

وضعیت سند جاری

این سند برای جامعه‌ی اینترنت یک پروتکل در استانداردهای شبکه را بیان می‌کند و درخواست می‌کند که پیشنهادها و نظرهای خود را برای بهبود آن ارائه کنند. برای آگاهی از وضعیت استانداردسازی این پروتکل، به نسخه‌ی جاری استانداردهای رسمی پروتکل اینترنت (STD1) رجوع کنید. انتشار این سند بدون محدودیت امکان‌پذیر است.

چکیده

این سند نسخه‌ی دوم پروتکل SNMP را توضیح می‌دهد. این سند نحو و عناصر رویه‌های ارسال، دریافت و پردازش بسته‌های SNMP را توضیح می‌دهد. سند جاری RFC ۱۹۰۵ را منسوخ می‌کند.

فهرست مطالب

| | |
|-----------------------------------|----|
| ۱-۱- مقدمه | ۳ |
| ۱-۲- ایده‌ی کلی | ۳ |
| ۱-۲-۱- اطلاعات مدیریتی | ۴ |
| ۱-۲-۲- بازارسال درخواست‌ها | ۴ |
| ۱-۲-۳- اندازه‌ی پیام‌ها | ۴ |
| ۱-۲-۴- نگاشت‌های لایه‌ی انتقال | ۵ |
| ۱-۲-۵- نگاشت‌های نوع داده‌ی SMIV۲ | ۵ |
| ۱-۳- تعاریف | ۵ |
| ۱-۴- ویژگی‌های پروتکل | ۸ |
| ۱-۴-۱- ساختارهای رایج | ۸ |
| ۱-۴-۲- پردازش پیام | ۸ |
| ۱-۴-۳- پیام GetRequest | ۹ |
| ۱-۴-۴- پیام GetNextRequest | ۹ |
| ۱-۴-۵- پیام GetBulkRequest | ۱۱ |
| ۱-۴-۶- پیام SetRequest | ۱۴ |
| ۱-۴-۷- پیام SNMPv۲-Trap | ۱۷ |
| ۱-۴-۸- پیام InformRequest | ۱۷ |
| ۱-۵- اطلاع از مالکیت معنوی | ۱۸ |
| ۱-۶- قدردانی | ۱۸ |
| ۱-۷- ملاحظات امنیتی | ۱۹ |

۱-۱- مقدمه

در زمان نوشتن این سند، چهارچوب مدیریتی SNMP از پنج عنصر اصلی تشکیل شده است:

- یک معماری کلی که در STD ۶۲، RFC ۳۴۱۱ [RFC۳۴۱۱] ارائه شده است.
- سازوکارهایی برای توصیف و نام‌گذاری اشیاء و رویدادها برای اهداف مدیریتی ارائه شده‌اند. اولین نسخه از این ساختار همان SMIV۱ نام دارد و در STD ۱۶، RFC ۱۱۵۵ [RFC۱۱۵۵]، STD ۱۶، RFC ۱۲۱۲ [RFC۱۲۱۲] و RFC ۱۲۱۵ [RFC۱۲۱۵] تعریف شده است. نسخه‌ی دوم که SNMPv۲ نام دارد در STD ۵۸، RFC ۲۵۷۸ [RFC۲۵۷۸]، STD ۵۸، RFC ۲۵۷۹ [RFC۲۵۷۹] و STD ۵۸، RFC ۲۵۸۰ [RFC۲۵۸۰] توصیف شده است.
- پروتکل‌های پیام‌رسانی برای انتقال اطلاعات مدیریتی. اولین نسخه از پروتکل پیام‌رسانی SNMP SNMPv۱ نام دارد و در STD ۱۵، RFC ۱۱۵۷ [RFC۱۱۵۷] توضیح داده شده است. نسخه‌ی دوم از پروتکل پیام‌رسانی SNMP، SNMPv۲c نام دارد و جزء پروتکل‌های ردیابی استاندارد اینترنت نیست. این نسخه در RFC ۱۹۰۱ [RFC۱۹۰۱] و STD ۶۲ توضیح داده شده است. نسخه‌ی سوم نیز SNMPv۳ نام دارد و در STD ۶۲، RFC ۳۴۱۷ [RFC۳۴۱۷]، RFC ۳۴۱۲ [RFC۳۴۱۲] و RFC ۳۴۱۴ [RFC۳۴۱۴] به توضیح آن پرداخته شده است.
- پروتکل‌های عملیاتی برای دسترسی به اطلاعات مدیریتی اند. اولین مجموعه از پروتکل‌های عملیاتی و ساختار پیام‌های مرتبط با آن در STD ۱۵، RFC ۱۱۵۷ [RFC۱۱۵۷] توصیف شده‌اند. در این سند دومین مجموعه از پروتکل‌های عملیاتی و ساختار پیام‌های مرتبط با آن توضیح داده می‌شوند.
- مجموعه‌ای از برنامه‌های کاربردی پایه در STD ۶۲، RFC ۳۴۱۳ [RFC ۳۴۱۳] توصیف شده‌اند. در STD ۶۲، RFC ۳۴۱۵ [RFC۳۴۱۵] نیز، مکانیزم کنترل دسترسی مبتنی بر دید^۱ توضیح داده شده است. برای آشنایی دقیق‌تر با پروتکل SNMP می‌توانید در زمان نوشتن این سند، به RFC ۳۴۱۰ [RFC۳۴۱۰] مراجعه کنید.
- دسترسی به اشیاء مدیریت شده از طریق یک پایگاه داده به نام MIB صورت می‌گیرد. اشیاء داخل این پایگاه با استفاده از مکانیزم‌های تعریف شده در SMI تعریف می‌شوند.
- این سند (نسخه‌ی دوم از پروتکل ساده‌ی مدیریت شبکه) به عملیاتی از پروتکل می‌پردازد که مربوط به ارسال و دریافت پیام‌های منتقل شده توسط پروتکل پیام‌رسانی هستند.

۱-۲- ایده‌ی کلی

موجودیت‌های SNMP که از برنامه‌های تولیدکننده‌ی فرمان یا دریافت‌کننده‌ی اعلان^۲ پشتیبانی می‌کنند (که به صورت سنتی مدیر نامیده می‌شوند) با موجودیت‌های SNMP که از برنامه‌های پاسخ‌دهنده‌ی فرمان یا تولیدکننده‌ی اعلان پشتیبانی می‌کنند (که به صورت سنتی کارگزار^۳ نامیده می‌شوند) ارتباط برقرار می‌کنند. هدف این پروتکل انتقال اطلاعات و عملیات مدیریتی است.

^۱ view-based access control mechanism

^۲ notification

^۳ agent

۱-۲-۱- اطلاعات مدیریتی

کلمه‌ی "متغیر" به نمونه‌ای از یک نوع شیء غیرتجمعی^۱ اشاره دارد که با توجه به قراردادهای بیان شده در SMI [RFC۲۵۷۸] یا قراردادهای متنی تعریف شده براساس SMI [RFC۲۵۷۹]، تعریف می‌شوند. کلمه‌ی "انقیاد متغیر"^۲ به زوج‌سازی نام یک متغیر و مقدار نظیر آن اشاره دارد. با این حال اگر حین پردازش یک درخواست بازیابی، انواع مشخصی از شرایط غیرعادی رخ دهند، یک انقیاد متغیر یک نام را با اشاره‌گری به آن وضعیت غیرعادی زوج‌سازی می‌کند.

یک لیست variable-binding، لیست ساده‌ای از انقیاد متغیرها است. نام یک متغیر یک OBJECT IDENTIFIER است که از الحاق OBJECT IDENTIFIER نظیر object-type و یک قطعه‌ی OBJECT IDENTIFIER که نشانگر نمونه‌ی آن شیء است، ساخته می‌شود. OBJECT IDENTIFIER نظیر object-type، پیشوند OBJECT IDENTIFIER آن متغیر نام دارد.

۱-۲-۲- بازارسال درخواست‌ها

در شرایط عادی در این پروتکل، گیرنده باید برای تمام انواع درخواست‌ها پاسخی را تولید کند و به ارسال‌کننده‌ی درخواست بفرستد. تصمیم‌گیری در خصوص بازارسال پاسخ‌هایی که در محدوده‌ی زمانی مناسب دریافت نشده‌اند برعهده‌ی ارسال‌کننده‌ی درخواست است و به صورت معمول به اورژانسی بودن درخواست وابسته است. با این حال چنین برنامه‌ای باید درخصوص تعیین فرکانس و طول بازه‌ی زمانی برای بازارسال به صورتی مناسب تصمیم‌گیری کند. برای آشنایی با اصول کنترل ازدحام فوق به ۴۱ BCP [RFC۲۹۱۴] مراجعه کنید.

۱-۲-۳- اندازه‌ی پیام‌ها

بزرگ‌ترین اندازه یک پیام SNMP توسط حداقل‌های زیر تعیین می‌گردد:

(۱) بزرگ‌ترین اندازه پیامی که موجودیت مقصد SNMP می‌تواند بپذیرد؛ و

(۲) بزرگ‌ترین اندازه پیامی که موجودیت مبدا SNMP می‌تواند تولید کند.

اولین مورد به ازای هر گیرنده تعریف می‌شود و در شرایطی که این اطلاعات در دسترس نباشد، توسط دامنه‌ی انتقال که حین ارسال پیام مورد استفاده قرار می‌گیرد، تعیین می‌گردد. دومین مورد نیز توسط محدودیت‌های محلی وابسته به پیاده‌سازی تعیین می‌شوند.

هر نگاشت انتقال برای SNMP، نشانگر حداقل اندازه‌ی پیامی است که یک پیاده‌سازی SNMP باید بتواند تولید یا مصرف کند. گرچه پیاده‌سازی‌ها تشویق می‌شوند تا در صورت امکان از مقادیر بزرگ‌تری از اندازه‌ی بسته‌ها پشتیبانی کنند، یک پشتیبانی سازگار نباید هیچگاه پیام‌هایی تولید کند که اندازه‌ی آن‌ها بزرگ‌تر از اندازه‌ی مجاز پیام‌ها در موجودیت SNMP دریافت‌کننده باشد.

یکی از اهداف پیام GetBulkRequest که در این پروتکل مشخص شده است، کمینه نمودن تعداد مبادله‌های پروتکلی مورد نیاز برای دریافت حجم زیادی از اطلاعات مدیریتی است. این نوع پیام به یک موجودیت SNMP که از برنامه‌های تولیدکننده‌ی فرمان پشتیبانی می‌کند، اجازه می‌دهد تا پاسخ‌هایی را درخواست کنند که اندازه‌ی آن‌ها با توجه به محدودیت در اندازه‌ی پیام‌ها بیشینه باشد. این محدودیت‌ها شامل محدودیت‌هایی روی اندازه‌ی پیام‌هایی است که موجودیت SNMP پاسخ‌دهنده می‌تواند تولید کند و اندازه‌ی پیام‌هایی که موجودیت SNMP تولیدکننده‌ی فرمان می‌تواند دریافت کند.

با این حال ممکن است این اندازه‌ی بیشینه برای پیام‌ها، بزرگتر از بزرگ‌ترین واحد قابل انتقال^۳ (MTU) در

^۱ Non-aggregate object type

^۲ Variable binding

^۳ Maximum transmission unit

مسیر پیمایش شده توسط آن‌ها باشد. در این شرایط این بسته‌ها قطعه‌بندی^۱ می‌شوند. از آن‌جا که قطعه‌بندی موجب کاهش اعتمادپذیری در انتقال پیام‌ها می‌شود، به صورت معمول مضر است. بنابراین یک موجودیت SNMP که پیام GetBulkRequest را ارسال می‌کند، باید پارامترهای نظیر را بگونه‌ای مقداردهی کند که از مخاطره‌ی قطعه شدن پیام جلوگیری به عمل آورد. به خصوص در شرایط دشوار برای شبکه، باید مقادیر کوچکی را برای max-repetition انتخاب نمود.

۴-۲-۱- نگاشت‌های لایه‌ی انتقال

توجه به این نکته مهم است که مبادله‌ی پیام‌های SNMP تنها به یک سرویس دیتاگرام غیرقابل اعتماد نیاز دارد که در آن، هر پیام به صورت کامل و مستقل در یک دیتاگرام قرار می‌گیرد. نگاشت‌های لایه‌ی انتقال و قوانین کدگذاری^۲ در [RFC۳۴۱۷] توضیح داده شده‌اند. با این حال، استفاده از پروتکل UDP^۳ ترجیح داده می‌شود [RFC۷۶۸].

۵-۲-۱- نگاشت‌های نوع داده‌ی SMIV۲

SMIV۲ [RFC۲۵۷۸]، یازده نوع پایه (INTEGER, OCTET STRING, OBJECT IDENTIFIER, INInteger۳۲, IpAddress, COUNTER۳۲, Gauge۳۲, Unsigned۳۲, TimeTicks, Opaque, Counter۶۴) و نوع ساخت‌یافته‌ی BITS را تعریف می‌کند. نوع‌های پایه‌ی SMIV۲ به نوع متناظر در انتخاب‌های SimpleSyntax و ApplicationSyntax از تعریف پروتکل SNMP توسط ASN.۱ نگاشت می‌شوند. توجه کنید که نوع‌های پایه‌ی INTEGER و Integer۳۲ از SMIV۲ به نوع انتخابی integer-value از انتخاب SympleSyntax نگاشت می‌شوند. به صورت مشابه، نوع‌های پایه‌ی Gauge۳۲ و Unsigned۳۲ از SMIV۲ به نوع انتخابی unsigned-integer-value از انتخاب ApplicationSyntax نگاشت می‌شوند. نوع ساخت‌یافته‌ی BITS به نوع انتخابی string-value از انتخاب SimpleSyntax نگاشت می‌شود. یک مقدار BITS به صورت یک OCTET STRING کد می‌شود. این کدگذاری بدین صورت است که در آن، تمام بیت‌های نام‌دار^۴ در bitstring، که از اولین بیت آغاز و تا بیت آخر ادامه می‌یابد، در بیت هشتم (بیت پرارزش) تا بیت اول (بیت کم ارزش) از اولین هشت‌تایی قرار می‌گیرند و سایر بیت‌ها به ترتیب در هشت‌تایی‌های بعدی، تا جایی که مورد نیاز است، تا آخرین هشت‌تایی قرار می‌گیرند. اگر بیت‌هایی از آخرین هشت‌تایی بدون استفاده بمانند، فرستنده به جای آن‌ها صفر می‌گذارد و گیرنده این بیت‌ها را نادیده می‌گیرد.

۳-۱- تعاریف

نحو یک PDU با استفاده از نمادگذاری ASN.۱ تعریف می‌شود.

SNMPv۲-PDU DEFINITIONS ::= BEGIN

ObjectName ::= OBJECT IDENTIFIER

ObjectSyntax ::= CHOICE {
 simple SimpleSyntax,
 application-wide ApplicationSyntax }

SimpleSyntax ::= CHOICE {

-
- fragmentation^۱
 - encoding^۲
 - User Datagram Protocol^۳
 - Named bits^۴

```
integer-value INTEGER (-2147483648..2147483647),
string-value OCTET STRING (SIZE (0..65535)),
objectID-value OBJECT IDENTIFIER }
```

```
ApplicationSyntax ::= CHOICE {
    ipAddress-value IpAddress,
    counter-value Counter32,
    timeticks-value TimeTicks,
    arbitrary-value Opaque,
    big-counter-value Counter64,
    unsigned-integer-value Unsigned32 }
```

```
IpAddress ::= [APPLICATION 0] IMPLICIT OCTET STRING (SIZE (4))
```

```
Counter32 ::= [APPLICATION 1] IMPLICIT INTEGER (0..4294967295)
```

```
Unsigned32 ::= [APPLICATION 2] IMPLICIT INTEGER (0..4294967295)
```

```
Gauge32 ::= Unsigned32
```

```
TimeTicks ::= [APPLICATION 3] IMPLICIT INTEGER (0..4294967295)
```

```
Opaque ::= [APPLICATION 4] IMPLICIT OCTET STRING
```

```
Counter64 ::= [APPLICATION 5]
    IMPLICIT INTEGER (0..18446744073709551615)
```

```
-- protocol data units
```

```
PDU ::= CHOICE {
    get-request      GetRequest-PDU,
    get-next-request  GetNextRequest-PDU,
    get-bulk-request  GetBulkRequest-PDU,
    response         Response-PDU,
    set-request       SetRequest-PDU,
    inform-request    InformRequest-PDU,
    snmpV2-trap      SNMPv2-Trap-PDU,
    report            Report-PDU }
```

```
-- PDUs
```

```
GetRequest-PDU ::= [0] IMPLICIT PDU
```

```
GetNextRequest-PDU ::= [1] IMPLICIT PDU
```

```
Response-PDU ::= [2] IMPLICIT PDU
```

```
SetRequest-PDU ::= [3] IMPLICIT PDU
```

```
-- [4] is obsolete
```

```
GetBulkRequest-PDU ::= [5] IMPLICIT BulkPDU
```

```
InformRequest-PDU ::= [6] IMPLICIT PDU
```

```
SNMPv2-Trap-PDU ::= [7] IMPLICIT PDU
```

```
-- Usage and precise semantics of Report-PDU are not defined
-- in this document. Any SNMP administrative framework making
-- use of this PDU must define its usage and semantics.
```

Report-PDU ::= [λ] IMPLICIT PDU

max-bindings INTEGER ::= 2147483647

PDU ::= SEQUENCE {
 request-id INTEGER (-2147483648..2147483647),

 error-status -- sometimes ignored
 INTEGER {
 noError(\cdot),
 tooBig(\cdot),
 noSuchName(\cdot), -- for proxy compatibility
 badValue(\cdot), -- for proxy compatibility
 readOnly(\cdot), -- for proxy compatibility
 genErr(Δ),
 noAccess(\cdot),
 wrongType(\cdot),
 wrongLength(\cdot),
 wrongEncoding(\cdot),
 wrongValue(\cdot),
 noCreation(\cdot),
 inconsistentValue(\cdot),
 resourceUnavailable(\cdot),
 commitFailed(\cdot),
 undoFailed(\cdot),
 authorizationError(\cdot),
 notWritable(\cdot),
 inconsistentName(\cdot)
 },
 },

error-index -- sometimes ignored
 INTEGER (\cdot ..max-bindings),
variable-bindings -- values are sometimes ignored
VarBindList
}

BulkPDU ::= -- must be identical in
SEQUENCE { -- structure to PDU
 request-id INTEGER (-2147483648..2147483647),
 non-repeaters INTEGER (\cdot ..max-bindings),
 max-repetitions INTEGER (\cdot ..max-bindings),
 variable-bindings -- values are ignored
 VarBindList
}

-- variable binding

VarBind ::= SEQUENCE {
 name ObjectName,

 CHOICE {
 value ObjectSyntax,
 unspecified NULL, -- in retrieval requests

 -- exceptions in responses
 noSuchObject [\cdot] IMPLICIT NULL,

```

noSuchInstance [۱] IMPLICIT NULL,
endOfMibView [۲] IMPLICIT NULL
}
-- variable-binding list

VarBindList ::= SEQUENCE (SIZE (۰..max-bindings)) OF VarBind

END

```

۴-۱- ویژگی‌های پروتکل

۴-۱-۱- ساختارهای رایج

مقدار فیلد request-id در پیام پاسخ، مساوی مقدار فیلد request-id در پیام درخواست متناظر این پاسخ است. یک برنامه می‌تواند با استفاده از این فیلد، درخواست‌های منتظر را از هم تشخیص دهد و به این ترتیب پاسخ‌های دریافتی نظیر هر یک از درخواست‌ها را بیابد. در شرایطی که یک سرویس دیتاگرام غیرقابل اعتماد استفاده شود، request-id برای تشخیص پیام‌هایی که دومرتبه ارسال شده‌اند، بکار می‌رود. استفاده از request-id مشابه در بازارسال یک درخواست نیز این امکان را می‌دهد که پاسخ بسته‌ی ارسال شده یا باز ارسال شده نیاز درخواست ارسال شده را برآورده کنند. با این حال برای محاسبه‌ی زمان رفت و برگشت (RTT)^۱ برای انتقال و پردازش یک تراکنش درخواست-پاسخ، این برنامه باید از مقدار request-id متفاوتی برای پیام بازارسالی استفاده کند. این راهبرد برای استفاده در اغلب شرایط بکار می‌رود.

مقدار ناصفر فیلد error-status در پیام پاسخ نشان‌دهنده‌ی وقوع خطا در بسته‌ی ارسالی است که مانع پردازش آن شده است. در این شرایط، مقدار ناصفر فیلد error-index در پیام پاسخ نشان می‌دهد که کدام انقیاد متغیر در یک لیست موجب خطا شده است. هر انقیاد متغیر توسط مقدار شاخص^۲ آن شناسایی می‌شود. اولین شاخص در یک انقیاد متغیر یک، دومین دو و بقیه به همین ترتیب هستند.

SNMP مقادیر OBJECT IDENTIFIER را به حداکثر صد و بیست و هشت زیرشاخص^۳ محدود می‌کند که حداکثر مقدار هر زیرشاخص ۱-۳۲*۲ است.

۴-۱-۲- پردازش پیام

در عناصر رویه‌ی زیر، تمامی فیلدهای یک پیام که توسط رویه‌ی فوق مورد ارجاع قرار نگرفته‌اند، توسط موجودیت SNMP دریافت‌کننده نادیده گرفته می‌شوند. با این حال، تمامی اجزای یک پیام، شامل آن‌هایی که مقادیرشان توسط موجودیت دریافت‌کننده نادیده گرفته می‌شوند، باید از نحو و کدگذاری معتبر ASN.۱ پشتیبانی کنند. برای مثال در برخی پیام‌ها (برای مثال GetRequest) تنها نام متغیرها و نه مقدار آن‌ها مهم است. در این شرایط، بخش مقدار در انقیاد متغیر توسط موجودیت SNMP دریافت‌کننده نادیده گرفته می‌شود. مقدار unspecified برای فیلد مقدار در چنین انقیادهایی در نظر گرفته می‌شود.

در برقراری یک ارتباط مدیریتی، wrapper پیام‌ها برای کپسوله کردن PDU با استفاده از "عناصر رویه" ی چهارچوب مدیریتی مورد استفاده ساخته می‌شود. تعریف "max-bindings" یک کران بالا برای تعداد انقیاد متغیرها تعریف می‌کند. همچنین اندازه‌ی یک پیام توسط محدودیت‌های روی بیشینه اندازه‌ی پیام‌ها محدود می‌شود. یک پیاده‌سازی سازگار باید در هر پیام PDU یا BulkPDU حداکثر تعداد متغیرها را با رعایت محدودیت اندازه در SNMP مقداردهی کند. اما نباید بیش از ۲۱۴۷۴۸۳۶۴۷ انقیاد متغیر را انجام دهد.

^۱ Round Trip Time
^۲ index
^۳ sub-identifier

در دریافت یک ارتباط مدیریتی، "عناصر رویه"ی چهارچوب مدیریتی مورد استفاده پیروی می‌شوند و اگر این رویه‌ها نشان دهند که عملیاتی که در این پیام قرار دارند باید به صورت محلی اجرا گردند، این رویه‌ها نشانگر MIB view ی قابل مشاهده برای عملیات هستند.

۳-۴-۱- پیام GetRequest

یک پیام GetRequest با درخواست یک برنامه تولید و منتقل می‌گردد. موجودیت SNMP دریافت‌کننده با دریافت یک پیام GetRequest، هر انقیاد متغیر در یک لیست از انقیادهای را پردازش می‌کند تا یک پیام پاسخ تولید نماید. مقادیر تمامی فیلدهای پیام پاسخ با پیام درخواست دریافتی یکسان است و تنها در فیلدهای زیر با هم متفاوتند. هر انقیاد متغیر به صورت زیر پردازش می‌شود:

(۱) اگر نام استفاده شده در انقیاد متغیر دقیقاً با نام متغیر قابل دسترسی با این درخواست یکسان باشد، فیلد مقدار در انقیاد متغیر به متغیر نام‌برده داده می‌شود.

(۲) در غیر اینصورت، اگر پیشوند OBJECT IDENTIFIER نام استفاده شده در انقیاد متغیر، با پیشوند OBJECT IDENTIFIER هر متغیر قابل دسترسی در این درخواست دقیقاً یکسان نباشد، مقدار این فیلد به "noSuchObject" تغییر می‌یابد.

(۳) در غیر اینصورت، فیلد مقدار در انقیاد متغیر به "noSuchInstance" تنظیم می‌شود. اگر پردازش یک انقیاد متغیر به دلایلی به جز موارد اشاره شده در بالا با شکست مواجه شود، یک پیام پاسخ با فرمت جدید ساخته می‌شود که در آن، فیلدهای request-id و variable-binding مقادیر نظیر این فیلدها در پیام GetRequest را دارند و مقدار فیلد error-status به "genErr" تغییر می‌یابد و مقدار فیلد error-index نیز به شاخص انقیاد متغیر شکست‌خورده داده می‌شود.

در غیر اینصورت، مقدار فیلد error-status در پیام پاسخ به "noError" تغییر می‌یابد و فیلد error-index نیز مقدار صفر می‌گیرد.

پیام پاسخ تولید شده در یک پیام کپسوله می‌شود. اگر اندازه‌ی پیام حاصل کوچک‌تر یا مساوی با هردو محدودیت محلی و بزرگترین اندازه پیام پشتیبانی شده توسط تولیدکننده باشد، به تولیدکننده‌ی پیام GetRequest ارسال می‌شود.

در غیر اینصورت، یک پیام پاسخ جایگزین ساخته می‌شود. این پیام مقدار فیلد request-id در پیام GetRequest دریافتی را در فیلد request-id خود دارد. همچنین مقدار فیلد error-status آن "tooBig"، مقدار فیلد errorIndex آن صفر و فیلد variable-bindings آن خالی است. این پیام پاسخ جایگزین در یک پیام دیگر کپسوله می‌شود. در صورتیکه اندازه‌ی پیام حاصل کوچک‌تر یا مساوی هردو محدودیت‌های محلی و اندازه‌ی بزرگ‌ترین پیام پشتیبانی شده توسط تولیدکننده باشد، به تولیدکننده‌ی پیام GetRequest ارسال می‌شود. در غیر اینصورت شمارنده‌ی snmpSilenceDrops [RFC۳۴۱۸] افزایش می‌یابد و پیام حاصل دور انداخته می‌شود.

۴-۴-۱- پیام GetNextRequest

یک پیام GetNextRequest با درخواست یک برنامه تولید و منتقل می‌شود. با دریافت یک پیام GetNextRequest، موجودیت SNMP دریافت‌کننده هر انقیاد متغیر در لیست variable-binding را پردازش می‌کند تا یک پیام پاسخ تولید کند. تمامی فیلدهای پیام پاسخ مقادیر فیلدهای پیام دریافت شده را دارند و تنها در موارد زیر با یکدیگر متفاوتند. هر انقیاد متغیر به صورت زیر پردازش می‌شود:

(۱) مکان متغیر در لیستی از نام‌های تمام متغیرها که با ترتیب حرف به حرف مرتب گشته‌اند، مشخص می‌گردد. می‌توان با این درخواست به این متغیر دسترسی یافت و نام آن اولین نام در ترتیب حرف به

حرف پس از نام ذکر شده در انقیاد متغیر در پیام `GetNextRequest` دریافتی است. فیلدهای نام و مقدار در پیام پاسخ، نام و مقدار متغیر فوق را می‌گیرند.

(۲) اگر نام متغیر درخواست شده در انقیاد متغیر، پیش از نام هیچ متغیر دیگر قابل دسترس با این درخواست در ترتیب حرف به حرف نباشد، یعنی هیچ نام دیگری پس از آن در ترتیب حرف به حرف نیامده باشد، انقیاد متغیر فوق که در پیام پاسخ صورت می‌گیرد دارای فیلد مقدار با مقدار `endOfMibView` است و فیلد نام در آن به نام انقیاد متغیر در پیام درخواست، تنظیم می‌شود.

(۳) اگر پردازش یک انقیاد متغیر به دلیلی خارج از موارد بالا با شکست مواجه شود، پیام پاسخ دارای فیلدهای `request-id` و `variable-binding` یکسانی با پیام `GetNextRequest` دریافتی است. مقدار فیلد `error-status` آن `genErr` و مقدار فیلد `error-index` آن به شاخص انقیاد متغیر خطا خورده تنظیم می‌شود.

در غیر اینصورت مقدار فیلد `error-status` در پیام پاسخ به `noError` و فیلد `error-index` نیز به صفر تنظیم می‌شوند.

سپس پیام پاسخ تولید شده در یک پیام کپسوله می‌شود. اگر اندازه‌ی پیام حاصل کوچک‌تر یا مساوی هردو محدودیت‌های محلی و بیشینه‌ی اندازه‌ی پیام پشتیبانی شده توسط تولیدکننده باشد، پیام پاسخ به تولیدکننده‌ی پیام `GetNextRequest` ارسال می‌گردد.

در غیر اینصورت، یک پیام پاسخ جایگزین ساخته می‌شود. این پیام دارای فیلد `request-id` یکسان با پیام `GetNextRequest` دریافتی است و مقدار فیلد `error-status` آن `tooBig`، مقدار فیلد `error-index` آن صفر و فیلد `variable-bindings` آن خالی است. پیام پاسخ جایگزین سپس در یک بسته کپسوله می‌شود. اگر اندازه‌ی پیام حاصل کوچک‌تر یا مساوی هردو محدودیت‌های محلی و بیشینه‌ی اندازه‌ی یک پیام در تولیدکننده باشد، این پیام به تولیدکننده‌ی پیام `GetNextRequest` ارسال می‌گردد. در غیر اینصورت، شمارنده‌ی `snmpSilentDrops` [RFC۳۴۱۸] یک واحد افزایش می‌یابد و پیام حاصل دور انداخته می‌شود.

□ مثالی از پیمایش جدول

یک کاربرد مهم از پیام `GetNextRequest`، پیمایش جدول‌های اطلاعاتی متصور^۱ در یک MIB است. نحو این نوع درخواست همراه با روش شناسایی نمونه‌های منفرد از اشیاء در MIB، دسترسی به اشیاء فوق از MIB که به صورت جدولی سازمان‌دهی شده‌اند را فراهم می‌کند.

در تبادل پروتکلی نمایش داده شده در زیر، یک برنامه آدرس فیزیکی وابسته به رسانه و نوع نگاشت آدرس برای هر سطر از جدول ترجمه‌ی آدرس IP شبکه-رسانه [RFC۱۲۱۳] نظیر یک عنصر شبکه مشخص را بدست می‌آورد. این برنامه همچنین مقدار `sysUpTime` [RFC۳۴۱۸] که نگاشت‌ها در آن وجود داشتند را بدست می‌آورد. فرض کنید جدول آدرس IP-رسانه عنصر پاسخ‌دهنده‌ی فرمان سه سطر داشته باشد:

| نوع | آدرس فیزیکی | آدرس شبکه | شماره واسط |
|---------|-------------------|-----------|------------|
| static | ۰۰:۰۰:۱۰:۰۱:۲۳:۴۵ | ۱۰.۰.۰.۵۱ | ۱ |
| dynamic | ۰۰:۰۰:۱۰:۵۴:۳۲:۱۰ | ۹.۲.۳.۴ | ۱ |
| dynamic | ۰۰:۰۰:۱۰:۹۸:۷۶:۵۴ | ۱۰.۰.۰.۱۵ | ۲ |

موجودیت SNMP که از یک برنامه‌ی تولیدکننده‌ی فرمان پشتیبانی می‌کند، کار خود را با ارسال یک پیام `GetNextRequest` که شامل مقادیر OBJECT IDENTIFIER نظیر نام متغیرهای درخواستی است، شروع

می کند.

```

getNextRequest ( sysUpTime,
                  ipNetToMediaPhysAddress,
                  ipNetToMediaType )

```

موجودیت SNMP که از یک برنامه‌ی پاسخگوی فرمان پشتیبانی می‌کند یک پیام پاسخ به صورت زیر ارسال می‌کند:

```
Response (( sysUpTime. = "133456" ),
          ( ipNetToMediaPhysAddress.1.9.2.3.4 = "...1.54321." ),
          ( ipNetToMediaType.1.9.2.3.4 = "dynamic" ))
```

موجودیت SNMP که از یک برنامه‌ی تولیدکننده‌ی فرمان پشتیبانی می‌کند به صورت زیر ادامه می‌دهد:

```
GetNextRequest ( sysUpTime,  
                 ipNetToMediaPhysAddress.1.1...Δ1,  
                 ipNetToMediaType.1.1...Δ1 )
```

موجودیت SNMP که از یک برنامه‌ی پاسخگوی فرمان پشتیبانی می‌کند با پیام زیر پاسخ می‌دهد:

```
Response (( sysUpTime.0 = "123456" ),
          ( ipNetToMediaPhysAddress.2.1.0.0.15 = "...10987654" ),
          ( ipNetToMediaType.2.1.0.0.15 = "dynamic" ))
```

موجودیت SNMP که از برنامه‌ی تولیدکننده‌ی فرمان پشتیبانی می‌کند با ارسال پیام زیر ادامه می‌دهد:

```
GetNextRequest ( sysUpTime,  
                 ipNetToMediaPhysAddress.Ȳ.ȳ.ȴ.ȵ.ȶ,  
                 ipNetToMediaType.Ȳ.ȳ.ȴ.ȵ.ȶ )
```

از آنجا که سطرهای بیشتری در جدول وجود ندارند، موجودیت SNMP پاسخگوی فرمان با متغیرهایی که بعد از نام اشیاء قابل دسترسی در ترتیب حرف به حرف می‌آیند، پاسخ می‌دهد. برای مثال:

```
Response (( sysUpTime.⋅ = "123456" ),
          ( ipNetToMediaNetAddress.1.2.3.4 = "9.8.7.6" ),
          ( ipRoutingDiscards.⋅ = "7" ))
```

توجه کنید که چگونه با رسیدن به انتهای ستون `ipNetToMediaPhysAddress`، دومین انقیاد متغیر از برنامه‌ی پاسخگوی فرمان، در اولین سطر از ستون بعدی پیچیده شده است.

همچنین توجه کنید که چگونه با رسیدن به انتهای ipNetToMedia برای سومین انقیاد متغیر، برنامه‌ی پاسخگوی فرمان با شی موجود بعدی پاسخ داده است که این شی خارج از جدول است. این پاسخ به برنامه‌ی تولیدکننده‌ی فرمان اطلاع می‌دهد که جدول به پایان رسیده است.

GetBulkRequest ٥-٤-١ - پیام

یک پیام `GetBulkRequest` بنابر درخواست یک برنامه‌ی کاربردی تولید می‌شود و منتقل می‌گردد. هدف این پیام درخواست نمودن انتقال حجم زیادی از داده، شامل (و نه محدود به) درخواست سریع و کارآمد جداول بزرگ است.

با دریافت یک پیام `GetBulkRequest`، موجودیت `SNMP` دریافت‌کننده، هر انقیاد متغیر در لیست `variable-binding` را پردازش می‌کند تا یک پیام پاسخ با مقدار فیلد `request-id` مشابه پیام درخواست تولید کند. برای نوع پیام `GetBulkRequest`، پردازش موفقیت‌آمیز هر انقیاد متغیر در درخواست، صفر یا تعداد بیشتری انقیاد متغیر در پیام پاسخ تولید می‌کند. به این ترتیب وضعیت نگاشت یک به یک بین انقیاد متغیرها در پیام `GetRequest`، `GetNextRequest` و `SetRequest` و پیام‌های پاسخ حاصل برای نگاشت بین انقیاد متغیرهای پیام `GetBulkRequest` و پیام پاسخ حاصل، برقرار نیست.

مقادیر فیلدهای non-repeaters و non-repetitions در پیام درخواست، پردازش مورد درخواست را

مشخص می‌کنند. در پیام پاسخ برای N انقیاد متغیر اول در پیام درخواست یک پیام و برای هر یک از R انقیاد متغیر باقی‌مانده در درخواست، M انقیاد متغیر درخواست می‌شود. به این ترتیب تعداد کل انقیاد متغیرهای درخواست شده توسط $N+M*R$ نشان داده می‌شود و در آن، N کمترین میان (۱) مقدار فیلد non-repeaters در پیام درخواست و (۲) تعداد انقیاد متغیرها در درخواست، M مقدار فیلد max-repetitions در درخواست و R بزرگ‌ترین میان (۱) تعداد انقیاد متغیرها در پیام درخواست (N و ۲) صفر است.

موجودیت SNMP دریافت‌کننده یک پیام پاسخ با حداکثر تعداد کل انقیاد متغیرهای درخواست شده که توسط پیام درخواست تبادل شده است، می‌سازد. Request-id پیام پاسخ باید با request-id پیام GetBulkRequest دریافتی یکسان باشد. اگر N بزرگ‌تر از صفر باشد، اولین تا N امین انقیاد متغیر در پیام پاسخ به صورت زیر تولید می‌شوند:

(۱) یک متغیر مکان‌یابی می‌شود. این کار بدین صورت است که در یک لیست مرتب شده از نام تمام متغیرهای قابل دسترسی توسط این درخواست به صورت حرف به حرف^۱، مکان آن مشخص می‌گردد و نام آن اولین کلمه بعد از نام انقیاد متغیر در پیام GetBulkRequest در ترتیب حرف به حرف است. فیلدهای نام و مقدار در انقیاد متغیر در پیام پاسخ به نام و مقدار متغیر مکان‌یابی شده تنظیم می‌شود. (۲) اگر نام انقیاد متغیر درخواست شده پیش از نام هیچ یک از متغیرهای قابل دسترسی توسط این درخواست نیامده باشد، در واقع هیچ نام دیگری در ترتیب حرف به حرف پس از این نام نیامده باشد، انقیاد متغیر تولید شده در پیام پاسخ مقدار فیلد مقدارش به "endOfMibView" و مقدار فیلد نام آن به نام در انقیاد متغیر در پیام درخواست تنظیم می‌شود.

اگر M و N ناصفر باشند، ($N+1$) امین و نیز انقیاد متغیرهای بعدی در پیام پاسخ به صورت مشابه تولید می‌گردند. برای هر تکرار i ، به طوریکه i بزرگ‌تر یا مساوی صفر و کوچک‌تر یا مساوی M است و برای هر متغیر تکراری r که بزرگ‌تر یا مساوی صفر و کوچک‌تر یا مساوی R است، ($N+(i-1)*R+r$) امین انقیاد متغیر در پیام پاسخ به صورت زیر تولید می‌گردد:

(۱) متغیری که در لیستی از نام‌های تمام متغیرهای قابل دسترسی توسط این درخواست که به صورت حرف به حرف مرتب شده‌اند، مکان‌یابی می‌شود. نام این متغیر i امین بعدی در ترتیب حرف به حرف پس از ($N+r$) امین نام انقیاد متغیر در پیام GetBulkRequest دریافتی است. فیلدهای نام و مقدار انقیاد متغیر به نام و مقدار متغیر مکان‌یابی شده تنظیم می‌شود. (۲) اگر در ترتیب حرف به حرف هیچ نامی پس از i امین نام وجود نداشته باشد، انقیاد متغیر متناظر که در پیام پاسخ آمده است، دارای فیلد مقدار برابر با "endOfMibView" و فیلد نام مساوی با آخرین نام بعدی در ترتیب حرف به حرف و یا در صورت وجود نداشتن هیچ بعدی‌ای در ترتیب حرف به حرف، ($N+1$) امین نام در انقیاد متغیر در پیام درخواست است.

گرچه بیشینه تعداد انقیاد متغیرها در پیام پاسخ توسط $N+(M*R)$ محدود می‌شود، ممکن است پیام پاسخ با تعداد کمتری از انقیاد متغیرها که صفر نیز می‌تواند باشد، تولید گردد. علت این مسئله هر یک از سه مورد زیر می‌تواند باشد:

(۱) اگر اندازه‌ی پیام پاسخ پیچنده‌ی پیام پاسخ که شامل تعداد درخواستی از انقیاد متغیرها است بزرگ‌تر یا مساوی محدودیت محلی یا بیشترین اندازه‌ی پیام‌های تولیدکننده باشد، پیام پاسخ با تعداد کمتری از انقیاد متغیرها ساخته می‌شود. این تعداد کمتر در حقیقت یک مجموعه‌ی مرتب از انقیاد متغیرها است که تعدادی از انقیاد متغیرها در انتهای مجموعه از آن حذف شده‌اند، به طوریکه اندازه‌ی پیامی که پیام پاسخ را می‌پیچد به صورت تقریبی مساوی اما نه بزرگ‌تر محدودیت محلی یا محدودیت روی بیشینه اندازه‌ی پیام در موجودیت تولیدکننده باشد. دقت کنید که تعداد انقیاد متغیرهای حذف شده هیچ ارتباطی با مقادیر N ، M یا R ندارد.

۲) اگر برای برخی مقادیر تکرار i ، بطوریکه i بزرگتر از صفر یا کوچکتر از M است، تمامی فیلدهای انقیاد متغیر تولید شده دارای مقدار "endOfMibView" باشند، پاسخ نیز می‌تواند با تعداد کمتری انقیاد متغیر ساخته شود. در این شرایط، انقیاد متغیرها ممکن است پس از $(N+(i \cdot R))$ امین انقیاد متغیر صورت گیرد.

۳) در شرایطی که پردازش یک درخواست که تعداد زیادی repetitions دارد، مستلزم زمان پردازش بسیار بیشتری نسبت به یک درخواست عادی باشد، یک برنامه‌ی پاسخگوی فرمان ممکن است به پیام درخواست را با تعداد repetitionهای کمتری پردازش کند. البته با این شرط که حداقل یک repetition را کامل کند.

اگر پردازش یک انقیاد متغیر به دلیلی خارج از موارد لیست بالا متوقف شود، پیام پاسخ با فرمت جدید ساخته می‌شود که فیلدهای request-id و variable-bindings آن برابر با مقدار این فیلدها در پیام درخواست شده هستند. مقدار فیلد error-status نیز به "genErr" و مقدار فیلد error-index به شاخص انقیاد متغیر در درخواست اصلی تنظیم می‌شود که متناظر با انقیاد متغیرهای شکست خورده است.

در غیر اینصورت، مقدار فیلد error-status در پیام پاسخ به "noError" و مقدار فیلد error-index به صفر تنظیم می‌شود.

پیام پاسخ تولید شده (که ممکن است فیلد variable-bindings آن خالی باشد) در یک پیام کپسوله می‌شود. اگر اندازه‌ی پیام حاصل کوچکتر یا مساوی هردو محدودیت محلی و اندازه‌ی بزرگترین پیام پشتیبانی شده توسط تولیدکننده باشد، به تولیدکننده‌ی پیام GetBulkRequest ارسال می‌گردد. در غیر اینصورت، شمارنده‌ی snmpSilentDrops [RFC3418] افزایش می‌یابد و پیام حاصل دور انداخته می‌شود.

□ یک مثال دیگر از جدول Traversal

این مثال نشان می‌دهد که چگونه پیام GetBulkRequest می‌تواند به عنوان جایگزینی برای پیام GetNextRequest استفاده شود. بدین ترتیب همان پیمایش جدول IP شبکه-رسانه که در بخش قبل نشان داده شد، با تعداد تبادلهای کمتر بدست می‌آید.

موجودیت SNMP که از برنامه تولیدکننده فرمان پشتیبانی می‌کند با ارسال یک پیام GetBulkRequest با مقدار max-repetition دو، و شامل مقادیر OBJECT IDENTIFIER که به صورت نام متغیر درخواست شده نشان داده شده است، کار خود را آغاز می‌کند.

GetBulkRequest [non-repeaters = ۱, max-repetitions = ۲]

(sysUpTime,
ipNetToMediaPhysAddress,
ipNetToMediaType)

موجودیت SNMP که از برنامه پاسخگوی فرمان پشتیبانی می‌کند با پیام Response پاسخ می‌دهد:

Response ((sysUpTime.۰ = "۱۲۳۴۵۶"),
(ipNetToMediaPhysAddress.۱.۹.۲.۳.۴ = "۰۰۰۰۱۰۵۴۳۲۱۰"),
(ipNetToMediaType.۱.۹.۲.۳.۴ = "dynamic"),
(ipNetToMediaPhysAddress.۱.۱۰.۰.۰.۵۱ = "۰۰۰۰۱۰۰۱۲۳۴۵"),
(ipNetToMediaType.۱.۱۰.۰.۰.۵۱ = "static"))

موجودیت SNMP که از برنامه تولیدکننده فرمان پشتیبانی می‌کند، بدین ترتیب ادامه می‌دهد:

GetBulkRequest [non-repeaters = ۱, max-repetitions = ۲]

(sysUpTime,

ipNetToMediaPhysAddress.۱.۱۰.۰.۰.۵۱,

ipNetToMediaType.۱.۱۰.۰.۰.۵۱)

موجودیت SNMP که از برنامه‌ی پاسخگوی فرمان پشتیبانی می‌کند به صورت زیر پاسخ می‌دهد :

Response ((sysUpTime.۰ = "۱۲۳۴۶۶"),

(ipNetToMediaPhysAddress.۲.۱۰.۰.۰.۱۵ = "۰۰۰۰۱۰۹۸۷۶۵۴"),

(ipNetToMediaType.۲.۱۰.۰.۰.۱۵ = "dynamic"),

(ipNetToMediaNetAddress.۱.۹.۲.۳.۴ = "۹.۲.۳.۴"),

(ipRoutingDiscards.۰ = "۲"))

توجه کنید که چگونه (همانند مثال اول) انقیاد متغیرها در پیام پاسخ نشان می‌دهد که جدول به پایان رسیده است. چهارمین انقیاد متغیر، با برگرداندن اطلاعات از ستون موجود بعدی انجام می‌شود؛ پنجمین انقیاد متغیر، با بازگشت اطلاعات از اولین شیء موجود که به ترتیب حرف به حرف در جدول قرار گرفته است، انجام می‌شود. این پاسخ، رسیدن به انتهای جدول را به برنامه تولیدکننده‌ی فرمان اطلاع می‌دهد.

پیام Response

پیام Response توسط یک موجودیت SNMP تنها پس از دریافت پیام GetRequest، GetNextRequest، GetBulkRequest، SetRequest یا InformRequest تولید می‌شود، همانطور که در جاهای دیگر این سند توضیح داده شده است.

اگر فیلد error-status در پیام Response غیر صفر باشد، فیلدهای مقدار مربوط به متغیر در لیست انقیاد متغیرها، نادیده گرفته می‌شوند.

اگر هر دو فیلد error-status و فیلد error-index در پیام Response غیر صفر باشند، سپس مقدار فیلد error-index، شاخص متغیری در لیست variable-binding درخواست نظیر است که درخواست آن شکست خورده است. اولین انقیاد متغیر در لیست variable-binding درخواست، شاخص اول را دارد، دومین شاخص دوم را دارد و به همین ترتیب.

یک موجودیت SNMP که از یک برنامه‌ی تولیدکننده‌ی فرمان پشتیبانی می‌کند باید قادر باشد به درستی یک پیام Response را با فیلد error-status برابر با "noSuchName"، "badValue" یا "readOnly" اداره کند. (به بخش‌های ۱.۳ و ۴.۳ [RFC۲۵۷۶] مراجعه کنید.)

با دریافت یک پیام Response، موجودیت SNMP دریافت‌کننده، محتویات خود را به برنامه‌ای که این درخواست را با همان مقدار request-id ایجاد کرده است، ارائه می‌دهد. برای جزئیات بیشتر، [RFC۳۴۱۲] را ببینید.

۶-۴-۱ - پیام SetRequest

یک پیام به درخواست یک برنامه SetRequest تولید و ارسال می‌شود.

پس از دریافت یک پیام SetRequest، موجودیت SNMP دریافت‌کننده اندازه‌ی پیامی که پیام Response را کپسوله می‌کند، مشخص می‌کند. فیلدهای request-id و variable-binding در پیام کپسوله‌کننده برابر با مقدار این فیلدها در پیام SetRequest دریافتی است. همچنین بزرگ‌ترین اندازه‌ی ممکن برای فیلدهای error-status و error-index برابر با بزرگ‌ترین اندازه‌ی ممکن در پیام SetRequest دریافتی است. اگر اندازه پیام مشخص شده بیشتر از محدودیت محلی یا حداکثر اندازه پیام پشتیبانی شده توسط مبدأ باشد، سپس یک پیام Response جایگزین تولید می‌شود، و به فرستنده‌ی پیام SetRequest فرستاده می‌شود و پردازش پیام SetRequest بلافاصله پس از آن متوقف می‌شود. این پیام Response دارای فیلد request-id یکسان با پیام اصلی است، مقدار فیلد error-status به "tooBig"، مقدار error-index به صفر تنظیم می‌شوند و فیلد variable-bindings خالی می‌ماند. اگر اندازه پیام حاصل کمتر یا برابر محدودیت محلی و حداکثر اندازه پیام اصلی باشد، به فرستنده پیام SetRequest ارسال می‌شود. در غیر اینصورت، شمارنده

[RFC3418] snmpSilentDrops افزایش می یابد و پیام حاصل از آن حذف می شود. صرفنظر از این، پردازش پیام SetRequest نیز متوقف می شود.

در غیر این صورت، موجودیت دریافت کننده SNMP، هر variable-binding را در لیست variable-binding، برای تولید پیام Response پردازش می کند. همه فیلدهای پیام Response مقادیر مشابهی با فیلدهای متفاوتی از درخواست دریافت شده دارند به غیر از موارد زیر:

variable-binding به صورت مفهومی به عنوان یک عملیات دو مرحله ای پردازش می شوند. در مرحله اول، هر variable-binding تایید می شود؛ اگر تمام اعتبارسنجی ها موفق باشند، هر متغیر در مرحله دوم تغییر می یابد. البته، برای اجرای اولین یا دوم یا هر دو این مراحل مفهومی به عنوان مراحل تکامل چندگانه آزادی عمل وجود دارد. در واقع، در بعضی موارد، ممکن است مراحل تکامل چندگانه برای اطمینان از انسجام لازم باشد.

اعتبار سنجی های زیر در مرحله اول بر روی هر variable-binding تا زمانی که همه موفق شوند یا تا زمانی که یک شکست نباشد اجرا می شوند:

(۱) اگر نام variable-binding نام یک متغیر موجود یا غیر موجود را تعیین کند این درخواست /

دسترسی ممنوع است زیرا در MIB نمی باشد، سپس مقدار فیلد error-status در پیام Response برابر "noAccess" می شود، و مقدار فیلد index-index آن به عنوان متغیر ناموفق نامحدود تنظیم خواهد شد.

(۲) در غیر این صورت، اگر هیچ متغیری وجود نداشته باشد که یک پیشوند OBJECT IDENTIFIER همانند نام متغیر را داشته باشد و بتواند بدون توجه به مقدار جدید تعیین شده ایجاد یا اصلاح شود، مقدار فیلد error-status برابر notWritable تنظیم می شود، و مقدار فیلد index-index آن به نام متغیر fail شده تعلق می گیرد.

(۳) در غیر این صورت، اگر فیلد variable binding، مطابق با زبان ASN.۱، نوعی است که با آن مورد نیاز برای تمام متغیرهایی که یک پیشوند OBJECT IDENTIFIER را همانند نام متغیر مورد استفاده قرار می دهند، مشخص می کند، سپس مقدار پیام Response فیلد وضعیت error-state به "wrongType" تنظیم شده است و مقدار فیلد index-index آن به عنوان متغیر ناموفق نامحدود تنظیم شده است.

(۴) در غیر این صورت، اگر فیلد مقدار متغیر variable binding براساس زبان ASN.۱ مشخص شود، طولی که با مقدار مورد نیاز برای همه متغیرهایی که همان پیشوند OBJECT IDENTIFIER را به عنوان متغیر binding's name به اشتراک می گذارند، متناقض هستند، سپس مقدار فیلد وضعیت خطای Response-PDU's به مقدار "wrongLength" تنظیم میشود، و مقدار فیلد error-index آن به index فیلد variable binding تنظیم می شود.

(۵) در غیر این صورت، اگر فیلد variable binding حاوی کدگذاری ASN.۱ باشد که با برچسب ASN.۱ «errorEncoding» آن فیلد ناسازگار باشد، مقدار فیلد error-status در پیام Response روی «errorEncoding» تنظیم شده و مقدار فیلد error-index آن به index فیلد variable binding متغیر fail شده تنظیم می شود. (توجه داشته باشید که تمام استراتژی های پیاده سازی این خطا را تولید نمی کند.)

(۶) در غیر این صورت، اگر فیلد value-binding یک متغیر که در هیچ شرایطی نمی تواند به متغیر اختصاص داده شود به متغیر داده شود، مقدار فیلد error-status در پیام Response بر روی «wrongValue» تنظیم شده و مقدار فیلد error-index آن برابر index مربوط به انقیاد متغیر شکست خورده تنظیم شده می شود.

(۷) در غیر این صورت، اگر نام variable binding متغیری را ایجاد کند که وجود ندارد و هرگز نمی تواند ایجاد شود (حتی اگر برخی از متغیرهایی که یک پیشوند OBJECT IDENTIFIER به اشتراک گذاشته شوند ممکن باشد در بعضی شرایط ایجاد شود)، سپس مقدار پیام Response فیلد وضعیت

error-state به "noCreation" تنظیم شده، و مقدار فیلد error-index آن به index مربوط به variable-binding شکست خورده تنظیم می‌شود.

(۸) در غیر این صورت، اگر نام binding-variable متغیری را تعیین کند که وجود ندارد، اما نمی‌تواند تحت شرایط فعلی ایجاد شود (حتی اگر در شرایط دیگر ایجاد شود)، مقدار فیلد error-status در پیام Response بر روی "inconsistentName" و مقدار فیلد error-index آن به index مربوط به variable-binding شکست خورده تنظیم می‌شود.

(۹) در غیر این صورت، اگر نام variable-binding متغیری را تعیین کند که وجود دارد، اما قابل تغییر نیست فارغ از اینکه چه مقدار جدیدی تعیین شده، سپس مقدار وضعیت پیام error-state Response بر روی "notWritable" تنظیم شده و مقدار فیلد error-index آن به index مربوط به variable-binding شکست خورده تنظیم می‌شود.

(۱۰) در غیر این صورت، اگر نام variable-binding متغیری را تعیین کند که می‌تواند تحت شرایط دیگر توسط متغیر نگهداری شود، اما در حال حاضر ناسازگار است یا در غیر اینصورت قادر به اختصاص به متغیر نیست، سپس مقدار error-state مربوط به پیام Response برابر "inconsistentValue" تنظیم می‌شود و مقدار فیلد error-index آن به شاخص مربوط به variable-binding شکست خورده تنظیم می‌شود.

(۱۱) هنگامی که در طی مراحل فوق، اختصاص مقدار توسط فیلد variable-binding به متغیر مشخص شده، نیاز به تخصیص یک منبع دارد که در حال حاضر در دسترس نیست، سپس مقدار error-status در پیام Response برابر "resourceUnavailable" و مقدار فیلد error-index آن به شاخص مربوط به variable-binding شکست خورده تنظیم می‌شود.

(۱۲) اگر پردازش variable-binding به دلیل دیگری غیر از موارد فوق شکست بخورد، سپس مقدار error-status در پیام Response بر روی genErr تنظیم می‌شود و مقدار فیلد error-index آن به شاخص مربوط به variable-binding شکست خورده تنظیم می‌شود.

(۱۳) در غیر این صورت، اعتبارسنجی variable-binding موفق می‌شود.

در پایان فاز اول، اگر اعتبارسنجی تمام variable-binding ها موفق شود، مقدار فیلد error-status در پیام Response به "noError" تنظیم می‌شود و مقدار فیلد index-index آن صفر است و پردازش بشرح زیر ادامه دارد.

برای هر variable-binding در درخواست، متغیر نامی در صورت لزوم ایجاد می‌شود و مقدار مشخص شده به آن اختصاص داده می‌شود. هر یک از این تخصیص متغیرها به طور همزمان با توجه به تمام تخصیص های دیگر مشخص شده در همان درخواست رخ می‌دهد. با این حال، اگر یک متغیر همزمان یک درخواست را بیش از یک بار، با مقادیر مختلف نام گذاری کند، سپس تخصیص واقعی به آن متغیر، implementation-specific است.

اگر هر یک از این تخصیص ها شکست خورده باشد (حتی پس از تمام اعتبارسنجی های قبلی)، تمام انتصاب های دیگر لغو می‌شود، و پاسخ پیام اصلاح می‌شود تا مقدار فیلد error-status در آن را به "commitFailed" تنظیم کند و مقدار فیلد error-index آن به شاخص مربوط به variable-binding شکست خورده تنظیم می‌شود.

اگر و فقط اگر تمام تخصیص ها را لغو نکنیم، سپس پیام Response اصلاح می‌شود تا مقدار فیلد error-status در آن به «undoFailed» تنظیم شود و مقدار فیلد index-index آن به صفر برسد. توجه داشته باشید که پیاده سازی ها به شدت تشویق می‌شوند تا تمام اقدامات احتمالی را برای جلوگیری از استفاده از "commitFailed" یا "undoFailed" انجام دهند.

در نهایت، پیام Response تولید شده به یک پیام کپسوله شده تبدیل و به فرستنده پیام SetRequest ارسال می‌شود.

۷-۴-۱- پیام SNMPv۲-Trap

پیام SNMPv۲-Trap توسط یک موجودیت SNMP تولید شده و توسط یک برنامه اعلان ارسال می شود. پیام SNMPv۲-Trap اغلب برای اعلان یک برنامه گیرنده اعلانات در یک موجودیت کنترل از راه دور منطقی SNMP است که یک رویداد اتفاق افتاده یا اینکه در شرایط موجود است. هیچ تأییدی در ارتباط با این مکانیزم تحویل اعلان وجود ندارد.

مقصودی (هایی) که یک پیام SNMPv۲-Trap برای آن ها ارسال می شود، توسط یک موجودیت SNMP تعیین می شود. دو دسته انقیاد متغیر در لیست variable-binding پیام SNMPv۲ Trap به ترتیب [RFC۳۴۱۸] sysUpTime.۰ و [RFC۳۴۱۸] snmpTrapOID.۰ هستند. اگر OBJECTS clause در فراخوانی مربوط به TYPE NOTIFICATION وجود داشته باشد، هر یک از متغیرهای مربوطه به عنوان نمونه ای از این notification، به ترتیب، به فیلد variable binding کپی می شود. اگر هر متغیر اضافی شامل می شود (در گزینه ای از ساختار SNMP)، سپس هر کدام به فیلد variable bindings کپی می شود.

۸-۴-۱- پیام InformRequest

یک پیام InformRequest بوسیله ی یک موجودیت SNMP از طرف برنامه سازنده اعلانتولید و ارسال می شود. پیام InformRequest اغلب برای اعلان یک برنامه دریافت کننده اعلانه یک رویداد که اتفاق افتاده است یا وضعیت موجود مورد استفاده قرار می گیرد. این یک مکانیزم تحویل اعلانتأیید شده است، البته، تضمین تحویل وجود ندارد.

مقصودی (هایی) که یک پیام InformRequest برای آن ها ارسال می شود توسط برنامه ایجاد کننده اعلانشخص می شوند. دو دسته variable binding اول در لیست variable binding پیام InformRequest به ترتیب [RFC۳۴۱۸] sysUpTime.۰ و [RFC۳۴۱۸] snmpTrapOID.۰ هستند. اگر OBJECTS clause در فراخوانی مربوطه macro NOTIFICATION-TYPE وجود داشته باشد، هر یک از متغیرهای مربوطه به عنوان نمونه ای از این notification، به ترتیب، به فیلد variable-bindings کپی می شود. اگر هر متغیر اضافی را شامل شود (در گزینه ای از ساختار SNMP)، سپس هر کدام به فیلد variable-bindings کپی می شود.

پس از دریافت پیام InformRequest، موجودیت SNMP دریافتی اندازه پیام کپسوله شده Response را با مقادیر مشابه در request-id، error-status، error-index و variable-bindings به عنوان پیام InformRequest دریافت شده تعیین می کند. اگر اندازه پیام مشخص شده بیشتر از محدودیت محلی یا حداکثر اندازه پیام اصلی است، سپس یک واکنش پیام متناوب تولید می شود، به فرستنده پیام InformRequest ارسال می شود و پردازش پیام InformRequest بلافاصله پس از آن متوقف می شود. این پاسخ جایگزین پیام با مقادیر مشابه در request-id field خود به عنوان پیام InformRequest دریافت می شود، با مقدار فیلد error-index خود را صفر تنظیم می کند و یک فیلد variable-bindings خالی جایگزین می شود. اگر اندازه پیغام نتیجه کمتر یا برابر محدودیت محلی و حداکثر اندازه پیام اصلی باشد، به فرستنده پیام InformRequest ارسال می شود. در غیر اینصورت، شمارنده [RFC۳۴۱۸] snmpSilentDrops افزایش می یابد و پیام حاصل از آن حذف می شود. صرف نظر از این، پردازش پیام InfoRequest متوقف می شود. در غیر این صورت، یک موجودیت SNMP دریافتی:

(۱) مطالب خود را به نرم افزار مناسب ارائه می کند.

(۲) یک پیام Response با همان مقادیر در فیلدهای request-id و variable-bindings به عنوان پیام Receive InformRequest تولید می کند، که مقدار فیلد error-status آن که به "noError" و مقدار فیلد error-index آن را صفر تعیین می کند. و

(۳) پیام Response تولید شده را به فرستنده پیام InformRequest ارسال می کند.

۵-۱- اطلاع از مالکیت معنوی

IETF در مورد اعتبار یا محدوده هر مالکیت معنوی یا حقوق دیگر مربوط به پیاده سازی که ممکن است ادعا شوند یا استفاده از فن آوری توصیف شده در این سند یا میزان مجاز بودن هر مجوز که ممکن است یا ممکن نیست در دسترس باشند هیچ جایگاهی ندارد. نشان نمی دهد تلاش برای شناسایی صحت چنین حقوقی را انجام شده باشد. اطلاعات در مورد روش های IETF در رابطه با حقوق در مستندات مربوط به استانداردهای پیگیری و استانداردهای ارتباطات در BCP-۱۱ یافت می شود. کپی های ادعای حقوقی که برای انتشار و هر گونه تضمین مجوزها در اختیار قرار گرفته است و یا نتیجه تلاش برای کسب مجوز عمومی یا اجازه استفاده از چنین حقوق مالکیت توسط اجراکنندگان یا کاربران این مشخصات می تواند از دبیرخانه IETF به دست آید. IETF هر یک از طرفین ذینفع را دعوت می کند تا به هر گونه حق نسخه برداری، اختراع ثبت شده یا پرونده های ثبت اختراع و یا سایر حقوق مالکیت که ممکن است شامل تکنولوژی هایی باشد که ممکن است برای انجام این استاندارد مورد نیاز باشد، توجه کنند.

۶-۱- قدردانی

این سند محصول کارگروه SNMPv۳ است. از اعضای کارگروه زیر تشکر ویژه ای داریم:

Randy Bush
Jeffrey D. Case
Mike Daniele
Rob Frye
Lauren Heintz
Keith McCloghrie
Russ Mundy
David T. Perkins
Randy Presuhn
Aleksey Romanov
Juergen Schoenwaelder
Bert Wijnen

این نسخه از سند، ویرایش شده توسط رندی پرسون، در ابتدا بر اساس کار یک تیم طراحی بود که اعضای آن عبارت بودند از :

Jeffrey D. Case
Keith McCloghrie
David T. Perkins
Randy Presuhn
Juergen Schoenwaelder

نسخه های قبلی این سند، ویرایش شده توسط کیت مک کلگری، نتیجه کار قابل توجه چهار مشارکت کننده اصلی بود:

Jeffrey D. Case
Keith McCloghrie
Marshall T. Rose
Steven Waldbusser

علاوه بر این، کمک های گروه کاری SNMPv۲ به نسخه های قبلی نیز تایید شده است. به ویژه، تشکر ویژه ای برای مشارکت های:

Alexander I. Alten
Dave Arneson
Uri Blumenthal
Doug Book
Kim Curran
Jim Galvin
Maria Greene
Iain Hanson
Dave Harrington
Nguyen Hien
Jeff Johnson
Michael Kornegay
Deirdre Kostick
David Levi
Daniel Mahoney
Bob Natale
Brian O’Keefe
Andrew Pearson
Dave Perkins
Randy Presuhn
Aleksey Romanov
Shawn Routhier
Jon Saperia
Juergen Schoenwaelder
Bob Stewart
Kaj Tesink
Glenn Waters
Bert Wijnen

۷-۱- ملاحظات امنیتی

پروتکل تعریف شده در این سند به خودی خود یک محیط امن فراهم نمی کند. حتی اگر خود شبکه ایمن باشد (به عنوان مثال با استفاده از IPSec)، هیچ کنترلی وجود ندارد که چه کسی در شبکه امن اجازه دسترسی به اطلاعات مدیریت را دارد.

توصیه می شود که پیاده سازها ویژگی های امنیتی ارائه شده توسط چارچوب SNMPv۳ را در نظر بگیرند. به طور خاص، استفاده از مدل امنیتی مبتنی بر کاربر ۶۲ STD، RFC۳۴۱۴ [RFC۳۴۱۴] و کنترل دسترسی مبتنی بر View، مدل ۶۲ STD، RFC۳۴۱۵ [RFC۳۴۱۵] توصیه می شود.

سپس این مسئولیت مشتری/کاربر است که اطمینان حاصل کند که موجودیت SNMP به درستی پیکربندی شده است به طوری که:

- فقط آن دسته از افراد اصلی (کاربرانی) که حقوق قانونی دارند می توانند به مقادیر هر یک از اشیاء MIB پشتیبانی شده توسط آن نهاد دسترسی یا تغییر دهند.

- وقوع رویدادهای خاص در واحد تجاری به طور مناسب اطلاع رسانی خواهد شد.

- واحد مورد رسیدگی به رویدادها و اطلاعاتی که به آن اطلاع داده شده است، به طور مناسب و با اعتبار مناسب پاسخ می دهد.