The Viceroy Network

Algorithms for Modern Communication Networks

Andreas Barthels

October 12, 2008

Contents

1 Introduction

This paper contains the written composition of my Sarntal talk about the Viceroy peer-to-peer network.

Viceroy was designed with the goal of improving peer-to-peer network performance over previous approaches. Previously, the linkage cost on the peers in the network was

- not constant (Chord), or,
- not evenly distributed (Binary Trees), or,
- the network itself did not feature a well scaling lookup complexity (CAN).

Viceroy focuses on avoiding bottlenecks and minimizing the load of the peers while still yielding good lookup times.

The next section comprises a brief discussion about network design and the typical characteristics. Afterwards, the Viceroy network structure is described and evaluated.

2 Network Design

When it comes to network design, there are certain network characteristics which can be optimized.

2.1 Network Characteristics

The basic characteristics of a network can be described using the network degree and diameter.
Definition 1 (Network Degree deg)**.** *The Network Degree deg is given by the maximum number of outgoing links out of a single peer in the network.*
Definition 2 (Network Diameter dia)**.** *The Network Diameter dia is given by the longest of all shortest distances between two peers.*

Having defined these two terms, the next subsection analyses how they are related and defines a design goal for optimal networks.

2.2 Trade-Off in Network Design

In network design, there is a trade-off in between the network degree and the network diameter. Within a distance d of a peer there are at most deg^d peers reachable. It immediately follows, that the following unequation holds for the number of peers n:

$$deg^{dia} \geq n$$

by taking the logarithm on both sides, dividing by $\log(deg)$ and using $deg > 1$, we have

$$\Rightarrow dia \geq \frac{\log(n)}{\log(deg)}.$$

The optimum in terms of single peer load is found, when we assume $deg = \text{const}$, and still have $dia \in O(\log(n))$. In the following section, we will take a closer look at the Viceroy network structure that reaches this optimum.

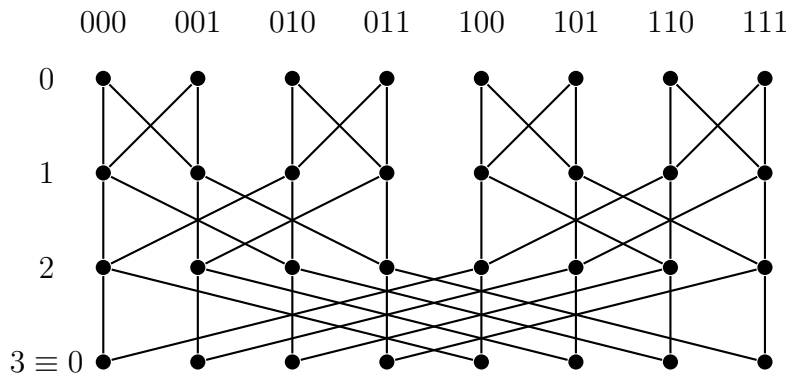


Figure 1: Butterfly-Network of dimension 3.

3 Viceroy Network Structure

Viceroy uses the concept of Butterfly-Networks which is presented in the following subsection.

3.1 Butterfly-Networks

Definition 3 (Butterfly-Networks)**.** *Butterfly-Networks of dimension k are defined as a Graph $BF(k) = (V_k, E_k)$, with*

$$\begin{aligned} V_k &= \{(i,u) \mid 0 \leq i < k, u \in \{0,1\}^k\} \\ E_k &= \{((i,u),(i+1 \mod k, u)) \mid 0 \leq i < k\} \cup \\ &\quad \{((i,u),(i+1 \mod k, \mathcal{F}(u,i))) \mid 0 \leq i < k\} \end{aligned}$$

using the bit-swapping function $\mathcal{F}(u,i)$

$$\mathcal{F}(u,i) \mapsto u \oplus (1 << i).$$

To illustrate this definition, $BF(3)$ is depicted in figure 1. Since one can "correct" one bit of u to a target v on each traversal of a level, routing into the correct column takes at most $O(k)$ steps. Furthermore, routing to the correct target level, again takes $O(k)$ steps. Routing is thus in total $O(k)$. In relation to the number of nodes n, we can express

$$\begin{aligned} n &= k2^k \\ \Rightarrow \log(n) &= \log(k) + k\log(2) \\ \Leftrightarrow k &= \frac{\log(n) - \log(k)}{\log(2)} \\ \Rightarrow k &< \log(n). \end{aligned}$$

Considering the constant degree of the network, routing is optimal $O(\log(n))$.

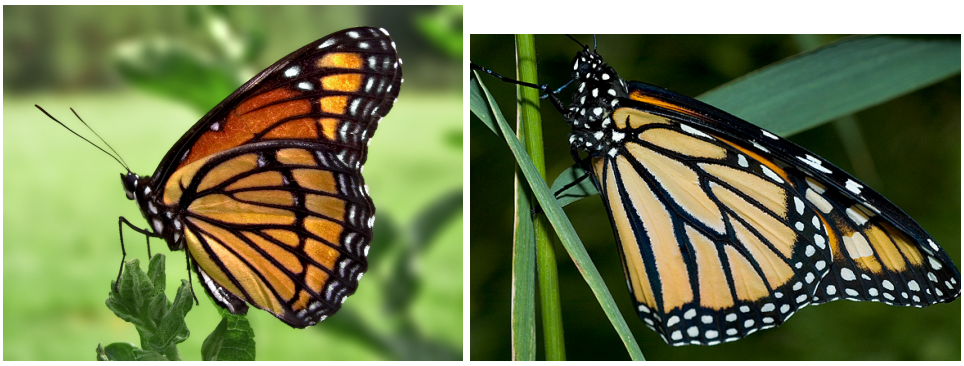


Figure 2: The Viceroy and Monarch Butterflies.
Copyright Namek Piccolo and April M. King. Taken from Wikipedia.

3.2 Viceroy: Emulation of Butterfly Networks

In this section, the Viceroy network structure is described. Before starting with this, a few words on the background of why the name "Viceroy" was chosen are given.

Viceroy is a butterfly, that looks very similar to the so called Monarch butterfly, as can be seen in figure 2. In the same way that the butterflies are similar, one can say, that the viceroy network is similar to butterfly networks.

There are several reasons, why one can not use butterfly networks straight away. The first is, the node count in a butterfly network is not flexible. Each dimension demands its own number of peers. The other is the need for a global coordination of the peers in establishing the butterfly structure.

Viceroy implements a 1-dimensional distributed hashtable. Keys are mapped to $[0, 1)$. Data is assigned to the clockwise-closest successor, as can be seen in figure 3.

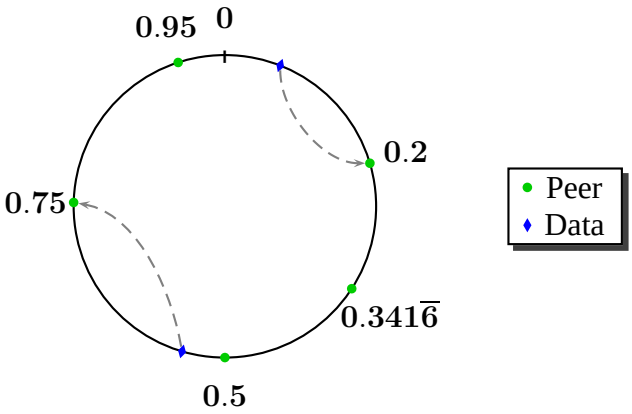
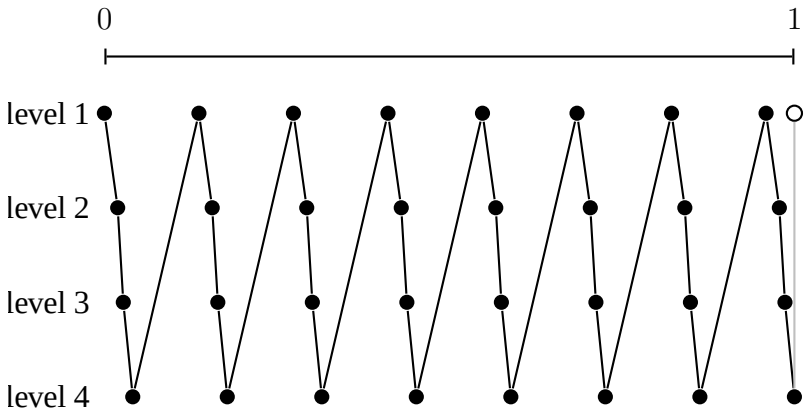
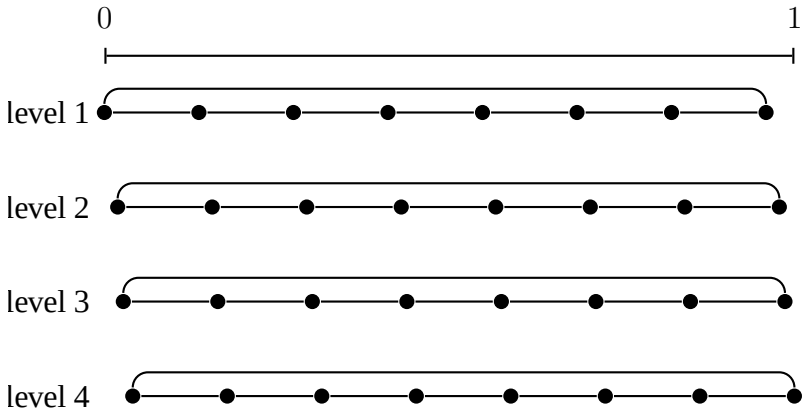


Figure 3: Onedimensional distributed Hashtable

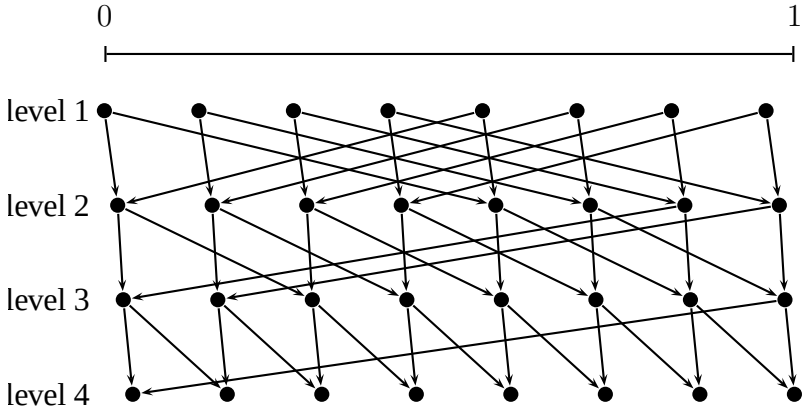
The butterfly network structure is approximated locally for peer s through creating links to ...



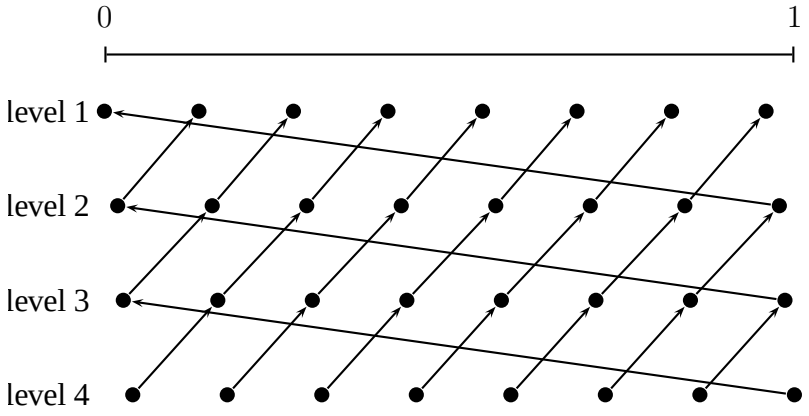
... the first successor and predecessor of s on the Interval $[0, 1)$, regardless of the level ...



... the first successor and predecessor of s on $s.level$ ...



... the clockwise-closest peer on $s.level + 1$ to $s.position$ and to $(s.position + (1/2)^{level}) \mod 1$ ...



... the clockwise-closest peer on level $s.level - 1$, if $s.level > 0$.

$\Longrightarrow$ Viceroy features a constant network degree of 7.

4 Performance Evaluation

This chapter deals with a brief performance evaluation of the Viceroy network. Giving all the performance details including their proof would exceed this work, so the reader must be refered to the literature ([MNR02] and [MS07]).

In order analyze the worst case performance, it is necessary to analyze the maximum level first.

4.1 Maximum Level

The level selection process is triggered locally in each node whenever the distance to the successing node changes. It is assumed, that the position of each peer is uniformly distributed. Let n denote the number of peers in the network. An estimation of the number of peers $\hat{n}$ can then be obtained through calculating

$$\hat{n} = \frac{1}{\text{distance to the next node}}.$$

In order to give boundaries for the estimation that hold with high probability, we first need the following lemma:

Lemma 1. *The length of stretches in between two nodes $\frac{1}{\hat{n}}$ is within the following bounds with high probability:*

$$\frac{1}{n^{c'}} \leq \frac{1}{\hat{n}} \leq \frac{c\log(n)}{n}, \qquad c \geq 1, c' \geq 2, \qquad \textit{w.h.p.}$$

Proof. At first, assume the n nodes are given unique numbers $N = \{N_1, N_2, ..., N_n\}$, $|N| = n$.

Let us proove the two inequations one after the other: Firstly, we will show

$$\frac{1}{n^{c'}} \leq \frac{1}{\hat{n}}, \qquad c' \geq 2, \qquad \text{w.h.p.}$$

Assume the distance in between N_1 and N_2 is $n^{-c'}$. Denote the stretch from N_1 to N_2 with s. Assume that the positions of the remaining $n-2$ nodes are distributed indepedently and uniformly. For any node to be placed within the stretch s, we have exactly the probability

$$P(N_i \text{ splits } s) = n^{-c'} = \frac{1}{n^{c'}}, \qquad i \geq 3$$

by the independency of the remaining $n-2$ nodes, we have:

$$P(\text{Any remaining node splits } s) = \sum_{i=3}^{n} \frac{1}{n^{c'}} = \frac{n-2}{n^{c'}} \leq \frac{n}{n^{c'}} = n^{1-c'}$$

Because $n^{1-c'}$ is an inverse polynomial of degree $c'-1 \geq 1$, we have shown that the first inequation holds with high probability.

Secondly, let us show

$$\frac{1}{\hat{n}} \leq \frac{c\log(n)}{n}, \qquad c \geq 1, \qquad \text{w.h.p.}$$

Assume there is an unoccupied stretch s of length $\frac{c\log(n)}{n}$. This yields the following probabilities:

$$P(N_i \text{ does not split } s) = 1 - \frac{c\log(n)}{n}, \qquad 1 \leq i \leq n$$

again, by the independency of the nodes:

$$\begin{aligned} P(\text{No node splits } s) &= \left(1 - \frac{c\log(n)}{n}\right)^n \\ &\leq e^{-\frac{c\log(n)}{n}n} = e^{-c\log(n)} = \left(e^{\log(n)}\right)^{-c} \\ &= n^{-c} \end{aligned}$$

Thus we have shown

$$\frac{1}{n^{c'}} \leq \frac{1}{\hat{n}} \leq \frac{c\log(n)}{n}, \qquad c \geq 1, c' \geq 2, \qquad \text{w.h.p.}$$

□

Using this lemma, we get the following bounds on the estimated number of peers:

Lemma 2. *For the estimation of the number of peers $\hat{n}$, the equation*

$$\log\left(\frac{n}{c\log(n)}\right) \leq \log(\hat{n}) \leq c'\log(n), \qquad c \geq 1, c' \geq 2$$

holds with high probability.

Proof. From the last lemma, we have:

$$\frac{1}{n^{c'}} \leq \frac{1}{\hat{n}} \leq \frac{c\log(n)}{n}, \qquad c \geq 1, c' \geq 2, \qquad \text{w.h.p.}$$

and thus, for the reciprocal quotients:

$$\frac{n}{c\log(n)} \leq \hat{n} \leq n^{c'}, \qquad c \geq 1, c' \geq 2, \qquad \text{w.h.p.}$$

applying the logarithm:

$$\log\left(\frac{n}{c\log(n)}\right) \leq \log(\hat{n}) \leq c'\log(n), \qquad c \geq 1, c' \geq 2, \qquad \text{w.h.p.}$$

□

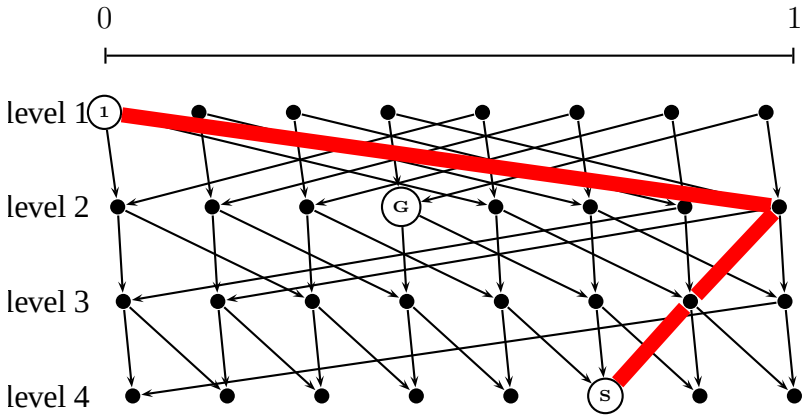
Having bound the maximum level to $O(\log(n))$ with high probability, we get the following results concerning the routing in the network.

4.2 Routing

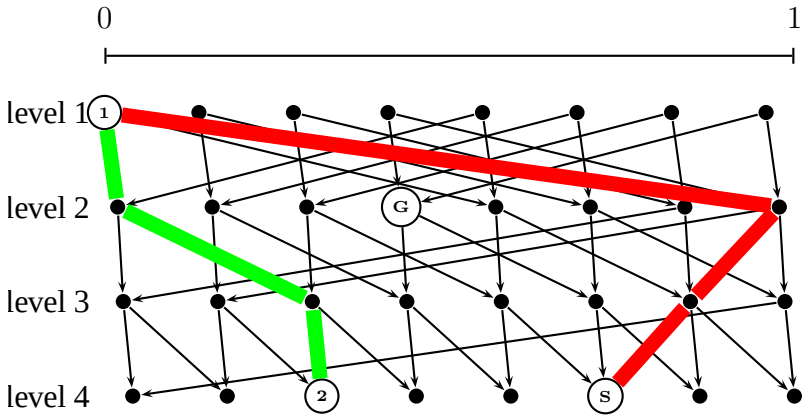
Routing involves the following three steps:

1. Route up to level1:
   This can easily be done by following up-links. Their number is $O(\log(n))$.
2. Route down to target:
   Is the target in between the left and the right down-link, route left, else right. $O(\log(n))$.
3. Traverse level and outer rings:
   If there are no down links anymore, go level-links in direction of target, if they are closer to the current peer than the target itself. Afterwards, route along the outer ring.

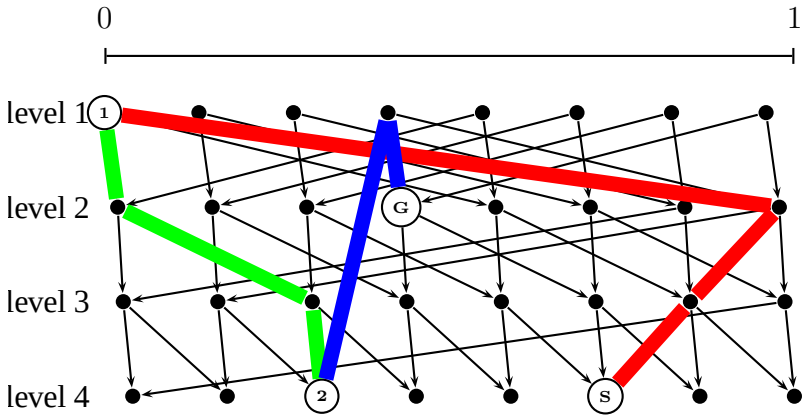
It can be shown, that the whole algorithm has logarithmic complexity with high probability. A routing example can be found in figure 4.



(a) Step 1.



(b) Step 2.



(c) Step 3.

Figure 4: Stepwise routing example through the Viceroy network.

4.3 Peer Insertion

Peer insertion involves the following steps:

1. Find peer responsible for SystemID. $O(\log(n))$
2. Reassign keys according to successor relationship. $O(1)$
3. Estimate number of peers $\hat{n}$ through distance to successing peer. $O(1)$
4. Choose Butterfly-Level evenly out of $1 \leq l \leq \lfloor \log(\hat{n}) \rfloor$. $O(1)$
5. Update links. $O(1)$ in expectation, $O(\log(n))$ with high probability
6. Additional to plain Butterfly, link peers on each level and create uplink. $O(1)$

In total, this whole algorithm features logarithmic complexity with high probability.

4.4 Ensuring Constant Indegree

Although the expected indegree is constant, it can still be logarithmic for some node. In order to limit the indegree to be always constant, the inventors of Viceroy came up with an algorithm that involves local coordination within "buckets" of $O(\log(n))$ peers. Among these peers, positions and levels are selected in a coordinated fashion. This guarantees a good and sane distribution of peers over stretches and levels.

4.5 Peer Failure / Leaving Peer

When a peer failure is detected, or a peer is leaving the network gracefully, there are certain steps that need to be taken:

1. The successing peer has to take over the data. $O(1)$
2. Every former link has to find a replacement. $O(\log(n))$

Again, the asymptotic complexity is logarithmic for this operation.

5 Summary

Viceroy was the first peer-to-peer network to combine a constant degree with a logarithmic diameter, while still preserving fairness and minimizing congestion. These strong asymptotic performance bounds are achieved through a quite complex architecture.

References

[MNR02] Dahlia Malkhi, Moni Naor, and David Ratajczak. Viceroy: A scalable and dynamic emulation of the butterfly. In *Proceedings of the twenty-first annual symposium on Principles of distributed computing*, pages 183–192, 2002.

[MS07] Peter Mahlmann and Christian Schindelhauer. *Peer-to-Peer-Netzwerke*. Springer Berlin Heidelberg, 2007.