L R I

SWITCHED CODE

KADI N / AL AGHA K

Unité Mixte de Recherche 8623
CNRS-Université Paris Sud – LRI

09/2009

Rapport de Recherche N° 1527

CNRS – Université de Paris Sud
Centre d'Orsay
LABORATOIRE DE RECHERCHE EN INFORMATIQUE
Bâtiment 490
91405 ORSAY Cedex (France)

Switched Code

Nour KADI & Khaldoun Al AGHA
LRI, Université Paris-Sud 11

1 Abstract

We present and analyze a novel degree distribution that outperform Robust Soliton distribution, used in LT code [1], when the source symbols are distributed over the network.

2 Binary Exponential Distribution

Definition 1. *(codeword and degree): A codeword is the result of XORing multiple source symbols. These source symbols are called the coding candididates. The number of coding candidates is called the degree of a codeword.*

Definition 2. *The Binary Exponential Distribution BED_k is given by*

- $\varphi(d) = \frac{1}{2^d}$, *For all* $d = 1, 2 \dots, k-1$
- $\varphi(k) = \frac{1}{2^{k-1}}$

where k represent the total number of source symbols.

Lemma 1. *For any $k > 0$, BED_k is a probability distribution*

Proof.

$$\sum_{d=1}^{k} \varphi(d) = \sum_{1}^{k-1} \frac{1}{2^d} + \frac{1}{2^{k-1}} = 2 * (1 - \frac{1}{2^k}) - 1 + \frac{1}{2^{k-1}} = 1$$

□

Definition 3. *(decoding probability): Let $D_{(r|d)}$ be the probability to recover the $(r)^{th}$ source symbol when decoding a codeword of degree d.Or, in other words, it is the probability to decode a codeword of degree d when $r-1$ of the source symbols has been recovered.*

Proposition 1.

$$D_{(r|d)} = \left\{ \begin{array}{lll} (k-r+1)/k & for & d=1 \\ \frac{d.(k-r+1).\prod_{i=0}^{d-2}(r-1-i)}{\prod_{i=0}^{d-1}(k-i)} & for & d=2,3,\ldots,r \\ 0 & if & d>r \end{array} \right\}$$

Proof. The destination, which has recovered $r-1$ symbols, is able to decode a received codeword of degree d if $(d-1)$ of the coding candidates are among the $r-1$ recovered symbols and only one candidate is among the $(k-r+1)$ uncovered symbol. So the decoding probability is

$$D_{(r|d)} = \frac{(k-r+1)\binom{r-1}{d-1}}{\binom{k}{d}} = \frac{(k-r+1).\frac{\prod_{i=0}^{d-2}(r-1-i)}{(d-1)!}}{\frac{\prod_{i=0}^{d-1}(k-i)}{d!}} = \frac{d.(k-r+1).\prod_{i=0}^{d-2}(r-1-i)}{\prod_{i=0}^{d-1}(k-i)}$$

In the case where $d > r$, certainly more than one coding candidates are among the uncovered symbols and hence it couldn't be decoded. □

Definition 4. *(symbol recover probability): Let R_r be the probability to recover the r^{th} source symbol. So $R_r = \sum_{i=1}^{k} p(i).D_{(r|i)}$.*

Definition 5. *Let E_y be The ecpected number of recovered symbols after sending y codewords. And the overhead $\Theta = Y - k$ where $E_Y = k$*

Our interest is to maximize E_y, $\forall y \leq k$ and at the same time to minimize Θ as possible. Or in other word, we want to maximize the symbol recover probability R_r, $\forall r \leq k$.

Lemma 2. *To recover the first symbol, it is more useful to use binary exponential distribution (BED) than using robust soliton distribution.*

Proof. Before sending any codeword, the number of recovered symbols $r-1=0$. So only a codeword of degree 1 could be decoded at this stage. The expected number of recovered symbols after sending the first codeword $E_1 = p(d=1) * D_{(1|1)} = p(d=1) * 1$. Let $p(d=1) = \frac{1}{2}$ is the probability to get an encoded symbol of degree 1 when using enxponential distribution and $p'(d=1) = (\frac{1}{k} + \frac{R}{k})/\beta$ represents the same probability but when using robust soliton distribution where $\beta = \sum_{i=1}^{\frac{k}{R}-1} \frac{R}{i} + R.ln(\frac{R}{\delta}) \leq 1 + \frac{R}{k}(\mathrm{H}(\frac{k}{R}) + \frac{1}{k}ln(\frac{R}{\delta})$ and $R = c.ln(\frac{k}{\delta})\sqrt{k}$. We will prove that $p(d=1) > p'(d=1)$ by contradiction.

Let's assume that

$$(\frac{1}{k} + \frac{R}{k})/\beta > \frac{1}{2}$$

$$\frac{1+R}{k + R(\mathrm{H}(\frac{k}{R}) + ln((\frac{R}{\delta})} > \frac{1}{2}$$

$$k + R[\mathrm{H}(\frac{k}{R}) + ln(\frac{R}{\delta}) - 2] < 2$$

But this is impossible $\forall k > 1$. Therefore using BED increases the expected number of recovered symbol when sending the first encoded symbol. □

Lemma 3. *To recover the last symbol, it is more useful to use soliton distribution.*

Proof. We will prove the lemma for ideal soliton distribution and the results follows for robust distribution by using the result in [1] which finds that the release probability for robust solition is superior to the release probability for ideal soliton.

In this case $r = k$. From proposition 1, $D_{(k|d)} = \frac{d}{k}$. Lets compare between the symbol recover probability for both distribution. When using binary exponential distribution

$$R_k = \frac{1}{k}\sum_{d=1}^{k} \frac{d}{2^d} = \frac{2}{k}.[1 - (\frac{1}{2})^{k+1} - \frac{k+1}{2^k}]$$

and when using Soliton Distribution

$$R'_k = \frac{1}{k}[\frac{1}{k} + \sum_{d=2}^{k} \frac{1}{d-1}] = \frac{1}{k}[\frac{1}{k} + \mathrm{H}(k-1)]$$

$$R'_k - R_k = \frac{1}{k}.[\frac{1}{k} + \mathrm{H}(k-1) - 2 + \frac{1}{2^k} + \frac{2k+2}{2^k}] = \frac{1}{k}.[\frac{1}{k} + \mathrm{H}(k-1) + \frac{2k+3}{2^k} - 2]$$

We will prove that $R'_k - R_k > 0$ by contradiction.

Lets assume that $R'_k - R_k < 0$. Which means

$$\frac{1}{k} + \mathrm{H}(k-1) + \frac{2k+3}{2^k} - 2 < 0$$

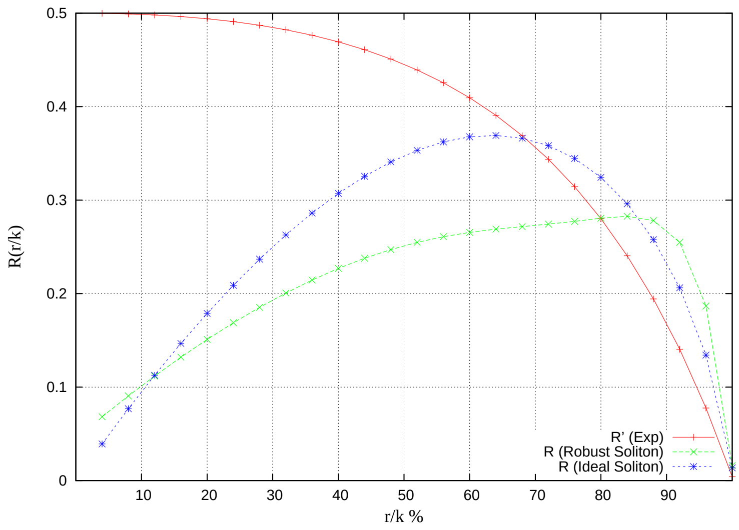

Figure 1: Symbol Recover Probability for Soliton distribution and binary exponential distribution

$$\frac{1}{k} + \mathrm{H}(k-1) + \frac{2k+3}{2^k} < 2$$

However $\mathrm{H}(k-1) > 2$ for $k \geq 5$ and also for $1 \leq k \leq 4$ this formula gives a result greater than 2. Hense our assumption was false which means that $R'_k > R_k$. □

Lemma 4. *Soliton distribution outperform binary exponential distribution only after recovering* 68% *of the overall source packets.*

Proof. If R_r, R'_r and R''_r be the symbol recover probability when using Ideal Soliton distribution, Robust Soliton distribution and binary exponential distribution respectively. Then we have

$$\begin{aligned} R_r &= \frac{k-r+1}{k^2} + \sum_{d=2}^{r} \frac{1}{(d-1)} \frac{(k-r+1).\prod_{i=0}^{d-2}(r-1-i)}{\prod_{i=0}^{d-1}(k-i)} \\ &= (k-r+1)[\frac{1}{k^2} + \sum_{d=2}^{r} \frac{1}{(d-1)} \frac{\prod_{i=0}^{d-2}(r-1-i)}{\prod_{i=0}^{d-1}(k-i)}] \end{aligned}$$

$$R'_r = \frac{k-r+1}{\beta}\Big[\frac{R+1}{K^2} + \sum_{d=2}^{\frac{k}{R}-1}(\frac{1}{d-1} + \frac{R}{k}).\frac{\prod_{i=0}^{d-2}(r-1-i)}{\prod_{i=0}^{d-1}(k-i)}$$

$$
\begin{aligned}
&+(\frac{R^2}{k(k-R)}+\frac{Rln(\frac{R}{\delta})}{k}).\frac{d.\prod_{i=0}^{d-2}(r-1-i)}{\prod_{i=0}^{d-1}(k-i)}\\
&+\sum_{\frac{k}{R}+1}^{k}\frac{1}{(d-1)}\frac{\prod_{i=0}^{d-2}(r-1-i)}{\prod_{i=0}^{d-1}(k-i)}\Bigg]\\
R''_r \;&=\; (k-r+1).\sum_{d=1}^{r}\frac{d}{2^d}\frac{\prod_{i=0}^{d-2}(r-1-i)}{\prod_{i=0}^{d-1}(k-i)}
\end{aligned}
$$

If we plot these three functions, we get the curves shown in fig 1. From this figure we see that when the number of recovered symbol is inferior to 68% of k then the recover probability of binary exponential distribution is superior to that of Soliton distribution and this is reversed as the number of recovered symbols increases. □

Now we have to know how many encoded packets should we use in order to recover 68% of the source packets using binary exponential distribution.

Definition 6. *Lets consider a new decoder S. If the decoder S receives r codewords then it decoeds them in assending order of their degree. a codeword which is considered by S for the first time will be droped and it will not be considered for later decoding. It is clear that the number of recovered symbols using our normal decoder, which keeps the codewords for later decoding, will be greater than if we use the decoder S.*

Proposition 2. *When using the decoder S with binary exponential distribution, The expected number of recovered symbols after sending k codewords is at least* $0.70 * k$

Proof. As mentioned earlier that decoder S decode the codewords sequentially depending on their degree. Assume that S receives Y codewords

- Step 1: For $d = 1$, all codewords of degree 1 are decoded and so their coding candidates are recovered. So the expected number of degree 1 codewords that could be decoded are $E_Y[1] = Y \times \varphi(d = 1) \times 1 = \frac{Y}{2}$

- Step 2: For $d = 2$, the expected number of degree 2 codewords that could be decoded are $E_Y[2] = \frac{Y}{4} * D_{((E_Y[1]+1)|2)}$

- Step 3: For $d = 3$, the expected number of degree 3 codewords that could be decoded are $E_Y[3] = \frac{Y}{8} * D_{((E_Y[1]+E_Y[2]+1)|3)}$

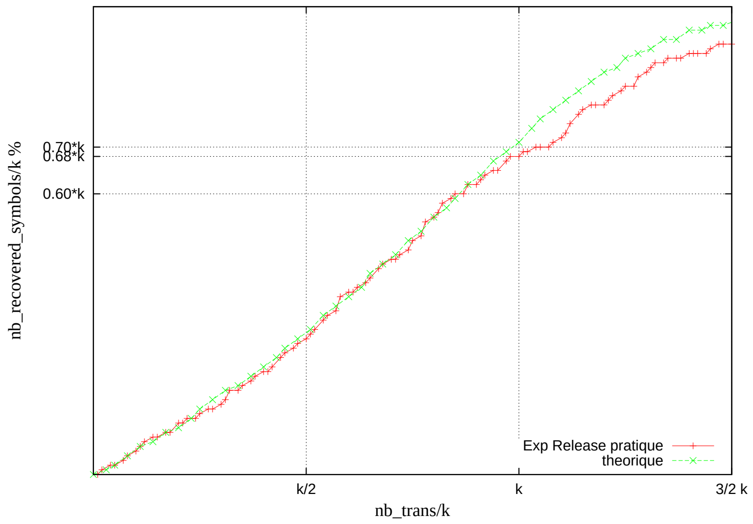

Figure 2: Number of released symbols at the destination in fnction of the number of the received codewords where $k = 1000$ for the pratique curve. The axis x represents Y which is the number of transmited codewords and the axis y represents the number of recovered symbol or E_y

- Step i: For $d = i$, ,the expected number of degree i codewords that could be decoded are $E_Y[i] = \frac{Y}{2^i} * D_{((\sum_{j=1}^{i-1} E_Y[j]+1)|i)}$

Then the total expected number of recovered symbol after sending Y codewords are $E_Y = \sum_{i=1}^{k} E_Y[i]$

$$E_Y = \frac{Y}{2} + \sum_{i=2}^{k} Y \times \varphi(d=i) \times D_{((\sum_{j=1}^{i-1} E_Y[j]+1)|i)}$$

$$= \frac{Y}{2} + \sum_{i=2}^{k} \frac{Y}{2^i} \frac{i.(k-\sum_{j=1}^{i-1} E_Y[j]).\prod_{q=0}^{i-2}(\sum_{j=1}^{i-1} E_Y[j]-q)}{\prod_{q=0}^{i-1}(k-q)}$$

We plot this function in figure 2 and we see that when $Y > k$ then the number of recovered symbols are superior to $0.70k$. Also to confirm this formula we plot the simulation result. In this simulation we propose that there is one source which has k symbols and which send codewords to a distination. The destination receives each codeword sent by the source. The destination decodes the codewords in the fly. So we plot each time the relation between the number of sent codeword and the number of recovered symbol. We see that the simulation curve is very close from the theorique result. □

3 Shifted Robust Soliton Distribution

This is a novel distribution which was proposed in [2] in order to adapt LT code to the case where some input symbols are already known at the receiver. In this case it is more useful to send encoding symbols with higher degree as the input symbols which are available at the receiver play the role of singletons and insure the existence of the ripple. This distribution is given by

$$\gamma_{k,n}(d) = 0 + \mu_{k-n}(d') \quad for \quad round\left(\frac{d'}{1-\frac{n}{k}}\right) = d$$

where k represents the total number of input symbols and n represents the number of input symbols already know at the decoder. The authors define the overhead of this distribution by $\left[n + O\left(\sqrt{k-n}\; ln^2\left(\frac{k-n}{\delta}\right)\right)\right]$

4 Switched Code

As mentioned earlier that our goal is to find a distribution which increases the symbol recovery probability at any time during the decoding process while keeping the overhead as small as possible. This characteristic is important when an itermediate node should decode the source symbols in order to reforward them after re-encoding. The new distribution could release enough symbols to be re-encoded even when small number of encoded symbols have been received.

In order to acheive this goal, we propose the switched distribution. The idea of this distribution is to switch from one distribution to another in function to the number of encoded symbols which have been sent. Following our previous analysis we see that the new distribution should start with the binary exponential distribution and then switch to robust soliton distribution after sending $\frac{5}{4}k$ encoded symbols where k is the total number of source symbols. In order to evaluate this distribution, we simulate a basic scenario that consists of one source and one distination and an ideal communication environment without any loss. Figure 3 shows that integrating ExpD with

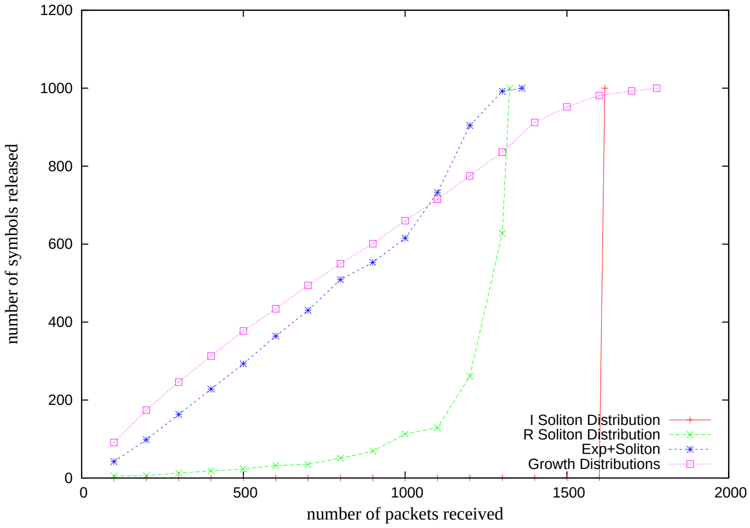

Figure 3: Number of recovered symbols at the destination in fnction of the number of the received codewords where N=1000

RSD improves the possibility of decoding even when few encoded packets are sent. Howevere, figure 4 shows that even this new proposed distribution decreases the overhead in comparing with growth code [3] but it is still a little bit higher than RS. In order to overcome this drawback we take into acount the folwoing remark. After sending k encoded symbol using BED, we are pretty sure that the destination has recovered at least 60% of the source symbols. So, as mentioned in [2], if we use shifted distribution instead of robust soliton distribution we can reduce the overhead.

Switched distribution could be defined as follow:

$$\varpi_{i,k}(d) = \begin{cases} \varphi_k(d) & \text{for } i < k \\ \gamma_{k,0.6k}(d) & \text{for } i \geq k \end{cases}$$

Where

$$\varphi(d) = \begin{cases} \frac{1}{2^d} & \text{for } d = 1, 2 \ldots, k-1 \\ \frac{1}{2^{k-1}} & \text{for } d = k \end{cases}$$

$$\gamma_{k,n}(d) = 0 + \mu_{k-n}(d') \quad for \quad round\left(\frac{d'}{1-\frac{n}{k}}\right) = d$$

and $\mu_k(d)$ is the robust soliton distribution.

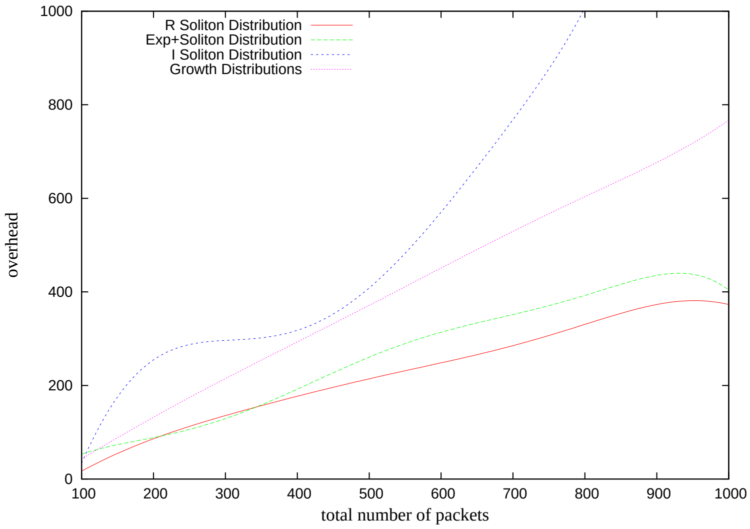

Figure 4: Overhead from different value of k

So the source node generates the i^{th} encoded symbols according to the distribution $\varpi_{i,k}(d)$ where k is the number of source symbols available at the node and this number could vary with time.

Lemma 5. *A decoder needs*

$$K' = 1.4k + O(\sqrt{0.4k}\; ln^2(\frac{0.4k}{\delta})$$

encoding symbols under switched code to decode all k input symbols with probability at least $1-\delta$.

5 simulation

we use the simulation and compare our distribution with RSD, ISD [1] and growth code [3]. We simulate the case of one source S which has k packets that want to send to a destination D. We don't consider the packet loss and we assume that each transmitted packet is received by the destination. For a fair comparaison we choose the parameters of Robust Soliton c and δ to be 0.2 and 0.1 respectivelty in order to give a small overhead as suggested in [4]. Figure 5 shows that switched code acheives the decoding progressively which

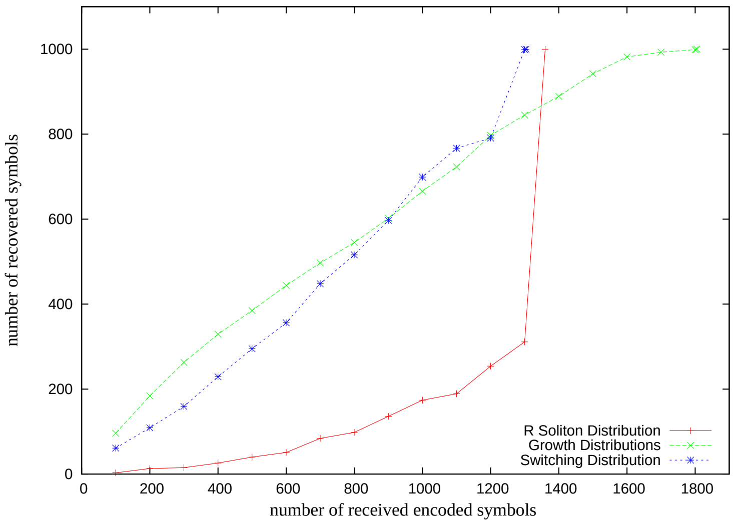

Figure 5: Number of recovered symbols at the destination in fnction of the number of the received codewords where $k = 1000$

means that even with small number of encoded symbols, switched code can recover a relatively high number of source symbols while we see that RSD recover a very few symbols at the begining and then it has a big jump when the decoding process approaches to the end. if we compare switched code with growth code, we see that the last recovers a little bit more symbols at the begining which is logical as growth code sends a large number of singleton at the begining but this is reversed quickly as we exceed a certain number of transmissions. More over it is clear from Figure 5 that switched code retrives all source symbols using fewer number of encoded symbols than other distribution. Figure 6 shows that the overhead acheived by switched code is about 60% lower than growth code. Moreover switched code reduces the overhead of RSD by nearly 32%

In fig7 we show the changing in the size of the encoded buffer during the simulation. The encoded buffer is used at the destination to keep the encoded packets that could not be decoded immediately. It is clear that our distribution decreases about 70% of the buffer size in comparing with Soliton distribution. This is because Soliton distributions give higher degree than ExpD but at the begining of the simulation the destination doesn't have too many native packets to use them on the decoding process. So the destination has to keep so much encoded packet in its buffer. Growth code reduces the

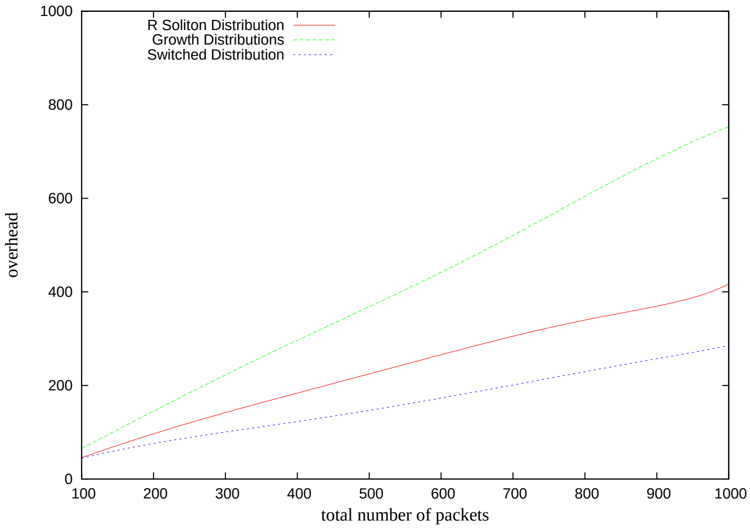

Figure 6: Overhead from different value of k

buffer size as thay send alot of native packets at the begining but this can increase the overhead as we have seen before.

References

[1] M. Luby. LT codes. Proceedings of The 43rd Annual IEEE Symposium on Foundations of Computer Science, November 16-19 2002, pp.271-282, 2002.

[2] S. Agarwal, A. Hagedorn and A. Trachtenberg, Adaptive rateless coding under partial information, Information Theory and Applications Workshop, UCSD, San Diego, USA, 2008

[3] Abhinav Kamra, Jon Feldman, Vishal Misra and Dan Rubenstein, Growth Codes: Maximizing Sensor Network Data Persistence, Proceedings of ACM Sigcomm, Pisa, Italy, September, 2006

[4] Information Theory, Inference, and Learning Algorithms: Published by Cambridge University Press (2003). Chapter 50.

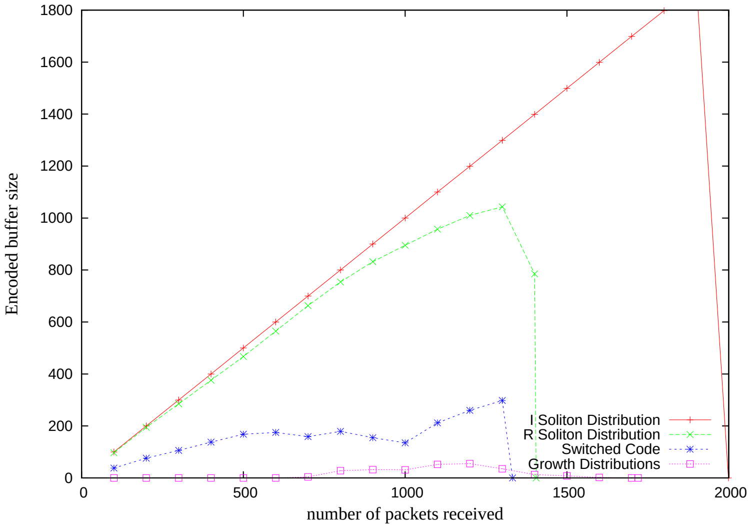

Figure 7: Size of the Encoded-buffer during the simulation