

# Cobol - aplikace v režimu klient - server

Jiří Veselý

## 1. Úvod

Cobol, programovací jazyk s dlouhou historií, prožívá svou renesanci. Zdrojové texty Cobolu jsou převoditelné mezi střediskovými počítači, mini a mikro počítačovou technikou. Cobolské aplikace lze provozovat současně v operačních systémech IBM VS, MS DOS, OS/2, Windows NT, UNIX, AIX a dalších.

Tradiční dobré vlastnosti Cobolu, jako práce se soubory, texty, dokumentační schopnosti, jsou obohaceny moderními programátorskými i neprogramátorskými postupy.

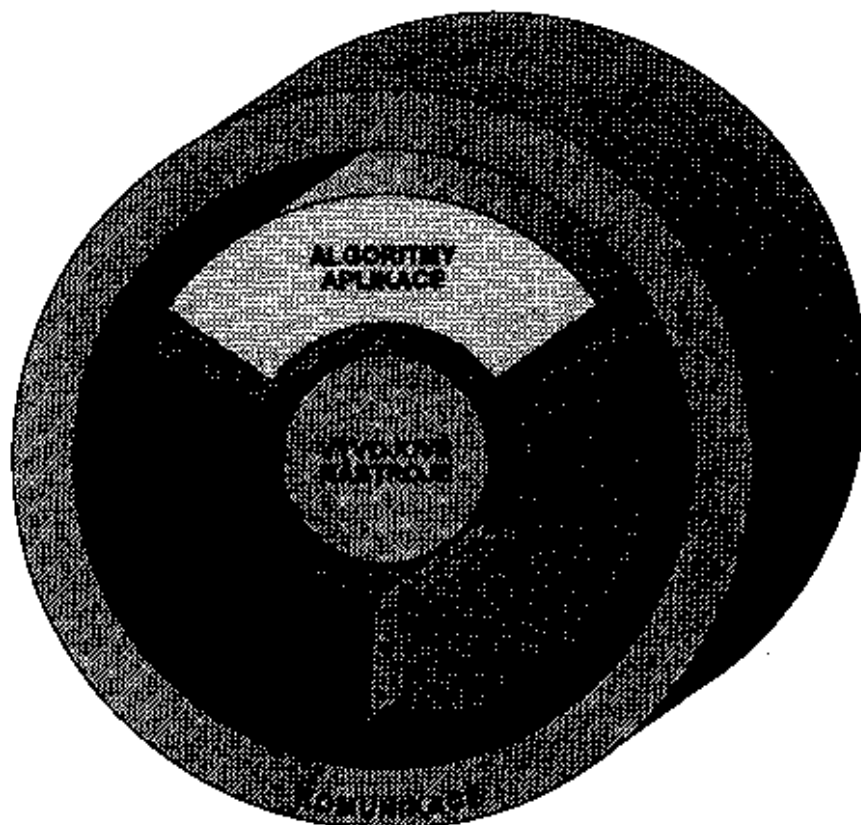
Velký rozvoj počítačové techniky v posledních letech vyžaduje přijetí nových metod programování. Z tohoto pohledu Cobol nezůstává pozadu za "novodobými" programovacími prostředky. Předcházející tvrzení je možno dokumentovat řadou informačních řídicích systémů, celých provedených v Cobolu, které jsou instalovány v ČR.

Na modernizaci a rozvoj programovacích systémů na bázi Cobolu se specializuje americko-britská společnost Micro Focus, firma s celosvětovou působností. Vývojová a provozní prostředí, založená na optimalizačním kompilátoru jsou vhodná pro projekce rozsáhlých interaktivních síťových aplikací.

## 2. Struktura aplikace klient - server

Následující obrázek dokumentuje strukturu prvků a postupů při vývoji aplikace klient - server. Klíčové komponenty jsou :

- algoritmy aplikace
- uživatelské rozhraní
- ovládání datové základny
- komunikace



Obr.1. Komponenty aplikace

## 2.1. Algoritmy aplikace

Micro Focus nabízí svým zákazníkům vhodné nástroje pro tvorbu moderních aplikací, tradičně založených na kompilátoru Cobolu. Cobol je stále ještě zdaleka nejvhodnější jazyk pro programování profesionálních aplikací hromadného zpracování dat.

Kompilátor a další nástroje pro vývoj jsou k dispozici na více než 500 platformách, což umožňuje nezávislost při volbě vhodného hardware a operačního systému. Aplikace může být vyvinuta a testována na pohodlném a laciném vývojovém prostředí (DOS, Windows, OS/2) a kompilovaný kód lze provozovat na uvedených platformách včetně systémů RISC a IBM/390 i bez nutnosti rekompilace.

## 2.2. Uživatelské rozhraní

Uživatelské rozhraní je částí "klient" aplikace klient - server. Produkt Dialog Systém zajišťuje vývoj cobolských aplikací typu GUI (Graphical User Interface), které jsou přenositelné podél grafických interaktivních prostředí (Presentation Manager, Windows, Motif) Dialog Systém (DS) je tedy spojovacím článkem mezi uživateli a vlastním procesem zpracování klient - server. Uživatelské rozhraní je část aplikace, kterou lze zcela oddělit od základní logiky algoritmů aplikace a správy datové základny. Lze ji tedy i samostatně distribuovat. Spojení obou částí aplikace zajišťuje jednoduché programové rozhraní ve formě příkazu CALL.

## 2.3. Ovládání datové základny

V této oblasti je poskytována široká podpora pro indexové a relativní soubory na počítačích kategorie mainframe, midrange (UNIX), PC a LAN. Ide o část "server" aplikace, kterou lze provozovat v režimu transakčního zpracování.

## 2.4. Komunikace

Spojení výše popsaných prvků zajišťují komunikační technologie. Komunikace spojují různé části téže aplikace, nebo dvě či více aplikací dohromady podél dvou nebo více platform. Základní komunikační platformou jsou produkty CCI (Common Communication Interface) a AAI (Application to Application Interface). Strategie těchto produktů umožňuje dosáhnout nezávislosti aplikačního software na použitém prostředí jak pro vývoj tak i pro provozování aplikace. Mezi základní výhody patří přenositelnost (portabilita), schopnost provozování formou "front-end", právě tak jako rozsáhlá podpora pro různé databáze. Výhodou pro programátora je možnost projektovat klient-server aplikace na standardní úrovni jazyka Cobol.

## 3. Technologie a produkty klient - server

Základem provozu aplikací jsou distribuované služby provozního systému OSX.

### 3.1. OSX (Operating System eXtension)

Produkt OSX umožňuje cobolským programátorům vyvíjet aplikace, které lze provozovat v různých operačních systémech, včetně DOS, MS Windows, OS/2, UNIX a AIX. OSX obsahuje funkce API (Application Program Interface), které může programátor využívat v programech bez

ohledu na cílovou provozní platformu aplikace. Výsledkem používání OSX je podstatné rozšíření možností programátora nad rámec makroinstrukcí jazyka. Již dříve zmíněnou přenositelnost aplikací podporuje OSX metodou dynamického rozhraní, díky kterému se požadavky aplikace vůči různým operačním systémům zdají shodné.

Přenositelnost (portabilita) je tedy schopnost přenášet aplikace z jednoho systému na druhý v binární formě bez modifikace zdrojového kódu a rekompile.

### 3.2. Podpora obecného komunikačního rozhraní (CCI)

CCI je aplikační programové rozhraní, které zajišťuje přístup k základní komunikační vrstvě - komunikačnímu protokolu. Moduly CCI (Common Communication Interface) umožňují programu zpřístupňovat síťová a jiná komunikační media v souvislé formě, nerespektující druh komunikačního protokolu. CCI poskytuje sadu funkcí, které dovolují aplikacím zapojit se do jiných pracovních stanic, posílat a přijímat data, potlačovat, řešit a ukončovat zapojení pracovních stanic. Programy sestavené a testované pro jeden systém nebo zařízení jsou snadno přenositelné na jiné systémy beze změny.

Přehled komunikačních protokolů a operačních systémů

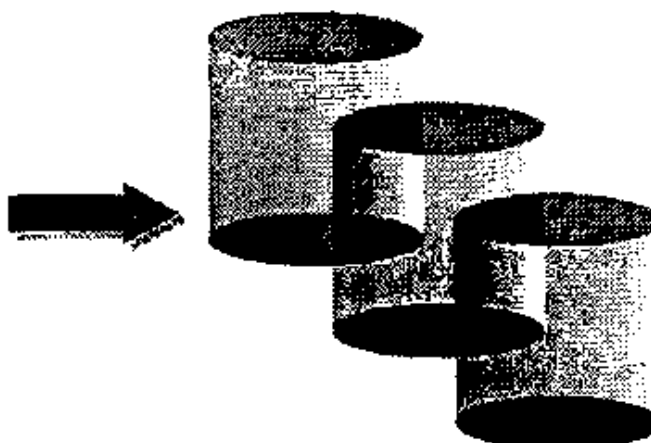
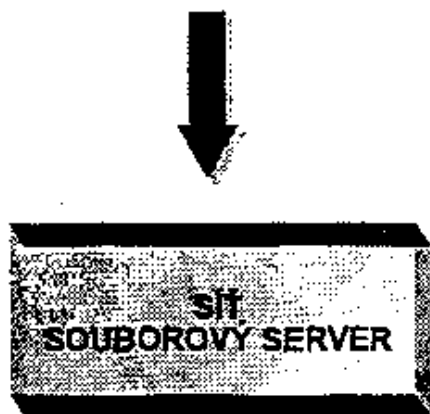
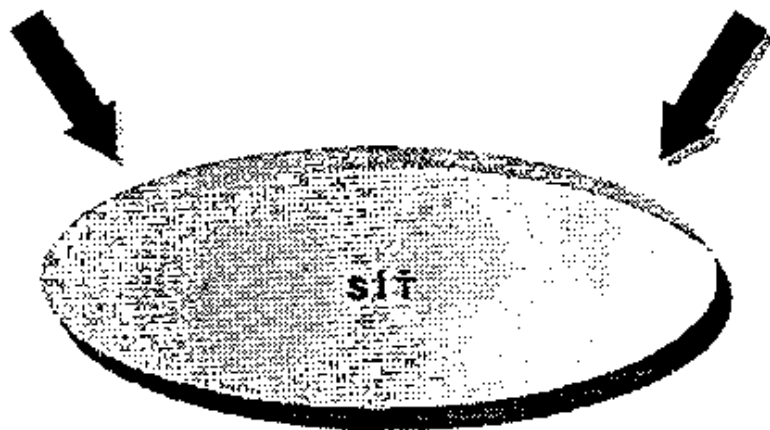
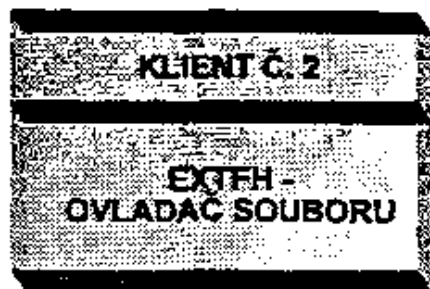
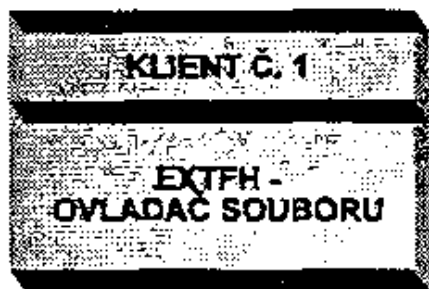
Protokol	Modul CCI	DOS	NLM	OS/2	WINDOWS	UNIX
LU6.2	CCIAPPS	X		X	X	
LU2.0	CCILU2	X		X	X	
Named Pipes	CCINAMP			X		
Named Pipes UNIX	CCINAMPU					X
NETBIOS	CCINETB	X		X	X	
Novell IPX	CCIIPX	X	X	X	X	(X)
TCP/IP	CCITCP	X		X	X	X

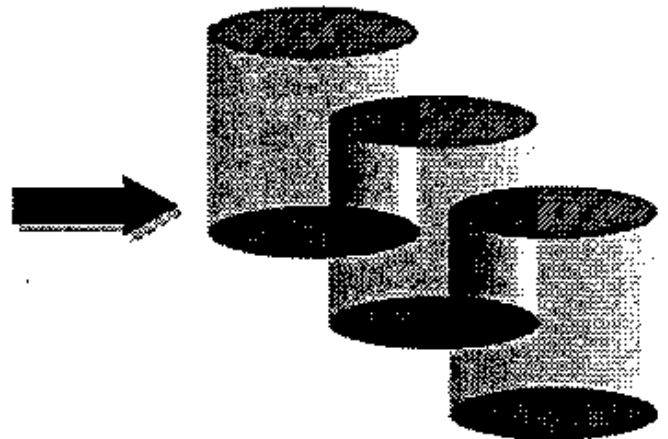
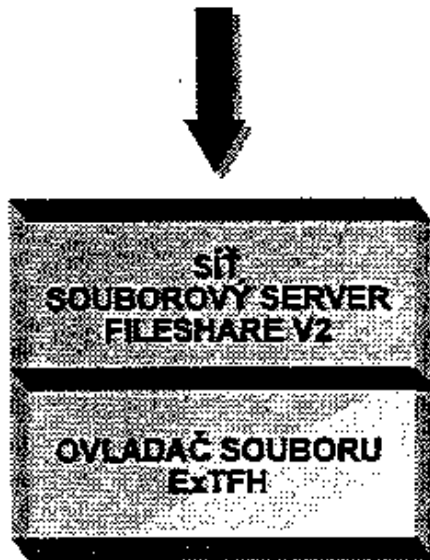
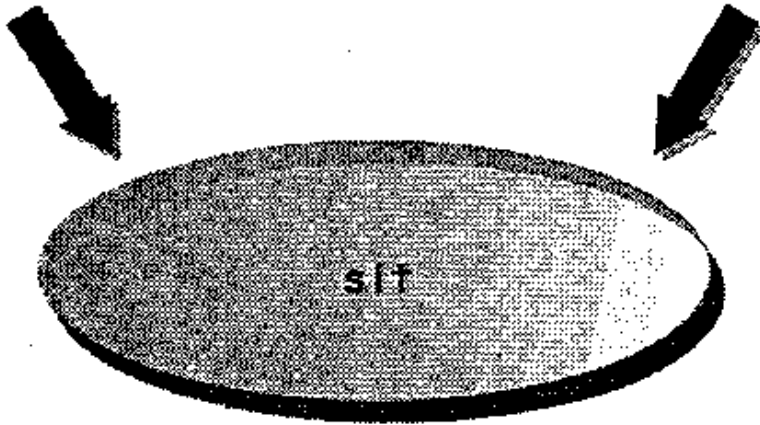
CCI je využit v MF produktech FILESHARE V 2 a AAI.

### 3.3. FILESHARE V 2

FS V 2 zajišťuje přístup k datovým souborům zpracovávaným v síťovém prostředí a jejich integritu. FILESHARE V2 reziduje na serveru a komunikuje s aplikačními programy prostřednictvím služeb programu FHREDIR, který běží na příslušné pracovní stanici. FHREDIR přebírá řízení od každého cobolského příkazu READ, WRITE nebo REWRITE a namísto provedení operace na lokální pracovní stanici předává jednoduchý požadavek centrálnímu ovladači souborů na serveru. Požadavek je následně zpracován systémem FILESHARE V2 a odpověď je vrácena zpět na pracovní stanici. Tento režim podstatně redukuje provoz v síti a umožňuje ovladači souborů zpracovávat vícenásobně používaná data v rychlé vyrovnávací paměti. Dochází tak ke značnému zvýšení výkonnosti síťového systému. Významnou výhodou je možnost transakčního zpracování s monitorováním transakcí, automatickou obnovou datových souborů, dvoufázovým režimem COMMIT a dalšími funkcemi.

Fileshare V2 je tedy ovladač souborů na síťovém serveru, který umožňuje konstituovat cobolské aplikace v režimu klient - server. Na následujících obrázcích je zřejmý rozdíl mezi standardním zpracováním souborů, file - server, ve víceuživatelském prostředí a metodou klient - server.





Tabulka 1 Praktické porovnání konvenčního systému a FILESHARE V 2

Aplikace : 1000 zápisů vět

Počet uživatelů	Konvenční systém [s]	FILESHARE V 2 [s]
1	497	50
2	1 284	68
3	1 874	99
4	3 467	136
5	5 216	180

## 4. AAI - kooperativní zpracování aplikací

AAI (Application-to-Application Interface) je nová kolekce produktů firem CSI (Creative Systems Interface Inc.) a Micro Focus, která umožňuje zapojení peer-to-peer více systémů klient-server a řízení aplikací napříč širokou paletou platform hardware a software. AAI zjednodušuje programování klient-server tím, že odstraňuje zápis specifického komunikačního zdrojového kódu.

AAI dovoluje volat a provozovat programy na libovolném z podporovaných systémů stejným způsobem jako když spouštíte program na individuálním počítači.

Produkty AAI mají standardní programátorské rozhraní na všech podporovaných systémech. Podmínkou je, že tento systém má podporu buď komunikačního protokolu APPC/LU6.2 (Advanced Program to Program), TCP/IP, MRO/XMS (pro komunikaci mezi regiony CICS) nebo Named Pipes. V současnosti jsou produkty AAI dodávány na tato prostředí :

- AAI/CICS pro MVS
- AAI/VTAM pro MVS batch a TSO
- AAI/400 pro OS/400
- AAI/OS pro OS/2
- AAI/DOS pro PC DOS
- AAI/WIN pro Windows
- AAI/AIX pro AIX

AAI umožňuje aplikačním programům rezidovat na kterémkoli z uvedených systémů, volat a provozovat aplikace na kterémkoli z ostatních. Libovolný systém může být klientem vyžadujícím služby a libovolný systém (mimo DOS) může být serverem, který požadované služby zajišťuje.

Jednoduše řečeno AAI je lepidlo vzájemně spojující provozní prostředí aplikací, které by jinak nemohly vzájemně komunikovat.

### 4.1. Jednoduché aplikační programování

Většina existujících aplikací je vyvíjena tradičními programátorskými technikami, které jsou jednoduché při vývoji a zavedení. Je-li nutno zpřístupnit určitou logiku algoritmu více programům, je tato logika zpracována ve formě podprogramu, který může být volán více programy. Tento návrh je možno snadno testovat, může jej použít kterýkoli aplikační programátor.

## 4.2. Kooperativní aplikační programování

Reziduje-li volaný program na různých systémech, je programování více komplexnější. APPC podobně jako MF CCI vymezuje řadu pravidel, která umožňují jednomu programu vyměňovat si data s jiným systémem. V porovnání s principem jednoduchého aplikačního programování nabírá programátor standardní příkaz CALL v programu A zdrojovým kódem APPC a doplňuje kód APPC i do volaného programu B, kde dříve žádný specifický komunikační kód nebyl (viz. následující obrázek).

Zdrojový kód APPC je obtížný pro zápis i testování a je i rozdílný pro všechny systémy, které APPC podporuje. Kódování funkcí APPC je obtížné proto, že příkazy a logika řízení komunikace je příliš komplexní. Kódy obou komunikujících programů musí být identické. Dojde-li k chybě, musí každý program přesně vědět kde se nachází v průběhu komunikace, zda-li zapisuje, čte nebo čeká na provedení externí operace. Každý program musí znát způsob ošetření chyby a činnost spolupracujícího programu. Aplikace APPC jsou komplexní v mnoha oblastech. Výše uvedené speciální zkušenosti jsou nutné pro návrh a práci v různých prostředích, jsou například nezbytné pro náročné testování aplikací. Aplikační program musí počítat s odstraněním všech možných chyb vzniklých v komunikačním prostředí. Při změně komunikačního prostředí je rovněž nutno provést i změny aplikačního programu.

## 4.3. Programování s AAI

Na následujícím obrázku program A volá program B s využitím systému AAI. Příkaz CALL je proveden stejným způsobem jako v jednoduchém aplikačním programování. Rozdíl je pouze v tom, že program A definuje dvě dodatečné datové oblasti, oblast U a R. Programátor zapíše v programu A standardní příkaz CALL programu B a program B přijímá řízení z programu A. Z pohledu programátora se zde uskutečňuje práce s lokálními procedurami. AAI řídí všechnu komunikaci včetně přesunu dat ze systému A do systému B.

*Viz obr. 4 Aplikační programování.*

## 4.4. Jazyková nezávislost

Základním pravidlem, které určuje možnost podpory funkcí AAI pro programovací jazyky je existence příkazu CALL. Prostřednictvím tohoto příkazu jsou vyžadovány služby AAI.

Podporovány jsou tedy programovací jazyky: Cobol, C, RPG, Assembler a většina jazyků čtvrté generace.

## 4.5. Výhody kooperativního zpracování

Vývojáři aplikací mohou nyní psát a testovat programy velmi pohodlně jak je znázorněno na obrázku č. 4. Oba programy A, B mohou běžet během testování na stejném stroji a pro praktický provoz jsou konfigurovány v odlišných systémech.

Ačkoliv návrh je stejný pro všechny kooperativní aplikace, AAI eliminuje všechny komunikační požadavky potřebné při použití APPC. Na rozdíl od jiných řešení, které podporují pouze část aplikace klient, AAI podporuje komunikaci jak klientů tak i serverů v opravdovém režimu Peer-to-Peer.

Jednoduché programové rozhraní dovoluje programátorům v Cobolu psát aplikace klient - server, kdy programátor nepotřebuje mít speciální znalosti komunikačního prostředí.

Síťový administrační systém zajišťuje služby ve všech prostředích, kde lze použít komunikační technologie Micro Focus. AAI podporuje více typů konverzace včetně rozhraní Remote Procedure

Call (RPC), Store & Forward a Peer-to-Peer a umožňuje vývoj smíšených jazykových aplikací. Programátoři jsou schopni vyvíjet aplikace kombinací Cobolu a jiných jazyků například použitím Cobolu na pracovní stanici a relační databáze na různých platformách hardware.

## **Autor :**

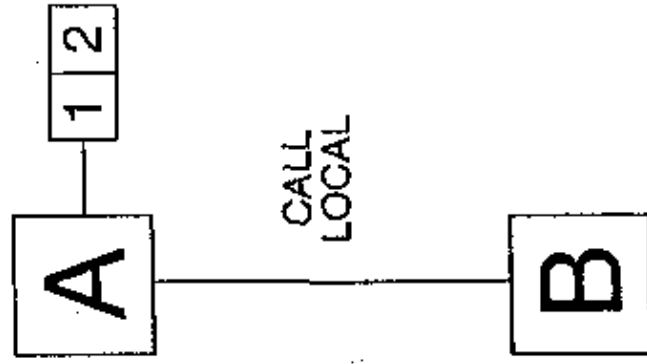
ing. Jiří Veselý  
MF Servis  
Veverkova 1631  
Hradec Králové, 500 02

tel. 049-395 466  
fax. 049-33848

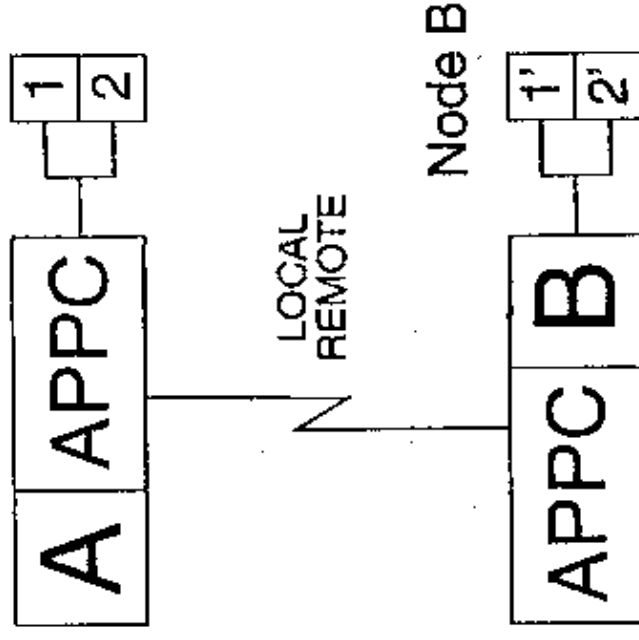


# Aplikační programování - AP

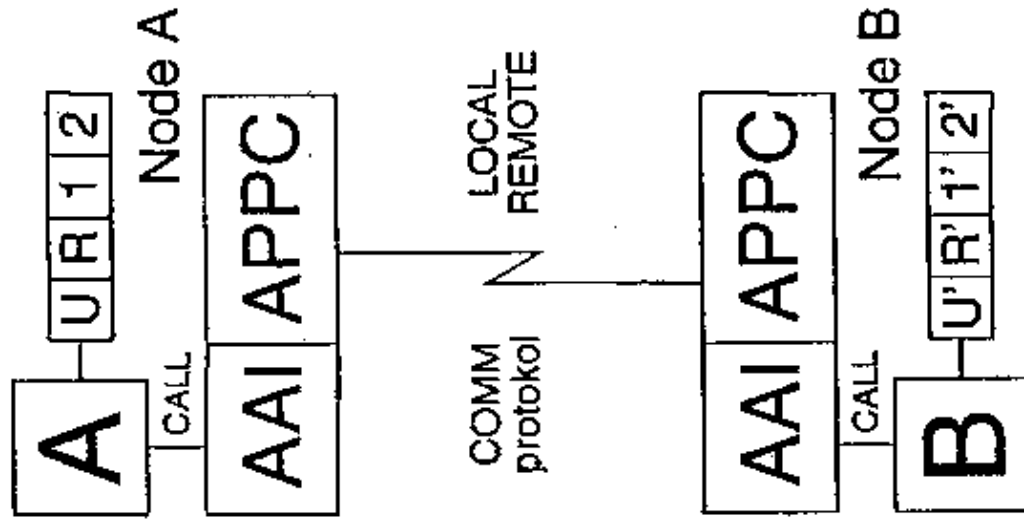
Jednoduché AP



Kooperativní AP



AAI - AP



- U ~ Universal - Request - Message - Header
- R ~ Request - Header
- 1 ~ Požadavek
- 2 ~ Odpověď