An overlay-based service architecture for distributed video surveillance system

Xuesong Cao
School of Computer, Wuhan University, China
caoxs1979@gmail.com

Zhaoping Wang, Ruimin Hu, Jun Chen
National Engineering Research Center for Multimedia Software, Wuhan University, China
seasonbird2004@gmail.com

Abstract
In this paper, a service architecture based on overlay network was presented. The architecture was designed to build a novel distributed video surveillance system, called as E-touch. In the model, the system is divided into three layers: underlying network layer (UNL), overlay service network layer (OSNL) and application layer (AL). OSNL is composed of many intelligent overlay nodes mapped by underlying network layer. Overlay nodes provide all kinds of media service for surveillance application in AL. OSNL allows media service composition of distributed stream processing applications dynamically. And a load balance of QoS-aware service composition algorithm (LBQSC) is employed to satisfy their end-to-end QoS demands. Large-scale experimental results demonstrate the scalability, efficiency and performance of the E-touch system.

1. Introduction
In past decade, the development of video surveillance systems has got the interests of both the research and industrial worlds. Applications of surveillance sensor become pervasive: They are increasingly exploited in transport monitoring, urban and building security, and bank protection [1-2].

A fully digital approach exploiting the use of distributed processing resources within intelligent networks has remained in a few years. Distributed intelligent Video surveillance system is becoming feasible in larger scale and also useful for simultaneous multi-functions applications. Research on the context of large distributed surveillance systems mainly focus on: video compression techniques [3], network and protocol techniques [4], distribution architectures and real-time communications [5].

However, those approaches suffer from the problems of: (1) scalability problem, where system got short of network resources (e.g., network bandwidth, service performance) in a large urban-surveillance application;(2) degraded quality-of-service (QoS), when different users need various level services for the limited system capacity.

Aiming at these problems, we have designed an overlay-based network architecture. The architecture can help to develop a novel service-oriented video surveillance system. The layout of the paper is as follows. Section 2 describes the relative works of overlay network. Section 3 describes our surveillance system model based on Overlay Network and proposes a QoS-aware routing scheme for service composition. Section 4 presents the simulation results
to evaluate the scalability and efficiency of the scheme. Section 5 introduces the actual application of the theory. Finally, we present our conclusions in section 6.

2. Overview of Overlay Network

Currently, many overlay models are designed for distributed Stream Application, such as multicasting [6], content distribution networks [7] and peer-to-peer file sharing [8]. The overlay networks effectively use the Internet as a lower-level infrastructure to provide higher-level services to end-users. The approaches are opening new ways to Internet usability, mainly by adding new services (e.g., built-in security) that are not available or cannot be easily implemented in the current Internet. Overlay network is developing rapidly within past two decades. MBone [9] deployed a large overlay multicast system in Internet based on IP tune technique. The main focus is to manage and allocate overlay links, router resources to different overlays. The OverQoS [10] project proposed an overlay-based QoS architecture for enhancing Internet QoS. The key building block of OverQoS is the controlled-loss virtual link (CLUL) abstraction. RON (Resilient Overlay Network) [11] is also based on strategically placed nodes in the Internet domains. It is proposed to quickly detect and recover from path outages or degraded performance.

The goal of our effort is as same as most of the above approaches. However, we mainly focus on media service are expected to be easily accessible and composable by users. It will help stream applications achieve QoS guarantee. We also address the load balancing problem in the service overlay system, and an adaptive and scalable load balancing technique is propped for fair allocation of resources. The research results can help us to build a large-scale overlay network for video surveillance or other distributed multimedia application. In following sections, we will describe our system model and present performance analysis results.

3. System Model

In this section, we first introduce some surveillance components for media data transport and processing. The functional modules are necessary for building a surveillance system. We have designed the system model based on the service objects, illustrated by Fig.2. Then, we formally put forward the QoS-aware service composition problem and load balancing problem, and present the resolving methods.

3.1 Service Components for Surveillance

Digital Video Surveillance Systems is a complex, distributed multimedia application for urban security. In [12], we have developed an object-based video surveillance system implemented with ICE middleware [13]. In the platform, we have classified several necessary surveillance components: Media Codec, Media Relay, Media Transcoder, Data Storage, Image Recognition, Media Player, etc, these components have usually been managed by centralized-group scheme, as shown in Figure 1.
Figure 1. Components in a surveillance system

**Media Codec:** All picture data captured by the sensors widely deployed in the city should be compressed into special media format, such as MPEG4, H.264 or AVS. The compressed data will be decompressed at the receiver before playing.

**Media Relay:** Multi-party users obtain media stream from Stream-Media Relay Servers through application-level multicast.

**Media Transcoder:** The service module transcodes different format media format or picture quality in order to satisfy different users’ requirements.

**Data Storage:** All media data can be stored in disks or multimedia database. Users search and obtain the video info according to index or key frame.

**Image Recognition:** Face recognition, Car licenses recognition and object motion-detection technologies provide security identification service for intelligent surveillance application.

Our system is designed for large-scale urban surveillance application. But actually, it is difficult to adapt to increasing QoS requirements of surveillance business. Many modules must take heavy load due to lots of task requests, such as Media Relay module and Transcoding module, and not be able to meet the end-to-end QoS either. So we raise our system model based on overlay network, and employ effective service composition and resource allocation scheme for media processing. The improved system model is applied in our E-Touch project.

### 3.2 Three-Layers Service Architecture

E-Touch project is implemented for DongGuan Urban Security System in China. The project builds a video surveillance system involved more than 2000 cameras, and stores about 600 TB video files at FC–SAN. So we must design a reasonable framework to control such great media data resource. We extract each sort of media service into one single overlay node and detach all service nodes to form an overlay service network. The middle network layer will provide all media processing modules for upper management system. Figure 2 describes a three-layer architecture of E-touch.
Figure 2. E-touch Architecture

The system model is composed of three layers: Underlying Networking Layer, Overlay Service Network Layer and Application Layer.

**Underlying Networking Layer (UNL):** UNL is a collection of all underlying physical nodes, including routers, hosts and terminals. UNL points to an existing physical topology such as backbone network, wireless channels, or LAN. In our project, the UNL is a heterogeneous entity contacts wired/wireless access.

**Overlay Service Network Layer (OSNL):** OSNL is a virtual map of UNL. It provides common media service such as Media Codec, Media Relay, Media Transcoding, etc. It also provides multi-control strategies for system balancing, such as topology discovery, overlay link performance estimation, overlay routing, resource allocation, etc. OSNL nodes are classified into three categories:

1) Resource Access node (RA Node), responsible for interaction with up-layer applications and allocation of service resources.

2) Media Service node (MS Node), responsible for implementing media processing functionalities.

3) Data node (DN), responsible for storage of all nodes info and entire overlay network topology. Each service node periodically updates own status information stored in the data nodes, helping OSNL to construct overlay topology.

**Application Layer (AL):** All overlay applications aggregates in AL. In our paper, overlay applications are mainly surveillance-oriented real-time tasks, including video monitoring, video replay, video retrieval, alarm handling and so on. The applications can be dynamically customized through OSNL. So, not only surveillance-oriented application but also other large-scale real time stream application, like Video Conference and IPTV, could acquire customizing services from OSNL.

### 3.3 Service Graph

All media processing applications in E-touch are modeled as sequences of invocations of services, which are executed in multiple nodes in OSNL. We use a directed acyclic graph, which we call application service graph, to map Overlay Service Network Layer of E-touch. We define $G = (V, E)$ as a server graph, $V$ is a set of all service nodes, and $E$ is a set of edges that represent connections between those nodes.
If the service output of a node \( v_i \) is the service input of another node \( v_j \), there is an edge \( E_{ij} \) between \( v_i \) and \( v_j \), described as Figure 3.

The services describe functionalities performed at nodes, such as encoding, decoding, transcoding and etc. When a new task \( T_i \) arrives at OSNL, OSNL can select appropriate sequences of nodes to achieve it. \( T_i \) is defined as a vector \( T_i = \{P, Q\} \). \( P \) describes details of task, including service type, service methods, source characteristic and destination characteristic. OSNL can select corresponding service nodes in term of these parameters. \( Q \) describes the QoS levels of task, as \( Q = \{q_1, q_2, ..., q_n\} \). Different QoS level means different requirements in capability of node, bandwidth and delay.

3.4 Service Composition Path

OSNL defines open interfaces for users to specify the basic services involved in a new composite service. It is desirable that the user does not need to know specific service interfaces and media formats when specifying a composite service. When tasks from AL arrive at RA Node, OSNL will perform validity check on service specifications and provides feedbacks to users. The DN stores the whole topology of OSNL and profiles of basic services. Based on DN, RA Node is able to check if OSNL can provide a composite service path for the task. Moreover, OSNL will select appropriate nodes to perform service request according to the application’s requirement and network load distributing. The service composition method is called LBQSC (load balance of QoS-aware service composition scheme), which is described in next section.
Figure 4 (a) shows a service composition path for a system that offers video view service. S1->M1->A path shows a relay model. Destination node A requires MPEG4, 128Kbps, CIF video stream from a Source Node, which may be a Digital Video Server (DVS). But the network bandwidth of DVS is limited. So, the video stream is relayed to A by Media Relay Service Nodes (M1). These relay nodes multicast video stream to users in order to save bandwidth. Similarly, S1->M1->T1->M2->B path depicts a transcoding model. Destination node B also requires video stream from S1. But B is more interested in a different video format as AVS, 64Kbps. So, the video stream will be transformed to adapt to B through a Media Transcoding Node (T1). Because T1 is a shared resource in network, the output stream is also relayed to B through M2. Figure 4 (b) shows a service composition graph for storage video demand. Destination node C requests to watch video from Storage device. Storage Node retrieve corresponding stream file in SAN Disk. T2 transcodes the stream into a special video format accepted by C.

3.5 Load Balance of QoS-aware Service Composition Algorithm

To find an optimal service path, we develop a load balance of QoS-aware service composition algorithm (LBQSC). The scheme can select the most appropriate QoS-satisfied path using the requested QoS requirements q. It calculates which paths satisfy the deadline by utilizing the current load information, handoff delay and residual bandwidth of paths (including links and nodes). Among the allocations that satisfy the QoS requirements, the algorithm returns the one which results to the maximum weight of paths. If no allocation that satisfies the given QoS exists, the scheme will refuse the task. The complete algorithm is shown as following:

**LBQSC Algorithm:**

1) According to user’s requirements (Delay and Bandwidth), take off disqualification nodes and links. Get Sub-graph: \( G' = \{V', E'\} \).

2) Select lightest load media service Node S in \( V' \) as Source Node. And select Destination node D based on service specifying.

3) Get the neighbor links collection of S: \( sE = \{E_1, E_2, ..., E_s\} \).

4) Define \( L_i \) as searching link, \( N_j \) as destination node of searching link. The selected weight is described as following:

\[
W(L_i) = K_{load} \cdot Load(N_j) + K_{bandwidth} \cdot Bandwidth(L_i) + K_{delay} \cdot Delay(L_i)
\]

\( K_{load}, K_{bandwidth} \) and \( K_{delay} \) are the weight values of load, bandwidth and delay for a link. All path delay must is less than delay of task, and path bandwidth must is more than bandwidth of task.

5) Search along all links of \( sE \). Find all QoS-satisfied paths by shortest path algorithm, as \( P = \{P_1, P_2, ..., P_n\} \).

6) Computing total delay of every path, \( D(P) = \sum_{i=1}^{n} D_i + \sum_{j=1}^{n-1} N_j \), \( D_i \) is delay of link \( i \), \( N_j \) is switch delay of Node \( j \).
7) Computing Load Fair Index [14] of every path: \( F(P) = \frac{\left( \sum_{i=1}^{n} L_i \right)^2}{n \cdot \sum_{i=1}^{n} L_i^2} \), \( L_i \) is the load of link \( i \) in the Path \( P \).

8) Computing the candidate path weight: \( \eta = F(P)/D(P)^k \), \( k < 1 \). \( k \) is the weight of delay, decided by service.

9) Select service path that result in maximum \( \eta \).

Figure 5. Path Search in LBQSC

The algorithm is described in Figure 5. The red line represents the search directions, and the destination path will be most optimal load-balancing in all alternative paths. LBQSC distributes the load across multiple nodes, driven by decisions of individual nodes. The approach allows service composition of distributed stream processing applications dynamically, to satisfy their end-to-end QoS demands with high probability.

4. Performance Analysis

In this section, we evaluate the effectiveness of the proposed LBQSC method through both simulation and stochastic analysis.

4.1. Simulations Setup

We have implemented a session-level simulator to evaluate and compare the performance of LBQSC method of E-Touch. The simulator, called as ONSP (Overlay Network Simulation Platform) is developed by object-oriented method. It can generate different overlay topology for different sessions. In the model, all components, including Overlay Node, Overlay Link, Overlay Path, Overlay Graph, and Overlay Request, are abstracted into object models. We can build a special overlay network by encapsulating and inheriting the objects.

In our simulation, we generate 2 groups of overlay topology (size =20 nodes and size =80 nodes) to evaluate the effect of the different algorithms. In the topologies, the overlay links’ bandwidths during the simulation are randomly selected between 512Kbps and 10Mbps, and delays are randomly selected between 10ms and 100ms. The
overlay links’ max processing request number is randomly selected between 15 and 20, and switch delay is between 50ms and 200ms. We simulate different number stream tasks to strike the overlay network for evaluate the system throughput. The requests require about 128kbps to 512kbps bandwidth, then bring 5-10 ms delay for overlay path.

4.2. Simulation Results and Discussions

Based on OSPN, We have got a group of compare experiments for LBQSC arming at load, bandwidth, average delay and recovery rate. The following algorithms are compared:

(1) FSC (Fixed Service Composition): The service composition method allocates a fixed source overlay node and random destination node to accept current tasks, with random path finding algorithm. If the fixed source node is overload, OSNL will select next service node to replace old source.

(2) FSC -S(Fixed Service Composition): The service composition method allocates sequential source overlay node and random destination node to accept tasks in turn, with random path finding algorithm.

(3) RSC (Random Service Composition): The service composition method allocates random source overlay node and random destination node to accept tasks, with random path finding algorithm.

(4) LBQSC-L: The decision weight of LBQSC depended on load of candidate path.

(5) LBQSC-D/L: The decision weight of LBQSC depended on load and delay of candidate path, $K_{delay} = 0.5$ and $K_{load} = 0.5$ for overlay link.

(6) LBQSC-B/L: The decision weight of LBQSC depended on load and bandwidth of candidate path, $K_{bandwidth} = 0.5$ and $K_{load} = 0.5$ for overlay link.

Algorithms (1) (2) (3) emulate traditional resource allocation method. The resource is unfairly allocated to session, without considering any factors. Algorithms (4) (5) (6) is different LBQSC with different link weight. The verified weight will affect on path finding result. Different scheme can dynamically satisfy users’ QoS requirements.

4.2.1 Success rate of task implementation

A service path finding is successful, if its resource requirements are satisfied throughout the service session. Figure 6 (a) shows the service path composition success rate achieved by the algorithms above, under request arrival rates ranging from 10 to 240 concurrency requests. Algorithm FSC achieve lower success rate than other algorithms. Algorithm LBQSC-L achieve highest success rate. And FSC–S constantly achieves almost the same success rate as RSC.
In Figure 6 (b), the result is the same as Figure 6 (a), but the gaps of algorithm performance are more obvious. In fact, if the resources have been fairly allocated, the throughput of system will be improved due to load-balance.

4.2.2 Load Fluctuation Comparison

We evaluate the load-balance degree of LBQSC method by a metrics: Load Fluctuation. The value means current load variance of all overlay nodes. We present the load by request numbers on node.

\[ L_i = \frac{\text{Current Request Number of node}}{\text{Max Request Number of node}}. \]

And Load Fluctuation is defined as:

\[ L' = \frac{\sqrt{\sum_{i=1}^{n} (L_i - \overline{L})^2}}{n}, \]

\( \overline{L} \) is average load of all nodes.

Figure 7 (a) and (b) shows the load fluctuation comparison of traditional method and LBQSC with overlay size = 20 nodes and size = 80 nodes. We observer the LBQSC can result in lower load fluctuation in the whole system. Other algorithms achieve very high load fluctuation under light load. It is because task distributions are unfair by allocating resource fixed and randomly. But all algorithms achieve lower load fluctuation. And it is because almost all nodes have been overload. So in the case, all load fluctuations of graph are same. From the result, LBQSC algorithm always achieves a more even distribution of the load, even under overload.

4.2.3 Different Weight Affection

If we consider delay except load, we can get following comparison result for different LBQSC. We Compare LBQSC-L, LBQSC-D/L and LBQSC-B/L:

1) The link weight of LBQSC-L is: \( K_{\text{load}} \times \text{Load(node)} \), i.e. \( K_{\text{load}} = 1 \).
2) The link weight of LBQSC-D/L is:
\[ K_{\text{load}} \times \text{Load}(\text{node}) + K_{\text{delay}} \times (\text{Delay}(\text{link}) + \text{Delay}(\text{node})) \], i.e. \( K_{\text{load}} = 0.5 \), \( K_{\text{delay}} = 0.5 \).

3) The link weight of LBQSC-B/L is:
\[ K_{\text{load}} \times \text{Load}(\text{node}) + K_{\text{bandwidth}} \times \text{Bandwidth}(\text{link}) \], i.e., \( K_{\text{load}} = 0.5 \), \( K_{\text{bandwidth}} = 0.5 \).

In the experiment, two metrics are defined: Average Delay, Bandwidth Fluctuation. Average Delay is measured to evaluate session latency quality under heavy load. The metric is important for low latency application, such as surveillance. And the goal of LBQSC is to have similar residual capacities at all links. We use the residual bandwidth fluctuation to evaluate this characteristic. Bandwidth Fluctuation is defined as:
\[ B' = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (B_i - \overline{B})^2} \], where \( B_i \) is average residual bandwidth of all links, \( \overline{B} \) is average residual bandwidth of link \( i \).

Figure 8 (a) and (b) illustrates the average end-to-end delay for each streaming sessions task. The figure shows that LBQSC with considering delay factor achieves the lowest end-to-end delay, regardless of the number of streams in the system. Figure 9 (a) and (b) illustrates the average bandwidth fluctuation for the whole system under increasing task load. The figure shows that LBQSC with considering link bandwidth factor achieves the lowest bandwidth fluctuation than the other two algorithms. The result shows that the selection of weights will greatly affect on the performance of our system. We must consider users’ QoS requirements to dynamically adjust our approach.
Fair and adaptive load distribution in a large-scale system can achieve scalability and maximize the probability of meeting the application requirements.

5. System Application

Figure 10 and 11 describe the implementation of E-touch project in DongGuan Urban. We deploy a large surveillance platform (including 12 relay service nodes and 600TB storage FC-SAN) in Municipal Center and many surveillance seats in Town Center. From screen of our terminal, we can observe scenes of different areas in the city. The system has been officially opened in March, 2007.

6. Conclusion

In this paper, we have presented a novel service architecture based on combined application-level overlay network. It can be used to build a large-scale video surveillance system, called as E-touch. E-touch can achieve better QoS provisioning and resource utilization than a traditional surveillance system that uses fixed resource allocation scheme. To find QoS-satisfied paths and adaptively route the overlay traffic, we employ a load-balance service composition algorithm: LBQSC. The scheme tries to allocate resources for tasks in such a way that these requirements are met and the load distribution of whole system is maximized. When system is overload, new tasks will be refused, avoiding to impact on services that accommodate already running tasks. E-touch can be easily deployed, which does not require any IP-layer multicast support. Large-scale experimental results demonstrate the efficiency of the E-touch system.
7. References


Authors

Xuesong Cao, received B.S.degree of Computer Science from Wuhan University, China, in 2002. He is a PhD candidate at school of computer, Wuhan University, China. Now he works as a team header of information security technologies for China Telecom in Guangdong. Since 2004 he has been involved in the development of digital video surveillance system in wide-area network. His current research interests include multimedia communication protocols, overlay network and QoS in distributed multimedia system. He is a student member of the CCF (China Computer Federation)

Zhaoping Wang, received B.S.degree from Wuhan University of Technology, China, in 2003. She is a PhD candidate in Wuhan University, China. Her current research concerns multimedia network and protocols, especially overlay multicast and QoS for IPTV system and wireless vehical network. She is a student member of the CCF (China Computer Federation).