



43rd International Conference
on Current Trends
in Theory and Practice of Computer Science
-
Student Research Forum, 2017
(SOFSEM SRF 2017)

Bandwidth performance analysis of Mobile VoIP Solutions

Rafael Dantas, Chris Exton and Andrew Le Gear

12 pages

Bandwidth performance analysis of Mobile VoIP Solutions

Rafael Dantas¹, Chris Exton² and Andrew Le Gear³

¹rafael.dantas@lero.ie

Lero - The Irish Software Research Centre
University of Limerick, Ireland

²chris.exton@ul.ie

Lero - The Irish Software Research Centre
University of Limerick, Ireland

³andrew.legear@horizon-globex.ie

Horizon Globex Ireland DAC
Nexus Innovation Centre, Ireland

Abstract: Despite the efforts to improve current 4G technologies by developing the new 5G mobile network, much of the world's population still relies on older 2G and 3G infrastructures.

Although old hardware can be replaced, the costs of such endeavour can be prohibitive for companies operating on developing nations. Network traffic optimisations can be used to provide a better experience for the increasing population of mobile devices connected.

This paper provides a base comparison of the bandwidth usage between Horizon Globex's Smart Packet VoIP Solution (SPVS) and a group of popular applications like Skype, Viber and WhatsApp. The experiment shows SPVS consumes less than 50% the amount of data when compared against the second place, WhatsApp. Nevertheless, more research is required to measure the impacts of the optimisations with respect to quality.

Keywords: VoIP, Mobile, Bandwidth, Consumption, Network

1 Introduction

As the 5G era approaches fast and the number of connected smart devices continues to increase, we find existing network infrastructures under increasing strain [Cis]. This change disproportionately affects developing nations, as old hardware will be required to support the increasing demand of growing populations [Cis].

While old hardware can be upgraded, and new equipment added, it can quickly become an expensive task, especially considering that most users will expect access to the Internet with 3G or faster connections.

With the majority of the world's mobile user population continuing to use 2G networks [Cis], network traffic optimisations can still provide benefits for mobile *Voice Over Internet Protocol (VoIP)* applications where modern mobile internet infrastructure is not available.

Horizon Globex's Smart Packet VoIP Solution (SPVS) [CD15] includes an Android and iOS mobile application, a series of node servers responsible for broker signalling between the phones' mobile applications, VoIP servers for hosting calls between mobile applications and gateways to standard telephony protocols, if necessary. It also employs a series of network and codec optimisations to allow real-time communication that is also stable, reliable and bandwidth efficient. This paper presents the results of an experiment comparing the bandwidth usage between the SPVS and nine popular mobile VoIP applications, which suggests that the former uses less bandwidth than the other applications tested.

2 Related Work

Performance and quality comparisons between applications are commonplace in mobile VoIP literature [DCR⁺95], [ZWL10], [SWB01], [KC11], [LL16], [MM01], [Räm10]. Often the objective of these tests is to evaluate the performance of existing technologies to select the best solution for the product under development. For example, [DCR⁺95] performed such an analysis for selecting the best codec for the Inmarsat mini-M system. Similarly, [ZWL10] also performed comparisons for the S3C2410 micro-controller.

The objective may also be to test the performance of a solution against the market standards and alternative solutions [SWB01], [KC11], [LL16], [MM01], [RT11]. When compared against other codecs like AMR and AMR-WB, the then newly developed Opus voice codec provided usable voice quality in [RT11]. To decrease the degradation of quality on VoIP systems caused by packet loss, [SWB01] proposed a new AMR codec and tested it against the usual codecs used with the H.323 protocol. A performance evaluation of Skype was made by [KC11] comparing it to industry standards. To improve bandwidth utilisation on mobile networks, [LL16] tested a TCP implementation with four alternatives and found significant improvements. Even classic algorithms can be subject to such tests. [MM01] tried to find which change would optimise Nagle's algorithm [Nag84] for network traffic control.

Lastly, quality tests may also be used to propose new ideas. [Räm10] used *Mean Opinion Scoring (MOS)* [SWH16] to test if subjects could distinguish between many narrowband and wideband codecs, and found a correlation between higher scores and higher bit rates. [WD15] evaluated the 3G network infrastructure in Bangkok, by testing how the MOS of Skype and LINE were affected by it.

3 The Smart Packet VoIP Solution

The SPVS is a complete mobile VoIP network solution including *Public Switched Telephone Network (PSTN)* breakout capabilities. While SPVS allows two users to communicate via traditional telephony infrastructure, it also supports one or both sides to be completely disconnected from regular PSTN if that user has access to the internet.

Figure 1 shows the basic workings of a typical VoIP call using SPVS:

- First the application contacts a central node, which will try to find the other user inside the network.

- The central node will then assign a VoIP server for handling the call, sending this information to both participants.
- Both ends will try to connect with the VoIP server. This is done to avoid any *Network Address Translation (NAT)* or proxy problems that would arise otherwise.
- Once connected, the VoIP server becomes responsible for relaying the call between the participants.
- If one or both the participants are not able to connect to the VoIP server, a *Session Initiation Protocol (SIP)* server will call the unreachable user using the normal telephony infrastructure.

3.1 Optimisations: Header Elimination

Most protocols carry some metadata alongside the payload. This header information is used by the receiver to understand and access the information contained inside, especially in cases where the configuration parameters can be changed for each individual packet.

To improve bandwidth efficiency, the SPVS' protocol doesn't carry metadata on every single packet. Instead, the developers of the solution use a series of metadata packets to coordinate the call between the devices, allowing the packet to carry only the actual encoded speech data.

In a similar way to protocols, codecs also carry a series of parameters that must be sent alongside the encoded data, which will be used by the decoding process. As an example, SPEEX parameters can be quite complex; as detailed by [TMA11].

If both sides of the communication agree on the configuration parameters of the payload the header section that carries this information becomes redundant. It is then possible to reduce header size, and ultimately remove it altogether, by standardising every parameter of the transmission.

The SPVS uses only one codec with a very specific set of parameters, making payload headers completely obsolete and allowing every packet to carry only the compressed speech signal.

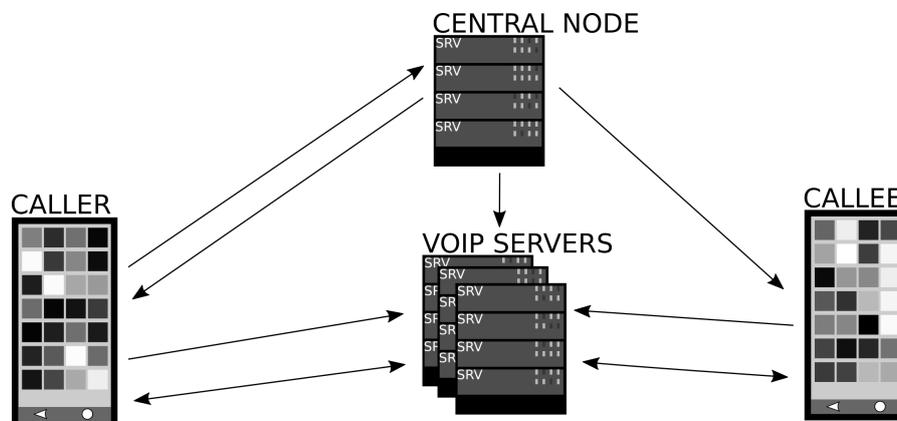


Figure 1: Typical workflow of a SPVS call.

Even if the payload does not carry a single bit of header information, every single network protocol will envelop all data produced by the layer above and append its own headers to it. Since those protocols are usually outside the scope of the application and only accessible by the OS layer or lower, the changes previously proposed in this section cannot be applied here.

To reduce the overhead imposed by the network protocol stack, we have used a sampling size of 100ms instead of 20ms. This change allows for a better usage of the network infrastructure, similar to the optimisations provided by Nagle's algorithm [Nag84] on TCP.

Since more data is being sent for each packet, the overall overhead size is drastically reduced as a larger part of the information is part of the actual payload.

3.2 Optimisation: Silence detection

In a normal conversation, usually only one of the participants is speaking while the other is listening to what is being said. If an application is not aware of this special case, it will always record and send speech information both ways. Using silence detection algorithms, it is possible to avoid sending useless data through the wire.

The SPVS has a special silence packet, which is sent by the application whenever it detects its user has gone silent. This packet also doubles as a keep-alive message, being sent periodically by the silent side to maintain the connection open.

3.3 Optimisation: Codec selection

As a freely available open source voice codec, SPEEX was selected to be integrated into Horizon Globex's solution. It also provides good results when compared with other narrowband voice codecs like AMR and iLBC [Räm10] [TMA11] [KC11].

SPEEX has many compression levels that decrease speech fidelity in order to decrease the overall size of the stream. This value ranges between 0 and 10, the former being the one with the highest compression but lowest quality while the latter is the exact opposite.

Horizon's developers decided to use compression level 3, since they have found that any higher setting does not provide improvement of the perceived speech quality through the phone speaker.

3.4 Optimisations: Transport protocol selection

There are two main protocols that operate over the *Internet Protocol, version 4 (IPv4)* [Pos81a]:

1. The *User Datagram Protocol (UDP)* [Pos80] is a very simple protocol, designed to provide a minimalistic mechanism for message exchange on an IP network. With simplicity as its main feature, a datagram has only 8 bytes of metadata. The error checking algorithm guarantees that every message received was not corrupted along the way but, since there is no error recovery built into the protocol, not all messages sent will arrive at their destination.
2. Another protocol is the *Transmission Control Protocol (TCP)* [Pos81b]. It was built for reliable communication between two computers inside the same network or between two

networks. To provide such reliability, the header is much bulkier than its simpler counterpart, containing at least 24 bytes of metadata. Although this feature is very important for all sorts of applications, its retransmission routines are usually more harmful to real-time applications like VoIP than the lost data they were designed to retrieve.

Due to its simplicity and smaller footprint, UDP was chosen as the best option as a transport layer protocol.

4 Evaluation

Section 4.1 will analyse the network performance of the SPVS, described in section 3, against the other solutions to test the assertions made in section 1.

For this experiment, we have selected some of the most commonly used applications in the western market, namely “Facebook Messenger”, Hangouts, Line, Skype, Telegram, Viber and WhatsApp. We have also selected two popular Chinese applications to represent that market, namely QQ and WeChat.

Table 1 presents the number of installed devices for all aforementioned applications. The numbers for all applications except WeChat and QQ were taken from Google’s Play Store, in 01/03/2017. Since WeChat and QQ’s mainly operate on the Chinese market, their numbers were taken from Huawei’s application store. Lastly, Hangouts’ numbers might not reflect its actual popularity since Android devices usually come with it installed by default.

4.1 Experimental Design

This experiment was designed to provide quantitative data for the analysis in section 4.3. We used two Android mobile phone where the tested application was installed with two GNU/Linux laptops acting as access points for both mobile phones and an earphone to reproduce a half minute-long audio file through both devices’ microphones. Figure 2 is an illustration of this experiment.

To properly intercept all data going through the Internet, both phones were connected to it via

Application	Installations
WeChat	2,344,124,542
QQ	2,004,169,512
Messenger	1,000,000,000
Hangouts	1,000,000,000
WhatsApp	1,000,000,000
Line	500,000,000
Skype	500,000,000
Viber	500,000,000
Telegram	100,000,000

Table 1: Number of installations per application.

the wireless access points provided by both laptops, which were connected to the Internet via two independent Ethernet cables.

Using a personalised *Command-Line Interface (CLI)* described in section 4.2 to schedule 40 test executions for each application described in section 1. This CLI was responsible for synchronising the actions of both computers during each test.

All tests performed followed these steps:

1. All tests are queued on the console, grouped by application.
2. Using a “next” command, the terminal loads the next test and waits for a user input. This input signals that both phones are connected in a call and ready for the audio playback.
3. Once the “enter” key is pressed, “tcpdump” is invoked as a background process, to record all network activity necessary for data collection.
4. Immediately after the previous step, the tool plays a half-minute-long audio file. This file contains phrases that alternate between the left and right channels to simulate a conversation with moments of activity and silence.
5. Both sides wait until the audio reproduction is complete. Then the terminal signals “tcpdump” to stop recording the network.
6. A counter is incremented to signals the completion of this test and the interface waits for a new “next” command or another user interaction.

All data collected by tcpdump was filtered using Wireshark, a network packet capture tool, which can read the format used by tcpdump and output the desired fields into the console, for further analysis.

Since all data going through the laptop was recorded into files, the most accessed IP address for that call was used to filter the data.

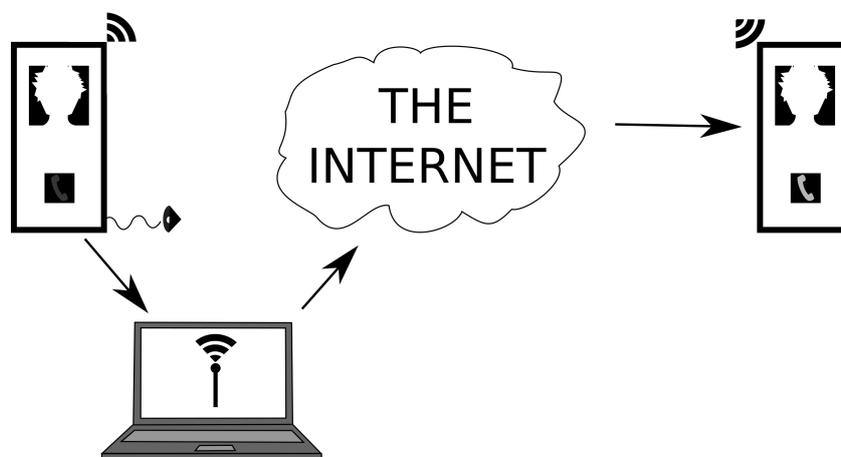


Figure 2: Representation of the experiment.

Application	Wire throughput (kb/s)	Payload throughput (kb/s)	Packet rate (p/s)	Frame size (bytes)	Payload size (bytes)	Payload-to-header ratio
Horizon	8.24	5.90	7.97	130	93	2.52
WhatsApp	23.59	17.44	22.23	133	99	2.83
QQ	28.64	20.07	31.08	116	81	2.34
Facebook	29.23	24.02	16.76	219	180	4.61
Viber	40.66	28.22	45.02	113	79	2.27
WeChat	44.85	35.83	33.13	170	136	3.98
Telegram	48.22	40.40	27.71	218	183	5.17
Line	53.82	42.21	42.33	159	125	3.64
Hangouts	59.06	32.35	50.80	146	80	1.21
Skype	84.54	61.69	83.98	126	92	2.70

Table 2: Comparison between various applications performances

The following section details the tools used for the experiment.

4.2 Tool Support

A personalised command-line interface was created to coordinate both laptops in all the tasks they were performing. This interface is able to connect the computers, interface with local network configuration by invoking “tcpdump” [JLM89] and “tc” [Kuz01], and it’s also capable of rolling back a test if needed.

“tc” [Kuz01] is a tool created to configure traffic control inside the Linux kernel. It is used for defining rules for incoming and outgoing packets on a device. This tool has many extensions, so called *queuing disciplines* or “qdiscs”, which are queues used to save packets while the network interface isn’t ready to handle them.

All data was collected using “tcpdump” [JLM89], a tool for inspecting all traffic on a network and storing all data onto a file. The output format is widely supported by many tools.

Two bash scripts were written for the filtering of data after its collection. These filters use “tshark” [Com12] to filter the pcap files created by “tcpdump”.¹

The audio sample was reproduced using “mplayer” [Tea05], a command-line interface player which was called by the script before changing the network properties.

For data analysis, “tshark” [Com12] and “Wireshark” [C⁺07] were selected for filtering the results, removing unrelated network traffic and for selecting only the fields necessary for the analysis.

In this experiment, every call happened after “tc” was used to create a new class. This class was configured to limit the bit rate to 100kbps, ensuring the application would use a narrow-band codec and giving the previously mentioned terminal some band to run commands without interfering with the experiment.

4.3 Quantitative Analysis: Bandwidth Usage

Table 2 has a comparison between all applications analysed as part of our evaluation.

- “Wire throughput” and “Payload throughput” are averages of the data rate sent to the network. The former’s numbers include the network overhead produced by all protocols in

¹ These scripts, along with set up advice, are available from the authors on request.

the network stack while the latter's numbers only include the data carried by the transport layer. Smaller numbers represent smaller data consumption and cheaper calls.

- “Packet rate” is the average number of packets sent per second. Applications with smaller values for this column might be less demanding on the network infrastructure while higher values may imply higher tolerance to individual packet losses.
- “Frame size” and “Payload size” are the average sizes of the individual packets sent by each solution. The former includes the size of the network overhead while the latter does not. Although higher numbers may imply higher network usage, this number only has true meaning when used together with packet rate.
- “Payload-to-header ratios” are simple ratios between the payload sizes and the average network overhead size, which is the frame size minus the payload size. Smaller ratios imply higher amount of data wasted on network headers.

The solutions were presented in ascending order by wire throughput, this is the values that most closely relates to the cost of the calls made by each solution.

When compared against the second closest solution in the first three categories, the SPVS uses, in average, 65%, 66% and 52% respectively less resources. The results are striking - using less than half the bandwidth than the second closest solution, WhatsApp. The SPVS has a network performance lead in this experiment.

Viber had the best results on both Frame size and Payload size. These numbers alongside a higher packet rate than the average of the other applications suggest they may be using a buffer size smaller than the other solutions.

Lastly, in the Payload-to-header ratio column, Telegram has the best payload-to-header ratio. They also have one of the biggest payload size, which naturally dwarfs the size of the header and inflates this ratio.

5 Discussion

The SPVS shows the best results with respect to the amount of data sent through the network, using at worst half as much data as the second closest application but on average 80% less throughput than other applications.

Table 3 shows the average cost per second of a call using each solution tested here. To calculate the correct cost, we have assumed an approximate 0.10 USD/MB, the standard rate for inter-telecom company GSM data, since a precise figure would vary between contract agreements between the solution's developer and its internet service provider.

Although both QQ and WeChat performed well on their respective runs of this experiment, the sound received by the callee was of very poor quality. For all other applications, the voice message was adequate and understandable in its entirety.

Skype had the best payload-to-header ratio, but that was because the payload carried by each packet was also larger in comparison to the size of the header.

Since the only poor network condition tested was packet loss, most solutions probably could not trigger changes in protocol.

Application	Wire throughput (kb/s)	Cost (usd cents/s)
Horizon	8.24	0.08
WhatsApp	23.59	0.23
QQ	28.64	0.28
Facebook	29.23	0.29
Viber	40.66	0.40
WeChat	44.85	0.44
Telegram	48.22	0.47
Line	53.82	0.53
Hangouts	59.06	0.58
Skype	84.54	0.83

Table 3: Average approximate cost of a call (assuming 10cent/s cost)

It is also surprising that most of the tested solutions were using TCP as their transport protocol instead of UDP, which would be the more sensible choice for real time applications like mobile VoIP, given the reasons discussed in section 3.4.

We believe the authors of the solutions using TCP over UDP believe that modern day internet connections are stable enough that the extra cost of developing for the latter protocol might not be justifiable. TCP already has concepts that are useful to voice calls like session and packet ordering that would need to be implemented by the application when using UDP.

6 Threats to Validity

Network fluctuations could affect the results of this experiment, making one solution appear better or worse than it would have been otherwise. Traffic shaping rules could also affect the numbers in a similar fashion. To minimise the effects of this threat, the experiments were run 40 times for the same application.

A better separation between inbound and outbound traffic could also be more useful for analysis than the total amount of data exchanged between both sides of the call.

Finally, network performance isn't the only important measure for a mobile VoIP solution - the perceived quality of the audio must also be maintained. The experimenter listened to the transmitted audio for each data point and confirmed that quality audio was being maintained, but we feel that although these initial tests provide some bases for comparison, we intend to further investigate these results using a MOS test in the next phase of our research.

7 Future Work

The next step in this research is to develop a more comprehensive experiment for both assertions in section 1, which will record both how much data is being sent through the network and how much of it is getting to the other side of the call. This will require data from both cell phones to be recorded.

Bandwidth savings are irrelevant if the overall quality of the audio sample has been degraded. Therefore, another important step is to properly measure how much has this property been affected by the optimisations and how SPVS's audio quality compares against the other solutions using a MOS assessment.

Finally, we intend to analyse ways to improve the current protocol. Ideas for future refinements include:

- A dictionary that will store common patterns produced by the codec to further compress the information being transmitted,
- A permanent personal audio profile to improve codec compression when talking with the same person more than once.

8 Conclusion

Although the SPVS showed very promising results in this experiment, a more detailed experiment will provide us with better data regarding current mobile VoIP solutions, especially with respect to quality.

9 Acknowledgements

This work was supported with the financial support of the Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

This work was possible thanks to Brian Collins, CEO of the Horizon Globex Ireland DAC, for permitting technical analysis of the product. (SOW2016-034)

Bibliography

- [C⁺07] G. Combs et al. Wireshark. *Web page: <http://www.wireshark.org/last modified>*, pp. 12–02, 2007.
- [CD15] B. Collins, C. Dziedzic. Method and devices for routing in a satellite-based communication system. Sept. 15 2015. US Patent 9,137,729.
- [Cis] C. V. N. I. Cisco. Global Mobile Data Traffic Forecast Update, 2015–2020 White Paper, 2016.
- [Com12] G. Combs. Tshark-dump and analyze network traffic. *Wireshark*, 2012.
- [DCR⁺95] S. Dimolitsas, F. Corcoran, C. Ravishankar, A. Wong, S. de Campos Neto, R. Skaland. Evaluation of voice codec performance for the Inmarsat Mini-M system. In *Digital Satellite Communications, 1995., Tenth International Conference on*. Pp. 101–105. 1995.

- [JLM89] V. Jacobson, C. Leres, S. McCanne. The tcpdump manual page. *Lawrence Berkeley Laboratory, Berkeley, CA* 143, 1989.
- [KC11] K. Kim, Y.-J. Choi. Performance comparison of various VoIP codecs in wireless environments. In *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*. P. 89. 2011.
- [Kuz01] Kuznetsov, A. tc. 12 2001.
- [LL16] K. Liu, J. Y. Lee. On Improving TCP Performance over Mobile Data Networks. *IEEE Transactions on Mobile Computing* 15(10):2522–2536, 2016.
- [MM01] J. C. Mogul, G. Minshall. Rethinking the TCP Nagle algorithm. *ACM SIGCOMM Computer Communication Review* 31(1):6–20, 2001.
- [Nag84] J. Nagle. Congestion control in IP/TCP internetworks. Technical report, Internet Engineering Task Force (IETF), 1984.
- [Pos80] J. Postel. User datagram protocol. Technical report, Internet Engineering Task Force (IETF), 1980.
- [Pos81a] J. Postel. Internet protocol. Technical report, Internet Engineering Task Force (IETF), 1981.
- [Pos81b] J. Postel. Transmission control protocol. Technical report, Internet Engineering Task Force (IETF), 1981.
- [Räm10] A. Rämö. Voice quality evaluation of various codecs. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. Pp. 4662–4665. 2010.
- [RT11] A. Rämö, H. Toukoma. Voice Quality Characterization of IETF Opus Codec. In *INTERSPEECH*. Pp. 2541–2544. 2011.
- [SWB01] J. W. Seo, S. J. Woo, K. S. Bae. Study on the application of an AMR speech codec to VoIP. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*. Volume 3, pp. 1373–1376. 2001.
- [SWH16] R. C. Streijl, S. Winkler, D. S. Hands. Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives. *Multimedia Systems* 22(2):213–227, 2016.
- [Tea05] M. Team. MPlayer–The Movie Player. <http://www.mplayer.hq.hu>, 2005.
- [TMA11] E. Touloupis, A. Meliones, S. Apostolacos. Implementation and evaluation of a voice codec for zigbee. In *Computers and Communications (ISCC), 2011 IEEE Symposium on*. Pp. 341–347. 2011.

- [WD15] P. Wuttidittachotti, T. Daengsi. Quality evaluation of mobile networks using VoIP applications: a case study with Skype and LINE based-on stationary tests in Bangkok. *International Journal of Computer Network and Information Security (IJCNIS)* 7(12):28, 2015.
- [ZWL10] J. Zhou, T. Wu, J. Leng. Research on voice codec algorithms of SIP phone based on embedded system. In *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*. Pp. 183–187. 2010.