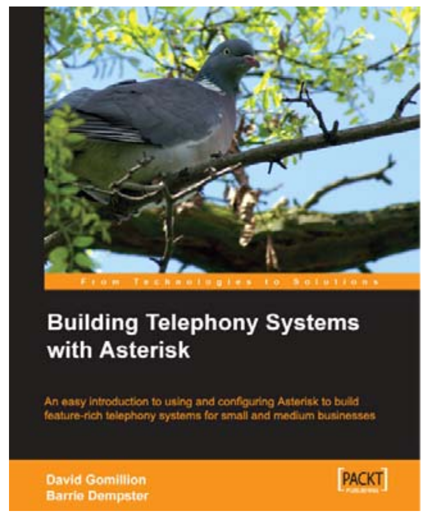
Building Telephony Systems with Asterisk

David Gomillion

Barrie Dempster



Chapter 2
"Making a Plan for Deployment"

In this package, you will find:

A Biography of the authors of the book

A preview chapter from the book, Chapter 2 "Making a Plan for Deployment"

A synopsis of the book's content

Information on where to buy this book

About the Authors

David Gomillion currently serves as Director of Information Technology for the Eye Center of North Florida. There, he orchestrates all of the technological undertakings of this four-location medical practice, including computers, software (off-the-shelf and custom development), server systems, telephony, networking, as well as specialized diagnostic and treatment systems.

David received a Bachelor's of Science in Computer Science from Brigham Young University in August, 2005. There he learned the theory behind his computer experience, and became a much more efficient programmer.

David has worked actively in the Information Technology sector since his freshman year at BYU. He has been a Networking Assistant, an Assistant Network Administrator, a Supervisor of a large Network and Server Operations unit, a Network Administrator, and finally a Director of Information Technology.

Through his increasing responsibilities, he has learned to prioritize needs and wants, and applies this ability to his Asterisk installations.

Barrie Dempster was a Network Administrator/IT Manager for a growing call center when he saw the convergence and dependence of telephony and IT-related fields on each other. He focused on integration of telephony with IT infrastructure, and took on security as a career. The increase of voice-over-IP communications has now led to high demand for these skills, which he now utilizes in his current position as a Scotland-based Infrastructure and Security consultant for a variety of clients primarily within the financial sector.

He has been involved in varied projects, from building and deploying web and database servers to creating custom communication and conferencing systems, most of which are secured highly in order to survive public networks. He has deployed and used a variety of PBX systems and, as a strong supporter and user of free and open-source software, has a serious interest in Asterisk as it combines all of these interests into one extremely powerful package.

2

Making a Plan for Deployment

Now that we have chosen Asterisk to meet our needs, we need to determine our course of action. We will go through some common requirements, discuss the most common choices for solutions, and finally make an informed decision. As we go along, we should make notes to help us on our way.

The Public Switched Telephony Network (PSTN)

Most of the telephones in the world are connected to a vast network, enabling any telephone to reach any other. This network is called the Public Switched Telephony Network (PSTN). The phones that are on it are all reachable by dialing a number, which may include country codes, area codes, and telephone numbers.

While there are instances in which interconnection with the PSTN is inappropriate, most users of telephones have the expectation that they can reach the world at large. Therefore, we will consider interconnection to the PSTN as a requirement.

Connection Methods

There are a number of different methods to connect to the PSTN. Each has advantages and disadvantages, most of which we will touch on. Since pricing varies depending on city or country, exact pricing will not be given. Pricing should be researched based upon the location of the Asterisk server.

We will handle each connection method one at a time.

Plain Old Telephone Service (POTS) Line

Probably the most common connection to the PSTN is a POTS line. This is an analog line, provided by a telephone carrier. Each POTS line can carry only one conversation at a time.

For small installations, POTS lines are usually the most cost-effective when connecting directly to our Local Exchange Carrier (LEC), a term used to refer to any company providing local telephone service. Eight lines is usually the point at which we should look seriously at another technology for our connection.

POTS lines from our LEC require a Foreign eXchange Office (FXO) interface to be usable in Asterisk. We will focus on Digium's offerings, namely the FXO module on a TDM400P. Each TDM400P can use up to 4 modules; therefore, if we have 1 line, we will have 3 empty module slots on the card.

Integrated Services Digital Network (ISDN)

ISDN is an all-digital network that has been available for over a decade. It is available in two major versions: Basic Rate Interface (BRI) and Primary Rate Interface (PRI).

ISDN divides a line into multiple channels. Each channel can contain either payload (Bearing, or B Channel) or signaling (Data, or D Channel). A BRI has 3 channels: 1 D channel and 2 B channels. Therefore, two phone calls can be in progress at a time on a single BRI. A PRI has 24 channels: 1 D channel and 23 B channels, resulting in up to 23 simultaneous calls.

ISDN is not limited to voice alone. Each channel can carry 64k of data, if so configured with the LEC. This gives ISDN a lot of flexibility over POTS lines, as the channels can be reconfigured between voice and data on the fly.

With its separate D channel, ISDN is able to do things POTS cannot, such as setting custom caller ID, receiving dialed number information, on-the-fly redirection of calls, and a whole host of other cool features. Of course, all of these features require cooperation from the LEC, which is not always forthcoming.

BRI does not have high penetration in the United States market. Some accuse LECs of vicious pricing, while others claim consumers are to blame for fearing new technology. Either way, the result is the same: if we call our LEC and request a BRI, they will assume it is for data.

PRI, on the other hand, is widely used in the US. It is the connection of choice for larger installations. PRIs are actually delivered over T1 connections, a proven technology. Although the author has many contrasting stories, T1s are usually very reliable.

T1 or E1

Technically speaking, when ordering service from an LEC, we order a DS1, which is delivered over a line referred to as a T1. However, this detail is usually overlooked. Therefore, we will refer to it in its vernacular: a T1.

A T1 is a line with 24 channels. Each channel can contain a call. Therefore, a T1 can contain an additional call when compared with a PRI. In Europe, E1s are more common. Much like a T1, they have 32 channels instead of 24. T1s have to signal the call somehow. They way they do this is through Robbed Bit Signaling. What this means is that a bit is robbed from time to time, as information needs to flow about the connection. While this is usually imperceptible to the human ear, it can be deleterious to data connections. Using a T1 to deliver both data and voice is common. Some of the 24 channels are designated to be used for data, and others are used for voice. There may even be unused channels. LECs are able to offer lower pricing when bundling services in this way, as a few channels may be voice, others used for an Internet connection, and yet others could be used for a private data connection to another office.

LECs are able to send information about the number that was dialed at the beginning of the call. In this way, one advantage of the PRI has been matched by T1s. If we intend to have about 8 to 12 lines as well as a Data connection, a T1 can be a good choice.

Voice over IP Connections

In recent years, a new way to connect to the PSTN has cropped up. Companies are using PRIs, T1, and other technologies to connect to the PSTN, and then resell those connections to consumers. The users connect to the companies offering these connections through Voice over IP technologies. By so doing, we can skip dealing with LECs completely.

This service is called Origination and Termination. Through these services, we can receive a real telephone number, with the area code depending on what the provider has access to. Not all providers can offer numbers in every locality. This means that our number could be long distance from our next-door neighbor, yet local to someone in the next state. The advantage of this, however, is that the provider will route most of the calls over their VoIP infrastructure and will then use the PSTN when they get to their most local point at the receiving end, which can mean that long distance charges are dramatically reduced. If we call a variety of countries, states or cities it can be worthwhile to research a provider that offers local PSTN access to the areas we call most.

The rates per minute are usually very attractive. Often, long distance is at the same rate as local calls. One thing to watch out for is that some providers charge for incoming minutes, much like on a cellular telephone, and some providers also charge for local calls.

Another thing to be aware of is that some providers require that you use their Analog Terminal Adapter (ATA). This means that they will send you a box that you plug into the Internet, which speaks Voice over IP. Then, you have a POTS line to connect a phone (or Asterisk) to.

Voice over IP makes sense in many installations. But for the quality to be acceptable, a reliable Internet connection with low latency is required. Another thing to watch out for is jitter. Jitter refers to the latency from packet to packet. Most protocols can handle latency a lot better if it is constant throughout the call.

A good candidate for Voice over IP is a site where interruptions in service will not endanger life and will not irreparably harm the company. While VoIP providers strive to achieve very high availability, we also have to rely on the Internet at large and our VoIP provider's ISP, as well as our own ISP.

If our telecommunication needs are such that periodic downtime is tolerable, VoIP will probably be our least expensive option. It requires less hardware in our Asterisk system as well, increasing the savings: to use VoIP with Asterisk all we need is a system capable of Internet access; we don't require any specialized telephony hardware.

Determining Our Needs

Now that we have examined some of the options, we need to determine what our needs are. Requirements will vary quite a bit from site to site. Something to keep in mind is that, although the previous choices are distinct, they can be mixed in an Asterisk installation. We can have VoIP providers and POTS lines, as well as a PRI if we desire. It's very common to have this type of setup; for example if we have an office in another country we can call them using VoIP but all local calls could use POTS. It is important to understand the calls our system will be making and where they will be going, so that we can arrange for the necessary services and ensure that the calls are routed accordingly. If we have an existing telephony system, we can take a look at the calls it's making just now and our current costs so that we can determine what technologies will be of most use to our system's users.

Now is the time to begin documenting what our plan is. If Asterisk is to replace an existing system, then we should start by writing down all of the current lines coming into our incumbent PBX. Once that is done, we need to look at what our requirements are.

First, we need to determine how many lines are needed. Telephone providers can generate a usage report that will tell us the maximum concurrent connections we have experienced in the last month. While they are able to do this, many providers are not very happy to have to run such a report; however, without that information we have nothing to gauge our needs other than gut feelings.

If we need more channels than we have, someone will get a busy/congested signal. Therefore, we should plan to have the maximum number of channels we have used plus a reasonable cushion. 125% of our current maximum is usually a reasonable cushion, this allows for instant 20-25% growth so that we can accommodate a sudden increase in calls without the system falling over. If we do increase calling to this level for a relatively long period, we must consider an increase in lines to prevent congestion. These numbers are a guideline and they can change depending on circumstances. In a call center where the main business purpose is to make and receive calls, 150% may be a more satisfactory figure. We also should take into account the time it takes to get new lines set up from our local operator. If a significant event occurs that generates a large number of calls we should have the capacity to handle this or be able to increase capacity quickly.

Now that we have a number of lines, we need to determine the technology to use for each. VoIP is usually cheapest, especially for long-distance calls. PRI is usually the most reliable, and for incoming calls is often cheaper than VoIP.

While pricing the options, we need to remember that POTS lines usually only have a single phone number, while a PRI can have hundreds of phone numbers. If we are a business that receives only a few calls, but needs the calls to have different phone numbers, then a PRI probably makes the most sense. Also with a PRI we can trunk more effectively, which may become essential.

Although a PRI can have hundreds of phone numbers, there is a charge for each number each month. Called DID (Directed Inward Dialing) numbers, these "virtual" numbers are usually sold in blocks of 10-20. If we do not order enough to begin with, it is usually not difficult to get new DIDs ordered; often, they can be available the same week, depending on the phone company. We assign these numbers to individual devices or groups of devices ourselves, once we have them allocated. This means we can decommission or reallocate numbers as necessary. We may have campaign DIDs that are reassigned to different groups depending on the current campaign, personal DIDs for key staff or our main DID, which would probably be assigned to a group of people responsible for handling these calls.

We should take this opportunity to write down what lines we want, what phone numbers we need, and get quotes if it differs from the currently installed PSTN connections.

Terminal Equipment

Now that we have decided on our PSTN interconnection, we need to decide on our internal connections. Our PBX can have modems, fax machines, hardware and software telephones, and other PBXs connected. We will refer to all these different machines as Terminal Equipment.

Types of Terminal Devices

There are four major types of Terminal Equipment: hard phones, soft phones, communications devices, and PBXs. We will cover each type briefly.

Hard Phones

The term Hard is short for Hardware. Hardware phones are physical devices that act as a telephone handset. Hard phones are available for POTS (as used in the typical household) or VoIP.

Voice over IP uses various protocols, depending on the handset, PBX, and the requirements. The major protocols supported by Asterisk are as follows:

H.323

The first protocol we will be looking at is H.323. Formally known as "ITU-T Recommendation H.323: Packet-based multimedia communications systems", this is a suggestion on how to accomplish conferencing over IP, which includes voice, video, and data. This recommendation actually came out at about the same time as SIP but has been more widely implemented.

The H.323 standard enjoys full backwards compatibility. Currently H.323v5 is out, and v6 is being discussed. Each new release keeps all of the pieces of the previous version. This gives a clear upgrade path and some assurance that equipment won't be quickly antiquated.

H.323 equipment is widely available. From gateways to telephone handsets, all of the needed equipment is relatively easy to find. Most of the telephone handsets are full-featured because the H.323 protocol has a robust feature set.

While the H.323 standard was not designed for wide-area networks, a whole set of rules allowing cross-domain addressing have been created. A system for reporting Quality of Service (QoS) back to a server has also been developed, allowing such information to be used to route future calls.

Finally, H.323 as a standard supports call intrusion. New endpoints can be added dynamically to any conference (i.e. a call) at any time.

Asterisk support for H.323 is not built in. Instead, an additional package, `asterisk-oh323`, must be installed. After installation, H.323 handsets and gateways can be addressed much like any other channel in Asterisk.

SIP

The Session Initiation Protocol, or SIP, is another method of signaling VoIP calls. SIP is part of the default installation of Asterisk.

Most of the newer VoIP equipment is supporting the SIP protocol. It has a number of advantages. One such advantage is that the code is smaller. The reason for this is that SIP only supports very basic features. All advanced features are supported through separate Internet standards. Another reason for its small footprint is that, as features are deprecated, the code to implement them is ousted.

Another advantage of SIP's design is its modular nature; as such, extending the protocol is easier to do. It also scales better and was designed with a large network in mind.

SIP seems to be the future of VoIP. There are many features that H.323 has but which are not available on SIP, though. This includes handset conference control, better Media Gateway definitions, and data sharing. However, SIP is a very good protocol for simple phone calls. Also, since we are using Asterisk, conferences are controlled by Asterisk, not the handsets. Asterisk is a clear Media Gateway, and when used as such, the ambiguity in SIP is not an issue.

IAX

The Inter-Asterisk eXchange (IAX) protocol is a protocol created by the programmers who brought us Asterisk. Because of the limitations of SIP and H.323, they chose to create a new de facto standard that would allow Asterisk servers to accomplish many things that are simply impossible with the other standards. They also support some features that are extremely difficult to do in SIP and H.323.

First, IAX pierces Network Address Translation (NAT) easily. Most firewalls and home Internet gateways use NAT, as well as some service providers. SIP and H.323 have worked hard to develop standards to allow them to break through the different types of NAT; however, IAX can work through most NAT devices right out of the box.

IAX is more configurable than the other protocols when dealing with Asterisk. Since the source code is available, we can modify it if we so desire, and then submit those changes to be evaluated for inclusion in future versions of Asterisk. Since IAX is not currently an Internet standard, per se, there is no standards body to work through, allowing more rapid improvement and growth.

IAX supports the trunking of calls. This means that multiple calls can be combined through a single stream. Through the trunking capability, a significant amount of bandwidth can be saved by not having the overhead of multiple streams.

IAX connections between servers support the switch command, with which information on how a call is routed can be efficiently shared between Asterisk servers.

IAX supports a large number of codecs. Any codec supported in Asterisk can be used with channels of this type.

Because IAX is an Asterisk-created protocol, there are not many handsets and gateways available. However, as time goes on, more and more devices are supporting the IAX protocol.

Just as a note, we sometimes see IAX and IAX2 differentiated. IAX2 has been merged into IAX, and IAX has been deprecated. Thus, if a device claims to support IAX2, it should really be supporting IAX.

Soft Phones

Much as hard phones are phones implemented in hardware, soft phones are phones implemented in software. Using all the same protocols available to hard phones, soft phones are far less expensive to implement. By using the general-purpose computing resources of a personal computer, the expensive proposition of replacing all telephones in a building can be avoided.

Before going further, we should recognize that most hard phones are in all actuality soft phones combined with bit of special-purpose hardware. The computing power of a hard phone is not as vast as that of a PC, but unlike a PC is specially tuned for carrying voice. Thus, we should not dismiss use of hard phones immediately.

The sound quality experienced on a soft phone will depend greatly on the available resources on the PC, the quality of the software used, and the quality of the data network between the PC and our Asterisk server.

Soft phones will have a hard time being accepted by some users, it is true. In addition to the political issue of having people use their computer to talk on the phone, we also have to address disaster planning. If we lose power, keeping a computer up that draws in excess of 400 watts will be far more difficult and costly than keeping power to a hard phone that draws 15 watts, especially for prolonged outages.

The most significant advantage of the soft phone is cost. In most businesses, desks contain a computer and phone at a minimum. If you can remove the phone there is an obvious reduction in hardware costs. There are a variety of soft phone products available and most operating systems come with a basic soft phone package by default. There are also a variety of open source products available. The choice of product, soft or hard, is equally as important as the PBX. You must be sure that the users will use the device and be sure that it will be reliable and supportable.

Communications Devices

Dedicated communications devices, such as modems and fax machines, are still very prevalent in business today. While these devices could be replaced with more modern, more reliable, and faster technologies, the new technologies have not yet been embraced.

Most of these devices will be analog (meaning they will require a POTS line). As mentioned before, a T1 can connect 24 lines, and a POTS line can only connect 1 line. With a device called a channel bank, a T1 can be split out into 24 POTS lines. When we require many POTS lines, channel banks are usually cost effective.

Communications devices do not all use analog signaling. One such device would be a T.38 fax gateway. This protocol allows regular faxes to be sent over UDP. At this time, Asterisk does not support the T.38 protocol, but hopefully will soon.

One extra note about faxing: Asterisk supports receiving and sending faxes via an add-on called SpanDSP. With this, Asterisk can receive a fax and turn it into a TIFF file. This TIFF file can then be further processed to become a PostScript or PDF file and e-mailed to the proper recipient. The installation of this add-on is not covered here, as it is changing rapidly.

These communications devices are usually supported for legacy reasons. We should continually strive to reduce outdated technologies and replace them with up-to-date solutions.

Another PBX

We can connect PBXs together to provide services to users hosted on another PBX. We can use SIP, PRI, T1, H.323, or IAX to connect the PBXs.

If we are connecting multiple Asterisk PBXs, we should use IAX. The IAX protocol has a number of features with this specific use in mind, such as the ability to have multiple conversations trunked into the same UDP stream, yielding greater efficiency.

Choosing a Device

Now that we have seen the broad offerings of terminal devices, we see how difficult it can be to choose one to meet our needs. After choosing a type of device, we then have to choose a manufacturer and model. This task can be daunting. Let's take a few minutes and discuss how we will make the best decision based on the available information.

Features, Features, and More Features…

As we review available phone handsets, we will be inundated with all the features that manufacturers can throw at us. These lists are overwhelming, even to the most seasoned experts. It is very difficult to compare two handsets solely on features, as some features have different names.

Determining the usability of a particular phone handset should be a straightforward process. This process has four major steps: requirement elicitation, prioritization, and documentation, followed by handset testing.

Requirement Elicitation

This is the brainstorming step. We should go to each user and determine what his or her needs are. We ask the user what features he or she uses on the current phone. We observe that person working for a period of time to get a good sampling of what he or she actually does.

We should then go to the user's manager and see what a person in that position is expected to do. We add these features to our list. While this list will be unique to each user, many will be very similar. We should see patterns of usage emerge between groups of employees.

Requirement Prioritization

In this step, we take our requirements list from the previous step and, working with the user and manger, determine which features are used most, which are most important to that user's role in the organization, and which features are simply nice to have. We should also attempt to recognize any deficiencies in the current technology. Changes are often embraced if the change adds value to the user by making a task easier or in some cases removing a task entirely. It's important that we recognize all nuances of the current system in order to provide the user with a replacement that will suit them.

We then should create a quantitative scale for each feature. For example, if we were working with an operator, a transfer button would be a 10, while a Do Not Disturb button will probably be a 1. If we had a phone handset with both, it would score an 11. By putting numbers on the required features, we can come up with a quantitative answer to a very subjective issue.

Requirement Documentation

This step is most important of all of the steps thus far, especially for consultants. We take the list of requirements and their weights and write them in a short document. We then have the user and the manager sign it off to indicate their agreement.

This may seem a little formal for picking a telephone handset, but it is an effective method of communicating expectations and plans between you, the implementer, and the users. This can help to prevent surprises or differing recollections of what was promised.

Phone Testing

This is the final step. After comparing the available handsets against the document we created in the previous step, we choose the highest scoring handset. We then take a handset of that type to the user and have him or her use it (if we have a test system installed by this point) or at least sign off on it conceptually.

Again, this is an opportunity to ensure our users' expectations are reasonable, that commitments are clearly defined, and that our users are kept informed during the decision-making process. It can also help us get buy-in from the users as we make the major adjustments that will invariably accompany a new phone system.

Determining True Cost

When we look at what handsets to compare our requirements document against, the issue of cost will have to be looked at. Before we offer a handset that would not be possible under our project budget, we should determine that the handset meets all of the requirements of the business, which includes the element of cost.

The issue of cost is not as simple as looking at the retail price of a handset. Each type of phone will have multiple types of cost. These costs will usually fall into one of the following categories:

- **Handset cost:** This is the easiest cost to determine. It is the actual amount of money that will have to be spent to acquire the telephone.
- **Port cost:** This cost is the element of what the phone connects to on the other end. On a VoIP phone, for instance, this could be a portion of the cost of a new network switch that supports Quality of Service (QoS) to enable reliable voice communications.
- **Headset cost:** If a phone will require a headset, then we should consider the cost of that headset as we choose the phone. Different connectors are available depending on the model.

- Software license cost: Some phones will require the purchase of G.729 licenses. Other phones may require a license for the software on the phone (usually referred to as firmware). We should not fail to consider this cost while computing the cost of the phone.
- Installation cost: Different phones require different amounts of time to install. That time translates into cost.

By considering each of these factors for each different handset, we get an idea for the true cost of each particular phone. With all of these costs defined, we can see which phones are within our budget and which are simply too expensive.

Compatibility with Asterisk

Not all handsets interoperate equally with Asterisk. Referring to the Asterisk Users mailing list archive, we can ensure that no serious incompatibilities have been discovered. Also, a wiki is available at `http://www.voip-info.org`. A vast array of useful information about Asterisk is available there. The site is searchable and is constantly updated.

We do not have to select a single protocol for all VoIP phones. Instead, we can mix and match protocols to our best advantage, thanks to the flexibility and power of Asterisk.

Sound Quality Analysis

Sound quality is a very subjective thing. Each user will have a personal threshold between acceptable and unacceptable.

Each phone will have varying sound quality. The variables that can affect the quality of a call are staggering. Network latency can significantly affect sound quality, but so can configurations of the phone. Determining which is the cause of low sound quality can be difficult to do.

Build quality from a manufacturer can also affect quality. When wide variations are allowed from one phone to the next, the result is usually inconsistency from handset to handset. Thus, we have to choose a manufacturer we can trust.

While there is no absolute, the quality of sound on telephone handsets, from highest to lowest, is usually as follows: analog hard phone, VoIP hard phone, analog soft phone, VoIP soft phone. If you are doing a comparison between different handsets, the main things to pay attention to are the amount of background noise (or hiss), distortion, drop outs, popping, and highly digitized voice. If we have users who are extremely sensitive to sound quality, analog will probably be our best bet. For those users who are a little more forgiving, VoIP allows us to use one network for our phones and our computers.

When determining what terminal equipment to use, we need to consider the sound quality of each device and match it against the needs and expectations of our users, and temper that with the cost of that device as compared to the budget.

Usability Issues

The world's most advanced VoIP handset is absolutely useless if our users cannot figure out how to use it. As we decide what equipment to provide for our users, we should consider where they are at in the continuum of technological awareness. While VoIP hard phones with context-sensitive buttons are useful for most users, some people find the interface confusing and frustrating.

This is one big issue that we need to address in the handset testing that we do after eliciting the requirements that our user has for a new phone. We have a duty to ensure that our users can use the handsets we choose. We must be careful not to assume that they will figure it out, as doing so often causes hurt feeling and resistance to change. The success of Asterisk will be largely measured by the response of our users.

Recording Decisions

It is time to decide what kinds of terminal equipment we will use with Asterisk. First, we should make a list of all users of our phone system. Based on the requirements we get from them and their supervisors, decide what type of device to use, whether it is a hard phone or a soft phone. Next, we should determine a protocol to use. Finally, determine a brand and a model of phone to use.

We should take the time to write this down. This list should be provided to the decision makers, and kept up to date as changes occur, which they inevitably will. Again, doing so will keep everybody informed and reign in the expectations to keep them reasonable.

How Much Hardware do I Need?

This is probably one of the questions most frequently asked by those who are new to the world of Asterisk. The answer depends largely on what we are going to do with our system.

Conversations that bridge between codecs (called transcoding) take the most power to handle. Voice over IP conversations seem to take a little more processing power than straight Time-Division Multiple-Access (TDM) calls. Having our server run scripts to find information will take more power than if we define everything statically. How many different conversations we have going at a time will affect how much horsepower we need our server to have. As will the features we use.

Do you see the complexity of answering this question? We have to figure out what we are going to use before we can figure out how big a server we will need. That said, there are some good rules of thumb we can start off with.

First, while we can run an Asterisk server on an old Pentium 90 with 64 MB of RAM, why would we want to? We are creating a robust phone system. We do not have to pay to license the use of the software, and we do not have to pay per extension. We can go spend some of the money we saved and buy a decently powerful server.

As we select the components for our server, we need to remember that we are not building an e-mail server or a web server. We are creating a PBX that people are going to expect to be running *all the time*. We should select a stable chipset, with an up-to-date BIOS, and match it with other current high quality components. By using high-quality components, we increase the likelihood of ending up with a highly-availability phone system.

On another note, we should select a server with as much redundancy as possible. A RAID-1 controller could save our phone system in the event of a hard drive failure. A pair of RAID-1 controllers that are mirrored could save our phone system in the event of a controller failure or a PCI slot failure. A server with redundant power supplies will help us in the event of power failure or a power supply failure. Of course, our phone system should be on an Uninterrupted Power Supply (UPS). This is not only for protection from power failures; it will also protect from spikes, and often even lightning.

Depending on the reliability requirements, we might need a redundant server. There are hardware devices that will detect if a PRI is down and automatically failover. Then again, for most installations, this is overkill.

The most important lesson to keep in mind is that people have grown to depend on phone systems. We should not skimp on hardware as doing so could, in the long run, cost us dearly. With the unique pricing structure of Asterisk, all we will have to pay for is any additional hardware to get increased reliability and capacity.

Along with hardware, the question is often asked "Which distribution of Linux should I use?" If we already have experience with some distribution of Linux, we should be able to make Asterisk work with that distribution. Asterisk is very flexible and has been built with commonly available dependencies, and any distribution of Linux should work. That said, some distributions will require more effort to enable some features such as automatically starting Asterisk when the server boots. Since each distribution treats startup scripts differently, most distributions will require a minor amount of tweaking.

We should check the wiki at `http://www.voip-info.org` for more information on the distribution we intend to use. It has up-to-date notes on compatibility problems, caveats, known issues, and often workarounds for those issues.

Choosing the Extension Length

While creating our phone system, we will need to create a set of extensions. Although Asterisk has no such requirement, these extensions should probably all have the same length to give comfort to our users. We must determine the length that we will use for all of our extensions.

When creating extensions, it is often advantageous to group certain extensions together. For example, all sales extensions could be in the 200's, support in the 300's, management in the 100's, etc. Or we could go further and say that all first-tier support will be in the 3100's, the second-tier support will be in 3200's, third-tier support will be 3300's, and so on.

We should keep in mind that it is easier to add extensions when there is an available number than it is to renumber all extensions in a building because we have filled up all of our available dial strings. For instance, suppose we chose 1-digit extensions and have the following phone list:

```
0 – Operator
1 – Reception Desk
2 – Break room
3 – Conference room
4 – John
5 – Sally
6 – Jennifer
7 – Fax Machine
8 – Voicemail Access
9 – Outgoing calls
```

This system will work fine until we add another extension. When we add another extension, we will have to give a new extension numbers to all of our users.

Now consider the following phone list:

```
1000000 – Kitchen
2000000 – Bedroom
3000000 – Office
8000000 – Voicemail
```

In this house, for someone in the kitchen to call the office (think "Dinner's ready; will you please leave that computer and *come eat*?!?"), the user has to dial 7 digits to accomplish what could have been done with one.

Therefore, we need to be smart about how long we make our extensions. Often, if we are replacing a phone system, we should just adopt the numbering already in place to make the transition a little easier for our users. Some phone systems may not have had extension numbers before, such as old analog systems. All lines were simply visible from all stations. In those instances, we should be sensitive to the new learning that will have to take place and make the extension number length as small as possible.

We also need to consider some special instances. First, most people do not want an extension that begins with a 0. Simply put, nobody likes to be a nothing and having a leading 0 for anybody but the operator makes them feel emotionally put-down. Also, we should reserve all extensions beginning with a 9 as outgoing telephone calls. Add to that the need to provide services like call recording, conferencing, and voicemail access. We will give all such services a prefix, such as 8. Thus we see that we have already lost 30% of all of the available extensions.

A good rule of thumb in computing is to take what we believe we will use and triple it, and then round up. Thus, if we believe that at the height of our system, we will have 100 users, we should assume that we will have 300 users. If we believe we will never have more than 10 users, we should assume 30.

With this in mind, here is a table of what we will need:

Expected number of extensions	**Our assumed number of extensions**	**Length to use for extensions**
2	7	1
22	70	2
222	700	3
2222	7000	4

Keep in mind when reading this chart that it is much easier to have people dial an extra digit than it is to make them learn all new extension numbers. Thus, if we are a border case, we should go ahead and move on up to the next extension length.

Another idea that we can take advantage of is using an extension that gives a lot of information about the destination. Take for instance a corporation with 7 locations. The first digit in the extension could designate the location. Then the second digit can designate the department, and the remaining digit(s) can designate which member of the group is sought. Thus, knowing the structure and an extension can give an idea of where that person is and what he or she does.

In some environments, such information is not desirable. For instance, on a college campus, some employees work very late at night. If the extension gives their precise location, stalking and threats of physical harm can prove problematic. Therefore, we need to be sensitive to such concerns.

One alternative to these layouts is to use the last few digits of a phone number to refer to each extension. This can work very well if all of such digit strings are unique; however, it can cause problems. Suppose we chose a 4-digit extension and have the phone numbers 555-1234 and 777-1234. Which one is extension 1234? Or suppose we use 7 digit extensions and have (800) 555-1234 and (866) 555-1234. Which one is extension 5551234? Thus, some organizations have moved to a full 10-digit extension length. While this allows 10^{10} extensions, it can cause some users to complain about usability and convenience.

With the flexibility of Asterisk, we can choose many different ways to allocate extensions, all of which will influence our decision on extension length. We must balance our users' expectations with our desire to leave room to grow. By so doing, we can create extensions that are easy to maintain and user-friendly.

Summary

Now that we have decided to use Asterisk, we must make a plan. This chapter has looked at the different types of hardware that an Asterisk system needs, namely:

- What technology we use to connect to the PSTN
- What technology or technologies we use to connect our handsets to Asterisk
- What server hardware we will use
- How we will architect our extensions to be easy to use while also allowing for the growth we can realistically expect

As we draw up our plan, we must address each of these options, before moving on to the next stage, the installation of the Asterisk software itself, which we cover in the next chapter.

Building Telephony Systems with Asterisk

The field of information technology has seen an explosion in the number of new businesses and startups. It is, however, important to take a realistic look at the needs as well as constraints of such a setup before investing in a solution—to not get extravagant, and also to stay practical. This is the thought that is behind the development of the Small Business Server 2003—to provide a server with a range of functions specially optimized for efficiently running such businesses.

What This Book Covers

Chapter 1 introduces Asterisk and talks about the possible scenarios that would demand its usage, and the realistic trade-offs that you should consider when choosing it.

Chapter 2 discusses a basic deployment plan, and takes you through various aspects such as requirements and the how-tos of choosing the right terminal equipment and hardware.

Chapter 3 discusses installation of Asterisk. It starts with a section on preparing a system for installation, takes you through installation of necessary components, and ends with an introduction to the way Asterisk behaves.

Chapter 4 deals with the basic Asterisk configuration, and discusses the Zaptel interfaces in detail, and then the configuration of protocols and various features.

Chapter 5 deals with creating a dialplan. This involves creating a context and extensions, and the chapter also discusses advanced call distribution and automatic attendants.

Chapter 6 discusses quality assurance issues that concern most companies, and gives an overview of call detail records, call monitoring, and recording, etc.

Chapter 7 talks about Asterisk@Home—a simplified Asterisk solution that retains most of its functionality for its so-called "home" users—and a customer relationship management system, SugarCRM.

In *Chapter 8* we've shown a few case studies of working Asterisk-based phone systems, and have discussed scenarios for home offices and small businesses.

Chapter 9 deals with Asterisk's maintenance and security aspects. The topics range from backups of configuration files to disaster management plans to server security. This chapter also discusses Asterisk's scalability aspects and support channels.

Where to buy this book

You can buy Building Telephony Systems with Asterisk from the Packt Publishing website: `http://www.packtpub.com/asterisk/book`. This book carries at least a 10% discount on the website as well as free shipping to the US, UK, Europe, Australia & New Zealand.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.