A Digital Fountain Approach to Reliable Distribution of Bulk Data

The authors define a “digital fountain” which allows any number of heterogeneous clients to acquire bulk data at any time with optimal efficiency, requiring no feedback data even with high loss rates. The authors also promise to that their protocol using a new erasure code closely approximates this optimal digital fountain. I will be looking for all of these claims to be addressed in the definition of the protocol, and demonstrated with the experiments. Also I would like to see a definition of “optimal efficiency” and a justification that their protocol “closely” approximates this optimal. The authors also clearly state that their erasure codes are “an order of magnitude” faster than standard erasure codes. I am no expert in encoding techniques but I think that the authors have made big claims here. It seems like the authors have claimed many of the things that I have learned to be very cautious of. Terms like optimal, and an order of magnitude seem a little dubious. This is an important paper so they have probably made good on the claims, but I will be checking. I would expect that in the introduction they will attempt to pare back these claims by defining exactly what they mean by optimal, closely approximates, and perhaps defining the conditions under which an order of magnitude performance increase is achieved. This will be an interesting review.

The authors, state that multicast is a natural solution for disseminating large amounts of content to a large number of users simultaneously. This is logical, but perhaps at least a mention of other methods such as CDN would be nice. They make the case for reliability and provide some applications, including database replication. I hadn't thought about db replication. It is obvious now that they mentioned it but still, I give them points for it. They also make the case that users will not all access the data at the same time. The authors make the statement that previous techniques such as polling, local repair, and clever use of hierarchy for providing reliability are inadequate. I don't know if this is true or not, but they did provide references. They make the case that retransmission requests are unworkable in satellite networks because of limited backchannel capacity. This may no longer be true, however, retransmission is unscalable in any case. They cite plenty of previous work on forward error correction and its benefits. They also cite a previous work that takes a hybrid multicast approach by retransmitting redundant codewords. Reed-Solomon codes are heavily cited. They mention that Reed-Solomon codes work well for small block sizes. This is likely going to be the basis for their “order of magnitude” increase in efficiency. The limitations of the “data carousel” approach are discussed.

The approach here is to construct packets such that all of the data can be reconstructed once a receiver has received any subset of packets equal to the length of the original data. The digital fountain idea is not new, the authors cite previous works which use this idea. The authors expect their approach to work well in a wide variety of networks, and they promise to demonstrate this by evaluating their erasure codes under a wide variety of packet loss rates. Finally, they promise to describe Tornado codes. There is no reference here so at this point I am assuming that this is their contribution.

The authors have done a good job of describing previous work and motivated the need for their new codes. Also they have provided a basis for their “order of magnitude” improvement. They will show that their codes are an order of magnitude more efficient than Reed-Solomon with large data blocks. This is reasonable since Reed-Solomon is known to provide poor performance with large data blocks.

In section 2 the authors define the “optimal” protocol. This has 4 characteristics. 1) reliability. 2) Efficiency, it should take no more time than an individual download. 3) On demand. Users should be able to start their download at any time. 4) Tolerant, the protocol should allow for a variety of end-to-end packet loss and data rates. The authors also state their expectations of network characteristics. The key being that all of the networks they are considering have limited back channel capacity. At this

point I feel pretty good about the authors ability to make good on their promises from the abstract. They have defined optimal for me, and have provided a reasonable basis for their claim of an order of magnitude performance increase. Very nice.

Sections 3 and 4 are used to define the digital fountain, and to describe in abstract how to build one. A file of k packets is encoded into n packets and transmitted repeatedly. The stretch factor n/k limits the extent to which the code can correct for. A typical value of n is 2k which in practice limits the amount of redundant transmissions. The key inefficiency here is that the Reed-Solomon codes are computationally intensive. The authors use of tornado codes reduces this computational burden at the cost of some extra packets. This tradeoff is the focus of the experiments in the paper.

Section 5.1 provides some basic theory for Reed-Solomon, and Tornado codes. This is very helpful for the reader such as myself, with no experience in encoding. It appears that (of course) a new Tornado code itself is not the contribution, but instead the application of tornado codes to the problem and the study of the tradeoffs between Reed-Solomon, and Tornado codes are the contribution of this work. Someone with encoding experience would probably have already known this. Reed-Solomon codes generally take k packets of data and encode it into n=2k packets. The n packets are transmitted repeatedly, and as soon as a receiver has k packets they may decode the original data. The matrix used to encode/decode is dense and the computation is intense. Tornado codes, on the other require slightly more than k packets to be received before decoding can occur. However, the decoding is done with simple xor, and substitution operations. Clearly for large data blocks the Tornado code will be an order of magnitude faster.

The results provided in the section on performance support the authors claims of an order of magnitude efficiency increase for larger block sizes. The graph used to generate the codes was created randomly and not specially chosen. The author mention that they used multiple runs when generating their results in tables 2 & 3, however, there is no mention of how many seeds were used to generate each result. This information along with confidence intervals would have been nice to have. In figure two, the authors show that the average inefficiency for tornado is 1.0536, and the maximum is 1.1 with a standard deviation of 0.0073. This seems very close to optimal. I can see now why the authors made the near-optimal claim in the abstract.

The simulation experiments performed compare the tornado codes to an interleaving technique. The problem with interleaving is that packets for bins already filled may arrive while waiting for the last few packets required to decode the final bins. This is a source of inefficiency. The simulation experiments are designed to answer the following two questions: 1) With k chosen such that the decoding inefficiency (packets more than k required for decoding) is comparable to the tornado code, how does the decoding time compare. 2) With k chosen such that the decoding time is comparable to the tornado code, how does the decoding inefficiency (packets more than k required for decoding) compare.

The simulations use a uniform distribution of loss. This is unrealistic, however for the tornado code it does not matter because any k + l packets received allow decoding of the whole file. However, with interleaving the loss pattern could have some effect. The authors think that it would make interleaving perform better. However, in any case, they promised to do a study using a large number of different loss models. In light of this I would feel cheated if they simply said all loss patterns are equivalent to uniform and left it at that. The authors also use trace driven results so their claim is satisfied.

In order to equate decoding times between tornado z and interleaving codes the authors first determine

through simulation the maximum interleaving block size to match the decoding efficiency of 1.076 for tornado. Based on the decoding times produced for the Cauchy (Reed Solomon) codes the decoding time is approximated.

The decoding inefficiency for a 1MB file is plotted with uniform packet losses of 10 and 50% over a range of 1 to 1000 receivers.

The tornado code is more efficient than interleaving for all values in the range with both packet loss models. The tornado code is more efficient than the interleaving code with similar decoding efficiency in all cases from 1 to 50% packet loss. I wonder how this would change with different processor speeds. However, this would have been difficult for the authors to determine. I think that faster processesors would favor the interleaving codes because the tornado codes are using xor & substitution. I think that xor is performed in hardware so the speedup would be linear to the increase in processor speed. The interleaving code however, would benefit from additional improvements such as better pipelining, etc.

I would like to have known the block size that was determined and once again it would be nice to see confidence intervals as well as to know the number of seeds.

Next the authors fix the decoding time and plot the decoding efficiency. They plot a graph at k=20 (roughly equivalent to tornado decoding time) and k=50 which is a little higher. The authors state that this is a reasonable value, however, I don't know enough about encoding to determine if this is true. The simulation is plotted for uniform loss of 10 and 50% with 10% representing the upper limit for loss on the wired internet and 50% for wireless. This seems reasonable, however, the modern wired Internet is probably a little less. A 1MB file was encoded over 2MB of packets and transmitted to a number of receivers ranging from 1 to 10,000. The graph for 10% packet loss is not very exciting. It seems that tornado z and interleaving codes have similar efficiencies. At 50% loss rate and a 10,000 receivers tornado z looks like it has nearly twice the efficiency of the interleaving code. The graph is a worst case or maximum plot. I would prefer to see confidence intervals. I suspect that the difference would be less dramatic in this case. Also as above, I wonder how modern processors would affect these graphs.

The graphs plotted against file size are a lot more interesting. The tornado codes start to become a lot more efficient than the interleaver code as the file size grows. This is interesting. Also since modern files can be very large, it would be interesting to see the difference for file sizes in the GB rather than megabytes.

The authors perform simulations using trace data with bursty loss patterns. The average loss in these traces was 11% and the results are very similar to the uniform loss pattern at 10% loss. This supports the authors conjecture that loss patterns make little difference and only the amount of loss is important.

In section 7.1 the authors describe a layered multicast system based on previous work. The sender in this scheme provides multiple layers of varying bandwidths and the receivers choose which layer to join. Joins are done at synchronization points and bursts are provided by the sender to allow the receiver to determine which layer to join. If a burst is received with no packet loss then the receiver joins a higher layer at the next synchronization point. Conversely if a receiver experiences packet loss then it will join a lower bandwidths layer. Synchronization points are provided more frequently for lower bandwidth channels. The key feature of this congestion control mechanism is that it does not require communication from the receiver to the sender. This allows it to function in networks with

limited or no backchannel capacity.

This scheme causes an additional source of inefficiency by introducing the possibility of receiving duplicate packets. The authors quantify this distinctness inefficiency and perform experiments using both the layered approach and a single multicast group at a fixed rate. There were four layers in the layered multicast protocol used in the experiment and the server had one unicast control channel. The server and clients were located on different subnets at three universities. The graphs are plotted showing efficiencies against packet loss. For the single layer approach with packet loss below 50% the distinctness efficiencies are almost always 1. Also at even higher loss rates the tornado code appears to be robust with an inefficiency of 1.4k packets at 70% loss. With the layered approach distinctness inefficiency increases (because of switching between layers) then at higher loss rates it becomes low again because most receivers are at the base rate and stay there.

The authors conclude by suggesting additional applications that may use the digital fountain. The first is disseminating routing data. The idea is that routing data can be sent along multiple paths and then the receiver can recover the data when a sufficient number of packets arrive (regardless of when they arrive). The second is mirroring applications. The receiver can get packets from multiple sources and aggregate them. Once it has a sufficient number of packets it can recover the data.

This work is interesting but I think that modern processors probably change the range of acceptable parameters. It is likely that the decoding times have become a lot closer to each other than they were with 1998 era processing power. Still the idea of a digital fountain is interesting and may have applications within modern networks.

The authors didn't use confidence intervals and in one case they presented only maximum values. I think the paper would have been better if this had been done. Also they only reported the number of seeds used in some of the experiments, not all of them. However, these are minor issues. The paper is very well written and a significant piece of work.