Advances in Internet of Things, 2012, 2, 23-36
http://dx.doi.org/10.4236/ait.2012.22004 Published Online April 2012 (http://www.SciRP.org/journal/ait)

Automatic Service Discovery of IP Cameras over Wide Area Networks with NAT Traversal

Chien-Min Ou[1*], Wei-De Wu[2]
[1]Department of Electronics Engineering, Ching Yun University, Chungli, Chinese Taipei
[2]Department of Computer Science and Information Engineering, National Taiwan Normal University, Taipei, Chinese Taipei
Email: *cmou@cyu.edu.tw

Received February 15, 2012; revised March 18, 2012; accepted March 30, 2012

ABSTRACT

A novel framework for remote service discovery and access of IP cameras with Network address Translation (NAT) traversal is presented in this paper. The proposed protocol, termed STDP (Service Trader Discovery Protocol), is a hybrid combination of Zeroconf and SIP (Session Initial Protocol). The Zeroconf is adopted for the discovery and/or publication of local services; whereas, the SIP is used for the delivery of local services to the remote nodes. In addition, both the SIP-ALG (Application Layer Gateway) and UPnP (Universal Plug and Play)-IGD (Internet Gateway Device) protocols are used for NAT traversal. The proposed framework is well-suited for high mobility applications where the fast deployment and low administration efforts of IP cameras are desired.

Keywords: IP Camera (IP CAM); Network Address Translation (NAT); Session Initial Protocol (SIP)

1. Introduction

An IP camera (IP CAM) [1-3] is a video camera that can be directly connected to the internet without the need for a separate computer. It contains a hardware video encoder for realtime compression of captured video sequences. It also has a built-in web server, which provides the ability for accessing digital images and configuring the camera. The camera can be easily integrated with a wide range of applications, including e-surveillance, web attractions and remote monitoring.

For applications network security is an important concern, the deployment of IP CAMs in the network address translation (NAT) [4] environments with dynamic locations are usually desired. However, without a static IP address information, accessing the web server associated with the IP CAMs will be difficult. Moreover, for a service consumer, it may not be possible to always have a complete overview over the availability of IP CAMs in an application. This is particular true when large number of IP CAMs are employed in the application. Without a protocol providing IP CAM service information, the effectiveness of IP CAMs for internet applications may be limited.

Many service discovery protocols, such as SLP (Service Location Protocol) [5], Jini [6], UPnP (Universal Plug-and-Play) [7], and Zeroconf [8,9], can be adopted for solving the problems. In the service discovery environments, IP CAMs and other devices advertise themselves, supplying details about their capabilities and the information one must know to access the service (e.g., the IP address). Nevertheless, existing service discovery protocols are limited only to local area networks (LANs). Some service discovery protocols are also not able to provide functions for NAT traversal. The goal of this paper therefore is to present a novel service discovery protocol for remote access of IP CAMs with NAT traversal.

The proposed protocol, termed STDP (Service Trader Discovery Protocol), is a hybrid combination of the SIP (Session Initiation Protocol) [10-12] and Zeroconf protocols. SIP is a protocol developed by IETF to assist in providing advanced telephony services across the internet. Basically it is a signaling protocol used for establishing sessions in an IP network. In the SIP, location of clients are maintained and updated in the registrar server. The IP address of the target node can be obtained by a query to the server. Although a direct deployment of SIP to an IP CAM is possible for accessing digital images, a number of modifications are desired. For many home network applications, costly manual pre-configurations should be avoided. However, the deployment of SIP requires the assignment of an unique pair of SIP URI and password to each IP CAM. This may result in a high manual pre-configuration cost when the number of IP CAM is large.

To reduce the pre-configuration cost, the Zeroconf

*Corresponding author.

protocol is employed in the STDP. The Zeroconf protocol is a light weight protocol supporting service discovery in a LAN. It operates without any kind of manual pre-configuration. As compared with other service discovery protocols, it imposes minimal implementation cost for an embedded system. The protocol therefore is well-suited for the IP CAMs.

In the proposed STDP, the SIP is required to be deployed only on a single node, termed trader, in the LAN. This assures the minimal pre-configuration cost for the system. The trader is responsible for collecting the service information provided by all the other nodes in LAN via the Zeroconf protocol. A remote access to any IP CAM in the LAN can be accomplished by first retrieving the service information from the trader using the SIP. Based on the information, the IP address of any IP CAM in the LAN can be found. A remote node can then access the web server associated with the target IP CAM based on the retrieved service information.

The employment of NAT may also be desired in STDP for network security enhancement. Nevertheless, the NAT traversal is required so that local nodes can be visible from remote nodes for service discovery. One simple way to accomplish this is by the employment of UPnP (Universal Plug and Play)-IGD (Internet Gateway Device) [13], where the gateway device will open a tunnel for each local node upon request. Although UPnP-IGD is simple to implement, the NAT mapping stored in the gateway device is subject to potential attack on internet. Since the trader in a LAN contains all the service information in the LAN, the exposure of trader is equivalent to the exposure of all nodes in the LAN.

Therefore, both UPnP-IGD and SIP-ALG (Application Layer Gateway) [14] protocols are used for NAT traversal in the STDP. All the local nodes other than trader use UPnP-IGD to open tunnels for remote access. Only the trader adopts SIP-ALG for NAT traversal. As a result, the trader is visible only to SIP servers. The security for the STDP-based network can then be effectively enhanced.

The proposed STDP protocol has been implemented in a dynamic network environment with NAT. Physical tests reveal that the IP CAMs supporting only simple Zeroconf and UPnP-IGD protocols can be easily accessed by a remote host. The proposed STDP protocol is therefore beneficial for a wide range of IP CAM applications requiring NAT and dynamic deployment.

2. Preliminaries

The proposed STDP is a hybrid combination of SIP and Zeroconf. Therefore, in this section, we give a brief description of these two protocols. The independent applications of these two protocols for accessing IP CAMs are also discussed.

2.1. SIP

SIP is a signaling protocol used for establishing sessions in an IP network. The user agents and servers are the major components of the protocol. A user agent is an end-user device. A user agent client (UAC) issues a request and a user agent server (UAS) responds to the request. When the SIP is applied for the remote access of an IP CAM, in the simplest form the UAC is a viewer and the UAS is the IP CAM, as shown in **Figure 1**. In this case, the location of the IP CAM should be fixed, and should be known to the viewer.

To support the mobility for the IP CAM, the employment of SIP servers are necessary. Commonly used SIP servers include the registrar and proxy server. A SIP registrar includes the registrar and proxy server. A SIP registrar databases (termed location server) containing user agent locations. A SIP proxy server can be viewed as the router in the SIP level that forward SIP requests and responses. In addition, it provides functions for authentication and authorization.

Figure 2 shows a simple example, which uses proxy, registrar and location servers with the INVITE message for session establishment. As shown in the Figure, an IP CAM first registers its location in the location server. A SIP proxy server then accepts an INVITE request made by a UAC and queries location server to find UAS location. Based on the address received from the server, the proxy server forwards the INVITE message to the UAS. The session will then be established after the acknowledgements from UAS are received. It can be observed from the example that the viewer does not have to know the IP CAM location prior to a connection establishment. In addition, the UAS is allowed to change its location without informing the UAC. Only a registration request

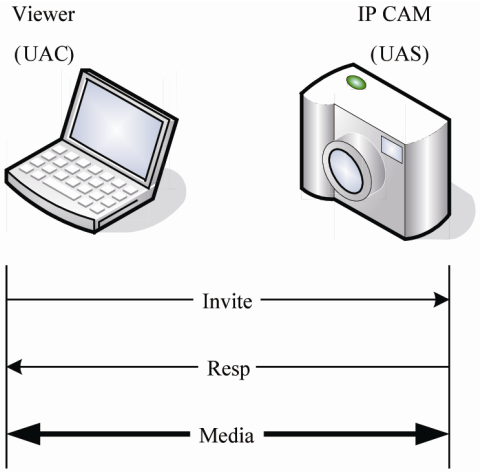


Figure 1. Basic application of SIP for IP CAM.

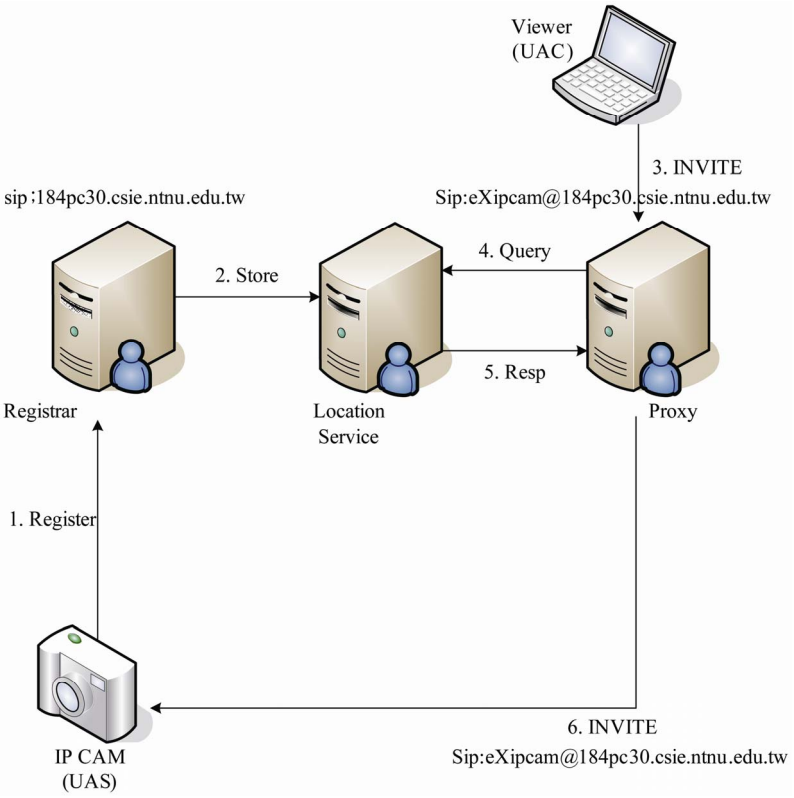


Figure 2. A simple SIP-based framework for remotely accessing IP CAM.

to the registrar server for location updating is required.

In addition to supporting the user mobility, the SIP offers the event notification framework [15,16], which uses SUBSCRIBE/NOTIFY messages for subscribing to, and receiving notifications of, SIP-related events within SIP networks. The ability to request asynchronous notification of events proves useful in many services for which cooperation between devices is required. Examples of such services for IP phone applications include automatic callback services (based on terminal state events), buddy lists (based on user presence events) and message waiting indications (based on mailbox state change events).

The SIP allows the remote access of IP CAM with mobility support and event notification. To use the SIP, however, each IP CAM should be associated with a pair of SIP URI and password. The high manual pre-configuration cost and administration efforts for the deployment of IP CAMs are therefore necessary. This is undesirable for many IP CAM applications.

2.2. Zeroconf

Zeroconf is a protocol for discovering services available in a local network. A Zeroconf network is one that can exist without a central control component, and works without any kind of manual pre-configuration.

Zeroconf can directly be adopted for discovering IP CAM in the LAN. It involves address assignment, name translation and service discovery without central servers. The address assignment for a node in Zeroconf network can simply be accomplished by randomly selecting an address in the range of 169.254.1.0 to 169.254.254.255. The node then does an ARP probe for the address. If there are any responses, the node chooses another IP address at random, and tries the ARP probe again.

The name translation in Zeroconf network is solved by the multicast DNS (mDNS) standard, which eliminates the requirement for DNS server. In the standard, all the nodes in the LAN listens to a specific IP multicast address. A node wish to publish a name will broadcast the selected name to this multicast address. Other nodes having the same name then reply to the requesting node. The name translation can be accomplished in a similar fashion. Instead of using fully qualified domain name (FQDN), a node name in the .local name space is used for mDNS.

Another standard, termed DNS Service Discovery (DNS-SD) can be used for the service discovery in Zeroconf. DNS-SD works particularly well with mDNS, since it also uses DNS records. Three basic operations are included in the DNS-SD: publication, discovery and resolution. The goal of publication is to advertise a service. The discovery operation is used to browse for available service. Based on the results of discovery op-

eration, the resolution operation is adopted for translating service names to addresses and port numbers.

Figures 3, **4** and **5** show a simple example of these DNS-SD operations for a local network consisting of an IP CAM. The publication operations of Zeroconf are shown in **Figure 3**, which consists of address selection (**Figure 3(a)**), name selection (**Figure 3(b)**), service start up (**Figure 3(c)**) and service broadcast (**Figure 3(d)**). In **Figure 3(a)**, the IP CAM randomly selects the IP address 169.254.0.1, and announces it to the network. Because no devices respond to the announcement, the IP CAM takes the address as its own. In **Figure 3(b)**, it starts up its own multicast DNS responder, requests the host name ipcam. local, verifies its availability, and takes the name as its own. In **Figure 3(c)**, the IP CAM starts up a video service on TCP port 80. Finally, in **Figure 3(d)**, it publishes the service instance, of type _http._tcp, under the name IP CAM, in the .local domain. It should be noted that the service type (*i.e.*, _http._tcp) contains two fields: the first field (*i.e.*, _http) is service dependent, and the second

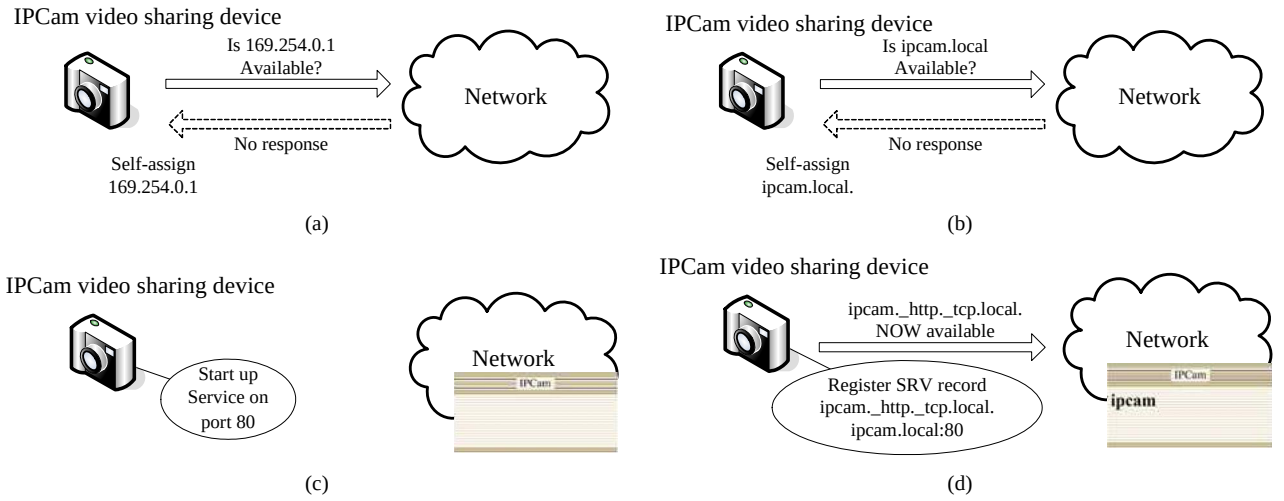


Figure 3. Publication operations of Zeroconf. (a) Address selection; (b) Name selection; (c) Service startup; (d) Service publication.

Figure 4. Service query and discovery in Zeroconf. (a) Query by service type; (b) Response.

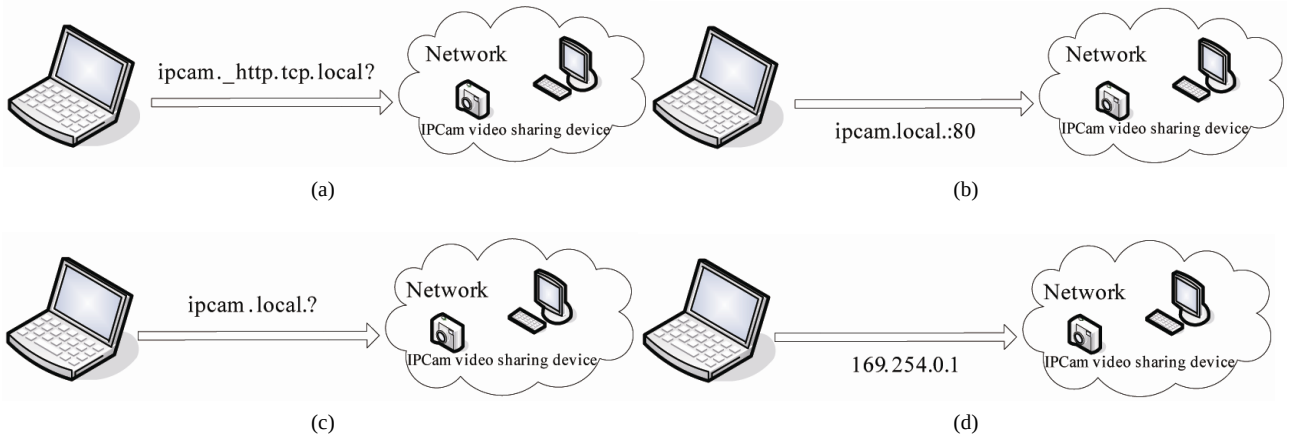


Figure 5. The query for domain name, port and IP address in Zeroconf. (a) Request domain name and port for instance name; (b) Receive domain name and port; (c) Request IP address for domain name; (d) Receive IP address.

field (*i.e.*, _tcp) indicates the transportation protocol used by the service. The service type will be used for service browsing and discovery. The instance name (*i.e.*, IP CAM) is device dependent. That is, devices sharing the same service type will have different instance names. The service instance therefore can be used for the query of port and IP address of a device.

Figure 4 depicts the service query and discovery in the Zeroconf network. In this example, the service type queried by the viewer shown in **Figure 4(a)** is _http._tcp. The service instance discovered from the network is ipcam._http._tcp.local, which represents an IP CAM. Based on the service instance, the viewer can further query for the port, domain name and IP address of the IP CAM using resolution operation, as shown in **Figure 5**.

Although Zeroconf requires no pre-configuration cost, it has the major drawback that the protocol can only be used in a local network. For IP CAM applications, however, remote accesses are usually desired.

3. STDP

The goal of STDP is to eliminate the drawbacks of accessing IP CAMs based only on SIP or Zeroconf protocols. It provides remote access of IP CAM with minimal pre-configuration cost. As shown in **Figure 6**, the STDP is an application layer control protocol that utilizes both SIP and Zeroconf. A STDP-based network contains three basic components: service provider, service requester, and service trader. In our design, the service provider and requester are an IP CAM and a viewer, respectively. Although the primary goal of the STDP is for the design of IP CAM systems, the STDP apply equally well to the broader group, where the service provider and service requester can be any networked appliances demanding low pre-configuration cost and efficient remote access.

The service traders are the nodes used for the delivery of service information over WAN. A service trader provides two functions. It can be adopted to collect/discover service information from service providers in a local network, and deliver the information to a remote node (which is also a trader) upon requests. Alternatively, it can also be used to subscribe and receive the service information from other traders in remote sites, and publish the service information to the service requesters within its local domain. A trader can be implemented in an independent device such as a computer. It can also be implemented in an IP CAM (or a viewer). In these cases, the device supports multiple roles as a trader and a service provider (or a requester).

The communications between two service traders are based on SIP protocol, as shown in **Figure 7**. Each trader can be an UAC and/or an UAS. Each local network needs only one service trader. Each of the service providers and requesters talk to its trader in the same LAN for the delivery of local service information. From **Figure 7**, we observe that the Zeroconf is adopted for the communication between a trader and a service requester (or a provider). Therefore, in our design, the service provider and requester need to support Zeroconf protocol.

To obtain information from a service provider to a service requester, both the SIP and Zeroconf protocols are used. The STDP provides a mechanism for the service information exchange between the SIP and Zeroconf. Based on the acquired service information, the viewer then can access the IP CAM using the HTTP protocol.

To discuss the STDP protocol in more detail, we divide the protocol into three parts, as depicted in **Figure 8**. The first part concerns with the communication between a trader and a service provider. It can be observed from **Figure 8(a)** that the trader will receive service information published by an IP CAM. The trader can also actively discover the service provided by an IP CAM. All the publish and discovery operations are based on Zeroconf protocol, which are illustrated in **Figures 3-5**.

The second part of the STDP protocol focuses on the interactions between traders. This part of the protocol is based on the SIP. Service traders accompanied by service providers are the UASs in the SIP. An UAS discovers/collects local services available, and delivers the service information to other UACs upon request. An UAC is the

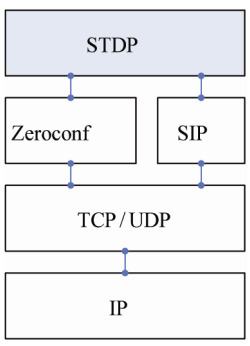


Figure 6. The protocol stack of STDP-based networks.

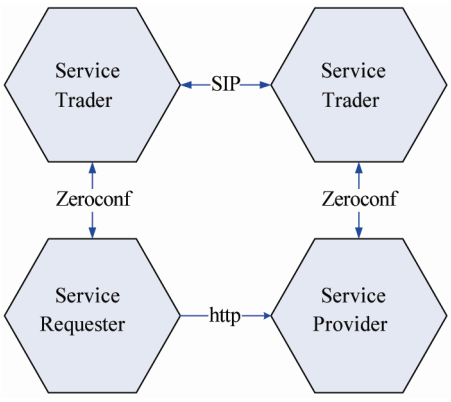


Figure 7. STDP topology.

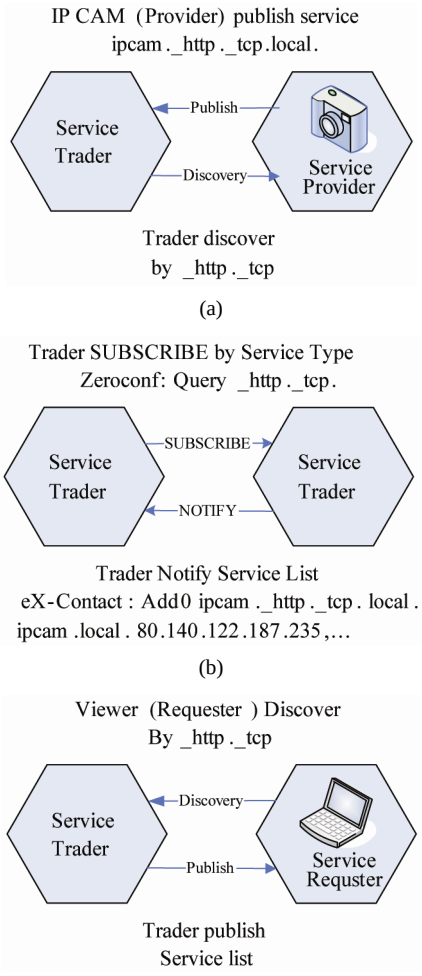


Figure 8. STDP protocol messages. (a) Communication between trader and provider; (b) Traders communication; (c) Communication between trader and requester.

service traders accompanied by service requesters. It sends subscription requests to UASs for acquiring the service information. Once the UAC obtains service notifications from UASs, it publishes the service information to its own service requesters.

In the STDP, the SIP SUBSCRIBE/NOTIFY messages are used for the service information delivery between an UAC and an UAS, as shown in **Figure 8(b)**. In the SIP, the original goal of SUBSCRIBE/NOTIFY messages is to provide the SIP related events subscriptions and notifications. The STDP extends the usage of SUBSCRIBE/NOTIFY for the service subscription and notification.

To use the SUBSCRIBE message for service subscription, the type of services desired should be specified in the message header. Here we augment a field (termed Zeroconf) in the header of SUBSCRIBE message for specifying the service type. The format of service type follows the DNS-SD format as _http._tcp, as depicted in **Figure 9(a)**.

NOTIFY messages are sent to inform traders of the service available for which the traders have a subscription. Subscriptions are established using the SUBSCRIBE method described above. Sending a NOTIFY message does not terminate the corresponding subscription. A single SUBSCRIBE request may trigger several NOTIFY messages. In each NOTIFY message, the list of services and the IP address of the corresponding service providers are carried. We also augment two fields (termed Zeroconf and eX-Contact) in the header of NOTIFY message to achieve this objective. It can be observed form **Figure 9(b)** that the Zeroconf field indicates the service type this message response to. The eXContact field contains 5 items: action, service instance, host name, port number and IP address.

The action item instructs how the service instance included in field should be handled. There are three actions: addition (denoted by Add), deletion (denoted by Del), and updating (denoted by Upd). The addition action instructs the target service trader to add the service instance to the service list. The deletion action informs the target service trader to remove the service instance. The update action directs the target trader to update the attributes of the service instance. The attributes considered here include the video coding standard adopted by the IP CAM, frame size and frame rate.

Use **Figure 9(b)** as an example, the NOTIFY message

SUBSCRIBE SIP URL of Trader A
From: SIP URL of Trader B
To: SIP URI of Trader A
Contact: SIP URO of Trader B
Call-ID: call identifier
CSeq: sequence number SUBSCRIBE
Event:presence
Expires:seconds until SUBSCRIBE expires
Allow-Events: presence, refer
Zeroconf: Query _http._tcp
Content-Length: 0

(a)

NOTIFY SIP URI of Trader B
From: SIP URI of Trader A
To: SIP URI of Trader B
Contact: SIP URI of Trader A
Call-ID: call Identifier
CSeq: sequence number NOTIFY
Event: presence
Subscription-State: active;expires=seconds until SUBSCRIBE expires
Allow-Events: presence, refer
Zeroconf: Response _http._tcp
eX-Contact: Add0 ipcam._http._tcp.local. ipcam.locol. 80 140.122.184.235,
Content-Length: …

(SDP not show)

(b)

Figure 9. Extensions of SUBSCRIBE/NOTIFY messages for STDP service subscription and notification. (a) SUBSCRIBE message for service subscription; (b) NOTIFY message for service notification.

instructs the target trader to add the service instance ipcam._http._tcp.local, with host name ipcam.local, port number 80, and IP address 140.122.184.235, to its service list. It should be noted that the service instance and domain name should also follows the DNS-DS format in the STDP.

The final part of STDP describes the communications between a trader and a service requester, which is also based on Zeroconf. It can be observed from **Figure 8(c)** that the trader will then publish the service information collected from other traders to the service requester. The service requester may also actively discover the service information from the trader.

Three parts of the STDP protocol depicted in **Figure 8** may operate independently. That is, the SIP and Zeroconf protocols are not required to operate at a pre-specified order in the STDP. **Figure 10** shows two examples of STDP message flows. For the sake of brevity, only two LANs are considered in each example. Nevertheless, the message flows can easily be extended to the scenarios containing large number of LANs. As shown in **Figure 10**, LAN A in each example contains a service requester and a trader (termed Trader 1). LAN B consists of two service providers (termed Service Provider 1 and Service Provider 2) and a service trader (termed Trader 2).

Figure 10(a) illustrates the scenario, in which Trader 1 and Trader 2 first find their own service requester and service providers via PUBLISH/DISCOVERY messages. The service information of the service providers is then delivered from LAN B to LAN A via SUBSCRIBE/NOTIFY messages. After receiving the service information, Trader 1 then publishes this information to the Service Requester. As shown in **Figure 10(a)**, after the Service Requester received the service information, it selects the Service Provider 2 as its target device. The Service Requester then issues directly a service request via HTTP protocol to Service Provider 2. Direct video delivery from Service Provider 2 to Service Requester then follows.

For the scenario shown in **Figure 10(b)**, it is assumed that the Service Requester and Service Providers are not online at the beginning. The communication between Traders 1 and 2 is first established. This is then followed by a series of service information updating/notification when service providers and service requester become available. Similar to the case shown in **Figure 10(a)**, the Service Requester finally selects the Service Provider 2 as its target device for the IP CAM service.

As shown in **Figure 7**, the service trader plays a major role in STDP. It connects different local networks, and operates in back-to-back mode. It acts as a SIP user agent on one side, and as a Zeroconf end device on the other side. A trader has 4 operations. To further elaborate these operations, **Figure 11** depicts their flowchart in detail.

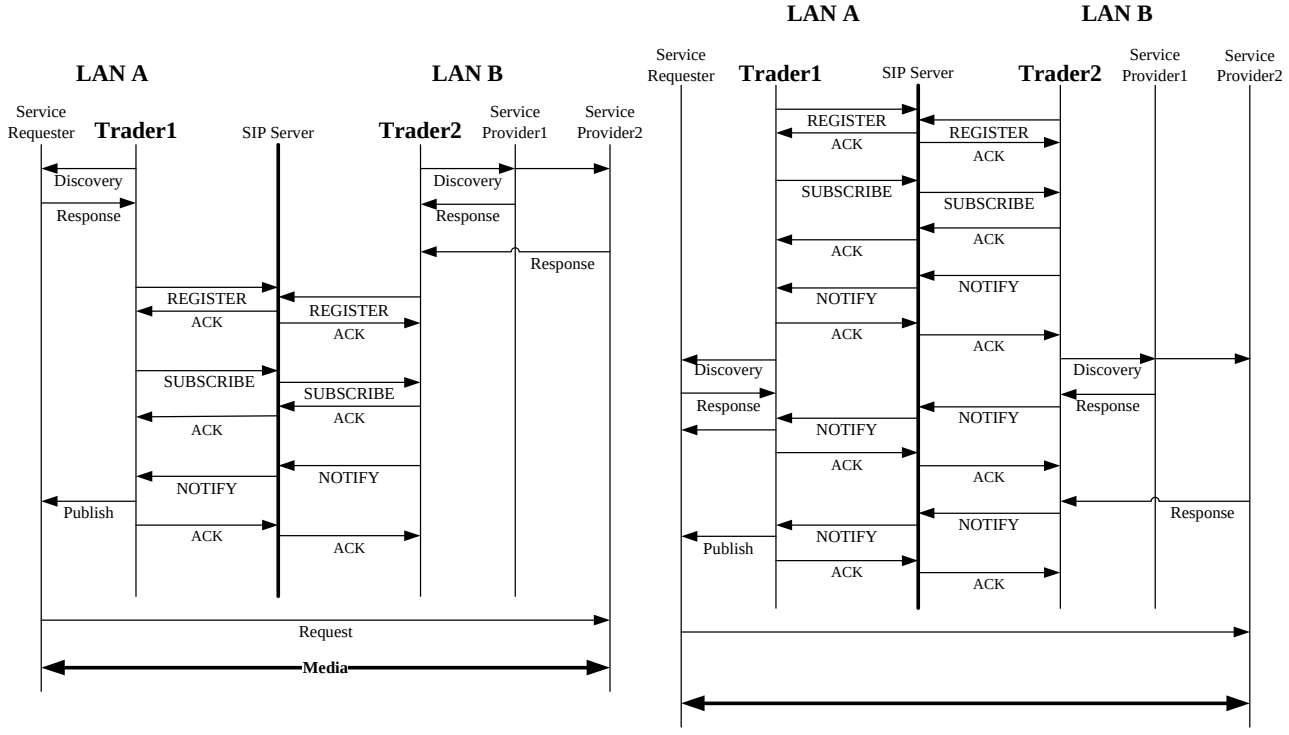


Figure 10. Two examples of STDP message flow: (a) Trader 1 and Trader 2 first find their own service requester and service providers via PUBLISH/DISCOVERY messages. (b) Trader 1 and Trader 2 first establish their connection via SUBSCRIBE/NOTIFY messages.

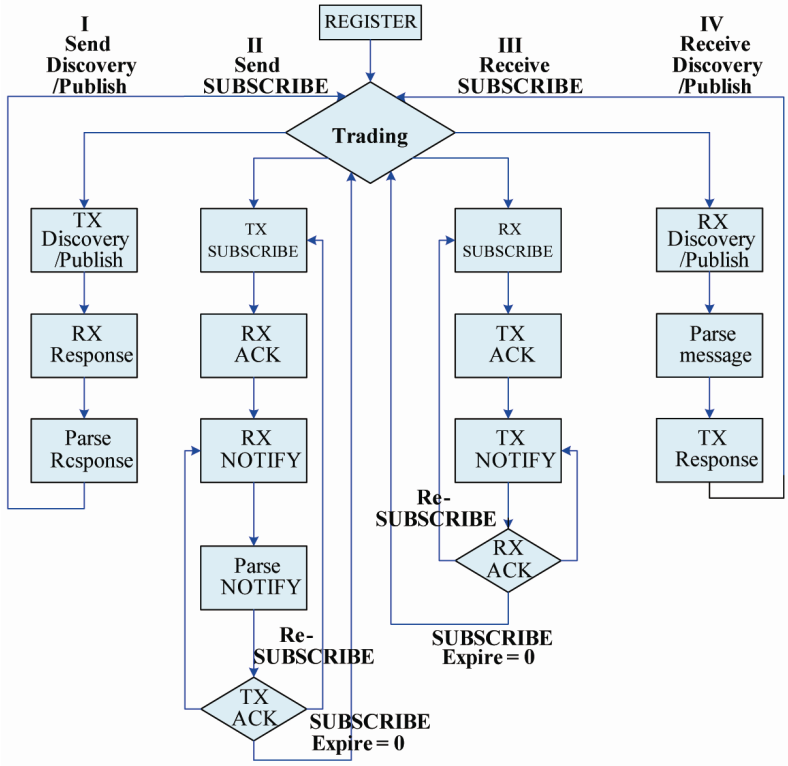


Figure 11. Operation flowchart of a trader.

The first operation is to send Discovery or Publish messages. In this operation, the trader acts as a Zeroconf end device searching or publishing the services available. In the second operation, the trader acts as a SIP UAC, and send SUBSCRIBE message for triggering the SIP event notification mechanism. After sending the SUBSCRIBE message, the trader will then waits and receives one or more NOTIFY messages for updating the service list. The trader behaves as a SIP UAS in the third operation, which receives the SIP SUBSCRIBE message. The trader will then send one or more NOTIFY messages to the subscribing node. The fourth operation receives the Discovery or Publish messages, where the trader functions again as a Zeroconf end device. This operation and the first operation are essential for a trader to acquire the service available from the service provider, or deliver the service to the service requester.

Note that the SIP is required to be installed in the service traders because of the operations of SUBSCRIBE/NOTIFY. Only one node in a local network needs to be the service trader. For the other nodes, only the implementation of Zeroconf is necessary. The employment of Zeroconf protocol can effectively reduce the pre-configuration efforts, because the protocol allows the simple plug-and-play. By contrast, the SIP devices require the assignments of SIP URI and password. For the stand alone embedded systems such as IP CAMs, the assignments may require considerable efforts especially when the number of IP CAMs is large. The STDP therefore provides an effective approach for lowering pre-configuration cost and providing remote access.

4. STDP with Nat Traversal

The employment of NAT may also be desired in STDP for network security enhancement. Nevertheless, the NAT traversal is required so that local nodes can be visible from remote nodes for service discovery. To see this fact in more detail, we first note that the proposed STDP employs HTTP/TCP for video streaming. Consequently, for a STDP-based system without NAT traversal, if a client wants to make a directly TCP connection to a IP CAM, the IP address of the IP CAM should be transmitted by service trader first. However, if the IP CAM is deployed in the residential environment behind a NAT, the information contained in STDP messages would be incorrect since the IP address of the IP CAM is private address.

To solve the problem, the UPnP-IGD protocol is adopted in our scheme. It has been found that UPnP-IGD is an effective solution for NAT traversal because of its simplicity. **Figure 12** shows that how UPnP-IGD-aware NAT works for NAT traversal. At first, the NAT joins in the multicast group 239.255.255.250 and listens on port 1900 for the request issued by a client. When the NAT receives a request, it will add the corresponding port mapping into its mapping table. The NAT subsequently

returns the public IP address and the allocated port to the client. After that, the remote host in the public network can connect to the client directly through the NAT.

To integrate the UPnP-IGD with the STDP, an IP CAM will sends a port mapping request to NAT when it is in the private network (*i.e.*, behind a NAT). It then publishes its service with the port mapping information via Zeroconf. As a result, the STDP messages will have correct IP address and port number. The IP CAM is then accessible from the public networks.

Although UPnP-IGD is simple to implement, the NAT mapping stored in the gateway device is subject to potential attack on internet. Since the trader in a LAN contains all the service information in the LAN, the exposure of trader is equivalent to the exposure of all nodes in the LAN.

Therefore, both UPnP-IGD and SIP-ALG protocols are used for NAT traversal in our design. All the local nodes other than trader use UPnP-IGD to open tunnels for remote access. Because the traders are actually the UACs or UASs in the SIP, the usual SIP-ALG protocol can be effectively used for the NAT traversal. In this way, the trader is visible only to SIP servers. The security for the STDP-based network can then be effectively enhanced.

Figure 13 shows an example of STDP system with NAT traversal. The system consists of an IP CAM, and IE browsers, traders, a SIP server, and two NATs. Note that, the IP CAM and IE browser are deployed in the residential environment with NAT, as shown in the **Figure 13**.

Note that the IP CAM with private IP addresses is only available in LAN B. The IP CAM has to send the port mapping request to NAT B via UPnP-IGD to be accessible from the public networks. NAT B receives the request and subsequently adds the port mapping into its mapping table. As a result, NAT B opens tunnels for the IP CAM, and returns the information of the public address and the ports to the IP CAM. After acquiring the responses from NAT B, the IP CAM is now available for remote access.

After finishing port mapping, the IP CAM publication services with their public address and ports (e.g., IP address 140.122.184.26, port 3000) in LAN B. This information can also be actively discovered by trader B. Thus trader B will have the complete service information in LAN B. After that, trader B, as an SIP UAS, connects to the SIP server and waits for SUBSCRIBE message from trader A.

Recall that for the basic STDP, trader A sends SUB-

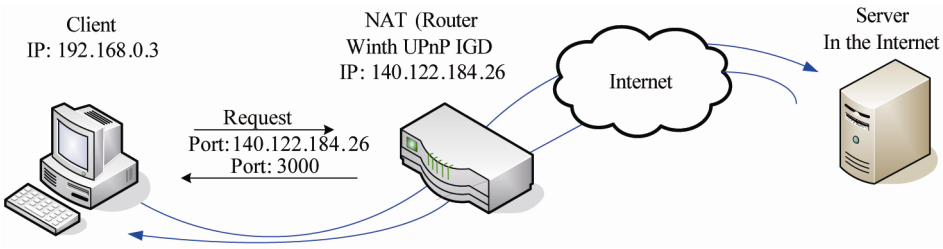


Figure 12. The NAT traversal with UPnP-IGD.

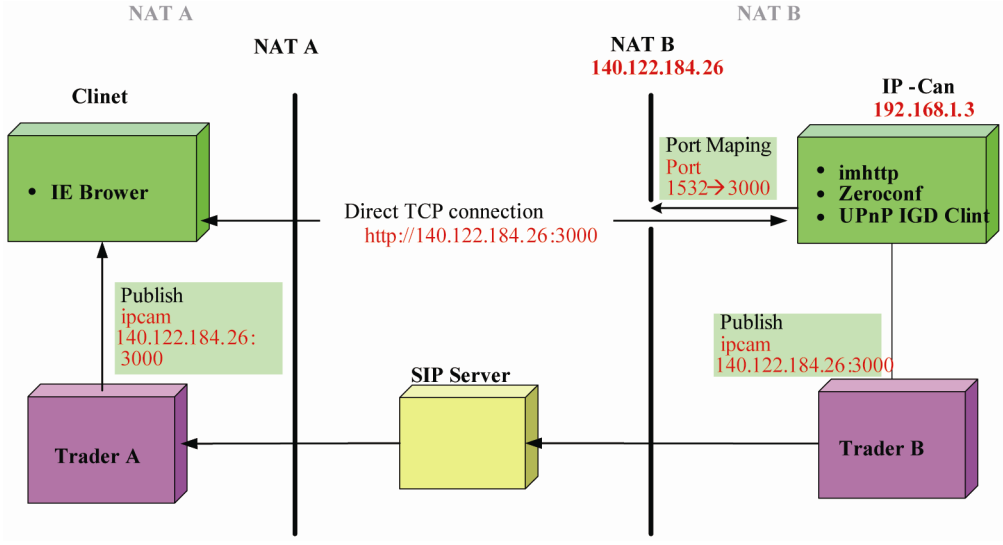


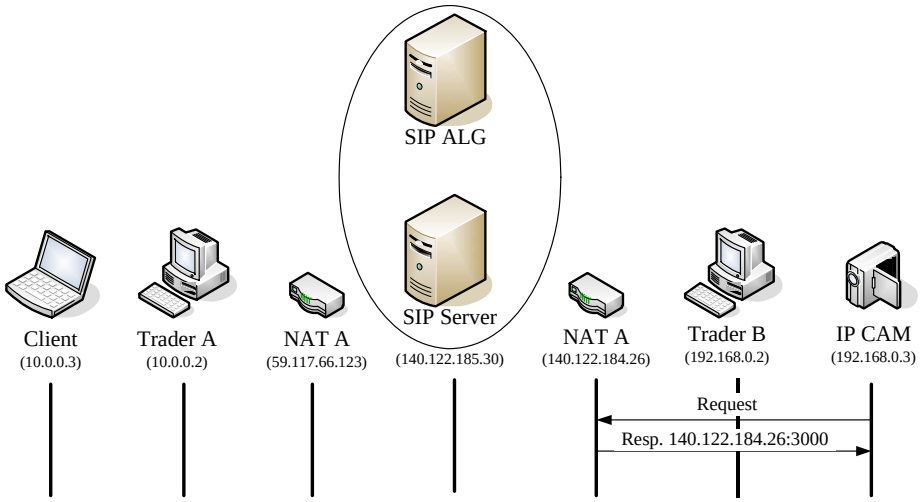
Figure 13. An example of STDP system with NAT traversal.

SCRIBE to trader B, and then trader B returns NOTIFY for the service information delivery over WAN. All the service information delivery is based on SIP protocol. Since the address information is in the packet payload in SIP, the SIP-ALG will be used for the NAT traversal of traders A and B in our design.

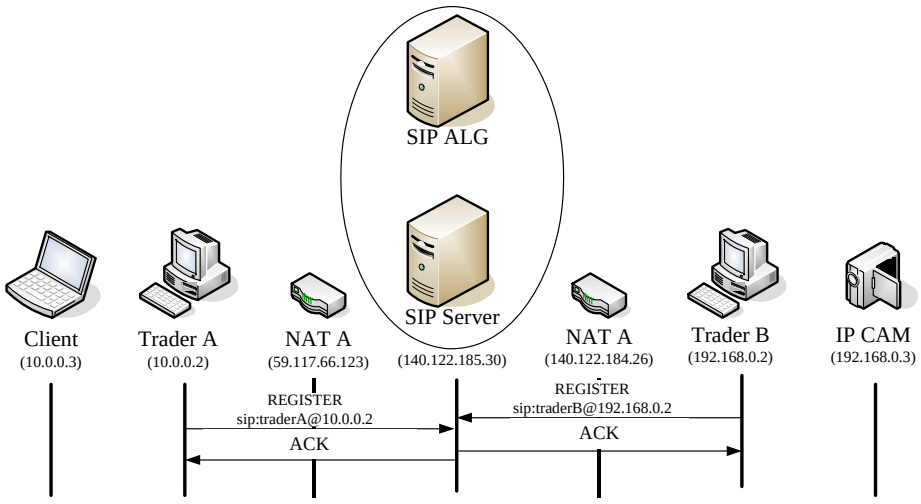
Consequently, for the NAT traversal, both SIP ALG and UPnP-IGD are adopted. **Figure 14** illustrates this fact in more detail. As shown in the Figure, the IP CAM first sends a port mapping request to NAT B via UPnP-IGD messages. The NAT B then opens a tunnel for TCP connection to the IP CAM, as shown in **Figure 14(a)**. In **Figure 14(b)**, trader A and trader B register themselves to SIP Server as UAC and UAS, respectively. The SIP server then activates the SIP-ALG protocol to process all the messages for NAT traversal. This allows trader A and trader B identify their own service requester and service providers via service discovery/publish operations, as shown in **Figure 14(c)**. Finally, the service information of the IP CAM is delivered from LAN B to LAN A via SUBSCRIBE/NOTIFY messages, as depicted in **Figure 14(d)**. After receiving the service information, trader A then publishes the information to the client. As shown in **Figure 14(d)**, the client acquires the service information, and it issues directly a service request via HTTP protocol to the IP CAM. Direct video delivery from the IP CAM to the client then follows.

5. Experimental Results

The STDP protocol has been implemented in a test-bed that realizes the scenario proposed in **Figure 15**. Similar to **Figure 13**, the scenario consists of two local networks (termed LAN A and LAN B in the Figure). LAN A consists of a number of laser printers, one IE browser and 2 IP CAMs. LAN B contains a number of laser printers, one personal computer and one IE browser. As shown in the Figure, personal computers serve as the



(a)



(b)

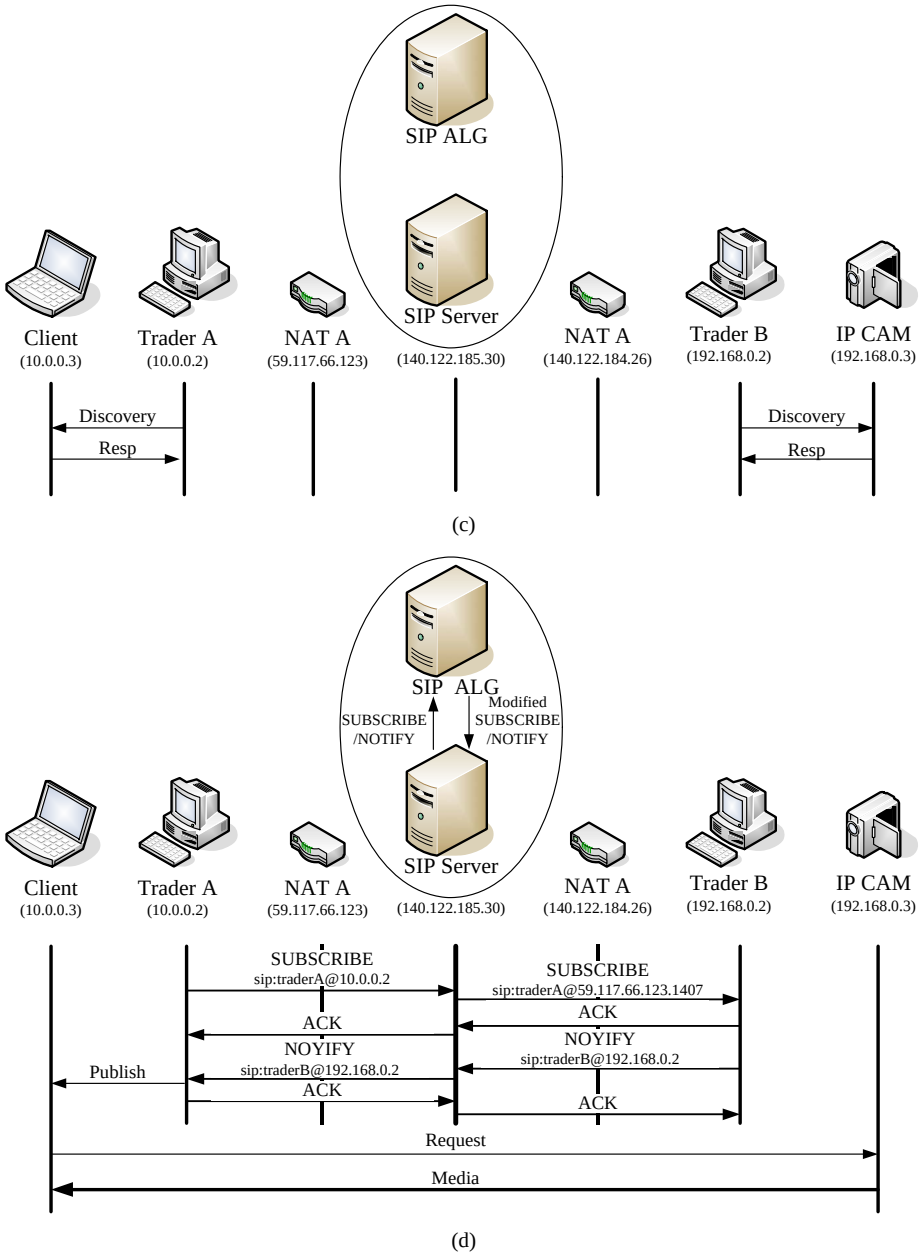


Figure 14. The message of STDP with NAT traversal: (a) IP CAM requests NAT B to open a tunnel for TCP connection; (b) Traders A and B register to SIP server with the private addresses; (c)Traders A and B find their own service providers and service requester; (d) Traders A and B establish their connection.

service trader in the LAN A and LAN B, respectively. The IE browser in each local network is the service requester in that local network. All the IP CAMs and laser printers in both LANs are the service providers. They are all of the type _http._tcp. Their IP addresses are dynamically assigned by a DHCP server. The service providers publish their service with their unique hostname. Only the traders in relative local networks require the manual pre-configuration, because the assignments of SIP URI and password are necessary to register to SIP server. Note that, in addition to the IP CAMs, the laser printers are included in this scenario. Since the laser printers supports Zeroconf, detecting the service provided by the laser printers in different local networks demonstrates the fact that the proposed protocol can be adopted for the discovery of services provided by various Zeroconf-based devices.

In our experiment, a reference design kit (RDK) based on a 200-MHz ARM 920 CPU and an MPEG4 encoder ASIC is used for the IP CAM design. The employment of MPEG4 ASIC allows the source video sequence to be encoded in real-time. Both the wired and wireless LAN

interfaces (*i.e.*, 802.3 and 802.11) are also available in the IP CAMs. The operating system of the IP CAMs is Linux. The Bonjour software development kit (SDK) [17] is adopted for the Zeroconf implementation in the IP CAM. Moreover, we adopt the Bonjour SDK and PJSIP [18] library for implementing the service trader.

Customarily, the NAT functions are accomplished by a router in the residential environment. Therefdore, we choose the routers supporting UPnP-IGD protocol in our experiments. Since the IE browser is used as the service requester in each LAN, it is necessary for the browser to support the Zeroconf. In our implementation, the Bonjour plug-in is adopted, which is able to discover the services of type _http._tcp.

Figure 16 shows all the services of type _http._tcp discovered by the IE browser in LAN A without the employment of STDP. It can be observed from the Figure that these services are actually the services provided by the laser printers in LAN A. Because all the devices in LAN A and LAN B are behind the NAT, without the proper NAT traversal schemes, the IE browser in LAN A is still not able to discover the services in LAN B even the STDP is employed.

When SIP-ALG is employed, the IE browser is able to

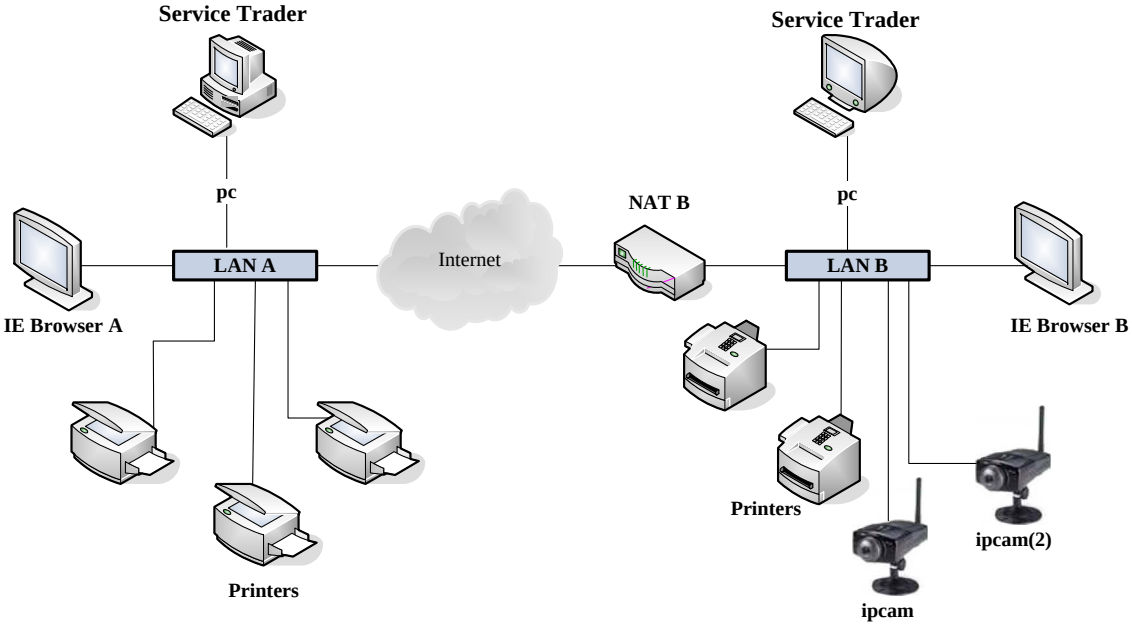


Figure 15. The scenario for our experiment.

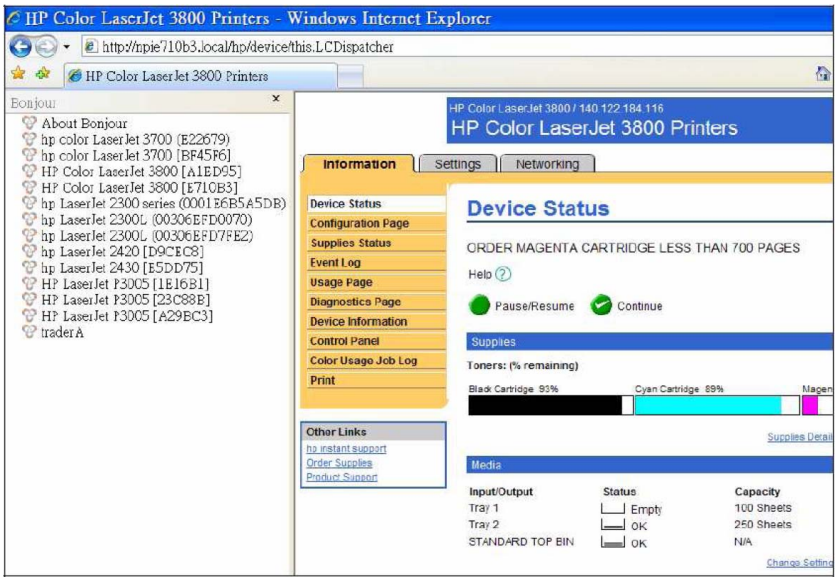


Figure 16. All the services of type _http._tcp discovered in LAN A without the employment of STDP.

discover services in LAN B, as shown in **Figure 17**. However, without the employment of UPnP-IGD, IP CAMs in LAN B still use private IP addresses. This implies the IE browser in LAN A is not able to access the IP CAMs in LAN B, as shown in **Figure 17**.

Figure 18 shows the results when the MiniUPnP-IGD [19] client is implemented in the IP CAMs in LAN B so that both SIP-ALG and UPnP-IGD are supported in our system. It can then be observed from the Figure that the IE browser is able to access the IP CAMs even when all the devices in both LANs are behind the NATs.

6. Conclusion Remarks

The proposed STDP protocol has been found to be effective for IP CAM applications. It allows both remote access and dynamic deployment of IP CAMs without the need of manual pre-configuration. In the STDP, the service lists from remote hosts are obtained by SIP SUBSCRIBE/NOTIFY event notification mechanism. The service discovery and publish in a local network are then based on Zeroconf protocol, which is also used for eliminating manual pre-configuration. When IP CAMs are deployed behind the NAT, both the UPnP-IGD and SIP-ALG protocols are adopted for NAT traversal. A test-bed verifying the STDP protocol with NAT traversal has been implemented. From the experiment, it is observed that a basic IE browser with Bonjour plug-in can be effectively used for the remote access of IP CAMs, which are installed with simple plug-and-play. All these facts demonstrate the effectiveness of the STDP.

Figure 17. All the services of type _http._tcp discovered in LAN B with the employment of STDP and SIP-ALG. The UPnP-IGD, however, is not adopted.

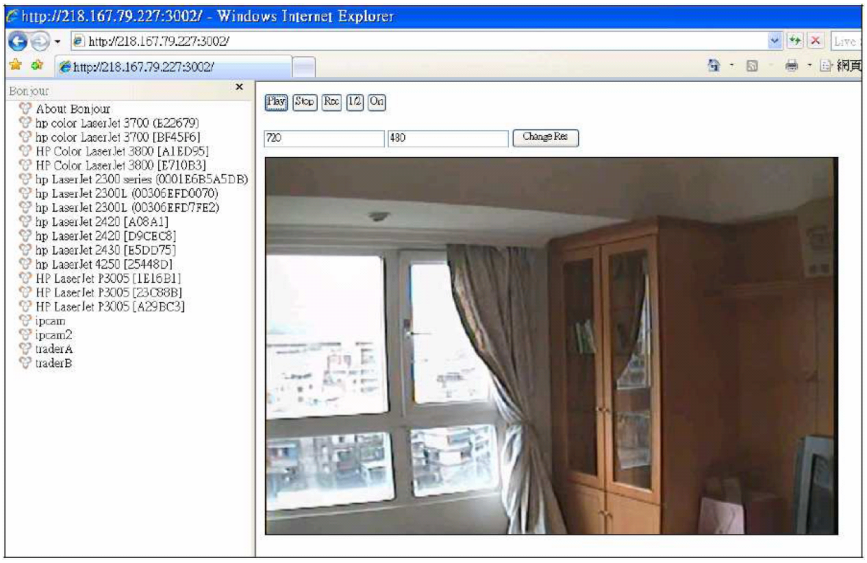


Figure 18. All the services of type _http._tcp discovered in LAN B with the employment of STDP, SIP-ALG and UPnP-IGD.

REFERENCES

[1] W. Hintermaier and E. Steinbach, "A system Architecture for IP Camera based Driver Assistance Applications," *IEEE Intelligent Vehicles Symposium* (*IV*), San Diego, 21-24 June 2010, pp. 540-547. doi:10.1109/IVS.2010.5548103

[2] N. A. Manap, G. Di Caterina, J. Soraghan, V. Sidharth and H. Yao, "Face Detection and Stereo Matching Algorithms for Smart Surveillance System with IP Cameras," 2010 2*nd European Workshop on Visual Information Processing European Workshop* (*EUVIP*), 2010, pp. 77-81. doi:10.1109/EUVIP.2010.5699107

[3] B. X. Li, W. H. Zhang, Z. H. Liu, M. J. Kang and S. Li, "Development and Implement of the IP Camera Based on DM6437," *IEEE Mechatronic Science, Electric Engineering and Computer International Conference* (*MEC*), Jilin, 19-22 August 2011, pp. 1961-1964. doi:10.1109/MEC.2011.6025872

[4] S. Guha and P. Francis, "Characterization and Measurement of TCP Traversal through NATs and Firewalls," *Internet Measurement Conference*, Berkeley, 19-21 October 2005, pp. 199-211.

[5] E. Guttman, C. Perkins, J. Veizades, "Service Location Protocol, Version 2," 1999.

[6] J. Waldo, "The Jini Specifications," 2nd Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, 2000.

[7] D. S. Kim, J. M. Lee, W. H. Kwon and I. K. Yuh, "Design and Implementation of Home Network Systems Using UPnP Middleware for Networked Appliances," *IEEE Transactions on Consumer Electronics*, Vol. 48, No. 4, 2002, pp. 963-972. doi:10.1109/TCE.2003.1196427

[8] S. Cheshire and D. H. Steinberg, "Zero Configuration Networking: The Definite Guide," O'Reilly Media, Inc., Sebastopol, 2005.

[9] E. Guttman, "Autoconfiguration for IP Networking: Enabling Local Communication," *IEEE Internet Computing*, Vol. 5, No. 3, 2001, pp. 81-86. doi:10.1109/4236.935181

[10] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "SIP: Session Initiation Protocol," 2002.

[11] W. Werapun, A. A. El Kalam, B. Paillassa and J. Fasson, "Solution Analysis for SIP Security Threats," *IEEE International Conference on Multimedia Computing and Systems* (*ICMCS*'09), Ouarzazate, 2-4 April 2009, pp. 174-180. doi:10.1109/MMCS.2009.5256707

[12] S.-W. Hsu, K.-D. Chang, C.-Y. Chen, H.-C. Chao and J.-L. Chen, "An Efficient Path-Migration Mechanism for IP Multimedia Subsystem," *IEEE Wireless Communications and Mobile Computing Conference* (*IWCMC*), Istanbul, 4-8 July 2011, pp. 1469-1474. doi:10.1109/IWCMC.2011.5982755

[13] UPnP Forum Official Website, "UPnP—Universal Plug and Play Internet Gateway Device v1.0," 2008. http://www.UPnP.org/standardizeddcps/documents/UPnP_IGD_1.0.zip.

[14] H. Kazuhito, S. Yuichi and S. Takaho, "A Study on Call State Model in SIP Application Level Gateway (SIP-ALG)," *IEIC Technical Report*, Vol. 103, No. 121, 2003, pp. 37-40.

[15] M. Rahman, D. Braun and D. Bushmitch, "A Framework to Access Networked Appliances in Wide Area Networks," *IEEE Consumer Communications and Networking Conference*, Princeton, 3-6 January 2005, pp. 261-266. doi:10.1109/CCNC.2005.1405180

[16] A. B. Roach, "Session Initiation Protocol (SIP)—Specific Event Notification," RFC 3265, 2002.

[17] Bonjour, "Aperture 2.1 SDK Overview," 2007. https://developer.apple.com/library/mac/#documentation/AppleApplications/Conceptual/AppleApp_Aperture_001/Overview/Overview.html#/

[18] PJSIP.ORG, "PJSIP-Open Source SIP Stack," 2008. http://www.pjsip.org/

[19] MiniUPnP Project, "MiniUPnP Project HomePage," 2008. http://miniupnp.free.fr/

