Development of a Prototype of RCAN P2P System

Djelloul Boukhelef

Advisor: Pr. Hiroyuki Kitagawa

1 Introduction

RCAN\(^1\) [1, 2] is a self-organizing peer-to-peer (P2P) overlay under development within KDE Lab\(^2\). RCAN is based on pure and flat P2P system design. RCAN uses a self-scaling multi-ring routing infrastructure, and provides high-performance routing and querying functionalities for efficient multi-dimensional data management in large scale networks. Our aim is to propose a new P2P overlay protocol with several optimized features simultaneously: small routing path, small routing state, low maintenance cost during nodes churn, and more routing flexibility and fault-tolerance.

In this short report, we will give a "zoom-in" description of RCAN’s design and implementation, highlighting its layered and modular architectures. We will show current realization as well as some future extensions of our protocol.

As shown in figure 1, our RCAN protocol is intended to be deployed in a real computer network (May be Internat-scale in the future), and and in which different P2P protocols may be supported.

![](image1.png)

Figure 1: The distributed architecture of RCAN.

For this reason, and besides of the performance and scalability requirements, our aim is to provide an architectural design with some key features such as: flexibility, extensibility, robustness...

\(^1\)RCAN stands for Multi-ring Content Addressable Network
\(^2\)http://www.kde.cs.tsukuba.ac.jp/

2 RCAN’s layered architecture

In this section, we present a "zoom-in" description of RCAN’s design and implementation, highlighting its layered and modular architectures. Figure 2 shows the architecture of RCAN’s prototype system (v0.1) that we have developed. It basically shows the differentiation of the three major layers: Management layer, overlay layer, and communication layer.

![](image2.png)

Figure 2: Layered architecture of RCAN prototype.

2.1 Overlay Layer

"Overlay layer" is the core component of our prototype system. This layer provides basic and essential functionalities of any distributed data/services management system, such as: Publish/subscribe, Lookup, Querying, ...

RCAN protocol is designed to be deployed in a real network environment. For that reason, the "Overlay layer" has been decomposed into two main sub-modules:"Overlay core layer", and "Overlay management layer".

- "Overlay core layer" is the running environment for "peers applications". It provides the basic data management functionalities (lookup, insertion, routing, ...). Basically, "overlay core layer" is protocol-dependent component. In our current realization, RCAN, CAN, LDP, and SSP protocols are supported.
- "Overlay management layer" provide high-level
overlay management functionalities: bootstrapping, security,... In a P2P framework, this layer may be implemented as a centralized component (our case), or distributed into several cooperating components. Unlike "Overlay core layer", "Overlay management layer" is protocol-independent. As a consequence, different protocols (Chord, ... ) can be implemented by the core layer (with slight changes if necessary), and integrated to the same "overlay management layer".

- "Connection middleware" is a software component installed in single-copy on each physical machine. Its main task is to ensure connection between a peer module in the "overlay core layer" (currently implemented as a pool of threads) and the "overlay management layer", by publishing some vital information such as: network addresses (IP@ and port numbers) of the overlay management layer, administration layer; creating and destroying peers modules,...

Our architectural choices offer more flexibility, extensibility, and robustness; and, which makes our RCAN' protocol an interesting framework for P2P protocols interoperability.

2.2 Management layer
The "Management layer" provide a high-level overlay visualization and administration interface. "Management layer" collect peers' status and overlay shell information, and construct statistics about the overlay system (routing, links,...). These information are presented to the system administrator in a user-friendly fashion.

The "Management layer" uses periodic probing and soft states to detect any changes in the overlay composition, the status of peers (including peer arrival, failure), query routing, and construct a "statistical" map of the overlay.

2.3 Communication Layer
The "Communication Layer" provides abstraction from the underlying communication network peculiarities. It provides a generic and uniform interface (classes and routines) for efficient communication. This layer is deployed on top of IP protocol stack, and offers interfaces for connection/connectionless oriented modes with both synchronous and asynchronous models. A simplified version of "IO completion protocol" model is also provided in this layer.

Note that all the inter and intra-application interactions are using the message-passing communication interface provided by the "Communication Layer". Which show the importance of this layer and the need for more performance and robust network communication interfaces.

3 Version 0.1 of RCAN
We have implemented a multi-threaded version of RCAN protocols (Figure 3) with network communication facilities as described above, following an Object-oriented design.

![Figure 3: RCAN prototype v0.1.](image)

This prototype is under intensive modifications and many extensions are planned to be included into the next version of it, such as:

- **Exception handling**: The most crucial expectation from any highly-available system is to run forever. To achieve, at least partially, this objective we will to integrate an "Exception handling layer" into the core application of RCAN Prototype to prevent access conflicts and crashes.

- **IOCP Model**: To offer a high-level of scalability, flexibility, and robustness to our design, we are also studying a highly-concurrent application architecture using Win32 IO completion port model.

- Other improvements include also full and easy-to-use user interface, rich statistics, decentralized "Overlay management layer"...

References
