TRANSZPORT FELÜLET

Bevezetés

A transzport felület, vagy más néven Transport-Level Interface (TLI), "C" nyelvű függvények halmaza, amely lehetővé teszi, hogy transzport-független hálózati alkalmazásokat fejlesszünk ki. A TLI és a socket interface általában standard hálózati interfészek a UNIX alapú operációs rendszerekben.

A transzport szolgáltatást nyújtó protokoll hajtja végre a TLI függvényeket. Ezeknek a függvényeknek a végrehajtását a transzport szolgáltatást felhasználó kezdeményezi megfelelő szolgáltatási kérések formájában.

A TLI-nek a következő két komponense van:

1. TLI könytári rutinok, amlyek az egyes szolgáltatásokat végrehajtják.
2. Állapotátmeneti szabályok, amelyek meghatározzák a transzport rutinok lehetséges hívási sorozatát.

Táblázatok definiálják a lehetséges állapotátmeneteket a potenciális események függvényében. Ilyen esemény lehet valamely, a felhasználó által kezdeményezett kérést, illetve választ megfogalmazó függvényhívás sikeres végrahajtása (**kimenő esemény**), vagy a szolgáltatást nyújtó felől érkező jelzések, illetve megerősítések átvételével kapcsolatos függvényhívások sikeres végrehajtása (**bejövő esemény**).

A TLI felhasználójának értenie kell az állapotátmeneteket, mielőtt olyan szoftver elkészítéséhez fogna, amelyik a transzport felületre épül.

Szolgáltatási módok

A TLI lehetséges szolgáltatási módjai a következők:

1. kapcsolatorientált szolgáltatás, és
2. kapcsolat nélküli szolgáltatás.

Kapcsolatorientált szolgáltatás

A kapcsolatorientált szolgáltatásnak négy fázisa van:

- lokális kezelés,
- kapcsolatfelépítés,
- adatátvitel,
- kapcsolatbontás.

Kapcsolatorientált szolgáltatás esetén az adatok sorrendtartóan továbbítódnak egy a végfelhasználók között már létrehozott adatátviteli kapcsolaton. A adatátviteli kapcsolat létrehozása után már nincs szükség további címkezelésre az adatátviteli fázisban.

A lokális kezelés elemei

A lokális kezelés fázisa a lokális műveleteket foglalja magába a szolgáltatást nyújtó és a szolgáltatást használó között. Ebben a munkafázisban például a felhasználónak kommunikációs csatornát kell létrehoznia a transzport szolgáltatóval. Minden ilyen csatorna egy meghatározott kommunikáció végpontja, és ezért transzport végpontnak is

Typeset by $\mathcal{AMS}$-TEX

szokás nevezni. A **t_open** függvény lehetővé teszi, hogy a felhasználó kiválassza a megfelelő transzport protokollt, amely alkalmas a kívánt kapcsolatorientált szolgáltatásra, és egyben megalapítja a transzport végpontot is. A transzport végpont szemléletesen a transzport felületen helyezkedik el, és szükségképpen valamely transzport rétegbeli entitáshoz kell kapcsolni. Ez gyakorlatilag a transzport végpontnak valamely transzport címhez való lekötését jelenti. Természetesen egy felhasználó egyidejűleg akár több transzport végpontot, illetve kapcsolatot kezelhet. A kommunikációs csatorna lokális végpontja nem a traszport entitásnál "ér véget", hanem tovább kell folytatni a kiépítést az internet réteg választott interfészén keresztül a kapcsolatos alhálózatig. A lekötést csak az interfész meghatározásáig kell folytatni, mivel ettől kezdve az összeköttetés az alhálózatig már a konkrét szoftver installáció által egyértelműen meghatározott. Az egyes transzport szolgáltatást adó protokolloknak sajátos mechanizmusuk van az egyes rétegek cimzésének kezelésére, és a végfelhasználók azonosítására. A **t_bind** függvény hívásával lehetséges valamely végponthoz címeket hozzárendelni. Valamely transzport végpont megalapítával kapcsolatos tevékenységeket az alábbi függvények segítségével lehet elvégezni.

t_alloc	Allokálja az adatszerkezeteket
t_bind	Transzport címet kapcsol a transzport végponthoz
t_close	Lezárja a transzport végpontot
t_error	Kiírja a TLI hibaüzenetet
t_free	Felszabadítja a **t_alloc** által lefoglalt területeket
t_getinfo	A transzport protokoll paramétereit adja át a felhasználónak
t_getstate	Visszaadja a transzport végpont pillanatnyi állapotát
t_look	Informálja a kurrens transzport réteg eseményről a felhasználót
t_open	Létrehoz egy transzport végpontot, és hozzárendel a végponthoz egy traszport protokollt
t_optmgmt	Egyezteti a protokoll-specifikus opciókat a traszport szolgáltatóval
t_sync	Szinkronba hozza a transzport végpontot a transzport szolgáltatóval
t_unbind	Leválasztja a transzport címet a transzport végpontról

Több TLI rutinhoz is van vele ekvivalens socket szolgáltatási rutin. Ezeknek a TLI rutinoknak a neve a megfelelő socket rutin nevéből a "t_" prefix alkalmazásával kapható meg.

A kapcsolatfelépítés elemei

A kapcsolatfelépítési fázisban a kommunikációban résztvevő végfelhasználók kapcsolatokat létesítenek egymással a transzport szolgáltatón keresztül.

Így például valamely szerver meghirdeti a szolgáltatását a kliensek egy csoportjának, és ezután blokkolódik egy **t_listen** hívásnál, és várakozik a kapcsolatkialakítási kérésekre. Valamely kliens megpróbál kapcsolatba lépni a szerverrel a meghirdetett címére hivatkozva a **t_connect** hívásban. A kapcsolatkialakítási kérelem megérkezése a szerverhez a szerver blokkolt álapotának feloldását eredményezi, amely ezután a **t_accept** hívással teheti teljessé részéről a kapcsolatkialakítást.

A következő tábázat összegzi azokat a rutinokat, amelyek a kapcsolatkialakítást támogatják.

t_accept	Elfogad egy kapcsolatkialakítási kérelmet
t_connect	Kezdeményez egy kapcsolatkialakítást valamely meghatározott címen levő transzport végpontnál
t_listen	Átvesz egy kapcsolatkialakításra irányuló jelzést a transzport réteg felől
t_rcvconnect	Átvesz egy korábbi kapcsolatkialakításra adott válasznak megfelelő megerősítést a transzport réteg felől

Az adatátviteli fázis elemei

Az adatátiteli fázisban a transzport felhasználóknak lehetőségük van adatátvitelre mindkét irányban egyidejűleg. A **t_send** adatokat küld, a **t_rcv** pedig adatokat fogad a már létrehozott kacsolaton keresztül. Az adatátvitel mindkét irányban sorrendtartóan garantált. A következő tábázat tartalmazza az adatátviteli fázist támogató TLI rutinokat.

t_rcv	Adatot fogad, amely a hivatkozott transzport kapcsolaton érkezett
t_snd	Adatot küld egy már létrehozott transzport kapcsolaton keresztül

A kapcsolatbontás elemei

A TLI a kapcsolatbontásra két lehetőséget is kínál. Az abortív jellegű bontás a kapcsolat azonnali megszüntetését jelenti. Minden megelőzően elküldött adat, amely még nem került átvételre, elveszhet. A **t_snddis** hívás kezdeményezi az abortív kapcsolatbontást, amellyel kapcsolatos jelzést a másik oldalon a **t_rcvdis** hívással lehet átvenni. Ezután a kapcsolat mindkét oldal számára törlődött. Az abortív bontás lehetőségét minden transzport szolgáltatást nyújtó szükségképpen biztosítja.

A transzportot szolgáltató biztosíthatja az egyeztetett bontás lehetőségét. Egyeztetett bontás esetén nincs adatvesztés. Ebben a szolgáltatásban a **t_sndrel** és a **t_rcvrel** rutinok vesznek részt. A kapcsolatban résztvevők bármelyike a **t_sndrel** kérés kiadásával jelzést küldhet a partner felé, hogy a felőle irányuló adatátvitelt befejezte, és ezután már nincs is a továbbiakban lehetősége adatküldésre. A másik irányú adatforgalmat mindez nem érinti. Ha a másik irányú adatforgalom is hasonlóképpen lezárásra kerül, akkor ez a kapcsolat bontását jelenti. Látható, hogy ebben az esetben nincsen adatvesztés.

A kapcsolatbontási fázis rutinjainak összefoglalását az alábbi táblázat adja.

t_rcvdis	Abortív bontással kapcsolatos jelzést ad át a partner végpontnál
t_rcvrel	Egyeztetett bontással kapcsolatos jelzést ad át a partner végpontnál
t_snddis	Abortiv bontási kezdeményezés, vagy elutasító válasz valamely kapcsolatkialakítási kérelemre
t_sndrel	Egyeztetett bontási kezdeményezés

Kapcsolat nélküli szolgáltatás

A kapcsolat nélküli szolgáltatásnak két fázisa van:

- lokális kezelés,
- adatátvitel.

A kapcsolat nélküli szolgáltatás üzenetorientált. Valamely adatcsomag az átvitelhez szükséges összes információt tartalmazza, beleértve a forrás és a rendeltetési címeket is. Minden ilyen információ egyetlen szolgáltatási kérés formájában adódik át, és az egyes adatcsomagok átvitele egymástól teljesen független. A kommunikációban résztvevő végfelhasználóknak kell értelmezniük az átvitt adatcsomagokat a hierarchia egy magasabb szintjén definiált protokoll alapján. A kapcsolat nélküli szolgáltatást megbízhatatlannak is szokták nevezni, mivel az átvitel nem garantáltan sorrendtató.

A kapcsolat nélküli szolgáltatás esetén a lokális kezelés fázisa ugyanazokat a lehetőségeket foglalja magába, mint a kapcsolatorienált esetben. Az adatátviteli fázisba a **t_sndudata** küldi, a **t_rcvudata** pedig fogadja a datagramokat. A transzport réteg felől a hibás datagram-átvitelekkel kapcsolatos jelzéseket a küldőnél a **t_rcvuderr** függvény segítségével lehet átvenni.

A következő táblázat a kapcsolat nélküli adatátviteli szolgáltatás rutinjait foglalja össze.

t_sndudata	Datagramot küld a redeltetési állomás felé
t_rcvudata	Datagramot fogad valamely állomás felől
t_rcvuderr	Hibaüzenetet ad át az előzőleg elküldött datagram hibás átviteléről

Rétegkommunikáció

A transzport interfészt felhasználó folyamat és a transzport réteg **szinkron**, illetve **aszinkron** módon kommunikálhat egymással a transzport réteg felőli jelzések és megerősítések átvétele szempontjából. Ezek a jelzések, illetve megerősítések a megfelelő függvényhívások sikeres végrahajtásával vehetők át. Ha a felhasználó egy ilyen jelzésre, illetve megerősítésre számít, akkor kiadhatja a megfelelő függvényhívást. Ha a jelzés, illetve megerősítés rendelkezésre áll, akkor a szóbanforgó függvény sikeresen visszatérhet. Ha viszont a kívánt jelzés, illetve megerősítés nem áll rendelkezésre, akkor a megfelelő függvényhívás kétféleképpen hajtódhat végre. Szinkron rétegkommunikáció esetén a függvényhívás blokkolódik, amely blokkoltsági állapotot valamely jelzés, illetve megerősítés beérkezése oldja fel automatikusan. Sikeres visszatéréskor a kívánt jelzés, illetve megerősítés érkezett be, sikertelen visszatéréskor pedig a **t_errno** érétke alapján dönthetünk a kialakult helyzetről. Aszinkron rétegkommunikáció esetén az illető függvényhívás végrehajtása nem blokkolódik, és a visszatérés sikeres vagy sikertelen voltából, és a **t_errno** alapján lehet a kalakult helyzetet analizálni.

A **t_errno** változó egy speciális értéke a **TLOOK**. Ha valamely TLI függvényhívás visszatérése hibás, és a **t_errno** értéke **TLOOK**, akkor a **t_look** függvény hivásakor a sikeres végrahajtásnál a visszatérési érték alapján lehet eldönteni, hogy milyen jelzés, illetve megerősítés érkezett, és ennek alapján lehet a megfelelő TLI függvényhívást kiadni a beérkezett jelzés, illetve megerősítés átvételére. Ha valamely függvényhívás abortál és a **t_errno** értéke **TLOOK**, akkor a függvényhívás semmilyen változást nem idéz elő.

A **t_look** függvény lehetséges visszatérési értékeit, és a megfelelő jelséseket, illetve megerősítéseket a következő táblázat tartalmazza.

Név	Leírás
T_LISTEN	Kapcsolatkialakítási kérelem érkezett a transzport végponthoz
T_CONNECT	Megerősítés érkezett egy, az illető végpontnál előzőleg kiadott kapcsolatkialakítási kérelemre a transzport végponthoz
T_DATA	Felhasználói adat érkezett
T_EXPDATA	Expressz felhasználói adat érkezett
T_DISCONNECT	Elutasítás érkezett egy, az illető végpontnál előzőleg kiadott kapcsolatkialakítási kérelemre a transzport végponthoz, vagy abortív bontás jelzése érkezett a létező kapcsolaton a transzport végponthoz
T_ORDREL	Jelzés érkezett egyeztett bontással kapcsolatban egy létező kapcsolat esetében a partner végpont felől
T_UDERR	Jelzés érkezett a transzport réteg felől azzal kapcsolatban, hogy az előzőleg elküldött datagram átvitelében hiba történt

TLI állapotok

Állapot	Leírás	Szolgáltatás
T_UNINIT	Inicializálatlan - az interfész kezdő és végállapota	T_COTS, T_COTS_ORD, T_CLTS
T_UNBND	Inicializált, de címhez nem kötött	T_COTS, T_COTS_ORD, T_CLTS
T_IDLE	Nincs kapcsolat létrehozva	T_COTS, T_COTS_ORD, T_CLTS
T_OUTCON	A kapcsolatkérelem megtörtént a partner felé	T_COTS, T_COTS_ORD
T_INCON	A kapcsolatkérelem érkezett a partner felől	T_COTS, T_COTS_ORD
T_DATAXFER	Adatátvitel	T_COTS, T_COTS_ORD
T_OUTREL	Egyeztetett bontás kérelme elment a partner felé	T_COTS_ORD
T_INREL	Egyeztetett bontás kérelme érkezett a partner felől	T_COTS_ORD

Kimenő események

A kimenő események az érintett transzport rutinok sikeres végrehajtásakor következnek be. Ezek a transzport rutinok szolgáltatási kéréseknek, illteve válaszoknak felelnek meg. Bizonyos eseteben az eseményeket meg kell külnböztetni annak függvényében, hogy milyen környezetben történtek. A környezet a következő paraméterektő függ:

ocnt - beérkezett, de még fel nem dolgozott kapcsolatkérések száma,
fd - a kurrens transzport végpont file leírója,
resfd - annak a transzport végpontnak a file leírója, ahol a kapcsolat kiszolgálásra kerül.

Esemény	Leírás	Szolgáltatás
opened	Sikeres visszatérés a **t_open** hívásból	T_COTS, T_COTS_ORD, T_CLTS
bind	Sikeres visszatérés a **t_bind** hívásból	T_COTS, T_COTS_ORD, T_CLTS

optmgmt	Sikeres visszatérés a **t_optmgt** hívásból	T_COTS, T_COTS_ORD, T_CLTS
unbind	Sikeres visszatérés a **t_unbind** hívásból	T_COTS, T_COTS_ORD, T_CLTS
closed	Sikeres visszatérés a **t_close** hívásból	T_COTS, T_COTS_ORD, T_CLTS
connect1	Sikeres visszatérés a **t_connect** hívásból szinkron módban	T_COTS, T_COTS_ORD
connect2	**TNODATA** hiba **t_connect** híváskor aszinkron módban, vagy **TLOOK** hiba amiatt, hogy **disconnect** érkezet a transzport végpontnál	T_COTS, T_COTS_ORD
accept1	Sikeres visszatérés a **t_accept** hívásból feltéve, hogy $ocnt == 1, fd == resfd$	T_COTS, T_COTS_ORD,
accept2	Sikeres visszatérés a **t_accept** hívásból feltéve, hogy $ocnt == 1, fd! == resfd$	T_COTS, T_COTS_ORD
accept3	Sikeres visszatérés a **t_accept** hívásból feltéve, hogy $ocnt > 1$	T_COTS, T_COTS_ORD
snd	Sikeres visszatérés a **t_snd** hívásból	T_COTS, T_COTS_ORD
snddis1	Sikeres visszatérés a **t_snddis** hívásból feltéve, hogy $ocnt <= 1$	T_COTS, T_COTS_ORD
snddis2	Sikeres visszatérés a **t_snddis** hívásból feltéve, hogy $ocnt > 1$	T_COTS, T_COTS_ORD
sndrel	Sikeres visszatérés a **t_sndrel** hívásból	T_COTS_ORD
sndudata	Sikeres visszatérés a **t_sndudata** hívásból	T_CLTS

Bejövő események

A bejövő események a megfelelő transzport rutinok sikeres befejeződését jelentik. Általában jellemző, hogy ezek a transzport rutinok valamely jelzés, vagy megerősítes átvételével kapcsolatosak. Az egyetlen bejövő esemény, amely nem kapcsolódik egyetlen transzport rutinhoz sem, a **pass_conn** esemény, amely akkor következik be, amikor valamely kapcsolatkérést a végpont elfogad, de egyúttal át is adja azt egy másik már előre létrehozott, és transzport címhez lekötött végponthoz. Ez az esemény annál a végpontnál következik be, ahová a szóbanforgó kapcsolat kiszolgálása átkerül, habár egyetlen transzport rutin hívására sem került sor ennél a végpontnál.

Esemény	Leírás	Szolgáltatás
listen	Sikeres visszatérés a **t_listen** hívásból	T_COTS, T_COTS_ORD
rcvconnect	Sikeres visszatérés a **t_rcvconnect** hívásból	T_COTS, T_COTS_ORD
rcv	Sikeres visszatérés a **t_rcv** hívásból	T_COTS, T_COTS_ORD
rcvdis1	Sikeres visszatérés a **t_rcvdis** hívásból feltéve, hogy $ocnt <= 0$	T_COTS, T_COTS_ORD
rcvdis2	Sikeres visszatérés a **t_rcvdis** hívásból feltéve, hogy $ocnt == 1$	T_COTS, T_COTS_ORD
rcvdis3	Sikeres visszatérés a **t_rcvdis** hívásból feltéve, hogy $ocnt > 1$	T_COTS, T_COTS_ORD
rcvrel	Sikeres visszatérés a **t_rcvrel** hívásból	T_COTS_ORD
rcvudata	Sikeres visszatérés a **t_rcvudata** hívásból	T_CLTS
rcvuderr	Sikeres visszatérés a **t_rcvuderr** hívásból	T_CLTS
pass-conn	Átadott kapcsolat átvétele	T_COTS, T_COTS_ORD

Transzport-felhasználó akciói

Az eseményektől függő állapotátmeneteket táblázatosan adjuk meg. Bizonyos állapotátmenetek esetén a felhasználónak meghatározott akciókat kell végrehajtania. Az egyes akciókra sorszámokkal hivatkozunk. A sorszámokhoz rendelt akciók a következők:

1. Az *ocnt* változó nullára állítása
2. Az *ocnt* változó eggyel növelése
3. Az *ocnt* változó eggyel csökkentése
4. A kapcsolat átadása egy másik végponthoz a **t_accept**-nek megfelelően

Állapotátmeneti táblázatok

Lokális kezelés fázisa

	T_UNINIT	T_UNBND	T_IDLE
opened	T_UNBND		
bind		T_IDLE[1]	
optmgmt			T_IDLE
unbind			T_UNBND
closed		T_UNINIT	

Kapcsolatnélküli szolgáltatás - adatátvitel

	T_IDLE
snudata	T_IDLE
rcvudata	T_IDLE
rcvuderr	T_IDLE

Kapcsolatorientált szolgáltatás - kapcsolatkezelés és adatátvitel

	T_IDLE	T_OUTCON	T_INCON	T_DATAXFER
connect1	T_DATAXFER			
connect2	T_OUTCON			
rcvconnect		T_DATAXFER		
listen	T_INCON[2]		T_INCON[2]	
accept1			T_DATAXFER[3]	
accept2			T_IDLE[3][4]	
accept3			T_INCON[3][4]	
snd				T_DATAXFER
rcv				T_DATAXFER
snddis1		T_IDLE	T_IDLE[3]	T_IDLE
snddis2			T_INCON[3]	
rcvdis1		T_IDLE		T_IDLE
rcvdis2			T_IDLE[3]	
rcvdis3			T_INCON[3]	
sndrel				T_OUTREL
rcvrel				T_INREL
pass_con	T_DATAXFER			

	T_OUTREL	T_INREL
connect1		
connect2		
rcvconnect		
listen		
accept1		
accept2		
accept3		
snd		T_INREL
rcv	T_OUTREL	
snddis1	T_IDLE	T_IDLE
snddis2		
rcvdis1	T_IDLE	T_IDLE
rcvdis2		
rcvdis3		
sndrel		T_IDLE
rcvrel	T_IDLE	
pass_con		