Front-End Communications Processors

Their Place in an IP world

Angus Telfer

angus_telfer@inetco.com

February 15, 2002

INETCO Systems Limited
201 – 3773 Still Creek Ave.
Burnaby, B.C.
Canada, V5C 4E2

www.inetco.com

Telephone: (604) 451-1567
Facsimile: (604) 451-1565

Introduction

Communications Front End Processors (FEPs) are responsible for linking client applications and their associated networks to host computer based applications. With the advent of the Internet and of IP as a universal protocol, it is often assumed that there is no longer any need for FEPs as "everything is IP". This may well be true where FEPs provide only straight connectivity (and assuming IP never changes). However, FEPs also perform a number of other vital communications related functions that are closely linked to transaction applications including message and transaction switching, multiplexing, transaction security, QoS guarantors, and end-to-end transaction management and reporting. The need for these functions is especially important in mission critical transaction environments such as point-of-sale, security, and health care applications. In these environments, FEP functionality is more necessary than ever before.

This paper examines the functions performed by Front End Processors (FEPs) in relation to IP network environments. It also outlines different approaches that may be used to implement this functionality and discusses the advantages and disadvantages of each.

FEP Functionality

FEP functionality is primarily at the communications level. In terms of the Open System Interconnection (OSI) model this means the transport, network, protocol, link and physical layers. Functionality provided can be broken down into the following categories:

- Connectivity - involves the connection of networks, devices, and servers.
- Switching - involves the switching of packets and messages based on network addresses and message contents.
- Multiplexing - involves the multiplexing and de-multiplexing of data streams such as the aggregation of transactions in retail Point-of-Sale (POS) applications.
- Reliability - involves issues such as end-to-end delivery confirmation, maintaining a necessary quality of service (QoS), load sharing, and appropriately pacing transactions so as to not overload any device or server.
- Security - involves data security issues such as encryption, firewall type issues such as examination data address information before routing packets, and device security issues such as ensuring that there is no way to penetrate and take control of a device remotely.
- Network management - involves network monitoring, and network control.

- Reporting - involves the reporting of network and communications activities for analysis. These reports may be used for customer billing, network planning, or for proactively finding and eliminating network problems.

Each of these categories, and how it pertains to the IP world, is explained below.

Connectivity

A traditional use of FEPs is to act as a multiple protocol gateway for connecting diverse equipment. This is how it got the name "FEP" in the first place. It "front ended" the mainframe computer.

At first glance, this function is no longer needed in an IP environment as all devices appear to be converging to a single communications infrastructure. However, things are not that simple in reality. New IP based protocols such as WAP and SMS are required to support wireless "always-on" applications. Other new IP protocols such as TTCP and QTP are required to properly support transaction based services. Finally, even existing protocols, such as IP, HTTP, etc. are being evolved to meet increased requirements for security, addressing, reliability, priority traffic, etc. The result is that the IP infrastructure is steadily moving to having as many (or more) different protocol implementations than there were in the "bad old days" where each equipment manufacturer supported their own suite of protocols.

Integration of these new IP protocols and protocol variants over time could presumably be done by upgrading all equipment simultaneously. However, this is not practical. This is especially true in these days of network appliances where each appliance performs specific, well-defined functions. Even if it were possible within a particular enterprise, connection to the network outside the enterprise would necessitate a multiple protocol gateway of some sort.

Switching

Another important traditional use of FEPs involves the switching of packets and messages based on network addresses. In modern implementations, this may also be based on message contents. One message may be sent to one server while another message may be sent to another server based on such items as the "terminal identifier" or "bank identifier" field within the message.

With the Internet, switching is accomplished at the client (i.e. PC) end of the network. Control of all communications, including making simultaneous connections to several destinations, is clearly left with the client (i.e. Internet browser, email, etc.). While this approach works quite well in client centric systems, it is potentially disastrous architecture for mission critical applications where the enterprise must maintain constant control over the device. With a client centric architecture, there is no reliable way to manage, monitor, or audit the device in real time as it may be communicating with

several remote servers simultaneously. As well, such an approach provides a large opening for hackers to steal control of the device.

The answer is to have a communications processor (i.e. an FEP) through which all the device's communications goes and through which it can be monitored, audited, and controlled. This device can then route the messages appropriately and can also be interrogated at any time with respect to the status of any device as well as being able to report any abnormal events that occur.

Multiplexing (aggregation)

One particular type of FEP that is used within many financial networks is a Network Access Controller (NAC). As well as the other functions described in this paper, a NAC is capable of concentrating multiple data streams onto a single data stream by acting as a gateway between circuit switched data and message switched data. This multiplexing (or aggregation) of data allows there to be fewer circuits on the host side of the NAC than there is on the device side. It also means that the host side circuits can be static as opposed to the device side where, in the transaction world, circuits are often transient and generally last only a few seconds each.

This function is less important in the IP world than it is in the legacy world where circuit management traditionally placed a large load on the mainframe. However, it is still important in large networks as TCP connections are still relatively expensive and the maximum number of connections into a server is generally much less than the number of concurrent transactions that could otherwise be handled by that server.

Reliability

Reliability involves issues such as end-to-end delivery confirmation, maintaining a necessary quality of service (QoS), load sharing and transaction pacing so as to not overload any device or server, and alternate routing in case of server failure.

Traditionally, the FEP would be responsible for these functions and would handle them as follows:

- Message delivery would be acknowledged end-to-end. That is, a message would not be acknowledged back to the application until confirmation was received from the remote device either by a direct acknowledgement or a delivery confirmation message.

- Quality of service information would be maintained on an end-to-end (i.e. client device to server application) basis by measuring the time before a confirmation was received, by noting retransmissions and other recovery actions and, if necessary, by alerting the network operator.

- Load sharing and transaction pacing was done by the FEP holding onto transactions and spreading them over available circuits (i.e. via rotary hunt groups, etc.) so as to not overload any device. Channels to the host could be "paced" by allowing transactions through at only a specified rate. Alternatively, channels could also be configured as half duplex (i.e. one-way only) to prevent the turn around delays encountered in many hosts.

- Failure recovery was done by the FEP transparently routing transactions to an alternate host should the primary host fail. Note that it is generally important in these scenarios that all transactions be routed to either one host or the other in order to maintain the operational integrity of the application and its database. If the client device controls the switching between hosts, it is invariable that some devices will be connected to the primary server and some (due to purely transient issues) will end up connected to the secondary server. At this point the enterprise is, whether it wants to or not, running a distributed database.

In pure Internet based IP networks, some of these functions are provided by the router network while many are not done at all. This results in a poor (and generally unknown) quality of service. In addition, the host must be thoroughly tested in advance to ensure that it can always take the simultaneous, worst case transaction hit as there is nothing in the network to spread load or to pace the rate of incoming transactions.

Security

Security in IP networks is primarily accomplished by the use of firewalls and VPNs. These approaches provide a level of security, but does not eliminate the possibility of insider attacks on the network. For example, a client device could be being taken over by hackers or an attack could be mounted on the server.

Traditional FEPs do not provide much in this area other than the examination of addressing information before packets are sent on. However, FEP like functionality is increasingly needed that would allow the client to be blind to all incoming requests except those from the FEP. While this is not strictly in tune with the concept that everything should be SNMP manageable, should respond to pings, should allow downloads by FTP, etc., it is much more secure than the approaches used in most current IP networks.

Another area where FEP like functionality is required is for filtering out messages that do not conform strictly with expected transaction types before they get to the host. Firewalls do this to some extent currently. As mission critical applications expand on IP networks, it will become increasingly necessary to expand this capability to the actual transaction message itself.

Network Management

One of the areas in which traditional FEPs excel is in their ability to monitor the health of all network elements, including that of the client devices themselves. The FEP is in the middle of every message and so always knows the state of the device without the need to poll it.

In most current Internet based IP implementations there is no central network entity. Instead, network management functions are generally placed off to the side where an SNMP server such as HP OpenView polls devices, receives alarms, and notifies the operator as it deems appropriate. Putting network management off to the side like this is, at best, a very poor substitute for providing in-line network management. It eliminates the possibility of using network management as part of a dynamic feedback system where corrective action can be taken automatically in real time. It also increases the bandwidth required due to the management requests that must be made over the network.

Reporting

Reporting on communications activities is important as it generally affects core business operations such as network planning, billing for network use, auditing (for security), and the interception of data as required by some governments for digital wire tapping.

Centrally located FEPs are an ideal place to collect the data required for such reports with minimal network load or changes to server applications. The alternatives are to collect the information from the devices themselves (impractical for large networks and for places where the devices are owned by third parties) or to collect the information from the server (impractical where multiple server applications are involved or where the servers are owned by third parties).

FEP Implementations

As can be seen, FEP functionality is very important in IP networks used for mission critical transaction processing. As our dependency on real time transaction systems grows, it will become increasingly vital. Given this, how can such functionality be achieved? Following are some of the methods that have been used.

Client as FEP

One way that has been promoted is to move the responsibility to the client as is done with web browsing over the Internet. In this model, the client, such as an Automatic Teller Machine, connects to each server application directly. That is, it connects to one server for financial transactions, another server for device management, another server for network management, another server for providing local advertising, and may even connect to other servers for such things as requesting cash replenishments, etc. In case of network or server failure, the client simply finds another server to connect to independently of any central control.

The primary advantages of this solution are in favor of the client manufacturers. It is simple to implement and it moves the control of the device from the central server application to the client. This reduces the reliance of the client manufacturers on the producers of the central servers and allows them to make their client application more proprietary.

Key disadvantages of this approach are:

- The client is operating essentially independently of any central application. This makes it very difficult to control or monitor. If security breaches do occur, they can be difficult to detect and it can be very difficult to regain control in order to limit the damage or correct the problem. This lack of a central control or audit point can be disastrous in mission critical environments.

- As the clients operate independently, there is no good way for switching them from one server to another server in any synchronized manner. In normal operation, clients will tend towards being distributed over all available servers due to variable loading conditions on the network. This is not a problem if servers are identical and meant to be operated in this fashion. However, in the case of most enterprise servers there is the concept of primary and secondary servers (if for no other reason, then to provide a method of updating server software). Distribution over these servers without appropriate operator control can result in problems maintaining consistency within the database, with encryption keys, etc.

- As all clients are operating independently, they must all be individually managed. In large systems, this results in a requirement for extensive network management bandwidth for querying and otherwise managing all the devices. With a centralized FEP approach, this would be accomplished simply by management of the FEP itself.

- The need for multiple communications connections from the device using the client centric approach opens it up for attacks from outside. This is especially a problem as most client devices use either Microsoft Windows™ or other well known IP stacks that have been extensively "hacked".

Given these problems, why is a client approach to providing FEP functionality considered at all? The answer is simple. It allows terminal manufacturers, be they manufacturers of client software, Automatic Teller Machines or terminal adapters, to quickly put a product out with minimal or no interaction with the producers of the central server application.

Server as FEP

The direct opposite to having the FEP functionality in the client, is to put it all in the server. This approach is similar to that used by Tandem in the legacy world.

The primary advantage of this embedded approach is that the server application is tightly coupled to the networking functionality, simplifying management and control as well as increasing security.

The primary disadvantage of this approach is that most current applications involve multiple disciplines, each with sophisticated software applications. For example, even a simple Point-of-Sale (POS) terminal may use biometrics for identifying the customer as well as for doing the financial transaction. There may also be loyalty card programs, checks regarding card usage for security reasons, inventory information regarding what is purchased, management of the POS terminal itself and, of course, management of the network links. Putting all of this in one server is, realistically, impossible. Even if achievable at some point in time, it forces enterprises to accept inferior (or no) solutions in one area in order to get superior solutions in another area. Adding new capabilities outside the server vendor's product range becomes expensive or impossible. Any attempt to move to another vendor in the future also becomes extremely expensive. Such lock-in to a single supplier is not in anyone's interests even if it is possible.

Network Based FEP

This is the approach generally promoted by the network equipment providers and telcos. With it, there would be a large network box at the central site that would control communications between the servers and the different network access points to which the clients would be connected.

When fully developed, this approach has the possibility of solving many of the communications problems inherent in using today's IP networks for mission critical applications. It can provide centralized network management, a centralized audit point, separation between the network and server protocol stacks, flow control of data into the servers, and server fail-over and load balancing.

The primary disadvantages of this approach are: (a) it is not yet fully implemented in terms of the features required for large enterprises, (b) it requires all network devices (including terminal adapters) to come from a single vendor if they are to be properly managed end-to-end, and (c) it is purely a network device with no knowledge of either the client or server application whatsoever.

Most of the disadvantages of this approach may be resolved over time should the manufacturers see sufficient revenue from such an integrated solution to warrant the work necessary and should they also open up the management interfaces to third party device manufacturers. However, current network boxes sold for this application are very limited and manage only the vendor's own equipment. This often leaves third party devices operating in the "client FEP" model, the worst of all models from a mission critical transaction application perspective.

Transaction FEP

In this approach, a box is inserted between the network and the application servers which knows of the different types of transactions that will pass through it. This box acts both as a network management and control point and also as a message switch.

The advantages of this approach from a network perspective are the same as for the "network based FEP". In short, it provides centralized network management, a centralized audit point, separation between the network and server protocol stacks, flow control of data into the servers, and server fail-over and load balancing. In addition, as it is transaction knowledgeable, it can also provide concentration of transactions on few communications channels to the servers, content based message switching, and even message conversion. Finally, as it is aware of the end devices, it can provide management on a true end-to-end basis and as well as providing guarantees of message delivery not otherwise possible.

The primary disadvantage of this type of solution is that transaction FEPs are very specific to different verticals. That is, a transaction FEP for POS terminals is of little use in security, Automatic Teller Machine, or other applications. For this reason, the spread of transaction FEPs has been slow and most have been created by system integrators for particular customers as opposed to being created as products for particular markets.

Summary

As can be seen, FEP functionality is vital within the IP world and will become even more important as the Internet evolves and as our commerce and personal interactions become increasingly reliant on IP technologies. Without FEPs, it will be difficult to guarantee the quality of service, the scalability, or the evolution of network standards and applications that will be necessary to provide true mission critical enterprise solutions.

The conventional approach for providing FEP functionality in the IP world is the most prone to problems as it is based on a client-centric architecture that places the client clearly in control of all communications. Likewise, the approach of putting the functionality in the server application is not advisable as it is based on a mainframe architecture whereby all services and applications on the mainframe were purchased from a single supplier. A network-based architecture whereby the network is managed as a single point is better, but suffers from the weaknesses of being closed to third party vendors, being incomplete, and being ignorant of the transactions that pass over it. Transaction-based FEP architectures are clearly preferable to the other approaches. However, they are not always available as this field is just emerging..

The functionality that FEPs provide may come in a variety of guises under a variety of names (none of which are "FEP"). It is up to the purchaser to do the research and weigh the pros and cons of each approach before deciding the best approach for their mission critical application at this point in time. This is a new era that is evolving quickly.