The HP OpenCall SS7 Platform

The HP OpenCall SS7 platform allows users to build computer-based signaling applications connected to the SS7 signaling network.

by Denis Pierrot and Jean-Pierre Allegre

Today's telecommunications operators need to offer more and more services to their customers. Because of deregulation and the resulting competition, network operators have to be able to bring to the market useful value-added services to differentiate themselves from the competition. To support new functionalities, telecommunications networks have undergone an important restructuring starting in the mid-1980s. This restructuring resulted in the separation of the signaling functions from the voice transmission functions.

Signaling includes all of the necessary procedures to set up, tear down, and control calls. Before this split was made, the networks were using inband signaling—the signaling information was conveyed over the same channel as the voice with some predefined tones (see Fig. 1a). This technique had many drawbacks, including:

- Long call setup times. Addressing information needed to be outpulsed one digit at a time for each intermediate switch in the voice path.
- Security problems. Billing fraud was possible by faking the inband signaling and billing tones.
- Limitations on the amount of new services that could be provided.

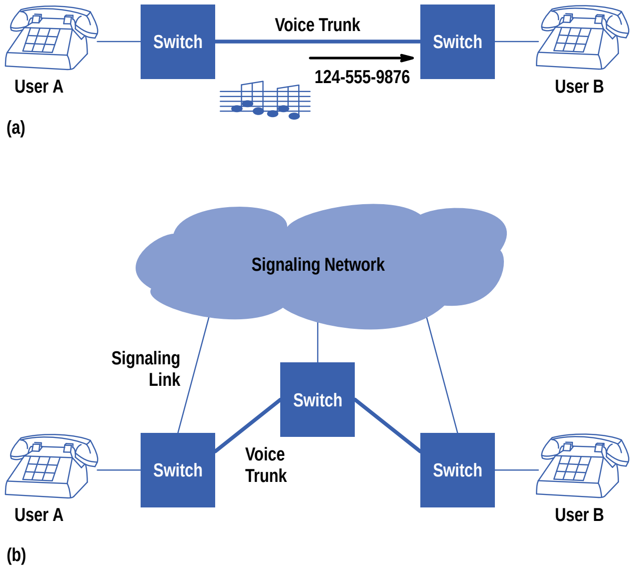

Fig. 1. *(a) Before signaling was separated from voice transmission, networks used inband signaling—the signaling information was conveyed over the same channel as the voice. (b) After the split, all of the connection setup, teardown, and control is effected via the signaling network and the voice trunks are dedicated to transporting voice only.*

The Signaling Network

With the separation of signaling and voice transmission, the concept of the *signaling network* was introduced. The signaling network is a digital, robust, packet network with built-in redundancy to achieve a high degree of availability. Fig. 1b shows the typical topology. All of the connection setup, teardown, and control is effected via the signaling network and the voice trunks are dedicated to transporting voice only.

The creation of the signaling network, often called *common channel signaling* or CCS, makes it possible to implement an important set of new services because of the global control it provides over the transmission network. The current implementation of the signaling network is called Signaling System #7, or SS7.

The signaling network is the foundation for the *intelligent network*, which makes it possible to deliver new services to network operators' customers in a timely and cost-effective manner. The intelligent network is programmable so that new services can be easily provisioned. It uses vendor independent interfaces so that multivendor networks can be built. It allows rapid introduction of new services, and it distributes the intelligence in the network into a few intelligent elements. For more information on intelligent networks, refer to ***Article 6***.

Elements of the Signaling Network

The signaling network is a packet network built using the following elements:

- The *service switching point* is a switch that is able to interact with the signaling network.
- The *signaling transfer point* is a packet switch that routes messages between end points of the SS7 network. Signaling transfer points are often compared to IP routers. Signaling transfer points have no connections to voice trunks or telephone lines.
- The *service control point* is the place of execution of value-added services. Historically, service control points were seen as databases only. The Advanced Intelligent Network architecture describes them more as the place of execution of the service logic.
- *Signaling links* are the physical connection between elements of the SS7 network. They provide a full-duplex 64-kbit/s digital path conforming to the V.35, DS0A, or T1/E1 standards. A group of signaling links connecting the same two elements can be grouped logically into a *linkset*. The SS7 protocol provides procedures for redundancy and load sharing between links of the same linkset. For example, if a link within a linkset fails, the protocol will automatically move the traffic to the nonaffected link and will try to restore the failed link.

Fig. 2 shows typical elements of an SS7 network. The signaling transfer points are provisioned in pairs. Service control points and service switching points always connect to two different nodes. Similarly, links are redundant such that messages can always find an alternate route in case of a failure.

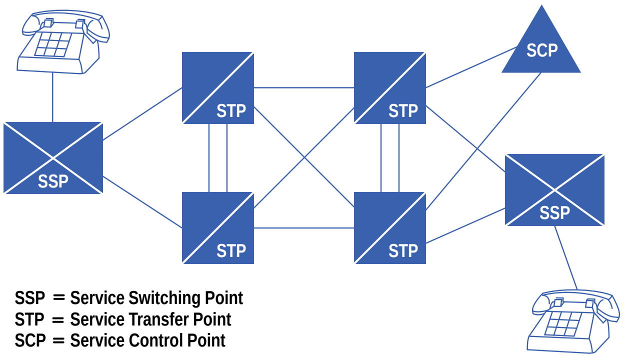
SCP
STP
STP
STP
STP
SSP
SSP
SSP = Service Switching Point
STP = Service Transfer Point
SCP = Service Control Point

Fig. 2. *Elements of the signaling network.*

Use of the Signaling Network

The signaling network is used for many purposes: It is used for regular calls, allowing rapid setup and secure operation. It is used in the wireline fixed network to provide additional services requiring specific service logic and databases (800 numbers, alternate billing, etc.). It is used in the mobile network to manage mobility information. For example, when a mobile phone is switched on, the *home location register* containing the subscriber profile is queried using the signaling network.

The signaling network is now the basic infrastructure for the global telecommunications network. SS7 networks are deployed in almost all countries now, with variable coverage. North American networks are defined by ANSI and Bellcore, while the rest of the world usually follows the ITU standard. The two flavors of the standard are similar, but (of course!) incompatible. The ITU version is used at the boundary of the international networks.

The SS7 Protocol Stack

The SS7 reference model is based on the Open Systems Interconnection (OSI) reference model of the International Organization for Standardization (ISO), following similar principles with layers of protocols. However, the SS7 model is more specialized, being designed for signaling information transfer with a specific focus on low latency and built-in robustness.

Fig. 3 shows the SS7 protocol stack. MTP stands for Message Transfer Part. It represents the three lower layers of the protocol stack. The Signaling Connection Control Part (SCCP) is built on top of MTP Level 3. The ISDN User Part (ISUP) sits on top of MTP (and potentially SCCP also). The Transaction Capabilities Application Part (TCAP) resides on top of SCCP. Let's look at each layer's functions.

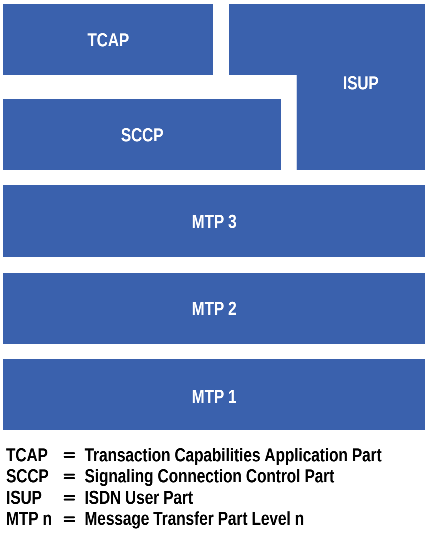
TCAP
ISUP
SCCP
MTP 3
MTP 2
MTP 1
TCAP = Transaction Capabilities Application Part
SCCP = Signaling Connection Control Part
ISUP = ISDN User Part
MTP n = Message Transfer Part Level n

Fig. 3. *The SS7 protocol stack.*

MTP Level 1. The physical layer of the SS7 protocol is based on digital transmission channels known as signaling links, connecting two digital elements with a rate of 56 kbits/s (ANSI) or 64 kbits/s (ITU). The physical network can be composed of V.35, DS0A, or T1/E1 links.

MTP Level 2. MTP Level 2 maps onto layer 2 of the OSI seven-layer model and provides a basic message exchange with an error correction mechanism based on the retransmission of unacknowledged messages. An alignment procedure ensures, if successful, that links are able to convey messages between two points.

Unlike most level 2 protocols, MTP 2 has some unusual features. For example, it keeps filling the available bandwidth by sending small messages (fill-in signaling units or FISU), especially when there is no user traffic. This allows it to promptly detect any physical link failure and to react accordingly. From an implementation point of view, this is a very stressful feature and usually requires specific hardware and firmware. Another interesting feature of MTP 2 is its ability to return unacknowledged frames stored in its buffers to the upper layer (MTP 3) in case of failure. This allows MTP 3 to retrieve the frames from a failed link and send them again on another link without any data loss.

MTP Level 3. MTP Level 3 handles the routing functions and network management procedures of the SS7 network. MTP 3 is the key contributor to the built-in robustness of the SS7 network. The network management functions are the most complex features of the SS7 protocol. They are in charge of maintaining the integrity of the signaling network. These functions can be split into three areas: link management, traffic management, and route management.

Link management is responsible for the integrity of one link. It uses services (especially counters) provided by MTP 2 to monitor the quality of a link. If the link is considered to be in error (excessive error rate, for example), then the link is removed from service, messages are rerouted to alternate links, and the adjacent node is notified to do the same. MTP 3 then starts an alignment procedure to try to restart the link in a clean state.

Traffic management handles traffic on the links within a linkset. It is in charge of load-sharing the traffic over all the active links and of rerouting traffic from a failed link to another.

Route management is in charge of maintaining information on the network topology and the availability or unavailability of certain paths to reach destination nodes. The interesting feature of MTP 3 is that it only knows adjacent nodes and destination nodes and not other intermediate nodes (Fig. 4). It maintains a table of available routes to reach a destination node via an adjacent node.

The other functions of MTP 3 are message routing, discrimination, and distribution. On the outbound path, this means finding the right link to reach the target destination node. On the inbound path, it has to determine for which higher-level protocol the packet is intended (ISUP, SCCP, etc.). It can reroute the packet via the network if it determines that it is not the intended target.

SCCP. SCCP is built on top of the MTP 3 layer and provides additional end-to-end services such as connectionless or connection-oriented service, extended addressing, and network management functions.

SCCP Users are assigned a specific address called a *subsystem number* which, along with the MTP 3 address (called a *point code*), makes it possible to uniquely address an SCCP user in the SS7 network. The extended addressing feature of SCCP allows the use of a label or *global title* in place of the subsystem number and point code to address an application. This allows for symbolic addressing and provides a level of indirection with respect to the physical structure of the SS7 network.

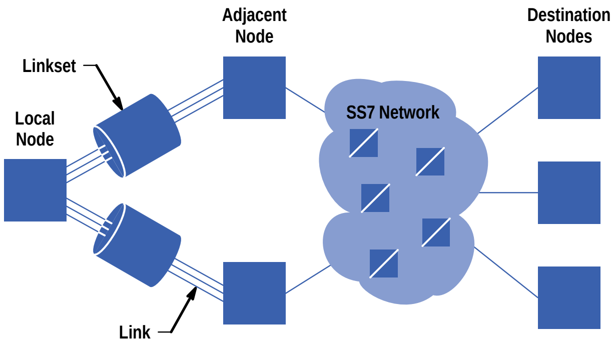

Fig. 4. *The SS7 network as seen by the Message Transfer Part Level 3 (MTP 3) on the local node.*

The translation of the global title to the subsystem number and point code is accomplished either in the signaling transfer points or in the endpoints. Very often, the dialed digits are used as the global title.

In connectionless mode, SCCP operates a bit like UDP (User Datagram Protocol) operates in the Internet world: messages are sent to a target address (either a global title or a subsystem number and point code) and are transmitted from node to node by MTP 3 to the final destination. There is no guarantee of delivery of the message, nor is it guaranteed that the messages will arrive in the order in which they were sent. The connectionless mode is the most widely used, especially because TCAP uses it.

In connection-oriented mode, SCCP operates a bit like X.25. A virtual circuit must first be opened before data transfer can take place. Once the circuit is open, there is a guarantee that messages are delivered and in the right order.

SCCP also has built-in network management functions. Each node in the network maintains the state of its SCCP users identified by their subsystem number. The SCCP layer is in charge of broadcasting the state of its own subsystem numbers to the other nodes, so that at any time, an SCCP user knows about the state of the remote subsystems.

TCAP. The objective of the Transaction Capabilities Application Part is to provide the means of transferring noncircuit-related information (unlike ISUP, which handles circuit-related information) between different nodes of the SS7 network. TCAP is especially used to access service control points in the fixed network or to access home location registers, a short message service center (SMSC), or an equipment identification center (EIC) in the mobile network.

The TCAP layer is divided into two sublayers. The first, the *transaction sublayer*, deals with the exchange of TCAP messages. A transaction, called a *dialogue* at the user level, can be *unstructured* (composed of one unidirectional TCAP message) when no explicit initiation or termination is needed. For more interaction, a *structured dialogue* is used with a beginning, an exchange, and a termination or an abortion. This sublayer uses the SCCP connectionless service.

The upper sublayer is the *component sublayer* and is dedicated to operations. An *operation* is an action (with parameters) to be performed by the remote end. Each operation is encoded into *components*, which are part of a TCAP message payload. Components convey either an operation request or an operation response. Simultaneous operations are allowed inside a transaction and TCAP is able to support multiple simultaneous transactions with different remote TCAPs. The addressing for each TCAP user is the addressing provided by SCCP (point code and subsystem number or global title).

Fig. 5 shows a TCAP interaction with the separation of the transaction layer and the component layer.

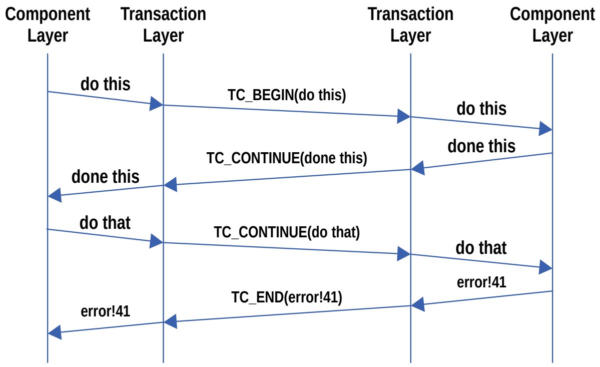

Fig. 5. *A TCAP (Transaction Capabilities Application Part) transaction, showing the transaction layer and the component layer. (ITU-T terminology is shown. ANSI has equivalent services).*

ISUP. The ISDN User Part (ISUP) is a circuit-related protocol, which means that it defines and transports the necessary messages to set up, tear down, and control voice and data circuits. It uses the MTP 3 services to transport messages from switches to switches.

A typical ISUP interaction is shown in Fig. 6. User A takes the receiver off the hook and dials 555-xxxx. The local switch (333) looks up its routing table and finds out that it should route the call to switch 444, which is an *access tandem* (not the final destination). It then sends an *initial address message* to switch 444 via the SS7 network and reserves a voice circuit to switch 444. When switch 444 receives the initial address message, it reserves the other end of the voice circuit, finds out that the call should be routed to switch 555, and sends another ISUP initial address message via SS7 to switch 555. Switch 555 accepts the call, reserves the voice circuit with switch 444, and sends back an *address complete message* to switch 444, which forwards it to switch 333, triggering the ring-back of user A (via the voice path). Switch 555 also rings the destination phone. When user B takes the receiver off the hook, switch 555 sends an *answer message* over the SS7 network to switch 444, which forwards it to switch 333. The call can now proceed.

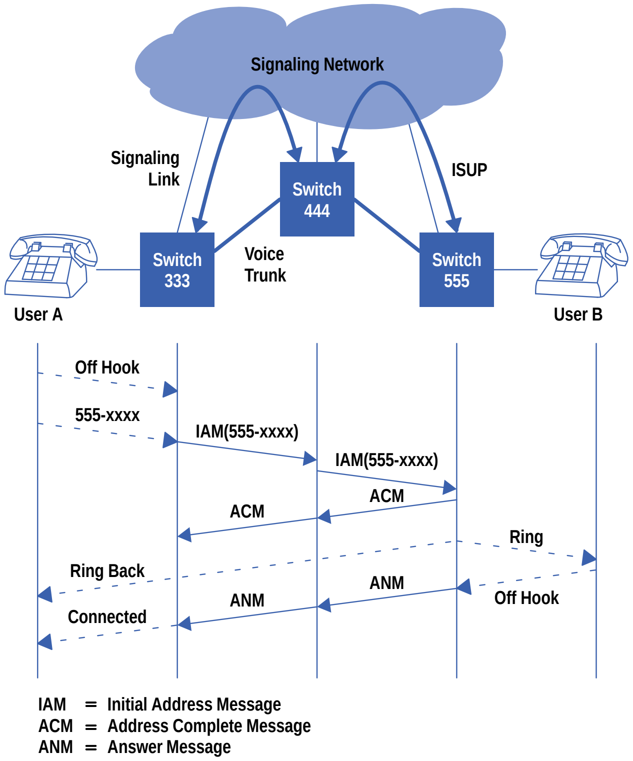

Fig. 6. *A typical ISUP (ISDN User Part) interaction.*

The release phase uses the same kind of message interaction. ISUP also allows many other supplementary services.

The HP OpenCall SS7 Platform

The HP OpenCall SS7 platform allows users to build computer-based signaling applications connected to the signaling network. Using computers to achieve some of the intelligent network functions is one of the key benefits of the intelligent network architecture. Compared to modifying switch software, it is less expensive, faster, and easier to program computers in the intelligent network. The HP OpenCall SS7 platform provides the hardware and the middleware necessary to use a computer in a signaling network.

The main characteristics of the HP OpenCall SS7 platform are:

- It provides the protocols to connect to the SS7 network. This mostly consists of specialized hardware (for MTP Levels 1 and 2) and protocol stacks (MTP, SCCP, TCAP, ISUP) for various flavors (ANSI, ITU, Chinese, hybrid).
- It provides high availability—most of the target applications are mission-critical (see ***Article 8***).
- It provides the necessary components for the computer to be integrated in a central office. This means, for example, support of a −48Vdc power supply, antiseismic capabilities, compliance with established standards, and so on.
- It provides open application programming interfaces (APIs) for users to write applications.

The HP OpenCall SS7 platform is a platform in the sense that it does not provide the application itself but rather allows users to build the application. The platform can be instantiated under several options that will be described later.

Core Protocol Implementation

The network connection is made by a dedicated communications unit called the *signaling interface unit* (see Fig. 7). Each signaling interface unit has a SCSI interface card for the host connection and three slots for signaling link interface cards. These cards provide two links each, with different options for each supported type of link (V.35, DS0A, and T1/E1). These cards come from the HP 37900 SS7 protocol analyzers. The signaling interface unit can also be expanded by means of a dedicated expansion box, which provides four additional slots (eight more links) for a single SCSI attachment. Signaling interface units can be chained on the SCSI bus and the platform can currently support up to 64 links.

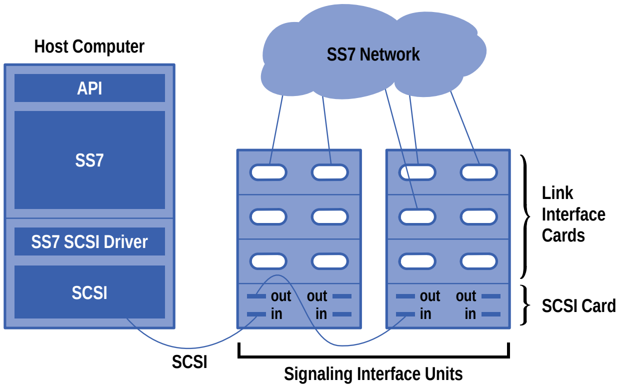

Fig. 7. *The network connection is made by a dedicated communications unit called the signaling interface unit. Each signaling interface unit has a SCSI interface card for the host connection and three slots for signaling link interface cards.*

Each signaling link interface card runs the MTP 1 and MTP 2 protocols and sends and receives messages to and from the host via the SCSI interface.

On the host side, messages are read by an SS7 driver built on top of the SCSI driver in the HP-UX* kernel. On top of these, a single user space process implements MTP 3, SCCP, and TCAP, sending and receiving messages to and from the SS7 driver.

APIs are provided as a library to be linked with the user process. The library is in charge of managing the interaction with the user application, implementing interprocess communication between the application and the SS7 stack, and supporting the flow control.

For each layer of higher-level protocols, an operation, administration, and maintenance (OA&M) programmatic access is provided (Fig. 8). This allows an application developer to control the state of the protocol stack or to implement management applications (monitoring, configuration, etc.).

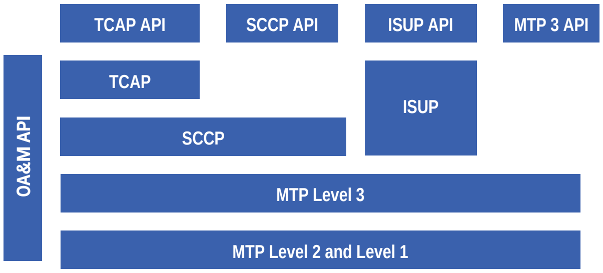

Fig. 8. *For each layer of higher-level protocols, an operation, administration, and maintenance (OA&M) programmatic access is provided.*

Each layer is directly accessible via a direct API. Some APIs are simple wrappers that get the user parameters and marshall them to the stack. Other APIs, such as the ISUP and TCAP APIs, implement some part of the protocol in the library itself, allowing wider distribution of the processing load. All of the APIs are asynchronous to allow for high transaction rates.

High Availability

As explained above, one of the key aspects of the platform is high availability. The SS7 network has built-in high availability capabilities and it is important that the end node also provide these capabilities.

Our solution is based on the *active/standby model* (see ***Article 8*** for more details). To eliminate any single point of failure, every element is replicated (see Fig. 9). Two hosts are used, one being the active host and the other the standby host. Only

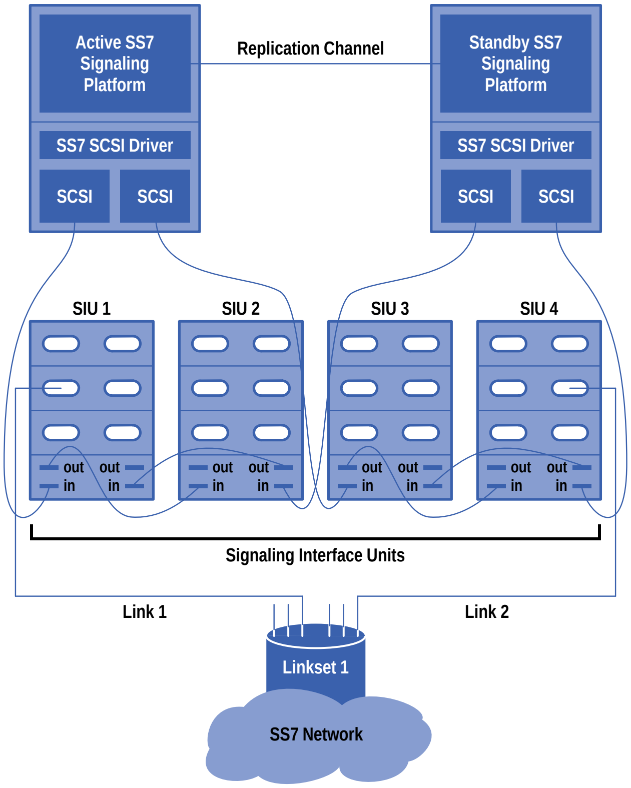

Fig. 9. *High availability of the HP OpenCall SS7 platform is based on the active/standby model.*

the active host processes the traffic, while the standby just keeps its state up to date. For network attachment, we use dual-ported signaling interface units, and each unit is connected to two different SCSI chains terminating at each host. The dual-ported signaling interface unit has built-in logic such that one and only one SCSI bus can be active at any point in time.

Each host has two SCSI interface cards, each connected to one half of the signaling interface unit set. Fig. 9 also shows how signaling interface units are chained. The active SS7 stack, running on the active host, uses the two SCSI chains terminating at it. The standby stack controls its two SCSI chains but does not have control of the dual-ported signaling interface units. The active stack has control of the signaling interface units via its two SCSI interfaces. In case of a failure of the active side, the standby side will take over and will take control of all the signaling interface units by using its two SCSI buses. The switchover happens in less than six seconds to be transparent from the SS7 network point of view. Refer to ***Article 8*** for more details on the mechanism.

The SS7 links of a given linkset must be connected to two different signaling interface units so that if one signaling interface unit happens to fail, the SS7 traffic will be routed transparently by the network to the surviving signaling interface units. Therefore, from a network attachment point of view, the architecture is more a load-sharing architecture, whereas it is an active/standby architecture at the host level.

From an application point of view, the API hides the fact that there are in fact two SS7 stacks running. Application developers are free to use their own high availability mechanism, either load shared or active/standby.

Distribution

Another important aspect of the HP OpenCall SS7 platform is its ability to support distributed applications. The key concept here is a *front-end/back-end mode*. The front-end computer supports the SS7 connection and protocol and the back-end computer supports the application. A typical configuration is shown in Fig. 10.

The SS7 stack is able to distribute the traffic among several instances of the application running on *back-end nodes*. The application instances can run on several nodes and several instances can run on the same host. The API completely hides the distribution and the active and standby instances of the stack. Thus, an application can be configured to run either on a simplex node (no high availability), on a duplex node (active/standby), or on a back-end node without modifying anything in the code. All connections between the various systems are made over a dual LAN (possibly FDDI for high-end systems) to eliminate any single point of failure.

This flexibility allows users to use their own high availability and distribution schemes.

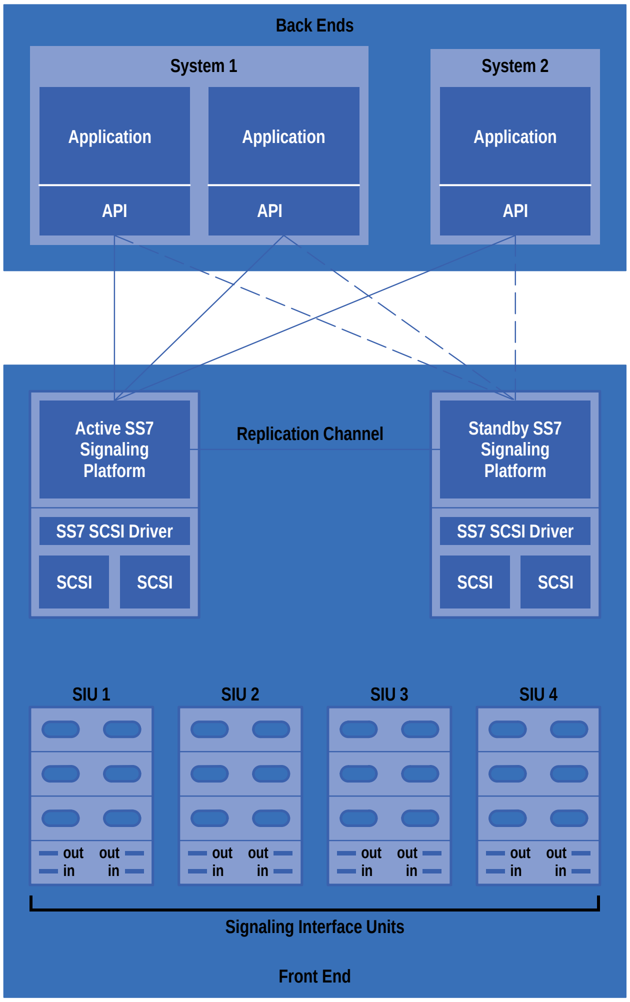

Fig. 10. *A front-end/back-end mode allows the HP OpenCall SS7 platform to support distributed applications.*

Stack Implementation

As mentioned earlier, MTP 3, SCCP, and TCAP are implemented in a single user space process. The protocol implementation started in 1988. The SS7 stack uses object-oriented technology and a message passing bus for interobject communication. Fig. 11 shows the stack implementation.

Each layer has a software bus instantiated. Entities can dynamically register on the bus, specifying what kind of messages they are interested in. Entities are object classes that model elements of the protocol. Typical objects are MTP 3 links, SCCP remote subsystem numbers, and so on. Each object instance is associated with a unique key (object identifier), usually extracted out of the protocol information, which allows very efficient dispatching by the bus. Entities can send messages on the bus to be multicast to the target entities, calling one of their base class methods.

This method has proved to be very efficient in terms of encapsulation and coupling between objects. (Note that the MTP 2 layer does not implement the MTP 2 protocol but rather provides the interface with the MTP 2 implemented in the signaling interface units).

Message Set Customization

To extend the capabilities of the SS7 platform, it is necessary to provide more built-in protocols as they are adopted. These new protocols are built on top of TCAP and are used in the intelligent network or in the mobile network. Although standardized, the flavors of these protocols vary from network to network and from vendor to vendor. The same applies for ISUP, for which there is about one version per country.

To deal with the diversity of message formats at the product level without having to do a special version for every new flavor of the protocols, we have developed a *message set customization* technology to automate the customization of a protocol. This consists of a message set compiler in conjunction with a generic run-time engine to process the encoded messages (see Fig. 12).

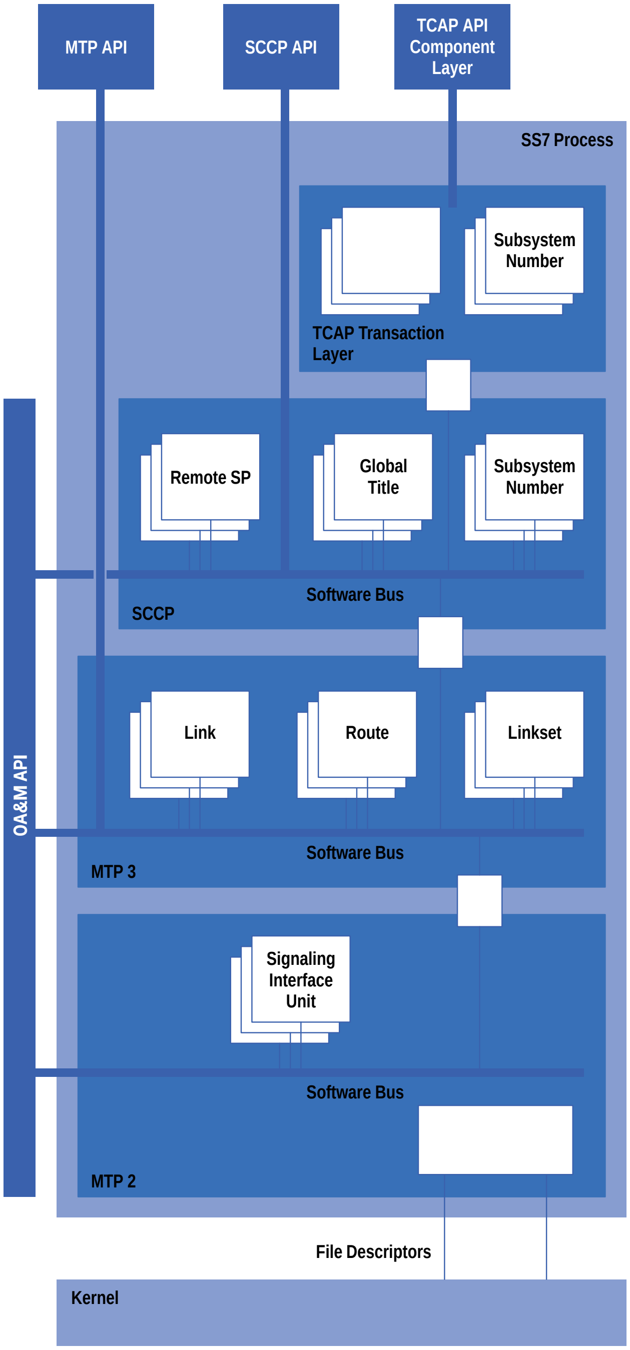

Fig. 11. *SS7 stack implementation in the HP OpenCall SS7 platform.*

The format of the messages used by the protocol is defined in Abstract Syntax Notation 1 (ASN.1) with some specific annotations. ASN.1 is the OSI standard for defining data structures and is used by most of the protocols that we implement. However, the message set compiler technology is not restricted to ASN.1. A protocol such as ISUP, which is not defined in ASN.1, can also be accommodated.

The compiler generates a *metadata file*, which contains the definition of the messages. The run-time engine loads these metadata files and can immediately encode and decode new definitions of messages without impacting the API or requiring recompilation or relinking. The benefit of this technology is that it can adapt the product to the user's exact specifications at the latest possible stage, without impacting the core product.

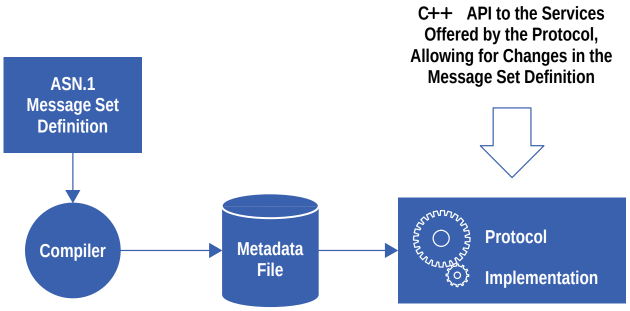
C++ API to the Services Offered by the Protocol, Allowing for Changes in the Message Set Definition
ASN.1 Message Set Definition
Compiler
Metadata File
Protocol Implementation

Fig. 12. *Message set customization technology consists of a message set compiler in conjunction with a generic run-time engine to process the encoded messages.*

Performance

The HP OpenCall SS7 platform can handle more than 2100 SS7 transactions per second, a transaction being defined as one message into a dummy application and one message out from the application. These figures were measured on an HP 9000 Model K420 host computer. The implementation is CPU-bound, so its capacity automatically increases whenever more powerful system hardware becomes available.

The constraints set up for the development were rather stringent and very similar to in-kernel development. For example, no file system access is allowed except at startup, and all calls must be asynchronous. We are closely watching the I/O traffic to avoid falling into I/O bottlenecks. One of the reasons why we do not get I/O bottlenecks is that we group as many SS7 messages as possible together before doing any transfer. For SCSI, this is mandatory because SCSI is architected for rather large data transfers whereas SS7 handles very small messages (~100 bytes). Therefore, the signaling interface unit and the SCSI driver purposely introduce some latency to transfer larger data blocks.

▶ Go to Next Article

▶ Go to Journal Home Page