

Performance of QVidium™ Advanced FEC versus Pro-MPEG Forum's Interleaved Parity FEC

Overview

Forward Error Correction (FEC) potentially provides a low-latency method for correcting packet loss. In contrast with other error correction and recovery mechanisms, such as automatic repeat request (ARQ), FEC does not require feedback or an upstream channel, and can immediately recover lost packets without having to wait for feedback from the source. This is convenient for long latency links and communications connections without an upstream channel, such as satellite links.

Various forms of FEC have long been applied to digital audio-video streams, most notably for satellite transmission and more recently for Internet streaming, to help minimize the adverse impact of channel impairments on the audio-video signal. Advantages of FEC over other error correction mechanisms include scalability to large systems and its inherent multicast compatibility.

FEC augments a media stream with redundant data, called checksum packets, to help restore stream integrity based upon anticipated levels of packet loss. FEC groups data packets into an FEC block and generates checksum packets for each FEC block. A complete FEC block includes all the data packets for that block plus all the checksum packets for that block. The checksum packets generated from a given block are said to *cover* that block since missing data packets can be restored by combining the remaining checksum and data packets in that block. FEC *coverage* is the number of missing data packets that FEC can recover within a block. FEC algorithms generally limit the number of recoverable data packets in an FEC block to the number of checksum packets within that block.

FEC algorithms generally determine the number of checksum packets with which to augment each data block as a fixed percentage of the number of data packets in a block. Because packet loss is limited to the number of checksum packets, the ability for FEC to

recover lost packets is therefore directly proportional to the additional bandwidth overhead allotted for FEC.

Basic FEC Implementations: Pro-MPEG Forum's Code of Practice

A relatively simple and well-known implementation of FEC for packet-switched transport is Parity-FEC, based upon the Internet Engineering Task Forces' (IETF) RFC 2733 [1]. Recently, the Pro-Mpeg Forum, a video transport standards body, adopted FEC implementations based upon RFC 2733 for the transport of video streams over IP networks. RFC 2733 computes the Exclusive-OR operation across corresponding bits of all data packets within an FEC block to create a single checksum packet for that block, called a parity packet. The complete FEC block has the property that the Exclusive-OR across all packets of that block, data and parity, yields a packet with all zeros in all bit locations. Any single bit error in any packet of the block would show up as a "one" bit when computing the Exclusive-OR. More importantly, if a receiver lost only a single packet within an FEC block, taking the Exclusive-OR of all received packets along with the parity packet would reproduce the lost packet. Through this technique, a system implementing Parity-FEC can recover any single lost packet within an FEC block.

In order to protect against large contiguous losses of packets, called a burst-drop, RFC 2733 includes a packet interleaving mechanism for grouping FEC blocks. In this implementation of interleaving, a transmitter sequentially fills each row of a two-dimensional matrix with copies of the outgoing data packets. When the last data packet fills the last data row of the matrix, the FEC engine computes a final checksum row, generating one parity packet for each column to fill the checksum row, and then sends the entire checksum row as a burst of parity packets.

At the receiver, interleaving introduces a processing delay equal to the time required to fill the entire receiver's matrix. The receiver waits for the last data and parity packet within a block to arrive before it applies the received parity packets to the received block of data packets to recover any missing packets. If the last packet in a block was lost, then either a timeout, the appearance of a packet from a following block, or a combination of both may force FEC immediate processing for the current FEC block. Interleaving

introduces processing jitter both at the transmitter and at the receiver as a result of the periodic processing time in waiting to fill the interleave matrix.

The Pro-MPEG Code of Practice #3.1 (COP #3 Release 1) [2] uses this interleaving method to create a parity row and can therefore protect against a single burst-drop of packets up to the number of columns in the matrix. However, this technique also introduces an additional FEC processing latency in the stream equal to the time it takes to generate and receive all the packets in the matrix.

Furthermore, this technique cannot handle arbitrary patterns of packet loss with the data block. For example, an FEC generator generated N parity packets for N columns of a matrix, and if outside of a burst-drop of $N-M$ packets, a receiver were to lose an arbitrary M packets elsewhere in the block, then such a interleaved parity FEC approach would generally fail to recover all N dropped packets, even though there were N parity packets for the FEC block, because there likely would be more than a single packet loss per matrix column. Yet, this approach only generates a single parity packet per column and can therefore only recover a single packet loss per column.

To mitigate this problem, the Pro-MPEG Forum recently introduced a new Code of Practice #3.2 (COP #3 Release 2), which augments the last row of column parity packets with a last column of parity packets for each row. Thus, whereas COP #3.1 generates N parity packets for an M -row by N -column matrix, COP #3.2 generates $N+M$ parity packets and can protect up to a theoretical maximum of $N+M-1$ packet drops. However, as with COP #3.1, COP #3.2 is still very limited in the number of packet loss patterns it can recover.

Drawbacks and Limitations of Parity FEC and Pro-Mpeg Forum FEC

Parity FEC in general and the interleaved implementations of RFC 2733 and the Pro-MPEG Forum in particular suffer from a number of serious limitations. Parity FEC can only recover a single lost packet per block. Thus, block size must be constrained to a small number of packets in order to obtain a reasonable recovery rate. However packets loss often occurs in bursts, so these FEC implementations must resort to interleaving to cover multiple contiguous packet losses.

Interleaving brings with it its own set of constraints. Interleaving adds additional latency equal to the total number of packets in the interleaving matrix. Each row and column in the matrix must be the same so that the block sizes can remain constant. Furthermore because of the additional complexity of interleaving, Pro-MPEG FEC constrains the matrix size to remain constant throughout an entire video session. Because an entire matrix must fill before sending the parity packets, and because only constant bit rate (CBR) streams can guarantee a constant matrix fill rate, Pro-MPEG FEC works only with CBR streams, despite the fact that video compression is more efficient under variable bit rate (VBR) compression. (Pro-MPEG FEC is also only specified for MPEG-2 transport streams.)

Lastly, but perhaps the most serious constraint is that interleaved parity implementations protect against only a small percentage of the total possible packet loss patterns. Thus, as we show later in the Performance Comparison section of this report, interleaved parity FEC has a high likelihood of having at least one unrecoverable packet when multiple packets in a block are lost, even through the total number of lost packets in a block may be less than or equal to the number of parity packets for the interleaved FEC block.

Description of QVidium™ Advanced FEC

To address these limitations, and produce a more robust forward error correction, we developed a more sophisticated, patent-pending FEC implementation called QVidium™ Advanced FEC, or QVidium-FEC. QVidium-FEC is able to generate multiple checksum packets without the need to interleave the data and recover from any pattern of packet losses in an FEC block, up to the number of checksum packets generated for that block. This results in a significantly more bandwidth efficient, flexible, and effective forward error correction algorithm. QVidium-FEC places no limitations on either block size or number of checksum packets per block (up to the number of data packets per block).

As long as the number of lost packets in a block is less than or equal to the number of checksum packets in the block, QVidium-FEC recovers 100% of all packet loss. QVidium-FEC places no limitations on data packet length among packets within a block.

Furthermore, FEC block size and the number of checksum packets per block can vary dynamically from one block to the next. QVidium-FEC works with any data stream type and can work with either CBR or VBR streams. Furthermore, QVidium-FEC can take advantage of the specific sensitivity to the playable recovered stream for different content types to optimize the recovered playable frame rate for media streams. If desired, QVidium-FEC can also be used with interleaving, where it adds the capability for generating multiple checksum packets per column or row and can therefore recover multiple lost packets per interleaved column or row.

Performance Comparison

Historically, systems designers have incorporated interleaved parity FEC to help improve system resilience in communications channels that may experience large burst drops. Yet, in general a channel often exhibits random combinations of packet and bit errors in addition to random occurrences of contiguous burst drops. In this section we have characterized the behavior of COP #3.1 and COP #3.2 as two common implementations of interleaved parity FEC to help quantify the performance characteristics of interleaved parity FEC in comparison to QVidium-FEC without interleaving.

We first look at all combinations of patterns of packet loss for increasing sets of interleaving matrices. For each combination of packet loss, we determine the number of packets that interleaved parity FEC implementations COP #3.1 and COP #3.2 can recover. We then tabulate and plot the number of unrecoverable packets as a function of the number of columns of the interleaving matrix in Tables 1 and 2 and Figures 1 and 2, and compare with QVidium-FEC, which, under the same conditions, would recover 100% of the lost packets.

The tables in Appendix A show the results of applying interleaved Pro-MPEG Column-only FEC (COP #3.1), and soon-to-be-released interleaved Pro-MPEG Row/Column FEC (COP #3.2) to all possible combinations of loss patterns from 1 to the number of parity packets in the interleaved FEC block. The last row in the tables of Appendix A shows the accumulation of the results from all combinations of loss patterns.

Tables 1 and 2 contain the last row of the tables of Appendix A. Appendix A shows several different FEC block sizes, varying from a 3-column by 3-row matrix up to a 6-column x 5-row matrix. More real-world FEC block sizes, such as 10-columns by 10-rows require excessive simulation time to calculate. Therefore, we use the results of Tables 1 and 2 from smaller matrices up to 6 x 5 to produce graphs to infer the unrecoverability percentage of dropped packets.

Number Columns	Block Size	Number Combinations	Unrecovered Packets	Dropped Packets	COP 3.1 % Recovered	QVidium % Recovered
3	9	129	135	333	59.46	100.0
4	16	2,516	4,432	9,216	51.91	100.0
5	25	68,405	168,875	323,775	47.84	100.0
6	30	768,211	2,345,700	4,397,880	46.66	100.0

Table 1. ProMPEG COP #3.1, Column Interleave FEC: Percentage Recovered Packets

Number Checksums	Block Size	Number Combinations	Unrecovered Packets	Dropped Packets	COP 3.2 % Recovered	QVidium % Recovered
6	9	465	576	1,971	70.78	100.0
8	16	39,202	110,976	262,144	57.67	100.0
10	25	7,119,515	32,303,300	64,478,250	49.90	100.0
11	30	107,636,401	577,278,150	1,095,586,680	47.31	100.0

Table 2. ProMPEG COP #3.2, Row/Column Interleave FEC: Percent Recovered

These tables tabulate the total number of packet loss patterns under the Number Combinations column for simulated packet losses of up to the theoretical maximum number of recoverable packets. This maximum possible number of recoverable packets is limited to the number of parity packets generated in the matrix, which is equal to the Number Columns column in Table 1 for COP #3.1. For COP #3.2, the number of parity checksum packets equals the sum of the number of columns plus the number of rows in the matrix, and is shown under the Number Checksums column for COP #3.2 in Table 2. These loss patterns simulate the loss of one or more packets, up to the number of parity packets in the matrix. The Dropped Packets column lists the total number of simulated dropped packets corresponding to the Number Combinations column of these tables. Unrecovered Packets is the number of packets that the FEC implementation could not recover when presented with all Number Combinations of packet loss patterns. Finally,

the last column, Percent Unrecovered, is simply the ratio of the number of Unrecovered Packets to the total number of Dropped Packets.

The data in these tables show COP #3.1 to be as effective as COP #3.2. This is because the number of dropped packets increases geometrically, as per the increase in the number of combinations of loss patterns. For example a 6-column by 5-row interleaved parity FEC, having 6 parity packets as per COP #3.1 FEC, has 593,775 combinations of packet loss totaling 4,397,880 dropped packets. Of these, 53.34% of these packets are recoverable under COP #3.1, while all the packets would have been recovered under QVidium-FEC.

The same dimensional block for COP #3.2 FEC has almost twice the number of parity packets, so that it has a much larger number of potential combinations of dropped packets. Specifically, from 11 parity packets and all combinations of up to 11 lost packets, there are a total of 1,095,586,680 dropped packets. However of these, only 577,278,150 of the packets, or 52.69% are unrecoverable.

From these tables and graphs we infer that in larger, more practical FEC block sizes, such as 10-columns by 10-rows, the effectiveness in either protection scheme, COP #3.1 or COP #3.2, will also be less than 50 percent. Again, QVidium-FEC is able to recover all lost packets in these patterns. Of course, combinations of more packet losses are less likely to occur in practice than combinations having fewer packet losses, perhaps making Pro-MEPG FEC more practical than may be inferred. However, it is clear that looking over all packet loss patterns, COP #3.2 does not provide an overall efficiency improvement over COP #3.1.

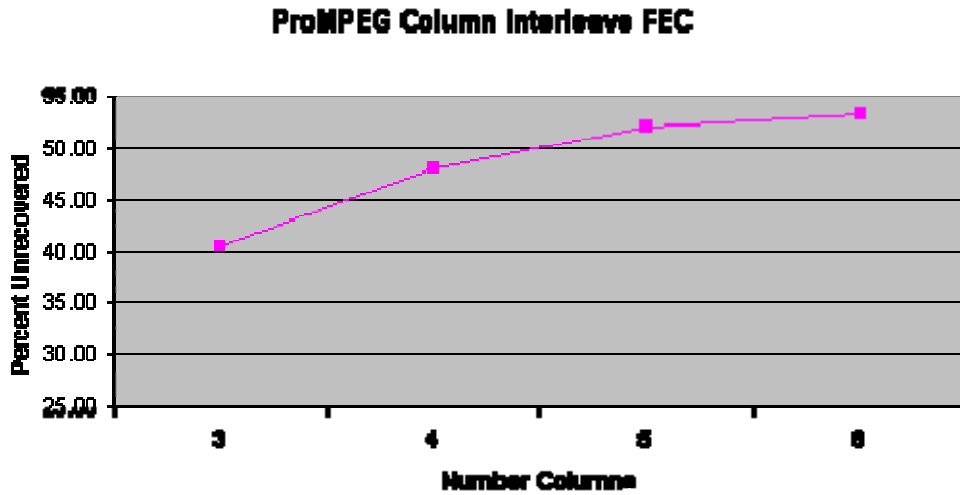


Figure 1. ProMPEG COP #3.1, Column Interleave FEC: Percentage Unrecovered Packets

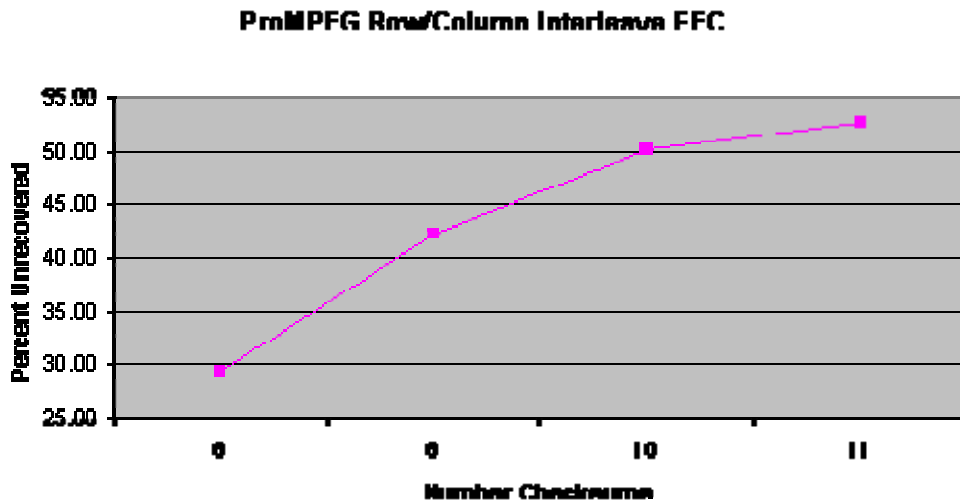


Figure 2. ProMPEG COP #3.2, Row/Column Interleave FEC: Percent Unrecovered

Table 3 shows tabulates the number of combinations of packet loss patterns having the packet loss equal to the number of parity packets and the number of these patterns where all the lost packets can be recovered. The last column shows the ratio of these recovered patterns to the total number of combinations as a percentage. This last column clearly shows that the percentage of fully recoverable packet loss patterns diminishes exponentially with increasing block size.

Number Columns	Block Size	Number Combinations	Full Recovery	Recovery Percent
3	9	84	27	32.14
4	16	1,820	256	14.07
5	25	53,130	3,125	5.88
6	36	1,947,792	46,656	2.40
7	49	85,900,584	823,543	0.96
8	64	4.43E+09	1.68E+07	0.38
9	81	2.61E+11	3.87E+08	0.15
10	100	1.73E+13	1.00E+10	0.06

Table 3. PromPEG COP#3.1 Full Recovery

For all the loss patterns shown in all the above tables, QVidium-FEC would have resulted in 100% packet recovery (no packet loss). These results highlight the fact that interleaved parity FEC provides diminishing performance in comparison to QVidium-FEC.

In particular, media file transport over satellite where the links have long latencies (on the order of seconds) and the upstream channel has very limited capacity often employ FEC as a practical way to improve transport efficiency without increasing transmission delay. In many such systems, 100% packet recovery is a requirement. Where an upstream channel exists, automatic repeat request (ARQ) mechanisms provide a means for achieving 100% recovery in the face of dynamic and unpredictable packet loss conditions. However, the long latencies quickly multiply with each ARQ request. Augmenting ARQ with FEC can provide a means to eliminate some of the upstream retransmission requests and can thereby drastically reduce the delays for 100% file transfer completion.

In such systems the incorporation of new, more efficient FEC techniques, such as QVidium-FEC, can overcome the residual packet losses inherent in interleaved parity FEC and help to achieve the goal of minimizing retransmission requests to minimize file transfer times. Furthermore in contrast to interleaved FEC implementations, QVidium-FEC can dynamically change block and checksum size to quickly adapt to changing network conditions.

Summary

QVidium Technologies, Inc. has developed a new patent-pending FEC technology called QVidium-FEC. QVidium-FEC overcomes many of the problems associated with other FEC implementations, most notably common limitations associated with interleaved parity FEC implementations such as RFC 2733 and Pro-MPEG FEC. Results of simulations confirm the improvements seen by implementations of QVidium-FEC for packet recovery. Furthermore, considering all combinations of packet loss patterns, COP #3.2 does not provide an overall efficiency improvement over COP #3.1.

Legal Notice

The information contained in this document represents the current view of QVidium Technologies, Inc. (“QVidium”) on the issues discussed as of the date of publication. Because QVidium must respond to changing market conditions, it should not be interpreted to be a commitment on the part of QVidium, and QVidium cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. QVIDIUM MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of QVidium Technologies, Inc.

QVidium may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from QVidium, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2005 QVidium Technologies, Inc. All rights reserved.

Appendix A.

K = Number data packets dropped each FEC block
 A.FEC = Packets dropped after FEC applied
 B.FEC = Packets dropped before FEC applied
 Drp % = (A.FEC / B.FEC) percent
 Columns 5-7 are accumative statistics.
 bRowColumn: false means COP#3.1, true means COP#3.2
 Recover All K counts combinations where all packets are recoverable
 Recover percent = (#Combinations / Recover All K) percent

PromPEG C(9,1-3): 1.29e+002, nColumns: 3, nRows: 3, bRowColumn: false)

K	#Combin- ations	A.FEC Dropped	B.FEC Dropped	Drp %	A.FEC AccDrp	B.FEC AccDrp	Drp %	Recover All K	Recover Percent
1	9	0	9	0.00	0	9	0.00	9	100.00
2	36	18	72	25.00	18	81	22.22	27	75.00
3	84	117	252	46.43	135	333	40.54	27	32.14

PromPEG C(9,1-6): 4.65e+002, nColumns: 3, nRows: 3, bRowColumn: true)

K	#Combin- ations	A.FEC Dropped	B.FEC Dropped	Drp %	A.FEC AccDrp	B.FEC AccDrp	Drp %	Recover All K	Recover Percent
1	9	0	9	0.00	0	9	0.00	9	100.00
2	36	0	72	0.00	0	81	0.00	36	100.00
3	84	0	252	0.00	0	333	0.00	84	100.00
4	126	36	504	7.14	36	837	4.30	117	92.86
5	126	180	630	28.57	216	1467	14.72	81	64.29
6	84	360	504	71.43	576	1971	29.22	0	0.00

PromPEG C(16,1-4): 2.52e+003, nColumns: 4, nRows: 4, bRowColumn: false)

K	#Combin- ations	A.FEC Dropped	B.FEC Dropped	Drp %	A.FEC AccDrp	B.FEC AccDrp	Drp %	Recover All K	Recover Percent
1	16	0	16	0.00	0	16	0.00	16	100.00
2	120	48	240	20.00	48	256	18.75	96	80.00
3	560	624	1680	37.14	672	1936	34.71	256	45.71
4	1820	3760	7280	51.65	4432	9216	48.09	256	14.07

PromPEG C(16,1-8): 3.92e+004, nColumns: 4, nRows: 4, bRowColumn: true)

K	#Combin- ations	A.FEC Dropped	B.FEC Dropped	Drp %	A.FEC AccDrp	B.FEC AccDrp	Drp %	Recover All K	Recover Percent
1	16	0	16	0.00	0	16	0.00	16	100.00
2	120	0	240	0.00	0	256	0.00	120	100.00
3	560	0	1680	0.00	0	1936	0.00	560	100.00
4	1820	144	7280	1.98	144	9216	1.56	1784	98.02
5	4368	1728	21840	7.91	1872	31056	6.03	3936	90.11
6	8008	9792	48048	20.38	11664	79104	14.75	5632	70.33
7	11440	32544	80080	40.64	44208	159184	27.77	4096	35.80
8	12870	66768	102960	64.85	110976	262144	42.33	0	0.00

PromPEG C(25,1-5): 6.84e+004, nColumns: 5, nRows: 5, bRowColumn: false)

K	#Combin- ations	A.FEC Dropped	B.FEC Dropped	Drp %	A.FEC AccDrp	B.FEC AccDrp	Drp %	Recover All K	Recover Percent
1	25	0	25	0.00	0	25	0.00	25	100.00
2	300	100	600	16.67	100	625	16.00	250	83.33
3	2300	2150	6900	31.16	2250	7525	29.90	1250	54.35
4	12650	22100	50600	43.68	24350	58125	41.89	3125	24.70
5	53130	144525	265650	54.40	168875	323775	52.16	3125	5.88

PromPEG C(25,1-10): 7.12e+006, nColumns: 5, nRows: 5, bRowColumn: true)

K	#Combin- ations	A.FEC Dropped	B.FEC Dropped	Drp %	A.FEC AccDrp	B.FEC AccDrp	Drp %	Recover All K	Recover Percent
1	25	0	25	0.00	0	25	0.00	25	100.00
2	300	0	600	0.00	0	625	0.00	300	100.00

3	2300	0	6900	0.00	0	7525	0.00	2300	100.00
4	12650	400	50600	0.79	400	58125	0.69	12550	99.21
5	53130	8400	265650	3.16	8800	323775	2.72	51030	96.05
6	177100	86400	1062600	8.13	95200	1386375	6.87	155900	88.03
7	480700	565000	3364900	16.79	660200	4751275	13.90	347500	72.29
8	1081575	2578400	8652600	29.80	3238600	13403875	24.16	515625	47.67
9	2042975	8506900	18386775	46.27	11745500	31790650	36.95	390625	19.12
10	3268760	20557800	32687600	62.89	32303300	64478250	50.10	0	0.00

PromPEG C(30,1-6): 7.68e+005, nColumns: 6, nRows: 5, bRowColumn: false)

K	#Combin- ations	A.FEC Dropped	B.FEC Dropped	Drp %	A.FEC AccDrp	B.FEC AccDrp	Drp %	Recover All K	Recover Percent
1	30	0	30	0.00	0	30	0.00	30	100.00
2	435	120	870	13.79	120	900	13.33	375	86.21
3	4060	3180	12180	26.11	3300	13080	25.23	2500	61.58
4	27405	40620	109620	37.06	43920	122700	35.79	9375	34.21
5	142506	333030	712530	46.74	376950	835230	45.13	18750	13.16
6	593775	1968750	3562650	55.26	2345700	4397880	53.34	15625	2.63

PromPEG C(30,1-11): 1.08e+008, nColumns: 6, nRows: 5, bRowColumn: true)

K	#Combin- ations	A.FEC Dropped	B.FEC Dropped	Drp %	A.FEC AccDrp	B.FEC AccDrp	Drp %	Recover All K	Recover Percent
1	30	0	30	0.00	0	30	0.00	30	100.00
2	435	0	870	0.00	0	900	0.00	435	100.00
3	4060	0	12180	0.00	0	13080	0.00	4060	100.00
4	27405	600	109620	0.55	600	122700	0.49	27255	99.45
5	142506	15600	712530	2.19	16200	835230	1.94	138606	97.26
6	593775	200100	3562650	5.62	216300	4397880	4.92	544525	91.71
7	2035800	1657200	14250600	11.63	1873500	18648480	10.05	1641000	80.61
8	5852925	9799800	46823400	20.93	11673300	65471880	17.83	3645000	62.28
9	14307150	43272600	128764350	33.61	54945900	194236230	28.29	5400000	37.74
10	30045015	145523850	300450150	48.44	200469750	494686380	40.52	4050000	13.48
11	54627300	376808400	600900300	62.71	577278150	1095586680	52.69	0	0.00