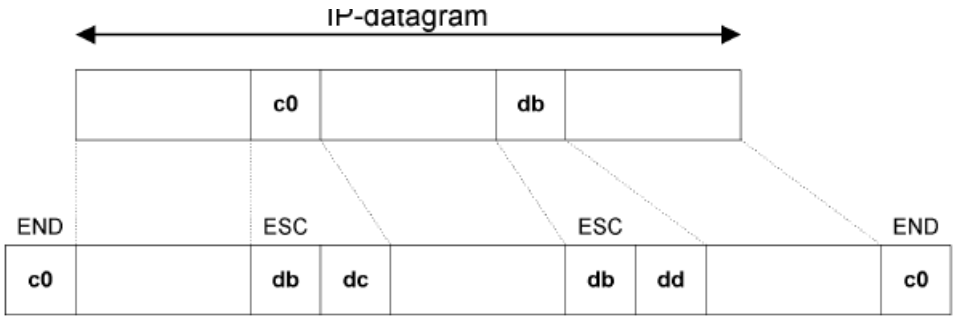
Zdroj: Velký průvodce protokoly TCP/IP a systémem DNS

Linkových protokolů je velké množství - např. SLIP, CSLIP, HDLC, PPP, Frame Relay, Ethernet, FDDI a ATM.

SLIP - Serial Line IP

SLIP je velice jednoduchý protokol, který je určen pro přenos paketů síťových vrstev. **SLIP vkládá IP pakety přímo do sériové linky**. **Pro řízení linky j**sou mezi data vkládány tzv. **Esc-sekvence** (analogicky jako při komunikaci počítače s terminálem či tiskárnou).Protokol SLIP je specifikován normou RFC-1055.



Každý rámec protokolu SLIP je ukončen tzv. Esc-sekvencí END ($c0_{16}$).
Většina implementací protokolu SLIP však Esc-sekvenci END dává navíc i na počátek rámce
Jestliže se vyskytne znak $c0_{16}$ v přenášených datech, pak je nahrazen tzv. SLIP Esc-sekvencí: dvojicí db_{16},dc_{16} (ASCII Esc-sekvence je$1b_{16}$).Znak db_{16} je nahrazován dvojicí db_{16},dd_{16}.

Protokol SLIP je velice jednoduchý, ale nezabezpečuje:

_ **Detekci chyb při přenosu**.
Je proto nebezpečné dávat za linky s protokolem SLIP např. DNS-servery nebo NFSservery, které nemají zapnut kontrolní součet v UDP-datagramu.

_ Rámec protokolu **SLIP nenese informaci o přenášeném protokolu** síťové vrstvy. Je proto možné přenášet vždy pouze jeden síťový protokol, tj. není možné na jedné lince mixovat např. pakety protokolu IP s pakety protokolu IPX.

_ Není možné, aby se oba konce např. informovaly o své IP-adrese či jiných konfiguračních parametrech.

_ Nelze jej použít pro synchronní linky.
Protokol SLIP má díky své jednoduchosti i jednu výhodu. Díky tomu, že neposkytuje téměř žádné služby, tak přenáší minimum služebních informací, takže na méně poruchových pomalých sériových linkách je poměrně oblíben.

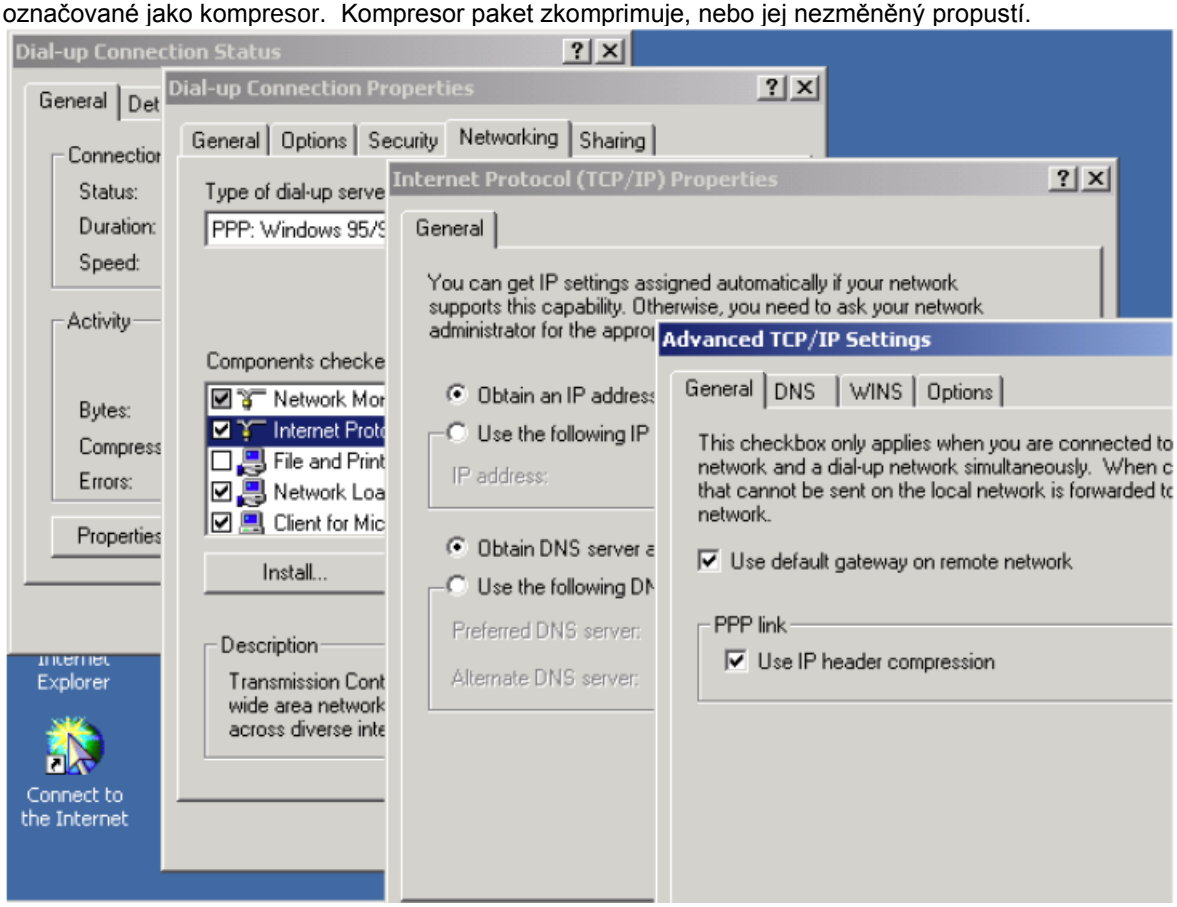
CSLIP - Compressed SLIP

Varianta protokolu SLIP s kompresí. Redukuje 40 bajtů záhlaví protokolů TCP a IP (20 bajtů z IP-záhlaví a 20 bajtů z TCPzáhlaví) na 3 až 16 bajtů. Komprimuje se IP a TCP záhlaví, nikoliv data. Komprese dat je ponechána na modemech.

Zjistilo se, že mnohé údaje v těchto záhlavích se během **TCP spojení nemění nebo se mění jen málo**, takže **stačí přenášet jen změněné položky IP a TCP** záhlaví nebo dokonce jen přírůstky těchto položek. Mění se pouze položky: **identifikace IP-datagramu, pořadové číslo odesílaného bajtu, pořadové číslo přijatého bajtu, některé příznaky, délka okna, kontrolní součet TCP záhlaví a ukazatel naléhavých dat**.
Položky: celková délka IP-datagramu a kontrolní součet IP-záhlaví jsou zase postradatelné.

Komprese záhlaví komprimuje záhlaví pouze v případě, že se jedná o TCP protokol a v záhlavích se mění pouze uvedené položky. V opačném případě (např. je odeslán ICMP paket, je odeslán UDP datagram,

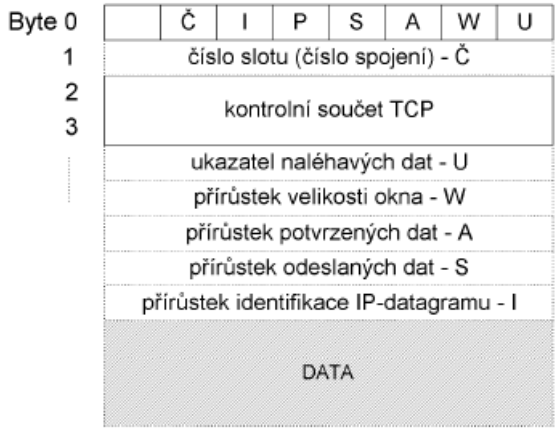
Pokud odesílatel chce přenést TCP/IP paket, pak je paket na straně odesílatele předán komponentě označované jako kompresor. Kompresor paket zkomprimuje, nebo jej nezměněný propustí.



Ve Windows 2000 je v případě konfigurace protokolu PPP též možné nastavit kompresi IP a TCP záhlaví

Kompresor komprimuje jednotlivá spojení. Pro každé spojení si udržuje slot, ve kterém má všechny informace z IP i TCP záhlaví nutné pro kompresi i pro dekompresi, tj. zpětné sestavení obou záhlaví.

Struktura komprimovaného paketu je na obrázku (nepovinné položky jsou znázorněny tečkovaně).



Komprimované záhlaví
Maska – 0B. Jednotlivé bity masky specifikují, které položky v záhlaví originálního paketu se změnily, a proto celé položky nebo jejich přírůstky musí být přenášeny i v komprimovaném záhlaví. Je-li příznak nastaven, pak v komprimovaném záhlaví je uvedena konkrétní položka komprimovaného záhlaví, pokud není nastaven, pak příslušná položka není v komprimovaném
záhlaví přítomna.

Kontrolní součet z TCP-záhlaví se přenáší vždy.

Jednotlivé bity masky:
_ **Č – označuje číslo slotu**. Číslo slotu není povinné, pokud není uvedeno, pak se předpokládá, že je shodné s číslem slotu předchozího komprimovaného paketu přenášeného linkou. Číslo slotu je dlouhé jeden bajt (tj. nabývá hodnoty v intervalu 0-255), protože číslo slotu se mezi kompresorem a dekompresorem přenáší v poli "Protokol vyšší vrstvy *(Protocol)*", které je dlouhé právě jeden bajt.
Na lince je tedy možné v jednom okamžiku komprimovat max. 255 spojení. Proto je komprese určena spíše pro linky připojující PC k Internetu a není příliš vhodná pro propojení páteřních směrovačů.
_ **U – ukazatel naléhavých dat**. Signalizuje, že je v paketu vyplněno pole obsahující ukazatel naléhavých dat.
_ **W – přírůstek velikosti okna**. V komprimovaném záhlaví se nepřenáší hodnota celého okna, ale pouze její přírůstek. Pokud by přírůstek byl záporný nebo větší než 64K (tj. nevešel by se do dvou bajtů), pak se paket nekomprimuje. Obdobně je tomu i u bitů A, S a I.
_ **A – přírůstek potvrzených dat.**
_ **S – přírůstek odeslaných dat.**
_ **I – přírůstek identifikace IP-datagramu.**
_ **P – příznak PUSH.** Tento příznak se odlišuje od ostatních příznaků tím, že mu neodpovídá konkrétní položka komprimovaného záhlaví. Je-li tento příznak nastaven, pak originální paket má nastaven příznak PUSH v záhlaví TCP segmentu.

TCP spojení je plně duplexní, tak komprimace IP + TCP záhlaví se provádí pro každý směr zcela samostatně, tj. jako by šlo o dva samostatné simplexní spoje.

V případě, že by v komprimovaném záhlaví byly současně nastaveny příznaky A, W a U, pak se přenášený paket nekomprimuje – jedná o výjimku připravenou pro protokoly telnet, rlogin apod. Pro tyto protokoly se komprimované záhlaví skládá pouze z masky s nastavenými příznaky A, W a U a kontrolního součtu, tj. komprimované záhlaví se zkracuje na 3 bajty.

V tomto případě se opravdu při stisknutí jedné klávesy na terminálu místo 41 bajtů přenáší pouze čtyři bajty (3 bajty komprimovaného záhlavía 1 bajt dat). Blíže viz RFC-1144.

HDLC High-Level Data Link Control

Protokol HDLC vznikl z protokolu SDLC firmy IBM (vyvinutý pro architekturu SNA - Systems Network Architecture).

Protokol SDLC byl primárně **určen pro synchronní přenos** (**Synchronous data-link control).**

Vlastnosti:
- Bitově orientovaný (né znakově)
- Plně duplexní
- Provádí **detekci chyb**
- Zajišťuje **řízení toku dat**

Dnes se protokol SDLC vesměs chápe jako podmnožina protokolu HDLC, i když ne všechny možnosti protokolu SDLC byly do HDLC zahrnuty.

Později byla norma **HDLC rozšířena i pro asynchronní přenos. Asynchronní varianta je dále řešena protokolem PPP,** který je podmnožinou protokolu HDLC.

Dále řešíme synchronní bitově orientovaný přenos na fyzické úrovni.

Protokol **HDLC je velice rozsáhlá norma** mající velké možnosti (mnohé jsou volitelné či se dokonce vzájemně vylučují). Proto většina firem **dodává programy nejen pro vlastní realizaci** protokolu HDLC, ale **i pro implementace svých nejdůležitějších konkurentů** (CISCO HDLC, DEC HDLC)..

Protokol HDLC rozeznává tzv. módy:

_ *ABM (Asynchronous balanced mode),* který je určen **pro propojení dvou stanic plně duplexním spojem**. **Většinou se dnes setkáváme právě s tímto módem**.

_ *NRM (Normal response mode).* mód **odpovídá protokolu SDLC**. Jedná se o situaci, kdy je **propojeno více stanic na polo-duplexním spoji.** Tento mód je v Internetu používán jen výjimečně. (Povolení se nastavuje P/F bitem v řídícím poli HDLC-rámce – vysvětlení významu P/F bitu).

_ *ARM (Asynchronous response mode)* je dnes **málo používaný** režim.

Struktura rámce

Flag	Adresa	Řídicí pole	Data	FCS	Flag
8 bitů	8 nebo více bitů	8 nebo 16 bitů	Proměnná délka, 0 nebo více bitů	16 nebo 32 bitů	8 bitů

Křídlová značka (Flag) HDLC rámec začíná a končí křídlovou značkou. Křídlová značka se skládá z **osmi bitů**: 0111 1110. **Šest po sobě jdoucích jedniček určuje křídlovou značku**. **Dvě křídlové značky** po sobě, **znamenají prázdný rámec**, se kterým se dále nepracuje.

Pokud **vstupní data** obsahují **více než pět jedniček za sebou, vloží se za každou pátou jedničku automaticky jedna nula**. Je-li pak ve výstupních datech za pěti jedničkami nula, pak se tato nula vypustí. Tento proces se nazývá **bit-stuffing. Toto se dá využít jen u bitově orientovaného přenosu.**

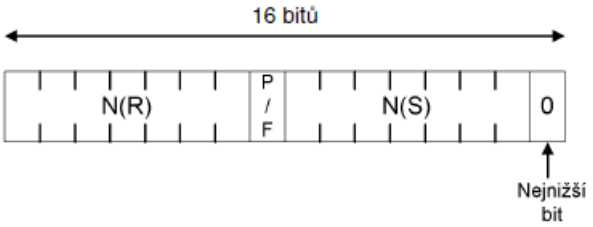
Adresní pole Adresní pole je **dlouhé 8 bitů**. Označuje adresu stanice, které je paket určen**. Účelně se využívá jen v módu NRM, kdy mezi sebou komunikuje více stanic, tj. jedná se o linkovou adresu.**

Kontrolní součet Z **přenášených dat, adresního a řídicího pole se počítá kontrolní součet**. Z přijatého rámce se spočte kontrolní součet, který se porovná s kontrolním součtem v přijatém rámci. Jsou-li shodné, pak přenos proběhl správně. **Nejsou-li shodné, tak se může přenos u číslovaných rámců zopakovat.**

Řídící pole Podle **nejnižších dvou bitů řídícího pole rozlišujeme 3 typy HDLC-rámců**:

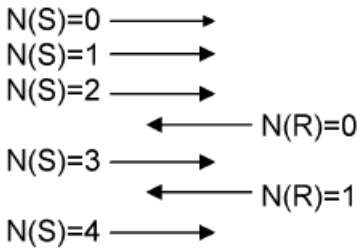
- I-rámce
- U-rámce
- S-rámce

_ **Informační rámce neboli I-rámce (v nejnižším bitu je 0)**, které jsou **primárně určeny pro přenos dat.** Mohou však ve svém řídícím poli přenášet i **některé řídící informace (např. pozitivní potvrzení přijatých rámců).**



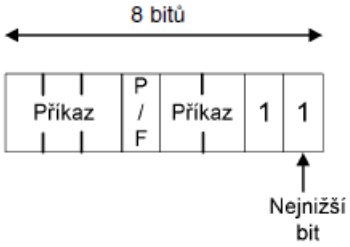
Pole N(S) a N(R) je pro číslování rámců.
N(S) - číslo odesílaného rámce.
N(R) - potvrzení přijatého rámce
Jelikož je komunikace obousměrná, potvrzují se v protisměru správně přijaté rámce.

Rámce se potvrzují pomocí tzv. okna (okénkové potvrzování). Je-li např. okno rovno třem, pak až po odeslání tří paketů se čeká na potvrzení prvního z nich. Po potvrzení prvního se odešle čtvrtý, po potvrzení druhého se odešle pátý, ... **Ve vyrovnávací paměti odesílatele je nutné udržovat celé okno nepotvrzených rámců** pro případ vyžádání chybného nebo ztraceného rámce.



Okno o velikosti 3.

_ **Nečíslované rámce neboli U-rámce (v nejnižších dvou bitech je 11)**, které se používají nejen **pro přenos dat, ale i pro řídící funkce** (**úvodní inicializační dialog, řízení linky a diagnostiku).**



U-rámce přenáší i **příkazy a odpovědi**:

_ **SABM** (*Set Asynchronous Mode* = nastavení módu ABM). **Příkaz nastavuje linku do módu ABM** s osmibitovým řídícím polem.

_ **SABME** (*Set Asynchronous Mode* = nastavení módu ABM). **Příkaz nastavuje linku do módu ABM s šestnáctibitovým řídícím polem –** jedná se tedy o tvar protokolu HDLC zobrazovaný v této kapitole.

_ **SNRM** (*Set Normal Response Mode* = nastavení módu NRM). Příkaz nastavuje linku **do módu**

NRM s osmibitovým řídícím polem.

_ **SNRME** (*Set Normal Response Mode* = nastavení módu NRM). Příkaz nastavuje linku do módu NRM s šestnáctibitovým řídícím polem.

_ **UA** (*Unnumbered Acknowledgment* = nečíslované potvrzení). Používá se **pro potvrzení** SABM, SABME, SNRM, SNRME a DISC.

_ **DISC** (*Diconnect* = odpojení). U komutovaných linek znamená **požadavek na zavěšení**. U pevných linek umožňuje **zrušit aktuální operační mód** (aktuální nastavení).

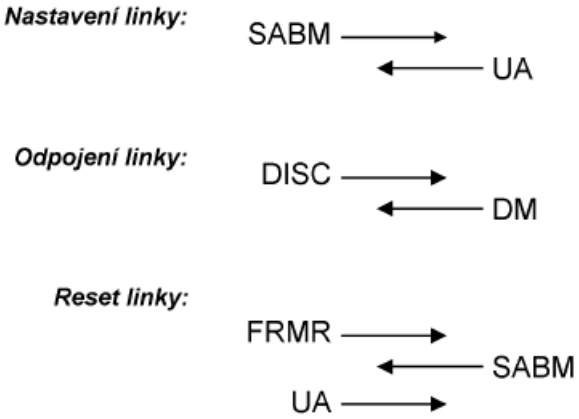
_ **DM** (*Disconnect Mod*). Pozitivní **potvrzení příkazu DISC**.

_ **FRMR** (*Frame Reject* = odmítnutí rámce). Používá se k **indikaci přijetí vadného rámce**. Přitom není možné dosáhnout opravy opakováním vysílání (retransmisí).

_ **XID** (*Exchange Station Identification* = výměna konfiguračních informací). Příkazy a odpovědi XID se používají k **úvodní inicializační sekvenci, kdy se stanice domlouvají na délce kontrolního součtu, přenášeném protokolu vyšší vrstvy** atd.

_ **UI** (*Unnumbered Information* = nečíslované datové rámce). Používá se pro **přenos nečíslovaných datových rámců.** Tyto rámce mohou na počátku datového pole obsahovat specifikaci přenášeného protokolu, takže **je pak možné na přenosové lince mixovat různé protokoly** (např. IP a IPX).

Příklad dialogu protokolu HDLC



_ **Rámce supervizoru neboli S-rámce (v nejnižších dvou bitech je 10)**. Používají se pro **řízení toku dat** (požadavek na vysílání, potvrzování I-rámců atd.). S-rámce mohou být používány až když je linka inicializována, tj. když mohou být používány I-rámce. S-rámce **neobsahují datové pole.**

S-rámec **může potvrzovat správně přijatý rámec**. V poli příkaz nese následující příkazy resp.odpovědi:

_ **RR** (***Receiver Ready* = přijímač připraven**). Informuje protějšek, že přijímač **je připraven akceptovat I-rámce**. Dále se používá jako **signalizace, že linka je opět volná** (poté co tomu tak nebylo).

_ **RNR** (***Receiver Not Ready* = přijímač nepřipraven**). Informuje protějšek o dočasné **neschopnosti přijímat I-rámce**, zároveň **potvrzuje dosud přijaté rámce**.

_ **REJ (*Reject* = odmítnutí).** Přijetí chybného rámce, tj. používá se jako příkaz nebo jako odpověď **pro zopakování vysílání**.

Linka v protokolu HDLC – stavy linky

Nachází se v:

- **Odpojeném stavu.**
- **Ve stavu nastavování linky**, kdy se mohou používat **pouze U-rámce**.
- **Ve stavu přenosu informací,** tj. za běžných okolností se přenášejí pouze **I a S-rámce** (nepoužívají-li se k přenosu dat U-rámce).
- **Odpojování linky**, opět se přenáší jen **U-rámce**.

Protokol HDLC umožňuje:

- **Pomocí kontrolního součtu zjišťovat chybné rámce.**
- **Při přijetí chybného číslovaného rámce je možné vyžádat zopakování** vysílání rámce (nečíslované chybné rámce se zahazují).
- **Pomocí nečíslovaných rámců je možné na lince mixovat více síťových protokolů**.

PPP Point-to-Point Protocol

Protokol PPP využívá rámce tvaru protokolu HDLC. Nevyužívá všechny možnosti protokolu HDLC.

- Na fyzické úrovni je **schopen používat rozhraní podle doporučení V.24, V.35** atp. Nevyžaduje žádné řídící signály (RTS, CTS, DCD, DTR atp.). Řídící signály však mohou být využity pro zvýšení efektivity.

- **Může používat jak asynchronní, tak bitově či znakově synchronní přenos dat.**

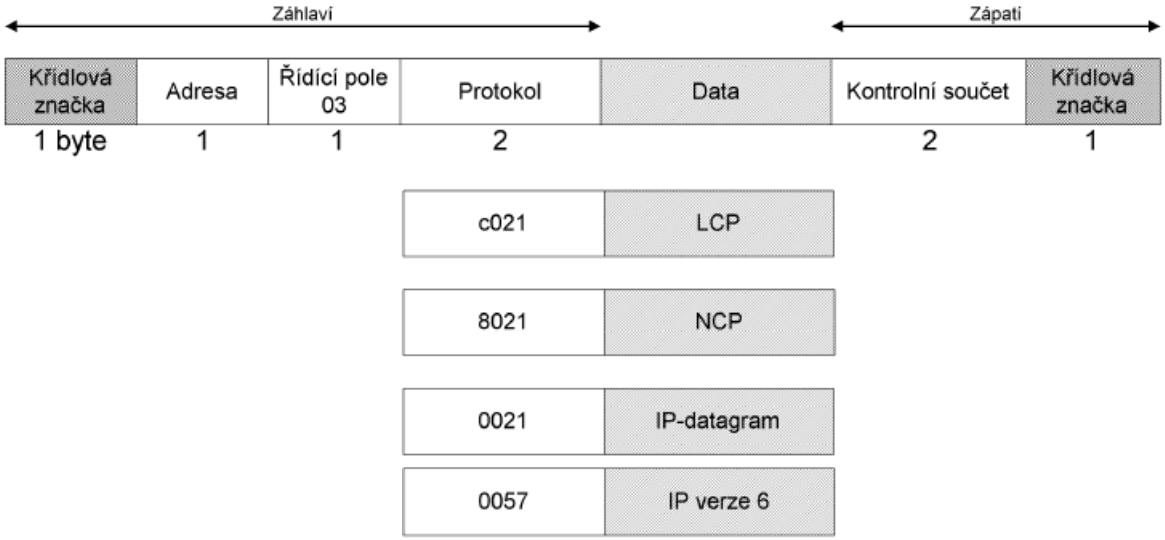
- Pro **asynchronní přenos použije 1 start bit, 8 datových bitů a 1 stop bit (bez parity).**

- **Vyžaduje plně duplexní dvojbodové spoje (*point-to-point*)**, které mohou být **pevné i komutované**.

- Využívá zpravidla **16 nebo 32 bitů pro kontrolní součet**, aby mohl zjistit, zda nebyl rámec během přenosu poškozen.

- Cílem protokolu PPP je **umožnit po jedné lince přenos více síťových protokolů současně** (mixovat protokoly). **Nepoužívá I-rámce,** ale **přenos dat provádí pouze pomocí U-rámců.** Nemůže použít číslování rámců, a tedy ani možnost opakování rámce v případě zjištění chybného rámce.
- Na **počátku datového pole umísťuje osmi nebo šestnáctibitovou identifikaci přenášeného síťového protokolu**.

Protokol PPP specifikuje RFC-1661. Tvar rámce PPP-protokolu specifikuje RFC-1662.



Rámec protokolu PPP **obsahuje v poli adresa ff16 (oběžník)** a **v řídícím poli vždy 0316** (U-rámce s nastaveným P/F bitem na 0). Pokud se na lince vyskytují **rámce pouze s těmito adresami a řídícími poli**, pak **oba konce linky mohou použít kompresi** (*Address-and-Control-Field-Compression*). Při této kompresi se při vysílání obě pole vypustí.

Křídlová značka (flag). Křídlovou značkou je **uvozen i ukončen každý rámec protokolu PPP**.Křídlová značka obsahuje binárně 0111 1110, tj. 7e16. Co ale když je třeba znak 7e přenášet v datech?

U binárně **synchronních linek byla popsána technika *bit stuffing***. Pro **asynchronní spoje (též pro znakově synchronní linky) se použijí Esc-sekvence** (podobně jako u protokolu SLIP).

Znak 7e se nahradí dvojicí 7d 5e. A znak 7d se nahradí dvojicí 7d 5d. Implicitně se uvozují escape sekvencí 7d i všechny řídící znaky ASCII (tj. znaky s kódem desítkově menším než 32). Navíc se k hodnotě těchto znaků připočte desítkově 32 (tj. 20 šestnáctkově). Např. místo znaku 03 se přenáší 23_{16}. Takže ani terminálový ovladač nemůže přenášeným znakům uškodit tím, že by je chybně interpretoval např. jako zvonek, BACKSPACE atp.

Na binárně synchronní lince se escape sekvence nepoužívají. Ale není tomu tak vždy. S escape sekvencemi je možné se setkat i na binárně synchronních linkách. Proč? V případě, že je třeba konvertovat přenos z asynchronních na bitově synchronní (a naopak), pak escape sekvence přejdou z asynchronního přenosu do synchronního jako znaky. Při odpovědi musí naopak synchronní strana obohatit synchronní data o escape sekvence, aby bylo po konverzi možné komunikovat s protějškem. Takže i synchronní stanice mohou používat příkaz *Async – Control – Character – Map.*

Součástí protokolu PPP jsou dva služební protokoly:

- **Protokol LCP sloužící k navázání spojení, autentizaci stanic** apod.

- **Skupina protokolů NCP.** Každý **síťový protokol, který využívá linkový protokol PPP má definovánu vlastní normu pro protokol NCP.** Součástí této normy je vždy i číslo protokolu, které se použije v poli protokol rámce, a to jak pro příslušný protokol NCP (číslo začíná číslicí 8), tak i pro datové rámce (číslo začíná číslicí 0).

Máme např.
▮ **Protokol IPCP** (číslo protokolu 8021_{16}), je **variantou protokolu NCP pro IP-protokol verze 4,** IPCP je specifikován RFC-1332. Datové rámce používají v poli protokol hodnotu 0021_{16}.

▮ **Protokol IPV6CP** (číslo protokolu 8057_{16}), je **variantou protokolu NCP pro IP-protokol verze 6** (RFC-2023). Datové rámce přenášené protokolem IP verze 6 používají číslo protokolu 0057_{16}.

▮ **Protokol SNACP** (číslo protokolu 804d), tj. **protokol NCP pro IBM SNA** (RFC-2043). Datové rámce používají číslo protokolu 004d.

▮ **Protokol DNCP** (číslo protokolu 8027), tj. **protokol NCP pro DECnet Pahse IV** (RFC-1762). Datové rámce používají číslo protokolu 0027.

▮ **Protokol IPXCP** (číslo protokolu 802b), tj. **protokol NCP pro IPX** (RFC-1552). Datové rámce používají číslo protokolu 002b.

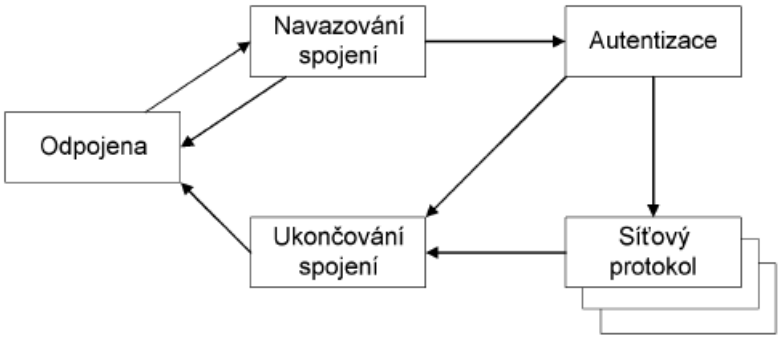
▮ **Protokol OSINLCP** (číslo protokolu 8023), tj. **protokol NCP pro OSI protokoly,** tj. např. protokoly ES-IS, IS-IS atd. (RFC-1377). Datové rámce používají číslo protokolu 0023
.

Protokol LCP

LCP je **společný protokol pro všechny síťové protokoly.** Je určen pro:

- navázání spojení
- ukončení spojení
- výměnu autentizačních informací apod.

Linka se nachází postupně ve fázích spojení, autentizace, síťový protokol a ukončování spojení



Linka odpojena je fáze, ze které se vždy začíná a končí. Když dojde k nějaké externí události (např. modemy ztratí mezi sebou spojení), přechází linka do této fáze.

Navazování spojení. Navazování spojení se provádí výměnou konfiguračních paketů. V případě výskytu datového paketu během navazování spojení se takový paket zahazuje.

Autentizace je fáze, kdy klient prokazuje svou totožnost. Kdo je to klient? Klientem je ta strana (stanice), která je vyzvána k prokázání své totožnosti. Po prokázání totožnosti jedné stanice si mohou stanice svou roli vyměnit a k prokázání své totožnosti může být vyzvána druhá strana. V praxi většinou prokazuje svou totožnost jen jedna strana (např. uživatel PC proti poskytovateli Internetu).

Autentizace je nepovinná, tj. může být přeskočena. Protokol LCP nepopisuje žádný autentizační algoritmus, pouze přenáší data, která pak následně využijí autentizační protokoly.

Jako autentizační protokol se používá protokol PAP nebo CHAP. Navíc ještě je zpravidla možná terminálová autentizace.

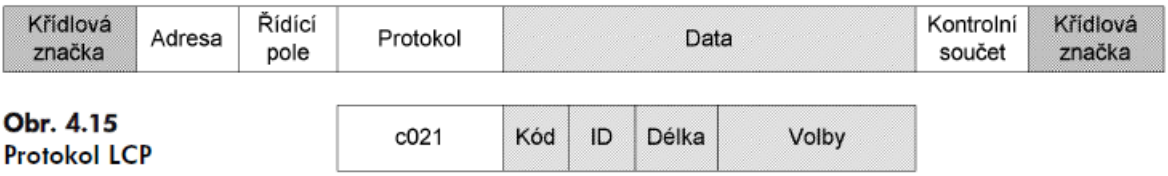
Fáze označena jako **síťový protokol** v sobě může obsahovat celou řadu kroků.V tomto okamžiku přicházejí ke slovu jednotlivé protokoly NCP. Každý síťový protokol, který chce linku využívat si musí přivést pomocí svého protokolu NCP linku do otevřeného stavu pro tento protokol.

Např. mají-li se na lince přenášet pakety protokolu IP verze 4 a protokolu IS-IS, pak se musí linka otevřít dvakrát, jednou pomocí protokolu IPCP a podruhé pomocí OSINLCP.

Linka může být otevřena pro více síťových protokolů současně.

Ukončování spojení. Během této fáze jsou všechny jiné pakety než protokolu LCP zahazovány. Fyzické vrstvě je signalizováno ukončení spojení. Fyzická vrstva může reagovat např.zavěšením komutované linky.

Formát rámce LCP protokolu



Obr. 4.15
Protokol LCP

Kód - osmibitové pole kód specifikuje typ příkazu (resp. odpovědi) protokolu LCP:

kód	Název (anglicky)	Význam
1	Configure-Request	Konfigurační paket nesoucí požadavky na změnu implicitních parametrů linky.
2	Configure-Ack	Konfigurační paket s kladným potvrzením požadavků na změnu implicitních parametrů linky. Tj. všechny požadované změny parametrů jsou akceptovány.
3	Configure-Nak	Konfigurační paket s odpovědí. Avšak protější strana neakceptuje všechny požadavky na změnu parametrů linky. Ty, které neakceptuje, jsou v tomto paketu specifikovány. Ostatní požadavky jsou akceptovány (tj. nespecifikované požadavky v paketu Configure-Nak jsou akceptovány).
4	Configure-reject	Konfigurační paket odmítající všechny požadavky. Může být důsledkem i chybného kódu požadavku.
5	Terminate-Request	Požadavek na ukončení spojení.
6	Terminate-Ack	Potvrzení požadavku na ukončení spojení.
7	Code-Reject	Odmítnutí požadavku z důvodu neznámého kódu. Může být způsobeno i tím, že protější stanice používá jinou verzi protokolu.
8	Protokol-Reject	Protější strana nepodporuje uvedený protokol.
9	Echo-Request	Podpora testovací smyčky na linkové úrovni.
10	Echo-Reply	Povinná odpověď na Echo-Request.
11	Disckard-Request	Zahoď tento paket. Používá se pro testování linky při zátěži, tj. odesílatel generuje pomocí těchto paketů umělou zátěž linky.

ID - Osmibitové pole **ID** je identifikace požadavku. Odesílatel zvolí identifikaci do tohoto pole a adresát ji zkopíruje do své odpovědi. Pomocí tohoto pole se **určuje příslušnost odpovědi k danému požadavku**.

Délka - Šestnáctibitové pole **délka** obsahuje číslo udávající součet délek polí: kód, ID, délka a volby
.

Volba - Pole **volby** obsahuje jednotlivé požadavky (resp. odpovědi) na změnu implicitních parametrů linky.

Autentizace

Prokazovat totožnost lze v případě protokolu PPP trojím způsobem (neuvažujeme-li jako čtvrtou možnost eventualitu, že autentizace je zcela vynechána):

- **Terminálový dialog**. Terminálový dialog nesouvisí s protokolem PPP, nýbrž s jeho implementací. Uživatel se přihlašuje po sériové lince k serveru. Na této lince terminálový proces **vyžaduje jméno uživatele a heslo**.

Teprve ze **jména uživatele pozná, že se nejedná o běžného uživatele** terminálu, ale uživatele, pro kterého má na lince startovat protokol PPP (např. proces pppd). Pokud je takováto autentizace na serveru možná a je dostačující, pak je možné autentizační fázi protokolu PPP přeskočit.

- **Protokol Password Authentication Protocol (PAP).** Tento protokol je obdobou autentizace pomocí terminálového dialogu. Tj**. uživatel prokazuje svou totožnost také pomocí jména uživatele a hesla.** Pro výměnu autentizačních informací se ale **použije protokol LCP**, tj. jméno uživatele a heslo se nevkládá přímo na linku, ale balí se do protokolu LCP.

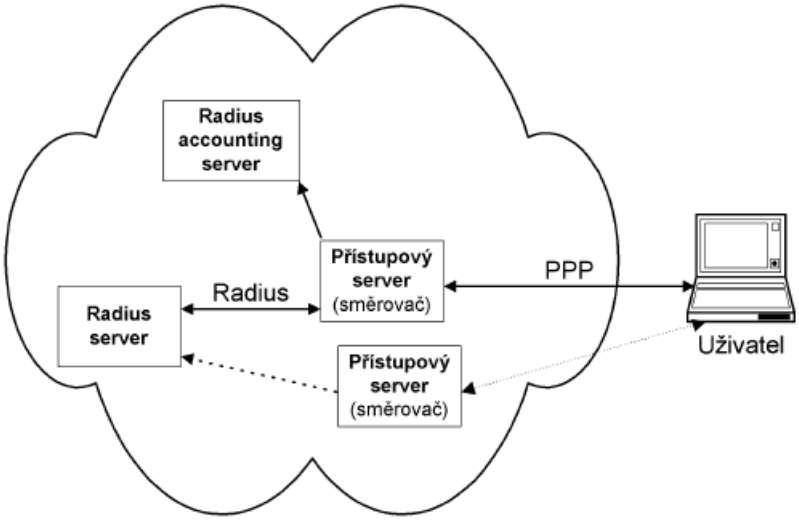
- **Challenge Handshake Authentication Protocol (CHAP)**. Je považován za dokonalejší. Oba konce sdílí stejné tajemství (v podstatě je to šifrovací klíč symetrické šifry). Stanice, která autentizaci inicializuje, vygeneruje náhodný řetězec jako dotaz (challenge), který odešle druhé straně. Druhá strana tento řetězec zašifruje pomocí sdíleného tajemství a odešle zpět. Stanice, která autentizaci inicializovala, tak obdržela zašifrovaný řetězec, který dešifruje. Porovná oba řetězce.Jsou-li oba řetězce stejné, pak protějšku potvrdí úspěšný výsledek autentizace. V opačném případě odpoví, že autentizace proběhla neúspěšně a může se začít znovu s navazováním spojení.

Výhodou protokolu CHAP je skutečnost, že **oba konce znají sdílené tajemství** – je tak snadno možné **provádět oboustranně autentizaci**. Sdílení tajemství je současně nevýhodou protokolu CHAP, nelze zabránit zneužití tohoto tajemství druhou stranou (na rozdíl od autentizace heslem, kde druhá strana má přístup pouze k zašifrovanému heslu). Protokol CHAP specifikuje RFC-1994.

Další problém s autentizací spočívá v tom, že **se klient bude chtít přihlašovat nikoliv stále na jeden přístupový server, ale na různé přístupové servery**. Klasickým případem je připojení k poskytovateli Internetu, který má své přístupové body v různých městech. V takovém případě by autentizační informace musely být udržovány na každém přístupovém serveru. Myšlenka spočívá v centralizaci autentizačních informací. V síti je jeden (nebo i více záložních) serverů, které udržují autentizační informace o každém uživateli. Kromě autentizačních informací mohou být udržovány i konfigurační informace (např. IP-adresa uživatele, přístupové filtry atd.). Přístupový server pak vůči takovémuto serveru vystupuje jako klient, který požaduje službu: prověření autentizační odpovědi či poskytnutí IP-adresy, kterou má protokolem IPCP předat uživateli atd.

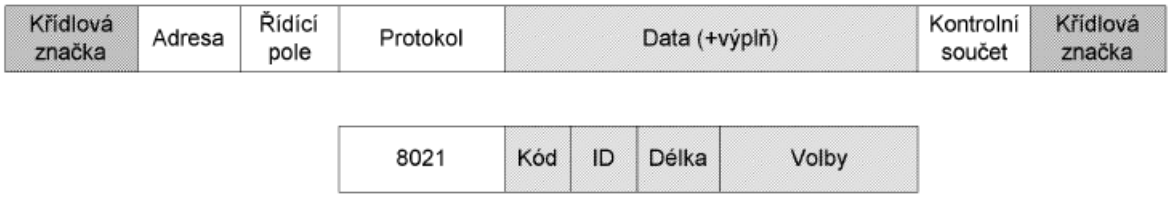
Jako protokol mezi přístupovým serverem a serverem s autentizačními a konfiguračními informacemi se dnes často používá protokol RADIUS nebo protokol TACAC+. ProtokolRADIUS je aplikační protokol.

Obr. 4.17
Radius a radius accounting protokol



Protokol IPCP (NCP pro IPv4)

Protokol IPCP je protokol NCP pro protokol IP verze 4. Formát rámce IPCP protokolu je vyjádřen na následujícím obrázku:



Kód - Osmibitové pole kód specifikuje typ příkazu (resp. odpovědi) protokolu IPCP:

ID - Osmibitové pole **ID** je identifikace požadavku. Odesílatel výplní nějaký údaj do tohoto pole a adresát jej zkopíruje do své odpovědi. Pomocí tohoto pole se určuje příslušnost odpovědi k požadavku.

Délka - Šestnáctibitové pole **délka** obsahuje číslo udávající součet délek polí: kód, ID, délka a volby.

Volby- Pole **volby** obsahuje jednotlivé požadavky (resp. odpovědi) na změnu implicitních parametrů linky. Toto pole se skládá z jedné nebo více voleb. Jednotlivé volby jsou ukládány sekvenčně za sebou.Pole volba a délka jsou jednobajtové, jejich formát je obdobný formátu voleb protokolu LCP

Příklad:

kód	Název kódu (anglicky)	Význam
2	IP-Compression-Protocol	Komprese TCP/IP záhlaví (viz protokol SLIP). Pole data obsahuje šestnáctkově 002d. Délka je 6, protože další 2 bajty obsahují parametry komprese.
3	IP Adress	Předání IP adresy protějšku. Takto je možno dynamicky přidělovat IP-adresy. Chce-li protějšek použít jinou IP-adresu, pak odpoví paketem Configure-Nak, kde tuto adresu specifikuje. Pole data obsahuje čtyřbajtovou IP-adresu a délka je 6.
129	Primary-DNS-Address	Specifikace primárního jmenného serveru pole data obsahuje (RFC-1877) čtyřbajtovou IP-adresu primarního jmenného serveru, délka je 6.
131	Secondary-DNS-Address	Specifikace sekundárního jmenného serveru, pole data obsahuje (RFC-1877) čtyřbajtovou IP-adresu primárního jmenného serveru, délka je 6.

V rámci protokolu PPP se pro přenos datových paketů s protokolem IP verze 4 použije identifikace protokolu 0021_{16}. V případě komprese je situace komplikovanější, protože ne všechny pakety mají komprimované záhlaví (ne všechny IP pakety nesou protokol TCP, např. pakety ICMP se nekomprimují). Je tedy nutné rozlišovat v přenášených paketech pakety s komprimovaným TCP/IP záhlavím a pakety s nekomprimovaným záhlavím. Proto v záhlaví PPP-rámce v poli protokol mají nekomprimované pakety identifikaci 0021_{16} a pakety s komprimovaným IP-záhlavím identifikaci $002d_{16}$.

Příklad rámců protokolu IPCP (Configure Request a Configure Ack).
odchycených MS Network Monitorem ve Windows 2000:

Frame: Base frame properties
Frame: Time of capture = 3/15/2000 9:39:21.948
Frame: Time delta from previous physical frame: 0 microseconds
Frame: Frame number: 37
Frame: Total frame length: 42 bytes
Frame: Capture frame length: 42 bytes
Frame: Frame data: Number of data bytes remaining = 42 (0x002A)
PPP: Unknown Frame (0x0)
PPP: Destination Address = SEND_

PPP: Source Address = SEND_
PPP: Protocol = Internet Protocol Control Protocol
IPCP: Configuration Request, Ident = 0x07
IPCP: Code = Configuration Request
IPCP: Identifier = 7 (0x7)
IPCP: Length = 28 (0x1C)
IPCP: Option: Compression Protocol = 0x002D (Van Jacobson Compressed TCP/IP)
IPCP: Option Type = Compression Protocol
IPCP: Option Length = 6 (0x6)
IPCP: Compression Protocol = Van Jacobson Compressed TCP/IP
IPCP: Max Slot ID = 15 (0xF)
IPCP: Comp Slot ID = Slot Identifier may be compressed
IPCP: Option: Address = 195.47.37.205
IPCP: Option Type = Address
IPCP: Option Length = 6 (0x6)
IPCP: Source Address = 195.47.37.205
IPCP: Option: Primary DNS Server Address = 194.149.105.18
IPCP: Option Type = Primary DNS Server Address
IPCP: Option Length = 6 (0x6)
IPCP: Primary DNS Server Address = 194.149.105.18
IPCP: Option: Secondary DNS Server Address = 194.149.103.201
IPCP: Option Type = Secondary DNS Server Address
IPCP: Option Length = 6 (0x6)
IPCP: Secondary DNS Server Address = 194.149.103.201
00000: 20 53 45 4E 44 07 20 53 45 4E 44 07 80 21 01 07 SEND. SEND.^!..
00010: 00 1C 02 06 00 2D 0F 01 03 06 C3 2F 25 CD 81 06 .....-....Ă/%ÍL.
00020: C2 95 69 12 83 06 C2 95 67 C9 Â•i.ı.Â•gÉ
****

Frame: Base frame properties
Frame: Time of capture = 3/15/2000 9:39:22.69
Frame: Time delta from previous physical frame: 120172 microseconds
Frame: Frame number: 38
Frame: Total frame length: 42 bytes
Frame: Capture frame length: 42 bytes
Frame: Frame data: Number of data bytes remaining = 42 (0x002A)
PPP: Unknown Frame (0x0)
PPP: Destination Address = RECV_
PPP: Source Address = RECV_
PPP: Protocol = Internet Protocol Control Protocol
IPCP: Configuration Acknowledgement, Ident = 0x07
IPCP: Code = Configuration Acknowledgement
IPCP: Identifier = 7 (0x7)
IPCP: Length = 28 (0x1C)
IPCP: Option: Compression Protocol = 0x002D (Van Jacobson Compressed TCP/IP)
IPCP: Option Type = Compression Protocol
IPCP: Option Length = 6 (0x6)
IPCP: Compression Protocol = Van Jacobson Compressed TCP/IP
IPCP: Max Slot ID = 15 (0xF)
IPCP: Comp Slot ID = Slot Identifier may be compressed
IPCP: Option: Address = 195.47.37.205
IPCP: Option Type = Address
IPCP: Option Length = 6 (0x6)
IPCP: Source Address = 195.47.37.205
IPCP: Option: Primary DNS Server Address = 194.149.105.18
IPCP: Option Type = Primary DNS Server Address
IPCP: Option Length = 6 (0x6)
IPCP: Primary DNS Server Address = 194.149.105.18
IPCP: Option: Secondary DNS Server Address = 194.149.103.201
IPCP: Option Type = Secondary DNS Server Address
IPCP: Option Length = 6 (0x6)
IPCP: Secondary DNS Server Address = 194.149.103.201
00000: 20 52 45 43 56 07 20 52 45 43 56 07 80 21 02 07 RECV. RECV.^!..
00010: 00 1C 02 06 00 2D 0F 01 03 06 C3 2F 25 CD 81 06 .....-....Ă/%ÍL.
00020: C2 95 69 12 83 06 C2 95 67 C9 Â•i.ı.Â•gÉ