

Top 10 VoIP IP telephony interview questions and answers with an extra C# softphone development tutorial

Introduction

Due to the rapid spread of VoIP technology, increasing number of companies are interested in VoIP programming in relation to the improvement of their telecommunication network. So in certain instances, basic knowledge of VoIP is required. Therefore, even during a simple software developer interview, it is conceivable that you will have to face with some VoIP-related questions as well.

For this reason, I have compiled a list of the potential VoIP-related questions and I have added a simple softphone example project in C# to get started easily if you would like to get a taste into this field of software development. My article has been written for software engineers primarily, but it can be useful for network engineers as well.

This interview guide includes the most common questions within the VoIP topic. If you think that my article should contain some further important questions, feel free to suggest your ideas in the 'Comments and Questions' section below.



Potential VoIP-related questions

1. What is VoIP?

VoIP stands for Voice Over Internet Protocol or Voice Over IP. VoIP technology makes it possible to convert analog voice signal into digital data and transmits it over the Internet. (There are more likely possible pronunciations, as well as vo-ipp, have been used, but generally, the single syllable - voyp, as in voice - may be the most common within the industry.)

2. Why VoIP is better than traditional phone services?

Due to its cost efficiency, VoIP is more and more popular largely over traditional telephone networks. VoIP cuts companies' monthly phone bill by approximately fifty percent. In addition to its cost

efficiency, VoIP technology ensures many advanced features, like conference calling, IVR, call forwarding, automatic redial, call recording, etc. without extra fees.

VoIP offers cheaper international long distance rates that are generally one-tenth of what is charged by traditional phone companies.

Due to its portability VoIP is a really good option to avoid expensive hotel phone and cell phone roaming charges. Only a high speed broadband connection (and a plugged adapter) is needed and anyone can reach you at your local number - independently of your location. Most of the times in-network calls to other VoIP service subscribers are free even if the calling parties are located in different parts of the world.

By using Internet connection for both data traffic and voice calls, it is possible to get rid of one monthly payment that are usually charged by most Internet service providers. In addition, the Internet-based voice and data transmission enables to avoid wireless roaming fees and long distance rates.

3. What are the advantages of VoIP?

In addition to its cost efficiency, the feature-rich services and the metaphorical disappearance of geographical boundaries as that were mentioned above, VoIP has many other benefits as follows:

- VoIP technology enables to detect and process touch tones and DTMF responses
- VoIP systems can be automated easily
- VoIP systems allow to use more than one codec
- VoIP provides rich media service as more file formats can be used with these systems
- VoIP ensures a much more flexible system than hardware based solutions
- Most VoIP service providers provide a user control interface, typically a web GUI, to their customers so that they can change features, options, and services dynamically.
- VoIP protocols run on the application layer and are able to integrate or collaborate with other applications such as email, web browser, instant messenger, social-networking applications, etc.

4. How does VoIP work?

VoIP is usually based on the SIP system that is the recognized standard. Any SIP compatible device can talk to any other. Any SIP telephone can call another over the Internet - any additional equipment is not needed for this. You only need to plug your SIP phone into the Internet connection, configure it then dial the other person. You can also connect traditional analog phones to your VoIP network – in this case an ATA device is needed.

In VoIP systems, your analog voice is converted into packets of data (as little files), and then transmitted to the recipient through the Internet and decoded back into your voice at the other end. To make it quicker, these packets are compressed before transmission, a bit like zipping a file (it will be decompressed of course at the other party).

The advantages of converting analog signals into digital data can be summarized as follows: Digital format can be better controlled as it can be compressed, routed, converted, etc. In addition, digital signals are more noise tolerant than analog signals. Quality of Service (QoS) ensures real-time errorless data streaming that allows interactive data voice exchange as well.

5. What is the actual cost of VoIP telephony?

If you only want to use VoIP to communicate with other users in your VoIP network, you can do that free of charge. If however you want to be able to use VoIP to make and receive calls to/from people who are out of your VoIP network or do not have VoIP, you will need to subscribe to a VoIP service provider plan, and a gateway service may be also needed that provides a bridge between VoIP and the conventional phone networks.

6. Is it possible to replace the current traditional corporate PBX with a VoIP one?

Definitely yes. VoIP is a very cost-effective option for those companies who want to upgrade their old PBX systems and VoIP ensures new features that traditional PBX systems simply do not. To change to a VoIP system, companies can buy an IP PBX, but it is also possible to add some VoIP functionalities into an existing phone system.

7. What kind of equipments do I need for creating a VoIP system?

Getting started with VoIP is fairly simple. Assuming that you already have the 2 most important ingredients (a Windows PC or Mac computer and a broadband Internet connection), all you need to get started is the following:

- Some telephone or messaging software
- A microphone
- Headphones or speakers

You can use a headset of course rather than a microphone and speaker to leave your hands free.

In order to choose which software to use, it is worth to consider the followings. Using voice chat in G-Talk or Yahoo Messenger could be regarded as VoIP, so could the highly publicised Skype; but these are all proprietary systems. You can download them free of charge, but to talk to someone using G-Talk, the person at the other end also needs G-Talk. The same applies to Yahoo and, to a great extent, to Skype. They use their own special system that is not open and will not connect to other systems easily. So – especially for corporate using – it is rather recommended to use such a softphone as [ConterPath X-Lite](#) with a SIP enabled IP PBX or access to an Internet Service Provider.

8. Which protocols describe VoIP connections?

VoIP has been implemented in various ways using both proprietary protocols and protocols based on open standards. You can see the VoIP protocols below:

- H.323
- Media Gateway Control Protocol (MGCP)
- Session Initiation Protocol (SIP)
- H.248 (also known as Media Gateway Control (Megaco))
- Real-time Transport Protocol (RTP)
- Real-time Transport Control Protocol (RTCP)
- Secure Real-time Transport Protocol (SRTP)
- Session Description Protocol (SDP)
- Inter-Asterisk eXchange (IAX)
- Jingle XMPP VoIP extensions
- Skype protocol

- Teamspeak

The most commonly used one is SIP. Session Initiation Protocol is IETF signaling protocol used for VOIP and other text and multimedia communication sessions such as voice and video calls over Internet Protocol (IP).

SIP can be used for creating, modifying and terminating two-party (unicast) or multiparty (multicast) sessions. These sessions include Internet telephone calls, multimedia distribution, multimedia conferences, instant messaging, file transfer and online games.

9. How to get started on VoIP programming?

The best way for creating any VoIP application is using a VoIP development kit. These SDKs are intended to provide a background support for your VoIP project by offering prewritten VoIP components. It is quite effective and comfortable to use these prewritten components, as you can save time and money as well. (During the softphone development that will be described below, [Ozeki VoIP SIP SDK](#) has been used for this purpose, that supports all the .NET programming languages, so C# as well.)

For using these toolkits, you have to add your preferred SDK as reference in you IDE. After you have added it, you can reach all VoIP components that are needed to be able to define the behaviour of such VoIP applications as softphones, call recorders, IVR menu systems, software-based IP phone systems (PBX), etc.

10. How to make VoIP calls?

In order to be able to make voice calls by using your own software application you need to connect your system to the telephone network. This can be done in three ways:

- **Option 1: Use a VoIP telephone adapter**

A VoIP telephone adapter is a hardware device that can be connected to your Ethernet LAN or to your PC. There are VoIP telephone adapters for GSM lines, for standard analog telephone lines and for ISDN lines. When you connect this hardware to your Ethernet LAN, it will receive an IP address. You need to configure this IP address in your VOIP SDK.

- **Option #2: Use a SIP Account provided by VoIP telephone service provider**

There are many VoIP telephone service providers worldwide that offer phone service over the Internet. You need to subscribe to their service, and you will receive a SIP account (including an IP address, a username and a password). You need to configure the SIP account details in your VoIP SDK.

- **Option #3: Use your existing office PBX if it is a VoIP system.**

If you already have an IP telephone system, you need to connect your VoIP SDK to that over the LAN. The SDK can log in to the phone system by using a SIP account and it can make telephone calls just like any other office desktop phone would.

Code example: How to create a C#.NET softphone?

Concerning to the fact that VoIP technology allows you to be able to communicate through the Internet, you will need a device that makes this telecommunication possible. That is why a softphone will be needed in case of each VoIP application – even if its GUI is not necessary. If you want to develop a simple software-based VoIP telephone, an autodialer program or a call recorder application, it is evident that you need a phone. But in case of other applications (such as IVR, PBX, call center, etc.) it is also required to have a softphone to be able to make and receive VoIP phone calls.

In the following section I will introduce briefly how to create a softphone using C#.NET. (All parts of the explanation below can be followed in the code snippet that has been inserted under the textual section.)

For building a softphone

First of all, you will need some using lines. After this, you need to create a softphone object and a phoneline object by using the ISoftPhone and the IPhoneLine interfaces. Then you need to specify the SIP account to be used for the phone line (you can find detailed explanation about this process in lines 14-34 – the commented parts will help to understand).

After finishing the SIP account specification, you need to define the PBX which the softphone will be registered to (see line 42). In most cases it is needed to be able to communicate through firewalls. Line 43 shows how you can implement it. 'None' in this line means that the communication will be completed on local network.

Now you need to configure the previously created phoneLine object to be able to communicate and to register to the PBX (see lines 44-63). When the application is running, the state of the phone line state can change. For following these changes, you need to listen to its change event. The last step of this task is calling the RegisterPhoneLine() method to register the phone line to the softphone. If the registration is required, this method will also send the SIP REGISTER command. The registration procedure will happen asynchronously in the background, and your program will be notified about the success or failure via the mySoftphone_PhoneLineStateChanged callback method. If the state of the phone line is RegistrationTimedOut (the allowed time for registration is up) or RegistrationFailed (the registration is required, but it could not be made), the 'Registration failed!' status message will be printed. If the state of the phone line is RegistrationSucceeded (the registration is needed and it was successful) or NoRegNeeded (the registration is not required), the 'Registration succeeded - Online!' status message will be printed.

```
1. using System;
2. using System.Threading;
3. using Ozeki.VoIP;
4. using Ozeki.VoIP.SDK;
5. using Ozeki.Network.Nat;
6.
7. namespace SIP_Register
8. {
9.     class Program
10.    {
11.        static ISoftPhone mySoftphone; // softphone object
12.        static IPhoneLine phoneLine; // phoneline object
13.
14.        private static void Main(string[] args)
15.        {
```

```

16. //Create a softphone object with RTP port range 5000-10000, and SIP port 5060
17. mySoftphone = SoftPhoneFactory.CreateSoftPhone(5000, 10000, 5060);
18.
19. // SIP account registration data, (supplied by your VoIP service provider)
20. var registrationRequired = true;
21. var userName = "sipusername";
22. var displayName = "sipdisplayname";
23. var authenticationId = "authenticationid";
24. var registerPassword = "Password";
25. var domainHost = "pbxip.voipprovider.com";
26. var domainPort = 5060;
27.
28. // Send SIP registration request
29. mySoftphone_Register(mySoftphone, registrationRequired, displayName, userName,
30. authenticationId, registerPassword,
31. domainHost, domainPort);
32.
33. // prevents the termination of the application
34. while (true) Thread.Sleep(10);
35. }
36.
37. static void mySoftphone_Register(ISoftPhone softphone, bool registrationRequired, string
38. displayName, string userName,
39. string authenticationId, string registerPassword, string domainHost, int domainPort)
40. {
41. try
42. {
43. var account = new SIPAccount(registrationRequired, displayName, userName,
44. authenticationId, registerPassword, domainHost, domainPort);
45. var natConfiguration = new NatConfiguration(NatTraversalMethod.None);
46. phoneLine = softphone.CreatePhoneLine(account, natConfiguration);
47. phoneLine.PhoneLineStateChanged += mySoftphone_PhoneLineStateChanged;
48. softphone.RegisterPhoneLine(phoneLine);
49. }
50. catch (Exception ex)
51. {
52. Console.WriteLine("Error during SIP registration: " + ex.ToString());
53. }
54. }
55.
56. static void mySoftphone_PhoneLineStateChanged(object sender,
57. VoIPEventArgs<PhoneLineState> e)
58. {
59. if (e.Item == PhoneLineState.RegistrationTimedOut || e.Item ==
60. PhoneLineState.RegistrationFailed)
61. Console.WriteLine("Registration failed!");
62.
63. if (e.Item == PhoneLineState.RegistrationSucceeded || e.Item ==
64. PhoneLineState.NoRegNeeded)
65. Console.WriteLine("Registration succeeded - Online!");
66. }
67. }
68. }

```

Hereby the first step has been carried out. To be able to make and receive phone calls by using your own softphone, you only need to add some new lines to your code as follows:

For making a phone call

To be able make a phone call, first you need to create a call object, subscribe to the events of the call, and call the Start() method of the call object. As you can see below, first of all the softphone checks whether there is an active call or not. If not, the correct methods will be called.

```
1. if (call == null)
2. {
3.   call = softphone.CreateCallObject(phoneLine, numberToDial);
4.   WireUpCallEvents();
5.   call.Start();
6. }
```

Please note that calls are being handled as objects, more specifically as IPhoneCall objects.

For receiving a phone call

The softphone has an IncomingCall event. After you have subscribed to this event, you will be notified if there is an incoming call. When this event occurs, you can accept the call by calling the Accept() method (or you can reject that by calling the Reject() method). (For hanging up any call, use the HangUp() method of the call object.)

```
1. softphone.IncomingCall += softphone_IncomingCall; // Subscribe to the IncomingCall event
```

Note for the code snippets: Please keep in mind that the code snippets above have been written by using the prewritten VoIP components of the Ozeki SDK. So if you want to try them out, you need to install its [freely available trial version](#).

Conclusion

In my opinion, in today's business world, a basic knowledge of VoIP is essentially needed if you are interested in system networking – whether you are CEO, system administrator or software developer. Certain level of knowledge can be required during a developer job interview if the potential employer is interested in VoIP in any way. In this article I run through the most common VoIP-related interview questions and I have presented a simple softphone example project in C# that you can use as a first step if you would like to get a taste into this field of software development.

Further reading and references

To draw up this VoIP interview guide and this short C# tutorial I used the following knowledge base:

1. [Introduction to VoIP technology](#)
2. [Voice over IP - in the Wikipedia Encyclopedia](#)
3. [What is VoIP and how it works?](#)
4. [Voice over Internet Protocol \(VoIP\) Interview Preparation Guide](#)
5. [Download Ozeki VoIP SIP SDK to reach the necessary VoIP components](#)
6. [Download CounterPath X-Lite to test your application](#)