# COIPV4

## KNIHOVNA PROTOKOLU IP

Příručka uživatele a programátora


***SofCon*®spol. s r.o.**
Střešovická 49
162 00 Praha 6
tel/fax: +420 220 180 454
E-mail: sofcon@sofcon.cz
www: http://www.sofcon.cz

Verze dokumentu 1.10

Informace v tomto dokumentu byly pečlivě zkontrolovány a SofCon věří, že jsou spolehlivé, přesto SofCon nenese odpovědnost za případné nepřesnosti nebo nesprávnosti zde uvedených informací.
SofCon negarantuje bezchybnost tohoto dokumentu ani programového vybavení, které je v tomto dokumentu popsané. Uživatel přebírá informace z tohoto dokumentu a odpovídající programové vybavení ve stavu, jak byly vytvořeny a sám je povinen provést validaci bezchybnosti produktu, který s použitím zde popsaného programového vybavení vytvořil.
SofCon si vyhrazuje právo změny obsahu tohoto dokumentu bez předchozího oznámení a nenese žádnou odpovědnost za důsledky, které z toho mohou vyplynout pro uživatele.

Datum vydání: 16.05.2003
Datum posledního uložení dokumentu: 16.05.2003
(Datum vydání a posledního uložení dokumentu musí být stejné)

# Obsah :

# 1. O dokumentu

## 1.1. Revize dokumentu

| Verze dokumentu | Verze SW | Autor | Datum vydání | Popis změn |
|---|---|---|---|---|
| 1.00 | 1.XX | Čr | | První vydání |
| 1.10 | 1.XX | Tu | 16.05.2003 | Úprava dokumentu dle ISO9000 |
| | | | | |

## 1.2. Účel dokumentu

Tento dokument slouží jako popis jednotky implementující protokol IP.

## 1.3. Rozsah platnosti

Určen pro programátory a uživatele programového vybavení SofCon.

## 1.4. Související dokumenty

Pro čtení tohoto dokumentu je potřeba seznámit se s manuálem CoBase.

Popis formátu verze knihovny a souvisejících funkcí je popsán v manuálu LibVer.

# 2. Termíny a definice

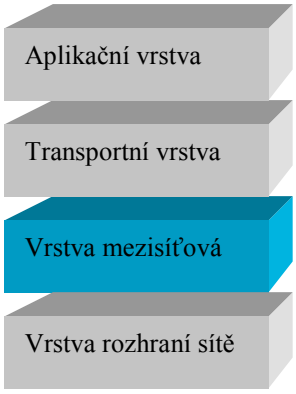
Používané termíny a definice jsou popsány v samostatném dokumentu Termíny a definice.

# 3. Úvod

## 3.1. Účel knihovny CoIPv4

Knihovna **CoIPv4** implementuje mezisíťovou vrstvu rodiny architektury TCP/IP. Mezisíťová vrstva je implementována pomocí zařízení TCoIPv4. Stručný popis architektury TCP/IP je uveden v dokumentaci ke knihovně CoBase.

**TCP/IP**




## 3.2. Funkce mezisíťové vrstvy

Mezisíťová vrstva TCP/IP odpovídá svými funkcemi síťové vrstvě referenčního modelu OSI. Mezisíťová vrstva zajišťuje především směrování a přepojování datagramů přes komunikační podsíť. Mezisíťová vrstva je tvořena následujícími protokoly:

- **Protokol intersítě** (Internet Protocol, IP)
- **Protokol mapování adres** (Address Resolution Protocol, ARP)
- **Protokol zpětného mapování adres** (Rerverse Address Resolution Protocol, RARP)
- **Protokol řídících hlášení** (Internet Control Message Protocol, ICMP)
- **Protokol správy skupin** (Internet Group Management Protocol, IGMP)
- **Směrovacími protokoly**

Funkce protokolu RARP byly úspěšně nahrazeny aplikačními protokoly BOOTP (Boot Protocol), příp. DHCP (Dynamic Host Configuration Protocol). V současnosti není nezbytně potřeba, aby aplikace stavebnice KIT umožňovaly komunikaci založenou na skupinových adresách, kterou podporuje protokol IGMP.

V další části textu budou rozebrány pouze implementované protokoly. Protokol RARP, IGMP ani směrovací protokoly nebudou zmiňovány.

## 3.2.1. IP protokol

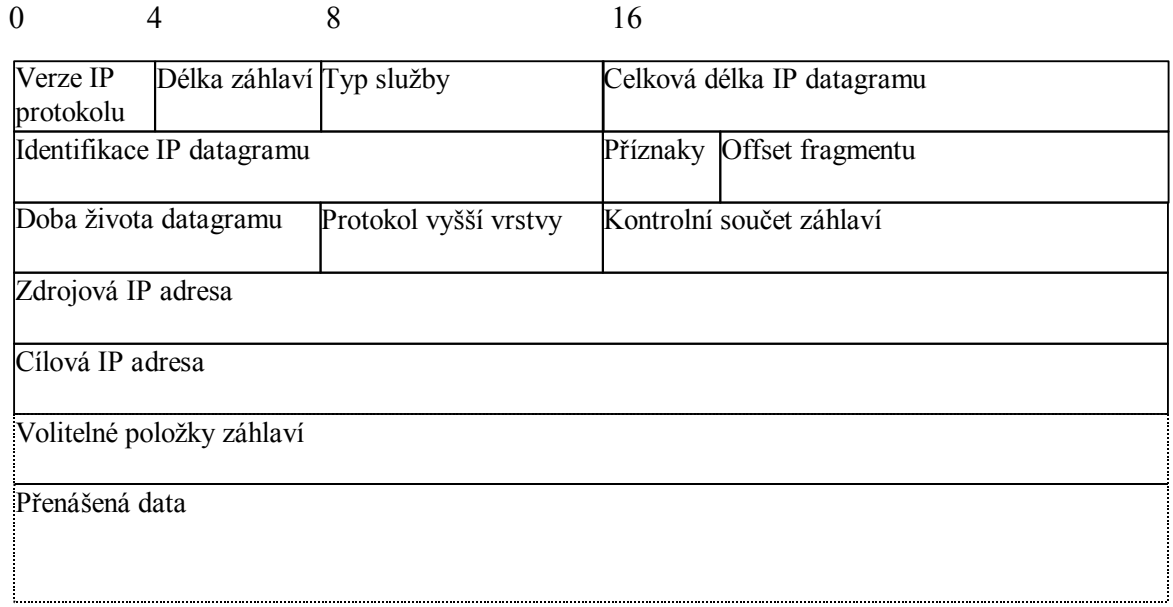
V současné době existují dvě verze protokolu IP: verze 4 a verze 6. Specifikace IP protokolu verze 6 existuje již několik let, existují i jeho implementace, nicméně prozatím nebyl oficiálně schválen, a tím pádem se prakticky nepoužívá. Z tohoto důvodu se budeme dále zabývat pouze protokolem verze 4, jehož zrod se datuje k začátku 80. let dvacátého století.

Síťový protokol IP poskytuje službu bez spojení, tzn. že nezaručuje bezpečné doručení dat. Architektura TCP/IP nepoužívá na mezisíťové vrstvě žádný spolehlivý protokol. Bezpečné doručení dat zajišťují protokoly transportní nebo aplikační vrstvy. IP protokol provádí především odesílání a příjem datagramů. V terminologii TCP/IP se místo pojmu paket používá spíše pojmu datagram. Každý datagram je samostatná jednotka opatřená adresou odesílatele a adresou příjemce. IP protokol dále provádí fragmentaci a znovusestavování datagramu do/z tzv. fragmentů, pokud je maximální povolená velikost datové jednotky v lokální síti menší než je celková délka datagramu. Fragmenty jsou pro tento účel číslovány podle pořadí v původním datagramu.

### 3.2.1.1. Struktura IP datagramu

Každý odesílaný datagram je opatřen záhlavím, jehož struktura je zobrazena na obrázku níže. Jeden řádek obrázku reprezentuje 32 bitů. Celková délka záhlaví může kolísat v rozmezí 20 až 60 oktetů podle velikost volitelných položek záhlaví.

**Obrázek 3.1 Struktura IP datagramu**




**Verze IP protokolu** Tato položka má 4 bity a obsahuje číslo verze IP protokolu. Položka obsahuje vždy hodnotu 4 (IP verze 4)

**Délka záhlaví** Tato položka obsahuje délku záhlaví (4bity). Je uváděna

| | |
|---|---|
| | ve násobcích čtyřech oktetů. Vzhledem k počtu bitů na položku je maximální délka záhlaví 64 oktetů. Běžně se však využívá minimální délka záhlaví 20 oktetů. |
| **Typ služby** | Tato položka určuje, jak má datagram zpracovávat protokol vyšší vrstvy z hlediska priorit a zpoždění. Tato položka v praxi nenašla zatím rozumné uplatnění, víceméně se nevyužívá a obvykle je nulová. |
| **Doba života datagramu** (TTL – Time To Live) | Slouží k zamezení nekonečného šíření paketů Internetem. Vysílající stanice nastaví tuto položku na jistou hodnotu (např. 64, 128). Každý směrovač na cestě sníží položku alespoň o 1. V okamžiku, kdy se její hodnota sníží na nulu, je datagram bezpodmínečně zahozen. |
| **Identifikace IP datagramu** | Identifikace datagramu je číslo, které vyplní síťová vrstva odesílatele datagramu. Jedná se o jedinečné číslo a spolu s položkami příznaky a posunutí fragmentu je využívána mechanismem fragmentace a znovusestavování datagramu. Více v kapitole 3.2.1.6. |
| **Protokol vyšší vrstvy** | Obsahuje číselnou identifikaci protokolu vyšší vrstvy, jemuž se příchozí paket doručí. V praxi se nesetkáváme s tím, že by se komunikovalo přímo pomocí protokolu IP. Vždy je použit nějaký protokol vyšší vrstvy (UDP nebo TCP). |
| **Kontrolní součet záhlaví** | Obsahuje kontrolní součet, avšak pouze záhlaví IP datagramu. |
| **Volitelné položky záhlaví** (Options) | Dovolují doplnit datagram o další informace: např. pro zabezpečení, záznam cesty sítí nebo dodržení předepsané cesty sítí. Vzhledem ke složitosti zpracování datagramů obsahujících tyto informace se toto pole většinou nepoužívá. |
| **Přenášená data** | Maximální velikost přenášených dat spolu s záhlavím je 64kB. Podle RFC-791 by však celková délka datagramu neměla přesáhnout 576 bajtů, tj. 512 bajtů dat a 64 bajtů pro záhlaví. Každá stanice v síti musí zaručit příjem datagramu této délky. Pokud dvě stanice chtějí komunikovat pomocí delších datagramů, musí se předem navzájem ujistit, že toto lze, pomocí mechanismů na vyšších protokolových vrstvách. Např. transportní protokol TCP k tomuto účelu obsahuje ve volitelné části záhlaví položku MSS, viz. dokumentace ke knihovně CoTCP. |

### 3.2.1.2. Volitelné položky záhlaví

Volitelné položky záhlaví slouží k upřesnění parametrů odesílaného datagramu. Datagram nemusí obsahovat žádnou volitelnou položku. Součástí volitelné položky je její typ a délka (pokud se nejedná o standardní položku fixní délky). Všechny implementace IP protokolu musí podporovat příjem volitelných položek záhlaví, a jestliže se v záhlaví vyskytne neznámá položka, není jednoduše zpracována. Pomocí volitelných položek lze upřesnit např. zabezpečení datagramu, přinutit směrovače na cestě datagramu, aby dodržely předepsanou cestu, zaznamenávat cestu datagramu

nebo zaznamenat čas odeslání datagramu. Volitelné položky nejsou nezbytně nutné pro funkci IP protokolu a dále si jimi nebudeme zabývat.

### 3.2.1.3. IP adresa

Síťové adresy IP verze 4 jsou 32-bitové a obvykle jsou zapisovány jako čtveřice desítkových čísel oddělených tečkou. (např. 192.168.1.1). Adresa má dvě části:

- **Adresu sítě**, což je první část adresy přidělená správcem IP adres (zpravidla poskytovatelem přístupu k Internetu).
- **Adresu uzlu v síti**, což je poslední část adresy přidělená správcem sítě.

Adresy mohou být tvořeny podle jednoho z následujících pěti formátů:

- **Třída A** používá první bajt adresy pro identifikaci sítě a zbývající tři bajty pro určení adresy uzlu v síti. Tato třída poskytuje největší rozsah adres stanic v rámci jedné sítě, ale počet sítí je omezen na 126. Získaní adresy třídy A je dnes prakticky nemožné.
- **Třída B** používá první dva bajty adresy pro identifikaci sítě a zbývající dva bajty pro určení adresy uzlu v síti. Adresy třídy B jsou již téměř rozděleny mezi poskytovatele přístupu k Internetu.
- **Třída C** používá první tři bajty adresy pro identifikaci sítě a zbývající bajt na určení uzlu v síti. Adresy třídy C poskytují minimální prostor pro adresování uzlů v síti. Dnes jsou prakticky jedinými dostupnými adresami pro přímé propojení k Internetu.
- **Třída D** (na rozdíl od adres A, B a C, které jsou určeny pro adresaci jednotlivých stanic v Internetu,) slouží pro skupinovou adresaci.
- **Třída E** je určena pro experimentální účely.

| | Struktura adresy (S=síť,U=uzel) | První bity adresy dvojkově | Platné hodnoty prvního bajtu | Počet adres sítí | Počet uzlů adresovatelných v rámci sítě |
|---|---|---|---|---|---|
| Třída A | S.U.U.U | 0… …. | 1 – 126 | 126 | $2^{24} – 2$ |
| Třída B | S.S.U.U | 10.. …. | 128 – 191 | $2^{14}$ | $2^{16} – 2$ |
| Třída C | S.S.S.U | 110. …. | 192 – 223 | $2^{21}$ | $2^{8} – 2$ |
| Třída D | - | 1110 …. | 224 – 239 | - | - |
| Třída E | - | 1111 …. | 240 – 255 | - | - |

Všechny platné adresy v rámci jedné sítě neoznačují stanice, protože v adresovém prostoru IP adres jsou některé adresy vyhrazeny pro speciální účely:

- 0.0.0.0 „Tato“ stanice na „této“ síti. Počítač např. ještě nezná svou adresu, ale přesto musí komunikovat např. při zavádění systému.
- S.0 V části identifikace stanice jsou dvojkově 0 (např. adresa 192.168.1.0 neoznačuje stanici, ale síť třídy C).
- 0.U V časti identifikace sítě jsou dvojkově 0 (např. 0.0.0.5

označuje stanici na této síti, pokud stanice nezná adresu sítě)

- 255.255.255.255 Všeobecná lokální adresa (broadcast), tuto adresu obsahují všeobecné oběžníky. Tyto oběžníky nejsou směrovači propouštěny do jiných sítí.
- S.255 Všeobecná adresa v síti, v části identifikace stanice jsou dvojkově 1. Adresou jsou specifikovány všechny stanice na konkrétní síti.

### 3.2.1.4. Podsíťování a síťová maska

Vzhledem k neefektivnímu rozdělení adres, především nedostatku adres stanic pro adresy třídy C a naopak nevyčerpatelnému přebytku adres uzlů ve třídě A, se přistoupilo k mechanismu tzv. podsíťování (subnetting). Část adresy původně určená stanici v rámci sítě se rozdělí na dvě části: adresu podsítě a adresu stanice.
Jakmile se používá podsíťování, je třeba přesně rozlišovat mezi adresou sítě, podsítě a uzlu v rámci podsítě. Zatímco v případě adresace bez použití podsíťování bylo při pohledu na první bajt adresy zřejmé, jaká je její struktura, pak s podsíťováním to již tak jednoduché není. Pro určení adresy uzlu musí být IP adresa doplněna tzv. podsíťovou (síťovou) maskou. Maska má stejný formát jako adresa. Na pozicích bitů označujících adresu sítě a adresu podsítě je hodnota bitů masky 1 a na pozicích sloužících pro adresaci uzlu v rámci podsítě má maska hodnotu 0. Implicitní masky bez použití podsíťování mají následující hodnoty:

- třída A – implicitní maska 255.0.0.0
- třída B – implicitní maska 255.255.0.0
- třída C – implicitní maska 255.255.255.0

### 3.2.1.5. Směrování

Pro každý odesílaný datagram musí odesílatel zjistit následující stanici na cestě. Pokud se cílová stanice nachází ve stejné lokální síti, což odesílatel rozpozná na základě adresy stanice, masky podsítě a adresy sítě, je situace triviální. V případě, že se cílová stanice nachází mimo lokální síť, pak je potřeba datagram předat vhodnému směrovači, což je stanice ve stejné lokální síti, která přepošle datagram dále.
Ke zjištění následující stanice na cestě datagramu slouží tzv. směrovací tabulka (Route Cache). Směrovací tabulka obsahuje minimálně následující informace:

- Lokalní IP adresu (pro stanice s více síťovými rozhraními)
- Cílovou IP adresu společně s maskou podsítě.
- IP adresu příští stanice na cestě

Do směrovací tabulky nejsou zanášeny stanice na lokální síti. Směrovací tabulku lze plnit staticky, ale i dynamicky na základě příchozích ICMP zpráv od směrovačů (viz. Kapitola 3.2.2). Obvykle je ve směrovací tabulce uvedena i položka pro tzv. implicitní směrovač (gateway), na jehož adresu jsou posílány všechny datagramy mimo lokální síť, pokud pro danou cílovou adresu není ve směrovací tabulce uvedena explicitně jiná následující stanice.

### 3.2.1.6. Fragmentace a znovusestavení IP datagramů

Teoreticky může mít IP datagram velikost až 64kB. Maximální velikost dat rámce Ethernetu je přibližně 1500 oktetů. Datagramy, delší než maximální velikost rámce na spojové vrstvě, je potřeba na vysílací straně rozdělit na menší fragmenty a následně na přijímací straně sestavit do původního datagramu. K tomu slouží mechanismus fragmentace a znovusestavení. Stanice připojené k síti Ethernet obvykle neposílají takové datagramy, které by bylo potřeba fragmentovat. Posílání fragmentovaných datagramů může v případě ztráty jednoho nebo více fragmentů vést k nadměrné paměťové zátěži příjemce. K fragmentaci dochází spíše ve směrovačích, kdy se přechází ze sítě s větší do sítě s menší maximální velikostí rámce (MTU) na spojové vrstvě. Takovou fragmentaci lze ale explicitně zakázat nastavením bitu DF (Don't Fragment) v položce **Offset fragmentu** záhlaví IP datagramu.
Odesílatel rozdělí datagram na fragmenty o maximální velikosti. Poslední fragment může být samozřejmě menší. Všechny fragmenty jednoho datagramu musí mít v záhlaví uvedeno stejné identifikační číslo (**Identifikace IP datagramu**). Pořadí fragmentu v datagramu je dáno položkou **Ofset fragmentu**. Všechny fragmenty jsou postupně odeslány.
Příjemce dostává postupně jednotlivé fragmenty. To, že příchozí datagram je fragment většího datagramu, rozpozná podle obsahu položky **Ofset datagramu**. Fragmenty jednoho datagramu mají stejné identifikační číslo. Fragmenty nemusí vždy přicházet ve správném pořadí a proto by měl příjemce fragmenty po určitou dobu držet v paměti (obvykle se jedná o dvě minuty) a až v okamžiku, kdy má k dispozici všechny fragmenty datagramu, předá výsledný datagram vyšší vrstvě. Pokud vyprší časový limit pro sestavení datagramu, musí být všechny přijaté fragmenty zahozeny.

## 3.2.2. ICMP protokol

ICMP protokol slouží k přenosu řídících hlášení týkajících se chyb a zvláštních okolností při přenosu datagramů. ICMP zprávy se vysílají v situaci, kdy je síť nebo stanice zahlcená nebo má nějaký jiný problém. Mechanismus odesílání ICMP zpráv je koncipován tak, aby tyto zprávy samy nepřispívaly k dalšímu komplikování situace. ICMP zprávy se proto negenerují v souvislosti s problémy datagramů vysílaných na všeobecnou nebo skupinovou adresu, neposílají se jako reakce na problémy s jinými ICMP zprávami apod. Úplná specifikace ICMP protokolu je uvedena v RFC-732 některá další rozšíření v RFC-1122.

Formát ICMP zprávy je uveden na následujícím obrázku. ICMP zprávy se zapouzdřují do datové části IP datagramu. Položka **Protokol vyšší vrstvy** záhlaví IP datagramu obsahuje hodnotu 1.

| Typ zprávy | Kód (8b) | Kontrolní součet (16b) | Data zprávy |
|---|---|---|---|

**Typ zprávy** Specifikuje typ zprávy (viz tabulka 1.2)

**Kód** Je součástí parametrů zprávy. Dalo by se říct, že se jedná o podtyp zprávy.

| | |
|---|---|
| **Kontrolní součet** | Zabezpečuje obsah zprávy a pokrývá celou ICMP zprávu včetně záhlaví. |
| **Data zprávy** | Obsah zprávy závisí na daném typu zprávy. |

Následující tabulka ukazuje nejdůležitější typy ICMP zpráv. Tyto zprávy je téměr bezpodmínečně nutné implementovat.

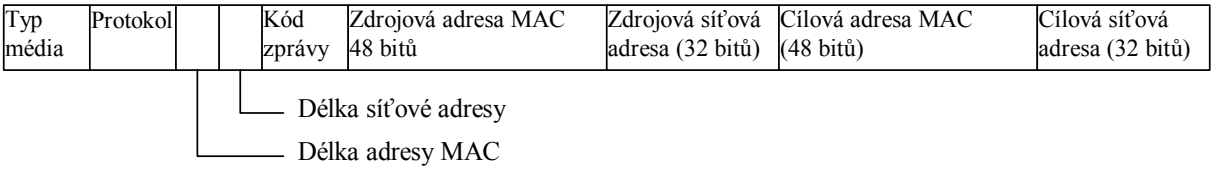
| Typ zprávy | Obsah zprávy |
|---|---|
| **Echo reply** 0 | Odpověď na žádost o odezvu. Generuje se v souvislosti se zprávou Echo Request (8). |
| **Destination Unreachable** 3 | Cíl je nedostupný, protože cílový port, protokol, stanice nebo síť jsou nedostupné nebo neznámé. Případně byla nutná fragmentace, ale bit DF v datagramu ji nepovolil, a v důsledku toho byl fragment směrovačem na cestě zahozen. Konkrétní důvod určuje položka **Kód** záhlaví zprávy. |
| **Source Quench** 4 | Tuto zprávu posílá stanice nebo směrovač na cestě datagramu, pokud zahodil datagram z důvodu nedostatku paměti vyhrazené pro příjem. Stanice, které je zpráva adresována, by měla pokud možno snížit frekvenci odesílání datagramu. |
| **Redirect** 5 | Přesměrování. Tuto zprávu generuje směrovač na lokální síti, pokud obdržel datagram, který je určen pro stanici mimo tuto síť a na lokální síti je ještě jiný směrovač s kratší cestou k cílové stanici. Příjem této zprávy vede obvykle k modifikaci směrovací tabulky zdrojové stanice. |
| **Echo Request** 8 | Žádost o odezvu. Cílová stanice odpovídá zprávou Echo Reply (8). |
| **Time Exceeded** 11 | Životnost datagramu překročena, dosáhla nuly nebo vypršel časový limit čekaní na zbývající fragmenty téhož datagramu. |
| **Parameter Problem** 12 | Chyba v položkách záhlaví datagramu. Součástí zprávy je ukazatel na problémovou část záhlaví. |

Všechny ICMP zprávy, uvedené v tabulce výše (kromě Echo Request a Echo Reply), obsahují v datové části IP záhlaví a minimálně 8 oktetů záhlaví datagramu transportního protokolu, ke kterému se zpráva vztahuje. ICMP zprávu je tak možné v cílové stanici přiřadit ke konkrétnímu transportnímu, příp. aplikačnímu protokolu.

### 3.2.3. ARP protokol

Protokol ARP je používán pro zjištění fyzické adresy rozhraní (MAC adresy) při znalosti cílové IP adresy stanice. Úplná specifikace protokolu ARP s ohledem na síť Ethernet je obsažena v RFC-826.
Každá stanice si udržuje v paměti tabulku (tzv. ARP cache), ve které je kromě dalších položek uvedena dvojice: fyzická adresa (MAC adresa) a odpovídajícími IP adresa. Před každým vysíláním mezisíťová vrstva tuto tabulku prohledává, a pokud zde nalezne síťovou adresu odesílaného datagramu, datagram odešle na fyzické rozhraní příslušejících této adrese uvedené v tabulce. V opačném případě se pokusí MAC

adresu rozhraní zjistit pomocí protokolu ARP.
Na následujícím obrázku je uveden formát rámce protokolu ARP:




Zdrojová stanice vyšle do lokální sítě jako oběžník žádost (ARP dotaz, ARP Request) tj. neúplně vyplněný ARP rámec, v němž není vyplněno pouze pole **Cílová adresa MAC**.
Stanice se síťovou adresou specifikovanou v žádosti odpoví podobnou zprávou (s jiným **kódem zprávy**), ve které zamění pole **Zdrojová síťová adresa** a **Zdrojová MAC adresa** za obsahem polí **Cílová síťová** adresa a **Cílová MAC** adresa. Pole **Zdrojová MAC adresa** vyplní adresou svého vlastního fyzického rozhraní. Přijatá odpověď příslušné stanice (tzv. ARP Reply) je zanesena jako jeden řádek do ARP cache a pozdržený datagram je odeslán. Pokud v lokální síti není žádná stanice s požadovanou síťovou adresou, vyslaný ARP dotaz zůstane bez odpovědi. Síťová vrstva se může pokusit o opakování dotazu. Po vyčerpání určitého počtu pokusů hlásí chybu vyšším protokolovým vrstvám.
Každý řádek v ARP cache má určitou životnost (obvykle dvě minuty), po této době je odstraněn, a pokud je i nadále potřeba MAC adresa stejné stanice, musí se znovu zjistit vysláním ARP dotazu. Tímto mechanismem je zaručeno zotavení komunikace po změně MAC adres, např. po výměně hardware nebo změně síťové adresy stanice.

# 4. Konstanty a jednoduché typy

## 4.1. Konstanty

### 4.1.1. Návratové kódy CST_IPV4_xxx

Současná verze knihovny CoIPv4 nedefinuje žádné vlastní návratové kódy CST_IPV4_.

Návratový kód je možné přeložit pomocí funkce a metody **CoStatusToStr** na textový řetězec.

### 4.1.2. Identifikátory tříd zařízení DEV_CLASS_xxx

Třída TCoIPv4 (protokol IP) má přidělen identifikátor **DEV_CLASS_IPV4**.
Tento identifikátor určuje zařízení v řetězci při volání funkcí CoIoctl, CoGetOption, CoSetOption apod. Zaregistrované jméno třídy pro použití s funkcí CoCreateDevice je **IPV4**.

### 4.1.3. Identifikátory řídících operací IOCTL_IPV4_xxx

Identifikátory s prefixem **IOCTL_** specifikují operaci prováděnou funkcí **CoIoctl** v řetězci zařízení. Knihovna CoIPv4 definuje několik nových konstant s prefixem IOCTL_IPV4_.

**IOCTL_IPV4_GETIPADDRMASK**
: Pomocí tohoto příkazu lze zjistit aktuální IP adresu a masku síťového adaptéru. Po provedení příkazu budou aktuální IP adresa a maska překopírovány do připraveného bufferu.

  Formát dat: TIpIoctlAddrMask

**IOCTL_IPV4_RESETCOUNTERS**
: Nulování čítačů událostí. Použijte tento příkaz k vynulování všech čítačů událostí.

  Formát dat: žádná data se nepředávají

**IOCTL_IPV4_GETCOUNTERS**
: Čtení stavu čítačů událostí. Po provedení příkazu je naplněna předaná struktura stavem všech čítačů událostí. viz. kapitola 4.2.6.

  Formát dat: TIPCounters

**IOCTL_IPV4_ICMP_DESTUNREACH**
: Odeslání ICMP zprávy DESTINATION UNREACHABLE. Tento příkaz je určen pro vyšší transportní vrstvy.

Formát dat: TIoctlIcmpDestUnreach

### 4.1.4. Identifikátory parametrů zařízení COPT_IPV4_xxx

Identifikátory s prefixem **COPT_** specifikují nastavovaný příp. vyčítaný parametr zařízení pomocí funkce **CoSetOption** příp. **CoGetOption**. Knihovna CoIPv4 definuje několik nových konstant s prefixem COPT_IPV4_:

**COPT_IPV4_IPADDR**
IP adresa síťového adaptéru. IP adresu lze měnit pouze tehdy, pokud je zařízení v neaktivním stavu.

Formát dat: TIPAddress Implicitní hodnota: 192.168.1.64

**COPT_IPV4_NETMASK**
Síťová maska sítě, ve které je síťový adaptér umístěn. Síťovou masku lze měnit pouze tehdy, pokud je zařízení v neaktivním stavu.

Formát dat: TIpAddress Implicitní hodnota: 255.255.255.0

**COPT_IPV4_DEFGW**
Implicitní brána. Na tuto adresu jsou směrovány všechny pakety jdoucí mimo lokální síť, pokud ve směrovací tabulce není pro danou stanici či síť určeno jinak. Neplatná adresa 0.0.0.0 zruší implicitní směrování. Implicitní bránu lze měnit pouze tehdy, pokud je zařízení v neaktivním stavu.

Formát dat: TIpAddress Implicitní hodnota: 0.0.0.0

**COPT_IPV4_TTL**
Hodnota položky TTL, která se doplní do každé hlavičky odchozího IP datagramu.

Formát dat: Byte Implicitní hodnota: 64 (Min: 0, Max: 255)

**COPT_IPV4_ECHO**
Povolení odpovídání na ICMP zprávu ECHO REQUEST. Pokud je tato služba zakázaná, nebude fungovat příkaz **ping** směrovaný na tento síťový adaptér.

Formát dat: Boolean Implicitní hodnota: True

**COPT_IPV4_ARPTO**
Doba platnosti položky v ARP cache v sekundách. Po uvedené době je položka z ARP cache vyjmuta. Doba platnosti položky se udává v sekundách.

Formát dat: Word Implicitní hodnota: 120s (min: 5s, max: 10000s)

**COPT_IPV4_ARPRETRY**
Počet pokusů o zjištění MAC adresy ARP dotazem, pokud cílová stanice neodpovídá. Dotaz je opakován vždy po jedné sekundě.

Formát dat: Word Implicitní hodnota: 3s (min: 1s, max: 100s)

**COPT_IPV4_REASMTO**
Doba platnosti fragmentů čekajících na složení do datagramu. Po této nastavené době jsou neúplné příchozí datagramy zahozeny. Doba platnosti se udává v sekundách.

Formát dat: Word Implicitní hodnota: 16s (min: 1s)

**COPT_IPV4_MTUR**
Maximální délka datagramu, který lze přijmout. Delší datagramy jsou automaticky zahazovány.

Formát dat: Word Implicitní hodnota: 2048
(min: 1024, max: 16384)

### 4.1.5. Identifikátory transportních protokolů IP_PROTO_xxx

Knihovna definuje následující identifikátory protokolů vyšší vrstvy. Protokoly UDP a TCP jsou implementovány knihovnami CoUdp a CoTcp. Protokol ICMP je implementován přímo touto knihovnou.

**IP_PROTO_ICMP** = 1
**IP_PROTO_TCP** = 6
**IP_PROTO_UDP** = 17

### 4.1.6. Identifikátory IP_ADDR_xxx

Knihovna definuje dvě standardní síťové adresy (viz. kapitola 3.2.1.3)

**IP_ADDR_ANY** = TIPAddress( $00000000 )
**IP_ADDR_BROADCAST** = TIPAddress( $FFFFFFFF )

### 4.1.7. Identifikátory ICMPDU_xxx

Konstanty ICMPDU_ slouží k upřesnění ICMP zprávy DESTINATION UNREACHABLE. Viz. struktura TIoctlIcmpDestUnreach.

**ICMPDU_PROTO_UNREACHABLE** = 2
Transportní protokol není na cílové stanici dostupný.
**ICMPDU_PORT_UNREACHABLE** = 3
Transportní vrstva nemůže demultiplexovat příchozí datagram, protože je určen pro nezaregistrovaný port.

## 4.2. Typy

### 4.2.1. Třída TCoEth01

```
  TCoEth01 = object( TCoDevice );
```

Pokud použijeme knihovnu CoEth01 (uvedeme ji někde v seznamu za klíčovým slovem **uses**), bude tato třída zaregistrovaná v globálním seznamu tříd zařízení pod jménem **ETH01**. Toto jméno lze použít v případě vytváření třídy funkcí **CoCreateDevice**. Číselný identifikátor třídy je **DEV_CLASS_NIC** (Network Interface Card).

### 4.2.2. Struktura TIpAddress

```
PIpAddress = ^TIpAddress;
TIpAddress = Longint;
```

Struktura TIpAddress specifikuje IP adresu stanice. Pro vytvoření IP adresy je možné použít dva postupy:

```
var
   IP: TIpAddress;

IP := $C0A80101;
IP := MakeIpAddr( 192, 168, 1, 1 ); { Přehlednější zápis }
```

Pro zobrazení adresy IP adresy lze použít funki **IpAddrToStr** (převádí IP adresu na textový řetězec), při zadávání IP adresy, lze použít funkci **StrToIpAddr** (převádí textový řetězec na IP adresu).

### 4.2.3. Struktura TIpIoctlAddrMask

```
TIpIoctlAddrMask = record
  Address : TIpAddress;
  NetMask : TIpAddress;
end;
```

Struktura **TIpIoctlAddrMask** slouží k předání síťové adresy a masky síťového adaptéru pomocí metody **CoIoctl**. Viz. kapitola 4.1.3 (identifikátor IOCTL_IPV4_GETIPADDRMASK).

### 4.2.4. Struktura TIpDevAddress

```
PIpDevAddress = ^TIpDevAddress;
TIpDevAddress = record
  Size : Byte;          { = SizeOf( TIpAddress ) }
  Protocol : Byte;
  case Integer of
    0 : (B1, B2, B3, B4: Byte);
    1 : (Node: TIpAddress);
end;
```

Struktura **TIpDevAddress** popisuje adresu na síťové vrstvě, která se použije při volání aplikačních metod třídy TCoIPv4, např. při volání metod CoSendBufferTo, CoRecvBufferFrom, CoBind a CoConnect apod.

Položka **Size** udává velikost struktury TIPDevAddress. Před použitím struktury se vždy nezapomeňte ujistit, že položka **Size** obsahuje správnou hodnotu, tedy SizeOf(TIPDevAddress). V položce **Protocol** je obsaženo číslo transportního protokolu, který bude uveden v hlavičce IP datagramu. V položce **Node** je IP adresa cílové stanice.

### 4.2.5. Struktura TIoctlIcmpDestUnreach

```
TIoctlIcmpDestUnreach = record
  Header : PIpHeader;
  Code   : Byte;
end;
```

Struktura **TIoctlIcmpDestUnreach** slouží společně s metodou **CoIoctl** a příkazem IOCTL_IPV4_ICMP_DESTUNREACH k odeslání ICMP zprávy DESTINATION UNREACHABLE. Položka **Header** ukazuje na IP hlavičku paketu, jež se odešle v ICMP zprávě společně s dalšími prvními 8 bajty dat datagramu. Položka **Code** specifikuje typ zprávy, viz. konstanty ICMPDU_.

### 4.2.6. Struktura TIpCounters

Tato struktura slouží k předání aktuálních hodnot čítačů událostí. Čítače událostí jsou 32 bitové proměnné, které se zvyšují o 1 pokaždé, co nastane při přijmu nebo vysílání určitá událost. Strukturu TIPCounters lze načíst pomocí metody **CoIoctl** a příkazu IOCTL_IPV4_GETCOUNTERS.

```
TIPCounterType = (

  IPCTR_TX_REQUESTS,
  IPCTR_TX_DATAGRAMS,
  IPCTR_TX_ROUTINGERR,
  IPCTR_TX_DISCARDS,
  IPCTR_TX_DGRAMFRAG,
  IPCTR_TX_FRAGMENTS,

  IPCTR_RX_DATAGRAMS,
  IPCTR_RX_HEADERERR,
  IPCTR_RX_ADDRERR,
  IPCTR_RX_UNKNPROTO,
  IPCTR_RX_DISCARDS,

  ARPCTR_TX_REQUESTS,
  ARPCTR_TX_REPLIES,
  ARPCTR_RX_REQUESTS,
  ARPCTR_RX_REPLIES,

  ICMPCTR_TX_MESSAGES,
  ICMPCTR_RX_MESSAGES
);

TIPCounters = array[TIPCounterType] of Longint;
```

**IPCTR_TX_REQUESTS**

Počet požadavků na odeslání datagramu.

**IPCTR_TX_DATAGRAMS**
Počet odeslaných IP datagramů.

**IPCTR_TX_ROUTINGERR**
Počet zahozených IP datagramů k odeslání z důvodu chyby směrování.

**IPCTR_TX_DISCARDS**
Počet zahozených IP datagramů k odeslání z jiných nespecifikovaných důvodů.

**IPCTR_TX_DGRAMFRAG**
Počet fragmentovaných datagramů

**IPCTR_TX_FRAGMENTS**
Počet odeslaných fragmentů.

**IPCTR_RX_DATAGRAMS**
Počet přijatých IP datagramů.

**IPCTR_RX_HEADERERR**
Počet přijatých IP datagramů s chybami v záhlaví. Takové pakety jsou tiše zahazovány.

**IPCTR_RX_ADDRERR**
Počet přijatých IP datagramů s chybami adresy**.** Takové pakety jsou tiše zahazovány.

**IPCTR_RX_UNKNPROTO**
Počet přijatých IP datagramů s neznámým číslem protokolu. Takové pakety jsou tiše zahazovány.

**IPCTR_RX_DISCARDS**
Počet zahozených příchozích IP datagramů z jiných nespecifikovaných důvodů.

**ARPCTR_TX_REQUESTS**
Počet odeslaných ARP dotazů.

**ARPCTR_TX_REPLIES**
Počet odeslaných ARP odpovědí.

**ARPCTR_RX_REQUESTS**
Počet přijatých ARP dotazů.

**ARPCTR_RX_REPLIES**
Počet přijatých ARP odpovědí.

**ICMPCTR_TX_MESSAGES**

Počet odeslaných ICMP zpráv.

**ICMPCTR_RX_MESSAGES**

Počet přijatých ICMP zpráv.

### 4.2.7. TIpAddrString

```
TIpAddrString = string[15];
```

Textový řetězec reprezentující IP adresu.

# 5. Textový konfigurační řetězec

Parametry zařízení TCoIPv4 lze nastavovat pomocí textového konfiguračního řetězce metodami **CoSetOptionString**.

V následujícím seznamu jsou uvedeny všechny povolené identifikátory parametrů:

**IPADDR** IP adresa síťového adaptéru. IP adresu lze měnit pouze tehdy, pokud je zařízení v neaktivním stavu. (viz. COPT_IPV4_IPADDR)
Formát dat: text, IP adresa
Implicitní nastavení: 192.168.1.64

**NETMASK** Maska sítě ve které je síťový adaptér umístěn. Síťovou masku lze měnit pouze tehdy, pokud je zařízení v neaktivním stavu. (viz. COPT_IPV4_NETMASK)

Formát dat: text, IP adresa
Implicitní nastavení: 255.255.255.0

**DEFGW** Implicitní brána. Na tuto adresu jsou směrovány všechny pakety jdoucí mimo lokální síť, pokud ve směrovací tabulce není pro danou stanici či síť určeno jinak. Neplatná adresa 0.0.0.0 zruší implicitní směrování. Implicitní bránu lze měnit pouze tehdy, pokud je zařízení v neaktivním stavu.

Formát dat: text, IP adresa
Implicitní nastavení: 0.0.0.0

**TTL** Hodnota položky TTL, která se doplní do každé hlavičky odchozího IP datagramu.

Formát dat: číslo
Rozsah hodnot: 0 až 255
Implicitní nastavení: 64

**ECHO** Povolení odpovídaní na ICMP zprávu ECHO REQUEST. Pokud je tato služba zakázaná (hodnota 0), nebude fungovat příkaz **ping** směrovaný na tento síťový adaptér.

Formát dat: číslo
Rozsah hodnot: 0 (ne), 1 (ano)
Implicitní nastavení: 1

**ARPTO** Doba platnosti položky v ARP cache. Po uvedené době je položka z ARP cache vyjmuta. Doba platnosti položky se udává v sekundách.

Formát dat: číslo
Rozsah hodnot: 5 až 10000s
Implicitní nastavení: 120 s

**ARPRETRY** Počet pokusů o zjištění MAC adresy ARP dotazem, pokud cílová stanice neodpovídá.

Formát dat: číslo
Rozsah hodnot: 1 až 100s
Implicitní nastavení: 3s

**REASMTO** Doba platnosti fragmentů čekajících na složení do datagramu. Po této nastavené době jsou neúplné příchozí datagramy zahozeny. Doba platnosti se udává v sekundách.

Formát dat: číslo
Rozsah hodnot: 1 až 120s
Implicitní nastavení: 16s

**MTUR** Maximální délka datagramu, který lze přijmout. Delší datagramy jsou automaticky zahazovány.

Formát dat: číslo
Rozsah hodnot: 1024 až 1638
Implicitní nastavení: 2048

**Příklad:**

```
var
  Sock : PCoDevice;

.
.

Sock^.CoSetOptionString( DEV_CLASS_IPV4,
                         'IPADDR="192.168.1.1"', nil );
```

# 6. Funkce a procedury

## 6.1. Pomocné funkce pro práci s IP adresou

### 6.1.1. MakeIpAddr

Funkce **MakeIpAddr** slouží k přehlednému vytvoření IP adresy.

```
function MakeIpAddr( A3, A2, A1, A0: Byte ): TIpAddress;
```

**Parametry:**

A3, A2, A1, A0

**Návratová hodnota:**

Funkce **MakeIpAddr** vrací IP adresu specifikovanou v parametrech A3 až A0.

**Příklad:**

```
var
  LocalIp : TIpAddress

.
.

LocalIp := MakeIpAddr( 192, 168, 1, 1 );
{ Je ekvivalentní s LocalIp := $C0A80101; }

.
.
```

### 6.1.2. IpAddrToStr

Funkce **IpAddrToStr** převádí binární tvar IP adresy do textové podoby.

```
function IpAddrToStr( Address: TIpAddress ): TIpAddrString;
```

**Parametry:**

Address Převáděná IP adresa.

**Návratová hodnota:**

Funkce vrací textový řetězec ve tvaru xxx.xxx.xxx.xxx, např. 192.168.1.1

### 6.1.3. StrToIpAddr

Funkce **StrToIpAddr** převádí textový řetězec reprezentující IP adresu do binární podoby.

```
function StrToIpAddr( const AText: String;
                      var AAddress: TIpAddress ): TCoStatus;
```

**Parametry:**

| | |
|---|---|
| AText | Textový řetězec popisující IP adresu ve formátu xxx.xxx.xxx.xxx. |
| AAddress | Proměnná, do které bude uložena převedená IP adresa. |

**Návratové hodnoty:**

V případě úspěšné konverze vrací funkce návratový kód CST_SUCCESS. V případě chyby vrací CST_ERR_SYNTAX.

# 7. Poznámky

Až na speciální případy, aplikace nikdy nepřistupuje k síťové vrstvě (IP protokolu) přímo, ale téměř vždy využívá vyšší transportní vrstvu (protokol UDP nebo TCP).

## 7.1. Inicializace zařízení pomocí CoBind

Metoda **CoBind** provede inicializaci instance IP protokolu a zaregistruje ji u nižší vrstvy.

První parametr metody **CoBind**, tj. AAdress je ukazatel na lokální adresu (strukturu TIpDevAddress). Pouze položky Size a Node struktury TIpDevAddress musí být vyplněny. Pokud místo ukazatele na strukturu TIpDevAddress předáme hodnotu **nil**, pak se použije implicitní nastavení lokální IP adresy. (viz. COPT_IPV4_IPADDR nebo parametr konfiguračního textového řetězce IPADDR).

## 7.2. Navazování spojení pomocí CoConnect

Protokol IP neumožňuje navázat spojení mezi dvěma stanicemi a metody **CoConnect** ani **CoDisconnect** nejsou implementovány.

## 7.3. Posílání a příjem dat

K posílání a příjmu dat jsou určené metody **CoSendBufferTo** a **CoRecvBufferFrom** K předávání adresy cílové příp. zdrojové adresy je určena struktura TIpDevAddress. V případě metod pro odesílání a příjem dat jsou využity všechny položky struktury TIpDevAddress.

IP protokol umožňuje posílat data na všeobecnou adresu (broadcast). Všeobecná adresa je 255.255.255.255 (konstanta IP_ADDR_BROADCAST)