

Churn Resilience of Distributed Content Service System on Smart Terminals Network

Hanxing Xue* **, Jiali You* **, Jinlin Wang* **

*National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, Beijing, China

**University of Chinese Academy of Sciences, Beijing, China

xuehx@dsp.ac.cn, youjl@dsp.ac.cn, wangjl@dsp.ac.cn

Abstract— Recently, the content service system is shifting from a centralized model to distributed model with the capabilities of terminal devices increasing. In addition to providing service to local users, the smart terminals utilize unused bandwidth and storage resources to provide proximity services to neighbor terminals in the overlay network. How to ensure the performance of the distributed system with the continuous process of terminals joining or leaving the overlay network (churn) is a crucial issue. Existing strategies maintain the system's service under churn through the central server scheduling or constructing complex topology. In this paper, we propose a hierarchical topology for terminals network. The inter-cluster and intra-cluster overlay use the unstructured network. The connectivity of topology is easy to be maintained under the churn. Further, we offer a decentralized replications diffusion and maintenance mechanism on the topology. Under the churn of terminals, the number of replications can be maintained autonomously to a reasonable amount. In the ordinary churn, the success rate of services in our network only drops by about 9% and is well kept. The mechanism can also quickly restore the performance of the network under the extremely churn.

Keywords— churn resilience, smart terminals network, proximity content service, distributed replications diffusion

I. INTRODUCTION

With the development of technology, the processing and storage capabilities of terminal devices have been significantly improved. Furthermore, terminals become smarter by running intelligent algorithms. These smart terminals can form an overlay network to coordinate the resources of multiple devices to provide proximity services to users [1], [2]. For example, as one of the representative products of the smart home, the Smart-router [3] can provide proximal service to local users and neighbor Smart-routers in the overlay network by proactive caching popular content replications. At the peak of users' requests, the load on the backbone network and bandwidth consumption of the ISP in the "last-mile delivery" [4] will be reduced. Actually, in the real system, the terminals will frequently join or leave the overlay network with users turning the terminal on or off, called churn [5]. The churn has a high impact on the performance of terminals network. Therefore, how to maintain the performance of the terminal's network under the churn is a crucial issue.

Researchers analyze the churn by measuring the terminals' session time in the real systems [6], [7]. They show that the

median of all terminals' session time is varying from one minute to one hour [8], [9]. Specifically, the impact of churn on the performance of the network mainly includes three parts, including: (1) The connectivity of the network is much affected, failing content retrieval. (2) The departure of the cached terminals reduces the availability of content in the network. (3) The frequently joining and leaving of smart terminals will result in a lot of overhead to maintain the topology and replications. The commercial Smart-routers system alleviate the problem of churn by employing a centralized scheduling mechanism [3]. The servers collect information about all terminals and direct the service of content. The centralized scheduling leads to the performance of the system is limited by the capabilities of servers. Therefore, to improve the scalability of terminals network, we study the autonomous and decentralized strategy to restrain the impact of churn in this paper.

There have been many kinds of research on the resilience of churn for a distributed system. We can divide all strategies into two categories: looking up in parallel and constructing a particular network to improve the success rate of retrieval; placing redundant replications to enhance the availability of content. Since looking up in parallel [10] doesn't repair the connectivity of the network, it doesn't perform well under the high churn. Researches in [11] design hierarchical topology based on the terminals' stability to maintain the connectivity of the network. While the departure of stable terminals has a large impact on the network and the structured backbone network is difficult to maintain. In [12], researchers predict the terminal's remaining session time and combine several terminals with complementary session time to form a "general" terminal to provide service. This method requires using historical data and prediction models. In [13], researchers place redundant replications of different contents in the unstructured P2P network. The number of different content's replications are uniformly or proportional to their popularity. These strategies require manual configuration and can't adaptively adjust replications. In the DHT network, the replications strategies including descendant replications [14], multiple key replications [15], path replications [16]. The descendant replications and path replications are easy to maintain the number of replications. But the storage load of terminals will be unbalanced. The multiple key replications

place replications in different areas can balance terminals' storage. However, the maintenance of replications is difficult.

Different from existing strategies, we consider the connectivity of the network and the redundant deployment of replications simultaneously. The main content and contribution of this article include:

- (1) We design a new hierarchical network for smart terminals. Terminals are clustered into clusters. The impact of churn is limited to each cluster. The inter-cluster and intra-cluster overlay use the unstructured network. The management terminals of each cluster will easily ensure the connectivity of the network under the churn.
- (2) We propose a decentralized diffusion and maintenance mechanism of replications. The popular contents' replications will be diffused from the seed terminal to each cluster. Under the churn, the management terminal of each cluster can maintain the number of replications based on the requirement of coverage rate autonomously. Experimental results show that our mechanism can well keep the performance of terminals network under the churn.

The rest of the paper is organized as follows. In Section II, we introduce the overview of the system. Section III presents the details of implementation. Section IV presents the simulation experiments. Finally, we conclude the paper in Section V.

II. OVERVIEW OF SYSTEM

In this section, we introduce the framework of the system and the decentralized content diffusion algorithm.

A. Framework of System

The entire content service system consists of three parts: management servers, CDN servers, and smart terminals. Figure 1 shows a simplified structure of the system.

The management servers work for managing the smart terminals and formulating the requirement of coverage rate. It will help the new terminal join the terminals' network. Meanwhile, the management server periodically analyzes users' requests. Specifically, it will select some popular content and set appropriate requirement of coverage rate for each of them. After that, the management servers send content to CDN servers. Accordingly, in addition to the normal provision of reliable service about requests, the CDN servers will also need to send the diffusion content to a random management terminal, called seed terminal. The content will be diffused in the terminal's network autonomously. Smart terminals are the core of our content service system. They construct the overlay network and cache popular content's replications to provide proximity services.

To achieve the terminals network's resilience of churn, we design a hierarchical overlay topology for them. In the topology, smart terminals are clustered into several clusters based on network proximity. Specifically, terminals with the similar status of the network are aggregated into one cluster. Terminals in the same cluster construct an unstructured and

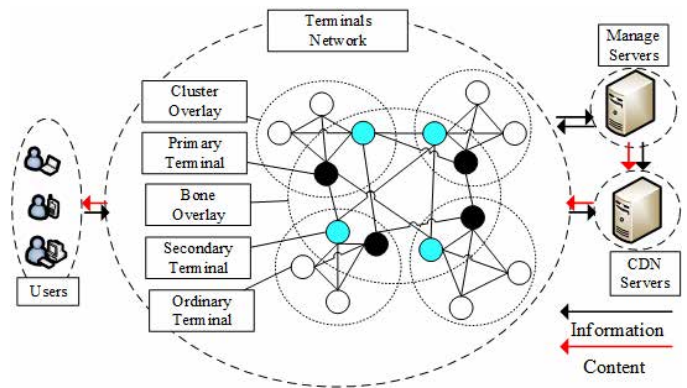


Figure 1. The simplified structure of the system.

fully connected network, called *cluster overlay*. The terminal can directly get the service of popular content from smart terminals in the same cluster overlay. If the size of the cluster is too small, the storage of all smart terminals in the cluster is not enough to cache replications of all popular contents. While, if the size of the cluster is too large, the smart terminal needs to maintain too much information. Therefore, due to the limited capabilities of each terminal, we approximately limit the size of each cluster to $[Th_c, 4Th_c]$. The value of Th_c can be manually configured.

As shown in [6], [7], old terminals are more stable than younger. We select the two oldest terminals in the cluster as the management terminals. Except for the management terminals, the remaining terminals in the cluster are ordinary terminals. In addition to providing service as well as ordinary terminals, the management terminals also need to manage the cluster, including terminal joining, terminal leaving, cluster partition, cluster merging, and replications maintenance. The propose of selecting two management terminals is to alleviate the impact of the departure of one management terminals. The two management is the primary terminal and secondary terminal, respectively. The secondary terminal is a backup of the primary terminal. Meanwhile, the management terminals between different clusters construct an overlay network, called a *bone overlay*. The bone overlay is responsible for the diffusion and retrieval of content. To make the bone overlay is easy to maintain under churn, we construct it using the unstructured network. An example of the hierarchical network is also shown in figure 1.

B. Decentralized Content Diffusion Mechanism

The target of the smart terminals' network is that the smart terminal can get low-latency service of popular content in its cluster. Therefore, we need to place popular content's replications on each cluster. As mentioned above, a management terminal is randomly selected by the CDN servers as the seed terminal for one content. The seed terminal will autonomously diffuse it to all clusters in a gossip-based way. Specifically, it will send the message of new content to its neighbors in the bone overlay. These neighbors will select a smart terminal from their cluster to cache the replication. The address of the cached terminal is also notified at the same time. The selected terminal will send a request of content to

the cached terminal to complete the diffusion of content. After that, the seed terminal's neighbors will diffuse the message to its neighbors in the bone overlay. In this way, the replications of content will be diffused to all clusters in the network. After each cluster has a replication of the content, the diffusion in the bone overlay completes. It is worth noting that if there is no change for the same content, the management terminal of each cluster processes and diffuses the message once to avoid infinite loops.

We define the coverage rate to reflect the degree of redundancy of the replications in the cluster. Cluster I 's coverage rate of content j at time t is $C_{I,j}(t)$, which is defined as the ratio of the number of replications about j in cluster I to the size of cluster I . It can be computed by the formula (1), where the $x_{m,j}(t)$ indicates whether terminal m will cache the content j ($x_{m,j}(t)=1$) at time t or not ($x_{m,j}(t)=0$). The management server will set the requirement of coverage rate for each content. The requirement of coverage rate of content j ($C_{d(j)}$) is set as the ratio of the number of requests for j to all requests. We use F and NR_j present the set of all files and the number of requests for j . The computation for $C_{d(j)}$ is showed in the formula (2). The value of $C_{d(j)}$ ranges in $[0,1]$. Assuming the situation of requests in all clusters is same, we can conclude that if the $C_{I,j}(t) \geq C_{d(j)}$, the number of j 's replications in the cluster I are sufficient to satisfy all requests about j in the cluster. Therefore, the seed terminal in each cluster will continuously diffuse content replications in its cluster overlay until the coverage rate of j meets the requirement.

$$C_{I,j}(t) = \frac{\sum_{m \in I} x_{m,j}(t)}{|I|} \quad (1)$$

$$C_{d(j)} = \frac{NR_j}{\sum_{j \in F} NR_j} \quad (2)$$

Figure 2 shows an example of the decentralized diffusion in the bone and cluster overlay. Assuming the content $F1$ will be diffused in the network ($C_{d(1)} = 0.25$). The primary terminal of Cluster1 is the seed terminal. In the first slot, $F1$ is diffused to the ordinary terminal in Cluster3. In the second slot, $F1$ is diffused to the ordinary terminal in Cluster2 and secondary terminal in Cluster4. The diffusion in the bone overlay finishes. In the third slot, the secondary terminal of Cluster4 diffuses $F1$ to an ordinary terminal in the same cluster to meet the $C_{d(1)}$. The entire diffusion completes.

III. IMPLEMENTATION DETAILS

There are several main events in the system: smart terminal joining, smart terminals leaving, cluster partition, cluster merging, cluster overlay diffusion, replications maintenance,

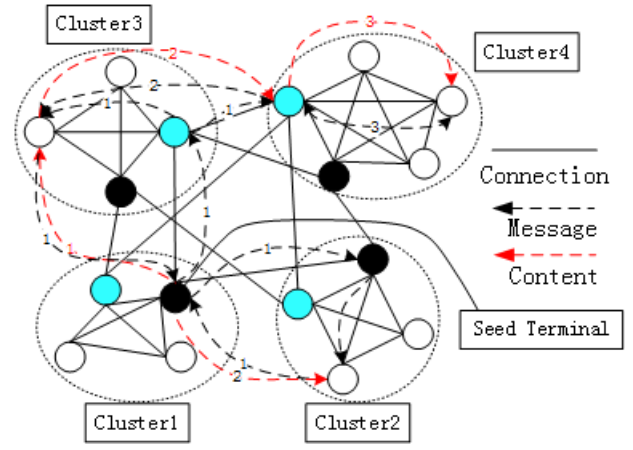


Figure 2. An example of decentralized diffusion in the bone overlay and cluster overlay.

and content requesting. In this section, we introduce the implementation details of processing each event.

A. Smart Terminal Joining

The management servers assist the joining of terminals by providing a list of partial primary terminals. The new smart terminal (m) will communicate with terminals in the list to get information. By considering the network proximity and size of the cluster, m chooses the appropriate cluster to join. Specifically, m will send a request of joining to the corresponding primary terminal (p). p will return a list of all terminals in the cluster. Further, m establishes connections by communicating with these terminals. All terminals in the cluster update their list of neighbors. Meanwhile, the size of the cluster will also be updated. The primary terminal of the cluster will partition the cluster if the size of the cluster is greater than the upper limit, i.e. $4Th_c$.

B. Smart Terminal Leaving

The departure of terminals can be divided into two categories according to the type of terminal, i.e. ordinary terminal leaving and management terminal leaving.

Ordinary terminal leaving. The departure of the ordinary terminal has two effects on the cluster. Firstly, the replications cached by the leaving terminals will not be available. Secondly, the size of cluster decreases. When the primary terminal finding some terminals have left the cluster, it will broadcast the list of leaving terminals to the remaining terminals in the cluster to update their list of neighbors. If the current size of the cluster is smaller than Th_c , the primary terminal will look for another cluster to merge. The primary terminal will also confirm the list of replications that leaving terminal cached. It will supplementally diffuse replications in the list to the cluster. In this way, these contents' requirement of coverage rate will still be met.

Management terminal leaving. Compared with the process of handling the departure of the ordinary terminal, handling the departure of management terminals firstly needs to select new management terminals and maintain the

connectivity of the bone overlay. If the primary terminal leaves the network, the secondary terminal acts as the new primary terminal. It will manage the cluster and select a new secondary terminal. The message about changing of primary and secondary terminal is broadcast to all terminals in the cluster. The new secondary joins the bone overlay by obtaining a list of other clusters' management terminals from the primary terminal. If the secondary terminal leaves the network, the primary terminal selects a new secondary terminal. It is worth noting that if the primary terminal and the secondary terminal leave the network at the same time, the two oldest terminals are selected as the new primary and secondary terminal, respectively. They join the bone overlay by requesting some clusters' management terminals from the management servers.

C. Cluster Partition

If the size of the cluster is larger than $4Th_c$, the primary terminal of the cluster will partition the cluster to relieve the pressure on each smart terminal. Specifically, the primary terminal will randomly select half of the ordinary terminals in the cluster to form a new cluster. The secondary terminal acts as a new primary terminal and organizes the remaining terminals to form a new cluster. These two primary terminals will broadcast a list of terminals in their respective cluster. Ordinary terminals update the list of neighbors in the new cluster, i.e. removing the terminals in the other cluster from the list of neighbors. Two primary terminals will also select new secondary terminal and help it join the bone overlay. Finally, the primary terminals in two clusters will maintain the replications of different contents to meet the $C_{d(j)}$.

D. Cluster Merging

If the size of the cluster is smaller than Th_c , the primary terminal of the cluster will try to merge with another cluster. Firstly, the primary terminal communicates with other neighbors in the bone overlay to acquire the size of other clusters. It will select the cluster with the smallest size to send a request of merging. If two clusters start to merge, their primary terminals will tell each other the list of all terminals in its cluster. Further, the list is broadcast to all ordinary terminals within the cluster. All terminals in the two clusters update the list of neighbors. Meanwhile, the primary and secondary terminal for the merged cluster is selected from the previous four management terminals. The other two management terminals degenerate into ordinary terminals. The new primary terminal updates the size of the merged cluster and confirms if the new cluster needs to partition. Finally, the primary terminal will also maintain the replications of different contents to meet the $C_{d(j)}$.

E. Cluster Overlay Diffusion

In the process of diffusion in cluster overlay, seed terminal will continuously place replications in its cluster until the coverage rate of j is larger than $C_{d(j)}$. Selecting which smart terminal to cache the replication also needs to be considered.

In our system, the selection of caching terminal is based on the terminals' free storage, i.e. selecting the terminal with the largest free storage to cache. In this way, the storage load of all terminals in the cluster will be balanced. Due to the limited storage of terminals, there may be cases that the storage of all terminals in the cluster is exhausted. In this case, the seed terminal will randomly select a neighbor to cache the replication. The selected terminal will replace a cached replication with the new replication. Due to the $C_{d(j)}$ reflects j 's popularity, the strategy of replications replacement can consider the $C_{d(j)}$. The terminal can select the replication with the least value of $C_{d(j)}$ to replace.

F. Replications Maintenance

After the partition and merging of a cluster, the primary terminal will maintain the replications of different contents to meet their requirement of coverage rate. Each smart terminal maintains a list of popular contents' information. These popular contents are recently diffused in the cluster. In the process of maintenance of replications, the primary terminal will traverse the content in the list. For each diffused content j in the list, it will confirm whether the number of replications of j in the cluster meets $C_{d(j)}$. If not, the primary terminal will supplementally select some terminals to cache the replications of j to meet $C_{d(j)}$. The terminal which already cached the replication of j can transmit the replications to these selected terminals. If there is no replication of j in the cluster, the primary terminal will retrieve the replication of j in another cluster through the bone overlay. These selected terminals will send a request of content to the cached terminal to complete the diffusion of content.

G. Content Requesting

In the network, each terminal maintains a list of popular contents that the system has deployed in each cluster. If the requested content (j) is not in the list, the requesting terminal (rt) will directly send the request to CDN servers. Otherwise, the rt will first try to get the service from the terminals' network. If there is a terminal in the cluster has cached j , the rt will directly get the service from the terminal. If the terminal that cached j has left the network, rt will send the request to the primary terminal to search j in adjacent clusters. To reduce the overhead of retrieval, the primary terminal only forwards the query message to one-hop neighbors in the bone overlay. If there are terminals in these clusters that can provide services, the rt directly gets the service from the terminals. If not, the rt sends the request to CDN servers.

IV. SIMULATION AND RESULT

In this section, we implement the content service system and evaluate it by simulation results.

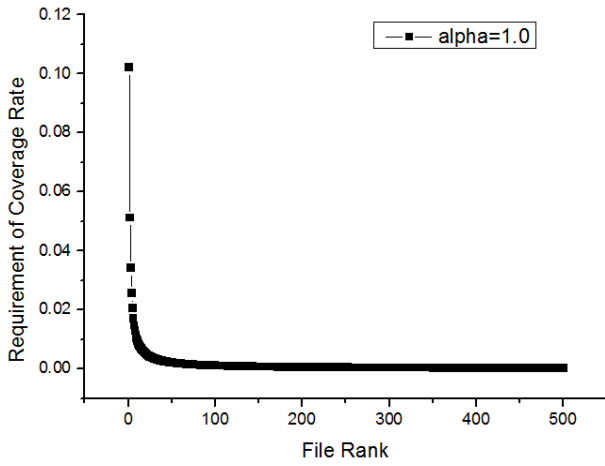


Figure 3. The $C_{d(j)}$ of different contents when the skewness of Zipf is 1.0.

A. Experiment Setup

We implement the content service system on the Peersim framework [17]. There are 5000 smart terminals in the system. We assume that the smart terminals are homogeneous. Their uplink and downstream bandwidths are 2Mbps and 4Mbps. They have 8GB eMMC high-speed storage. There are 10000 contents in the system. The average size of all contents is 200MB. In this way, each smart terminal can cache 40 replications of different content. Meanwhile, the number of requests for different content follows the Zipf's distribution (the skewness of Zipf is 1.0). We can get the $C_{d(j)}$ of the most popular 500 contents by formula (2) and show it in figure 3. We set the value of Th_c to 25. The size of each cluster is limited to [25,100]. We can find out that the storage resources of all terminals in each cluster can easily meet the $C_{d(j)}$ of 500 files. The bone overlay is built using the typical BA network.

We simulate the users' requests for content and use the success rate of services to evaluate the performance of the system. The **success rate of services** is the ratio of the number of successful servicing in smart terminals network to all requests. The terminal's capabilities are limited. In addition to the local users, the smart terminal can only provide services to one terminal at the same time. It will take the FCFS (first come first served) strategy and refuse the incoming requests which exceed its capabilities.

B. Different Scenarios of Churn

We evaluate the system's reliance of churn under two different scenarios: ordinary churn and extreme churn. The specific introduction is as following:

1) Ordinary Churn. Similar to the simulation in [18], we assume that about 50% of all terminals join the network before the session began. Five hundred files' replications have been diffused in the network. After the session begins, we will periodically select some terminals which are not online to join the network. All terminals' session time follows the Weibull distribution [6], [7]. The parameter of Weibull distribution is

set according to the real measurement results [19], i.e. $k = 40$ and $\lambda = 0.4$. The media session time is set to 1 hour.

2) Extreme Churn. The extreme churn is characterized by a large number of terminals joining or leaving the network instantaneously. These cases are common during the peak hours at night. We assume that there are 2500 terminals in the network and their session time are infinity. Similar to the simulation in [20], we instantly remove 1250 terminals in the network. Meanwhile, we also evaluate the performance of network after adding 2500 new terminals to the network instantaneously.

C. Evaluating the System

In our system, the proposed hierarchical network, replications diffusion, and maintenance mechanism work together to ensure the performance of smart terminals network. It is a comprehensive system and difficult to be directly compared with other existing strategies under the same condition. Therefore, we examine the following four networks.

UBA (Unstructured BA). All terminals are organized by the BA network.

Laurel [20]. All terminals are organized by a hierarchical topology. The bone overlay uses the model of the chord (structured network). It can maintain the connectivity of network under the churn.

HN (Hierarchical Network). The network proposed in this paper that only maintains the connectivity of topology. The bone overlay uses the unstructured network.

HN-RM (Hierarchical Network and Replications Maintenance). The network proposed in this paper that maintains the connectivity of topology and the number of replications simultaneously. The bone overlay uses the unstructured network.

At the initial moment, the replications of 500 files have been diffused in each terminals' neighbors.

1) Performance of System Under Ordinary Churn

Figure 4 presents the four networks' success rate of services under the ordinary churn. The "Upper Limit" in the figure is the ratio of requests for diffused files to total requests. In Figure 3, we can easily find that the four strategies can well satisfy the users' requests at the initial time. The reason is that our content diffusion mechanism has deployed enough replications in each terminal's neighbors. While their performance drops with the appearance of churn. The UBA network has the worst performance. After 2 hours, its success rate of services has dropped to 21%. It is because UBA lacks the maintenance of connectivity of network and number of replications. The Laurel and HN maintain their topology under the churn to ensure the retrieval of content. Therefore, compared with UBA, they perform better. The performance of HN having a slight gap with Laurel, about 2%. It is due to the structured DHT network in Laurel has better retrieval results. However, they ignore the maintenance of the replications. The reduction in the number of replications results in a decline in the smart terminal network's capability of services. The HN-RM has the best performance. It can keep the success rate of

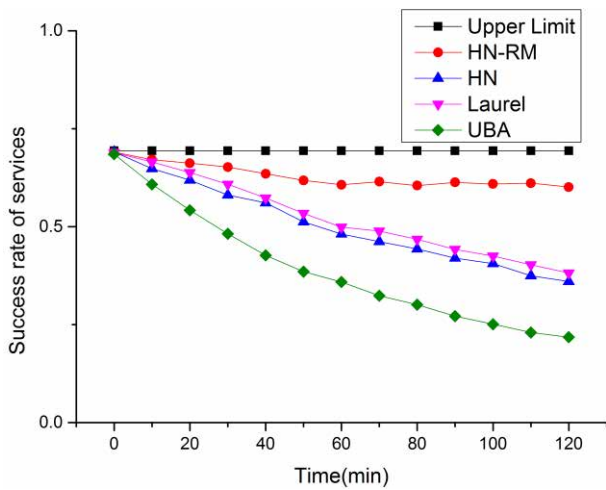


Figure 4. The performance of networks under the ordinary churn.

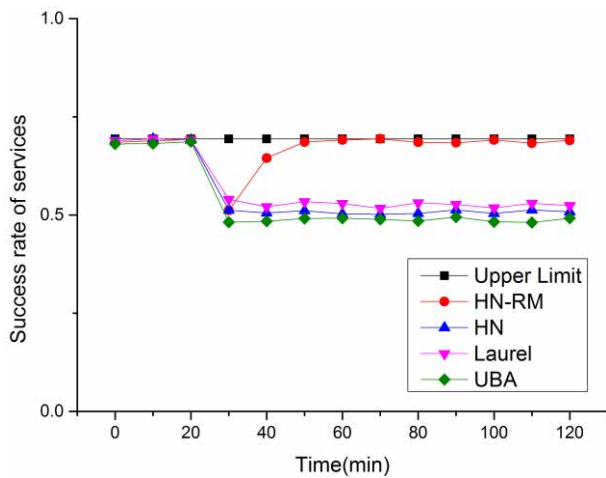


Figure 5. The performance of networks with 2500 terminals joining instantaneously.

services at about 61%. It is due to the primary terminal in HN-RM maintains the connectivity of network and replications simultaneously.

2) Performance of System Under Extreme Churn

At the 30min, we add or remove a large number of terminals in the network. Figure 5 shows the performance of networks with 2500 terminals joining instantaneously. The performance of four networks drops significantly at the 30min. It is because the number of users' requests are increased considerably. At the 30min, the Laurel has the best performance is due to the DHT network has a better efficiency of retrieval. However, due to the fixed number of replications, the Laurel will not restore the performance of the system to the best level. However, HN-RM can quickly restore the performance of the system to the best by dynamically adjusting the number of replications.

Figure 6 shows the performance of networks with 1250 terminals leaving instantaneously. The performance of four networks also significantly drop at the 30min. The main reason is that the departure of a large number of terminals substantially damages the connectivity of the network. The

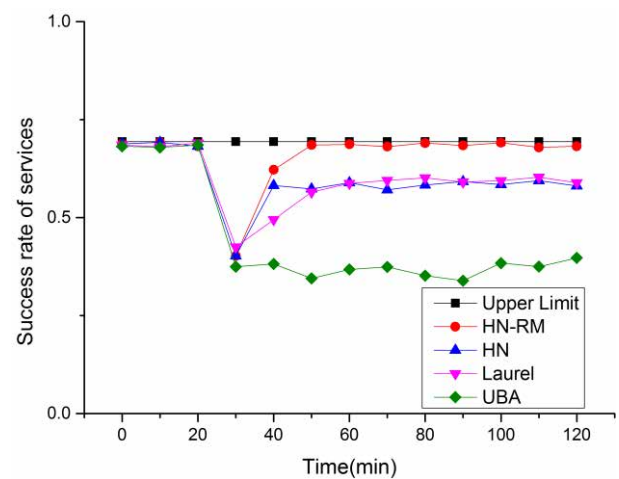


Figure 6. The performance of networks with 1250 terminals leaving instantaneously.

number replications in the network also drop significantly. Therefore, the success rate of retrieval and services are greatly reduced. The Laurel and HN repairing the connectivity of network will gradually restore the performance of the system. While ignoring the maintenance of replications causes them are impossible to restore the best performance. Meanwhile, compared with the Laurel, the HN restores the topology faster. It is because the bone overlay in HN is unstructured BA network which is easy to restore. The primary terminal HN-RM will repair the network and supplementally diffuse rare replications. In the HN-RM, the performance of the system can be well restored.

D. Discussion

The proposed HN-RM maintains the performance of the smart terminal's network well under the churn. The maintenance overhead of the HN-RM can be divided into the maintenance of topology and replications. Due to the size of the cluster is limited to a certain range, the overhead of topology maintenance in each cluster is small. In the HN-RM, all replications are evenly placed on the terminals in the cluster. Meanwhile, according to the $C_{d(j)}$, most of the contents requires only one replication in the cluster. In the simulation, the average number of replications in each terminal is 7. The primary terminal will select seven different terminals to cache the supplemental replications simultaneously. Therefore, the overhead of maintenance of replications will not be considerable.

In the system, the Management Server will periodically update popular contents' $C_{d(j)}$ according to the users' requests. Accordingly, the HN-RM will diffuse the replications and manage the number of replications in each cluster regularly.

V. CONCLUSIONS

In this paper, we propose a hierarchical network for the smart terminal's network. The unstructured inter-cluster and intra-cluster overlay network can autonomously restore the connectivity under the churn of terminals. Further, we propose a decentralized replications diffusion and

maintenance mechanism for the network. The popular content's replications will be diffused from the seed terminal to each cluster in a decentralized way. According to users' requests, we define the requirement of coverage rate in the cluster for each content. Under the churn, the management terminal of each cluster can maintain the number of replications based on the requirement of coverage rate autonomously. The experimental results show that the success rate of services in our network is well maintained in ordinary churn. The mechanism can also quickly restore the performance of the network under the extremely churn.

ACKNOWLEDGMENT

This work is partially supported by "5G and Information Center Network (ICN) Fusion Technology Development" (No.2017ZX03001019). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

- [1] G. Zhang, W. Liu, X. Hei, and W. Cheng, "Unreeling Xunlei Kankan: Understanding Hybrid CDN-P2P Video-on-Demand Streaming," *IEEE Trans. Multimedia*, vol. 17, pp. 229–242, 2015.
- [2] P.Y. Yang, Z.Q. Zhao, P. Sun, "A Data Prefetching Cache Scheme for Data Sharing System Between Smart Devices," *J. Netw. New Media*, vol. 1, no. 4, pp. 31–36, 2012.
- [3] M. Ma, Z. Wang, K. Su, and L. Sun, "Understanding content placement strategies in smartrouter-based peer video CDN," presented at the Proceedings of the 26th International Workshop on Network and Operating Systems Support for Digital Audio and Video, 2016, p. 7.
- [4] H. Yin, X. Liu, G. Min, and C. Lin, "Content delivery networks: a bridge between emerging applications and future IP networks," *IEEE Netw.*, vol. 24, no. 4, pp. 52–56, Jul. 2010.
- [5] D. Wu, Y. Tian, K.-W. Ng, and A. Datta, "Stochastic analysis of the interplay between object maintenance and churn," *Comput. Commun.*, vol. 31, no. 2, pp. 220–239, Feb. 2008.
- [6] M. Steiner, T. En-Najjary, and E. W. Biersack, "Long Term Study of Peer Behavior in the kad DHT," *IEEEACM Trans. Netw.*, vol. 17, no. 5, pp. 1371–1384, Oct. 2009.
- [7] B. Li, M. Ma, Z. Jin, and D. Zhao, "Investigation of a large-scale P2P VoD overlay network by measurements," *Peer-to-Peer Netw. Appl.*, vol. 5, no. 4, pp. 398–411, Dec. 2012.
- [8] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "Measurement study of peer-to-peer file sharing systems," in *Multimedia Computing and Networking 2002*, 2001, vol. 4673, pp. 156–171.
- [9] S. Sen and J. Wang, "Analyzing Peer-To-Peer Traffic Across Large Networks," *IEEE ACM Trans. Netw.*, vol. 12, no. 2, pp. 219–232, Apr. 2004.
- [10] D. Stutzbach and R. Rejaie, "Improving Lookup Performance Over a Widely-Deployed DHT," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, Barcelona, Spain, 2006, pp. 1–12.
- [11] M. E. Dick, E. Pacitti, and B. Kemme, "A Highly Robust P2P-CDN under Large-Scale and Dynamic Participation," in *2009 First International Conference on Advances in P2P Systems*, 2009, pp. 180–185.
- [12] X. Meng, "A churn-aware durable data storage scheme in hybrid P2P networks," *J. Supercomput.*, vol. 74, no. 1, pp. 183–204, Jan. 2018.
- [13] E. Spaho, L. Barolli, and F. Xhafa, "Data replication strategies in P2P systems: a survey," presented at the *Network-Based Information Systems (NBIS)*, 2014 17th International Conference on, 2014, pp. 302–309.
- [14] Z. Ou, E. Harjula, O. Kassinen, and M. Ylianttila, "Performance evaluation of a Kademia-based communication-oriented P2P system under churn," *Comput. Netw.*, vol. 54, no. 5, pp. 689–705, Apr. 2010.
- [15] Z. Trifa and M. Khemakhem, "A novel replication technique to attenuate churn effects," *Peer-to-Peer Netw. Appl.*, vol. 9, no. 2, pp. 344–355, Mar. 2016.
- [16] B. J. Kim, C. N. Yoon, S. K. Han, and H. Jeong, "Path finding strategies in scale-free networks," *Phys. Rev. E*, vol. 65, no. 2, Jan. 2002.
- [17] <http://peersim.sourceforge.net/>
- [18] O. C. Kwon and H. Song, "Adaptive tree-based P2P video streaming multicast system under high peer-churn rate," *J. Vis. Commun. Image Represent.*, vol. 24, pp. 203–216, 2013.
- [19] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," presented at the *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, 2006, pp. 189–202.
- [20] Z. Zheng, Y. Wang, and X. Ma, "Laurel: A Hybrid Overlay Network for Data Distribution," *J. Softw.*, vol. 22, no. 04, pp. 722–735, 2011.



Hanxing Xue was born in 1992. He received his B.S. degree in Communication Engineering, at Beijing Jiaotong University in June 2014. He is currently working towards his Ph.D. degree in Signal and Information Processing at the University of Chinese Academy of Sciences. His current research interest is about the content distribution and service on the edge network.



Jiali You is an associate professor of the National Network New Media Engineering Research Center, Institute of Acoustics (IOA), Chinese Academy of Sciences (CAS). She received her Ph.D degree in Signal and Information Processing from the Institute of Acoustics, Chinese Academy of Sciences in 2008. Between January 2015 and January 2016, she was a visiting scholar at University of Massachusetts Amherst. Her research interests include future network, content distribution network and cloud computing.



Jinlin Wang was born in 1964. He is a Professor, Director of Network and New Media Technology Research Center. He graduated from Mathematics Department of University of Science and Technology of China with his bachelor degree in 1986. After he got his master degree from Institute of Acoustics, Chinese Academy of Sciences in 1989, he began to work in the institute and was engaged in the study of digital signal processing. He had been the principal of many projects affiliated to "863 Program" in China. His current research interests include digital signal processing, application of DSP, IP network technology, structure and new service of wideband network