The Social Sciences 10 (6): 1574-1582, 2015
ISSN: 1818-5800

The Implementation of Peer-To-Peer Bandwidth Estimation Mechanism in Multimedia Streaming Networks

[1]Dong-Liang Cai, [1]Po-Hsuan Wang, [1]Yung-Chung Wang,
[1]Fu-HsiangTsai and [2]Jenn-Shing Wang
[1]Department of Electrical Engineering,
National Taipei University of Science and Technology, Taipei, Taiwan
[2]Deptartment of Computer Science and Information Engineering, JUST, New Taipei, Taiwan

Abstract: Due to the rapid development of the internet, P2P (Peer-to-Peer) data communication system has become one of the most convenient and fastest transmission methods. But the transmission quality will be affected by the traffic flow, so the available bandwidth estimation technique has to be designed and adjusted adequately according to the characteristics of the network service. Up to date, the multimedia streaming network bandwidth estimation technique has become the most popular topic. And this P2P related technology can be applied to local area communicating architecture, such as college campus or community etc. So that this study implements pathChirp bandwidth estimation mechanism to an embedded system with CPU of TI DM365 under Linux Operating System (OS). At the beginning, NAT (Network Address Translation) must be penetrated. Then the available bandwidth of the network environment will be calculated to determine the compression rate of codec H.264. After experimenting in three different network environments, we find the proposed mechanism that can really get a smooth video picture no matter what the traffic circumstance changes.

Key words: P2P (Peer-to-Peer), pathChirp, NAT, H.264, TCP, UDP

INTRODUCTION

Owing to the development of the instant messaging network to transmission real-time voice call in the internet, so an extension of the current video conference to video streaming live video multimedia services as have been developing. Then, over TCP (Transmission Control Protocol) protocol sending more of its video packets through the application software program UDP (User Datagram Protocol) and video transmission packets which will be our discussed keynotes. In short, the main purpose of this study is:

- In Linux OS, interface bandwidth with pathChirp point estimation mechanism to estimate the available bandwidth
- By DM365 digital media processor with digital signal processing DSP (Digital Signal Processing) technology; bandwidth adjustment under different H.264 compression ratios, to maximize the effectiveness of the limited bandwidth

Relative literature review: On the issues in heterogeneous network real-time available bandwidth, up to date in the internet, the bandwidth estimation is divided into passive (Passive) and active (Active) two categories (Wu, 2008). Because, the active estimation can quickly estimate the existing bandwidth, so most of the literatures have adopted active packet estimation method.

Generally in the market, it has many estimation tools such as IPerf (Xiao *et al*., 2007), ThruLay (Jain and Dovrolis, 2010), TOPP (Chimmanee and Wipusitwarakun, 2008), PathLoad (Melander *et al*., 2000) and pathChirp (pathChirp, 2012). We select TOPP, PathLoad and pathChirp three kinds of bandwidth estimation methods for comparison and choose the most efficient and immediacy pathChirp for our implementation.

P2P: The P2P emphasizes on the End-to-End, Program-to-Program, Computer-to-Computer (PC to PC), People-to-People the four kinds of direct relationship established network architecture and it allows client being a server.

Corresponding Author: Cai, Dong-Liang, Department of Electrical Engineering,
National Taipei University of Science and Technology, Taipei, Taiwan

Nat firewall: NAT has internal and external IP mutual conversion technology from the address limitation of physical IP in IPv4. An external IP address through the IP router can be divided into a group of addresses to provide for internal IP address 192.168.0.x assigned to all internal computers.

STUN: STUN set in RFC3489, the main probe is penetrating into NAT IP and Port Number of auxiliary methods and as a solution to commonly used in VOIP (Voice Over Internet Protocol) penetrating NAT firewall.

pathChirp bandwidth estimation: pathChirp (Alim *et al*., 2007) principle is based on the concept of self-induced congestion (Li, 2009) by the packet Chirps Method to dynamically measure the remaining network bandwidth shown in Fig. 1. PathChirp substitutes SLoPS (Self-Loading Periodic Stream) (Chen *et al*., 2007) equidistant packet interval with exponentially increasing packet interval.

PathLoad bandwidth estimation: PathLoad is a proactive End-to-End available bandwidth estimation tool, using SloPS (Zhang *et al*., 2012) estimation method, by PCT (Pairwise Comparison Test) and PDT (Pairwise d Diference Test) (Wen, 2011) methods detecting network individual delays. The packet transmission rate is R bit/sec and each probe packet will mark the time stamp when the receiver receives packet stream, then it can calculate each packet delay time OWD (One-Way Delays).

TOPP bandwidth estimation: TOPP's (Trains of Packet Pairs) principle is based on PRM (Probe Rate Model) (Yu *et al*., 2009; Zhang *et al*., 2010) and the sender sends a packet pair (Jain and Dovrolis, 2003; Chen, 2010) to the receiver, assuming the packet pair size L bits, the interval between packet pairs, the packet transmission rate Ri = L/s. While R is greater than the available bandwidth A, the second packet occurs in queuing situation and the receiver receiving rate Ro will be less than the transmission rate Ri. If R is less than the available bandwidth A, then Ri = Ro, same as the SloPS (Castellanos *et al*., 2006; Sargento and Valadas, 2006; Strauss *et al*., 2003).

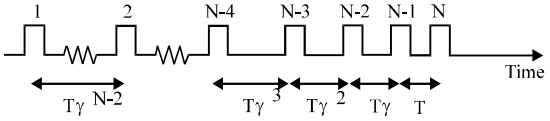

Fig. 1: Chirps packet transmission diagram

DM365 platform: DM365 has DaVinci platform with DSP processor-based combination development tools which can help developers to achieve ultra high definition digital video images in multimedia design. DM365 integrates numerous components such as H.264, MPEG-4, MPEG-2, MJPEG and VC1 encoder.

MATERIALS AND METHODS

Community-video intercom software design: Based on community-video intercom with pathChirp bandwidth estimation method as measures the sender and receiver immediacy available bandwidth. So, coupled with RTP header, the captured image can adjust H.264 compression ratio and transmission rate then put the encoded compressed live video packet to receiver, finally show live video image through LCD (Liquid Crystal Display) receiver shown in Fig 2. Community-video intercom operation steps shown as following:

- The user presses door sender call button and sends a request signal to door receiver
- When receiver receives a signal, server will send a signal to inform pathChirp to enter the bandwidth estimation process
- Then pathChirp server will send -s (Sender IP and Port Number), -r (Receiver IP and Port Number), -t (bandwidth estimation time) three parameters to receiver, then, it will start estimating the measurement bandwidth and inform sender the available measured bandwidth
- The sender camera captures images, based on the amount of available bandwidth and dynamically adjusts the compression ratio and the transmission rate of H.264
- Through RTP (Real Time Protocol) header sender gets the used encoder data sources, a series of compression position packets and chronological order, so that receiver can find the corresponding decodes and correct playing order

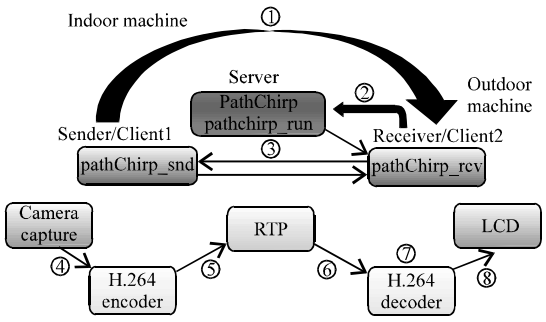

Fig. 2: Community-video intercom software system

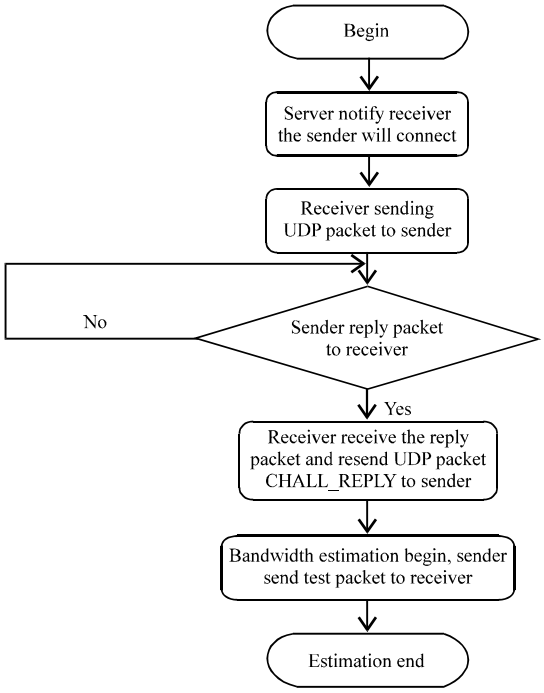

Fig. 3: pathChirp program flowchart

- When the receiver receives RTP header compression packets, the dismantling gets Payload Type, sequence number, timestamp three parameters
- After getting sender's encodes by Payload Type, then it begins decoding operation and getting packt sequence through sequence number

pathChirp bandwidth estimation software process: To achieve pathChirp Linux OS, the program unit is divided into pathChirp_rcv, pathChirp_snd and pathChirp_run. pathChirp_run installed on pathChirp server host, its aim is to inform the position both sender and monitor; pathChirp_rcv installed on receiver, establishing connecting between sender and receiver, it needs to informn receiver through server host then UDP packets send notifice to establish a connection; pathChirp_snd mounted to sender, it estimates estimation packet to measure real-time bandwidth between sender and receiver. pathChirp server installed on monitor, it performs the receiver and sender bandwidth estimation process, shown in Fig. 3

Whole processes in estimation, pathChirp server is only responsible for original receiver sending a packet to sender for connecting and the packet contains -s, -r and -t. As is completed by sender and receiver and the detailed schematic between these parties shown in Fig. 4. pathChirp detailed procedures as follows:

- Receiver pathChirp_rcv program opens TCP port 7365, UDP port number 8365
- Sender pathChirp_snd also will open UDP port number 8365
- Server monitor pathchirp_run program connects receiver and passes the -s, -r, -t three parameters to receiver
- After receiver receiving monitor information to initialize and attempting to connect sender, it sends setup socket to sender, asking for REQ_CONN, written packet, cc = 52, run select requesting to establish a connection
- Initializing receiver, sending a program creates chall number. After receiving connection message, it will send chall pkt to receiver
- Receiver receives chall pkt as coming from sender to obtain chal no and it will send CHALL_RE PLY to notify sender
- When sender receives challenge_reply, it begins sending Chirp probe packet to receiver and proceeding bandwidth estimation

The bandwidth estimation of pathChirp will set bandwidth estimation interval between 10 Mbps to 200 Mbps. So, repeating interval estimation procedures, it converges to a value and the value after convergence being available bandwidth.

Firewall penetration software procdure: The steps described in section (A) are prerequisites for both sender and receiver performing in the entity IP environment. So in Fig. 4 procedures, NAT firewall blocks receiver to send connecting packets to sender, resulting in bandwidth estimation stop.

Then, we are particularly joining NAT firewall penetration P2P in bandwidth estimation program. And, it needs a firewall penetration to send data between sender and receiver as defined Client1 and 2, through firewall P2P traversal way to allow sender and receiver getting connected shown in Fig. 5.

NAT firewall of Client1 and 2 open each active UDP protocol port number and send UDP packet to STUN server. If Client1 is going to connect with Client2 P2P then Client1 takes the initiative in sending probe packets to Client2. If Client2 receives the packet and replies to Client1, then it does not need to pass through firewall program and directly go into bandwidth estimation procedures. Then, Client2 will take the initiative in sending UDP packets to Client1 and Client1 will get IP and

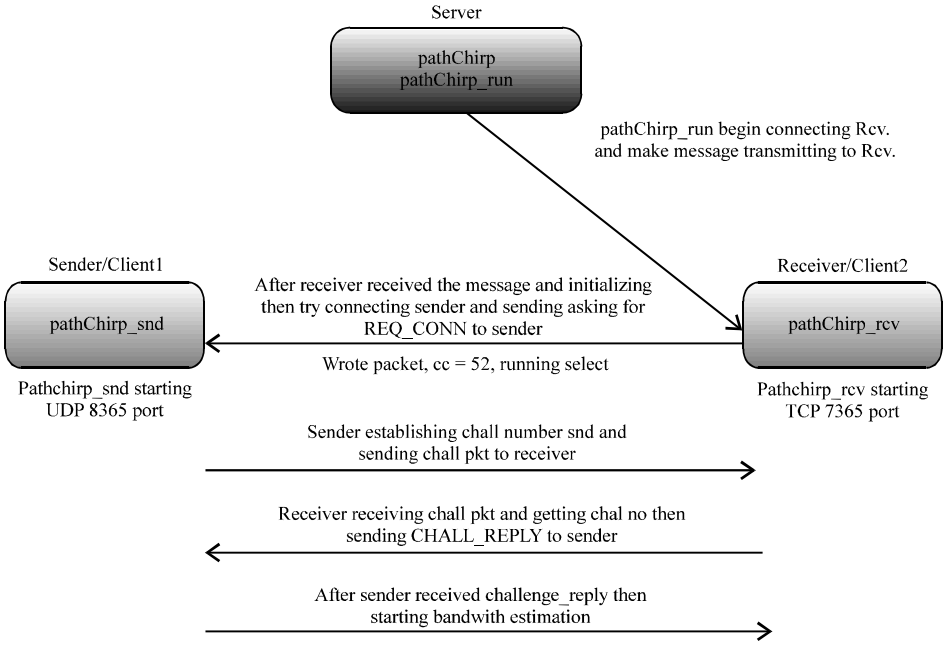

Fig. 4: pathChirp detailed proceeding scheme

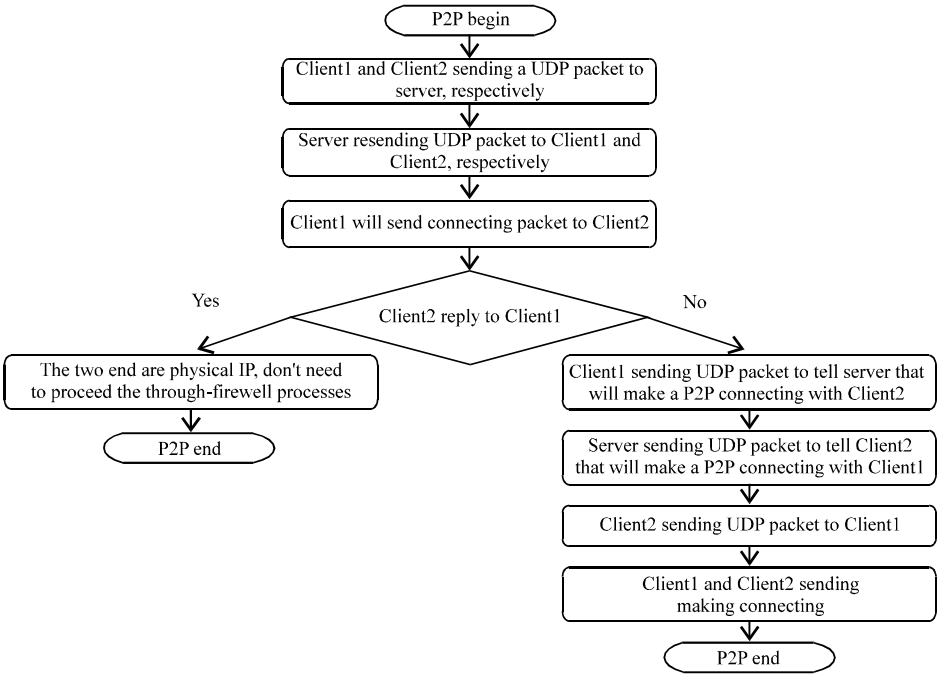

Fig. 5: Firewall penetration program flowchart

port number to pass through firewall and open up connection between these two Clients shown in Fig. 6

As the following step 1-4 which take the initative in logging on registration from the Client to Server. After the Step 5, it will take firewall penetration action:

- Client1 opens the UDP port number and then sends an UDP packet to server of packets containing Client1 physical IP and port number
- Server receiving Client1 packet which takes down the physical IP and port number of Client 1 and 2

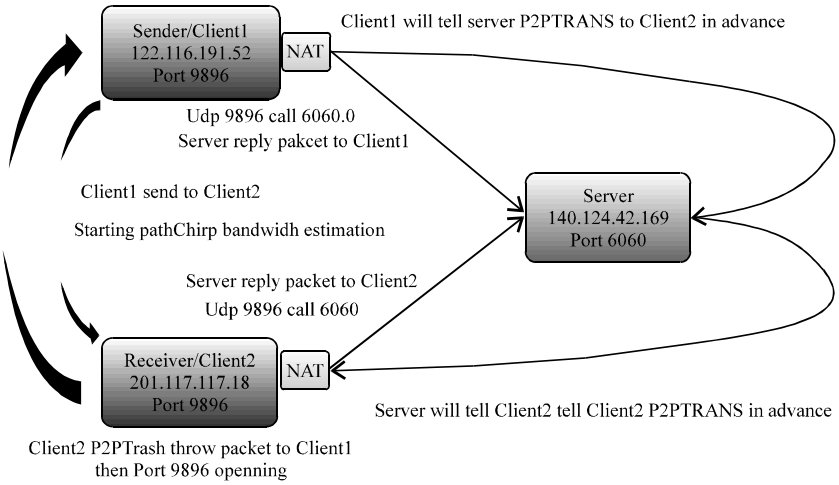

Fig. 6: P2P Firewall penetration shcematic diagram

- And after writing to the packet and sending back to Client1, Client1 take the position of IP and port number of entities located outside NAT firewall
- During connection, Client2 also opens UDP port number and sends UDP packet to Server, packets containing Client2 physical IP and port number
- On receiving Client2 packet, it records the physical IP and port number of Client1 and 2. After waiting the packet and sending it back to Client2, Client2 takes side with the physical IP and port number outside NAT firewall
- When Client1 intends to establish a connection with Client2, we must issue a request to server and establish P2P connection with Client2
- Server sends a notified packet to Client2 and advises Client1 intending to establish P2P connection, while the closure containing Client1 address and other information
- After Client2 receiving server connection packet, it will take the initiative in sending a UDP packet to Client1 and make Client1 recognize Client2 address and other information
- After Client1 receiving Client2 packet, it will send a packet to Client2 and make its connection successful or not
- If P2P connection is successful, then it starts pathChirp bandwidth estimation

Software implementation of image accessed unit: Image accessed unit: the implementation uses the camera lens module provided by OmniVision company and the camera model is OV9650 and 1.3-megapixel image quality. The image transfer interface adopts an 8-bit UYVY format connected to DM365. Of SXGA (Super eXtended Graphics Array) resolution, the highest resolution attains a level of 15 images per second. In DM356's pin from CCD-DATA0 to DATA3 and from CMOS_D8 to D11 which eight pins are used to make a simple image transfer data line. And we use I2C (Inter-IC) for controlling OV9650 to read and write shown in Fig. 7.

Once the control signals and data lines connected to CPU, the next part of driver must be implemented in order to communicate with Linux OS. Since, we implement Linux Video4Linux (Video4Linux V4L2, June 2012) as API (Application Program Interface) access interface layer, it must be appropriately adjusting the driver's part to connect the necessary image information.

The H.264 encoding/decoding implementations: The driver unit can be accessed by V4L2 (Video4Linux2) (Video4Linux V4L2, June 2012). interface and the image access unit's coding processes shown in Fig. 8.

DSP through V4L2 interface initializes encoder/ decoder engine (CERuntime_init) and DSP and ARM multimedia application interface DMAI (DaVinci Multimedia Application Interface) (Dmai_init). Then, it sets the input and output buffer space (Buffer_create) and evokes camera image access thread which image information through V4L2 will continue to remove the original image data for encoding/decoding operation (Venc1_process). So that the camera lens module images obtain UYVY format, it is converted to H.264 encoder for DSP which can accept NV12 format.

RTP implementations: All audio and video packets meet real-time which must provide a large and fast transfer rates through UDP protocol.

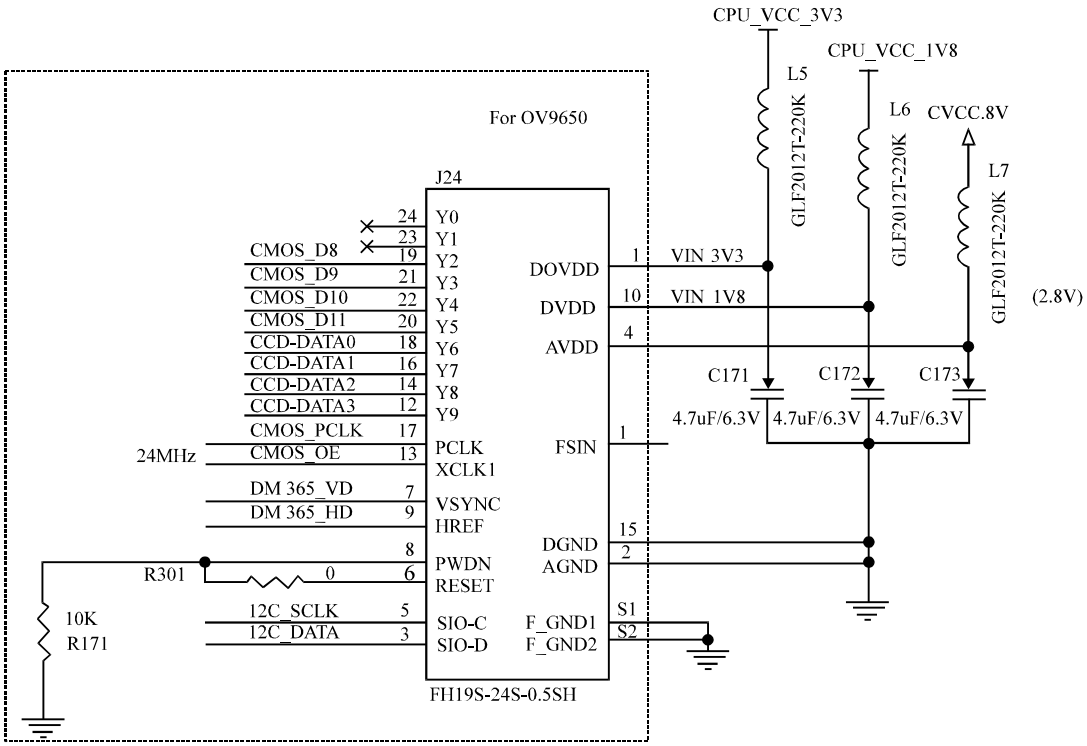

Fig. 7: Pins diagram

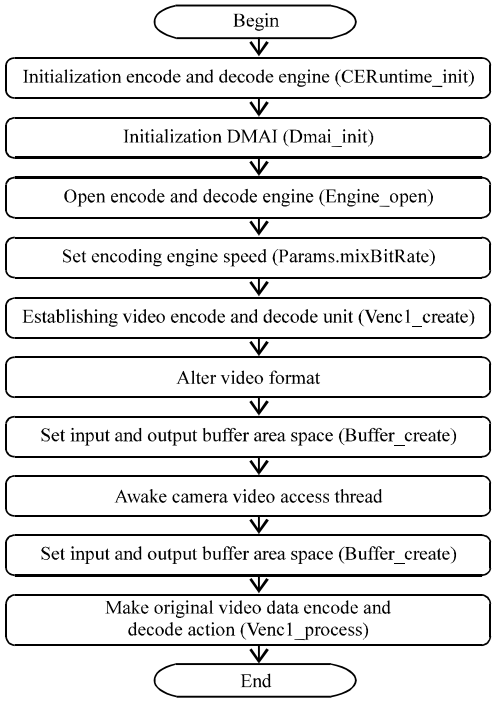

Fig. 8: DSP encode/decode flowchart

Although, the transmission speeds quickly, there is frequently packet loss, duplicated packets, failing order situation.

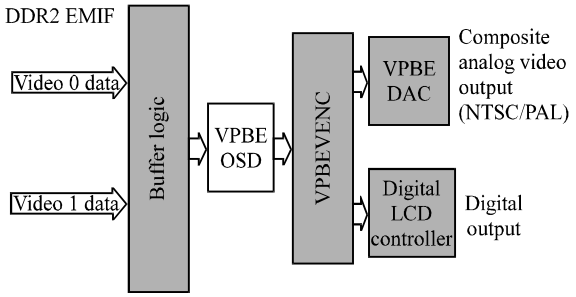

Fig. 9: DM365 showing interface simplified block diagram

LCD presentation unit: For an image shown, we use AUO (AU Optronics) 10.1-inch TFT-LC panel which has a resolution of 1024x600 and a communication interface of LVDS (Low-Voltage Differential Signaling) where DM365 interface has RGB format. Therefore, we need to add an interface to accomplish the conversion.

Figure 9 is a simplified block diagram showing DM365 interface, once the first step in video data as is available through the control of the buffer. And then, after VPBE (Video Processing Back End) processing, the image will show on the display panel.

RESULTS AND DISCUSSION

Hardware implementation

Introduction to implementing hardware platform: The implemented hardware environment has a LCD touch

Fig. 10: The door machine physical outside-view

Fig. 11: The door machine physical inside-view

Fig. 12: The indoor machine outside-view

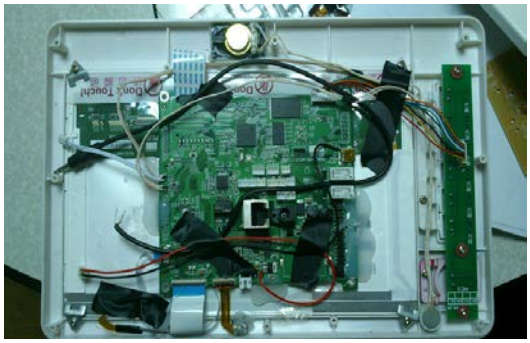

Fig. 13: The indoor machine physical inside-view

display and directional microphones door machine with photographic lens which communicate over the network.

Figure 10 and 11 is the physical picture of our experimental hardware platform door machine which the bandwidth estimation sender is in pathChirp. When door camera receives the available bandwidth value, as will immediately be adjusted by H.264 video packet transmission rate. Then, it puts video compression coding into the packet header of RTP packet and via UDP transmitting to door machine (receiver) to decode and proceed. Figure 11 shows the structure of motherboard and other hardware door machine, sending its video signal over the network port by UDP protocol interfaces.

Figure 12 and 13 is the experimental hardware platform photos for our indoor machine. Figure 13 is a motherboard and internal structure of indoor machine which is also transmitting video information over the network.

Implemented case; both devices on WAN and all have real IP addresses: The equipment, bandwidth estimation and P2P penetrating firewall servers are set up in physical IP address: 140.124.42.161 on host computer. The indoor machine is put on a physical IP address: 140.124.42.163 and the outdoor machine: 122.116.191.52 under LAN. And then, the outdoor machine under LAN, as is put a virtual IP address is: 192.168.0.195. Through DMZ (Demilitarized Zone) (DNZ, June 2012), we set outdoor machine NAT outside firewall port number to get a stable connection and physical IP. The practical case device with its physical IP address shown in Fig. 14.

Moreover, we will set up NAT port number corresponding to the outdoor machine as a physical IP address. This part does not affect the terms of the two devices communicating with each other, so the flow of operation are summarized as follows:

- After the devices being connected, the two devices take the initiative in registering the 140.124.42.161 server. Server returns the packet to the two host devices and then both devices will be informed the external physical IP and port number with each other by the packet
- After outdoor machine being pressed the button which starts judging of network type. Server will inform outdoor and indoor machine the physical IP and port number. After the judging, it will find that the two devices all have physical IP. Then, it can directly communicate with specified port number
- After confirming the communication with each other, bandwidth detection action will go on. And, indoor machine woken by main program to evoke pathchirp_rcv which is to wait receiving. Finally, outdoor machine executing pathchirp_run, as is to perform the assigned initial bandwidth detection, test time 30 sec. Then, outdoor machine starts sending estimated packets to indoor machine, shown in Fig. 15 In the figure, we can get the sender and receiver physical IP address and preset the bandwidth estimation range at 10-200 Mbps

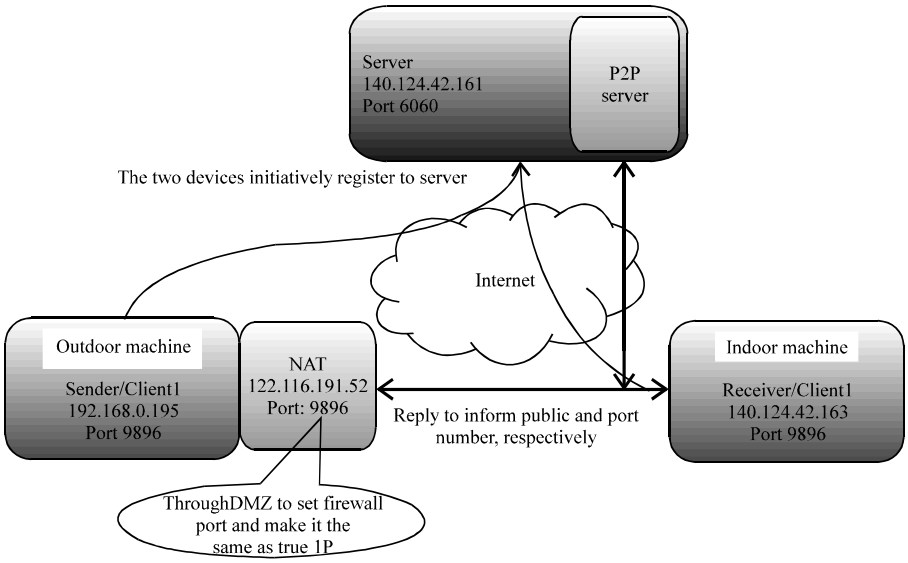

Fig. 14: The indoor and outdoor machine lie in WAN schematic diagram

```
Sender host is:  122.116.191.52
Receiver host is:  140.124.42.163
Opening file: 122_140_1342366213.instbw
```

Fig. 15: pathChirp bandwidth estimation execution

```
1342342497.171250  2.381000
1342342494.611250  2.260000
1342342501.051240  2.199000
1342342498.751240  2.287000
1342342496.251250  2.287000
1342342502.891250  2.287000
1342342500.301240  2.387000
1342342498.031250  2.340000
1342342504.481240  2.340000
1342342501.751250  2.191000
1342342499.481240  2.264000
1342342505.801240  2.143000
1342342503.451250  2.264000
1342342500.911250  2.365000
1342342507.421250  2.244000
1342342504.871250  2.422000
1342342502.301240  2.361000
1342342509.021250  2.409000
1342342506.501250  2.369000
1342342504.171240  2.490000
1342342510.631250  2.639000
1342342508.111250  2.566000
```

Fig. 16: The measured bandwidth under the public IP

- As shown in Fig. 16, the measurement results from each case under Public IP (external) and indoor machine will send the available bandwidth values, measured between outdoor and indoor machine. By Fig. 17, both ends measure the available bandwidth 2.566 Mbps in the lower right corner under physical IP
- While outdoor machine gets the available bandwidth data as will be adjusted in accordance with H.264 transfer rate. Because of the available bandwidth of 2.5 Mbps, we will adjust H.264 transfer rate to 1.5 Mbps. Shown in Fig. 17, outdoor machine has frame transmission rate about 10 KB. Therefore, the average transmission rate of packets per second is 10 KB*8*15 = 1.2 Mbps

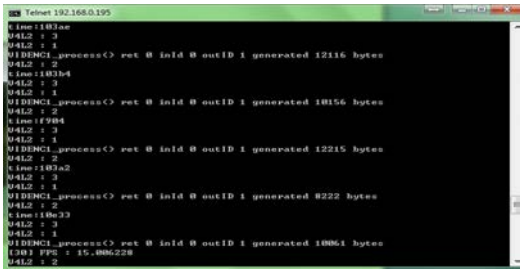

Fig. 17: Adjust the H.264t transfer rate to 1.5 Mbps

CONCLUSION

Through a network of real-time audio and video presentation which has become an integral part of life that now we need. Whether it is a network of P2P video download communication network or the real-time video surveillance cameras delivery, we are likely to encounter the heterogeneous bandwidth because of different networks connecting. In order to solve this problem, we try to get the available bandwidth between clients through pathChirp bandwidth estimation techniques, according to the available bandwidth as the transmission rate resizing H.264 video coding compression ratio. Of practicality, we will use pathChirp bandwidth estimation method, implemented in DM365 embedded CPU, then put it into the community-based video intercom system.

Although, we have measured three types of network environments, we still lack the practical measurement of bandwidth fluctuations in the internet. Therefore in the future, we hope to continue obtaining real-time information for estimating the available bandwidth in the

network, coupled with the ongoing dynamic adjustment of H.264 video compression ratio and to maintain immediate image fluently.

REFERENCES

Alim, M., B. Orlie and A. Arun, 2007. Bandwidth Estimation for Network Quality of Service Management. IEEE Conference on Military Communications, Orlando, pp: 1-7.

Chimmanee, S. and K. Wipusitwarakun, 2008. Regression based SSH-Telnet Metric for Evaluating End-to-end Path Capability Over the Internet for Supporting QoS. 8th International Conference on Telecommunications, Phuket, pp: 264-269.

Chen, Zheng-Hua, Yang, Zhe-Nan and Zhang Yi-Zheng, 2007. IEPM-BW on TWAREN realization. TANet2007 Taiwan Internet, Taipei, pp: 27-30.

Chen, B.Y., 2010. Information Technology and Applications Symposium academia. Master Thesis, Institute of Information Engineering Private Tatung University, Taipei.

Castellanos, C. U., D. L. Villa, O.M. Teyeb, J. Elling and J. Wigard, 2006. Comparison of Available Bandwidth Estimation Techniques in Packet-Switched Mobile Networks. 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2006, Sept.), Shanghai, pp: 1-5.

Jain, M. and C. Dovrolis, 2003. End-to-end Available Bandwidth: Measurement Methodology, Dynamics and Relation with TCP throughput. IEEE Transactions on Networking, pp: 537-549.

Jain, M. and C. Dovrolis, 2010. Ten fallacies and pitfalls on end-to-end available bandwidth Estimation. IEEE Symposium on Computers and Communications, Italy, pp: 272-277.

Li, M.F., 2009. Research Bidirectional Multimedia Streaming Network Available Bandwidth Estimation Mechanism. National Telecommunications Seminar, Kaohsiung, pp: 15-18.

Melander, B., M. Bjorkman and P. Gunningberg, 2000. A New End-to-end Probing and Analysis Method for Estimating Bandwidth Bottlenecks. IEEE Conference on Global Telecommunications, IEEE, San Francisco, pp: 415-421.

pathChirp, 2012. http://www.spin.rice.edu/Software/pathChirp.

Sargento, S. and R. Valadas, 2006. Accurate Estimation of Capacities and Cross-traffic of all Links in a Path using ICMP Timestamps. Telecommunication Syst., 33: 89-115.

Strauss, J., D. Katabi and F. Kaashoek, 2003. A Measurement Study of Available Bandwidth Estimation Tools, 11th IEEE Symposium on Computers and Communications, New York, pp: 39-44.

Wu, Li., 2008. Available bandwidth measurement method study. Master's thesis, Beijing University of Technology, Institute of Computer Applications, Beijing.

Wen, Y., 2011. Internet research available Bandwidth Measurement Method. Master's Thesis, Institute of Computer Architecture, Shandong University, Shandong, China.

Xiao, Y., S. Chen, X. Li and Y.Li. Li, 2007. A New Available Bandwidth Measurement Method Based on Self-Loading Periodic Streams. International Conference on Wireless couumuications Networking and Mobile Computing, Shanghai, pp: 453-458.

Yu, Jing, Yang, Ying-jie and Chang, Der-Yu, 2009. Available Bandwidth Measurement Algorithm Research. Computer Eng., 35 (6): 136-138.

Zhang, M., C. Luo and J. Li 2010. Estimation Available Bandwidth Using Multiple Overloading Streams. IEEE International on Communications, Nanjing, pp: 495-502.

Zhang, Da-Lu, Hu, Zhi-Guo, Zhu, An-Qi and Zhang, Jun-Sheng, 2012. A self-loading drop Out Rate Package Available Bandwidth Measurement Method. J. Software, 23 (2): 335-351.