Experimental estimation of channel width of IP BPX subscribers in OpenVPN IP networks

A Y Telkov and N V Olimpiev
Department of Electronics, Voronezh State University, Universitetskaya Sq., 1, Voronezh, Russia
E-mail: telkov@dpo-it.ru

Abstract. This paper presents the methodology and results of an experimental estimation of the channel width required to transfer traffic from VoIP subscribers connected via VPN technology to IP ATC Asterisk. OpenVPN technology with typical settings (tun interfaces, routed network segments and TCP transport protocol) was used. An updated tool for estimating the volumes of transmitted and received traffic was used, based on data processing from the ipcad sensor — an emulator of the Cisco NetFlow mechanism for linux routers. The operator representation of extracting information about single network sessions is used.

1. Introduction
It is known that the use of corporate IP telephony in order to provide remote subscribers with communication gives positive results. There are two approaches. The first is the use of cloud-based PBXs, and in this case, it is enough to connect the correctly configured IP-phones of subscribers to a local network with Internet access; the second is the creation and support of its own IP-PBX with the ability to connect remote units via the Internet, including via VPN channels, and communication with external subscribers via SIP trunks. In the second case, not cloud PBXs are rented, but group multichannel external numbers. Each of the mentioned options for organizing IP PBXs has lists of advantages and disadvantages, but in short it can be stated that small organizations often choose cloud IP PBXs, medium and large ones often deploy their own server infrastructure. Despite the presence of a priori data that the packing of SIP traffic into VPN channels leads to an increase in the bandwidth required for transmitting the voice traffic of each IP PBX subscriber [1–3], the authors did not find an experimental estimate of such an increase in open sources. At the same time, knowledge of this kind of information will allow VoIP engineers to use the updated data when calculating speeds of connected Internet channels both in units with hosted IP PBXs and in units with subscriber terminals (SIP clients). The aim of this work is to estimate experimentally the channel width occupied by the IP PBX subscriber’s traffic when corresponding SIP client operates on alaw, ulaw and G729 codecs in a local network, via the Internet, and through OpenVPN channels.

2. Materials and methods
To solve the task posed in the work, it was necessary to create a working model of the system in the form of IP-PBX and provide the ability to connect subscribers to it. In addition, we need to use an accurate tool for detailed traffic analysis and VoIP channel width measurement.
This model was created on the basis of a server with the CentOS linux operating system, with Asterisk software IP PBX and the FreePBX interface. Traffic data were collected using ipcad, a Cisco NetFlow emulator for linux routers [4]. The operator representation of the procedure for extracting data about traffic sessions was used [5]. On the same server, the OpenVPN server role was installed to provide the ability to connect remote SIP subscribers through OpenVPN channels. In the process of research, measurements were carried out in a simulated telephone conversations, data was recorded every 15 seconds. The amount of information for this time interval was measured in bytes. The experimental network is shown in figure 1. All measurements were carried out in three locations: a university laboratory, a home laboratory and the central office of a partner company, for which necessary equipment was installed and networks shown in the diagram were configured. Traffic measurements to calculate the requirements for the required channel width were carried out for 3 codecs: G711 (alaw, ulaw) and G729. Measurements for each codec were made for three different cases — inside the OpenVPN channel, inside the local network and in the open network. Each measurement cycle for the selected codec lasted about five minutes in order to obtain a sufficient amount of reliable data. The measurements were carried out repeatedly for each codec.

Let’s consider in detail steps for preparing the model:

1. Firstly, a linux server with the iptables service was installed in the central office of the partner company. The IPv4 network address of eth0 on the LAN was set to 192.168.132.251/24. The gateway address on this network was 192.168.132.25. Inbound and outbound traffic on eth0 was allowed.

2. Secondly, on the above linux node, the OpenVPN server role was deployed. The address of the virtual adapter tun0 was set on 10.8.2.1/24. Thus, the 10.8.2.0/30 network was allocated for the virtual end address of the OpenVPN tunnel on the server, and the addresses 10.8.2.4/30, 10.8.2.8/30, 10.8.2.12/30, etc. have been configured to issue for remote OpenVPN clients. The VPN server provided connections to the OpenVPN hardware and software clients during the
experiments. The ability to connect to the server was provided by translating one of the private port TCP ports from an external routable IPv4 address to port 1194 TCP with the HQ Border Router (figure 1).

3. Thirdly, ipcad sensor was installed and capture of information about transited packets through the eth0 interface was configured, taking into account IP addresses and TCP and UDP traffic port numbers.

4. Fourthly, an Asterisk server with support for the ability to work with the codecs alaw, ulaw, g729 and the FreeBPX interface would be installed. The server could accept connections from all hardware and software SIP clients participating in the experiment.

5. Fifth, sets of keys and configuration files were generated that enabled the connection of OpenVPN software and hardware clients to the server. And for the experiment with measurements from the university laboratory, in addition, routing of the remote segment 172.16.11.0/24 was organized using the corresponding configuration file on the OpenVPN server with the iroute directive [6].

6. For convenience of working with ipcad data, an operator view was used. As was shown in [5] data from ipcad sensor may be written in PostgreSQL database, in table ipcaddump with structure, shown in table 1. Very small selective data example shown with restricted columns in table 2.

7. To record information coming from the ipcad sensor, a special updated traffic collector was developed. It was implemented using two scripts, one of which was run by the cron task scheduler once a minute (every minute). The first shell-script calculates the number of function calls for writing files with ipcad data. The first script also recorded these files. The second script was created with command-line php and called by the first script at the end of its execution. At the end of the first script, a cycle was launched that was executed for each of the recorded files, which erased these files and wrote information from files to the database. In order to ensure unique file names within one recording cycle, their names contained string values obtained by converting the system date and time include seconds. To illustrate the operation of the algorithm, it is advisable to bring its enlarged block diagram (see figure 2). With a value of $N = 4$, the measurement interval becomes equal to 15 seconds and we come to our case.

### Table 1. Fields and types in ipcaddump table.

| Field name | Field type    |
|------------|---------------|
| code       | bpchar(19)    |
| sourceip   | bpchar(15)    |
| sourceport | bpchar(15)    |
| destip     | bpchar(15)    |
| destport   | bpchar(15)    |
| packets    | int(4)        |
| bytes      | int(4)        |
| protocol   | bpchar(5)     |
| interface  | bpchar(5)     |
| info       | bpchar(16)    |
| ipcdate    | date          |
| ipctime    | interval(2147418114) |
Further, after the preparation of the model network, studies were carried out as follows.
1. Determination of the subscriber’s channel width within the local network. To solve this problem, we used an IP telephone connected to a segment of the local network of HQ Office and an external subscriber to the cellular network. Multiple calls were made from an IP phone to an external number through one of the IP telephony providers. At a five-minute interval during a telephone conversation, the bandwidth $H$ was calculated for each of the 15-second sub-intervals that make up this interval

$$H = \frac{V}{(8 \times T)},$$

**Figure 2.** The algorithm for writing information to the ipcaddump table.

**Table 2.** Fragment of ipcaddump table.

| sourceip       | sourcep. | destip       | destp. | packets | bytes | prot. | ipctime     |
|----------------|----------|--------------|--------|---------|-------|-------|-------------|
| 192.168.132.251| 5060     | 212.53.40.40 | 5060   | 4       | 2636  | 17    | 14:31:45    |
| 192.168.132.251| 1194     | 128.70.43.145| 5328   | 23      | 8254  | 6     | 14:31:45    |
| 192.168.132.251| 53782    | 192.168.132.25| 53    | 1       | 55    | 17    | 14:31:45    |
| 192.168.132.25 | —        | 224.0.0.5    | —      | 1       | 64    | 89    | 14:31:45    |
| 192.168.132.25 | 37797    | 255.255.255.255| 5678  | 1       | 148   | 17    | 14:32:00    |
| 192.168.132.251| 33150    | 192.168.132.25 | 53   | 1       | 55    | 17    | 14:32:00    |
| 212.53.40.40   | 5060     | 192.168.132.251| 5060  | 3       | 1535  | 17    | 14:32:00    |
where $V$ is the volume expressed in bytes; $T$ — time interval in seconds.

Information about the amount of data transmitted by the IP phone in the direction of the IP PBX was extracted from the database using queries similar to the example below

```
SELECT SUM(bytes) FROM ipcaddump WHERE ((ipcdate='2019/05/15') and (ipctime >'10:15:00') and (ipctime <'10:15:15') and (sourceip='192.168.132.239') and (destip='192.168.132.251') and (protocol='17')).
```

This example corresponds to a specific date and time interval for measurements and assumes that the IP phone received the address 192.168.132.239 on the local network.

2. Determination of the subscriber’s channel width within an open network. To solve this problem, an IP phone and laptop were connected to the LAN segment of the VSU LAB. Using a laptop and a service for determining external IPv4 addresses, an external IPv4 address was determined, with which packets from this local segment go to the Internet. Then, from this phone, a call was made to an IP phone in the HQ Office segment and, similar to case 1, the bandwidth was calculated. The request for information retrieval in this case had the form like

```
SELECT SUM(bytes) FROM ipcaddump WHERE ((ipcdate='2019/05/15') and (ipctime >'02:15:00') and (ipctime <'02:15:15') and (sourceip='64.76.168.254') and (protocol='17')).
```

For the period of measurements for this option, the translation of ports 5060, 10000 - 20000 UDP from the external IP address to the IP address of the PBX (192.168.132.251) was temporarily configured with the restriction of the addresses from which the SIP client can connect to the IP PBX. The study for this case was conducted outside of working hours, IP phone in the HQ Office segment was configured with “auto-pick-up” option in order to be able to conduct research in the absence of a person.

3. Determining the subscriber’s channel width within the VPN network. To solve this problem, a router — an OpenVPN LinkSys WRT54GL client with dd-wrt firmware — was connected to the VSU LAB LAN segment. Behind the router was an IP phone. Further, a call was made from this phone in the same way as options 1, 2 and the bandwidth was calculated. The request to extract information in this case had the form like

```
SELECT SUM(bytes) FROM ipcaddump WHERE ((ipcdate='2019/05/15') and (ipctime >'15:15:00') and (ipctime <'15:15:15') and (sourceip='64.76.168.254') and (destport='1194') and (protocol='6')).
```

In addition, in a similar way, the measurement of the subscriber’s channel width inside the VPN network and the measurement of the subscriber’s channel width inside the open network was carried out from the home laboratory, using the programmatic OpenVPN and SIP clients with laptop.

### 3. Results and discussion

The reasons for the differences in the channel width required by the SIP client when working in various conditions. Before analyzing the research results, we consider the main reasons for the differences between the theoretical and actual band occupied by VoIP traffic.

Firstly, differences are related to MTU size. MTU (Maximum Transmission Unit) — the maximum size of the useful data block of one packet, which can be transmitted by the protocol without fragmentation. One of the problems associated with MTUs is that higher layer protocols can create larger packets that are not supported by other network nodes. To address this Internet Protocol supports fragmentation, which allows to break a datagram into smaller pieces. However, despite this method of solving the problem, packet fragmentation has its drawbacks, for example, due to the copy of the IP header, the bandwidth is inefficiently used for each fragment, increasing the channel width requirements [7].

Secondly, the differences are due to the features of the packet headers. In order to calculate the throughput for VoIP traffic, use the following formulas [8–10]:

```
Total packet size = (L2 header: MP, FRF.12 or Ethernet) + (IP / UDP / RTP header) + (amount of useful voice data);

\[ PPS = \frac{\text{codec bit rate}}{\text{(voice payload size)}}; \]

\[ \text{Throughput} = (\text{total packet size}) \times PPS. \]

All VoIP packets consist of two components: voice samples and IP / UDP / RTP headers. Although voice data samples are compressed by a digital signal processor (DSP) and may vary in size based on the codec used, these headers are constant 40 bytes in length. For example, compared to 20 bytes voice samples in the default call G.729, these headers make up a significant proportion of the service data. Thus, without using any kind of compression, packet headers cause differences in the theoretical and practical bandwidth of VoIP traffic.

Thirdly, the encapsulation of traffic during a VPN connection has a direct effect on the increase in channel width. When converting VPN traffic, the data packets are “fouled” with additional information, which increases the channel bandwidth requirements.

The measured histograms and the average values of the SIP channel calculated from the measured data are presented below.

Measurement of traffic within the OpenVPN channel. The average value for the alaw codec is 118.7 kbit/s; the histogram is shown in figure 3. The average value for the ulaw codec is 121.1 kbit/s; the histogram is shown in figure 4. The average value for the G729 codec — 19.4 kbit/s, the histogram is shown in figure 5. The comparative histogram for these three codecs is presented in figure 6.

Measurements of traffic on the local network. The average value for the alaw codec is 66.6 kbit/s; the histogram is shown in figure 7. The average value for the ulaw codec is 65.1 kbit/s; the histogram is shown in figure 8. The average value for the G729 codec — 7.9 kbit/s codec; the histogram is shown in figure 9.

The histograms measured in an open network are similar. For this reason, their graphic images are not shown here. The resulting values occupy an intermediate position between values calculated for the VPN channel and values calculated for the local network for the corresponding codecs. Quantitative values of the corresponding characteristics are given later in the work.

To summarize, it is convenient to present previously obtained results in tabular form — see table 3.
Figure 4. ulaw in VPN channel.

Figure 5. g729 in VPN channel.

| Parameter                                               | alaw | ulaw | g729 |
|---------------------------------------------------------|------|------|------|
| Theoretical value, kbit/s                              | 64   | 64   | 8    |
| Measured data in OpenVPN channel, kbit/s               | 118.7| 121.1| 19.4 |
| Measured data in local network, kbit/s                 | 66.6 | 65.1 | 7.9  |
| Measured data in open network, kbit/s                  | 95.8 | 92.6 | 12.2 |

4. Conclusions
The results of measurements and calculations based on them show that the actual channel width occupied by the active SIP subscriber of the IP PBX is more than theoretical values.
Measurements of data on SIP traffic packed in an OpenVPN channel showed that the required channel width increases approximately by 2 times. Using Wireshark, observations were made illustrating that the size of the transmitted frames is significantly increased compared to the standard value. For example, for the G729 codec, the frame size is increased from 32 bytes to 74 bytes. The measurements were made for the case of a typical OpenVPN setup (with tun adapters, routed segments, and TCP transport protocol).

Measurements in an open network showed that the requirements for channel width increase by about 1.5 times compared with theoretical values for various codecs.

The measurement data for the local network are most consistent with the theoretical ones. This is due to the fact that in the local network from the point of view of packet transformation, only one operation occurs — encapsulating IP packets in Ethernet frames when sending for transmission through Ethernet switches and decapsulating these frames when receiving during
exchange between IP PBX and SIP clients, and measurements traffic volumes are produced using tools that operate with packet data (after this conversion) on the IP PBX itself. Differences from theoretical values in this case allow us to estimate the measurement errors, which is 3.5 percents.

The results of this research can be used by VoIP engineers in calculating speeds of connected Internet channels in network segments with hosted IP PBXs and in network segments with subscriber terminals (SIP clients).

References
[1] Amirisetti, Sushma and Sanguankotchakorn Implementation of IPsec VPN with SIP Softphones using GNS3 ICNCC 2018 Proceedings of the 2018 VII International Conference on Network, Communication and Computing pp 152–156 (DOI: 10.1145/3301326.3301333)
[2] Kuhn D, Walsh T and Fries S 2005 Security Considerations for Voice Over IP Systems Recommendations of the
National Institute of Standards and Technology. "National Institute of Standards and Technology (800–58), pp 1–93." "Coonjah I, Catherine P and Soyjaudah K Experimental performance comparison between TCP vs UDP tunnel using OpenVPN 2015 International Conference on Computing, Communication and Security (ICCCS) Pamplemousses, Mauritius 4–5 Dec 2015 (DOI: 10.1109/CCCS.2015.7374133)"

"Mukhtarov M, Miloslavskaya N and Tolstoy A Network security threats and cloud infrastructure services monitoring, The Seventh International Conference on Networking and Services, ICNS 2011 IARIA, May 2011, pp 141–145." "Telkov A, Krokhin E 2017 Proceedings of VSU. Series: Systems analysis and information technologies.51–56 (URL: http://www.vestnik.vsu.ru/pdf/analiz/2017/02/2017-02-07.pdf)"

"Keijser J OpenVPN Cookbook 2017 Packt Publishing (URL: https://edgeconsult.me/lib/Software Development/Networking/VPN/OpenVPN/openvpn-cookbook-2nd.pdf)"

"Resolve IPv4 Fragmentation, MTU, MSS, and PMTUD Issues with GRE and IPsec Cisco Systems (URL: https://www.cisco.com/c/en/us/support/docs/ip/genericrouting-encapsulation-gre/25885-pmtud-ipfrag.htm)"

"Davidson J 2007 Voice over IP Fundamentals – 2nd Edition, 432 p"

"Roslyakov A V 2003 IP telephony (Moscow: Eco-Trends), 252 p"

"Goldshtein B S 2014 Directory (St. Petersburg: BHV-Peterburg), 456 p"