Prototype Model for Real Time Home Surveillance System over the Internet

Rathika Rajaravivarma
Computer Engineering Technology Department
New York City College of Technology
rrajaravivarma@citytech.cuny.edu

Abstract

Internet is dominating almost every aspect of our life. Internet applications are too many in today's business world. It is hard to imagine any office or home without a computer network. This paper focuses on a video Internet technology freely available in the Microsoft operating system. The purpose of this paper is to use this feature to design, test, and implement a home surveillance system that will contact its remote owner whenever an intruder is detected. The home surveillance system consists of a home computer connected to the internet that has a motion detector and web camera attached to it. The remote computer must be connected to the internet with the home owner's security features to check. The first part of this paper will focus the purpose of the home surveillance system and its application. Then the hardware and software design used to create the application plus the technologies used to support the software are discussed in detail. Next, the steps taken to implement the application as well as the limitations of the software are presented. Finally the results of the real time testing of the home surveillance system are summarized. In addition, the technologies discussed in this paper can be further expanded to design other types of applications for remote home monitoring and robotic applications.

Introduction

As the use of the Internet has grown, businesses and home applications have found that the Internet is an inexpensive way for mobile users and fixed sites to connect to the corporate and home network. Clearly it is more economical to connect over the Internet to a private network, than to pay for a leased line or lines to do so. One such application is to have a real time home surveillance system over the internet [1]-[2]. The home surveillance system as shown in Figure 1 consists of a home computer connected to the internet that has a motion detector and web camera attached to it. The home computer continuously monitors the motion detector. When the home computer detects motion it will automatically start a video conference call with a remote computer that the owner is currently at. The remote computer must be connected to the internet and the owner must be logged on to the IPTEL SIP (Session Initiation Protocol) Registrar Server.

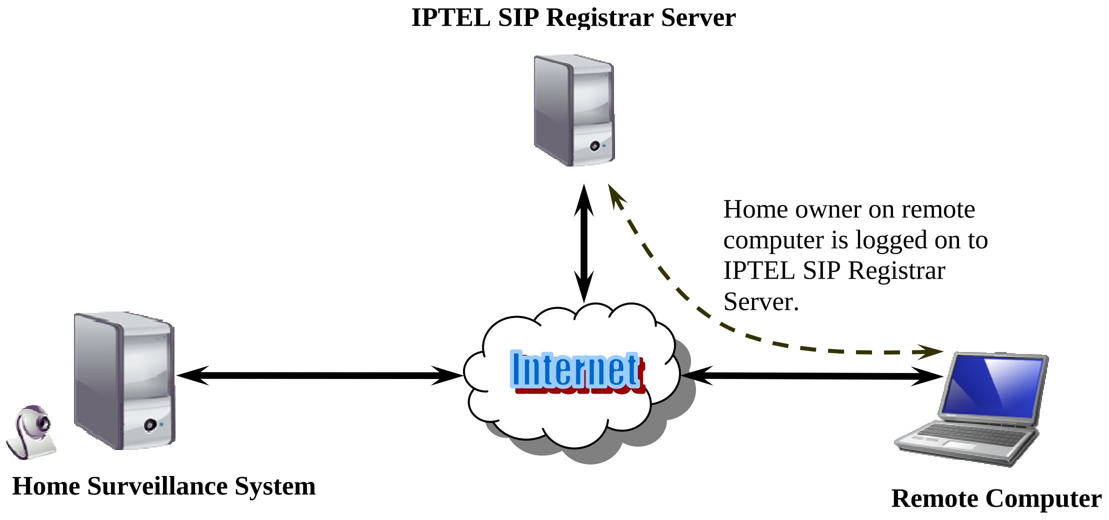
IPTEL SIP Registrar Server
Home owner on remote computer is logged on to IPTEL SIP Registrar Server.
Internet
Home Surveillance System
Remote Computer

Figure 1. Home Surveillance System

Home Surveillance System Hardware

The Home Surveillance System in Figure 1 consists of a motion detector, web camera, PC, and circuit board. The motion detector is an Eltec 442-3 Pyroelectric sensor that detects the motion of a heat source. The motion detector has 3 wires that can be connected to an A/D converter. When the detector senses a heat source in motion it sends a different voltage on its output pin. When a heat source moves from right to left, the detector output voltage increases. The opposite occurs when a heat source moves left to right, the output voltage decreases. The web camera is a Creative Labs Webcam Pro. However, any USB web camera that is compatible with the WDM Video for Windows Capture driver can be used. The web camera is a USB1.0 camera capable of supporting 640x480 pixels. However, this resolution is never seen in the application due to software restraints. The PC is any computer running Windows 2000 or higher operating system.

As shown in Figure 2, the circuit board is a Brainstem GP1.0 module that converts the analog signal received from the detector to a digital value. It can also perform many more functions and was mainly developed to interface sensors and servos to a computer’s serial port. The input analog signal received must be a value between 0 to 5 volts. The output digital value is a 10 bit number between the values of 0 to 1023. The GP1.0 module connects to the serial port on the PC. It allows applications to break out of the computer domain and expand in to the robotics domain. A home computer could act as the brains, receiving and controlling signals through the GP1.0 module.

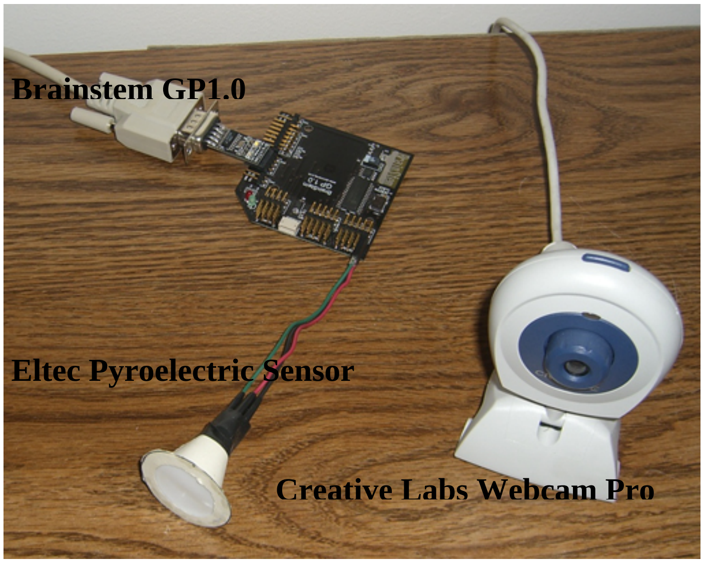

Figure 2. Home Surveillance System Hardware

The chart below shows the cost of the parts and where they were obtained.

Part Description	**Purchased From**	**Cost**
Eltec Pyroelectric Sensor	www.acroname.com	$70.00
Brainstem GP1.0 module	www.acroname.com	$81.00
Creative Labs Webcam Pro	Staples	$40.00
Microphone	Radio Shack	$10.00

Home Surveillance System Software

Two executables were created for this application, the surveillance executable and the remote client executable. Both applications require Windows 2000 or higher and were compiled each executable using the Microsoft Visual C++ 2005 Express Edition compiler. The surveillance executable runs on the home computer and the client executable runs on the remote computer. As shown in Figure 3, the two applications communicate with each other across an internet or intranet connection using the Microsoft Real Time Client (RTC) API [3]. The Microsoft RTC API provides functions built on top of the SIP and H.323 protocols.

The surveillance application runs two threads simultaneously. One thread monitors the motion detector sensor. This is a polling thread and continuously loops to check the status of the motion detector. It communicates with the GP 1.0 circuit board through libraries provided by Acroname. The other thread is event driven. It traps and processes Real Time Client events and user interface events. User events occur only when a user interacts with the user interface on the screen. These include pushing buttons, setting options, making calls, etc.

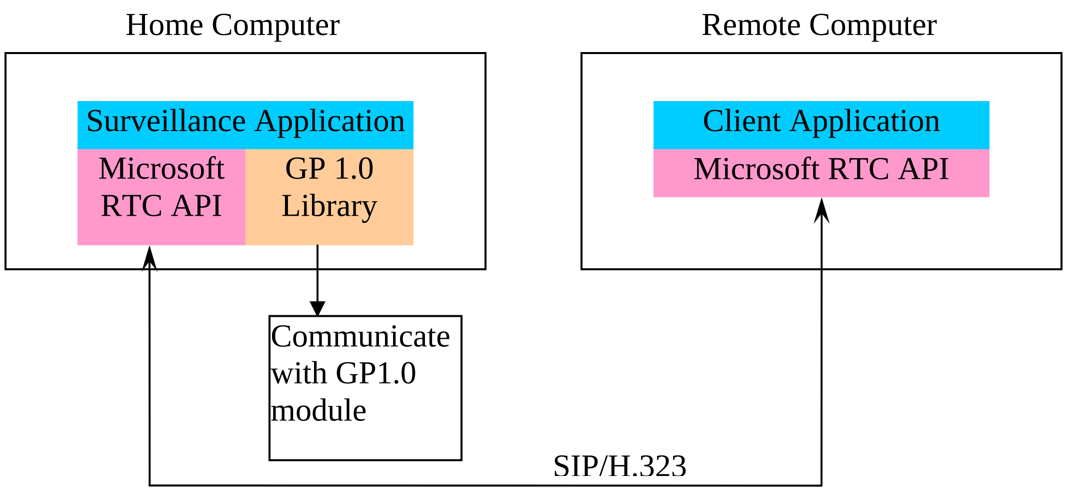

Figure 3. Home to Remote Communication

RTC events occur when a SIP message is received from the internet. These include SIP messages from someone trying to call, someone ending a call, a change in the presence status of a buddy, etc. The client application running on the remote computer is identical to the surveillance application except it does not contain a thread to poll the web camera.

The GP1.0 library contains functions for the PC to communicate with the GP1.0 module. The library is statically linked to the surveillance executable that is run on the home computer. It contains functions for controlling servos, reading analog signals, reading digital signals, etc. Different versions of the library exist for different operating systems. The functions are called directly form C code. The surveillance executable continuously calls on these functions to poll the A/D converter connected to the motion detector.

Session type	Media types	Participants
PC-PC	Audio send and receive Video send and receive Whiteboard (T120) Application sharing (T120)	2
PC-phone	Audio send and receive	2
Phone-phone	Audio send and receive	2 or more
IM	Text only	2
Multiparty IM	Text only	2 or more
Application	Media streaming handled by the application.	2

The Microsoft Real Time Client API is a dynamic link library found in the Microsoft Platform SDK [3]. The chart below shows the type of communications offered by the RTC API.

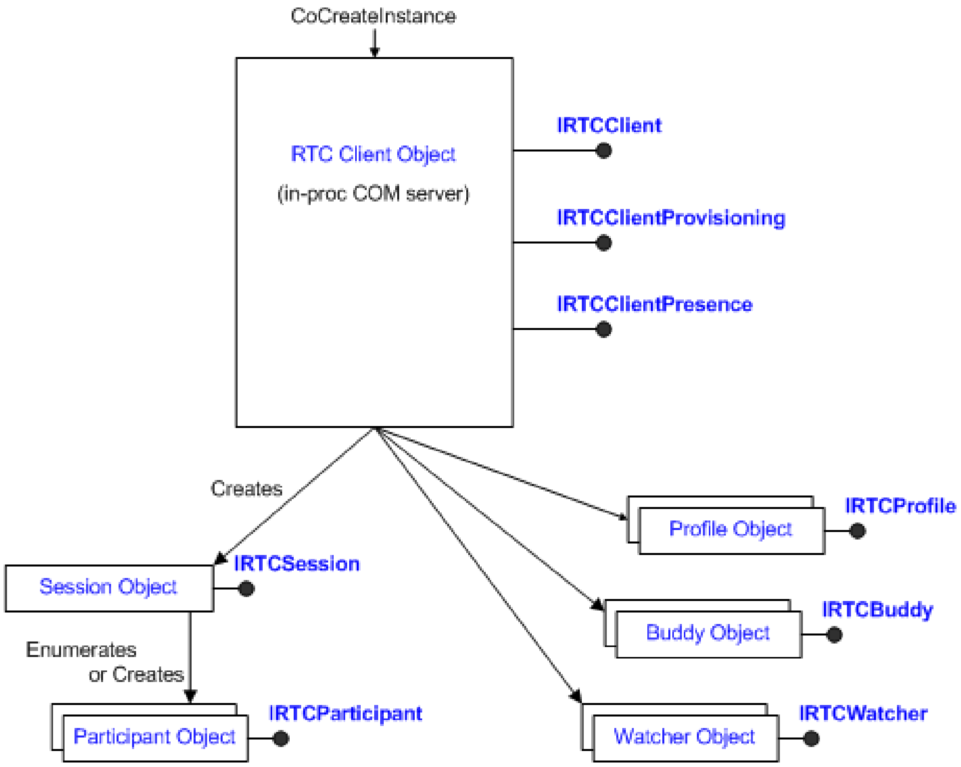

Figure 4. Microsoft Real Time Client API Architecture

The RTC API follows Microsoft's Component Object Model architecture where objects and their interfaces contain the necessary functions for creating sessions. The main function of the RTC API is to provide an interface to the features provided by the H.323 and SIP internet protocol standards. Figure 4 and Table 1 below shows the objects and interfaces of the RTC API. Objects are shown in boxes and their interfaces are dots connected to the boxes. CoCreatInstance is just a function that is called to retrieve a pointer to the RTC Client Object.

It should be noted that Microsoft's Real Time Client API is not only found in its Win32 COM technologies but also in its new .NET framework [4].

Table 1. Microsoft Real Time Client Interface Descriptions

Object Name	Interface Name	Interface Description
RTC Client Object	IRTCClient Interface	-Creates a Session object - PC to PC, PC to Phone, Phone to Phone, Instant Message, or Application -Can specify user profile containing user settings.. -Set preferred audio or video device -Set max bit rate for communication -Can start tuning wizard to adjust micophone, video settings -Create video window if video conferencing
	IRTCClient Provisioning Interface	- Manages user profiles. Provides functions for creating, disabling, or enabling user profiles.
	IRTCClientPresence Interface	- Can set current presence status – offline, online, away, idle, busy, be right back, on the phone, out to lunch. - Manage Buddies and Watchers. Can remove, add, and get Buddies and Watchers from profile.
Session Object	IRTCSession Interface	-Answers incoming sessions. -Terminates sessions. -Creates participant objects and adds them to a session. -Removes participants from a session. -Sends an instant message. -Adds media streams to a session. -Removes media streams from a session.
Participant Object	IRTCParticipant Interface	-Gets the user URI of a Participant. -Gets the name of a Participant. -Returns the current state of a Participant.
Profile Object	IRTCProfile Interface	- provides functions for accessing and setting options for a particular user profile
Buddy object	IRTCBuddy Interface	- includes methods to obtain presence status information on a **Buddy** object
Watcher Object	IRTCWatcher Interface	-includes methods to manage the state of a **Watcher** object

Internet Technologies for Video Conferencing

The Microsoft RTC API hides the complexity needed for making video teleconferencing applications. The API is built upon two internet standards, the Session Initiation Protocol (SIP) and H.323. The internet protocol hierarchy is depicted in the Figure 5 below.

SIP is an Internet Engineering Task Force (IETF) protocol standard that is defined in RFC 3261. It is a signaling protocol that manages sessions between nodes on an IP network. Sessions are connections between one or more nodes that allow video, sound, or data to be transferred between nodes. They are similar to phone calls on a PSTN network but allow for additional types of media to flow. Just as SS7 manages phone calls on the PSTN network, SIP manages sessions on an IP network. SIP sets up the session, monitors the session, and ends the session when the call is ended. Network nodes must have software installed to receive and send SIP messages in order to use the SIP protocol to establish sessions with

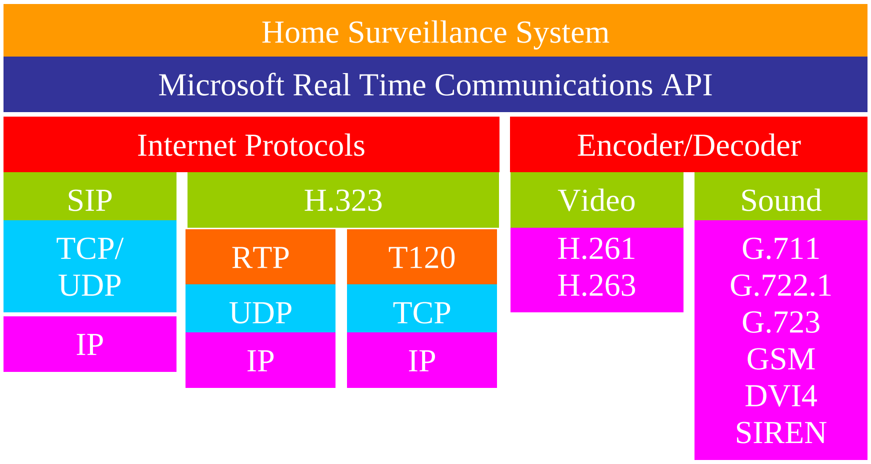

Figure 5. Internet Protocol Hierarchy

other nodes. SIP has the following characteristics [6],

- Lightweight, in that SIP has only six methods, reducing complexity
- Transport-independent, because SIP can be used with UDP, TCP, ATM & so on
- Text-based, allowing for humans to read SIP messages

Even though SIP is mostly used as a signaling protocol it can also be used for text messaging between nodes on an IP network. Once a session is established SIP uses the H.323 standard to actually transfer the type of media being used in the session.

SIP provides special features depending on the way that it is configured on the network [7]-[8]. On the most basic level, SIP can establish peer to peer sessions between two nodes on an IP network. The calling user would have to know the IP address of the computer that the person wishing to be contacted is at. Once the IP address is provided, the connection is made. If a SIP Registrar server is added to the network, then users on the network can use presence information on the server to contact each other. A user could simply provide the name of the person that they wish to contact. The information is passed on to the SIP Registrar server which finds the computer the person wished to be contacted is currently at. Presence information can be used for other features, such as monitoring the location and presence status of users online. If a PSTN gateway server is added to the network, calls can be made from the network to the PSTN. A user on a computer could establish a session with another person using a POTS telephone. SIP is a powerful communications protocol and is widely used today in many IP telephony services [9].

The other internet protocol that Microsoft's RTC API is built upon is H.323. "**H.323** is an umbrella recommendation from the ITU Telecommunication Standardization Sector (ITU-T), that defines the protocols to provide audio-visual communication sessions on any packet network"[6]. Even though H.323 tries to achieve the highest level of performance, it is not a guaranteed quality of service protocol. H.323 defines a collection of protocols to provide video conferencing services [10]-[11]. This collection of protocols is described in the table below. The main goal of H.323 is to provide a set of standards so that 3rd party vendors can create software/hardware for video teleconferencing applications that can communicate with one another.

Table 2. H.323 Protocols for Video Conferencing

H.323 Sub Protocol Name	Description
H.225	Defines call control and framing format. Allows more than one media stream to exist between two nodes in a session. Allows nodes to negotiate codec and performance settings to be used.
T.120	Provides data sharing between two nodes. This protocol supports file transfer and program sharing capabilities
RTP	Used to transport real time data across the network. Video and sound data are placed in RTP packets.
RTCP	Provides a way to monitor and control RTP performance.
RAS	Registration Admission Status. Provides gatekeeper control on the network.

The H.323 network is composed of several different types of nodes. H.323 terminals are the clients where the video conference takes place. Multipoint Control Units (MCU's) allow "three or more H.323 terminals to connect and participate in a multipoint conference"[12]. Conference gateways allow terminals on one LAN establish video conferences to terminals on other LAN's. Gateways also provide signal conversion if terminals on different sides of the gateway use different call signaling protocols. Gatekeeper devices manage video conferencing sessions across the network such as how many sessions can exist at once, or what calls car permissible. It is not necessary to have MCU's, gateways, and gatekeepers, but they provide additional functionality.

The H.323 umbrella recommendation also defines several encoder/decoder (codec) standards. These standards define how audio and video data is digitized and compressed before being packaged in to RTP packets. These codecs fall in to two categories, codecs for sound, G.7XX, and codecs for video, H.26X. The Microsoft RTC API uses the G.723 codec for audio and the H.263 codec for video by default. Both codecs are favored towards low bit rate transmissions and therefore sacrifice on quality of the signal being transmitted.

Practical Aspects of Home Surveillance System

The following steps were followed to implement the Home Surveillance System.

1. Purchase necessary hardware as described in hardware section.
2. Download Microsoft Microsoft Visual C++ 2005 Express Edition compiler from Microsoft for free.
3. Download Microsoft platform SDK from Microsoft for free.
4. Find example code in SDK to start from – RTCEXAMPLE.
5. Remove all compile time errors from example code.
6. Download Brainstem GP1.0 library from Acroname website.
7. Add software to monitor motion detector.
8. Find 3rd party SIP Registrar server on the internet offering free SIP services – www.iptel.org.

9. Create account on www.iptel.org. The surveillance application will access this server/account to find my remote computer location.
10. Test software.

Practical Limitations of Microsoft Real Time Client API

The RTC API was too high level and did not provide enough flexibility. Even though its functionality served the purpose for home surveillance system application, they cannot be extended to support other types of applications. Its functions were too distant from the H.323 and SIP standards it implemented while trying to extend this application that would allow a remote computer to control a pan/tilt mount attached to the home computer. It would be possible to implement this feature by first creating an executable to control the pan/tilt mount, allowing the executable to run on the home computer, and then sharing the executable with the remote computer. This would be possible using the application sharing protocol T.120 provided by H.323 protocol. However, the interface to the T.120 protocol is limited by the RTC API and cannot provide this type of capability. Also, the RTC API would not give control over which encoders/decoders to use. It was not possible to select the use of H.264 video codec feature. In addition, the RTC API does not provide good video performance. The best it can provide is QCIF at 176x144 pixel resolution.

In order to receive all the benefits of SIP such as presence information and being able to make calls to the PSTN network, one must still set up the necessary SIP Registrar and PSTN Gateway servers. The RTC API only provides the capability to communicate with those servers. Software must be created on those servers to provide the additional functionality. For this purpose, refer [5] at www.iptel.org to find a SIP Registrar server on the internet free of charge.

The RTC API also faces obstacles in trying to establish sessions on a more complicated network containing firewalls and proxy servers. A non-UPnP enabled NAT or firewall will block the media stream. These limitations of being too high level, having to create the necessary SIP servers, and amount of work needed to have the RTC API run across complex networks truly hinder the usefulness of the API.

Conclusion

A better software architecture for this type of application would have been to use software with closer ties to the H.323 and SIP protocols such as those provided by the ITU. Microsoft's high level Real Time Client API is good for building video conferencing applications but cannot be expanded upon for other uses. The RTC API provides limited video conferencing performance. To take full advantage of all the benefits of SIP, it is difficult to build own SIP Registrar server and PSTN gateway server. Standards developed by the ITU and IETF provide a software layering architecture that hides the complexity of communication at the application layer. Software implementations of ITU/IETF standards do not necessarily provide all the features of the standard being implemented. Care must be taken to understand the functionality provided by a software implementation.

References

[1] H. Kruegle, “CCTV Surveillance: Analog and Digital Video Practices and Technology “, Butterworth-Heinemann Publisher, 2006

[2] E.M. Hardwood, “Digital CCTV: A Security Professional's Guide”, Butterworth-Heinemann Publisher, 2007

[3] http://msdn2.microsoft.com/en-us/library/ms775893.aspx

[4] http://groups.google.com/group/microsoft.public.win32.programmer.rtc/topics?hl=en

[5] http://www.iptel.org/

[6] http://en.wikipedia.org/wiki/Session_Initiation_Protocol

[7] http://www.ietf.org/html.charters/sip-charter.html

[8] http://www.ietf.org/html.charters/avt-charter.html

[9] M.A. Rosengrant, “Introduction to Telecommunications”, Prentice Hall, 2006

[10] http://www.itu.int/rec/T-REC-H.323/en

[11] http://en.wikipedia.org/wiki/H.323

[12] http://www.microsoft.com/windows/NetMeeting/Corp/reskit/Chapter11/default.asp

Biography

Professor Rathika Rajaravivarma is currently a faculty member in the Computer Engineering Technology department at CUNY, New York City College of Technology, Brooklyn, NY. Dr. Rajaravivarma has over 17 years of experience as an engineer, software developer, and educator. Her areas of research interest are Signal Processing, Database Management, and Computer Networks for Multimedia Applications.