties of rateless codes.

The remainder of this paper is organized as follows. In Sec. 2, we present our network model for using rateless codes, and discuss the recoding approach. The baseline protocol and dynamics handling protocol are presented in Sec. 3. We present simulation results, discuss related work and conclude the paper in Sec. 4, Sec. 5 and Sec. 6.

2 Outburst: Efficient Content Distribution with Rateless Codes

In this paper, we consider content distribution in mesh overlay topologies, consisting of one data *source* S and multiple *receivers* in T. Each receiver is served by one or more *upstream nodes*, and may serve one or more *downstream nodes*. We divide the bulk data file to be distributed into *segments* $s_1, s_2, \ldots$. Each segment contains k blocks, and each block has a fixed length of L bits. In *Outburst*, we code each segment with a rateless code and deliver coded blocks for each segment in the network.

2.1 Source Coding with Rateless Codes

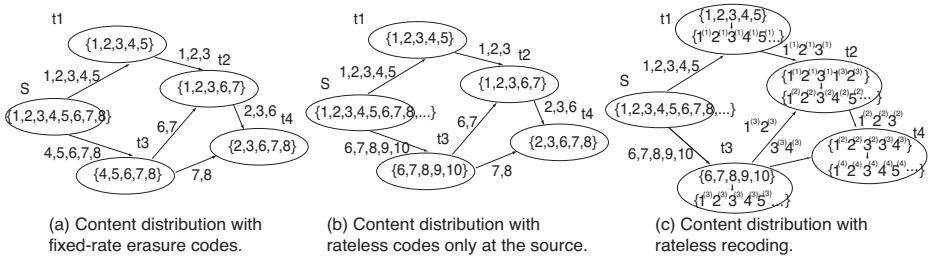

Fig. 1. Content reconciliation problem with different coding schemes: a comparison.

We now motivate the use of rateless codes in *Outburst*. In contrast with traditional erasure codes, the benefits of rateless codes are related to the fundamental challenges in overlay content distribution: *volatile network dynamics* and *content reconciliation*.

Erasure source coding has been used in recent years to cope with network dynamics in content distribution [1, 5]. A traditional (n, k) erasure code, such as Reed-Solomon codes and Tornado codes [5, 6], is a forward error correction code with parameters k, the number of original symbols, and n, the number of coded symbols. The ratio k/n is referred to as the *rate* of the code.

An erasure code is loss-resilient, since if any k (or slightly more than k) of the n coded symbols are received, the k original symbols can be recovered. This makes erasure codes an ideal solution for reliable transmission over an unreliable transportation protocol, such as UDP. Also, since any distinct symbol from any upstream nodes can be used for decoding, a receiver does not rely on a specific upstream node for the supply of certain original symbols. This makes erasure codes resilient to node failures.

In addition, the use of erasure source coding mitigates the need for content reconciliation in parallel downloading. By expanding the k original symbols to a larger symbol space of size n, the probability of different nodes holding the same symbols

decreases. However, since the total number of coded symbols is fixed, the problem is not completely solved. We show an example in Fig. 1(a), where S generates 8 coded blocks from 5 original blocks with an $(8, 5)$ erasure code and transmits them to four receivers. It is apparent that t_2 and t_4 still need to reconcile their parallel downloading from upstream nodes t_1, t_3 and t_2, t_3 respectively.

To further address the content reconciliation problem, as well as to provide better resilience to network dynamics, we propose to use *rateless codes* as the foundation of *Outburst*. Rateless codes constitute a category of recently proposed erasure codes, including LT codes [2], Raptor codes [3] and online codes [4]. They are named as "*rateless*" as the number of coded symbols that can be generated from k original symbols is potentially unlimited. Rateless codes are failure-tolerant as they retain the desirable property that the k original symbols are decodable from any slightly more than k coded symbols with high probability. Furthermore, rateless codes possess *two key advantages*, which make them a more suitable solution for overlay content distribution.

1) Efficient encoding and decoding. We briefly illustrate the basic idea in the encoding and decoding process of a rateless code.

Given k input symbols, the basic operation performed by a rateless-code encoder is to exclusive-or a *subset* of the input symbols, which is randomly chosen based on a *special* degree distribution, such as *Robust Soliton distribution* for LT codes. This simple encoding process makes it possible to produce coded blocks on the fly when required. A *decoding graph* that connects coded symbols to input symbols is defined by the encoding process. The encoding information for each coded symbol, *i.e.*, degree and set of neighbors in the decoding graph, is communicated to the receiver for decoding.

In the *Belief-Propagation* (BP) decoder of rateless codes, it constructs the decoding graph when it receives slightly more than k coded symbols and their encoding information. In each round of decoding process, the decoder identifies a coded symbol with degree one, and recovers the value of its unique neighbor among the input symbols. Then the value of the recovered input symbol is exclusive-or'ed to the values of all its neighboring coded symbols, and all the incident edges are removed. Such a process is repeated until all the input symbols are recovered.

As both encoding and decoding only involve exclusive-or operations, rateless codes are very computationally efficient.

2) Better solution towards content reconciliation. The rateless property of rateless codes is useful towards finding a complete solution to the content reconciliation problem. Compared to a traditional erasure code which generates a fixed number of coded symbols from k original symbols, rateless codes can potentially provide a nearly unlimited number of coded symbols to be delivered throughout the network, further decreasing the probability of block conflicts in parallel downloading.

Nevertheless, coding with rateless codes at only the data source may not completely eliminate the need for reconciliation. As shown in Fig. 1(b), t_4 still needs to reconcile its downloading from t_2 and t_3, which inevitably share some common blocks as t_2 downloads from t_3. To completely solve the content reconciliation problem throughout the topology, we propose to generate new coded blocks for a segment on each receiver node, instead of purely relaying the received blocks.

2.2 Recoding with Rateless Codes

The basic idea in *Outburst* is to generate freshly coded blocks at each receiver, so that all the received blocks from any upstream nodes are unique, and useful for decoding at receivers. To this end, we seek to find an efficient recoding scheme at each receiver.

At the first thought, a question to ask is: *Is it possible to directly recode incoming coded blocks of a segment at each receiver, such that the new generated blocks are also useful for decoding at other receivers?* If so, we can employ such direct recoding at each receiver. Unfortunately, with the example of LT codes, we show that the favorable property of efficient decoding is not maintained and the decodability is not guaranteed, if we directly recode received blocks with the same Robust Soliton distribution.

Direct Recoding with LT Codes is not Feasible With an LT code, a segment is encoded with the Robust Soliton distribution. This degree distribution plays a significant role in the success of BP decoding. With it, the probability for a coded block to have a small degree in the decoding graph is high, but the probability quickly decreases as the degree becomes larger. For example, if 10 input blocks are encoded, a coded block has a probability of 0.5 to have degree 2, or a probability of 0.2 to have degree 3.

We show that, if we directly recode the received coded blocks on the same Robust Soliton distribution at a receiver, such a degree distribution is not retained in the decoding graph connecting recoded blocks to original blocks. The expected degree of a recoded block in this decoding graph tends to increase. For an example in Fig. 2, from 8 original blocks, S generates 6 coded blocks to transmit to n_1, and 5 additional blocks to n_2. n_1 directly recodes the 6 received blocks into 5 new blocks to serve t, while n_2 recodes its 5 received blocks to produce 3 new blocks for t. At t, the decoding graph connecting the 8 received blocks to the 8 original blocks is depicted. This graph, with average degree of 3.6, is much denser than that at source S with average degree of 2.5.

Since the desirable Robust Soliton distribution is not retained with direct recoding, it is unlikely that the same superior decoding efficiency of BP decoder in LT codes can be achieved. Further, the decodability with such recoded blocks is not guaranteed with the same high probability as the original LT codes.

Outburst's Recoding Scheme To design a recoding scheme which retains a degree distribution, we investigate another favorable property of rateless codes — the receiver may decode from coded symbols generated by different devices operating a same rateless-code encoder, as long as they are generated from the same set of input symbols [3].

In *Outburst*, the data source encodes blocks of each segment with a rateless code based on a certain special degree distribution, such as the LT code with Robust Soliton Distribution, and transmits the coded blocks. After a receiver receives slightly more than k coded blocks for segment s_i, it decodes and obtains the k original blocks. Upon requests for segment s_i from its downstream nodes, it generates freshly coded blocks from the recovered original blocks, using a rateless-code encoder based on the same degree distribution, and delivers them to these downstream nodes. In what follows, we show that such a recoding process is *correct* and *efficient*.

Correctness. In *Outburst*, the coded blocks a node receives for segment s_i are either encoded by the source or recoded by a receiver, both from the same set of k original blocks of s_i. Since all the encoders follow the same encoding steps and generate each

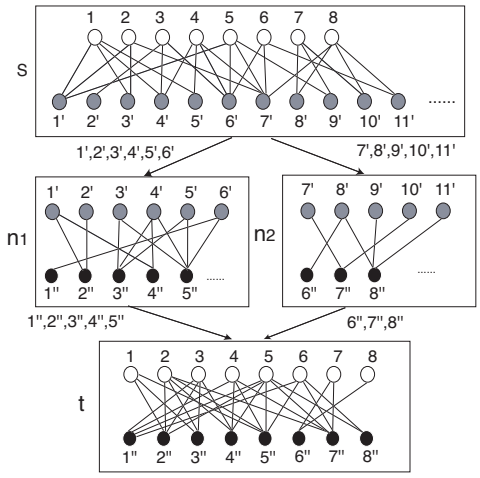

Fig. 2. Direct LT recoding with the Robust Soliton distribution: an example.

block independently from any other one based on the same degree distribution, the coded blocks are all potentially unique as if they are produced by a same encoder. Thus, after collecting slightly more than k coded blocks from any upstream nodes, the receiver can recover the k original blocks with the same high probability as the original code. By guaranteeing the potential uniqueness of all the coded blocks in the network, our recoding scheme successfully eliminates the need for content reconciliation.

In Fig. 1(c), for example, all receivers decode their received blocks and recode them into a potentially unlimited number of coded blocks. Since the blocks in transit are all freshly encoded by their senders, no reconciliation is needed for parallel downloading at any receiver. An analogy is to describe this situation as having many *"mini-sources"* in the overlay, each serving at least one segment with unlimited number of coded blocks.

Efficiency. As previously mentioned, rateless codes are highly efficient with respect to encoding and decoding, which makes it feasible to recode on-the-fly at the receivers. For the example of LT codes, it takes on average $O(\ln(k/\delta))$ block exclusive-or operations to generate a coded block from k input blocks, and $O(k\ln(k/\delta))$ block exclusive-or operations to recover the k original blocks from any $k + O(\sqrt{k}\ln^2(k/\delta))$ of coded blocks with probability $1-\delta$. Each block exclusive-or operation includes L bitwise exclusive-or operations. Even better linear-time encoding and decoding are provided by Raptor codes. For decoding, the decoding graph can actually be constructed on the fly while receiving coded blocks; based on belief propagation, original blocks can be recovered whenever there is enough information to recover it. Thus, our recoding scheme does not introduce much delay and computation overhead, but eliminates the need for content reconciliation required for every parallel retrieval.

3 Outburst: Protocols

We now present practical protocols the source and receivers employ in *Outburst*.

3.1 Baseline Protocols

In *Outburst*, a receiver can essentially choose any available segment to download from any upstream node. Even when it is concurrently downloading coded blocks for a same

segment from multiple upstream nodes, the received blocks can all be used for decoding the segment with high probability. In the practical retrieval protocol design, we consider two problems. First, as a receiver needs to decode a segment before the segment becomes *available* to be recoded and served to other nodes, a potential problem may arise that an upstream node may hold partially coded blocks for many segments at a specific time, but not sufficient coded blocks for recovering a single segment. Second, as the data source and overlay nodes may fail unexpectedly, segment diversity needs to be guaranteed throughout the network for better failure tolerance.

Our strategies to address the above problems are as follows. When a receiver v decides which segment to retrieve from upstream node u, it first checks whether any of the segments it partially holds (which is currently being downloaded from other nodes or has previously been downloaded from a failed node) is available at u. If so, it randomly chooses one such segment and requests it from u; otherwise, it randomly selects an available segment at u and requests its coded blocks.

At the upstream side, when node u receives a request for a specific segment from v, it starts generating coded blocks for the segment and keeps pushing them to v. When the segment is successfully decoded at v, v will send a "stop" message to u to terminate generation and delivery of coded blocks for this segment, and request a new available segment if there exists one.

3.2 Handling Node Departures and Failures

In *Outburst*, upon detecting the departure or failure of an upstream node, a downstream node tries to increase its download bandwidths from the remaining upstream nodes. Appearing intuitive, we note that such simple node dynamics handling — practically compensating the throughput loss from other upstream nodes — is only efficient due to rateless recoding in *Outburst*. As rateless recoding guarantees all coded blocks in the entire overlay are unique, we can rest assured that the compensating download bandwidths are indeed fully utilized to deliver useful blocks for decoding, without the need for reconciliation. Also, our segment retrieval strategy maximizes the diversity of segments in the network and minimizes the chance of holding only partial blocks of a segment in case of node departures or failures. Working together, these measures are able to provide excellent failure tolerance for the content distribution.

4 Performance Evaluation

Our simulations are conducted over random network topologies generated with BRITE [7], based on power-law node degree distributions. The average number of neighbors per node is six. Each node, including the data source, has $1.5 - 4.5$ Mbps of download bandwidth and $0.6 - 0.9$ Mbps of upload bandwidth. Unless otherwise stated, each segment of the data file to be distributed consists of 100 blocks.

4.1 Maximization of Bandwidth Utilization

We first compare *bandwidth efficiency* among four different schemes: source coding (SC) and recoding (RC) with rateless codes, source coding only with rateless codes, source coding with erasure codes ($n/k = 8$), and no coding. Under each scheme, the content blocks are delivered without reconciliation among upstream nodes. For the

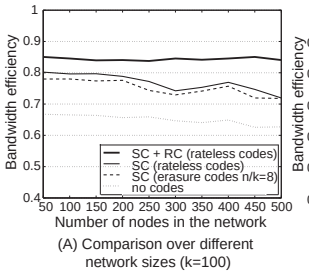

Fig. 3. A comparison of bandwidth efficiency.

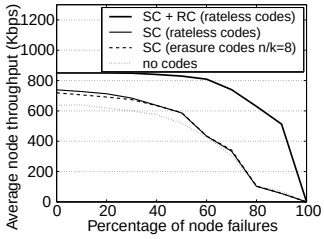

Fig. 4. A comparison of failure tolerance.

scheme without coding, we eliminate duplicated blocks obtained from different upstream nodes and calculate the throughput of distinct content blocks at each receiver; for the schemes with coding, we eliminate duplicated and non-useful coded blocks, decode the content and compute the throughput of decoded original blocks. The bandwidth efficiency of each system is computed as the aggregate throughput at all receivers divided by the total bandwidth consumption.

We can see in all the comparison scenarios shown in Fig. 3, *Outburst*'s rateless source coding and recoding scheme always achieves the highest bandwidth efficiency, as it best eliminates delivery redundancy.

Fig. 3(B) further shows that increasing the number of blocks in each segment helps improving bandwidth efficiency. Nevertheless, the other schemes never outperform *Outburst*, and their bandwidth efficiency becomes stable when k exceeds 100.

4.2 Tolerance against Node Failures

In our next experiments over a $300-$node network, we randomly choose different percentages of nodes to fail concurrently, and calculate the remaining throughput of receiving original content at the remaining receivers. Under all schemes, the receivers perform the same failure handling protocol as discussed in Sec. 3.2. Fig. 4 reveals that the average throughput in *Outburst* remains almost unaffected with failure percentage up to 40%. For the other schemes, their throughput starts to drop whenever failure occurs, and drops faster than that for *Outburst* when failure percentage is high. All these exhibit the excellent failure tolerance of *Outburst*.

5 Related Work

To enhance delivery bandwidth and reliability, mesh-based proposals have become typical in recent overlay content distribution systems [8, 9]. In a mesh overlay, each receiver decides which upstream node to retrieve a specific block from. In *Outburst*, we make every coded block from any upstream nodes equally useful, sparing the receivers from the burden of reconciliation.

As a well-known work on content reconciliation, Byers *et al.* [1] provide algorithms for reconciliation of symbols between node pairs. The algorithms are quite resource intensive as for computation and messaging.

Some existing overlay content distribution proposals advocate erasure codes, *e.g.*, Reed-Solomon codes and Tornado codes, to provide reliability and flexibility [1, 8]. A

more recent work by Maymounkov *et al.* [4] uses online codes, a type of rateless codes. These proposals only encode at the source but do not recode at the receivers, and thus mitigate the need for content reconciliation but do not eliminate it.

For recoding with erasure codes, Byers *et al.* [1] discuss direct recoding of Tornado-code encoded symbols to mitigate delivery redundancy. They advocate heuristics to construct recoding degrees, and do not prove the decodability of recoded symbols.

Network coding has been studied to allow encoding at intermediate nodes in a network [10]. Avalanche [11] is a well-known content distribution scheme with network coding. Similar to *Outburst*, it is robust to node dynamics and reduces delivery redundancy. However, decoding of network coding involves matrix inversions over a finite field up to GF(2^{16}), which is known to be more complex than decoding with only XORs in rateless codes.

6 Conclusion

This paper presents *Outburst*, an excellent solution for efficient content distribution over overlay mesh topologies. Using rateless codes in a novel way — encoding at both the source and the receivers — it effectively battles the fundamental challenges of dynamics, reconciliation, and limited bandwidth in overlay content distribution. With examples, analysis and simulation results, we demonstrate that *Outburst* achieves high bandwidth efficiency and excellent failure tolerance, as compared to traditional schemes with or without erasure codes. The benefits inspire us to further work towards its implementation in realistic large-scale content distribution applications.

References

1. Byers, J., Considine, J., Mitzenmacher, M., Rost, S.: Informed Content Delivery Across Adaptive Overlay Networks. In: Proc. of ACM SIGCOMM 2002. (August 2002)
2. Luby, M.: LT Codes. In: Proc. of the 43rd Symposium on Foundations of Computer Science. (November 2002)
3. Shokrollahi, A.: Raptor Codes. In: Proc. of the IEEE International Symposium on Information Theory. (June 2004)
4. Maymounkov, P., Mazieres, D.: Rateless Codes and Big Downloads. In: Proc. of the 2nd Int. Workshop Peer-to-Peer Systems (IPTPS). (February 2003)
5. Byers, J., Luby, M., Mitzenmacher, M., Rege, A.: A Digital Fountain Approach to Reliable Distribution of Bulk Data. In: Proc. of ACM SIGCOMM 1998. (September 1998)
6. Luby, M., Mitzenmacher, M., Shokrollahi, M., Spielman, D., Stemann, V.: Practical Loss-Resilient Codes. In: Proc. of the 29 th ACM Symp. on Theory of Computing. (1997)
7. Medina, A., Lakhina, A., Matta, I., Byers, J.: BRITE: Boston University Representative Internet Topology Generator. Technical report, http://www.cs.bu.edu/brite (2000)
8. Kostic, D., Rodriguez, A., Albrecht, J., Vahdat, A.: Bullet: High Bandwidth Data Dissemination Using an Overlay Mesh. In: Proc. of the 19th ACM Symposium on Operating Systems Principles (SOSP) 2003. (October 2003)
9. Sherwood, R., Braud, R., Bhattacharjee, B.: Slurpie: A Cooperative Bulk Data Transfer Protocol. In: Proc. of IEEE INFOCOM 2004. (March 2004)
10. Ahlswede, R., Cai, N., Li, S.Y.R., Yeung, R.W.: Network Information Flow. IEEE Transactions on Information Theory **46**(4) (July 2000) 1204–1216
11. Gkantsidis, C., Rodriguez, P.: Network Coding for Large Scale Content Distribution. In: Proc. of IEEE INFOCOM 2005. (March 2005)