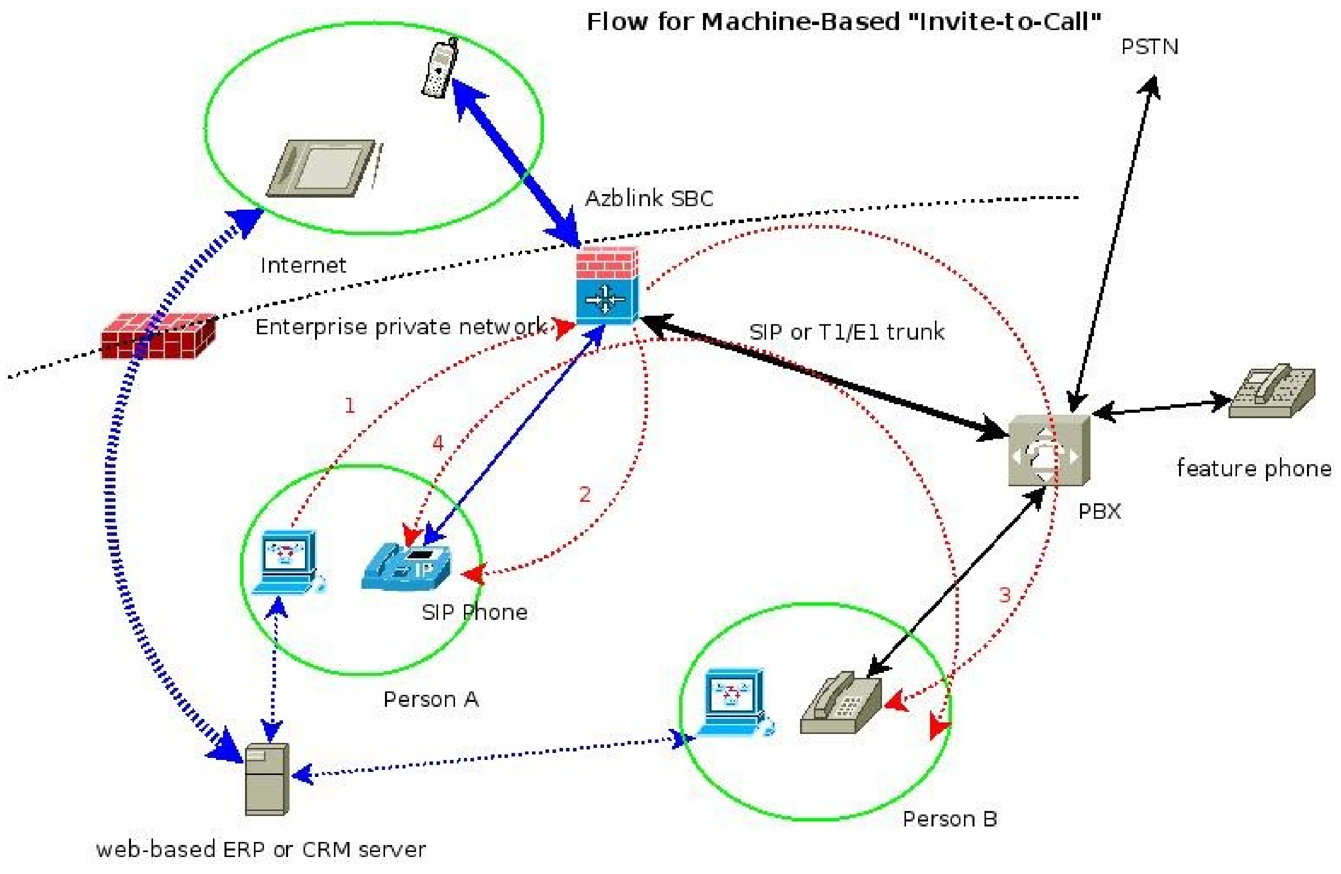
Azblink API for Machine-Based “Invite-to-Call”

Abstract: This document is to describe the method to trigger Azblink SBC for inviting two people to talk over the phone.

Introduction

The following diagram describes the use scenario: “**Person A**” is with his/her computer and a SIP phone on the desk.   Usually, the ERP or CRM server is accessed via the desktop computer. When “**Person A**” uses his/her ERP program to trigger Azblink SBC, the Azblink SBC will ring his/her desktop phone. Once **A** picks up his/her phone, Azblink SBC will ring the desktop phone of **B** and bridge the two calls together so that **A** and **B** can talk over the phone.


Flow for Machine-Based "Invite-to-Call"
PSTN
Azblink SBC
Internet
Enterprise private network
SIP or T1/E1 trunk
1
4
2
3
PBX
feature phone
SIP Phone
Person A
Person B
web-based ERP or CRM server

If “**Person A**” fails to pick up the phone at the first place, the system will try to ring the phone after 60 seconds for two times. And only when the first call is successfully picked up will the system ring the other party.

The scenario described does not have to be web-based ERP or CRM server; Azblink SBC takes input via HTTP GET or HTTP POST. As long as the application can send HTTP GET or HTTP POST, it does not have to be web-based. Please note: if the HTTP GET or POST request is sent from Web Browser, the route from the desktop computer to SBC should be set up in advance. Otherwise, the request can not reach Azblink SBC.

Operation Principle

Either to use HTTP GET or HTTP POST, the values of the two variables “origNum” and “destNum” should be set.

origNum: the_origin_number
destNum: the_destination_number

It depends where to place the PHP wrapper on Azblink SBC; you can send request from Web browser directly as follows:

http://192.168.11.23:8082/t/invite.php?origNum=0911222333&destNum=8886992

Then, the SBC will initiate a call from “0911222333” to “8886992”. Please note that the numbers entered here are with respect to SBC; they might be different from the numbers you are using directly from your PBX.

Sample Codes in Javascript

The following is an example to use Javascript in HTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.or
g/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<meta http-equiv="content-type" content="text/html; charset=UTF-8">

<head>
    <title> Invite two parties for phone conversation from Server</title>

<script language="javascript">
<!--
function httpGet(theUrl)
{
    var xmlHttp = null;

    xmlHttp = new XMLHttpRequest();
    xmlHttp.open( "GET", theUrl, false );
    xmlHttp.send( null );
    return xmlHttp.responseText;
}

//-->
</script>

</head>

<body>

    <li><a href="#" onClick=" httpGet('http://192.168.11.209:8082/t/invite.php?o
rigNum=0911222333&destNum=8886992'); "> Steve Chang (x6992) </a></li>
```

Sample Codes in C Programming Language for HTTP GET

```
/* --------------------------------------------------------------------
   - httpGet.c
      This program is an example to  send http "GET".
     Usage:
           ./httpGet hostip port  query_string
           where
                  hostip - IP address of the host
                  port   - port number to accept HTTP GET
                  query_string - the variables and the values you
                                 with to send via HTTP GET

     Example:
          ./httpGet 192.168.11.209 8082  "/t/invite.php?origNum=0911222333&destN
um=8886992"

------------------------------------------------------------------------- */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>

#include <arpa/inet.h>

////////////////////////////////
// get sockaddr, IPv4 or IPv6:
////////////////////////////////
void *get_in_addr(struct sockaddr *sa)
{
    if (sa->sa_family == AF_INET) {
        return &(((struct sockaddr_in*)sa)->sin_addr);
    }

    return &(((struct sockaddr_in6*)sa)->sin6_addr);
}



int main(int argc, char **argv )
{

    char host[128];
    char portno[128] ;

    struct hostent *server;
    struct addrinfo hints, *servinfo, *p ;
    int rv;
    int sockfd, bytes, sent, received, total;
    char message[4096];
    char response[32768];
    char page[1024];
    char postString[1024];
    char s[INET6_ADDRSTRLEN];

    if (argc < 4)
    {
       printf("%s hostip port  query_string  \n", argv[0]);
       printf(" Example: %s 192.168.11.209 8082  \"/t/invite.php?origNum=0911222
333&destNum=8886992\"\n", argv[0] );
       exit(0);
    }

     sprintf( host, "%s" , argv[1] );
     sprintf( portno, "%s" , argv[2] );
     sprintf( postString, "%s", argv[3] );

    /* fill in the parameters */
    sprintf(message,"GET %s HTTP/1.0\r\nHost: %s\r\n\r\n\r\n", postString, host)
;




    memset(&hints, 0, sizeof hints);
    hints.ai_family = AF_UNSPEC;
    hints.ai_socktype = SOCK_STREAM;

    if ((rv = getaddrinfo(host, portno, &hints, &servinfo)) != 0)
    {
        fprintf(stderr, "getaddrinfo: %s\n", gai_strerror(rv));
        return 1;
    }

    for(p = servinfo; p != NULL; p = p->ai_next)
    {
        sockfd = socket(p->ai_family, p->ai_socktype, p->ai_protocol);
        if ( sockfd ==  -1)
        {
            perror("client: socket");
            continue;
        }

        if (connect(sockfd, p->ai_addr, p->ai_addrlen) == -1)
        {
            close(sockfd);
            perror("client: connect");
            continue;
        }

        break;

   }

    if (p == NULL)
    {
        fprintf(stderr, "client: failed to connect\n");
        return 2;
    }

    inet_ntop(p->ai_family, get_in_addr((struct sockaddr *)p->ai_addr),
            s, sizeof s);

    freeaddrinfo(servinfo);


    /* send the request */
    total = strlen(message);
        bytes = write(sockfd,message,total);
        if (bytes < 0)
        {
                  perror("error: socket write");
                  exit(0);
        }


    /* receive the response */
    memset(response,0,sizeof(response));


    bytes = read(sockfd,response,4096);
        response[bytes]='\0';

    close(sockfd);

    /* process response */
    //printf("------------------ \n");
    //printf("Response:\n");
    //printf("bytes received: %d \n", bytes );
    //printf("%s\n",response);

     return 0;
}
```

Sample Codes in C Progarmming Language for HTTP POST

```
/* ---------------------------------------------------------------------
     This program is used to send http "POST".
     Usage:
            ./httpPost  hostip port action post_string
            where
                   hostip - IP address of the host
                   port   - port number to accept HTTP POST
                   action - the action script to take HTTP POST
                   post_string - the variables and the values you
                                   with to send via HTTP POST

     Example:
          ./httpPost  192.168.11.209 8082  /t/invite.php "origNum=0911222333&des
tNum=8886992"




-------------------------------------------------------------------------- */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>

#include <arpa/inet.h>

////////////////////////////////
// get sockaddr, IPv4 or IPv6:
////////////////////////////////
void *get_in_addr(struct sockaddr *sa)
{
    if (sa->sa_family == AF_INET) {
        return &(((struct sockaddr_in*)sa)->sin_addr);
    }

    return &(((struct sockaddr_in6*)sa)->sin6_addr);
}



int main(int argc, char **argv )
{

    char host[128];
    char portno[128] ;

    struct hostent *server;
    struct addrinfo hints, *servinfo, *p ;
    int rv;
    int sockfd, bytes, sent, received, total;
    char message[4096];
    char response[32768];
    char page[1024];
    char postString[1024];
    char s[INET6_ADDRSTRLEN];

    if (argc < 5)
    {
       printf("%s hostip port action post_string  \n", argv[0]);
       printf(" Example: %s 192.168.11.209 8082  /t/invite.php \"origNum=0911222
333&destNum=8886992\"\n", argv[0] );
       exit(0);
    }

     sprintf( host, "%s" , argv[1] );
     sprintf( portno, "%s" , argv[2] );
     sprintf( page, "%s", argv[3] );
     sprintf( postString, "%s", argv[4] );

    /* fill in the parameters */
    sprintf(message,"POST %s HTTP/1.0\r\nHost: %s\r\nContent-type: application/x
-www-form-urlencoded\r\nContent-length: %d\r\n\r\n%s\r\n", page, host, strlen(po
stString), postString);




    memset(&hints, 0, sizeof hints);
    hints.ai_family = AF_UNSPEC;
    hints.ai_socktype = SOCK_STREAM;

    if ((rv = getaddrinfo(host, portno, &hints, &servinfo)) != 0)
    {
        fprintf(stderr, "getaddrinfo: %s\n", gai_strerror(rv));
        return 1;
    }

    for(p = servinfo; p != NULL; p = p->ai_next)
    {
        sockfd = socket(p->ai_family, p->ai_socktype, p->ai_protocol);
        if ( sockfd ==  -1)
        {
            perror("client: socket");
            continue;
        }

        if (connect(sockfd, p->ai_addr, p->ai_addrlen) == -1)
        {
            close(sockfd);
            perror("client: connect");
            continue;
        }

        break;

   }

    if (p == NULL)
    {
        fprintf(stderr, "client: failed to connect\n");
        return 2;
    }

    inet_ntop(p->ai_family, get_in_addr((struct sockaddr *)p->ai_addr),
            s, sizeof s);

    freeaddrinfo(servinfo);
    /* send the request */
    total = strlen(message);
        bytes = write(sockfd,message,total);
        if (bytes < 0)
        {
                  perror("error: socket write");
                  exit(0);
        }


    /* receive the response */
    memset(response,0,sizeof(response));


    bytes = read(sockfd,response,4096);
        response[bytes]='\0';

    close(sockfd);

    /* process response */
    //printf("------------------ \n");
    //printf("Response:\n");
    //printf("bytes received %d \n", bytes );
    //printf("%s\n",response);

     return 0;
}
```