scannex

ip.buffer App Note

AN006 : Cisco Call Manager

Date	*Author*	*Release*
2008-06-26	MP	Initial draft

Scannex Electronics Ltd, UK
t: +44(0)8707 48 65 65
f: +44(0)8707 48 67 67

http://www.scannex.co.uk
info@scannex.co.uk

Scannex LLC, USA
t: 1-866-4BUFFER
(1-866-428-3337)

http://www.scannex.com
info@scannex.com

Table of Contents

1.Introduction..1

2.FTP Push...2

2.1.Could I just tell the CCM to push to my PC?..2

2.2.Can the CCM FTP across the Internet?...2

2.2.1.Is it safe?...2

2.2.2.Is it reliable?...2

2.3.Can I get real time CDR output?..3

2.4.If I have a slow link, can I regulate the flow of CDR data for a low bandwidth connection?.........3

2.5.What if the CCM disconnects, powers off, or fails?.................................3

2.6.Can I collect from CCME as well?...3

3.Other considerations..4

3.1.When providing connection from CCM and other PBXs.................................4

3.2.Applying European Privacy Laws...4

3.3.Using “pro-active” techniques...4

4.Setting up the ip.buffer..5

1. Introduction

The Cisco Call Manager (CCM) version 5 and above provide Call Detail Record (CDR) output via an FTP push operation.

This document discusses some of the issues surrounding this type of transfer and its use in the real world.

2. FTP Push

The CCM can be programmed to push the CDR data to an FTP Server. The CCM supports up to 3 FTP servers, and can push the data at an interval of between 1 minute and 1440 minutes (24 hrs).

2.1. Could I just tell the CCM to push to my PC?

Yes, that is technically possible. Since the FTP protocol is a standard Internet protocol, you can of course tell the CCM to connect straight to a PC and deliver the data there.

If the PC is offline, then the CCM should buffer the data. The CCM should keep retrying the FTP connection and eventually will deliver the data.

The ip.buffer supports the CCM. The CCM can FTP straight into the ip.buffer. In this case it is preferable to site the ip.buffer directly by the CCM – minimising possible points of failure.

2.2. Can the CCM FTP across the Internet?

Yes, that is possible if there is a route from the CCM onto the Internet – say across an ADSL link.

However, there are a few serious considerations:

2.2.1. Is it safe?

As long as an encrypted FTP transfer is used – then yes, you could consider this safe. However, the firewalls and routers between the CCM and the remote PC should also be carefully considered.

The ip.buffer provides optional high-strength, industry standard SSL encryption. Rather than just providing FTP encryption, SSL can protect HTTP (web), SMTP (email), and TCP traffic back to the central PC.

As an alternative, the ip.buffer can be programmed to deliver the data through a PPP modem dial-up connection directly into your central PC system – skipping the Internet altogether.

2.2.2. Is it reliable?

Using the public Internet, which involves a vast infrastructure, introduces the real possibility of failure. What happens when the Internet, or the ADSL connection, goes down? Will the system recover correctly? How long will it take before the system recognises this failure?

The ip.buffer can deliver the collected data either across the LAN/WAN/Internet, or across a PPP modem-dial up connection. In the scenario where the ADSL link dies the ip.buffer can be programmed to take an alternative route to your main PC by dialling up to a local ISP.

When the link fails, the ip.buffer knows within 2 minutes and can signal such a failure by sending out an alert email and/or an SNMP trap to the local network.

2.3. Can I get real time CDR output?

The CCM does not provide a real-time output. However, it can push CDR every minute – providing "near" real-time output.

The FTP protocol does have a reasonable overhead as it negotiates the transfer of the two Cisco files every minute. Pushing this data across your LAN, WAN, or the Internet may be an issue.

If the ip.buffer is sited locally to the CCM, the ip.buffer can offload some of the processing. For example, real-time may be required for specific needs – perhaps looking for a certain type of call. The ip.buffer can be programmed to delivery infrequently, but to look for these calling patterns and deliver immediately if it sees the pattern.

2.4. If I have a slow link, can I regulate the flow of CDR data for a low bandwidth connection?

You cannot directly program the CCM to limit the bandwidth from the CCM to the FTP Server.

With the ip.buffer you can specify the maximum data transfer rate from the ip.buffer across the Ethernet link (whether LAN, WAN, or Internet) and take careful administrative control over the bandwidth used by the CDR logging system (without having to program this in the network's switches and routers).

Additionally, you can choose to zlib compress the data that is output from the ip.buffer. This will typically give around 10:1 compression ratio on CDR data – saving 90% of the network bandwidth!

2.5. What if the CCM disconnects, powers off, or fails?

Early versions of CCM had an issue where the FTP connection would appear to remain "alive", but the Cisco would not push data. When using standard PC FTP server packages this may not be apparent until a user intervenes.

The ip.buffer has specific code for the CCM to detect this "stale" situation. If detected, it will automatically close the connection and allow the Cisco to form another.

All connects and disconnects can be monitored, either locally or remotely, by using the email alert system of the ip.buffer and/or the SNMP trap output of the buffer.

2.6. Can I collect from CCME as well?

The Cisco Call Manager Express uses the "syslog" protocol for CDR delivery. As this protocol uses the UDP/IP network protocol you cannot reliably use this across a WAN or the Internet – CDR records can easily be lost with no way of knowing.

A locally connected ip.buffer can easily collect the syslog packets from a CCME. Since the ip.buffer should be directly next to the PBX there is practically no chance of losing records. The ip.buffer can then deliver the CDR data by a reliable protocol back to the central system.

3. Other considerations

3.1. When providing connection from CCM and other PBXs

If there is a situation where there are CCM PBXs, as well as other PBXs that do not FTP the data you have to provide two different solutions for collecting the data and bringing it back to the central system.

With the ip.buffer you can provide a single central system and interface all types of PBX back to the central system. You can choose to use whatever delivery method is most appropriate, whether Email, FTP Push, FTP Server, or raw TCP delivery.

In this situation you only have one software system to test, maintain, and debug – the ip.buffer unifies the collection.

3.2. Applying European Privacy Laws

Some EU countries require that telephone CDR records be partially obscured. For example, they may require that the last 4 or 5 digits of the telephone number be masked – as a legal requirement. (These type of legal requirements may become more wide-spread.)

The ip.buffer can easily collect the data from the CCM and apply these rules to the CDR before it is actually stored. Then the data that is delivered from the ip.buffer (either across LAN, WAN, or Internet) is “safe” from a legal and privacy point of view.

3.3. Using “pro-active” techniques.

The ip.buffer has a large number of “pro-active” facilities that can be used to greatly assist in the detection of fault or alarm situations with the PBX.

For example, the ip.buffer can:

- Quickly detect that the PBX has disconnected and send out an email or trap.
- Send an email alert or trap if the CCM has *not* sent data within a predetermined time.
- Analyse the incoming data stream and send emails, traps, or trigger data deliveries.
- Perform local temperature measurement and email alerts when critical temperatures are reached (ip.2 and ip.4 only)

4. Setting up the ip.buffer

- Decide what ip.buffer channel to collect the CCM data on.
- Configure that channel's source parameters:
    - Set the collection mode to “FTP Server”
    - In the FTP Server parameters, make a note of the “Username” and “Password”. Enter these values in the CCM configuration screen.
    - Set the “Timeout” value to some value larger than the interval the CCM is programmed to FTP push to the ip.buffer. If this timeout is exceeded, the ip.buffer will kick the FTP connection.
    - Choose whether you want “File Markers”. The ip.buffer stores everything in a single linear stream. The file markers can be added so you can determine the original FTP filename, and the source of the transfer (ie the IP address of the CCM).
    - Set the “Protocol” parameter to “ASCII Lines”
- Configure the delivery method as appropriate to your overall system.
- Once saved, make sure the CCM is programmed to FTP to the address of the ip.buffer.
    - The status screen of the ip.buffer should show the “Source” as Connected, and the cell should be green all the time the CCM is connected.
    - To check the data, use the ip.buffer menu “Tools” and “Live Record View”[1]
    - Hovering the mouse over the “Source” cell on the status screen will show when data was last received (but not the content).

[1] When using File Markers for the FTP collection, you will typically just see the last file marker line after a transfer has completed.