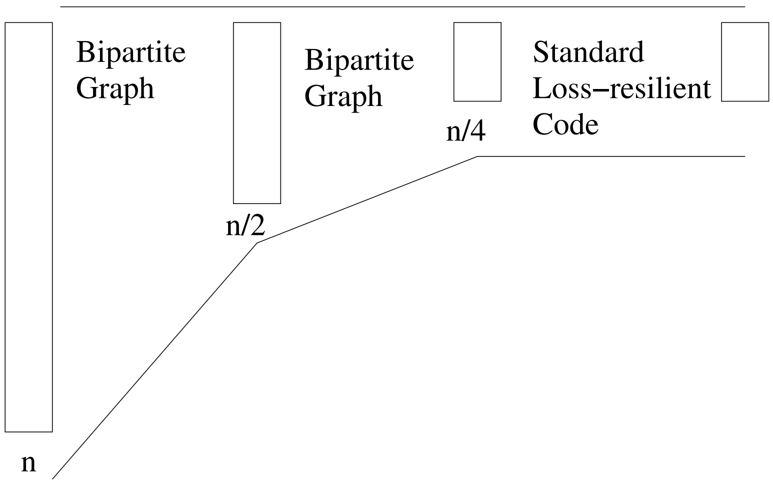

Figure 1: Encoding structure of tornado code

CS 493: Algorithms for Massive Data Sets	**Tornado Codes and Error-Correcting Codes**
3/7/02	**Scribe: Ming Zhang**

1 Reception efficiency

Suppose the length of original message is n, the length of encoded message is cn, the length of received message is rec, reception efficiency is defined as the ratio of packets in message to packets needed to decode: n/rec. Optimally, when reception efficiency is 1, the original message can be decoded from any n words of encoding.

Practically, we want to decode from any $(1+\varepsilon)n$ words of encoding. Accordingly, the reception efficiency is $1/(1+\varepsilon)$.

2 Regular graph

In tornado code, a message of length n is encoded to length cn with $c = 2$ in several steps. As shown in figure 1, in each step, the length is shrinked by a factor of $\beta = 1 - 1/c$. The goal is to recover the message from close to β erasures.

We first considered using 3-6 regular graph for the design and analysis of tornado codes. As shown in figure 2, suppose the probability of packet not recovered in previous step is x, and in current step is y, they satisfy $y = a(1-(1-x)^5)^2$. For the decoding to proceed, we need $y < x$ for all $0 < x < a$ ($a < 0.43$). The performance of regular graph is shown in figure 3, the maximum reception efficiency is achieved when left degree is round 3.

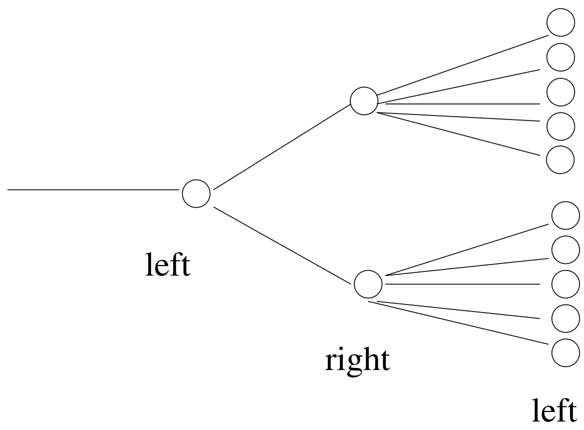

Figure 2: 3-6 Regular Graph Analysis

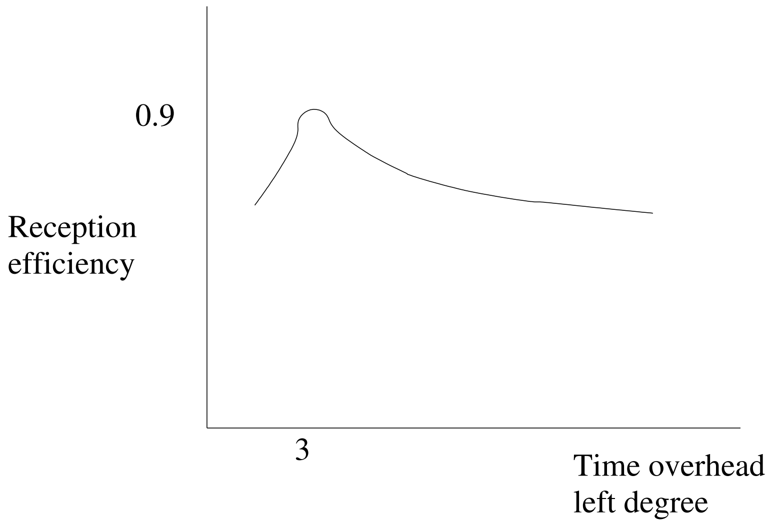

Figure 3: Regular graph performance

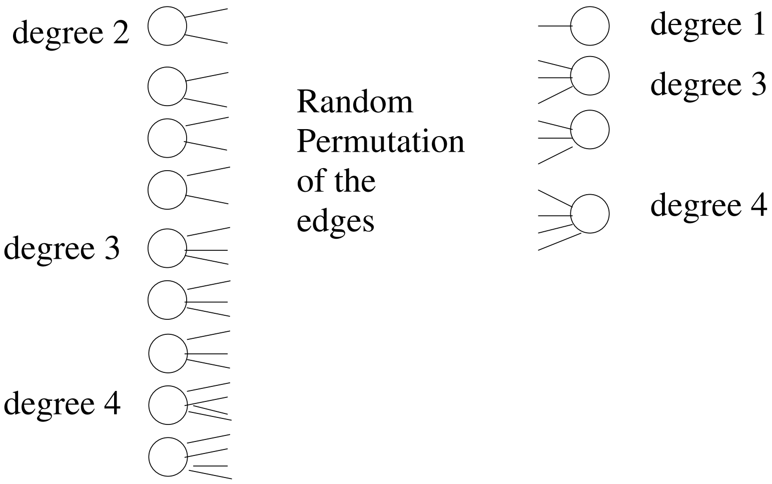

Figure 4: Irregular graphs

The performance of regular graphs are not very good. Because the right degree is $2d$, so Pr[right degree = 1] = $1/2^{2d-1}$. This implies the expected number of left nodes with neighbor of degree 1 is $d/2^{2d-1}$.

3 Irregular graph

Using irregular graph can achieve a better performance than using regular graph. An example of irregular graph is shown in figure 4.

Let l_i be the fraction of edges of degree i on the left in the original graph, r_i be the fraction of edges of degree i on the right in the original graph. We define the degree sequence functions as:

$l(x) = \sum l_i x^{i-1}$ and $r(x) = \sum r_i x^{i-1}$

Similar to the analysis of regular graphs in figure 2, suppose the probability of packet not recovered in previous step is x, and in current step is y, they satisfy $y = a \times l(1 - r(1 - x))$ and for decoding to proceed, we want $y < x$ for all $0 < x < a$.

A good left degree sequence is got from truncated heavy tail distribution and right degree sequence is got from Poisson distribution:

$l_i = \frac{1}{H(D)(i-1)}$ $(i = 2, 3, \ldots, D+1)$ and $r_i = \frac{e^{-a} a^{i-1}}{(i-1)!}$

$H(D)$ is the harmony function. Let a_l and a_r be the average node degree on the left and right, then:

$a_l = \frac{H(D)(D+1)}{D} \approx ln(D)$ and $a_r = \frac{ae^a}{e^a - 1} \approx ln(D)/b$

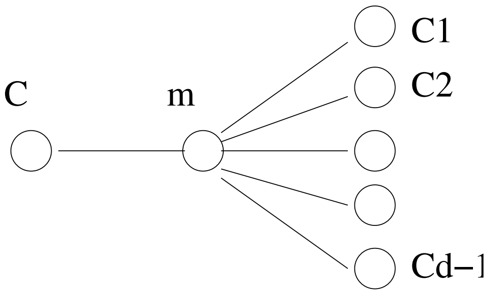

Figure 5: Decoding

They satisfy $a_l = ba_r$. From l_i and r_i, we know that

$$l(x) = \frac{-ln(1-x)}{H(D)} \text{ and } r(x) = e^{\frac{(D+1)H(D)}{bD}(x-1)}$$

So, $r(1-x) = e^{-\frac{D+1}{D}\frac{H(D)}{b}x}$, and for decoding to proceed, it must satisfies:

$$y = al(1-r(1-x)) = \frac{-\frac{D+1}{D}\frac{H(D)}{b}x}{H(D)} < x \text{ for all } 0 < x < a => a < \frac{D}{D+1}b$$

Because the average right degree is $2ln(D)$, Pr[right degree=1]$\sim 1/D$. So the expected number of neighbors of degree 1 is 1. From this, it's easy to see that irregular graph will have better performance than regular graph.

4 Error-correcting codes

In error-correcting codes, the check bit is computed as the XOR of its incident message bit. The decoding proceeds in *rounds* and can be described as a cascading series of bipartite graphs as in figure 5. One layer is corrected each time by assumming previous layers corrected. The belief propagation works as follows:

- From my other check bits, I believe I am ...
- From my other message bits, I believe I am ...

With belief propagation, we want to reduce the number errors to a very small fraction. We use the following strategy:

- message bit m will send received bit to check bit c unless all other check bits say otherwise
- check bit c sends to message bit m the XOR of the values it received in this round from its adjacent message bits

Let d_l be the degree of message nodes and d_r be the degree of check nodes. With probability p a message node receives the wrong bit. Let p_i be the probability that message bit sends check bit a wrong value in round i. Initially, $p_0 = p$.

We can define a recursive equation describing the evolution of p_i over a constant number of rounds. Consider the end of the *ith* round, and the probability that a check bit receives an even number errors is:

$$\frac{1+(1-2p_i)^{d_r-1}}{2}$$

the probability that message bit is received wrong but sent correctly in round $i+1$ is:

$$p_0[\frac{1+(1-2p_i)^{d_r-1}}{2}]^{d_{l-1}}$$

the probability that message bit is received correctly but sent wrong in round $i+1$ is:

$$(1-p_0)[\frac{1-(1-2p_i)^{d_r-1}}{2}]^{d_{l-1}}$$

This gives an equation for p_{i+1} interms of p_i:

$$p_{i+1} = p_0 - p_0[\frac{1+(1-2p_i)^{d_r-1}}{2}]^{d_{l-1}} + (1-p_0)[\frac{1-(1-2p_i)^{d_r-1}}{2}]^{d_{l-1}}$$

We want for any $\varepsilon > 0$, the fraction of incorrect edges p_i can be reduced to ε in a constant number of rounds.

5 Reference

A digital fountain solution to reliable multicast of bulk data, John W. Byers, Michael Luby, Michael Mitzenmacher and Ashutosh Rege, proceedings of ACM SIGCOMM, 1998.