

P2P 분산 모바일 네트워크상에서 퍼지 유사도를 이용한 멀티-레벨 스트리밍

Multi-Level Streaming Using Fuzzy Similarity in P2P Distribution Mobile Networks

이종득*

Chong-Deuk Lee*

요 약

P2P 분산 모바일 네트워크 환경에서 스트리밍 미디어 서비스의 QoS는 혼잡, 지연, 간섭 등과 같은 오버헤드에 많은 영향을 받는다. 미디어 객체의 중요도는 시간에 따라 변하기 때문이다. 본 논문에서는 이러한 문제를 해결하기 위하여 새로운 퍼지 유사도 기반의 멀티-레벨 스트리밍 기법 FSMS+를 제안하며, 제안된 기법은 오버헤드로 인하여 발생하는 스트리밍 서비스의 성능 저하를 최소화한다. 이를 위해서 퍼지 유사도 기반의 적합도 기법을 적용하며, 이것은 오버헤드가 최소인 스트리밍 미디어 객체를 동적으로 스트림할 수 있게 한다. 시뮬레이션 결과 제안된 기법은 거리 기반 기법, 디스크 캐싱 기법, 그리고 프리픽스 기법과 비교할 때 재전송율, 혼잡 제어율, 그리고 지연율의 성능이 보다 우수함을 보인다.

Abstract

In P2P distribution mobile networks, QoS of streaming media services are under heavy influence from overheads such as congestion, latency, and interference. The problem is further complicated by the fact that the popularity of media objects changes over time. This paper proposes a new FSMS+ (Fuzzy Similarity-based Multi-level Streaming Scheme) which minimizes performance degradation of streaming services due to overhead. We then utilize fuzzy similarity-based relevance that can dynamically stream the streaming media object with minimum overhead. The simulation result showed that the proposed scheme has better performance in retransmission rate, congestion control rate and latency rate than the other existing methods of distance method, DC (disk caching) method, and prefix method.

Key words : wireless mobile network(무선 모바일 네트워크), congestion(혼잡), jitter latency(지터지연), fuzzy relevance(퍼지 적합도)

I. Introduction

A lot of multimedia services based on distributed

network methods are proposed for streaming multimedia data in P2P distribution mobile networks. Efficient streaming of multimedia data requires functions such as

* 전북대학교 공과대학 전자공학부(Div. of Electronic Engineering, Chonbuk National University)

· 제1저자 (First Author) : 이종득

· 투고일자 : 2011년 4월 6일

· 심사(수정)일자 : 2011년 4월 6일 (수정일자 : 2011년 6월 7일)

· 게재일자 : 2011년 6월 30일

streaming resources management for continuous flow of media data, channel management for the minimization of average latency time of data for transmission, data storage management for efficient manage of limited storage, and minimization of network bandwidth and access latency. Proxy caching mechanism, which satisfies the listed requirements, is proposed in distributed environment for the streaming of user service oriented static media objects [1],[2],[3],[4].

This mechanism streams media objects continuously without the access of clients to the server by using data blocks in proxy server. For the efficient streaming of multimedia data in distributed environment, however, we should consider the minimization of jitter for service objects, latency time, and transmission hit rate. Segment-based proxy caching scheme, proposed to solve this problem, however, has the problem of streaming delay during classification since media objects are partially cached for streaming [2],[5].

The CDN is one technique to alleviate these problems. CDNs cache partial or entire media objects at proxies deployed close to clients, and thereby reduce network and server load and provide better quality of service to end-clients [6],[7]. CDN method is based on numerous servers in the internet. Thus, streamed packets are supposed to be distributed in the numerous servers and transmitted to neighbor clients through servers. Disk caching method includes cache storing policy and disk cache replacement issues [7]. Prefix caching method is to hide the startup latency, network jitter and perform look ahead smoothing into the client playback buffer [8].

However, existing methods for caching media objects are not appropriate for caching media objects. The main reason is due to the large sizes of typical media objects. Therefore delay problem is inevitable due to the limited resources when many data packets are required at the same time. The P2P-based streaming service method is relatively efficient for multimedia streaming [9],[10]. In P2P-based distribution mobile applications, a client, as a peer, forwards streaming of data packets from the proxy

as well as to neighbor clients. A client must help the server by the cooperation with other clients. The performance of the system can be improved by alleviating the burden of the server through the cooperation of clients. In distributed mobile applications, however, peers cannot control data packets adaptively due to limited resources and mobility.

Thus, this paper proposes a new FSMS+(Fuzzy Similarity-based Multi-level Streaming Scheme) for the adaptive streaming of large size multimedia data. The proposed FSMS+ optimizes streaming services by: i) Use of fuzzy relevance metric, ii) Minimization of packet loss due to congestion and conflict, and iii) Minimization of re-streaming due to disconnection and delay. Simulation showed that the proposed FSMS+ has better performance than the other existing methods of distance method, DC method, and prefix method

II. Related Works

Users with excellent network service environment may enjoy high quality transmission service while users with poor network service environment may not. Thus, most proxy structures perform services with the bit rate unequal to the bandwidth for the streaming of media clip objects. Typical media proxy mechanisms do neither consider the streaming by partitioning segment versions of media objects nor perform services by integrating various versions of media objects. Chang and Chen [1] proposed a mechanism which partitions different versions of objects during the caching. This mechanism is efficient for the segmentation of web objects. For media objects, however, this mechanism has restrictions due to the character of large size. Miao and Ortega [5] proposed selective caching mechanism to improve playback quality. This is a intermediate frame selection mechanism which caches frames according to the transmission rate. This mechanism, however, has the problem that cannot cache frames after the initial

segment directly. Wu et al. [6] proposed a dynamic control video buffer mechanism and transmission scheduling mechanism for flexible playback. Generally in terms of video reference, frames are classified as high usage frames and low usage. This kind of scenario appears mainly internet video such as live news.

Segment-based proxy caching scheme does not cache all segments for streaming but caches only a part. Prefix caching scheme caches segments classified as prefix or suffix and caches segments divided into a fixed size according to the access pattern of a client. These schemes may increase the hit rate owing to the fixed segment size. However, the performance of caching decreases because of disregard of dynamic segments. Pre-fetching scheme minimizes proxy jitter by the pre-fetch of uncached segments. But the pre-fetch of uncached segments leads the increase of network traffic and the waste of storage. In the worst case client sessions may be closed before the access of segments. Network overlay scheme improves QoS by using the relevance of peers in a group and decreases the load of the server by the cooperation with other peers. This scheme has lower reliability than Client-Server Architecture scheme [2] or CDN scheme [4] because of the problems for storage space and network bandwidth due to many peers distributed on the network. In addition to these schemes, Zhu et al proposed caching by using video staging which caches frame larger than the critical value [7]. This scheme improves the performance of caching by reducing the bandwidth. But the hit rate of the scheme decreases when the number of frame increases or profile information is not provided. The caching scheme by using congestion control mechanism [3],[10],[12] pre-fetches uncached layer media frames. The scheme analyzes whether there exist frames to be cached and applies pre-fetching window for cached frames.

III. The proposed FSMS+ mechanism

In P2P distribution mobile networks, streaming media services are under heavy influence from overheads such as congestion, latency, and jitter. This section describes the FSMS+ for an efficient streaming service. The proposed FSMS+ measures the relevance using fuzzy similarity μ [11].

3-1 Multi-level Streaming Structure

In the proposed method stream objects are stored into buckets during the initial streaming stage for adaptive multi-level streaming. The burst of packets, data size and time stamp are very important factors to determine the relevance metric in the proposed method. Let $[ts, te]$ to denote the timestamp interval ts for multi-level adaptive streaming. ts , and te are the start time and the end time of the time interval, respectively. At first L-0 level is composed of bursts and the burst block B_h is formed for streaming.

The next upper level L-1 is composed of stream pairs with higher relevance metric and lower cost determined by average relevance cost of stream pairs. This process continues iteratively until the end of time stamp or ts_e^L . Figure 1 shows the multi-level stream structure with $L=4$, $B_h=8$, and $ts=8$.

In Fig. 1, L-0 is the initial level and L-1, L-2, and L-3 represent level-1, level-2, and level-3, respectively.

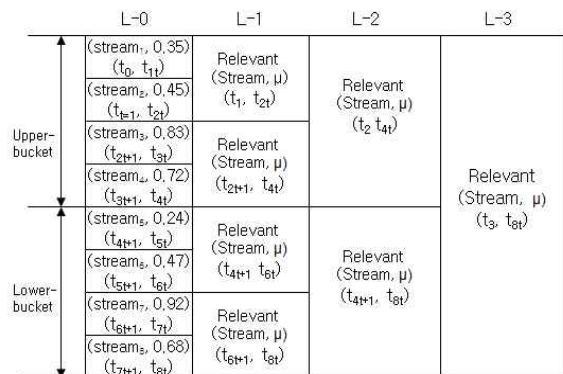


그림 1. 멀티레벨 스트리밍 구조
Fig. 1. Multi-level Stream Structure

Each level maintains the fuzzy similarity value μ of

the latest level stream. If level L-k has an input of new streams, then the fuzzy similarity value of L-k generates the new level L-(k+1) by using the similarity parameter μ and updates the end time of L-(k+1) or t_e^{K+1} .

3-2 Stream Relevance Measurement

The calculation of relevance metric to find best-fit streams starts in stream buckets by using the fuzzy similarity value μ for each stream. The relevance metric proposed in this paper is to reduce the overhead due to congestion and conflict found in some routings and clients in distribution mobile environment and to minimize re-streaming due to disconnection and delay [2],[8],[9]. For the measurement of relevance metric the search to the upper/lower levels is performed by the query in stream bucket structure. Stream objects which are satisfied the fuzzy similarity conditions is eligible for the upper level, has a small number of burst operations, and generates less packet loss due to congestion and delay. Contrary to this case the lower layer streams, especially L-0 streams, are those without fuzzy similarity test and thus requires a lot of burst operations and may suffer bandwidth limitation and delay. In order to reduce this kind of problem, the lower level streams should be described in more detail than the upper layer streams. Existing streaming methods, however, do not describe so detail as to reduce packet loss and re-streaming due to congestion and delay. The proposed method tests relevance metrics of streams to minimize packet loss and re-streaming at each level. Relevance check ensures stable streaming and improved performance. Relevance is checked as the following.

Definition 1 In order to measure the relevance of stream s_j in stream bucket S_{Bi} at each level, the relevance of j-th stream object is defined as the following.

The relevance of stream object s_j in stream bucket S_{Bi}

is $\{(sa(L1) | \mu_1 |), (sa(L2) | \mu_2 |), \dots, (sa(Lj) | \mu_j |)\}$, where L_j is the j-th stream level at level L.

After the relevance check for each stream object at each level, cost for the relevance is measured. The total cost for stream objects at level L is determined by the cost for each stream object and given as the following.

Definition 2 Relevance cost for the stream object searched in S_{Bi} at level L is

$$SR_{costS_{Bi}}(sa) = \frac{\sum_{i=1}^N \omega_i Cost_{S_{Bi}}(s_\alpha)}{N}$$

where N is the number of searched stream objects and ω_i is the weight for the stream object s_i .

Relevance cost for a stream object is a parameter for the improvement of performance. The smaller relevance cost shows the better performance. Relevance metric is measured for stream pairs as another similarity metric. The similarity metric $SR(x)$ between two substream pair of stream1 and stream2 in a stream bucket at level L is defined as the following.

Definition 3 The similarity of two sub-streams, stream object1 and stream object2, of a stream bucket at level L is

$$SR(x) = \sum_k (|stream\ object_{2k} - stream\ object_{2k-1}|)^2 \times \mu \times \alpha \times |L_n|$$

where μ is the similarity between streams, α is the weight for the similarity, and L_n is the number of stream pairs used in the measurement of burst time and data size.

After the measurement of relevance cost and

similarity at each level, relevance is tested for stream object pairs satisfying conditions. The stream object with best-fit similarity μ is selected as the most appropriate stream object. The relevance metric $HRL(x)$, used to select the most appropriate stream object pair, is defined as the following.

Definition 4 The relevance metric to select the most appropriate stream object pair at level L is $HRL(x)=\max\{\text{pairs}(\text{stream object1}, \mu), (\text{stream object2}, \mu)\}$

Thus the measurement of relevance metric helps to improve the performance of streaming.

3-3 FAS

FAS (fuzzy average similarity) represents the order of similarity between stream objects. The order of stream object is determined by the similarity measurement. This method is used to determine whether a stream object relates with the other stream objects and to solve the problem of similarity duplication. For the stream bucket SB_i and any stream object s_j in SB_i , let K denote the set of all child node of SB_i and N denote the number for stream objects in SB_i . Then FAS for SB_i of stream objects is defined by the following Eq. (1).

$$FAS = \sum_{s_j \in \text{child of parent } SB_i} \left\{ \frac{SB_i(s_j)}{N} \right\} \geq \mu - cut \tag{1}$$

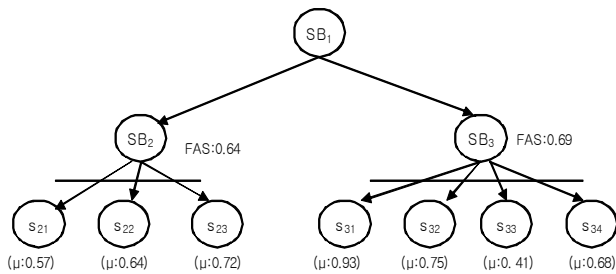


그림 2. FAS
Fig 2. FAS

As an example, let's assume that stream objects in the stream bucket depicts as Figure 2. Then SB2 is s21, s22, and s23 and SB3 is s31, s32, s33, and s34. Therefore FAS is 0.64 for SB2 and 0.69 for SB3 by "Eq. (1)".

3-4 Streaming Strategy

Best-fit strategy of streaming is the process to search the best-fit similarity of stream objects. Stream objects which are decided the best-fit similarity has higher priority and enters into the upper level. Stream objects entered into the upper level is less sensitive to the distortion of encoding rate and decoding rate. However, due to burst time and large data size, there exists distortion of encoding rate and decoding rate for the low level stream object whose level is not undetermined. This paper measures best-fit relevance to solve this problem. At each level the maximum fuzzy relevance between stream objects, $F_{\text{best-fit-rel}}$, is determined as the following Eq. (2).

$$F_{\text{best-fit-rel}} = \max \left\{ \sum_{i=1}^n \sum_{j=1}^m \frac{1}{1 + (SB_i - SB_j) \times \mu} + SR(x) \right\} \tag{2}$$

where SB_i and SB_j is the i-th stream bucket and the j-th stream bucket, respectively.

Multi-level stream structure decides whether to try to enter into the upper level by comparing with the predetermined threshold after the determination of best-fit fuzzy similarity. In order to adjust the relevance in the multi-level stream structure, threshold μ is classified as 3 groups of large burst for $\mu \leq 0.5$, middle burst for $0.6 \leq \mu \leq 0.7$, and short burst for $\mu \geq 0.8$. Fuzzy similarity is measured only if similarity is not less than the threshold value. After this procedure multi-level stream structure reallocates levels according to the updated fuzzy similarity. The updated multi-level stream structure manages streaming stably by classifying streams into congestion and uncongestion. This reduces

overhead due to disconnection and retransmission by reducing packet loss. Without best-fit similarity decision in buckets, however, there occurs packet loss due to start delay and congestion. The stream object, whose relevance is finally determined, is cached first in the proxy due to short burst time and higher priority. The next caching is performed for the stream object with best-fit similarity. The proxy continues streaming for stream objects only if the similarity is greater than the threshold. The reason for this is to reduce overhead for objects with similarities less than or equal to the threshold. After the caching of objects with similarity greater than threshold, the proxy searches the stream list in order of similarity.

The following procedure, SL_{best-fit-rel}, shows the way to determine the stream list with maximum relevance.

```

Procedure SLbest-fit-rel
SB={s1, s2, ..., sn}
//Stream objects in buckets
//Decision of fuzzy relevance for streams
StreamObjectListIndex=0
//Initialization for empty stream list for every
stream object in buckets
if (StreamListIndex >  $\sum_{i=1}^n |SB_i| \geq \mu - cut$ ) {
Fbest-fit-rel =
max  $\left\{ \sum_{i=1}^n \sum_{j=1}^m \frac{1}{1 + (SB_i - SB_j) \times \mu} + SR(x) \right\}$ 
SLbest-fit-rel
=GetStreamObjectDataSet(StreamObjectList[StreamO
bjectListIndex]);
//Generation of list for stream objects satisfying
best-fit fuzzy similarity
(GetStreamObjectDataSet(StreamObjectList[Stream
ObjectListIndex]), StreamObjectListIndex+1);
//Call the next stream object to measure best-fit
fuzzy similarity
}
else if (StreamListindex <  $\sum_{i=1}^n |SB_i| \geq \mu - cut$ )
then {
//Control of best-fit fuzzy similarity for stream

```

objects

$$\text{Threshold} = \sum_{i=1}^n |SB_i| \geq \mu - cut$$

//Decision of level according to the threshold

$$\text{else } \{SL_{\text{best-fit-rel}} \cap \sum_{i=1}^n |SB_i| \neq 0\}$$

//Decide whether to continue or stop the measurement of best-fit fuzzy similarity for stream objects

$$\sum_{i=1}^n |SB_i| \leq 0.5 - cut$$

// Stop the search of F_{best-fit-rel}

IV. Simulation results

To evaluate the proposed FSMS+, we utilized event-driven simulation. This paper utilize the parameters of table 1.

표 1. 성능 파라미터

Table 1. Performance parameter

Parameter	Value
Number of stream object	1500
Stream object size	100MB-200MB
Encoding rate	1.5MB
Decoding rate	1.5MB
Request distribution	$\theta=0.67$
Fuzzy similarity	$\mu \geq 0.8$
Packet size	512Kbyte
L _n	750
Bandwidth	10/100Mbps
t _s	[1, 20s]
Weight	$\alpha \geq 0.5$
Max caching nodes	18

The weighting factor is determined by the similarity. The simulation is performed for the proposed FSMS+, distance-based method, D/C method, and prefix caching method. In the proposed method best multi-level stream structure is assumed since the initial L-0 influences heavily on the performance of the multi-stream structure. S_{number}=CB(Cache Block) is used to denote the size and the number of the caching blocks in the simulation. Simulation is performed 5 times for five packet objects sized 300, 700, 1000, 1200, and 1500. After the

simulation, the relevance is analyzed considering stream level, the size of caching block (CB), and the size of packet data. The simulation is performed according to the proposed FSMS+ metric. Simulation scenarios are as the following.

- Simulation scenario 1: Simulation is performed with 1,500 packet data to find the packet retransmission rate for the proposed FSMS+ and random, distance, D/C, and prefix. Random is the method without the use of relevance metric.

- Simulation scenario 2: Simulation is performed to find the congestion control rate and the latency rate for the multi-level stream structure with the similarity weight μ of 0.5, 0.6, 0.7, 0.8, and 0.9.

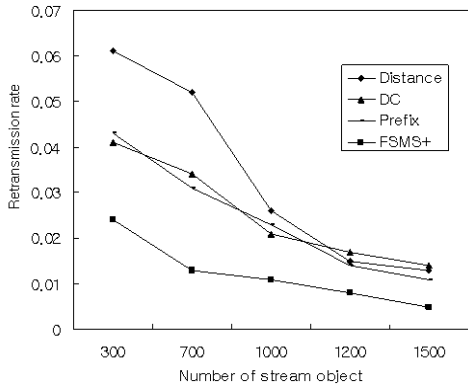


그림 3. $\mu \geq 0.8$ -cut 일 때 FSMS+ 와 기존 기법들과의 재전송율 비교

Fig 3. Retransmission Rate for FSMS+ and the Existing Methods with $\mu \geq 0.8$ -cut

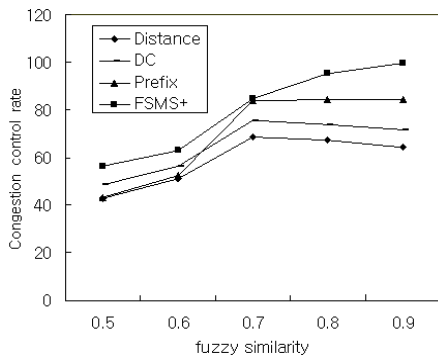


그림 4. FSMS+ 와 기존 기법들과의 혼잡 제어율 비교
Fig 4.. Congestion Control Rate for FSMS+ and the Existing Methods

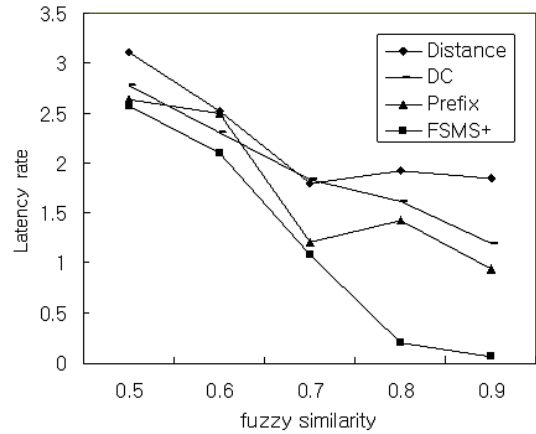


그림 5. FSMS+ 와 기존 기법들과의 지연을 비교
Fig 5. Latency Rate for FSMS+ and the Existing Methods

Figure 3, figure 4 and figure 5 show simulation results for scenarios 1 and 2. As shown in figure 4 and, the proposed method showed improvement of 20% in congestion control rate and 10% in latency rate compared with prefix method which is one of the best existing method. Thus, it is shown that the proposed FSMS+ performs streaming adaptively and stably. Therefore the simulation shows that the proposed method performs congestion control more stably and reduces packet loss more than the other existing methods.

V. Conclusion

This paper proposed a new FSMS+, to improve the performance of streaming in the P2P distribution mobile networks. The proposed method is capable of multi-level streaming service by the use of fuzzy similarity μ . For the improvement of streaming performance, next higher levels are organized successively to construct a multi-level stream structure by using the relevance metric after the loading of initial stream buckets. An best-fit stream is searched after the selection of pairs selected using the calculated cost and relevance metric by the fuzzy similarity μ . The searched best-fit stream object is selected for streaming. The NS-2 simulator is

used for the simulation. The simulation result showed that the proposed method has better performance of 20% for congestion control rate and 10% latency rate than the other existing methods.

References

- [1] Yang Guo, Zihui Ge, Bhuvan Uргаonkar, Prashant Shenoy, and Don Towsley, "Dynamic Cache Reconfiguration Strategies for Cluster-based Streaming Proxy", *Computer Communications* 29, pp. 1710-1721, 2006.
- [2] K.-L. Wu, P. S. Yu, and J. L. Wolf, "Segment-based Proxy Caching of Multimedia Streams", *Proc. Of the 10th International WWW Conference*, Hong Kong, pp. 36-44, 2001.
- [3] Bi-Ru Dai, Jen-Wei Huang, Mi-Yen, and Ming-Syan Chen, "Adaptive Clustering for Multiple Evolving Streams", *IEEE Trans. on Knowledge and Data Eng.*, vol. 18, no. 9, PP. 1,166-1,180, 2006.
- [4] A. Bulut and A. K. Singh, "Swat: Hierarchical Stream Summerization in Large Network", *Proc. Int'l Conf.*, pp. 303-314, 2003.
- [5] S. R. Gruber and J. Basso, "Protocol Considerations for a Prefix-Caching Proxy for Multimedia Streams", *Pro. 9th International World Wide Web Conference*, Amsterdam, 2000.
- [6] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDN's and WEB sites", in *Proc. Int'l www Conf.* 2002.
- [7] Zhe Xiang, Qian Zhang, Wenwu Zhu, Zhensheng, and Yu-Qin Zhang, "Peer-to-Peer Based Multimedia Distribution Services", *IEEE Trans. on Multimedia*, vol. 6, no. 2, pp. 343-354, 2004.
- [8] Song Wu, Hai Jin, Jie Chu, and Kaiqin Fan, "A Novel Cache Scheme for Cluster-based Streaming Proxy Server," *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)*, pp. 1-7, 2005.
- [9] Peng Zhu, Wenjun Zeng, and Chunwen Li, "Joint Design of Source Rate Control and QoS-Aware Congestion Control for Video Streaming Over the Internet", *IEEE Trans. on Multimedia*, vol. 9, no.2, pp.366-376, 2007.
- [10] V. Rakocevic, "Congestion Control for Multimedia Applications in the Wireless Internet," *International Journal of Communication Sys Mus*, Vol. 17, pp. 723-734, 2004.
- [11] K. Atanassov, *Intuitionistic Fuzzy Sets, Fuzzy Sets Syst.*, vol.20, pp. 87-96, 1996.

이 종 득 (李鍾得)



1983년 2월 : 전북대학교 컴퓨터과
학과(이학사)

1989년 2월 : 전북대학교 컴퓨터과
학과(이학석사)

1998년 2월 : 전북대학교 컴퓨터과
학과(이학박사)

1992년 3월~2002년 2월 : 서남대학교
컴퓨터통신학과 교수

2002년 2월~2011년 6월 현재 : 전북대학교 전자공학부 교수
관심분야 : 무선 모바일 네트워크, 무선센서 네트워크,
MIMO, 유비쿼터스 통신, 등