Windows Telephony Backgrounder

Microsoft At Work

1. Introduction:

Nearly every business desktop has a phone adjacent to a PC, yet there is no integration between these devices. While modems are widely used for data transport, they do not provide control of the phone network beyond simple connections on an analog network. The PC today is not a full participant in the global telephone network.

There are a number of motivations for integrating telephones and PCs. Organizations and individuals seek more effective communications media, as well as ways to manage their communications more efficiently. This desire is illustrated by a range of telecommunications developments. Richer communications, such as videoconferencing, can convey more information than a normal telephone call, as well as save on travel time and costs. The proliferation of portable phones, car phones, answering machines, voice mail, and pagers represent a desire for uninterrupted communications, regardless of time or location. Moreover, the traditional voice network has been supplemented with a variety of other communications media like fax and email, forcing consumers to deal with disparate communications systems. The telephone network is also the widest of wide area networks and offers connectivity to a far greater number of people than any other network.

There have been a number of past attempts to combine the phone and the PC, all of which have failed because of hardware, software or cost deficiencies. With the advent of fast CPUs, graphical user interfaces and multitasking operating systems, integration is now realistic. Applying a graphical user interface and programmability to the telephone network enables a myriad of new applications.

This document provides a conceptual introduction to how telephones work, the benefits of integrating them with PCs, and what hardware and software is required.

2. Telephone Basics:

The telephone network differs from the packet-switched networks of the computer industry in several important ways. These distinctions are important because historically the computer and telecommunications industries have had a very difficult time understanding each other's network paradigm (and have each lost lots of money as a result).

The telephone network is circuit-switched, which means a connection is of fixed, two-way bandwidth that is available at all times. Voice signals must be delivered synchronously: a conversation is unintelligible if data are delivered out of sequence or at irregular intervals. Thus, packet delays on a shared network are not acceptable. Achieving this "guaranteed bandwidth" has its costs. Every connection requires its own circuit, which means more costly network topologies. Further, it takes time to set up a circuit, which means longer connection times. Key differences between telephone and computer networks are summarized below:

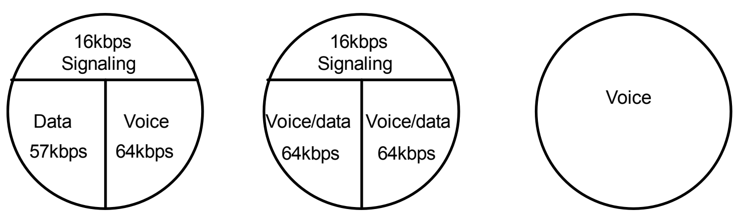
Telephone Network	Computer Network
Circuit-switched	Packet-switched
Fixed capacity channel with constant bit rate	Variable bandwidth requirement for variable bit rate
Always two-way	One or two-way
Tolerant of noise	Must be error-free
Continuous transmit and	Bursty communications

listen	
Always real-time	Can be store-and-forward
Long connection setup times	Quick or no connection setup time

It is important to note that by the end of the decade, these distinctions will be less important, as the world moves to isochronous networks like ATM that can support asynchronous or synchronous data equally well and with very high bandwidth.

The telephone network is conceptually quite simple. It uses a star network topology where each telephone is connected to a switch, which in turn is connected to other telephones and switches. Network interaction is achieved through messages between telephones and switches. A phone sends messages to the switch like the fact it has gone on or off-hook, or a button on the phone has been pressed. The switch acknowledges these messages and also sends call progress signals like dial tone, ringing or busy. In the case of simple analog phones, these signals are done "in-band" and audible to the user over the single channel between the telephone and the switch.

More sophisticated telephone systems support multiple discrete channels to the telephone device over a digital line. These channels fall into two categories: bearer (B channels) and signaling (D channels). Bearer channels are typically 64 kilobits/second, offering significantly greater bandwidth than the 4 kilohertz bandwidth of standard analog phones. Signaling channels are converging to 16 kilobits/second and carry messages about button presses and call progress tones between the phone and the switch. Signaling channels do not carry the actual tones, just messages that represent them. Thus, when you call a busy number, the switch sends a message to the phone that the called number is busy and the phone generates the busy tone you hear. More sophisticated features like the name of the calling party are also implemented over this channel as messages. The basic level of service for sophisticated phone systems is two bearer channels and a signaling channel, commonly known as 2B+D.   Some examples of the "pipe" between the switch and the phone are displayed below:



PBX (Microsoft)   ISDN (BRI)   Consumer phone

There are many different and incompatible line protocols that deliver 2B+D. These protocols cover the low-level electrical standards up to the actual messages exchanged between phone and switch. The result has been you need a proprietary telephone handset to connect to a given switch via a digital line, and proprietary telephones are quite expensive. ISDN is particular 2B+D protocol intended for the public telephone network. To date, implementations of ISDN from different manufacturers have not been interoperable. The telecommunications industry has finally gotten its act together and specified an interoperable ISDN standard (at least for the US with National ISDN-1; ISDN standards for Europe and Japan are still incompatible).

Telephone switches today are digital. That is, the traffic within a switch is handled digitally. However, a digital switch can support a variety of analog and digital lines. Each line enters the switch through a *line interface card*. This card essentially translates from whatever protocol the line uses to the internal switch protocol. A typical PBX might support both analog and digital trunks to the outside world, as well as analog (for a fax machine, for instance) and digital lines for internal users. So a call might come in on an analog trunk, go through the switch and then be delivered to a phone digitally.

There are a variety of telephone network types. Ultimately, they all interconnect, but there are many different standards that connect individual phones. To summarize the most common and important:

- POTS ("Plain old telephone service") - this is standard analog telephone service. Signaling is done in-band and there is a very limited set of network features. This is the least common denominator in the world of telecommunications, but it is the basis of a global network.
- CLASS (Custom Local Access Services) - this is richer but still analog service with some additional features for managing incoming calls. It provides the basis for Caller ID and other services. To be effective, it will require richer user interfaces and displays on telephones. Today, these features are accessed with codes like *62 and have very low penetration. This signaling is also in-band. There will be widescale availability in the next couple years.
- PBX - this is an on-premise switch that handles calls within an organization and has trunks to the public telephone network. These are all digital at the switch today and support both analog lines and proprietary 2B+D digital lines.
- Key System - while once different from PBXs in terms of architecture, these can today be thought of as very small PBXs designed for small businesses. Many incorporate signaling with 1B+D or 2B+D lines. The principal differences from a PBX are in customer size and the feature set on the handset.
- CENTREX - essentially PBX functionality where the switching occurs at the phone company's central office as opposed to at an on-site PBX. These have been very popular in recent years due to aggressive marketing by the Regional Bell Operating Companies.
- ISDN - the standard for end-to-end digital lines on the public network. Long awaited, much debated, but adoption has been very slow because to date it has not been a true standard. The National ISDN-1 standard due to be implemented at the end of this year should result in a real ISDN standard.
- Cellular - while this is a wireless service, it is fundamentally a circuit-switched architecture, substituting slices of frequency spectrum for physical circuits.

3. Applications:

By integrating the PC and the telephone networks described above, we can enable a number of classes of applications:

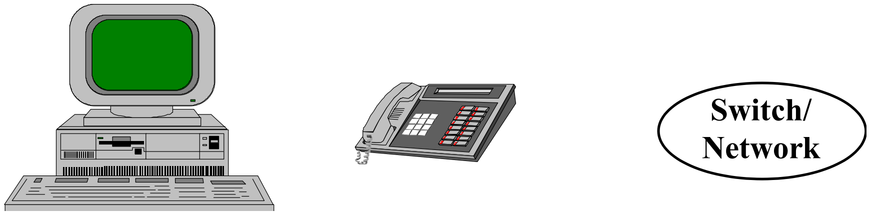
- Screen-based Telephony - visual interfaces permit much more usable access to existing phone features, as well as make new features possible that cannot be implemented due to the today's limited telephone user interface. Consider how difficult it is to use the slew of features found on office telephones. Richer user interface can make these features more accessible and easily customized. Setting up a conference call should be as easy as clicking on the names of the conferees.

- Communications Management - end-user programmability enables intelligent filtering of telephone communications. The user should be able to control who can reach them and where. Important calls should always get through, even if it requires trying the recipient at several locations.
- Personal Productivity - telephone calls should be automated and integrated into personal productivity software. Calls can be automatically dialed to save time and call details logged.
- Integrated Messaging - voice mail and fax are two common message types that use the phone system. Users should be able to access their different electronic communications media from a single point. Electronic mail, voice mail and faxes should all be accessible from a single in-box on the desktop. Similarly, they should be accessible remotely via a telephone or fax machine.
- Ubiquitous Voice on the Desktop - speakers and microphones for PCs are far from commonplace and pose a number of ergonomic problems. Yet nearly every PC has these devices adjacent in the form of a telephone. By providing access to the telephone's voice path to the PC, it is possible to digitize audio directly into the PC, or retrieve it seamlessly from a voice server and play back over a phone's speaker.
- Conferencing - wide-area connections between PCs can be used for videoconferences as well as less bandwidth intensive tasks like sharing documents or "virtual whiteboards", to create a richer communications medium and attain the benefits of proximity at a distance.
- Wide Area Data Networking - cleaner integration with the global telephone network facilitates fax and data communications from the PC. Today, for example, fax machines need a special analog line to be connected to a PBX.
- Vertical Solutions - telephone communications can be integrated with business information systems to yield tremendous value. Call centers that do lots of inbound or outbound calls already benefit from integrating their voice and data systems. An incoming call can be routed to the first available agent by a computer-based queuing system, and the customer's record "popped" onto the agent's screen before they even pick up the phone with Caller ID-like functionality.
- Unimagined Applications - as usual, we expect ingenious developers to find ways to exploit telephone services once they readily available to applications.

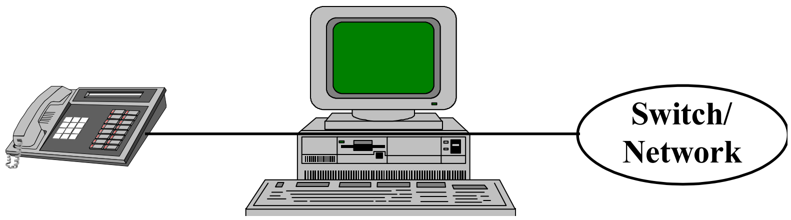
4. Connecting PCs and Telephones:

To enable the above applications, we need to integrate the PC with the telephone network. There are four basic models for connecting desktop PCs with telephones. The key is to give the PC access to the signaling channel. This may require special hardware, such as a DSP to recognize analog call progress tones, or just require software that can grab the messages coming over a digital D-channel. If the PC can send and receive the proper signals, it can emulate a telephone, and the switch can't distinguish between it and a standard phone.

The first model is a connection between the phone on the desktop and the PC. This is typically done with an RS-232 cable originating in the back of the phone or from an adjunct box. In a 2B+D environment, this solution only puts one channel onto the desktop, although most let you toggle between bearer channel carrying data or a signaling channel with escape sequences. The PDI box for the Microsoft internal phone system is an example of an adjunct box that terminates the line from the switch and then sends 1B or D channel to the PC via RS-232 and sends full 2B+D into the phone.



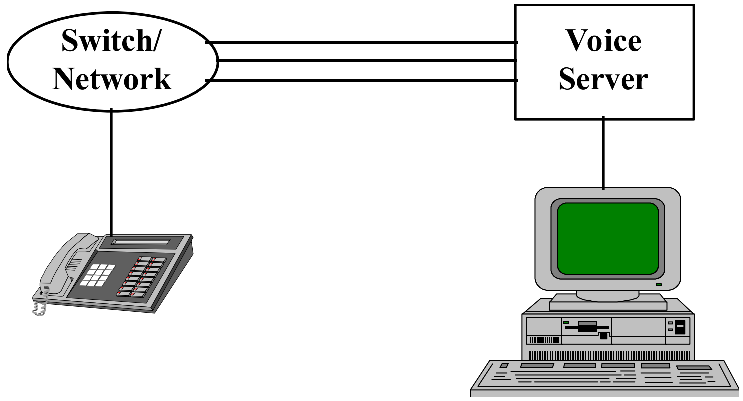
The second model involves a PC board that puts the full 2B+D onto the bus of the PC. The line from the switch terminates at the board and a telephone handset may hang off the board. This model provides full 2B+D access on the desktop and the primary functionality is in the PC. In the long term, this will be a connection on the motherboard, probably ISDN.



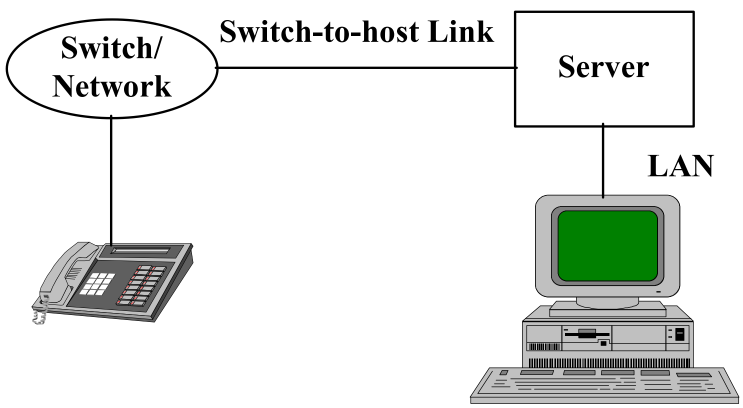
The third model is a voice server, illustrated below, which provides voice services under a client-server model. There is no direct connection between a particular PC and phone; there is only a virtual connection. A server has a number of voice processing boards in it and is connected by a number of telephone lines to the PBX. The server is connected with the PC over the LAN. This model can ensure universal access to voice services within an organization, without requiring special hardware at each desktop. The server model is also attractive for applications that store lots of voice, as it is very disk intensive.

The voice server model uses what is called *third party call control*. This differs from the previous two models, which use *first party call control*. First party call control is essentially the emulation of the telephone set. First party call control only permits you to do things you could do from your own telephone. Thus, there is no control of other lines. Third party call control, on the other, involves a direct connection to the switch and can control arbitrary lines.

Two examples will illustrate how the voice server works. Take the case of voice mail. A caller who gets transferred to voice mail ends up with their call connected to the voice server, which issues prompts and stores the message on the server. When the user goes to retrieve their mail, they can dial into the voice mail system from their phone and use tones to get their mail played back over the voice channel. Alternatively, a user might have a mail package that integrates both electronic mail and voice mail. By clicking on a voice mail message on their desktop, a message goes over the LAN to the voice server. The voice server could then ring that user's phone (or just turn on the speaker phone) and play back the message. Alternatively, the message could be sent over the LAN and played back on a PC with the appropriate audio hardware. Another example is enterprise-wide voice annotation. Any user can click on a document to begin an annotation. The annotation software (probably an OLE server) sends a message to the voice server, which rings the users' phone and they can speak their annotation. That annotation is stored on the server. When that document is forwarded to someone else, they can click on the annotation and the server will route that message to the reader's phone.



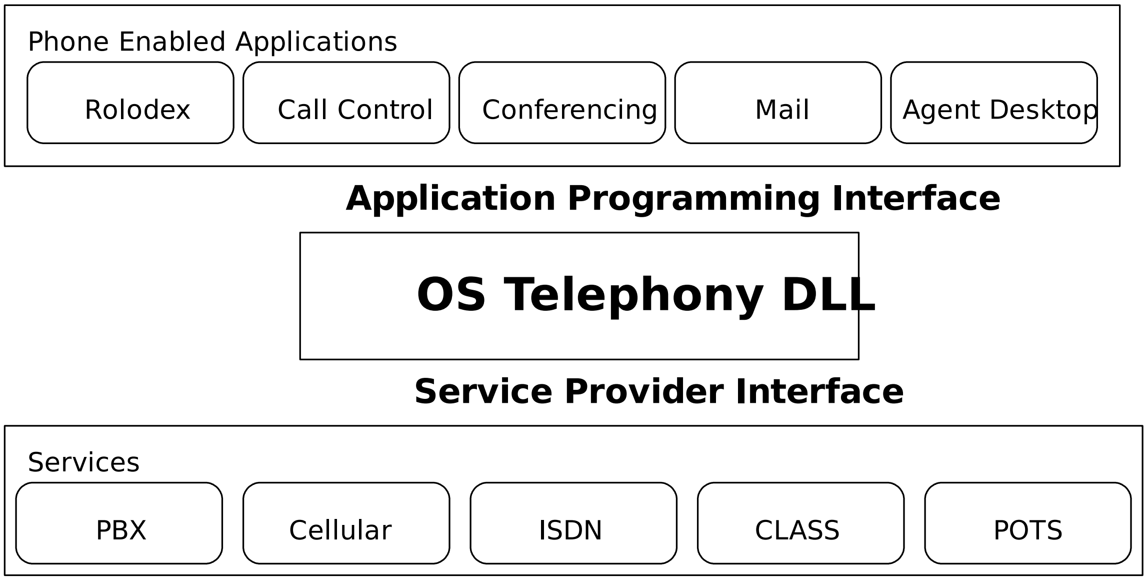
The fourth model, illustrated below, also works over the network, but instead of voice paths between the switch and the server, there is only a control channel. Thus, it can be used to set up calls and exploit basic network features, but it will not support applications like conferencing, voice annotation or anything else that uses the information carried on the voice path.



5. Windows Telephony:

Today, an application that wants to integrate the telephone must write "plumbing" code to talk to different combinations of hardware interface and network messaging for each incompatible telephone network. This is a daunting task today given the wide range of telephone networks, particularly for a simple application that only wants to use the telephone as a peripheral feature (e.g. a rolodex that wants to offer autodialing). The result is almost no telephony-enabled applications development.

Microsoft offers a solution to this applications bottleneck with the Windows operating system and a new industry standard for integrating PCs with telephones. Microsoft is developing a standard interface in consultation with independent software developers and telecommunications experts to help ensure complete system independence for telephony-enabled applications. This standard is called Windows Telephony and is a component of the Windows Open Services Architecture (WOSA). WOSA is a comprehensive design to hide programming complexities from users and developers, while providing seamless access across a variety of systems.



The preceding diagram illustrates how Windows Telephony will help solve the development problem that impedes telephony-enabled applications today. It provides a layer of functionality between applications and the underlying phone systems, allowing each to be developed independent of one another. An application only needs to write to the API, and not worry about what type of telephone network it is connected to or how that connection is handled. When the application wants to dial a number, it merely makes a single call and the particulars of dialing on the underlying network are handled at the service provider level. Similarly, a particular phone system needs only provide a single connection via the service provider interface and any application that uses Windows Telephony can utilize that phone system.

6. Conclusion:

Windows Telephony promises to unleash a new generation of applications that exploit the integration of PCs and the telephone network.

Overview of the Microsoft At Work™ Software Architecture

On June 9th, 1993, Microsoft Corporation announced the Microsoft At Work™ software architecture, a set of modular software technologies designed to bring ease of use, compatibility, and an enabling platform to devices in the workplace such as copiers, telephones, fax machines, printers, and handheld systems. These software components will be embedded inside these devices and in Microsoft® Windows™ based PCs to address many key problems that exist in the workplace today - devices are difficult to use, they don’t connect with Windows based PCs directly, and they cannot be easily programmed or customized to meet the needs of individuals and businesses.

Users will realize these benefits by using graphical user interfaces on devices such as copiers and telephones that make all features easy to access and use, and by having control of and connections to devices from Microsoft Windows based PCs. The Microsoft At Work software architecture offers full compatibility with the Microsoft Windows operating system, ensuring that data can move freely between Microsoft At Work based devices and the PC. This compatibility also allows the over 300,000 developers in the world today to use their existing development tools and knowledge to write applications that run on the Microsoft At Work platform.

Since the June 9 announcement, Microsoft has delivered the first Microsoft At Work software in Microsoft Windows™ for Workgroups v3.11, and multiple office equipment manufacturers are expected to ship devices based on Microsoft At Work software in 1994. Over 60 companies from the telecommunications, office automation, and personal computer industries are working with Microsoft to develop compatible hardware and software products.