# SNMP

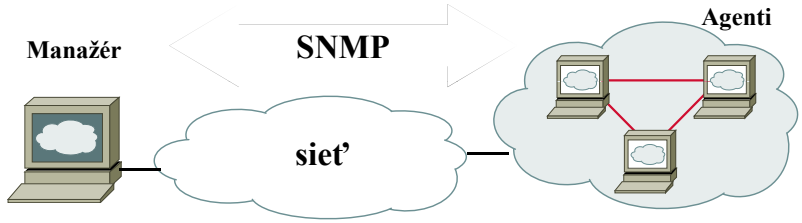
*(Simple Network Management Protocol)*





## SNMP

**SNMP** je manažmentový protokol určený na riadenie sietí.





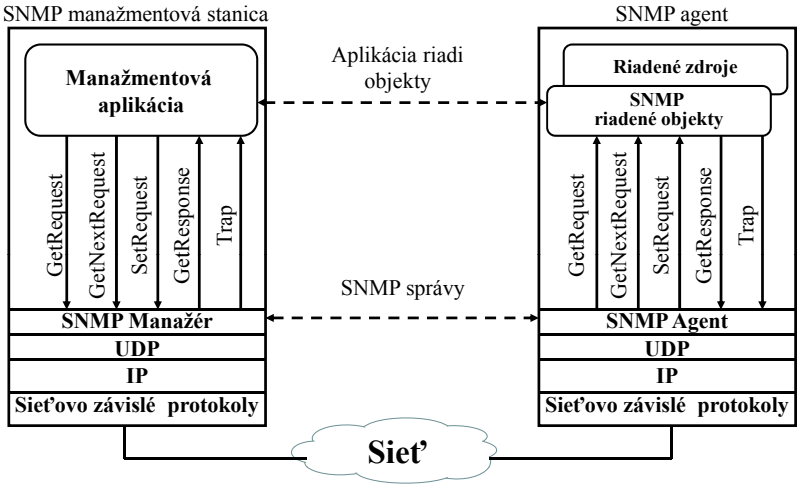



## SNMP

- **Základné špecifikácie:**
  - **RFC 1155** - Structure and Identification of Management Information for TCP/IP-based Internets
  - **RFC 1213** - Management Information Base for Network Management of TCP/IP-based internets: MIB-II
  - **RFC 1157** - A Simple Network Management Protocol (SNMP)







# Typy operácií

V rámci SNMP sú špecifikované tri všeobecné operácie so skalárnymi objektmi.

- **Get -** manažmentová stanica získava hodnotu skalárneho objektu od agenta.
- **Set** - manažmentová stanica mení hodnotu skalárneho objektu u agenta.
- **Trap -** agent posiela nevyžiadanú hodnotu skalárneho objektu manažmentovej stanici.





# Vlastnosti SNMP

- Nie je možné meniť štruktúru MIB pridávaním, alebo mazaním inštancií objektov.
- Je možné vydávať príkazy na vykonanie určitej činnosti.
- Je možný prístup len k objektom nachádzajúcim sa v koncových uzloch registračného stromu.
- Je možné vykonávanie operácií nad dvojrozmernými tabuľkami.





# SNMP manažment

**Jeden agent – viac manažmentových staníc.**

Každý agent si riadi svoju lokálnu MIB a musí byť schopný riadiť prístup k nej od viacerých manažmentových staníc.





# Tri aspekty riadenia prístupu

1. **Autentifikácia -** agent môže obmedziť právo na prístup k MIB len pre autorizované manažmentové stanice.
2. **Prístupová politika -** agent môže mať rôzne prístupové práva pre rôzne manažmentové stanice.
3. **Proxy služba** - agent môže slúžiť ako proxy pre ďalšie riadené stanice. Z toho vyplýva možnosť implementovania autentifikácie a prístupovej politiky pre iné riadené systémy na danom proxy systéme.

# Komunita

- SNMP komunita je vzťah medzi SNMP agentom a množinou SNMP manažérov, ktorý definuje autentifikáciu, riadenie prístupov a proxy charakteristiky.
- Komunita je definovaná na strane agenta. Agent môže ustanoviť viacero komunít, v rámci ktorých môže dochádzať k prekrývaniu sa jednotlivých manažmentových staníc.
- Názvy komunít musia byť v rámci agenta jednoznačné, avšak rôzni agenti môžu používať tie isté názvy komunít.





# SNMP správy

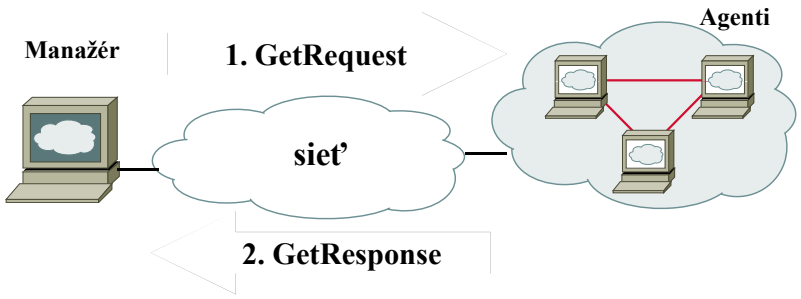
- V SNMP sú informácie medzi riadiacou stanicou a agentom vymieňané vo forme SNMP správ.
- **Typy SNMP správ**
  - GetRequest PDU
  - GetNextRequest PDU
  - SetRequest PDU
  - GetResponse PDU
  - Trap PDU





# GetRequest

**GetRequest** slúži na získanie hodnoty inštancie objektu od agenta.








# GetRequest - *vlastnosti*

- GetRequest PDU môže obsahovať zoznam viacerých objektov.
- Operácia GetRequest je ***atomická***.

***Atomickosť operácie***

*Ak nemôže byť zaslaná jedna z požadovaných hodnôt, tak nie sú zaslané žiadne hodnoty.*

*Príklad*

## Zistenie hodnoty viacerých objektov

- **Príkaz**

  ***GetRequest*** *(udpInDatagrams.0, udpNoPorts.0, udpInErrors.0, udpOutDatagrams.0)*

- **Odpoveď**

  ***GetResponse*** *((udpInDatagrams.0 = 100), (udpNoPorts.0 = 1), (udpInErrors.0 = 2),(udpOutDatagrams.0 = 200))*





*Príklad*

## Načítanie hodnôt jedného riadku tabuľky

| ipRouteDest | ipRouteMetric1 | ipRouteNextHope |
|---|---|---|
| 9.1.2.3 | 3 | 99.0.0.3 |
| 10.0.0.51 | 5 | 10.0.0.99 |
| 10.0.0.99 | 5 | 89.1.1.42 |

***GetRequest*** *(ipRouteDest.9.1.2.3, ipRouteMetric1.9.1.2.3, ipRouteNextHope.9.1.2.3)*





## GetRequest - *atomickosť operácie*

- Ak sa označenie požadovaného objektu nezhoduje ani s jedným identifikátorom objektu v MIB pohľade, alebo požadovaný objekt je agregátneho typu (napr. tabuľka) a preto nemá priradenú hodnotu inštancie - v odpovedi GetResponse PDU je ako **error-status** uvedené **NoSuchName**,

- Ak odpovedajúca entita môže zaslať hodnoty všetkých požadovaných objektov, ale odpoveď GetResponse PDU by prekročila miestne obmedzenie - v odpovedi GetResponse PDU je ako **error-status** uvedené **tooBig**,

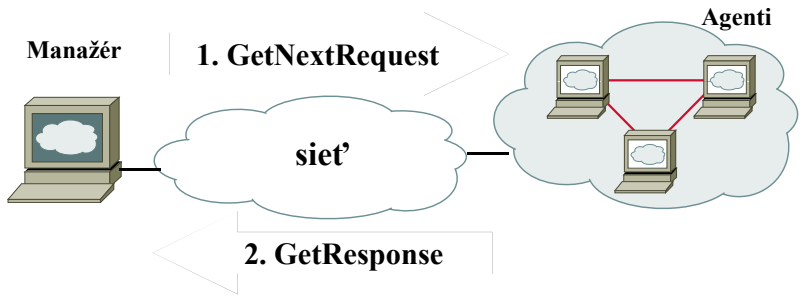
- Ak odpovedajúca entita nemôže z nejakého iného dôvodu zaslať hodnotu niektorého z požadovaných objektov - v odpovedi GetResponse PDU je ako **error-status** uvedené **genErr**.





## GetNextRequest

**GetNextRequest** slúži na získanie hodnoty inštancie objektu ktorá nesleduje v lexikografickom poradí za inštanciou uvedenou v zaslanej správe.



# GetNextRequest - *vlastnosti*

- GetNextRequest PDU môže obsahovať zoznam viacerých objektov.
- Operácia GetNextRequest je tiež ***atomická***.





*Príklad*

# Zistenie hodnoty viacerých objektov

- **Príkaz**
  ***GetNextRequest*** *(udpInDatagrams, udpNoPorts, udpInErrors, udpOutDatagrams)*
- **Odpoveď**
  ***GetResponse*** *((udpInDatagrams.0 = 100), (udpNoPorts.0 = 1), (udpInErrors.0 = 2),(udpOutDatagrams.0 =200))*





## Načítanie hodnôt tabuľky *Príklad*

| ipRouteDest | ipRouteMetric1 | ipRouteNextHope |
|---|---|---|
| 9.1.2.3 | 3 | 99.0.0.3 |
| 10.0.0.51 | 5 | 10.0.0.99 |
| 10.0.0.99 | 5 | 89.1.1.42 |

- **1. príkaz**
  ***GetNextRequest*** *(ipRouteDest, ipRouteMetric1, ipRouteNextHope)*
- **Odpoveď**
  ***GetResponse*** *((ipRouteDest.9.1.2.3 = 9.1.2.3), (ipRouteMetric1.9.1.2.3 = 3), (ipRouteNextHope.9.1.2.3 = 99.0.0.3))*





## Načítanie hodnôt tabuľky *Príklad -pokr.*

| ipRouteDest | ipRouteMetric1 | ipRouteNextHope |
|---|---|---|
| 9.1.2.3 | 3 | 99.0.0.3 |
| 10.0.0.51 | 5 | 10.0.0.99 |
| 10.0.0.99 | 5 | 89.1.1.42 |

- **2. príkaz**
  ***GetNextRequest*** *(ipRouteDest.9.1.2.3), (ipRouteMetric1.9.1.2.3), (ipRouteNextHope.9.1.2.3)*
- **Odpoveď**
  ***GetResponse*** *((ipRouteDest.10.0.0.51 = 10.0.0.51), (ipRouteMetric1.10.0.0.51 = 5), (ipRouteNextHope.10.0.0.51 = 10.0.0.99))*

## Načítanie hodnôt tabuľky *Príklad - pokr.*

| ipRouteDest | ipRouteMetric1 | ipRouteNextHope |
|---|---|---|
| 9.1.2.3 | 3 | 99.0.0.3 |
| 10.0.0.51 | 5 | 10.0.0.99 |
| 10.0.0.99 | 5 | 89.1.1.42 |

- **3. príkaz**

  ***GetNextRequest*** *(ipRouteDest.10.0.0.51), (ipRouteMetric1.10.0.0.51), (ipRouteNextHope.10.0.0.51)*

- **Odpoveď**

  ***GetResponse*** *((ipRouteDest.10.0.0.99 = 10.0.0.99), (ipRouteMetric1.10.0.0.99 = 5), (ipRouteNextHope.10.0.0.99 = 89.1.1.42))*





## Načítanie hodnôt tabuľky *Príklad - pokr.*

| ipRouteDest | ipRouteMetric1 | ipRouteNextHope |
|---|---|---|
| 9.1.2.3 | 3 | 99.0.0.3 |
| 10.0.0.51 | 5 | 10.0.0.99 |
| 10.0.0.99 | 5 | 89.1.1.42 |

- **4. príkaz**

  ***GetNextRequest*** *(ipRouteDest.10.0.0.99, ipRouteMetric1.10.0.0.99, ipRouteNextHope.10.0.0.99)*

- **Odpoveď**

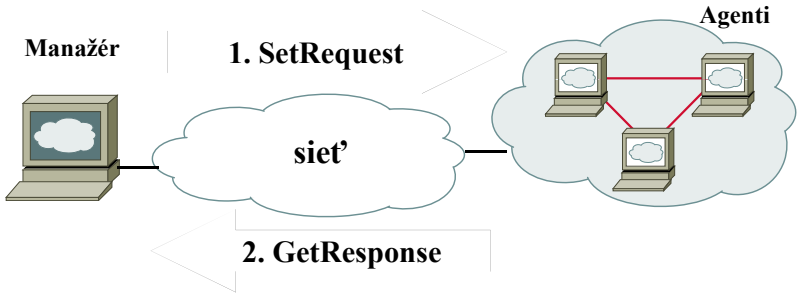
  ***GetResponse*** *((ipRouteMetric1.9.1.2.3 = 3), (ipRouteNextHope.9.1.2.3 = 99.0.0.3), (ipNetToMediaIfIndex.1.3 = 1))*





## SetRequest

**SetRequest** slúži na zmenu hodnoty inštancie objektu u agenta.








## SetRequest - *vlastnosti*

- SetRequest PDU musí obsahovať identifikátory inštancií objektov a im priradené hodnoty.
- SetRequest PDU môže obsahovať zoznam viacerých objektov.
- Operácia SetRequest je ***atomická -*** *sú zmenené hodnoty všetkých objektov, alebo žiadneho.*

*Príklad*

## Zmena hodnoty inštancie objektu

- **Príkaz**
  ***SetRequest*** *(ipRouteMetric1.9.1.2.3 = 9)*
- **Odpoveď**
  ***GetResponse*** *(ipRouteMetric1.9.1.2.3 = 9))*
- Uvedeným spôsobom je možno vykonať aj zmenu hodnôt tabuľky.





## Pridanie riadku do tabuľky

| ipRouteDest | ipRouteMetric1 | ipRouteNextHope |
|---|---|---|
| 9.1.2.3 | 3 | 99.0.0.3 |
| 10.0.0.51 | 5 | 10.0.0.99 |
| 10.0.0.99 | 5 | 89.1.1.42 |

- **Príkaz**
  ***SetRequest*** *((ipRouteDest.11.3.3.12 = 11.3.3.12),*
  *(ipRouteMetric1.11.3.3.12 = 9),*
  *(ipRouteNextHope.11.3.3.12 = 91.0.0.5)*





## Pridanie riadku do tabuľky - odpovede

**RFC 1212 umožňuje 3 spôsoby reakcie agent**

1. agent nepozná objekt **ipRouteDest.11.3.3.12** a operáciu zamietne
   **Odpoveď -** ***GetResponse*** *(error-status = noSuchName)*
2. agent akceptuje operáciu ako žiadosť o vytvorenie novej inštancie objektu, ale zistí, že niektorá z priradených hodnôt má zlú syntax, alebo hodnotu
   **Odpoveď -** ***GetResponse*** *(error-status = badValue)*
3. agent akceptuje operáciu a vytvorí ďalší riadok tabuľky
   **Odpoveď -**
   ***GetResponse*** *((ipRouteDest.11.3.3.12 = 11.3.3.12),*
   *(ipRouteMetric1.11.3.3.12 = 9),*
   *(ipRouteNextHope.11.3.3.12 = 91.0.0.5))*





## Zrušenie riadku tabuľky

- Nastavením hodnoty špeciálnej premennej *(napr. ipRouteType)* na hodnotu *invalid*
- ***Príklad***

| | | |
|---|---|---|
| **Príkaz** | ***SetRequest*** | *((ipRouteType.7.3.5.3 = invalid)* |
| **Odpoveď** | ***GetResponse*** | *((ipRouteDest.7.3.5.3 = invalid)* |

- *Pozn.: Nie je aplikovateľné na všetky tabuľky v MIB-II*

# Iniciovanie vykonania nejakej činnosti

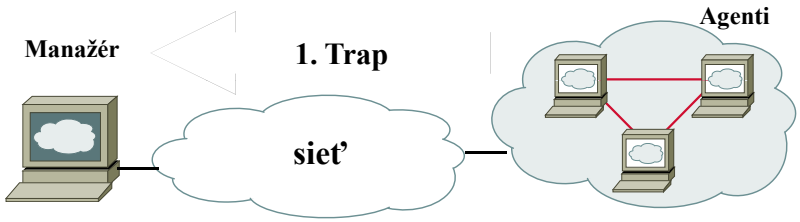
- SNMP neposkytuje špeciálny mechanizmus pre zaslanie príkazu na vykonanie nejakej činnosti agentovi.
- Iniciovanie vykonania určitej činnosti je možno realizovať zmenou hodnoty špeciálneho, na to určeného, objektu *(napr. reBoot = 0 --> 1).*





# Trap

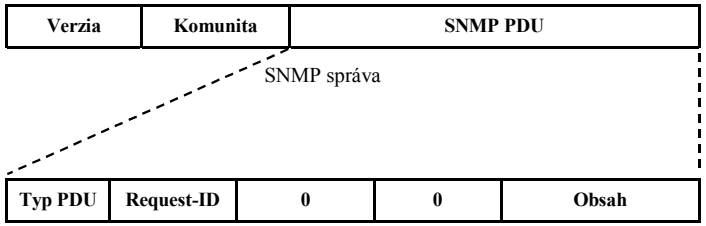
**Trap** slúži na zaslanie nevyžiadanej správy agentom manažérovi.







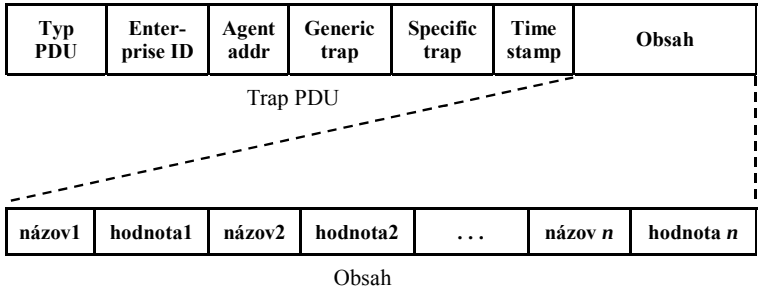

# Formáty SNMP správ








# Formáty SNMP správ - *pokr.*



# Formáty SNMP správ - *pokr.*

- **Verzia** - verzia SNMP
- **Komunita** - názov SNMP komunity, slúži ako heslo na autentifikáciu SNMP správy
- **Typ PDU** - označenie typu PDU *(napr. GetRequest PDU)*
- **Request ID** - slúži na rozlíšenie žiadostí. Každá žiadosť má jednoznačné ID.





# Formáty SNMP správ - *pokr.*

- **Error-status** – oznamuje typ mimoriadnej udalosti, ktorá sa vyskytla pri spracovaní žiadosti.
  - Možné hodnoty:

| | |
|---|---|
| **0** – žiadna chyba | *(noError)* |
| **1** – príliš veľké | *(tooBig)* |
| **2** – neexistujúce meno | *(noSuchName)* |
| **3** – zlá hodnota | *(badValue)* |
| **4** – iba na čítanie | *(readOnly)* |
| **5** – všeobecná chyba | *(genErr)* |

- **Error-index** - poskytuje ďalšie informácie o mimoriadnej udalosti
- **Enterprise** - udáva typ objektu generujúceho trap





# Formáty SNMP správ - *pokr.*

- **Agent-addr** - adresa objektu generujúceho trap
- **Generic-trap** – všeobecný typ trapu.
  - Možné hodnoty:

| | |
|---|---|
| **0** - studený štart systému | *(coldStart)* |
| **1** - teplý štart systému | *(warmStart)* |
| **2** - linka je neaktívna | *(linkDown)* |
| **3** - linka je aktívna | *(linkUp)* |
| **4** - zlyhanie autentifikácie | *(authenticationFailure)* |
| **5** - strata spojenia na susedný EGP uzol | *(egpNeighborLoss)* |
| **6** – firemne špecifické | *(enterpriseSpecific)* |

- **Specific-trap** – špecifický kód trapu
- **Time-stamp** – čas generovania trapu meraný od poslednej reinicializácie sieťovej entity





# Obmedzenia SNMP

- SNMP nie je vhodné pre riadenie veľkých sietí,
- SNMP nie je vhodné na získavanie veľkých objemov dát (napr. úplných smerovacích tabuliek),
- SNMP trapy sú nepotvrdzované,
- SNMP poskytuje iba minimálnu autentifikáciu,
- SNMP nepodporuje priame imperatívne príkazy,
- Model SNMP MIB je limitovaný,
- SNMP nepodporuje komunikáciu medzi manažérmi.

**Riešenie**

# SNMP v2

# Vylepšenia SNMPv2

Vylepšenia manažmentového protokolu SNMPv2 oproti SNMP je možné rozdeliť do nasledovných oblastí:

- **Štruktúra manažmentovej informácie**
- **Protokolové operácie**
- **Spolupráca medzi manažérmi**
- **Bezpečnosť**





# Štruktúra manažmentovej informácie

- Rozšírením makra definujúceho typy objektov bolo pridaných niekoľko nových dátových typov.
  - **Counter64**
  - **UInteger32** *(unsignet Integer)*
  - **NsapAddress**
- Zmenilo sa označovanie existujúcich dátových typov:
  - **Integer32** (-2147483648 .. 2147483648 )
  - **Counter32**
  - **Gauge32**
- Ostali typy: **IpAddress, TimeTics, Opague**





# Štruktúra manažmentovej informácie

- Bol pridaný nový typ prístupu k objektom *(read-create)*
- Zmenil sa spôsob vytvárania a rušenia riadkov v tabuľke -

  - realizuje sa pridaním stĺpcového objektu ktorý má:
    - **SYNTAX** **RowStatus**
    - **MAX-ACCESS** **read-create**

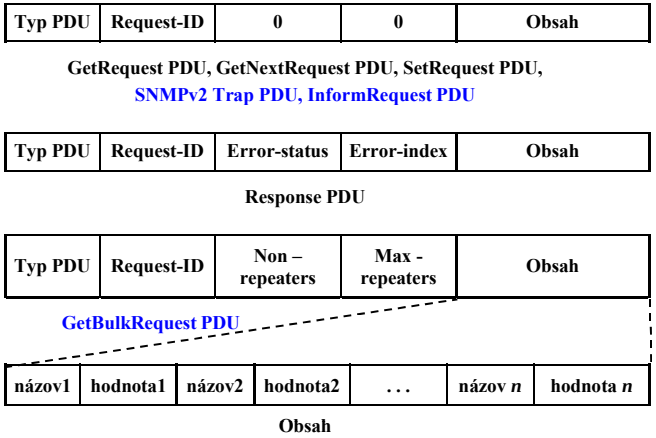




# Protokolové operácie

- Boli pridané dva nové typy PDU
  - **GetBulkRequest PDU**
  - **InformRequest PDU**
- Do SNMPv2 MIB boli pridané ďalšie informácie týkajúce sa konfigurácie SNMPv2 manažéra a agenta.

# Formáty SNMPv2 správ








# GetRequest

- SNMPv2 GetRequest PDU je formátom a sémantikou identická s SNMP GetNextRequest PDU.
- Rozdiel - operácia GetRequest **nie je** atomická.
- V prípade chyby, môže byť vrátená ako príslušná hodnota kód chyby

```
VarBind ::=
    SEQUENCE {name ObjectName,
        CHOICE {   value                 ObjectSyntax,
                   unSpecified           NULL,
                   noSuchObject    [0]   IMPLICIT NULL,
                   noSuchInstance  [1]   IMPLICIT NULL,
                   endOfMibView    [2]   IMPLICIT NULL}}
```





# GetNextRequest

- SNMPv2 GetNextRequest PDU je formátom a sémantikou identická s SNMP GetNextRequest PDU.
- Rozdiel - operácia GetNextRequest **nie je** atomická.
- V prípade chyby pri získavaní hodnoty niektorého objektu je správanie sa podobné ako pri SNMPv2 GetNextRequest PDU





# GetBulkRequest

- GetBulkRequest PDU umožňuje minimalizovať počet protokolových výmen potrebných pre prenos veľkého objemu manažmentových informácií.
- GetBulkRequest pracuje na podobnom princípe ako GetNextRequest.
- Rozdiel - GetBulkRequest umožňuje špecifikovať počet lexikografických nasledovníkov.

# GetBulkRequest - príklad

- Požiadavka na získanie hodnoty:
  - sysUpTime,
  - ipNetToMediaPhysAddress
  - ipNetToMediaType

| ipNetToMedia IfIndex | ipNetToMedia PhysAddress | ipNetToMedia NetAddress | ipNetToMedia Type |
|---|---|---|---|
| 1 | 10.0.0.51 | 00:00:10:01:23:45 | Static |
| 1 | 9.2.3.4 | 00:00:10:54:32:10 | Dynamic |
| 2 | 10.0.0.15 | 00:00:10:98:76:54 | Dynamic |





# GetBulkRequest - príklad *(pokr.)*

- **1. príkaz**

  GetBulRequest [nonrepeaters = 1, max-repeaters = 2]
  (sysUpTime, ipNetToMediaPhysAddress,ipNetToMediaType)

- **Odpoveď**

  Response (( sysUpTime.0 = „123456“,
  (ipNetToMediaPhysAddress.1.9.2.3.4 = „000010543210“),
  (ipNetToMediaType.1.9.2.3.4 = „dynamic“),
  (ipNetToMediaPhysAddress.1.10.0.0.51 = „000010012345“),
  (ipNetToMediaType.1.10.0.0.51 = „static“)





# GetBulkRequest - príklad *(pokr.)*

- **2. príkaz**

  GetBulRequest [nonrepeaters = 1, max-repeaters = 2]
  (sysUpTime, ipNetToMediaPhysAddress.1.10.0.0.51,
  ipNetToMediaType.1.10.0.0.51)

- **Odpoveď**

  Response (( sysUpTime.0 = „123477“,
  (ipNetToMediaPhysAddress.2.10.0.0.15 = „000010987654“),
  (ipNetToMediaType.2.10.0.0.15 = „dynamic“),
  (ipNetToMediaNetAddress.1.9.2.3.4 = „9.2.3.4“),
  (ipRoutingDiscards.0 = „2“)





# SetRequest

- SNMPv2 SetRequest PDU je formátom a sémantikou identická s SNMP SetRequest PDU.
- Rozdiel - v spôsobe spracovania odpovede.
- Vylepšenie - detailnejší popis typu chyby v odpovedi
  *(19 typov chybových hlásení)*
- Operácia SetRequest **je** atomická

# SetRequest - chybové hlásenia

- noError (0)
- tooBig (1)
- noSuchName (2)
- badValue (3)
- readOnly (4)
- genError (5)
- noAccess (6)
- wrongType (7)
- wrongLenght (8)
- wrongEncoding (9)
- wrongValue (10)
- noCreation (11)
- inconsistentValue (12)
- resourceUnavailable (13)
- commitFailed (14)
- undoFailed (15)
- authorizationError (16)
- notWritable (17)
- inconsistentName (18)





# Trap

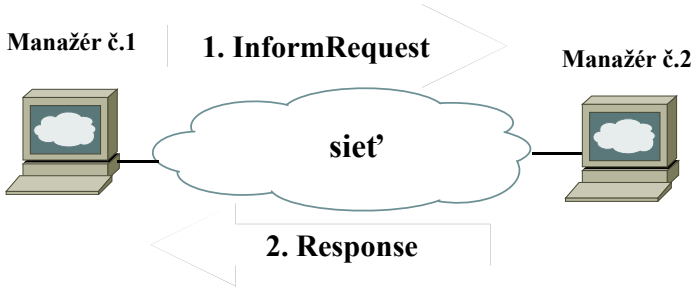
- SNMPv2 Trap PDU:
  - má rovnakú funkciu ako SNMP Trap PDU
  - má iný formát (ako všetky SNMPv2 PDU okrem GetBulkRequest)
  - podobne ako SNMP Trap PDU nie je potvrdzovaný.
- SNMPv2 Trap PDU má nasledovný obsah:
  - sysUpTime.0 - ako 1. premenná,
  - SNMPv2OID.0 - ako 2. premenná,
  - nasledujú inštancie objektov def. pre daný typ trapu





# InformRequest

- InformRequest PDU slúži na výmenu informácií medzi manažérmi.








# Spolupráca manažér - manažér

- SNMPv2 umožňuje výmenu manažmentových informácií medzi manažérmi (prostredníctvom InformRequest PDU)
- SNMPv2 definuje Manager-to-manager MIB.
  - Má dve skupiny:
    - **Alarm groupe,**
    - **Event groupe.**
- **Možnosť budovať distribuované manažmentové architektúry**

# Bezpečnosť

- SNMPv2 prináša snahu o zvýšenie bezpečnosti
  - **SNMPv2** - r.1993 - obsahuje určité bezpečnostné prvky.
  - **SNMPv2c** - r.1996 - [RFC 1901 - 1908] len funkčné vylepšenia, neobsahuje bezpečnostné vylepšenia (len „Community-based“ SNMPv2).
  - **SNMPv2u** - r. 1996 [RFC 1909 - 1910] } snaha o zlepšenie
    **SNMPv2*** - } bezpečnosti

**Riešenie**

**SNMP v3** *(RFC 3410 - 3418)*





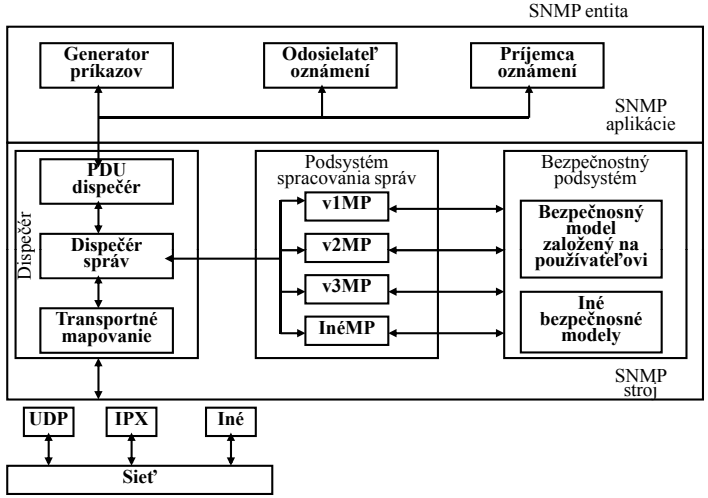
# SNMPv3 - vlastnosti

- SNMPv3 nie je priamou náhradou za SNMPv1 alebo SNMPv2 ale ide o rozšírenie SNMPv2 (alebo SNMPv1) o bezpečnostné mechanizmy.
- SNMPv3 umožňuje autorizáciu *(HMAC-MD5-96, HMAC-SHA-96)* a kryptovanie prenášaných správ *(DES)*
- SNMPv3 definuje novú architektúru agenta a manažéra (každá SNMP entita pozostáva z modulov, ktoré navzájom spolupracujú s cieľom poskytovať manažmentové služby).





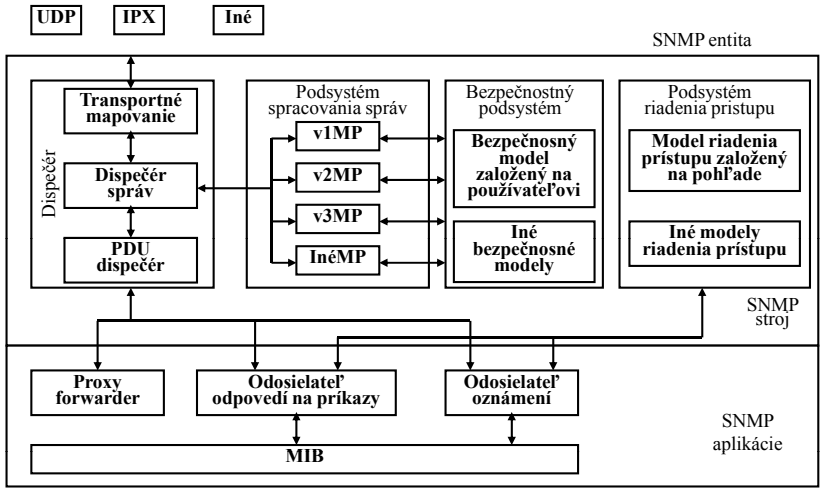
# Tradičný SNMP manažér



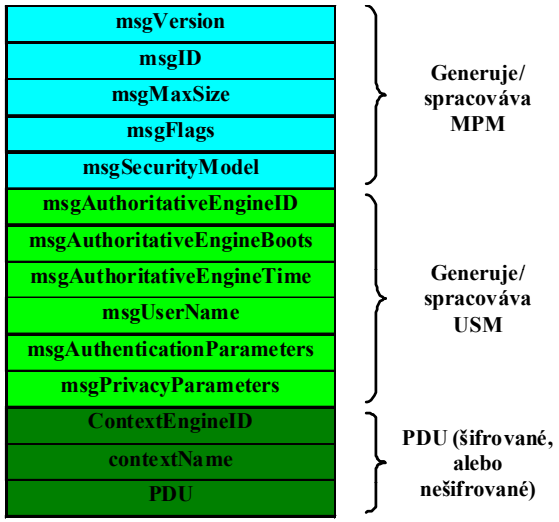





# Tradičný SNMP agent



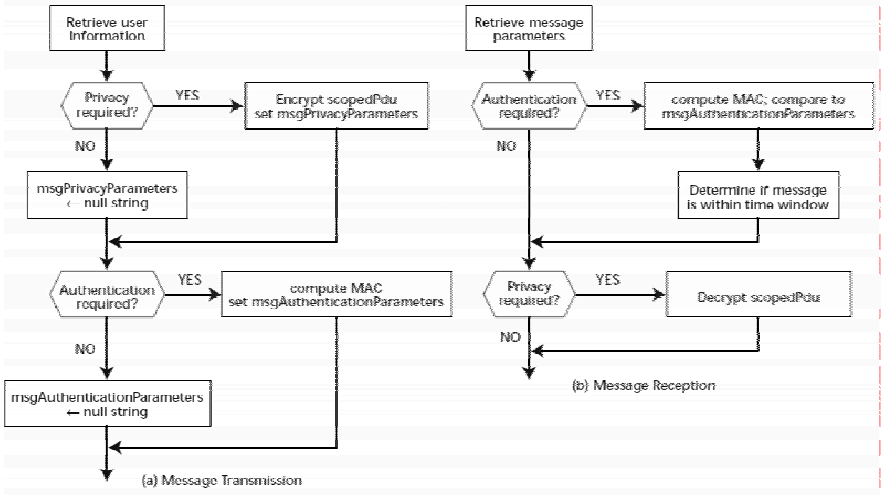
# Formát správy s USM








# Postup spracovania USM správy








# Model riadenia prístupu založený na pohľade

- SNMPv3 definuje päť prvkov umožňujúcich riedenie prístupu
  (RFC 3415):
  - skupina,
  - bezpečnostná úroveň,
  - kontext,
  - MIB pohľad,
  - prístupová politika.





# Logika riadenia prístupu založeného na pohľade