The adaptive Optimal Route Service design for Content Delivery Networks

Zhaohong Lai, Alex Galis, Richard Lewis, Chris Todd

University College London, Dept. of Electronic & Electrical Engineering, Torrington Place, London WC1E 7JE

{z.lai, a.galis, r.lewis,c.todd }@ee.ucl.ac.uk

Abstract: In the traditional Content Delivery Network (CDN) environment, it faces the increasing challenge to provide satisfactory QoS to deliver the delay-sensitive or mission-critical content such as medical videos. This paper presents the adaptive Optimal Route Service design to realise QoS functionalities for CDN applications, which considers the contextual information from both the service layer and network layer. It proposes an effective algorithm to compute the network topology for the ORS design. This paper also describes how to conduct the ORS experiment architecture based on DINA programmable platform.

1. Introduction

The advent of content distribution networks (CDN) has significantly improved Internet service performance and has a dramatic impact on the networking industry [1]. CDNs achieve its objectives by strategically deploying distributed Edges Servers (EDS) in different areas/countries in order to push content closer to clients [2]. Up to now, most CDN research has focused on how to effectively distribute the content from Origin Server to Edge Server [3] while the delivery network remains a bottleneck for CDN applications. Particularly, in the multimedia CDN environment [4](*multimedia Content Discovery and Delivery Network project*), as the content is highly time-sensitive and mission-crucial, how to provide the satisfied QoS services is a crucial issue for mCDN applications. Compared with other conventional networks, CDNs already contain valuable service and content metadata which describes each content QoS features before content was delivered from EDS to customers. This notable feature indicates that the QoS can be significantly improved once the network layer is aware of the service layer contextual information. In this paper, we present the adaptive Optimal Route Service to seamlessly interacting between the network layer and service layer as to realise QoS functionalities. Firstly, the scenario of adaptive ORS will be introduced. Then, it reviews the traditional context-aware services in the application and network layer and discusses its limitations in mCDN environment. After that, the adaptive ORS design algorithm will be presented and followed by its architecture design. Finally, the conclusion will be drawn.

2. Optimal Route Selection Scenario in CDNs

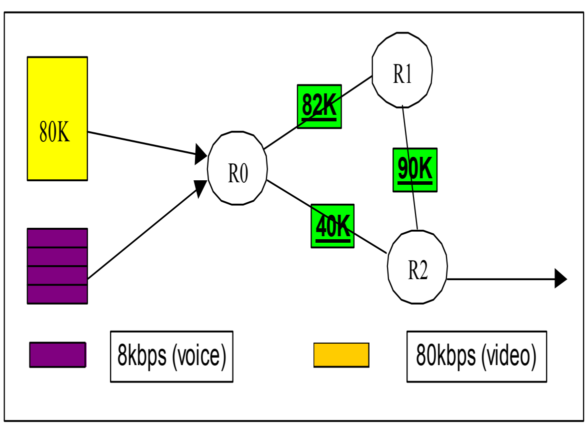

Figure 1: The optimal route service scenario in CDNs

As shown in figure 1, five content streams need to be delivered from R0 to R2 with one 80kb/s stream and four 8kb/s streams. In this network scenario, there are two links from R0 to R2: one link has a minimum 82k BW available while another link has 40K BW available. If one 8kb/s stream has been routed via link R0-R1, then the 80kb/s stream cannot be delivered from either the R0-R1-R2 link or the R0-R2 link with the required QoS. This situation may very likely happen with traditional QoS solutions. For example, DiffServ only considers the packet priority for different types of applications and partially considers network conditions like available bandwidth. Thus, in this case, to successfully deliver these contents, the network layer nodes need to be aware of the contextual information from service layer information. Specifically, in the adaptive Optimal Route Service design, firstly the service layer contextual information like explicit content metadata needs to be taken into account. Then, the network topology needs to be examined as well as the network metrics such as the available bandwidth, the delay etc. Based on the design, ORS will route the 80kbps content to link R0-R1-R2 while the other four streams are routed to R0-R2. If these five streams do not arrive at node R0 in the same time, ORS will optimise the network resource afterwards.

3. Review of Context-aware Service

In [7], *Sun et al* state that the context-awareness based information is regarded as the implicit input, which leads to the additional elements for service adaptation. They propose an application-centric context-awareness model to improve service availability and reliability. This model includes three methods: application transparent, application omniscient and application aware methods as shown below. However, concerning the ORS design in CDN environment, it is difficult to obtain application contextual information. But, the content metadata and other service context information has already been provided by mCDN such as Personalization and Profiling System (PPS) and Internet Media Guide (IMG) components [1] .

Many studies have been presented to address network-level context-aware service design. In [[5]][[7]][[8]], the proposed service model allows applications to trade off the quality of the transferred objects to adapt to changing network conditions, e.g., to select low-quality version content to compensate for a long Round Trip Time (RTT). In [[6]], an adaptive MPEG-2 video transcoding mechanism has been presented by employing the network-aware service to adjust transmission rate rather than congestion management provided by transport layer protocols. As network context input for these solutions are derived from either application or network transport level, limited functionalities can be achieved. Particularly, due to the lack of the network topology context input, the service quality is highly restricted. Moreover, many contents in CDNs can not be reduced like medical videos. In mCDN project desing, the DINA[2] [10] active platform has been chosen for retrieving the network-level contextual information.

4. Context-Aware ORS algorithm design

As discussed above, the first input required in ORS is content metadata information such as content service type, bandwidth requirement and the size of content. One example of content service type is shown in the Table 1. According to this table, ORS will verify the predefined policy database and make different decisions, e.g. the flow reservation required for the ‘0101’ type. The flow-based traffic information can greatly help ORS prevent the rerouting (flipping) as the network conditions are dynamically and frequently changing.

0001 ~ Best effort 0010 ~ voice content 0011 ~ video content 0100 ~ voice & video	0101 ~ mission critical (e.g.; medical video) 0111 ~ dynamic real-time content (financial stock data) 1000 ~ emergency content

Table 1: Content Service Types

Once ORS has obtained the service context data, the network-level contextual information needs to be provided. Most network metrics can be directly obtained from reading the MIB database such as the physical bandwidth, the available bandwidth and the link delay etc. However, it is difficult to obtain and maintain the accurate network topology in a large-scale network, e.g. AS-level [9]. Since ORS is deployed on the distributed DINA platform, we only focus on the algorithm how to obtain the network topology knowledge in one sub-AS area.

- **The algorithm to obtain the Network Topology in ORS**

The algorithm to compute the complete network topology is the core part for the ORS design. In one sub-AS area, the network topology can be deduced by the following (1) and (2) equations. If there are N nodes in one network topology, the total number of possible links φ is given as:

$$\varphi = 2 * C_n^2 \tag{1}$$

where n is equal to N. The number of possible routes Θ between any two nodes is given as:

$$\Theta = \sum_{r=0}^{n} \mathrm{P}_n^r \tag{2}$$

---

[1] The details of the mCDN project architecture are at [4]

[2] http://context.upc.es/principal_licence.htm

where n = N-2. This is can be examined by the following five-node's network structure as shown in figure 2.

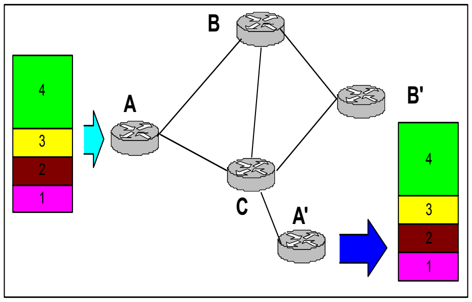

Figure 2: one five-node's network topology algorithm in ORS

Applying the above equations, from node A to A', theoretically, the total number of routes is:

$$\Theta = \sum_{r=0}^{3} P_{3}^{r} = P_{3}^{0} + P_{3}^{1} + P_{3}^{2} + P_{3}^{3} = 16,$$ as n is equal to 3 (N-2).

The total link number is: $\varphi = 2 * C_{5}^{2} = 20$. The ORS algorithm to compute available routes from A to A' is shown in the following steps:

(1) Physically, there exist 12 links in this network: AB, AC, BC, BB', CB', CA', BA,CA,CB,B'B,B'C,A'C;

(2) for P_{3}^{0} : as the link AA' can not be found in (1), the AA' boolean value ={0};

(3) for P_{3}^{1} : B, B',C = { {AB, BA'}, { AB',B'A'},{AC,CA'}}= boolean:{ {0,0},{0,0}, {1, 1}}={0,0,1}. As the last boolean value is equal to {1}, only the link ACA' is true;

(4) for P_{3}^{2} : BB', BC, B'B, B'C, CB, CB' = boolean:{0,1,0,0,0,0}; → the link ABCA' is true;

(5) for P_{3}^{3} : BB'C, BCB', B'BC, B'CB, CBB, CB'B = boolean:{1,0,0,0,0,0}; → the link ABB'C is true;

Thus, there are 3 links that are available from node A to A' in this scenario: ACA', ABCA' and ABB'C. Since there is more than one link available, initially, which one is the best ORS should consider? We can compute the usage possibility for every link using the above equations as shown in the following table.

Link	A'C	BC	CA'	CB	AC	B'C	CA	CB'	AB	BA	B'B	BB'
Value	10	10	10	10	11	11	11	11	12	12	12	12
P(L)	0.075	0.075	0.075	0.075	0.083	0.083	0.083	0.083	0.090	0.090	0.090	0.090

Table 2: the link usage probability in five-node's network topology

The table 2 shows that four links' (AB,BA,B'B and BB')usage probability is higher than others. The link ABCA' usage probability sum is equal to 0.24 (AB+BC+CA'), which is smaller than the link ABB'C value 0.263.As a result, the link ABCA' priority in ORS is higher than the link ABB'C. Once ORS has obtained the network topological context information, it will request Information Broker based on DINA platform to retrieve the minimum available BW for each available link. Finally, ORS can make decisions to optimise content delivery while also maximising network resource utilisation.

4. ORS experiment architecture design

This architecture employs the DINA [10] active platform to model the different layer contextual information by the active codes. Three Linux-based routers R0, R1 and R2 are included in this experiment which constructs three subnets: 192.168.[1-3].0 as shown below.

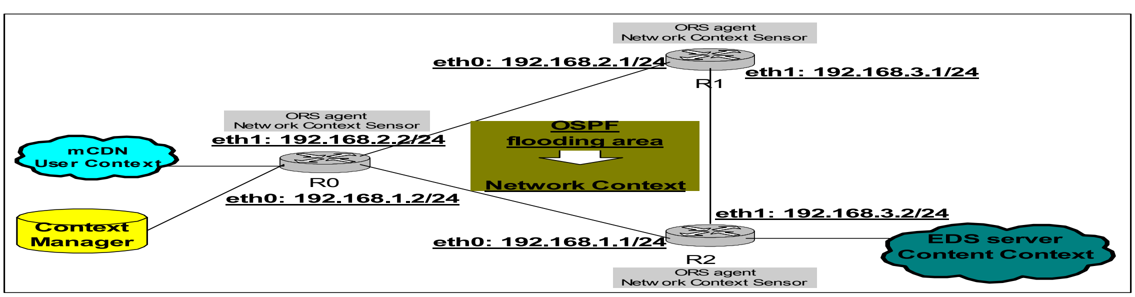

Figure 3: Network-aware ORS experiment architecture

The VLC video stream server is installed in R2 as a simulated Edge Server while R0 simulates the mCDN client. The ORS architecture comprises three sub components:

- Context Manager (CM): acts as server that receives the context data from clients and stores it in a context database. In this experiment, CM is deployed in R2. There is one CM daemon that is responsible to communicate with ORS agent and context sensor. It has the complete network-centric context paradigm picture, and performs the ORS algorithm based on predefined policies. CM programs the decision and sends out the active code to the ORS agent in IP active packets.
- Context Sensor (CS): designed for the purpose of capturing the required context information. In the OSPF environment, the interface has been designed in CS to access the OSPF MIB via SNMP to obtain network-level network feedback. CS periodically reads the MIB It sends updated contextual information to CM as it perceives changing conditions, such as a link becoming congested.
- ORS Agent: ORS Agent is able to execute active code to perform the re-route action. Moreover, since ORS agent has the flow knowledge for each content, it can reserve the bandwidth for the running flow, unless Context Manager issues a new action.

5. Conclusion

This paper presents an innovative optimal route service mechanism to realise QoS functionalities for CDN applications. After presenting the ORS scenario in CDNs, it argues that both service layer and network layer contextual information need to be considered for the ORS design. The proposed network topology computation algorithm shows it can greatly improve the ORS decision-making. Finally, how to design the ORS experiment architecture also has been described.

6. Acknowledgement

This paper describes work partially undertaken in the context of the mCDN (FP6-507993) - Information Society Technologies project, which is partially funded by the Commission of the European Union. We thank mCDN consortium for their good co-operation in the project.

7. Reference

[1] Vakali, A.; Pallis, G., Content delivery networks: status and trends, Internet Computing, IEEE,Volume 7, Issue 6, Nov.-Dec. 2003 Page(s):68 – 74

[2] Molina, B., et al, 2004, “A closer look at a content delivery network implementation”, Electro technical Conference, MELECON 2004. Proceedings of the 12th IEEE Mediterranean,Volume 2, 12-15 May 2004 Page(s):685 - 688 Vol.2

[3] Ercetin, O.; Tassiulas, L., 2003, “Market-based resource allocation for content delivery in the Internet”,Computers, IEEE Transactions on Volume 52, Issue 12, Dec. 2003 Page(s):1573 – 1585.

[4] EU FP6 mCDN (Multimedia Content Delivery Networks) project, http://www.comtec.e-technik.uni-kassel.de/ content/projects/mcdn.

[5] Bolliger, J., et al, 1999, “Bandwidth modelling for network-aware applications”,INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings. IEEE, Volume 3, 21-25 March 1999 Page(s):1300 - 1309 vol.3.

[6] Khan, J.I.; Qiong Gu,2001,”Network aware video transcoding for symbiotic rate adaptation on interactive transport”, Network Computing and Applications, 2001. NCA 2001. IEEE International Symposium on 8-10 Oct. 2001 Page(s):201 – 212

[7] Bolliger, J.; Gross, T.R.; 2001, “Bandwidth monitoring for network-aware applications”,High Performance Distributed Computing, 2001. Proceedings. 10th IEEE International Symposium on 7-9 Aug. 2001 Page(s):241 - 251

[8] Bolliger, J.; Gross, T.; 1998, “A framework based approach to the development of network aware applications”, Software Engineering, IEEE Transactions on Volume 24, Issue 5, May 1998 page(s):376 – 390.

[9] Aho, A.V.; Lee, D.; “Hierarchical networks and the LSA N-squared problem in OSPF routing”, 2000. 27 Nov.-1 Dec. 2000, GLOBECOM '00. IEEE, Volume 1, 397 - 404 vol.1

[10] Galis, A., Denazis, S., Brou, C., Klein, C. (ed) – “ Programmable Networks for IP Service Deployment“ ISBN 1-58053-745-6; pp450, June 2004; Artech House Books; www.artechhouse.com.