CHAPTER V

DISCUSSION AND CONCLUSIONS

5.1 Discussion

An Intelligent Domain Name System or IDNS has been proposed to justify a current name service where each unique name maps one object, especially, comparing with DNS.

Both IDNS and DNS architecture are name-to-object translation. Their name space structure is hierarchical. However, there is important features. Firstly, IDNS name space employs the concept of set to provide sharing name. Secondly, DNS names are originally restricted to ASCII-based characters known as the "LDH" rules for "letter-digit-hyphen". Though there are a number of DNS extensions for using non-ASCII characters such as an iDNS-compatible server or IDNA protocol, their implementation is only mapped non-ASCII name labels into ASCII characters. The IDNS, on the other hand, is designed to support Unicode characters which encoded in UTF-8. Thirdly, a domain name in DNS is a path in an inverted tree. However, a global name in IDNS is not only a path in its inverted hierarchical name space, but a global name can be referred to various forms mentioned in Chapter 3. Table 5.1 shows comparisons between IDNS and DNS.

Table 5.2 shows the features of the existing name service and IDNS. But this re-

Table 5.1: Comparison between DNS and IDNS features.

	DNS	IDNS
Name space	hierarchical	hierarchical and set
Mapping	name-to-IP address	name-to-object such as IP address, documented files, etc.
Name	unique name	sharing name
Character support	ASCII	Unicode

search does not focus on security issue and the process on wireless network has not been performed.

5.2 Conclusions

Name services are fundamental services of all computer networks. Various naming systems have been designed. Most of their structures are hierarchical. This structure contains a simple parent-child relationship which provides a unique name. Names at the leaf nodes of the tree represent individual objects. Thus, it limits names from being shared. This research proposes the IDNS architecture to facilitate one name mapping many objects while one result is still obtained.

The IDNS employs both concepts of sets and trees. The nature of trees aids the architecture to remain scalable. This means that even though the number of objects increase, the proposed architecture remains effective. The property of the set allows one name to refer to many objects with their different contents. Hence, names at the leaf nodes may not represent individual objects but a set of objects.

Table 5.2: Features of current name services and IDNS

	Name Service	Human-readable name	Hierarchical name space name space	Sharing unique name	Character Support	Anonymity
Existing Name Service	Grapevine	Yes	Yes	No	ASCII	No
	Clearinghouse	Yes	Yes	No	ASCII	No
	DNS	Yes	Yes	Yes[a]	{a-z, A-Z, 0-9, -}[b]	No
	GNS	Yes	Yes	No	ASCII	No
	NIS+	Yes	Yes	No	ASCII	No
	NDS	Yes	Yes	No	ASCII	No
	Handle System	Yes	Yes	No	Unicode	No
	CORBA Naming Service	Yes	Yes	No	Unicode	No
Proposed Name Service	IDNS	Yes	Yes	Yes	Unicode	No

[a]DNS allows a domain name mapping to multiple addresses for load balancing.

[b]any character (octet value) can be in any DNS label, but other applications (e.g. e-mail, www, etc) only handle {a-z, A-Z, 0-9, and hyphen (-)}, and no encoding label such as UTF-8 is in DNS.

The IDNS is based on Internet layering protocol. The experiment, compared the average transmission time using UDP with TCP protocol, ensures that sending data with TCP is slower than UDP. Therefore, the IDNS uses UDP to transmit queries and responses between a client and a server.

The IDNSRP is a process to find a query name mapping the object(s). If the IDNS name space contains m nodes, n is internal nodes. Using an m–way search tree, if J is a number of keys where $J = (m-1)\ n$. The search time complexity for doing linear search on a node takes $O(m\ log_m J)$ and the search time complexity for doing binary search on the node takes $O(log_2\ m\ log_m\ J)$. The IDNSRP with filtering mechanism allows users to obtain the correct objects. The time complexity of sorting n elements in the filtering algorithm is $O(n\ log_2\ n)$. Then IDNS is scalable.

The implementation of the IDNS prototype reveals that IDNS can work properly as required. The implementation maps between names and objects as an example. Finally, the definitions and theorem ensure that IDNS can function as designed.

5.3 Further Research

Although many limitations of the existing naming services have been eliminated, there are some areas have been left out. Therefore, the further researches can be performed under the following topics.

1. Wireless communication issue

   Since this research had been implemented and tested using wire line network, therefore there is no evidence that the process can be guaranteed under the wireless

communication environment. However, the author believes that the result of the naming service will be the same as the cabling network.

2. Network security issue

   The experiment in this research had been performed based on the assumption that the network is secured. However, in the real situation, all messages must be encrypted to protect the transferring contents. Therefore, the security of the queries that are sent along the network must be taken in the consideration for the next step of our research to ensure that the transferred contents will arrive safely.