The Deployment Experience and Survey of the Cooperative Caching Proxy Server

Jann-Perng Tseng

Information Science, Tatung University, Taipei, Taiwan

tseng@mail.moe.gov.tw

Abstract

The objective of this paper is two-folded. The former is to report the experience of the deployment of cooperative proxy server on TANet. The latter is the survey of current state-of-the-art cooperative proxy server that was purposed in the literature or proprietary product. We study deep into the architecture of each scheme. Then, we make the comparison of these alternatives. From the analysis, it sounds that no one can advantage over others completely. It depends on what your requirement is. This reveals that although theoretically no problem but it never guarantees to work well in certain network environment.

Keywords： Cooperative Proxy Sever, ICP, Cache Digest, Cache Summary, WARP, WCCP

1. Introduction

1.1. Deployment of Caching Proxy Server

During the last few years the use of the World Wide Web (WWW) server is growing exponentially. That leads the consequence that the traffic on the national and international networks also grows exponentially. From the prevalent usage of the WWW, it means that people need much more bandwidth to meet the requests. Network administrators are facing with the difficulty of the cost to provide more network bandwidth and server capacity. Without any action, the network will become congested and the load of origin web server will be unacceptable high. Both effects will cause increasing latency.

One way to reduce the network load is to install a web caching proxy server[1]. Proxy server migrates copies of requested objects from origin web servers to a place closer to the clients. Essentially, once the object pointed to by a URL has been cached in the proxy server, subsequent requests for the same URL will result in the cached copy being returned, and little or no extra network traffic will be generated. There are many project of deploying proxy server in the national wide network. These include NLANR (National Laboratory for Applied Network Research, United States), CHOICE Project (Europe), HENSA (United Kingdom), Academic National Web Cache (New Zealand), W3 CACHE (Poland), SingNet (Singapore), CINECA (Italy) and Korea Cache Project (Korea). [1, 2, 3, 4]

1.2. The Problems

A single proxy server has its limitation in capacity to serve the requests. Although the network bandwidth grows with respect to the requirement of user, the proxy server can not afford the capacity to serve the increasing requests. This is the problem of scalability, which also yields the problem of load sharing among stand-alone proxy server. Another Problem with a single proxy server is that the reliability of service. A system failure on account of any reason will hinder the normal operation of the service, which highly impact the user or the client. Thus, how to device a mechanism or protocol to cooperate the stand-alone proxy server become an important issue.

The objective of this paper is two-folded. The former is to report the experience of the deployment of cooperative proxy server on TANet. The latter is the survey of current state-of-the-art cooperative proxy server which were purposed in the literature or proprietary product. In section 2, we give a brief introduction of the current status of TANet. Then, we mention the deployment of cooperative proxy server on TANet. It includes the innovation and system architecture. After this, we discuss the lessons and experiences we learned from the deployment of cooperative proxy server. The related researches will be described in Section 3. Although there are many related techniques suggested by the research paper, there exist much deviation in practical deployment.

[1] The term of web caching proxy server will be abbreviated for proxy server in the follow context of the paper. It implies that we dedicate on HTTP request and the proxy server does have the capability of caching.

Many of them were intuitively and theoretically no problem, but the circumstance will not be the same with the theoretical say. We give the comparison of each cooperative proxy server mentioned in section 3. The pros and cons of each scheme are proposed in section 4. We describe what we can do in the consideration of the next stage. It might include the adjustment of the topology of the network to benefit the cache meshes, the expansion of the bandwidth of the international link, the construction of the cache hierarchy or meshes, the adaptation of caching policy. Finally, we point out what the future research direction will be continued.

2. Lessons Learned From the Deployment of Cooperative Proxy Server on TANet

2.1. Taiwan Academic Network

Computer Center, Ministry of Education (MOECC) and some national universities built TANet in July of 1990. The objective is to establish a common national academic network infrastructure to support research and academic institutes in Taiwan.

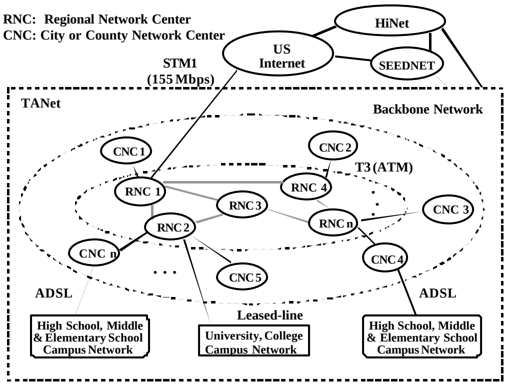

RNC: Regional Network Center
CNC: City or County Network Center
HiNet
US Internet
SEEDNET
STM1 (155 Mbps)
TANet
Backbone Network
CNC 1
CNC 2
T3 (ATM)
RNC 1
RNC 4
RNC 3
RNC n
CNC 3
RNC 2
CNC n
CNC 4
CNC 5
ADSL
ADSL
Leased-line
High School, Middle & Elementary School Campus Network
University, College Campus Network
High School, Middle & Elementary School Campus Network

Figure 1. The network architecture of TANet.

At present, there are 12 regional network centers (RNC), which are governed by 11 national universities and Ministry of Education; and 27 city or county network centers (CNC) which are run by Education Bureau of County. A STM1 (155Mbps) international link between the MOECC and Stockton in US California. All schools are connected to TANet during the past ten years. It consists of about 4100 schools and 100 academic related institutes including university, college, senior high school, and senior vocational school and K12 school. It estimates that the user is up to two million. All these efforts are to provide all teachers, students, educational administrators a comprehensive network to access resources they need and to exchange all kinds of information with one another. It has been an education and research network providing functions on teaching, research, and services for all teachers, students and educational administrators. This is also the main objective of constructing the network. TANet is a three-layered architecture as illustrated in Figure 1. These are RNCs, CNCs and campus networks. The RNCs are interconnecting with high-speed ATM circuit. The incoming and outgoing bandwidth of RNC is 120 Mbps. The CNC is the aggregate point, which connect K12 schools inside the city or county, and then connects to neighbor RNC with ATM T3 circuit. It is also a part of backbone network. Most university and college are being connected directly to the RNC instead of CNC.

The international link from TANet to U.S. Internet is a STM1 (155 Mbps) circuit. But there are only 100 Mbps for academic general-purpose usage. The other 35 Mbps is for the research network and 20 Mbps for Academia Sinica. With the rapid growth of institutes and users, it becomes congested to connect to other country via the international link. To solve the problem of limited international bandwidth, we strategically partition the 100 Mbps into two parts 36 Mbps for general-purpose use and another 64 Mbps for proxy server use only. In order to take advantage of the specific portion of 64 Mbps of the international link, a tentative-staged proxy server construction project had been applied since 1999 to improve the congestion situation.

2.2. Cooperative Proxy Server on TANet

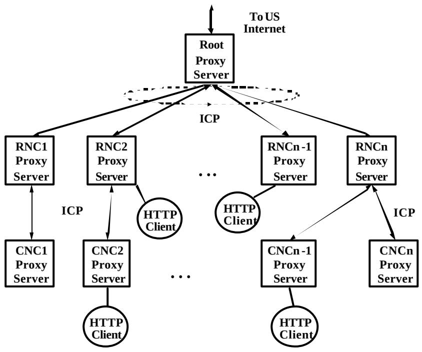


Figure 2. The architecture of proxy server on TANet. (1st Stage)

As we know that the WWW suffers from the problems of high latency, network congestion, and server overload. Caching documents throughout the web helps to alleviate the problem. For a campus or organization it is enough to set up a single proxy server. On the other hand, it is not the case of an ISP or network wide service provider. It needs a mechanism to make WWW proxy server cooperate. Most RNCs and CNCs on TANet use Squid proxy server. Squid support the Internet Cache Protocol (ICP), which make it possible to share cached object in other caches. The detail description of the ICP will be stated in next subsection.

The initial topology of cooperative proxy server on TANet is depicted in Figure 2. It is constructed with Squid proxy server in a hierarchical architecture of three levels. The top level is the root server in MOECC. The first level is RNC and the next level is CNC. There is a sibling relationship among RNCs through ICP message of Squid. The proxy server in RNC also serves proxy server of university directly connected to it. The CNC proxy server serves K12 schools. On account of too many ICP query message generated in RNC level which make the congested network become worse. Moreover, they heavily increase the latency time of client request. Another problem is that the root server can not handle the volume of requests.

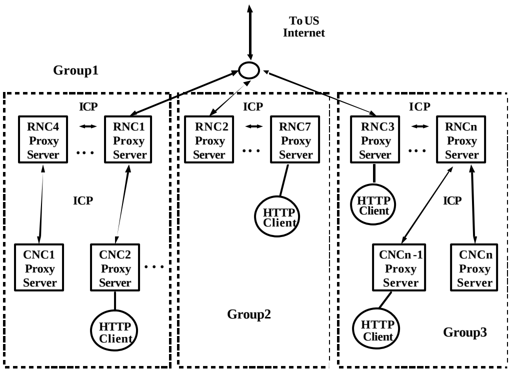


Figure 3. The architecture of proxy server on TANet. (2nd Stage)

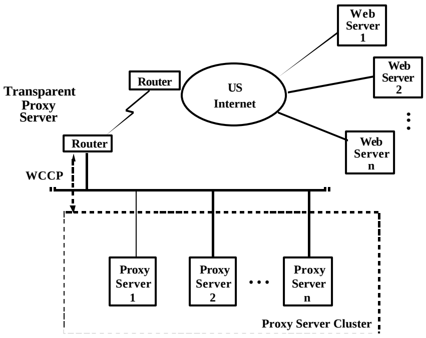


Figure 4. The architecture of cooperative proxy server in RNC.

To solve the above problem, we narrow down the hierarchical topology. The root server is also removed. The RNC level is partition into groups of caches based on geographic location. Each group can directly connect to the U.S. Internet. The sibling relationship is still existed among the groups. The ICP to used for the inter-cache communication. The CNC level remains in the same state of the initial stage. It becomes a two-level architecture and the architecture is illustrated in Figure 3. The result shows that it greatly improves the latency time. It also solves the problem of overload of the root server. The ICP traffic among groups still exists and occasionally in an unacceptable high latency.

2.3. Current Proxy Server in each RNC on TANet

In order to supply more friendly, effective and reliable service, MOECC and some RNCs have adjusted the schemes to support transparent proxy server on TANet. In addition to the cache schematic adjustment, we deploy the proprietor proxy server product to facilitate robust and scalable server.

Figure 4 illustrated the current scheme of proxy server in RNC on TANet. With the introduction of transparent proxy server, there is no need for the user to specify the proxy server in the case that the parent node is not robust enough. The detail description of the Cisco's Web Cache Communication Protocol (WCCP) will be described in section 3.

2.4. Lessons Learned

In the initial and the adaptation stage mentioned above, we know that ICP do really work with caching hierarchy or meshes. When the network is not overloaded it can sustain the ICP message and work smoothly. But under the circumstance of already congested network, it will get worse. You can not take the advantage of cooperation of caching proxy. Another issue is that we should estimate the server capacity of a parent proxy. All the requests under this parent node will serve by the parent node. It should be robust enough in capacity to supply the services; otherwise it will become the failure point in the proxy server hierarchy. The scalability of parent proxy should also to be taken into consideration. The performance of caching proxy server highly depends on available network bandwidth.

3. Related Works on Cooperative Proxy Server

Proxy servers tend to be composed of multiple distributed caches to improve system availability, scalability and load balance capability. In terms of scalability and availability, the existence of multiple distributed cache permit a system to deal with a high degree of concurrent client requests. Regardless of a logical cache system is composed of multiple distributed caches, it is often desirable to allow these caches to communicate with each other. Distributing objects among caches allows load balancing. The permission of subsequent inter-cache communication allows the overall logical system to efficiently resolve requests internally.

Different solutions are proposed to satisfy the specific requirements. There are many protocols and systems, either research domain or proprietary, deployed in web caching today. These include ICP [8], Cache Digests [12], CARP [15], WCCP [19] and so on. Additional protocols or dedicated devices are being invented to meet the innovated requirements. Although there are many related techniques suggested by the research paper, there exist much deviation in practical deployment. Many of which were intuitively and theoretically no problems, but the circumstance will not be the same with the theoretical say.

3.1. Internet Cache Protocol

Internet cache protocol (ICP) is a protocol used for communication among proxy caches. The ICP protocol is defined in two Internet RFCs. RFC 2186 [9] describes the protocol itself, while RFC 2187 [10] describes the application of ICP to hierarchical web caching. ICP is primarily used within a cache hierarchy to locate specific objects in sibling caches. It was implemented in Harvard project of Internet cache -- Squid proxy server package. If a Squid cache does not have a requested document, it sends an ICP query to its siblings, and the siblings respond with ICP replies indicating a ``HIT'' or a ``MISS''. The cache then uses the replies to choose from which cache to resolve its own MISS. ICP also supports multiplexed transmission of multiple object streams over a single TCP connection. ICP is currently implemented on top of UDP.

The ICP provides support for informed selection of a next -hop cache, including implicit indications of network congestion. There are parent and child relationship between lower level and upper lever proxy server. The top-level proxy sever behaves as parent and the second level acts as child proxy server in Figure 2 and Figure 3. The relationship among top-level proxy server can be sibling one another. It is true that the sibling relationship may exist either in a set of child or a set of parent. The difference between sibling and parent relationships is in their role during cache missing. The parent can help resolve misses, but the sibling must not. From the inter-cache relationship we specified, the proxy topology can be configured as a hierarchy or a mesh. Caches should not forward requests to sibling caches unless they know the sibling has the requested object.

3.2. Cache Digests and Summary Cache

Cache Digests are a response to the problems of latency and congestion associated with previous inter-cache communications mechanisms such as ICP. It supports peering between caching proxies and cache servers without a request-response exchange taken place. Instead, a summary of the contents of the server is fetched by other servers who peer with it. Using Cache Digests it is possible to determine with a relative high degree of accuracy whether a given URL is caches by a particular server. It is both an exchange protocol and a data format. A "lossy" technique is used for compression, which means that very high compression factors can be achieved at the expense of not having 100% correct information.

3.2.1. The System Architecture of Cache Digest

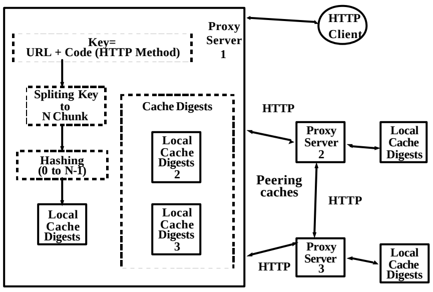


Figure 5. Architecture of Cache Digest proxy server.

3.2.2. The Theory Behind CD and CS

A Bloom filter is an array of bits, some of which are on and some of which are off. The process to determine which bit is on or off is depend on a specific number of hashing function while adding a key to the Bloom Filter. To check whether a specific entry is in the filter, we just calculate the same hashing function values for its key and examine the corresponding bits. If one or more of the bits is off, then the key is not in the filter. If all bits are on, there is some probability that the entry is in the filter.

The size of a Bloom filter determines the probability an "all-bits-on" lookup is correct. A smaller filter size will result in more errors than a larger one for the same data. The terms hit and miss is used to indicate whether or not the bits of the Bloom Filter predict that a given key is in the filter. Furthermore, the terms true and false describe the correctness of the prediction.

True hit: The filter correctly predicts the object is in the cache.

False hit: The filter incorrectly predicts the object is in the cache.

True miss: The filter correctly predicts the object is not in the cache.

False miss: The filter incorrectly predicts the object is not in the cache.

By its very nature, a Bloom Filter will always have a non-zero number of false hits. This i8s the price paid for its compact representation. When the Bloom Filter is perfectly synchronized with its source, there will be zero false misses. The detail descriptions of the management of local digests could be found in [11].

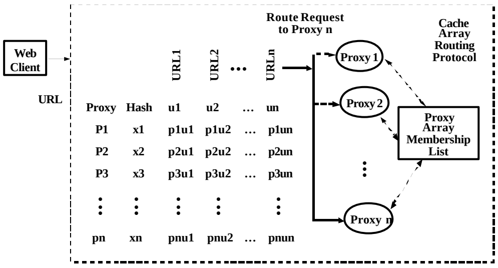
3.2.3. Summary Cache (SC)

Cache Digest and Cache Summary are designed and developed by Pei Cao and students at the University of Wisconsin Madison [13]. The main technique behind them is the Bloom Filter theory in the database field. Summary Cache extends ICP to allow "pushing" of Bloom Filter from parent caches to their children. Updates are supported via ICP as well. Summary Cache maintains a special table to keep track of deletions from a Bloom Filter. The size of that table is 4 times the size of local Bloom Filter. The table allows them to notify peers when objects are purged from the cache. The design objectives are probably the same, and there are a few differences in practical matter.

3.3. Cache Array Routing Protocol

Microsoft's Cache Array Routing Protocol (CARP) uses a hashing scheme to identify which proxy server has the requested object in contrast to ICP-based approaches, which proxy server can communicate with each other to locate the requested content. When a request come in from a client, a proxy evaluates the hash value of the requested URL with the name of the proxies it knows about, and the one with the highest value is realized to be the owner of that content. The CARP hash-routing scheme is proposed as a means for avoiding the overhead and scalability issues associated with intercache communication.

Since hashing can be used as the basis for cache selection during object retrieval, hash based routing is seen as an intercache communication solution. Its use can reduce the need for caches to query each other. Instead, requests are made to caches as a function of the hashing the URL key. There are also scenarios in which hash-based routing is used only to point the caller in the direction of the content. This can be the case for very large cache infrastructures, such as the type described by the adaptive caching project. When locating remote content, hash-based routing can be used as a means to point the local cache in the direction of other caches which either have the object or can get it from other caches or the origin server.



Proxy	Hash	u1	u2	...	un
P1	x1	p1u1	p1u2	...	p1un
P2	x2	p2u1	p2u2	...	p2un
P3	x3	p3u1	p3u2	...	p3un
⋮	⋮	⋮	⋮		⋮
pn	xn	pnu1	pnu2	...	pnun

Figure 6. The architecture of cache array routing protocol.

In order to provide with an easy way to understand how CARP works, we studied the system architecture of CARP and depicted in the Figure 6. The mechanism is described as followed.

- All proxy servers are tracked through an "array membership list", which is automatically updated through a time-to-live (TTL) countdown function that regularly checks for active proxy servers.
- A hash function is computed for the name of each proxy server.
- A hash function is computed for the name of each requested URL.
- The hash value of the URL is combined with the hash value for each proxy. Whichever URL+Proxy Server hash comes up with the highest value, becomes "owner" of the requested object.

The result is a deterministic location for all cached objects, meaning that the web browser or downstream proxy server knows exactly where a requested URL either already is stored locally, or will be located after caching. Because the hash functions used to assign values are so great (2^32 = 4294967296) the result is a statistically distributed load balancing across the array. The deterministic request resolution path that CARP provides means that there's no need to maintain massive location tables for cached information. The browser simply runs the same math function across an object to determine where it is. Detail descriptions of the hashing algorithm could be found in [15, 16].

3.4. Web Cache Communication Protocol

The Web Cache Communication Protocol (WCCP) is a Cisco-developed content-routing technology, which provides the function to integrate proxy server into network infrastructure as depicted in

Figure 4. WCCP enables platforms of router to

transparently redirect content requests. The main benefit of transparent redirection is that users do not have to configure their browsers to use a web proxy. Instead, they can use the target URL to request content, and have their requests automatically redirected to the proxy server group. The word "transparent" means that the end user does not know that a requested object came from the proxy server instead of fromthe specified origin web server.

When a proxy server receives a request, it attempts to service it from its own local cache. If the requested object is not present, the proxy server issues its own request to the originally targeted server to get the required object. When the proxy server retrieves the requested object, it forwards it to the requesting client and caches it to fulfill future requests, thus maximizing download performance and significantly reducing transmission costs.

3.4.1. The System architecture of WCCP

WCCP enables a series of proxy servers, called a proxy server cluster, to provide content to a router or multiple routers. Network administrators can easily scale their proxy servers to handle heavy traffic loads through these clustering capabilities. Cisco clustering technology enables each member proxy server to work in parallel, resulting in linear scalability. It greatly improves the scalability, redundancy, and availability of caching solution.

3.4.2. How WCCP Works

The following sequence of events details how WCCPv2 configuration works [19]:

- Each proxy server is configured with a list of routers.
- Each proxy server announces its presence and a list of all routers with which it has established communications. The routers reply with their view (list) of proxy servers in the cluster.
- Once the view is consistent across all proxy servers in the cluster, one proxy server is designated as the lead and sets the policy that the routers need to deploy in redirecting packets.

4. Discussions and Comparisons

4.1. The Pros and Cons of ICP

Pros

- It improves the hit ratio of the cache for each server. According to Wessels [8], we can expect an improvement of 10% for hit ratio as a consequence of the cooperation with neighbor caches.
- The improvement of server capacity makes it possible to handle more HTTP requests. A hierarchy is able to handle a heavier load since requests are distributed across the hierarchy.
- To achieve the load balance by controlling the number of requests each server can handles

Cons

- The cache coverage available to any proxy server does not include the content of its descendents or its cousins and their descendents. This is because queries only travel up the hierarchy, never down. This can result in a large number of false misses unless the parents are large enough to replicate most of the objects held by their descendents.
- The directory probe mechanism is combined with object transfer; fetched objects may pass through several intermediate caching servers on their way to the original requester. It also places unnecessary load on high-level servers that field the cache misses from all of their descendents.
- Multicasting can increase the probe load on all caching servers to unmanageable levels.
- Lost ICP messages or busy neighbor caches increase miss latencies. A caching server must wait for all neighbor caches to respond with a miss or timeout before directing the request upward through the hierarchy.
- It demands background and ability for the cache administrator to configure hierarchical cooperative proxy server. This becomes complicated and harder as the number of proxy server grows in the hierarchy.
- Manually configuration is needed when a proxy server of the cluster is down. There existed single point of failure.

4.2. The Pros and Cons of Cache Digests and Cache Summary

Pros

- Object retrieving latency is eliminated and client response time could be improved.

- Network utilization may be improved.

Cons

- Additional overhead is needed but it is advantages over the ICP scheme.
- Cache Digests are not always perfectly synchronized, there will be some number of false miss
- It is not yet clear that the tradeoffs between cache digest size and the effectiveness.
- Cache Digests are relative large data structure. Rebuilding a digest may be CPU intensive and it is also require to allocate temporary memory for storing the new digests while it is being built.
- If deleting from the digest is not supported, even a memory resident digest must be rebuilt from scratch on a periodic basis to erase the bits of stale object.

4.3. The Pros and Cons of CARP

Pros

- CARP uses hash-based routing to provide a deterministic request resolution path through an array of proxies. The result is single-hop resolution. The web browser or a downstream proxy will know exactly where each URL would be stored across the array of servers.
- It has positive scalability. Due to its hash-based routing, and hence, its freedom from peer-to-peer pinging, CARP becomes faster and more efficient as more proxy servers are added.
- It protects proxy server arrays from becoming redundant mirrors of content. This vastly improves the efficiency of the proxy array, allowing all servers to act as a single logical cache.
- It automatically adjusts to additions or deletions of servers in the array. The hashed-based routing means that when a server is either taken off line or added, only minimal reassignment of URL cache locations is required.
- It provides its efficiencies without requiring a new wire protocol. It simply uses the open standard HTTP. One advantage of this is compatibility with existing firewalls and proxy servers.
- It can be implemented on clients using the existing, industry-standard client Proxy Auto-Config file (PAC). This extends the systemic benefits of single hop resolution to clients as well as proxies. By contrast, ICP is only implemented on Proxy servers.

Cons

- It provides better load balancing of the proxy servers. There is a re-balancing that has to be done on the failure of a proxy server in the cluster.
- Hashed documents may be re-weighted to adapt to the load of a server on the system.
- It is suitable in the LAN environment.

4.4. The Pros and Cons of WCCP

Pros

- It is working in conjunction with existing network infrastructure, operates transparently to user. Client does not need to configure their browsers to point to a specific proxy server.
- WCCP-enabled routers perform a hashing function on the incoming request's destination IP address, mapping the request into one of 256 discrete "buckets". Statistically, this hashing function distributes incoming requests evenly across all buckets.
- If any proxy server in a cluster fails, the cluster automatically "heals" itself. The WCCP-enabled router redistributes the failed proxy server's load evenly among the remaining proxy servers. The cluster continues operation using one less proxy server but operation is otherwise unaffected.
- It enables a Multigroup Hot-Standby Router Protocol (MHSRP) router pair to share a proxy server cluster, creating a fully redundant caching system.
- It support the multi-homed routers, overload bypass and dynamic client bypass functions.

Cons

- Network equipments are involved in this scheme.
- We can not know if the load of each proxy server in the cluster is evenly distributed especially for the hot-spot problem.
- The integration of heterogeneous proxy

server, which support WCCP but with different capacity and computing power should be further explored.

4.5. The comparison of related coope rative proxy server

The comparison of related researches on cooperative proxy server is illustrated in Table 1. The criteria we used include: year, communication protocol, architecture, scalability, load balancing, network device, overhead bypass, single point of failure, additional overhead and paradigm. The proprietary is at large advantageous over other schemes in most criteria.

5. Conclusions and Future Works

In this paper, we study the different schemes of cooperative proxy server. We study deep into the architecture of each scheme. Then, we make the comparison of these alternatives. From the analysis, it sounds that no one can advantage over others completely. It depends on what your requirement is. This reveals that although theoretically no problem but it never guarantees to work well in certain network environment. This is also the reason why we do the study of cooperative proxy server.

In section 2, we describe the innovation of cooperative proxy server on TANet and lessons learned. From the experience, we may claim the following conclusions.

- Being a parent cache, the server capacity is significantly important.
- ICP could be used to facilitate the cache hierarchy or mesh. But it becomes invalid in the circumstance that the network is congested. The case of the first stage of cooperative proxy server on TANet is an example.
- Prefetching technique does help to improve the user latency. Since it consumes much more bandwidth than normal state, and gains tremendous load in an already congested or few bandwidth available environments.

The main contribution of this paper is in the following:

- The deployment experience of cooperative proxy server.
- Detail description of the different schemes of cooperative proxy server, it provided not only the theory but also the comparison among them.
- The comparison give the guideline for whomever want to do the deployment of cooperative proxy server.

In practical deployment, we should take the following criteria into consideration. These include functionality, scalability, reliability and load balancing of cooperative proxy server.

Finally, there are still more works both theoretical and practical left to be done in the future listed in the following. Some alternatives schemes, such as content delivery network (CDN), content peering and partially replicated data have been proposed recently. It is also necessary to do the analytical or quantitative analysis.

- Alternative cache protocols need to be studied to improve the performance of inter-cache group communication.
- With WCCP, the server scalability is well solved. But we don't know if the request is bypassed when the server is overloaded.
- It is worth while to do the study of content-aware mechanism to alleviate the load of router and proxy cache.

References

[1] Bradley Huffaker et al., "Visualization of the Growth and Topology of the NLANR Caching Hierarchy", Proceedings of the Third International WWW Caching Workshop, Manchester, England, June 1998.

[2] Jieun Lee et al., "Report on the Costs and Benefits of Cache Hierarchy in Korea", the Third International WWW Caching Workshop, Manchester, England, June 1998.

[3] "DESIRE: Development of a European Service for information on Research and Education", http;//www.desire.org/html/services/caching/.

[4] Michael Baentsch et al., "the Web's Infrastructure: From Caching to Replication", IEEE Internet Computing, Vol. 1, No. 2, 1997.

[5] Dean Provey, John Harrison, "A Distributed Internet Cache", Proceedings of the 20th Australian Computer Science Conference, Sydney, Australia, Feb. 5-7 1997.

[6] Nakul Saraiya, R. Vasudevan, "Measuring Network Cache Performance", The first IRCache Web Cache Bake-off, Jan. 24 1999.

[7] Duane Wessels, "Squid and ICP: Past, Present, and Future", August 6, 1997.

[8] K. Claffy, D. Wessels, "ICP and the Squid web cache", IEEE Journal on Selected Areas in Communications, pp. 345 - 357, 1997.

[9] D. Wessels, K. Claffy, "Internet Cache Protocol (ICP), version 2" RFC 2186, September 1997.

[10] D. Wessels, K. Claffy, "Application of Internet Cache Protocol (ICP), version 2" RFC 2187,

September 1997.

[11] Alex Rousskov, Duane Wessels, "Cache Digests", Proceeding of 3rd International WWW Caching Workshop, June 1998.

[12] Martin Hamilton, Alex Rousskov and Durane Wessels, "Cache Digest Specification Version 5", Dec. 1998.

[13] Li Fan, Pei Cao, Jussara and Andrei Z. Broder, "Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol", IEEE/ACM Transactions on Netwoking, Vol. 8, No. 3, June 2000.

[14] K.W. Ross, "Hash-Routing for Collections of Shared Web Caches", IEEE Network Magazine, pp. 37-44, Nov.-Dec. 1997.

[15] V. Valloppillil and K. W. Ross, "Cache Array Routing Protocol v1.0", Internet Draft, http://ircache.nlarnr.net/Cache/ICP/draft-vinod-carp-v1-03.txt, Feb. 1998.

[16] "Cache Array Routing Protocol (CARP) Version 2.0", Whitepaper,

[17] http://www.microsoft.com/Proxy/, "Using Cache Array Routing Protocol with Windows NT Load Balancing Service".

[18] Syam Gadde, Jeff Chase and Michael Rabinovich, "A Taste of Crispy Squid", Workshop on Internet Server Performance, June 1998.

[19] Cisco Systems, "Web Cache Communication Protocol Version 2", Cisco Cache Engine User Guide, Version 2.1.0.

[20] Hisakazu Hada, "Behavior of WWW Proxy Servers in Low Bandwidth Conditions", Proceeding of the WCW 1999.

[21] Duane Wessels, K. C. Claffy, "A Distributed Architecture for Global WWW Cache Integration", University of California, San Diego, 15 May 1996.

Appendix

Table 1. Comparison of related cooperative proxy server schemes.

Scheme	**Year**	**Communication Protocol**	**Architecture**	**Scalability**	**Load Balancing**
ICP	1997	UDP Unicasting	Hierarchy,	Poor	Poor
CD and CS	1997	HTTP	Distributed	Poor	Median
CARP	1998	Proprietary	Distributed	Well	Well
WCCP	1998	Proprietary	Distributed	Well	Well
Scheme	**Network Device**	**Overload Bypass**	**Single Point of Failure**	**Additional Overhead**	**Paradigm**
ICP	No	No	Yes	ICP Packet	Request-Reply
CD and CS	No	No	Yes	Digest Create and	Directory
CARP	No	No	No	Hashing Computation	Server / Client Hashing
WCCP	Yes	Yes	No	TCP Flow Redirection, Hashing Computation	Server Hashing