



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE ESTUDIOS SUPERIORES
CUAUTITLAN

**"Redes de Computadoras. CONNECT direct
como instrumento de transportabilidad
de datos."**

TRABAJO DE SEMINARIO

QUE PARA OBTENER EL TITULO DE:

LICENCIADA EN INFORMATICA

P R E S E N T A

S A N D R A G O M E Z C O R T E S

ASESOR L.C. CARLOS PINEDA MUNGZ

**TESIS CON
FALLA DE ORIGEN**

CUAUTITLAN IZCALLI, EDO. DE MEX.

1997



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN
UNIDAD DE LA ADMINISTRACIÓN ESCOLAR
DEPARTAMENTO DE EXÁMENES PROFESIONALES

U. N. A. M.
FACULTAD DE ESTUDIOS
SUPERIORES - CUAUTITLÁN



DEPARTAMENTO DE
EXÁMENES PROFESIONALES

DR. JAIME KELLER TORRES
DIRECTOR DE LA FES-CUAUTITLÁN
PRESENTE.

AT'N: ING. RAFAEL RODRIGUEZ CEBALLOS
Jefe del Departamento de Exámenes
Profesionales de la FES-C.

Con base en el art. 51 del Reglamento de Exámenes Profesionales de la FES-Cuautitlán, nos permitimos comunicar a usted que revisamos el Trabajo de Seminario:

Redes de Computadoras, Connectivity como
instrumento de transportabilidad de datos.

que presenta la pasante: Sandra Gómez Cortés
con número de cuenta: 8810113 - 8 para obtener el Título de:
Licenciada en Informática

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXÁMEN PROFESIONAL correspondiente, otorgamos nuestro VISTO BUENO.

ATENTAMENTE.

"POR MI RAZA HABLARA EL ESPIRITU"

Cuautitlán Escali, Edo. de México, a 10 de Octubre de 19 97

MODULO:	PROFESOR:	FIRMA:
<u>I</u>	<u>L. C. Carlos Pineda Muñoz</u>	<u>[Firma]</u>
<u>II</u>	<u>ING. Miguel Alvarez Pasayo</u>	<u>[Firma]</u>
<u>III</u>	<u>ING. Jesús Moisés Hernández</u>	<u>[Firma]</u>

DEP/V0805EM

Dedico ...

**..Al principio y fin de mis días y noches;
al que me permite mis triunfos, y que
me ayuda salir de mis fracasos: DIOS.**

**...A los seres más amados y respetados
por mí que me ayudaron a hacer
camino y estuvieron junto a mí
soportando los tiempos difíciles
y que se congratularon con los días de
gozo : MIS PADRES.**

**...A la Institución que me ayudó a
crecer, desarrollarme y brindarme
conocimientos para ser
mejor: La UNAM.**

GRACIAS

SANDY.

Gracias...

A mi asesor:

Lic. Carlos Pineda el más sincero agradecimiento, respeto y cariño por la dirección y supervisión que me brindo en todo momento para la mejor elaboración de este trabajo.

A mi hermano David:

Por el apoyo, la motivación con la que conté en el pasado y con la que cuento para el futuro, y sobre todo por formar parte de la que hoy soy.

A Jorge:

Por que me brindo su aliento, entusiasmo, cariño, por los momentos y sentimientos compartidos.

SANDY

INDICE

Introduccion	<i>Página</i>
Capítulo 1	
Necesidades de Conectividad	
1.1. Demanda de la comunicación de datos.	6
1.2. Antecedentes de comunicación.	7
1.3. Sistema básico de Comunicación.	9
1.4. Connect direct for Unix: Software para el intercambio de información..	10
1.4.1 Características	12
1.4.2 Automatización Inteligente	12
1.4.3 Procesamiento de Fácil Manejo	13
1.4.4 Confiabilidad y atrapado de errores.	13
1.4.5 Seguridad	14
1.4.6 Diversas Plataformas, Diversos Sistemas Operativos.	14
1.4.7 Alternativa de protocolos	14
1.5 Componentes de Connect direct para UNIX.	16
 Capítulo 2	
Protocolo de Control de Transmisión TCP/IP	
2.1. Introducción al Protocolo TCP	28
2.2. Características de TCP/IP.	29
2.3. Funcionalidad de TCP/IP.	30
 Capítulo 3	
Características para la transmisión de datos.	
3.1. Velocidad de transmisión.	37
3.2. Capacidad de transmisión.	37
3.3. Sincronización de la comunicación.	39
3.4. Compresión de datos.	41
3.5. Enlaces de Comunicación.	41
 Capítulo 4	
Caso practico: Instalación y administración de Connect:direct for UNIX	
4.1 Preparación para la instalación.	45
4.2 Instalación y administración de Connect:direct for UNIX	46
 Conclusiones	 69
 Bibliografía	 72

OBJETIVO GENERAL

Establecer las condiciones necesarias para el envío y recepción de paquetes de datos considerando ampliamente las cualidades de CONNECT direct como software de transportabilidad de información.

OBJETIVO ESPECIFICO

Instalar y administrar CONNECT direct como instrumento de transmisión y recepción de paquetes de datos del área de operaciones de Afore Santander Mexicano hacia Proccsar.

Capitulo 1

Necesidades de Conectividad

1.1 Demanda de la comunicación de datos

La creciente integración de computadoras y comunicaciones dentro de un sistema único, a ocasionado que la industria vaya evolucionando con un rápido crecimiento en la comunicación de datos. Aunque su antigüedad no es mucha, en la última década, los logros tecnológicos dentro de la industria han sido significativos. En diversos sectores tanto comerciales, industriales, educativos, financieros, etc. muchos usuarios necesitan el servicio de la computadora como medio de comunicación, existe una posibilidad cada vez mayor de que los servicios de comunicación de datos enlacen el servidor central con usuarios remotos. Esta tendencia a crecer rápidamente es en realidad totalmente universal. En todo el mundo se han experimentado considerables adelantos técnicos, así como un marcado incremento en la disponibilidad de servicios de comunicación.

Los adelantos de la tecnología permiten que las comunicaciones tengan un lugar muy importante ya que en este tiempo a través de la tecnología se comunican a grandes distancia cada vez con mayor eficiencia.

El teléfono se ha transformado en una necesidad y la computadora se está convirtiendo en una herramienta completamente necesaria para la vida cotidiana. Este rápido cambio ha forzado a nuevas ideas de diseño y conceptos tecnológicos revolucionarios. Hoy en día es cada vez mayor la interrelación y la interdependencia de oficinas y lugares de trabajo geográficamente dispersos. Para esto se exige una disponibilidad de datos que cumplan con las siguientes premisas:

"la persona adecuada, que debe recibir, la información adecuada en el momento adecuado." Esto obliga a los sistemas que demandan la información no importando la fuente, las distancias ni el lugar de destino.

En México, como en muchos otros países, se ha visto la convivencia de implantar redes dedicadas a la transmisión de datos que favorezca el procesamiento de información a distancia, el cual tiende a incrementarse de manera inusitada, ya que el desarrollo de las

aplicaciones utilizando las nuevas tecnologías constituyen un factor de transformación de la organización económica y social, y del modo de vida en general.

1.2 Antecedentes de comunicación

“El arte de la comunicación es tan antiguo como la humanidad. En la antigüedad se usaban tambores y humo para transmitir información entre localidades. A medida que pasó el tiempo se crearon otras técnicas. La era de la comunicación electrónica inicio en 1843 con el invento del telégrafo, y su código asociado. El invento del telégrafo adelantó la posibilidad de comunicación humana, no obstante se tenían muchas limitaciones. Uno de los principales defectos fue la incapacidad de automatizar la transmisión. Debido a la incapacidad técnica de sincronizar unidades de envío y recepción automáticas y a la incapacidad propias del código morse de apoyar la automatización.” En el año 1874 Emil Baudot en Francia ideó un código en el cual el número de elementos (bits) en una señal era el mismo para cada carácter y la duración (sincronización) de cada elemento era constante. Este código fue llamado de longitud constante.

En 1910, Howard Krum introdujo mejoras en este incipiente concepto de sincronización y lo aplicó al código de longitud constante. Este desarrollo llamado start/stop , condujo a la rápida difusión del uso de equipos automáticos de telegrafía. El primer equipo teleimpresor operaba sin ningún protocolo identificable, tan pronto como la máquina local comenzaba a transmitir, la máquina receptora copiaba la transmisión. A este sistema se le llama “no controlado” . A medida que las comunicaciones se volvieron más sofisticadas, en el comienzo de los años 50 se introdujeron dispositivos electromecánicos centrales para realizar tareas notificando en secuencia a cada estación del mismo circuito para transmitir su tráfico y notificando a una determinada estación que debe recibir un mensaje, esto fue el principio de enviar, recibir, reacondicionar o realizar funciones básicas. Más tarde se introdujeron las teleimpresoras , que transmitían ya fuera directamente por el teclado o por medio de la cinta y el producto final era una cinta perforada o bien una copia impresa, estas fueron instaladas en 1970 en todo el mundo empleando el código de Baudot.

Paralelamente el desarrollo del teléfono dio lugar al desarrollo básico de las telecomunicaciones, conmutación de mensajes y control de línea. Los sistemas se construyeron con base en comunicaciones a través de la voz y la transmisión de caracteres de datos. Después de la Segunda Guerra Mundial comenzó el desarrollo comercial de la computadora. Como estas primeras máquinas eran orientadas a lotes, no existía la necesidad de interconectarse con el sistema de comunicación que abarca a toda la nación, sin embargo más adelante la industria tomó conciencia de la conveniencia de que máquinas y gente hablaran entre sí. Dado que el único sistema de comunicación disponible era el teléfono, naturalmente, las computadoras en evolución, habrían de desarrollarse siguiendo vías que les permitieran usar este servicio.

El crecimiento del uso de la comunicación fue simultáneo al crecimiento de la tecnología de las computadoras y en parte, favorecido por él. Las redes de conmutación de mensajes, reservación y transacciones financieras en los años 50 y 60 usaban computadores centralizados sofisticados para controlar grandes volúmenes de dispositivos y terminales primitivas. A medida que esas redes crecían en lo que se refiere a volúmenes de tráfico y población de terminales, el aspecto no controlado de la operación de las terminales se volvió inaceptable. En los años 60 las aplicaciones de comunicación de datos se expandieron más allá del intercambio rutinario de tráfico de mensajes. Los patrones de tráfico, la extensión de los mensajes, los requerimientos de tiempo de respuesta, los parámetros relacionados con estas nuevas aplicaciones fueron significativamente diferentes de a las primeras aplicaciones. Esto condujo a la necesidad de nuevas técnicas de transmisión. Con la tecnología disponible se logran velocidades más altas, más terminales en un circuito dado, mejor control de errores y otras mejoras.

A finales de los 60, las operaciones sincronicas comenzaron a suplantar los métodos asincronicos. Las técnicas de transmisión fueron en gran parte el resultado de presiones provenientes de la creciente popularidad de las comunicaciones.

1.3 Sistema Básico de Comunicación

El sistema de comunicación más sencillo es el constituido por una sola entidad transmisora y sólo una unidad receptora. Cuando la conexión entre el receptor y el transmisor no tiene conexiones intermedia, más que amplificadores o repetidores, con esto se dice que se tiene un enlace directo, o sea que los dos únicos equipos compartiendo el medio de transmisión se dice que se tiene un enlace punto a punto. Cuando existen varios dispositivos compartiendo el medio de transmisión tenemos un enlace multipunto.

De esta forma un sistema básico de comunicación consiste en sólo una unidad transmisora y solo una unidad receptora. Esta pequeña red comienza con la estación primaria (transmisora) , la información es enviada de esta estación a una estación remota o secundaria en el otro lado del enlace.

La comunicación de datos se puede describir como el intercambio de código de información digital entre dos Dtes, (data terminal equipment) . La separación física de 2 equipos puede variar en distancias cortas por ejemplo entre dos computadoras personales, ó en distancias muy largas de cientos de kilómetros a través de una red pública.

Siempre que se habla de transmisión de datos se habla de transferencia de información entre dos puntos o dos soportes cualesquiera, utilizando para ello un medio físico que los interconecte.

El término dato es normalmente reservado para describir un segmento o un bloque de uno ó más caracteres alfabéticos o numéricos digitales codificados. Típicamente estos datos representan un documento almacenado el cual se intercambiaría entre dos nodos.

Tanto en los sistemas de comunicación como en los de procedimientos de datos, la información tiene que ser codificada de alguna forma para ser utilizada. El gran mérito de la clave Morse es el haber sentado las bases para el diseño de código binario (en base a dos símbolos solamente) para la transmisión de la información mediante un equipo como se mencionó en el primer párrafo, en términos generales esta idea fue el principio de la creación y desarrollo de sistemas modernos de comunicación y de procesamiento. Los datos de construyen mediante una unidad fundamental conocida como bit (BInary digiT) la cual

solo puede tener dos valores cero o uno. Existen diferentes códigos encargados de establecer la correspondencia entre bits y caracteres. Todos los códigos se basan en el mismo principio, existe un predeterminado grupo de bits dentro del cual cada diferente combinación de unos y ceros representa un carácter.

1.4 Software para el intercambio de información: CONNECT direct

En nuestros días la distribución de las redes, los multiprotocolos y los sistemas operativos en multiplataformas nos dan como resultado la creciente necesidad de evolucionar y desarrollar nuevas técnicas para el manejo de la información. Para ello es necesario contar con instrumentos inteligentes que permitan transferir la información de una manera eficiente hablando en un sentido aplicativo.

Connect Direct ofrece la posibilidad de proporcionar una amplia solución de transportabilidad de datos que se implante de una manera óptima y que sea transparente con las operaciones existentes y soporte la integración de nuevas aplicaciones (hardware o software).

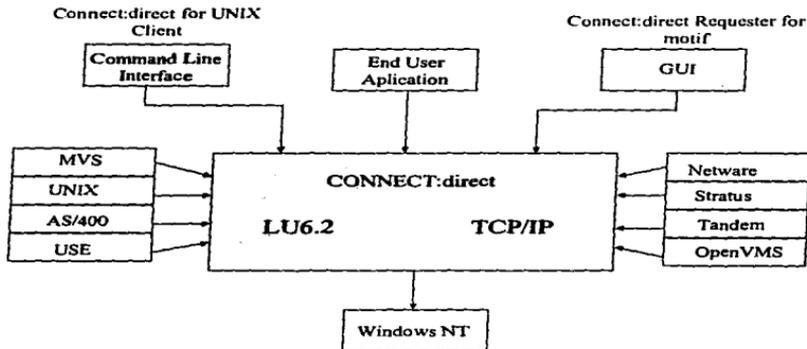
Este producto ofrece un acceso rápido a la información, cubriendo así la necesidad actual de eficiencia y de seguridad en la transmisión de datos a través de la red. CONNECT direct es una solución creada para responder a las necesidades de transferencia de datos entre las afores e instituciones gubernamentales que lo requieren, independientemente de la plataforma que utilicen. Explica que al inicio, la concentración de la información se realizaba vía mensajeros, quienes llevaban físicamente los datos a cada Afore en disco o en cintas magnéticas a la empresa creada para administrar la base de datos del SAR (Procesar), y ésta última a través de operadores procesaban dicha información, pudiendo cometer errores humanos.

Por tal motivo, Procesar en combinación con Avantel; realizó esfuerzos para enlazar el sistema mediante el protocolo TCP/IP, tanto en Afores como en instituciones gubernamentales, con el fin de mejorar y agilizar los modulos de procesamiento, sin embargo, aún no contaban con el medio adecuado para el intercambio de información.

Para esto se realizó una investigación en las instituciones de la industria bancaria y se encontró que la mayoría de las éstas contaban con software CONNECT:direct , constatando que dicho software presentaba un medio adecuado para las necesidades de la empresa concentradora.

Con la experiencia de implantar CONNECT:direct en las 16 Afores y 4 instituciones gubernamentales se explica que CONNECT:direct cuenta con componentes que aseguran la entrega de la información, realiza traslación de caracteres automática y permite la automatización de procesos, todo ello independiente de las plataformas utilizadas.

Fig. 1 CONNECT:direct Multiplataforma, Multiprotocolos, Diversos Sistemas Operativos



1.4.1 Características

Connect Direct mantiene un alto desempeño en la transferencia de archivos, transmite de manera automática la información ya sea punto a punto o permitiendo el intercambio a través de múltiples plataformas, siendo este un factor muy importante para el desarrollo de las actividades de las empresas. Connect direct es ideal para el intercambio de información automáticamente en grandes volúmenes que se pueden presentar internamente en una empresa o externamente con clientes o socios en otra zona.

Connect Direct es un software confiable, que primordialmente mantiene la integridad de la información. Este producto ha sido considerado para su uso en diversas empresas dedicadas especialmente al sector gubernamental.

1.4.2 Automatización Inteligente

Connect direct mejora la productividad de los sistemas, eliminando la necesidad de interacciones manuales cuando se procesa la información. La automatización facilita su manejo permitiendo:

- Inicialización local y remota de las aplicaciones, utilerías y comandos.
- Predefine los jobs a cierto tiempo que se están ejecutando de manera continua.
- Establecer periodos de ejecución con intervalos de tiempo.
- Asignación de prioridades en los procesos.
- Detección y documentación de errores provenientes de la recuperación de la información
- Generación de reportes o estadísticas de comunicación.
- Notificación a los usuarios de terminación de jobs.

Por todo esto se puede determinar que el contar con un control automatizado de las aplicaciones y jobs que se encuentren en el nodo central requerirán menor administración.

1.4.3 Procesamiento de fácil manejo

La integración completa entre las funciones de transferencia de datos en la red pueden ser realizadas sin la necesidad de reescribir o modificar las aplicaciones existentes. En este punto solo se requiere de un lenguaje único de script llamado process. Connect direct define el trabajo que se tiene que hacer al interactuar con la transferencia de archivos, utilizando diversas sentencias separadas por condiciones que interpreta process. De manera eficiente se pueden programar diversos scripts que efectúen múltiples tareas de envío y recepción de datos en pasos lógicos y secuenciales permitiendo incrementar la productividad de la empresa, esto es poder contar con actividades que se requieran en producción todos los días a cualquier hora.

1.4.4 Confiabilidad y atrapado de errores

Connect direct provee un alta confiabilidad en el ambiente de comunicaciones y con instrucciones sencillas que permitan atrapar errores de manera inmediata, con esto tiene la capacidad de reportar los diagnósticos de la operaciones que se realicen. Una de las características de este software se refiere a la posible restauración de los jobs en ejecución. Esto permite que Connect direct restaure un job desde el punto de falla y no comience desde el principio, en esta operación se garantiza la integridad de los datos. Automáticamente se proporcionan reportes de estadísticas que son comprensibles y que muestran las operaciones ejecutadas. Con estos reportes se pueden realizar análisis de las actividades efectuadas en las operaciones en un periodo de tiempo.

1.4.5 Seguridad

Las opciones de alta seguridad proporcionada por Connect direct facilita el control de acceso a los datos a través de la red. En este producto se maneja la encriptación de datos que trabaja de manera eficiente para resguardo de la información. Esta se puede complementar con sistemas de seguridad de otros sistemas que permitan la protección de los datos.

1.4.6 Diversas plataformas, Diversos Sistemas Operativos

Desde los grandes mainframes, LAN o las computadoras personales este software corre en MVS, VS, VSE, OpenVMS, Tandem NonStop Kernel, Stratus, Netware, Unix, Windows NT, OS/2, o MS-DOS. Connect direct utiliza al máximo los beneficios de ejecución sobre cada plataforma que se esté utilizando, Además Connect direct ofrece estructura y sintaxis común entre los diversos sistemas operativos de la actualidad con interfaces de usuario que son apropiados a cada sistema en la red.

1.4.7 Alternativa de Protocolos

Con los beneficios de Connect direct soporta los protocolos estándar, tiene fáciles opciones de conectividad que permiten métodos de acceso más eficiente.

Los protocolos que soporta :

- TCP/IP
- SNA (LU1, LU2, LU6.2)
- Netex (HyperChannel)
- ICE
- Decnet
- Async
- SNAX

- IPX/SPX
- X.25

Fácil instalación

Porque no necesita modificación de sistemas o software existente. Connect direct puede ser instalado en muy poco tiempo, y es un software flexible a la medida de la plataforma en la cual se trabaje, tiene una interfaz amigable y los requerimientos necesarios son mínimos.

Términos que se deberán conocer a cerca del tema

CONNECT:direct Es la familia de productos de software para la transferencia de datos, distribuyen la información y facilitan la administración de centros de datos no importando la localización física. Connect:direct fue conocido anteriormente como Redes de datos móviles (NDM Network Datamover).

CONNECT:direct for UNIX. El producto de connect:direct implementado específicamente sobre plataforma UNIX.

CONNECT:direct Node. Alguna computadora o workstation conectada en la red donde corre Connect:direct

CONNECT:direct Process. Es una serie de comandos que son almacenados en un archivo llamado script donde es generado el código fuente indicando las funciones que se deben realizar. Ejemplo:

```
get_file() {
/opt/cdirect/cdunix/ndm/bin/ndmcli -x << CDEOF
  sub proc1 process snode=INSCONS.UNIX

  step1 copy
  to( file=/export/home/rec/AFORES/A540/S100/RECEPCION/$FILE_NAME
  DISP=SHR
  snode )
  from ( file=$SPATH_NAME/$DATE_FMT/$FILE_NAME
  pnode )
  pend;
CDEOF
}
```

PNODE (Primary Node). Es el nodo en el cual se comienza la transmisión.

Session. Es la conexión entre dos nodos Connect:direct (SNODE Y PNODE)

SNODE (Secondary Node). Es el nodo que interactúa con el nodo primario durante la ejecución de procesos de transmisión, se le puede referir como nodo destino.

1.5 Componentes de Connect Direct para Unix

Este es el producto que se va instalar para que opere en Afore Santander Mexicano, bajo un ambiente de producción UNIX.

El producto Connect direct para Unix está basado en una arquitectura cliente/servidor, Sin embargo en este caso especial para Afore Santander no actúa de esta manera, sino los dos nodos son servidores y están esperando la llamada de uno u otro para poder establecer una sesión.

Vida de un proceso

1. Proceso es construido
2. Proceso es validado en su estructura es decir, revisa semántica, sintáxis, usuario identificado.
3. Proceso en espera para ejecución
4. Proceso ejecutado
5. Genera estadísticas y termina.

Los siguientes componentes de este producto interactúan entre sí para ejecutar sentencias y comandos aplicados a través de una interfaz:

1. **Administrador de Procesos (PMGR)** en la parte del servidor
2. **Administrador de Comandos (CMGR)** en la parte del servidor
3. **Administrador de Sesiones (SMGR)** en la parte del servidor
4. **API (Applications Programming Interface)** este es componente del cliente
5. **CLI (Command Line Interface)** en la parte del cliente
6. **Solicitud de un ambiente gráfico, Motif o Windows** en la parte del cliente

1. Administrador de Procesos (PMGR)

Es el proceso padre que mantiene el ambiente de operaciones de transmisión y recepción radica en la parte del servidor y está corriendo todo el tiempo en espera de realizar lo que se le pida por ejemplo:

- Inicializar el software de connect direct
- Acepta las peticiones de conexión con el cliente, utilizando el API para conexiones con nodos remotos las cuales son capaces de comunicarse con el PMGR.
- El PMGR crea consigo dos procesos hijo, el Administrador de Comandos CMGR y el Administrador de sesiones SMGR para comunicarse con los APIs y comunicarse con los nodos remotos respectivamente.
- Acepta las peticiones del CMGR y SMGR y las centraliza para realizar todos sus funciones
- Termina la ejecución de Connect direct.

El PMGR puede correr en diversas máquinas de la misma red. Y pueden existir múltiples copias en la misma máquina simultáneamente.

2. Administrador de Comandos (CMGR)

El CMGR es creado por cada conexión API que se establezca exitosamente. El número de procesos CMGR que el PMGR pueda crear depende del tipo de sistema y está limitado por el número de archivos disponibles por cada proceso UNIX.

Este proceso proporciona las siguientes funciones:

- Ejecuta los comandos enviados por los APIs y envía el resultado de regreso.
- Realiza la autenticación del proceso, en conjunto con el API para determinar el acceso a Connect Direct.
- Interactúa con el PMGR cuando ejecuta los comandos que se envían.
- El CMGR actualmente limita el tamaño del process a 60K porque es el parámetros por default pero este puede ser incrementado en el iniparm.cfg.

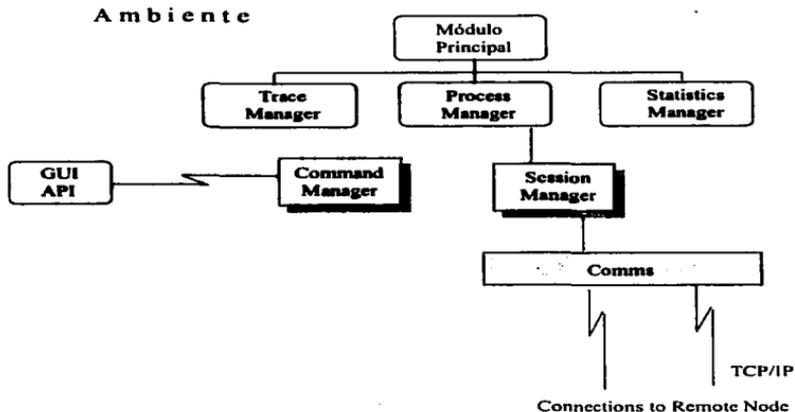
3. Administrador de Sesiones (SMGR)

El SMGR es creado e invocado por el PMGR cuando los recursos están disponibles y algún proceso está listo para ejecución o cuando un nodo remoto requiere un conexión con el nodo local.

Este proceso proporciona las siguientes funciones:

- Es un proceso importante para desempeñar el trabajo de Connect direct.
- Este actúa como un nodo primario (PNODE), el pnode es el que hace referencia a un nodo local o LNODE, es el nodo responsable de la inicialización de la ejecución de proceso.
- Este puede actuar como un nodo secundario (SNODE), el snode es el que hace referencia al (RNODE) o nodo remoto, este es el nodo que participa en la ejecución del proceso inicializador por otro pnode.

Fig 2. Ambiente de Operación de Connect:direct



Nodo local y nodo remoto

En cada transferencia de datos se involucran dos partes interesantes el nodo remoto y el nodo local. El nodo local se encuentra activo en la parte del servidor (LNODE) y el nodo remoto (RNODE) en un sitio distante. Los dos servicios trabajan conjuntamente para completar el trabajo de transmisión de información, cada uno de los nodos puede inicializar el trabajo, el nodo que inicializó la sesión de transferencia tiene el control primario sin hacer caso de la dirección del flujo de la información.

La conexión entre el nodo remoto y el nodo local se realiza a través de una ruta de comunicación común que debe estar perfectamente definida en el ambiente de red.

Proceso de Envío

El lenguaje que se maneja en Connect direct a través de instrucciones para la realización de los scripts de ejecución le permite realizar la transferencia de archivos, el ejecución de tareas, iniciar otro proceso de Connect direct y alterar la secuencia de los pasos de ejecución de algún script.

Un proceso se inicia con la definición de la sentencia (process) y puede recibir diferentes parámetros para su funcionamiento. En el siguiente cuadro se presenta la sentencia y su función (Cuadro 1).

<i>Process statement</i>	<i>Función</i>
copy	Copia archivos de un nodo a otro.
conditionals	Altera la secuencia del proceso basado en los comandos if, then, else, eif, goto y exit.
process	Define las características generales de process
run job	Permite especificar comandos de unix en el procesos. El proceso no espera hasta que el job finalice continua con los siguientes pasos del proceso.
run task	Permite especificar comandos de unix en el procesos. El proceso espera hasta que el job haya finalizado después continua con los siguientes pasos del proceso.
submit	Inicia otros procesos que pueden ser del nodo remoto o del nodo local durante la ejecución del proceso.
pend	Marca el final del proceso de Connect direct

Cuadro 1.

Ejemplo:

```
ckpt01 process snode=unix.node
step01 copy from (
    file=file1
    snode
)
ckpt=1k
to
(
    file=file2
    disp=new
    pnode
)
```

pend;

Estructura de la sentencia

Label	Statement id	Parameters
-------	--------------	------------

Label comienza en la primera columna del script de ejecución, esta etiqueta del proceso debe comenzar con un carácter alfabético.

sub **procl** process



Label

Statement id. Identifica la función a realizar

step1 copy



Statement id

Parameters. Especifica diversas funciones adicionales sobre la función descada separados por comas o espacios.

Parameters { compress=extended
ckpt=100k

El control de transmisión

El control de transmisión en espera (TCQ) controla la ejecución del proceso, es decir como opera Connect direct. El TCQ es dividido en 4 colas de espera lógicas: Ejecución, espera, tiempo programado en espera y detenido.

En cada parte del proceso es asignada un cola de espera basada en los parámetros que afecten el esquema, ejemplo de cada parámetro existente, detenido, retención y ejecución.

El producto Connect direct le asigna prioridades a cada proceso conforme al sistema primeras entradas, primeras salidas y de esta manera se conocen como están disponibles los procesos. Los procesos con mayor prioridad son seleccionados para ejecución antes que los procesos de menor prioridad. A través de los comandos de Connect direct se puede acceder a la cola de espera y manipular el proceso, con el uso de estos comandos se puede realizar:

- * Administración de Procesos en la cola de espera
- * Monitoreo de la ejecución de los procesos
- * Producir reportes en base a las actividades de los procesos
- * Detener la operación de Connect direct

Ejemplo:

```

.....
.
.      CONNECT:Direct for UNIX      .
.
.-----
. Copyright (c) 1983, 1996 Sterling Commerce, Inc. .
.      Version 1.4.00 GA      .
.....

```

Enter a ';' at the end of a command to submit it. Type 'quit;' to exit CLI.

Direct> sel proc;

SELECT PROCESS

PROCESS NAME	NUMBER	USER	SUBMITTER	NODE	QUEUE	STATUS
procl	6139	cdirect	AFOsame.unix		EXEC	EX
procl	6133	cdirect	AFOsame.unix		HOLD	HE

Aquí se presentan la lista de los comandos y sus funciones (Cuadro 2).

Comando	Función
change process	Cambios en el estatus y modificaciones en las características específicas de un proceso que no se ha ejecutado en el TCQ.
delete process	Borrar los procesos del TCQ no ejecutados
flush process	Borra los procesos en ejecución del TCQ.
trace	Corre traces, reporte de actividades por proceso.
select process	Despliega o imprime información acerca de los procesos en el TCQ.
select statistics	Despliega o imprime estadísticas en el archivo de log
stop	Detiene la operación de Connect direct
submit	someter un proceso para ejecución

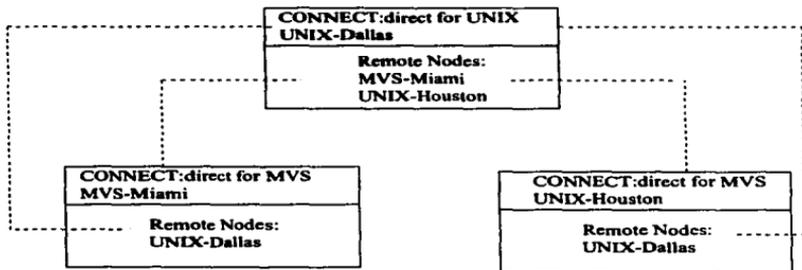
(Cuadro 2).

Arquitectura de la red

Durante la transferencia de datos, el servicio activo de Connect direct está en el nodo local y en la parte remota el nodo asociado. Connect direct identifica el nodo remoto con el cual el nodo local es capaz de establecer una comunicación a través de utilizar la arquitectura de la red, llamada Netmap

La Netmap incluye la información de los nombres de todos los nodos remotos que pueden comunicarse con el nodo local de Connect direct , también las rutas de acceso de estos nodos y las características de las sesiones para establecer comunicación.

El nodo remoto también cuenta con su Netmap sobre sus nodos remotos. El siguiente diagrama presenta la correspondiente Network map que se establece entre diversos nodos como son UNIX-Dallas, MVS-Miami y UNIX-Houston. Fig. 3.



Autorización de Usuarios

El producto Connect direct tiene la facilidad de autorizar usuarios locales y remotos para la ejecución de tareas de manera segura y certificada.

Shell scripts

El software trae consigo archivos de muestra para la creación de procesos que están destinados en el home de Connect direct , estos archivos pueden ser modificados según sean las necesidades o se pueden crear nuevos tomando como base estos. Estos deben existir para poder realizar los diversos tipos de servicios.

<i>Filename</i>	<i>Tipo de proceso</i>
cpunx.cd	copy
rtunx.cd	run task
rjunx.cd	run job
sbunx.cd	submit

<i>Filename</i>	<i>Tipo de shell script</i>
selstat.sh	select statistics
send.sh	send
recv.sh	receive

Interface en la línea de comandos

La interface de la línea de comandos (CLI) permite la utilización de comandos propios de connect direct para la operación de transmisión de archivos

Para invocar al CLI se utiliza el comando direct si la variable de ambiente NDMAPICFG ya está establecida en el archivo de .profile no será necesario exportar la variable de lo contrario se exportará la variable antes de invocar direct de la manera siguiente:

```
$ NDMAPICFG= $home_cdirect/ndm/cfg/cliapi/ndmapi.cfg
```

```
$ export NDMAPICFG
```

Forma de mandar a ejecutar el CLI es:

- Abrir una sesión en Unix con el usuario cdirect ya que esté solo será el usuario que tenga los privilegios de ejecución sobre CLI.
- Teclar : **\$ direct**

Capitulo 2

Protocolo de Control de Transmisión TCP/IP

2.1 Introducción al Protocolo TCP

En el nivel más bajo, las computadoras entienden únicamente sucesiones de unos y ceros. La manera como los usuarios comunican a las máquinas a través de estas cadenas de dígitos binarios es mediante el sistema operativo. A lo largo de la historia de la computación, se han desarrollado diferentes sistemas operativos. Cada tipo de computadora a desarrollado o mandado a hacer un sistema operativo. Sin contar con las actualizaciones de cada uno de ellos. La diferencia de sistemas operativos tienen poco de malo si una computadora no va a estar conectada con ninguna otra o eventualmente si sólo va a estar conectada a computadoras de su misma especie. Pero en la realidad actual todas las computadoras deben "hablar" entre ellas independientemente de su sistema operativo.

Para poder comprender la operación de transmisión de archivos es necesario conocer el protocolo de que me permita realizar estas funciones para este caso TCP es el protocolo que me permitirá comunicarme de mi ambiente de trabajo bajo plataforma UNIX en AFORE SANTADER A PROCESAR en un ambiente en IBM.

Un protocolo define las reglas de un procedimiento que una computadora debe obedecer cuando establece comunicación con cualquier otra. El nivel básico del protocolo se define cuando los paquetes serán transmitidos sobre una red. Este acceso entre protocolos permite el intercambio fundamental de información, en el nivel más alto el código de caracteres puede ser interpretado por una aplicación de software apropiada. Estas reglas establecidas por los protocolos siguen distintos requerimientos que son gobernados en varios niveles jerárquicos.

El desarrollo de TCP/IP comenzó a principio de los años setenta cuando DARPA (Agencia de Departamento de Defensa de Estados Unidos para proyectos de investigación Avanzada)

concedió varios contratos de desarrollo y fue adoptada rápidamente como arquitectura de red estándar por muchas redes de gobierno de Estados Unidos. DARPA y algunas otras dependencias de la defensa estadounidense contrataron con compañías de computo el desarrollo de software TCP/IP para las combinaciones de las máquinas y sistema operativo que tenía el gobierno estadounidense. DARPA contrató también a Bolt Beranek y Newman para hacer desarrollos de software para máquinas Unix e inició un programa de prueba y certificación de software para máquinas UNIX e inició un programa de prueba y certificación de software para que cubriera los estándares TCP/IP. La combinación unix TCP/IP se ha transformado en una combinación popular para los usuarios de redes de área local que desean mantener una arquitectura abierta, dado que los protocolos TCP/IP no son propiedad de ninguna casa comercial.

2.2 Características de TCP/IP

TCP/IP podrá ser parte fundamental en la interconexión de redes de diversos tipos por sus características únicas que han permitido crear una red de redes llamada internet. Pudiendo comportarse como una red simple compuesta por diversos nodos o servidores conectados para construir una red de comunicaciones conectados punto a punto

Este protocolo fué diseñado para poder ligar diversos tipos de hardware y sistemas operativos, que requieren ser robustos por la necesidades actuales y que permita detectar de manera eficiente tipos de errores que se presenten en los niveles del protocolo.

La implementación de TCP/IP provee tres servicios básicos como son : tranferencia de archivos, login remoto, y correo electrónico.

2.3 Funcionalidad de TCP/IP

Cuando dos computadoras se comunican entre ellas, una recibe los datos que la otra envía. Sin el protocolo TCP/IP la máquina que recibe no sabría que hacer con los datos que va recibiendo.

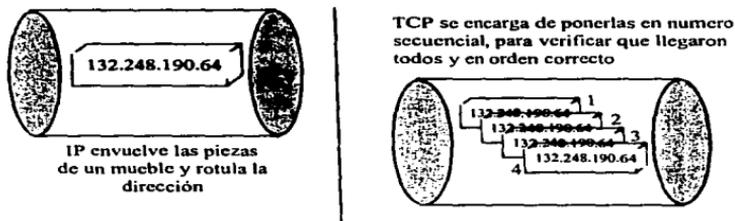
El procedimiento que se utiliza para intercambiar datos de manera segura en computadoras es de la manera siguiente:

1. Paquete de datos
2. Determinara la ruta que deben seguir los datos
3. Regular la tamaño de transmisión de datos considerando el ancho de banda y la capacidad de recepción del nodo receptor de datos.
4. Transmitir los datos en un medio físico.
5. Conservar la unión de los paquetes de datos a través de secuencias para no perder ninguna pieza.
6. Checar que los datos lleguen de manera adecuada.
7. Notificar el número de paquetes enviados contra los paquetes recibidos.
8. Entrega de los datos en la aplicación
9. Detectar errores o eventualidades.

La capa de red se denomina protocolo de entre redes (Internet Protocol IP); este protocolo para datagramas direcciona los mensajes a través de la red, intercambiando datos entre sistemas independiente de la topología de l red y del medio empleado. Este protocolo se encarga de direccionar la información entre los nodos de la red. IP proporciona los mecanismos para mandar los datos, pero no puede garantizar que llegue de manera correcta. IP forma paquetes de datos que envía a través de la red y pueden llegar a tener hasta 65 535 bytes de 8 bits.

Para poder enviar los paquetes a una máquina en particular a cada una de la computadoras conectadas a Internet se le asigna una dirección IP. Esta es un conjunto de cuatro números separados por un punto. Esto se así porque cada uno de los cuatro números que forman la dirección es un byte de 8 bits. La dirección total tiene entonces 32 bits. Lo que permite direccionar alrededor de 43 000 millones de computadoras. TCP hace de s cosas más que no hace IP: garantiza la entrega el orden correcto de los paquetes.

Fig. 3 Funciones de TCP e IP



TCP hace también algo adicional, lleva registro del número de puerto. Esto es importante sobre todo para la presentación de servicios de red. TCP/IP es responsable de que pueda haber correo electrónico entre máquinas de distinta arquitectura y con distintos software localizados en lugares geográficamente muy apartados.

Se dispone de un protocolo adicional de control de mensajes (ICMP) para proporcionar funciones de gestión de nivel de red y funciones de control.

La capa transparente se denomina protocolo de control de transmisión (TCP, Transmission Control Protocol) y es un protocolo de transporte orientado a la conexión que secuencia los mensajes en transacciones, y proporciona un servicio de transporte extremo a extremo similar al del protocolo ISO 8073 clase 4. El TCP dispone sólo de un tipo de TPDU y la

cabecera necesita 20 octetos, comparados con los 5 a 7 empleados en la cabecera del protocolo clase 4 del ISO 8073. Las capas de sesión y presentación, denominadas protocolos telnet soportan circuitos virtuales entre terminales y ordenadores centrales, permitiendo al usuario de una máquina conectarse con otra. El protocolo de transferencia de ficheros (FTP, File transfer Protocol), que corresponde rigurosamente con las capas de presentación y aplicación, facilitan la transferencia de ficheros entre máquinas y sistemas operativos diferentes.

Fig. 4 Los Niveles de TCP/IP comparados con el modelo OSI (Open System Interconnection)

Niveles OSI	Niveles TCP/IP
7. Nivel de Aplicación	Protocolo de Transferencia de archivos (FTP) entrada remota de trabajos
6. Nivel de Presentación	
5. Nivel de sesión	Telnet Protocolos para terminales virtuales
4. Nivel de transporte	Protocolos de control de transmisiones o protocolos de datagramas de usuario
3. Nivel de red	Protocolos Internet (IP)
2. Nivel de Enlace	Nivel de enlace dependiente de la red
1. Nivel Físico	Nivel físico dependiente de la red

Los datos son almacenados dentro de unidades llamadas frames o packets y enviados desde un interface en un sistema local hacia una interface destino unidos desde una red física como una red LAN o WAN que proveen las funciones de las capas más bajas.

El Internet Protocol (IP) rutea los datos entre diversos nodos o servidores que puede realizarse en una red simple o en una red distribuida de sistemas abiertos. Los datos son transportados en unidades llamadas datagramas. El IP es llamado connectionless porque cada datagrama es ruteado de manera independiente y el IP no garantiza la entrega puntual o en secuencia de los datagramas. El IP puede ser comparado con la capa 3 del modelo OSI que representa la capa de red.

TCP contiene los mecanismos adecuados para poder garantizar que los datos fueron entregados completos, libre de errores y en orden o secuencia. TCP envía unidades llamadas segmentos que son pasados de la capa IP quienes rutean al destino. TCP acepta los segmentos enviados del IP determinando cual aplicación será la receptora, y pasa los datos a la aplicación de la misma manera como fueron enviados.

Las aplicaciones invocan el UDP (User Datagram Protocol) en el orden para envío de mensajes que pudieran ser aislados por cada aplicación. Los paquetes de datos UDP se encuentran en unidades llamadas User datagrams y ellos pasan al IP para ser ruteados al destino.

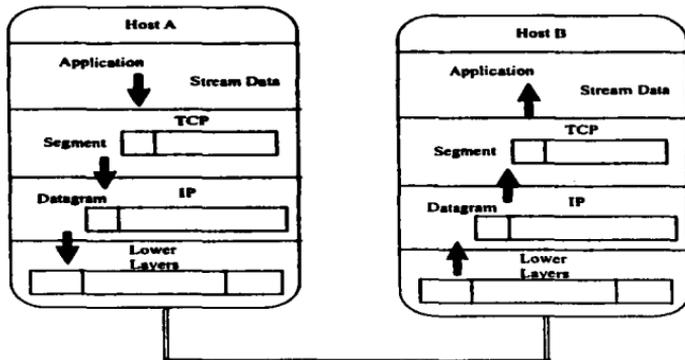
El servicio de aplicación del protocolo TCP/IP incluye servicios estándar para la comunicación programa a programa. Como FTP (File transfer Protocol), SMTP (Simple Mail transfer Protocol), Telnet (Terminal access) y DNS (Domain Name System) servicio de directorios.

TCP es implementado en los servicios de transmisión. La entidad de cada fin de conexión del protocolo TCP asegura que la transmisión cumplió con las siguientes funciones:

- * Transmisión correcta, exacta
- * De manera secuencial
- * Transmisión completa
- * Libre de paquetes duplicados

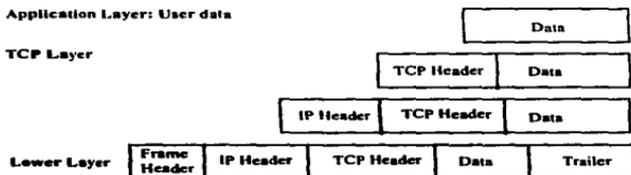
La aplicación transfiere un flujo de bytes a TCP. TCP rompe este flujo en piezas y adiciona un encabezado en cada pieza, formando segmentos, en este momento TCP pasa los segmentos a IP para transmitirlos en un datagrama.

Fig. 5 Flujo de los paquetes en el protocolo TCP/IP



Recibe TCP teniendo que guardar la información de si llegaron los datos de manera correcta, esto es un reconocimiento. (ACKs). Si uno de los segmentos no es recibido en un período de tiempo determinado, TCP realiza nuevamente el envío del segmento. Esta estrategia realizada se conoce como retransmisión con reconocimiento positivo. Ocasionalmente una retransmisión puede causar duplicidad de segmentos en la entrega para TCP. Cuando TCP recibe los segmentos los coloca de manera ordenada, descartando la posible duplicidad que pueda darse, ya que los encabezados cuentan con un número de secuencia la cual es identificado en la función de reconocimiento, así TCP libera los datos a la capa de aplicación en orden sin faltar piezas en la información.

Fig 6 Encabezados de la unidad de datos.



La unidad de datos que se encuentra en la capa más baja del protocolo es llamada : frame. Un encabezado de frame contiene campos que identifican los dispositivos físicos del origen así como del destino. Así como también contienen un frame check sequence conteniendo el resultado matemático que es utilizado para detectar errores de transmisión. Este resultado es comparado con el valor del trailer garantizando la integridad de los datos, que sean recibidos íntegramente.

TCP se comporta como un protocolo full duplex, es decir puede enviar y recibir en ambos sentidos en una conexión al mismo tiempo existiendo dos flujos de bytes en la transmisión.

Capitulo 3

Características para la Transmisión de datos

3.1 Velocidad de transmisión

En un circuito de transmisión de datos podemos medir, por un lado, la velocidad con la que los datos son transmitidos, y por otro lado la capacidad de transmisión que tienen los medios físicos o lógicos utilizados.

En cualquier terminal de transmisión de datos se pueden diferenciar tres partes: la fuente, el colector y el controlador de la comunicación. La fuente de la información es el lugar del terminal emisor donde está almacenada la información original que se pretende transmitir, el colector es el lugar del terminal receptor donde se guardarán los datos que se van recibiendo y el controlador de la comunicación es el programa encargado de llevar a cabo la transmisión, este software existe tanto en el terminal emisor como en el receptor.

Los controladores de la comunicación de los terminales emisor y receptor, como su propio nombre indica, son responsables de que la transmisión se realice de una forma eficaz y sin que haya errores.

En general la velocidad de transmisión de información se mide por el número de bits transmitidos por segundo o bps.

3.2 Capacidad de transmisión

La transmisión de datos siempre se basa en la existencia de una línea que une el terminal emisor con el receptor. Dicha línea puede ser, desde un simple cable que une dos computadoras cercanos hasta un medio de transmisión público, como puede ser la red telefónica o los circuitos alquilados punto a punto, o también conexiones vía microondas basadas en satélites.

Todos los medios de transmisión que envían señales a largas distancias utilizan equipos especiales de transmisión para asegurar que las señales introducidas por un extremo lleguen al otro en buenas condiciones.

Anteriormente los medios de transmisión eran analógicos, es decir estaban diseñados para transmitir únicamente señales analógicas por ejemplo una línea telefónica, para medir la capacidad de transmisión en este medio se utiliza el ancho de banda, esta es la diferencia entre las frecuencia mayor y la frecuencia menor que puede ser transmitida por un medio de comunicación, en el caso de líneas telefónica, el ancho de banda es de $3400-300=3100$ Hz. En la actualidad la mayoría de los nuevos sistemas de transmisión que se instalan son medios digitales. Un medio de transmisión digital se caracteriza porque recibe información digital binaria en un extremo y la transmite al otro extremo. El típico medio de la transmisión digital es la fibra óptica, sin embargo también existe transmisión digital vía radio, cable o satélite. La capacidad de un medio de transmisión digital se mide en bits por segundo. En el caso de la fibra óptica son muy normales capacidades de 140 o 512 Megabits por segundo (Mbps).

A medida que las necesidades de comunicación entre computadoras crecieron, se fueron desarrollando mecanismos para perfeccionar esa inter-conexión en especial en el área de los canales de comunicación que se usaba, para hacer más eficiente el envío de datos.

Lo primero en desarrollarse fue la conmutación de mensajes. La conmutación de mensajes es una tecnología que permite guardar la información, para enviarla posteriormente. La computadora a cargo examina la dirección en la cabecera del mensaje y lo conmuta (encamina) hacia la otra computadora que debe de recibirlo, con el tiempo se trató de mejorar la estructura de la conmutación de datos, las nuevas tecnologías dieron como resultado la conmutación de paquetes.

La conmutación de paquetes se conoce con este nombre, porque los datos se descomponen en trozos más pequeños ,estos trozos llamados paquetes, se envían por la red como entidades independientes ya que a cada paquete se le añade un información adicional al comienzo del mismo en la cabecera para que cada paquete se mueva por la red de manera independiente. Si en un momento dado, una ruta o un nodo de comunicaciones queda fuera de servicio, los paquetes que en un principio utilizan esos medios son enviados de forma automática por otras ruta, sin que quede interrumpida la comunicación. La conmutación de

paquetes se desarrolló inicialmente por el Departamento de Defensa de los E.U. por los años 70's con el fin de conseguir conmutar paquetes de conversaciones vocales. La idea era que si se dividía una conversación en fragmentos, podría enviarse por diferentes rutas. De este modo el enemigo, al interceptar la comunicación, captaría solo una mínima parte de todo el mensaje. La necesidad de contar con una red que garantizara la transmisión de los datos libres de errores da la pauta para la creación de una red de conmutación de paquetes; con esto se consigue una utilización más eficiente de la red, evitándose las saturaciones, disminuyendo los tiempos de establecimiento de la comunicación, aumentando la velocidad máxima disponible y mejorando consecuentemente la calidad de la comunicación.

3.3 Sincronización de la comunicación

Los procesos más importantes que intervienen en la transmisión de datos, es la coordinación de la transmisión y la recepción de los datos. Son tres los factores a tener en cuenta:

1. El primero es que los bits son enviados por la terminal origen de forma secuencial y con cierta cadencia. Si la terminal de destino tiene un mínimo error en la cadencia de lectura, puede llegar a leer un mismo bit 2 veces o saltarse algún bit sin leer.
2. El segundo factor es que la terminal receptora recibe los bits uno tras otro, por lo que debe tener algún procedimiento para diferenciar cada uno de los caracteres o bytes que componen la información transmitida.
3. Cuando se tienen que transmitir grandes volúmenes de información, esta no se transmite toda de una vez, ya que eso provocaría que de haber un error se tendría que retransmitir todo desde un principio, para evitar esto la información se divide en secciones más pequeñas, llamadas tramas, bloques o paquetes, esto quiere decir que hay que establecer un procedimiento que nos permita identificar qué carácter, de todos los recibido, es el primero de cada trama.

Transmisión asincrónica

La búsqueda de mayores velocidades de transmisión llevó a los diferentes diseñadores de sistemas de comunicación de datos a idear un sistema que aprovecharse mejor el tiempo de lo que lo hace el sistema asincrónico. La idea es producir dispositivos que envíen el máximo posible de bits por unidad de tiempo utilizando un mismo canal de comunicación. El resultado fue el sistema síncrono.

Con el sistema síncrono, la base de tiempo que genera la terminal emisor para transmitir de datos es recogida por el terminal receptor a partir de los propios cambios de estado de los datos recibidos. Esto es, la sincronización se lleva a cabo utilizando para ello los mismos cambios de estado en las señales transmitidas.

Para asegurar la sincronización, antes de empezar a transmitir los datos de información, el terminal emisor transmite uno o más caracteres de sincronización llamados SYN. Estos caracteres están formados por una combinación de 0 y 1 alternos (0101010). Por otro lado, los bytes de información son enviados agrupados en tramas, de forma que después de cada trama se envía de nuevo el carácter SYN.

La ventaja de los sistemas síncronos es que no se desperdicia tiempo en realizar el sincronismo, como ocurre con los sistemas asíncronos.

Características de las técnicas de transmisión

Síncrono

- La transmisión de datos es precedida por una serie de caracteres de sincronismo
- Entre cada trama o bloque de datos se vuelven a transmitir los caracteres de sincronismo para mantener la sincronización.
- Entre cada carácter y carácter no existe ningún periodo de inactividad
- La terminal emisora genera la señal de reloj que fija la cadencia de envío de los bits utilizando la terminal receptora la misma referencia
- Las terminales deben disponer de un memoria intermedia (buffer)
- Este sistema se suele utilizar para velocidades de transmisión iguales o superiores a los 2400 bits por segundo.

3.4 Compresión de datos

La compresión de datos es una técnica que permite transmitir más información en un mismo número de bits. Los modems que utilizan esta técnica comprime la información antes de transmitirla, de forma que eliminando los caracteres repetidos y siguiendo un algoritmo adecuado, la información puede ser reproducida exactamente igual a la original en el destino, pero habiendo transmitido de forma real mucho menos bits de los que originalmente se componía la información.

Los modems con sistemas de compresión de información sólo pueden sacar provecho de él si al otro extremo disponen de un módem con las mismas características. Por otro lado, la eficacia de los sistemas de compresión depende en gran medida de la información a transmitir. La información de texto puede verse reducida en un 25% de su tamaño original, y por lo tanto puede tardarse la cuarta parte del tiempo en su transmisión.

3.5 Enlaces de Comunicación

Los enlaces de comunicación están diseñados para satisfacer los requerimientos de algún sistema en particular. El enlace más sencillo es aquel que contiene una estación transmisora que es la encargada de establecer la comunicación y controlarla mientras la estación receptora, tan sólo se dedica a recibir la información y contesta a los requerimientos de la primera. Las dos estaciones debe usar el mismo protocolo, tener la misma capacidad para el manejo de datos y tener el mismo código de datos para establecer un buen enlace. Los métodos actuales de enviar y recibir información están divididos en tres :

Simplex (unidireccional) solo establece comunicación en un sentido. Este tipo de enlace es útil cuando es necesario transmitir grandes volúmenes de información a la estación receptora sin requerir de ésta ninguna señal de respuesta.

Half duplex (semidireccional). Estos enlaces permiten la transmisión de información en ambos sentidos, del transmisor al receptor y viceversa pero restringe esta transferencia a sólo un sentido a la vez. Hasta que la estación emisora haya terminado de enviar su mensaje la estación receptora podrá enviar la respuesta.

Full duplex completamente bidireccional, permiten la transmisión simultánea de información en ambos sentidos utilizando el mismo medio de transmisión al mismo tiempo.

Como trabaja Connect direct

De acuerdo a estas características de transmisión de datos, CONNECT:direct permite realizar la transmisión a través de líneas directas que transmiten a una velocidad de 10 Mbps estas líneas son: la línea primaria es de Avantel un enlace T1 a 1.5 Mbps. y la línea de contingencia es de Telmex es un enlace E1 con 2Mbps.

Los archivos que son enviados a través de la red son segmentados en varios paquetes que utilizan el protocolo TCP/IP y además son encriptados para proveer la seguridad de esta información, el software que se maneja no permite dar a conocer el método de encriptación pero a través de algunas pruebas realizadas por la empresa que distribuye este software en México (Grupo Pissa) la información transmitida no se puede decifrar, escuchando en la red no se comprende la información y además por eso se particionan en paquetes. La capacidad de transmisión según los parámetros permitidos para envío el valor máximo es de 2 Gigabytes y el valor por default es de 64 K que es el tamaño como se está mandando esta información a Procesar. Este software me permite considerar ciertos parámetros de compresión de la información, cuando los archivos recibidos o enviados son de tamaño grande como más de 60 Mb. se considera este parámetro para su uso permitiendo que el tiempo de transmisión o recepción sea más rápido.

Otro de los parámetros que Connect direct me permite utilizar es la función checkpoint, estos son puntos de inspección que marcan en los encabezados de los paquetes de ambos nodos (destino y origen) utilizando valores numericos en kbytes o Megabytes. Esto se utiliza para evitar perdida de la información por posibles problemas en la comunicación en cualquier parte de la transmisión. Conforme a esto en el momento de la recuperación del error la transmisión prosigue a partir del último checkpoint, sin necesidad de retrasmisión.

Por otra parte el enlace de transmisión que se utiliza en Afore Santander Mexicano es Full-duplex, porque se tiene la disponibilidad de poder realizar diversas transmisión o recepciones en ambas direcciones utilizando el mismo medio.

La manera de como podemos verificar que el nodo remoto nos recibió la información enviada es a través de las estadísticas generadas por Connect:direct de la siguiente manera:

SELECT STATISTICS

PROCESS RECORD Record Id => CTRC

Process Name => proc1 Stat Log Date => 10/10/1997
 Process Number => 6325 Stat Log Time => 19:10:04
 Submitter Class =>
 Submitter Id => cdirect@AFOSAME.UNIX

Step Start Date => 10/10/1997 Start Time => 19:09:46
 Step Stop Date => 10/10/1997 Step Stop Time => 19:10:04
 Step Elapsed Time=> 00:00:18

Step Name => step1
 SNODE => INSCONS.UNIX
 Completion Code => 0
 Message Id => SCPA0001
Short Text => Copy operation successful.
 Ckpt=>Y Lkfl=>N Rstr=>N Xlat=>N Scmp=>N Ecmp=>N
 Local node => P
 From node => P
 Src File => /afs/gfi/prod/afore/consar/envios/971010/19971010.080
 Dest File => /export/home/rec/AFORES/A540/RECEPCION/19971010.080
 Source Destination
 Ccode =>0 Ccode =>0
 Msgid =>SCPA0001 Msgid =>SCPA0001
 Bytes Read =>125378 Bytes Written=>125378
 Recs Read =>6 Recs Written=>6
 Bytes Sent =>125542 Bytes Recvd =>125542
 Rus Sent =>33 Rus Recvd =>33

Select Statistics Completed Successfully.
 Direct>
 CONNECT:Direct CLI Terminated...

Capitulo 4

Instalación de Connect direct

A través de los capítulos anteriores podemos entender el funcionamiento y toda la estructura básica de lo que comprende la transmisión y recepción de información, esto con el fin de cubrir la necesidad de compartir información sin restricciones de áreas geográficas. Este software Connect:direct for Unix versión 1.4.00 Ga se instaló en un servidor HP9000, modelo K460 que opera con el sistema operativo HP-UX versión 10.20 y teniendo un espacio físico para la instalación de este software de 25 Mb. y un espacio de 2.5 Gb de memoria. El tipo de red es Ethernet y se utiliza el protocolo de comunicación TCP/IP. Fig. 7.

4.1 Preparación para la instalación

Tipos de Configuración

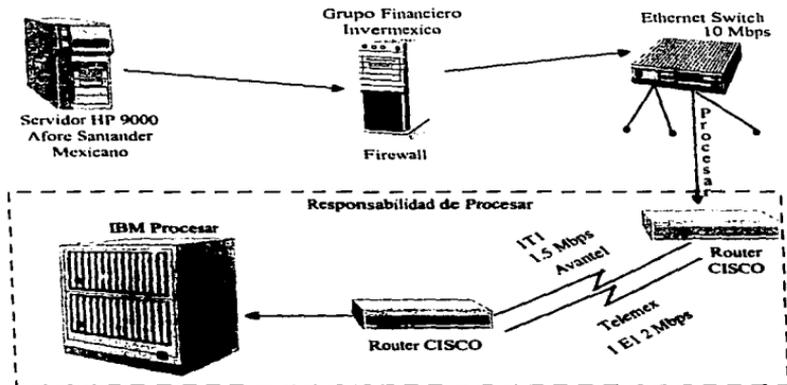
Connect Direct puede instalarse de dos diferentes formas de configuración, las cuales son las siguientes:

1. Instalación del servicio en una sistema local y el cliente (CLI/API) en un sistema remoto
2. Instalación del servidor y al menos un cliente (CLI/API) en un sistema local y el CLI/API permanecer en reserva en un sistema remoto.

Interfaces soportadas:

Ethernet
Token ring
SDLC
X.25

Fig. 7 Ambiente de Producción de Afore Santander Mexicano



4.2 Instalación de Connect direct para UNIX

Paso numero 1

Antes de comenzar la instalación es necesario crear un usuario o una cuenta en el sistema UNIX que tenga los privilegios requeridos para la instalación. No se debe instalar el software con el usuario root. En este caso especial la cuenta del usuario es cdirect.

Paso numero 2

Montar el CD-ROM en el driver apropiado para poder leerse

Paso numero 3

Una vez que fue montado el CD-ROM en el directorio correspondiente en este caso se encuentra listo en :

```
cpio -civdmB < /cdrom/opt/cdirect/ cdinst
```

Paso numero 4

Después de haber ejecutado el script de instalación, se teclea el siguiente comando.
cdinst

Los siguientes mensajes que se despliegan en el comienzo del procedimiento de instalación están relacionadas a preguntas como, especificar el directorio destino donde será almacenado el producto Connect direct. Después de que los archivos han sido cargados al directorio del servidor automáticamente se realiza la llamada del script de configuración de Connect direct para que opere en un ambiente UNIX .

En las preguntas que se derivan de la instalación del producto, se debe de contestar de la siguiente manera:

Para responder a las preguntas cuando lo indica el prompt se utilizan las letras y que quiere decir si, con **n** que indica no y **y** que implica todo. Las posibles respuestas son indicadas con corchetes y el valor por default es presentado en mayúscula. Si sólo se oprime RETURN se acepta el valor por default.

No se deben usar : para valores que se pidan. Para continuar con la instalación es necesario teclear RETURN.

Se puede terminar el proceso en el momento que se desee con sólo presionar Ctrl-c. Si este es el caso el script de instalación despliega un mensaje notificando que la instalación del producto es incompleta, esto es que los archivos de configuración no están completos y no se podrá hacer uso del software, pero los archivos que ya se copiaron al subdirectorio existen y para iniciar nuevamente la instalación hay que borrar esos archivos.

Sterling Commerce, Inc., TM Connect:direct TM for UNIX TM

Installation Procedure

You are beginning the Connect:direct for Unix Installation Procedure. You will be asked to specify a directory (called the destination directory) where the Connect:direct for Unix files will be stored.

After the files are extracted from the media, a customization procedure will be called. It will configure the Connect:direct for Unix operating environment for you.

Sterling Commerce, Inc., TM Connect:direct TM for UNIX TM are trademarks of Sterling Commerce, Inc., TM on the U.S.A. and the other countries.

UNIX is a trademark of UNIX Systems Laboratories, Inc.

Press RETURN when ready

Is the media drive attached to the local system: [Y/n]

Enter Device Name (e.g., /dev/rmt0, etc) of the drive: /dev/rmt0

*** Please insert the media if you have not done so already.*

Paso numero 5

En este paso se verá el contenido de la sección de ejecución del script que es llamado automáticamente después de la carga del software en el servidor:

1. Se debe leer la información que es desplegada en pantalla y presionar RETURN para comenzar la instalación.

Press RETURN when ready

2. Se indicará si la instalación es de manera local

Is the media drive attached to the local system [Y/n]: Y

3. Por default el sistema es local, de lo contrario se pregunta el nombre del sistema remoto

Enter host name of the remote drive: n

4. El script pregunta cual es el nombre del device en donde se encuentra Connect direct para Unix.

Enter Device name of the drive: _/cdrom/opt/cdirect _
Place insert the media if you have not already done so.
Press RETURN when ready.

5. El siguiente paso es determinar el tipo de instalación ,si el cliente CLI/API ,o solo el servicio , o ambos. El default es ambos, en esta etapa tienes la opción de salir de la instalación.

The installing procedure allows you to install:

[1] Connect: direct for UNIX sever;

[2] Connect: direct for UNIX CLI/API

[3] Connect: direct for UNIX sever and CLI/API;

[4] Exit

Enter your choise [3]: 3

Cuando es seleccionada la opción 3, se manda un mensaje de confirmación para la instalación la parte del servidor y cliente

Both Connect:direct for Unix Server Version 1400 and CLI/API Version 1400 will be installed in your system. Do you want to continue: [Y/n] : Y

Install both CONNECT:direct for UNIX server version 1400 and CLI/API Version 1400 for the user cdirect

18012 kbytes of disk space are required to install Connect:direct for Unix server and CLI/API.

Si se oprime RETURN por default es Si de lo contrario si es No se sale del script de instalación.

6. En este momento comienza un análisis de espacio libre en el disco en específico sobre el file system en el cual se va a bajar la información., esto es necesario para el proceso de instalación. Unos segundos después se presenta un reporte de la capacidad de espacio libre que se tiene.

Here is the current free disk space for each disk partition:

```
=====
Filesystem      kbytes      used        avail        %used      Mounted on
/dev/vg00/lvol3 127573      21061       93754        18%        /
/dev/vg00/lvol1 47829       29021       14025        67%        /stand
/dev/vg00/lvol11 196608      93201       96330        49%        /var
/dev/vg00/lvol5 393216      259493      133723       65%        /opt
..... , etc
```

*You have 133723 k of disk space available to you,
which is more than the required 18012 k.*

Do you want to continue [Y/n]: Y

Si se cuenta con el espacio requerido se procede con la instalación con RETURN si se necesita reorganizar el espacio en disco se teclea n.

7. En la siguiente pregunta se solicita el directorio completo del lugar donde va a ser instalado el software.

Enter the full path of the destination directory to install Connect direct, You can use \$HOME to shorten the name: [\$HOME/cdunix]: /opt/cdirec

Cuando el path de direct es presentado es diferente del default, a partir del último directorio se crearán los directorios de cdirect sin perder su estructura, se puede presentar un

directorio destino diferente o usar el sugerido en este caso es /opt/cdirect claro que el directorio debe contar con privilegios de escritura para el uso de Connect direct creado en el pado número 1.

El script de instalación pide la confirmación de que el directorio local sea el correcto.

You have chosen path name as destination directory, Please confirm it: [Y/n]: Y

8. La instalación se realiza automáticamente a partir del punto que se le indicó en el párrafo anterior. Los archivos son copiados del la fuente al directorio destino. El script sólo verifica el numero de archivos y bloques que son copiados. La instalación continua con la ejecución de script consecutivos los cuales son invocados automáticamente del script de instalación.

Paso numero 6

Personalización de Connect direct para UNIX.

La ejecución del script es de manera automática ya que es invocado después del script de instalación y en el caso de este script de personalización se puede recibir ayuda para el ambiente de operación. Cuando el script de instalación es terminado exitosamente, genera un mensaje en el cual se informa que se ha iniciado la personalización del ambiente. Para llamar la ayuda en línea se debe introducir un carácter ? en el lugar donde se espera la opción deseada. La terminación de ejecución del script de personalización puede realizarse con Ctrl-c, en este caso la instalación es incompleta, no funciona el software.

1 .Para realizar la personalización de Connect direct es necesario llamar un script :

\$ \$HOME_CDIRECT/etc/cdcusr

Inmediatamente se despliega la información siguiente:

Sterling Commerce CONNECT:direct for UNIX Customization

The customization procedure sets up the CONNECT:Direct for UNIX operating environment. In the following, please answer questions when in doubt, enter (?) followed by RETURN for help. To abort the process, enter Control-c. As a short hand, entering RETURN at the prompt means "yes" or the fault value, if any will be taken.

Press RETURN when ready

Espera RETURN para continuar con el proceso de personalización.

2. El script de personalización solicita que se especifique el tipo de configuración que se debe de tener para los archivos de control.

The customization procedure allows you to create configuration files for:

- (1) CONNECT:Direct for UNIX server*
- (2) CONNECT:Direct for UNIX CLI/API*
- (3) CONNECT:Direct for UNIX server and CLI/API*
- (4) exit*

Enter you choice [3] : 3

Para este caso el numero es el 3.

La opción por default es la numero 3 que se personaliza para ambos (servidor y cliente).

Este script crea 6 archivos requeridos para la operación correcta del servidor, estos son los siguientes:

Archivo de parámetros de inicialización

Archivo de información de usuarios autorizados.

Archivo del Mapa de la red (Netmap)

Archivo de llave de autenticación del servidor.

Archivo de configuración del cliente CLI/API

Archivo de llave de autenticación del cliente.

Continuando con el proceso de personalización se pide el nombre del path del directorio destino:

Enter full of CONNECT:Direct destination directory: [\$HOME/cdunix]

En este caso en particular la ruta es /opt/cdirect

Después se solicita el nombre del nodo que se quiere personalizar.

Please enter name of CONNECT:Direct node you want to customize: Afore3.prosis

El nombre del nodo puede tener una longitud hasta de 16 caracteres.

En el caso de Afore Santander Mexicano se tuvo que realizar los siguientes pasos en particular ya que la personalización que se requiere es para servidor y cliente CLI/API en el mismo nodo.

Paso numero 6.A

Archivo de Parámetros de Inicialización

El proceso de CONNECT:Direct inicializa parámetros durante el comienzo de la corrida del proceso.

Algunos parámetros para este archivo provienen de la información del servidor. Después de esta parte del script de personalización CONNECT:Direct asigna valores por default al resto de los parámetros de inicialización. Algunos de estos parámetros son susceptibles a

posibles modificaciones después de la personalización esto se puede realizar a través de un editor de texto . Aquí se muestra una parte de la descripción de este archivo (Cuadro 3).

No. de línea	Contenido	Notas
1	#Diversos conceptos	# en la primera columna se indican los comentarios.
2	ndm.path:path=/ndm/users/c:	Nombre del registro record name=ndm.path parameter=path value=/ndm/users/c:
3	proc.prio:default=8	record name=proc.prio parameter=default value=8
4	asset.protection:key=cpu 1234 pid 1234 seq 1234 \	record name=asset.protection: parameter= key= value=cpu 1234 pid 1234 seq 1234 rix 1234 gbi 1234 pn(1234) sys 1234 teer 1234: Se puede continuar una línea lógica con el símbolo backslash(\)
5	rix 1234 gbi 1234 pn(1234)\	\ indica continuación de una línea lógica
6	sys 1234 teer 1234:	: indica la terminación de los valores
7	#local CONNECT:direct connection information	#Indica solo comentario
8	local.node\	record name=local.node
9	:api.max.connects=11:\	Numero máximo de conexiones parameter=api.max.connects value=11
10	:conn.retry.stwait=00.00.12:conn .retry.stattempts=2:\	Esta línea contiene dos parámetros: parameter=conn.retry.stwait value=00.00.12 parameter=conn.retry.stattempts value=2
11	:conn.retry.ltwait=00.01.01:conn .retry.ltattempts=4:\	Esta línea contiene dos parámetros: parameter=conn.retry.ltwait value=00.01.01 parameter=conn.retry.ltattempts value=4
12	
22	:tcp.api=rusty;391:\	parameter=tcp.api value=rusty;391
23	:tcp.api.bufsize=10240	parameter=tcp.api.bufsize value=10240

Si se tiene dos parámetros sobre el mismo renglón es necesario separarlos por (:), para la asignación del valor a cada parámetros se realiza a través de (=), si los valores son más de dos entonces sólo son separados por espacio en blanco y en último valor se termina con :. Este archivo de parámetros reside en /opt/cdirect/ndm/cfg/cd_node/initparm.cfg. En el próximo cuadro se muestra la descripción de los posibles parámetros. (Cuadro 4).

Parámetro	Descripción
ndm.path	Especifica el path completo de todo el software de CONNECT:direct
proc.prio	Especifica la prioridad del proceso, esta prioridad por default es 10 conociendo que el rango es de 1 a 15
asset.protection	Especifica la información referente a las llaves para poder utilizar CONNECT:direct de manera adecuada
local.node	Define la información por default del software CONNECT:direct para poder ser usado en la transmisión entre nodos remotos.
api.max.connects	Especifica en número máximo de conexiones concurrentes permitidas hacia el nodo local. El número por default es 16
conn.retry.stwait	Especifica el tiempo de espera entre un inmediato reintento de transmisión a partir de una falla ocurrida. Los parámetros son hh.mm.ss el valor máximo permitido es 23:59:59 y el valor por default es 00:00:30 , 30 segundos.
conn.retry.stattempts	Especifica el número de veces que se va a reintentar la transmisión después de haberse detectado una falla. El valor máximo es un número de cuatro dígitos, y el valor por default es 6
contact.name	Especifica el nombre del administrador u operador de CONNECT:direct, y es longitud no limitada.
contact.phone	Número de teléfono del administrador u operador de CONNECT:direct
descrip	Apartado para la descripción o comentarios
name	En este parámetro se almacena el nombre del nodo local de CONNECT:direct, este parámetro es indispensable. La longitud de este registro es hasta 16 caracteres alfanuméricos
sess.total	Especifica el número máximo de sesiones concurrentes entre todos los nodos y el nodo local. Este tiene un valor por default de 64 y el máximo es un valor de 3 dígitos numéricos
sess.pnode.max	Especifica el número máximo de conexiones concurrentes en donde el nodo local es el iniciador de la sesión

sess.nnode.max	Especifica el numero máximo de conexiones concurrentes en donde el nodo local es el nodo remoto en la sesión. Nota el numero de sess.pnode.max y ses.nnode.max no debe exeder el numero de sess.total.
comm.outsize	Es el tamaño del buffer para transmitir los datos hacia un nodo remoto. Por default es de 4096 bytes, pero depende del sistema.
tcp.api	Especifica la información necesaria para que la comunicación se establezca entre el CLI/API usando TCP/IP. El parámetro hostname requiere en nombre de la maquina donde se encuentra CONNECT:direct como nodo local o la dirección ip nnn.nnn.nnn.nnn . El siguiente parámetro es port, identifica el numero de puerto de comunicación para CONNECT:direct para UNIX, el formato es nnn con números decimales.
tcp.max.time.to.wait	Cuando es usando TCP/IP este parámetro es el valor máximo de tiempo de espera del nodo local para escuchar un mensaje completo del nodo remoto. Este parámetro no tiene un valor por default.
pacing.send.dela y	Especifica el intervalo de tiempo de espera entre la operación de envío hacia el nodo remoto. El formato en nnn... dado en milisegundos, esto es el tiempo de espera a partir del final del paquete y el comienzo del próximo paquete. El valor por default es cero.
pacing.send.cou nt	Especifica el numero de operaciones que se envían en espera de la respuesta del nodo remoto. El valor por default es cero.
netmap.check	Provee de una mejor seguridad porque checa solo el nodo secundario (SNODE) Para las conexiones con TCP/IP la dirección remota IP que llega al socket es comparada con el parámetro comm.info del archivo netmap.cfg, este valor debe ser igual para que se puede establecer la comunicación De lo contrario no se podrá establecer el enlace.
Rnode.listen	Es el registro que contiene los parámetros usados por el nodo local para escuchar la conexión que llega.
Recid	Es el numero de identificación del registro rnode.listen. Parámetro no limitado en su valor.
Comm.info	Especifica la información necesaria para escuchar la petición de conexión desde el nodo remoto usando TCP/IP, este valor es requerido.
Comm.transport	Especifica el protocolo de transporte para el nodo remoto, acepta los siguientes valores tep para TCP/IP, lu62 para AIX SNA LU6.2 o blklu62 para otras conexiones LU6.2
max.age	Especifica el numero máximo de días que un proceso con estatus de error permanezca en le TCQ (Transmission Control Qecu) antes de ser borrado automáticamente, el default es de 8 días, el rango máximo está limitado a 3 dígitos numéricos.

Ckpt.interval	Especifica el intervalo del punto de transmisión en una operación de copia. El máximo valor posible de 2 gigabytes, pero el valor normal es de 64K. Las opciones que se tienen para este parámetros en n que no especifica el checkpoint, nnnnnnnnK número denotado en Kbytes y nnnnM número denotado en Megabytes.
Restart	Especifica la restauración de la tarea para que inicie en el paso en que se quedo antes de la falla, la opción por default es y.
File.size	Especifica el número máximo en bytes para la estadística individual del archivo enviado. El nombre para el archivo de estadísticas tiene el siguiente formato Syyymmdd.ext, donde yyyy indica el año, mm el mes y el dd indica el día. La extensión de estos archivos comienzan con un consecutivo de tres dígitos 001 y se va incrementado en una unidad.
Log.select	Especifica si se quiere registrar en un archivo log la salida de los comandos select process y select statistics, opciones para este parámetro y o n. La opción por default es n
server.program	Especifica el programa del servidor usado durante el procedimiento de autenticación. El valor por default ndmauths.
Server.keyfile	Especifica la llave del archivo usado durante el procedimiento de autenticación. El valor por default es keys.server

(Cuadro 4).

1. El script de personalización requiere de la llave de protección que es tan importante.

Please enter Asset Protection Key information at the following prompts.

CPU: xxxx
 PID: xxxx
 SEQ: xxxx
 RLX: xxxx
 GBL: xxxx
 PN : xxxx
 SYS: xxxx
 TEER: xxxx

Después de haber introducido la llave de protección presionar RETURN, esta llave es almacenada en el archivo de parámetros de inicialización en key parameter **asset.protection.record**.

Y nuevamente se solicita la verificación de la llave de protección que fue introducida.

Please make sure the asset protection key is entered correctly, Does it match with was shipped with the product ? [Y/n]: Y

Si la lleve de protección es incorrecta teclear n y se desplegará la ventana anterior para la corrección.

3. Continuando con la personalización de Connect:direct el script solicita el número de puerto de TCP/IP a través del cual se va a establecer la conexión con el nodo remoto. Es el puerto por donde se oye la petición del nodo remoto.

Enter TCP/IP port number to listen for a remote CONNECT:direct connection request [1364] : 1364

Esta información que se requiere es muy necesaria para el establecimiento de la interconexión con el nodo remoto. El valor por default es 1364, se recomienda utilizar este numero de puerto pero en caso de que este puerto sea utilizado por otro servicio no existe ningún problema para que se asigne otro puerto disponible. Se puede checar la lista de puertos en el archivo /etc/services. Esta información también es almacenada en el archivo de parámetros en el parámetro *comm.info*

4. Uno de los parámetros que también se necesitan en el archivo de parámetros es el numero de puerto para que el servidor escuche la requisición de los APIs.

Enter TCP/IP port number to listen for an API connection request [1363] : 1363

El valor por default es el 1363 que es el recomendado, pero se puede utilizar algún otro puerto disponible., este parámetros se almacena en el archivo de inicialización en tcp.api

Paso numero 6.B

Definir el archivo de información de usuarios autorizados.

Este archivo contiene la información de los usuarios locales o remotos que estén contemplados para la utilización de este servicio de transmisión. A cada registro del usuario local se le asigna un identificador y por cada usuario remoto se registra con el identificador que se le asignó localmente. Los usuarios que se deseen colocarse en este archivo es dependiendo las necesidades de la empresa. Después de la personalización se pueden modificar algunos de los parámetros editando el archivo de información de usuarios, los parámetros restantes Connect direct le asigna valores por default.

1. Se puede crear un registro para un usuario remoto

Insert a remote user record ? [Y/n]: Y

La opción por default es si y después es llenar la información necesaria para el registro. Si no se requiere dar de alta un usuario remoto entonces se solicita la información para el registro de un usuario local.

2. Si la opción fue y entonces se solicita el nombre de identificación del usuario remoto

Enter remote userid: cdirect

3. Inmediatamente se solicita el nombre del nodo remoto

Enter remote CONNECT:direct node name: INSCONS.UNIX

4. Para el registro de un usuario local es muy similar, se solicita el nombre del usuario

Enter local userid: _____

Este parámetro puede tomarse del usuario local del sistema UNIX. En este caso `cdirect`, es guardado en el parámetro `local.id`, en este caso por seguridad el usuario remoto utiliza este archivo para verificar el usuario local este autorizado para establecer la conexión y de la misma forma cuando el usuario remoto pretende establecer el intercambio de información identifica al usuario local en este archivo.

En caso de que se requiera el script de personalización solicita el registro de otro usuario remoto o local y se repiten los mismos pasos.

5. Se establece la posibilidad de poder otorgar privilegios de administrador al usuario local.

Grant administrative authority [Y/n] : Y

Este valor es almacenado en el parámetro `admin.auth` en el registro del usuario local, esto le permite a este usuario tener la todas las capacidades de un administrador, se puede introducir ? para detallar este tipo de autorizaciones que se otorgan al usuario. Los privilegios de administrador se le pueden otorgar a varios usuarios locales que se quieran claro que deben de estar ya registrados como usuarios locales.

Insert another local user record [Y/n]: Y

Paso numero 6.C

Creando el archivo de mapa de la red.

El network map es el archivo que contiene la información referente a la red y este es usando por `CONNECT:direct`, esta información describe el nodo local y otros nodos a los cuales se puede acceder a través de `Connect :direct` y puedan comunicarse , en este archivo se puede definir la información del nodo remoto por registro.

Cuadro de parámetros para la configuración de netmap. (Cuadro 5).

Parámetro	Descripción
node.name	Especifica el nombre del nodo remoto. Este valor no debe exceder de 16 caracteres.
Client.program	Especifica el programa que usa el cliente durante el procedimiento de autenticación. El programa por default es ndmauthc
client.keyfile	Especifica el archivo llave usado durante el procedimiento de autenticación. EL archivo por default es keys.client

(Cuadro 5).

En el script de instalación automáticamente genera el archivo network map con la información del nodo local y además contiene la información referente a la configuración de instalación del nodo.

Después de que el archivo es generado CONNECT:direct despliega el siguiente mensaje:

```
The Network Map is located at
  $ HOME_CDIRECT/ndm/cfg/netmap.cfg
  /opt/cdirect/ndm/cfg/netmap.cfg
```

Paso numero 6,D

Creación de un archivo de llave de autenticación a nivel servidor para verificar las peticiones de conexión, solo a los APIs autorizados se les otorga el privilegio de conexión.

CONNECT:direct genera automáticamente el archivo para la llave de automatización del servidor, después de que el archivo es generado CONNECT:direct despliega el siguiente mensaje:

```
A server authentication key file has been created. The default key is XXXXXXXX.
```

Paso numero 6.E

Creación del archivo de configuración de CLI/API

El archivo de configuración es creado por cada CLI/API, este archivo es necesario para levantar el proceso que se requiere para que el cliente trabaje. Después de haber completado esta parte de la personalización CONNECT:direct asigna los valores por default a los parámetros remanentes de este archivo. En este caso se cuenta con la opción de poder modificar los parámetros en cualquier momento.

1. Se confirma si se quiere utilizar las mismas llaves de protección que se otorgaron con opción a poder ser modificadas.

You entered the following Asset Protection Key last time.

```
CPU: xxxxx  
PID: xxxxx  
SEQ: xxxxx  
RLX: xxxxx  
GBL: xxxxx  
PN : xxxxx  
SYS: xxxxx  
TEER: xxxxx
```

Do you want to use the same key? [Y/n]: Y

Este parámetro de la llave es almacenada en el campo **asset.protection** en el archivo de parámetros, la llave de protección debe ser la correcta para que CONNECT:direct funcione apropiadamente y continua el script

2. En el siguiente paso se debe de indicar el numero de puerto del servidor el cual se va a configurar para el cliente CLI/API

Enter the port number of the CONNECT:direct for UNIX server this CLI/API will connect to [1363]: 1363

Este valor queda almacenado en el parámetro llamado **tcp.port** en el archivo de configuración de CLI/API

3. Se requiere conocer el nombre de host del servidor al cual se va conectar un API

Enter the host name of the CONNECT:direct for UNIX server this API will connect to: Afore3.prosis.bsantander.com.mx

Este valor queda almacenado en el parámetro llamado **tcp.hostname** en el archivo de configuración de CLI/API. A continuación se describen los parámetros de la configuración para el CLI:

En la configuración de los parámetros para el CLI de CONNECT:direct se define el nombre del directorio que identifica la localización de los archivos usados para formatear la salida de los comandos **select statistics** y **select process**. El nombre del registro se llama **cli.parms**. Parámetros para el CLI (Cuadro 6).

Parámetro	Descripción
script.dir	Especifica el nombre del directorio de los archivos usados para la salida de los comandos select statistics y select process , el directorio por default es ndm/bin .
Client.program	Especifica el programa que usa el cliente durante el procedimiento de autenticación. El programa por default es ndmauthc .
client.keyfile	Especifica el archivo llave usado durante el procedimiento de autenticación. EL archivo por default es keys.client .

(Cuadro 6).

Paso numero 6.F**Creación del archivo llave de autentificación del cliente**

Este archivo es usado por CONNECT:direct para identificar el servicio que atenderá al cliente y podrá conectarse al nodo. Se pueden tener multiples entradas de multiples servidores.

CONNECT:direct genera este archivo automáticamente después de haberse generado este archivo se despliega el siguiente mensaje:

A client authentication key file has been created. The default key is XXXXX.

Paso numero 7

Cambio de propiedad y permisos en los archivos para los procesos de CONNECT:direct

Después de haber personalizado CONNECT:direct, es necesario cambiar los atributos de los archivos que son utilizados por los procesos de CONNECT:direct como el Sesión Manager, el Process Manager y el Command Manager esto es:

Para modificar los permisos y el dueño de los archivos es necesario estar conectado al servidor con el usuario root.

Después cambiarse al directorio `cd /opt/cdirect/ndm/bin` y cambiar de dueño o propiedad a los archivos siguientes:

```
chown root ndmsmgr cdpmgr ndmcmgr
chmod u+s ndmsmgr cdpmgr ndmcmgr
```

Paso numero 8

1. Verificar la instalación
2. Levantar el servicio de CONNECT:direct para UNIX

Para levantar este proceso se ejecuta:

```
/opt/cdirect/ndm/bin/cdpmgr -i /opt/cdirect/ndm/cfg/afore3/initparam.cfg
```

3. Iniciar el servicio del cliente CLI

Para invocar el cliente direct es:

```
/opt/cdirect/ndm/bin/direct
```

Y es necesario contar con las variables de ambiente NDMAPICFG

```
NDMAPICFG= /opt/cdirect/ndm/cfg/cliapi/ndmapi.cfg
```

```
export NDMAPICFG
```

4. Realizar un ejemplo de transmisión para el proceso TCQ

Para esto se puede contar con ejemplo de copia de un directorio cdelete.me al directorio home uno es el nodo primario y otro el nodo secundario esta copia se realiza en intervalos fijos de 5k y se utiliza la compresión en la información que se transmite.

Ejecutar direct

```
direct> submit file=sample.cd;
```

5. Monitorear el estatus en el proceso de TCQ

Para poder llevar a cabo esta actividad es necesarios interactuar con el CLI a través de algunos comandos que nos pueden explicar a través de un reporte que especificar el nombre del proceso, usuario, nodo destino, cola de espera y estatus que está sucediendo con la transmisión o copia del los archivos. Esta información puede ser presentada de manera resumida o detallada para esto es necesario utilizar el siguiente comando:

```
Direct > select process pnumber=1;
```

Este comando proporcionará un reporte resumido conteniendo la información del proceso numero 1.

6. Conocer el resultado del proceso

Una vez completado el proceso de copia se puede revisar la estadística de este a través de otro comando en la línea de direct que nos presenta parte de lo realizado por CONNECT:direct

```
Direct > select statistics pnumber=1;
```

Detener o dar de baja el servicio de CONNECT:direct para UNIX

De manera manual puede darse de baja el proceso de CONNECT:direct a través de la línea de comandos direct.

```
Direct > stop
```

El cliente termina automáticamente el proceso en el servidor. Pero no afecta a cualquier proceso pendiente que pueda estar corriendo o reintentando enviar o recibir algún archivo.

Estructura del directorio de CONNECT:direct en una instalación completa:

Lista de archivos de CONNECT:direct

La siguiente es una lista de los archivos que integran a CONNECT:direct para UNIX: (Cuadro 7).

Nombre del archivo	Tipo	Descripción
ndm/bin/ndmpmgr	Server	Script que invoca al Process Manager
ndm/bin/cdpmgr	Server	Archivo binario del Process Manager
ndm/bin/ndmsmgr	Server	Archivo binario del Session Manager
ndm/bin/ndmcmgr	Server	Archivo binario del Command Manager
ndm/bin/ndmcli	Client	Script que invoca la interface de línea de comandos <i>direct</i>
ndm/bin/direct	Client	Archivo binario de la interface de línea de comandos <i>direct</i>
ndm/bin/ndmproc	Client	Script para el comando select process

ndm/bin/ndmproc.awk	Client	Programa awk para el comando select process
ndm/bin/ndmstat	Client	Script para el comando select statistics
ndm/bin/ndmstat.awk	Client	Programa awk para el comando select statistics
ndm/bin/ndmauthc	Cliente	Programa de autenticación para el cliente
ndm/bin/ndmauths	Server	Programa de autenticación para el servidor
ndm/bin/ndmmsg	Ambos	Despacha mensajes
ndm/bin/ndmxit	Ambos	Programa de traducción
ndm/cfg/cd_node/initparm.cfg	Server	Archivo de inicialización de parámetros.
ndm/cfg/cd_node/netmap.cfg	Server	Archivo de Network map
ndm/cfg/cd_node/userfile.cfg	Server	Archivo de información de usuarios autorizados
ndm/cfg/cd_node/msgfile.cfg	Server	Archivo de mensajes
ndm/cfg/cliapi/ndmapi.cfg	Client	Archivo de configuración del cliente CLI
ndm/include/ndmapi.h	Client	Incluye encabezado de los archivos
ndm/lib/ndmapi.a	Client	Librería para API que puede ser ligada con programas de aplicación
ndm/man/ndmmsg.1	Ambos	Páginas del manual para la Administración de Mensajes.
ndm/man/ndmpmgr.1	Ambos	Páginas del manual para el Process manager
ndm/security/keys.client	Client	Archivo llave de autorización para el cliente
ndm/security/keys.server	Server	Archivo llave de autorización para el servidor
ndm/xlate/def_recv.xit	Server	Interpretación de la tabla donde se registra los archivos recibidos
ndm/xlate/def_send.xit	Server	Interpretación de la tabla donde se registra los archivos enviados
work	Server	Directorio de trabajo por default
work/cd_node	Server	Directorio de trabajo de un nodo en particular, conteniendo TCQ y los archivos de estadísticas
ndm/src	Ambos	Archivos de Script, ejemplos de procesos
etc/cdcust	Ambos	Scripts de configuración
etc/tlflst1.0	Server	Listado de los archivos
etc/svcflst1.0	Server	Listado de los archivos del servidor
etc/cliflst1.0	Client	Listado de los archivos del cliente.

(Cuadro 7).

Conclusiones

Considerando el amplio campo de la computación y de las tecnologías actuales, se manifiesta a través de este trabajo la necesidad del intercambio de información entre diversas plataformas y sistemas operativos, dando como resultado que este intercambio se ha realizado de manera eficiente para el desempeño de las funciones de operación de Afore Santander Mexicano.

El objetivo principal de este trabajo fue la instalación de Connect:direct for UNIX ver. 1.4.0.0 GA en un equipo de producción HF300 K460 que opera bajo un sistema operativo UNIX así como la administración de dicho software como parte determinante en el buen funcionamiento de la transportabilidad de datos.

Para la instalación realizada creo conveniente resaltar que ésta se realizó de manera sencilla en un tiempo no mayor a 2 horas y que al término de ésta ya se estaban realizando pruebas concretas. Además sí se requieren conocimientos adicionales para poder configurar de manera adecuada el software de Connect:direct, para ello considero importante leer detenidamente los manuales de instalación ya que la ayuda en línea proporcionada en la instalación no es realmente útil.

Como parte del funcionamiento del software, me permito criticar el motif que Connect:direct for unix trae consigo como parte de un ambiente gráfico, ya que este motif presenta deficiencias en su presentación, debe de correr en un cliente en un ambiente Windows y se necesita de más inversión para el equipo que se utilizará para envío y recepción de archivos, otra parte de esta presentación que para mi punto de vista es un poco complicada la generación de scripts ya que presenta varias pantallas y se necesita un editor de texto posteriores modificaciones, claro que esto no desmerita la capacidad y excelente funcionamiento de este software para mi caso en particular no fue instalado el motif para la operación de Afore Santander Mexicano.

**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

En la parte de administración de este software es necesario estar pendiente del cambio de password de entrada ya que solo tiene una duración de 15 días el cual se está renovando constantemente, esto lo considero muy importante como parte de la administración del sistema ya que para la recuperación o envío de archivos, tiene que realizarse en el tiempo establecido por Procesar.

También se observó que la generación de script para la recuperación y envío de archivos se realiza de manera sencilla considerando la estructura secuencial de un script de Connect:direct, y manipulando los diferentes parámetros que pueden ocuparse para la obtención de archivos en diversos formatos. Connec:direct for Unix al instalarse genera un directorio con ejemplos de script que pueden ser ocupados como modelo para los propios. Otros de los puntos interesantes para la operación de Connec:direct for Unix como instrumento de transportabilidad de datos es la generación de estadísticas de cada uno de los procesos enviados o recibidos. Esta parte debe conocerse más a fondo, porque es sencilla cuando se requiere información no de manera detallada sin embargo cuando se presenta algunos errores para conocer más detalladamente esta situación es necesario conocer ciertos comandos para interactuar con CLI.

Dado que el sistema presenta una buena estabilidad en su configuración, esto me permite considerar que las próximas actualizaciones podrán ser realizadas de manera sencilla y no obstruirán la operación del sistema de intercambio.

Bibliografía

REDES DE COMPUTADORAS: Protocolos, normas e interfaces
Black, Uyles
Madrid 1987
Editorial: Macrobit
421 p.

TCP/IP: Architecture, Protocols and Implementation.
Sidnie, Feit.
E.U. 1993
Editorial: McGraw Hill
466 p.

Comunicaciones y Redes de procesamientos de datos.
González Sainz, Nestor
México 1993
Editorial: McGraw Hill
396 p.

REDES DE COMUNICACIONES
Huidabro, José Manuel
Madrid 1992
330 p.

EL LIBRO DE LAS COMUNICACIONES DEL PC
Carballar, José A.
México 1997
Editorial: Alfaomega grupo editor
743 p.

DATA COMMUNICATIONS, COMPUTER NETWORKS AND OPEN SYSTEMS
Halsall, Fred
U.S.A 1992
Editorial: Addison-Wesley
772 p.

REDES DE COMPUTADORAS: Aspectos técnicos y operacionales
Menasce, Daniel A.
Madrid 1988
168 p.

DISTRIBUTED PROCESSING AND DATA COMMUNICATIONS
Mcglynn, Daniel R.
New York 1978
305p.

COMPUTER NETWORKS AND THEIR PROTOCOLS
Chichester, J. Wiley
478 p.

TODO ACERCA DE REDES DE COMPUTACION
Stoltz, Kevin
Editorial: Prentice Hall
México 1995
518 p.

Manuales de Connect:direct

Connect_direct for UNIX. Installation and Administration Guide.

Connect:direct Process Guide Concepts and Examples. Volume 1

Connect_direct for UNIX. User's Guide.

Por Sterling Commerce versión 1.4 June 1996

First Edition

Direcciones electrónicas consultadas:

<http://ww.csg.stercomm.com>