

4. ویژگی های پروتکل

4.1. ساختارهای مشترک

مقدار فیلد request-id در یک Response-PDU همان مقدار فیلد request-id Request-PDU ای که به آن پاسخ داده است، می باشد. با استفاده از مقدار request-id یک برنامه میتواند بین درخواست های انجام شده (عموما چندتا) تمایز دهد و در نتیجه پاسخ های ورودی را با درخواست هایی که به آنها پاسخ داده نشده ربط دهد. هرگاه یک سرویس دیتاگرام غیرقابل اعتماد استفاده شود request-id همچنین ابزار ساده ای برای تشخیص پیام های تکراری شبکه خواهد بود. استفاده از یک request-id یکسان در ارسال مجدد یک درخواست به پاسخ اجازه میدهد تا یا درخواست اصلی یا درخواست دوباره ارسال شده را پاسخ دهد. هر چند اگر بخواهیم RTT یک ارسال و پردازش تبادل یک درخواست-پاسخ را محاسبه کنیم برای هر ارسال مجدد یک درخواست باید از request-id متفاوتی استفاده کنیم. استراتژی بعدی برای اکثر موقعیت ها توصیه میشود.

مقدار غیر صفر در فیلد error-status در Response-PDU بیانگر وقوع خطا و جلوگیری از پردازش پاسخ است. در این مواقع یک مقدار غیر صفر در فیلد error-index در Response-PDU اطلاعات اضافه ای راجع به این که کدام variable binding در لیست باعث خطا شده است را میدهد. VarBind ها با شماره اندیشان شناسایی میشوند. اولین VarBind با اندیس 1 دومی با اندیس 2 و به همین ترتیب اندیس گذاری میشوند. در SNMP مقدار OBJECT IDENTIFIER حداکثر تا 128 زیر شناسه محدود میشود که هر زیر شناسه حداکثر مقدار $2^{32} - 1$ را میگیرد.

4.2. پردازش PDU

در المان های پروسه زیر، هر فیلد PDU که توسط پروسه مربوطه ارجاع داده نشده در SNMP دریافتی صرفنظر میشود. هرچند همه اجزای PDU حتی آن مقادیری که توسط SNMP دریافتی صرفنظر میشوند باید syntax و رمزگذاری معتبر ASN.1 داشته باشند. مثلا برای بعضی از PDU ها (برای مثال GetRequest-PDU) فقط نام یک متغیر و نه مقدار آن مهم است. در این مواقع از بخش مقدار متغیر در SNMP دریافتی صرفنظر میشود. مقدار "UnSpecified" برای استفاده در بخش مقدار چنین متغیر هایی تعریف میشود.

در تولید یک ارتباط مدیریتی پیام "wrapper" برای encapsulate کردن PDU بر اساس "المان های پروسه" ی چهارچوب اجرایی مورد استفاده، تولید میشود. تعریف "max-binding" یک حد بالا بر تعداد variable binding ها

اعمال میکند. در عمل سائز یک پیام توسط "maximum message size" هم محدود میشود. یک پیاده سازی مورد قبول تا جایی که موتور SNMP اجازه میدهد از variable binding در PDU یا BulkPDU پشتیبانی میکند تا به اندازه حداکثر طول پیام برسد ولی نه بیشتر از 2147483647 variable binding !

در دریافت یک ارتباط مدیریتی "المان های پروسه" ی چهارچوب اجرایی مورد استفاده دنبال میشود و اگر آن پروسه ها مشخص کنند که operation ای که درون پیام است باید به صورت محلی اجرا شود همان پروسه MIB-view ای که برای پروسه قابل مشاهده باشد را هم مشخص میکند.

4.2.1. GetRequest-PDU (مشابه RFC 1157)

GetRequest-PDU توسط یک موجودیت پروتکل و تنها برای درخواست موجودیت کاربردی SNMP خود تولید می شود. به محض دریافت GetRequest-PDU، موجودیت پروتکل دریافت کننده مطابق با هر قانون قابل اجرا در لیست زیر پاسخ می دهد:

(1) اگر برای هر شی نام گذاری شده در فیلد VarBind ها، نام شی نظیر نام شی در دسترس برای عملیات get در دید MIB مربوطه نباشد، آنگاه موجودیت دریافت کننده به مبدا پیام دریافت شده یک GetResponse-PDU با فرمی برابر به جز مقدار فیلد error-status که برابر است با noSuchName و مقدار فیلد error-index که ایندکس مولفه نام شی گفته شده در پیام دریافتی می باشد، ارسال خواهد کرد.

(2) اگر برای هر شی نام گذاری شده در فیلد VarBind ها، شی از نوع تجمعی است (مانند چیزی که در SMI تعریف شده)، آنگاه موجودیت دریافت کننده برای مبدا پیام دریافت شده یک GetResponse-PDU با شکل یکسان به جز مقدار فیلد error-status که noSuchName می باشد و همچنین مقدار فیلد error-index که برابر با ایندکس مولفه نام شی گفته شده در پیام دریافتی می باشد، ارسال می کند.

(3) اگر اندازه GetResponse-PDU تولید شده به عنوان چیزی که توصیف شده از یک حدودی فراتر رود، آنگاه موجودیت دریافت کننده برای مبدا پیام یک GetResponse-PDU به همان شکل ولی با مقدار فیلد error-status برابر با toobig، و مقدار فیلد error-index برابر با zero ارسال می کند.

(4) اگر برای هر شی نام گذاری شده در فیلد VarBind ها، مقدار شی را به دلایلی جز موارد بالا بازیابی نشود، آنگاه موجودیت دریافت کننده به مبدا پیام یک GetResponse-PDU به همان شکل ولی با مقدار فیلد error-status برابر با genErr و مقدار فیلد error-index برابر با ایندکس مولفه نام شی گفته شده در پیام دریافتی ارسال می کند.

اگر هیچ یک از قوانین بالا اعمال نشود، آنگاه موجودیت دریافت کننده پروتکل برای هر شی نام گذاری شده در فیلد VarBind های پیام دریافتی، یک GetResponse-PDU مثل همان به مبدا پیام ارسال می کند، مولفه منتظر GetResponse-PDU نام و مقدار آن متغیر را نشان می دهد. مقدار فیلد error-status برابر با noError و مقدار فیلد error-index برابر با zero می باشد. مقدار فیلد request-id نیز برابر با همان مقدار فیلد در پیام دریافتی خواهد بود.

4.2.2. GetNextRequest-PDU (مشابه RFC 1157)

فرم GetNextRequest-PDU مشابه GetRequest-PDU می باشد، به جز نشانه ای که برای نوع PDU است.

GetNextRequest-PDU توسط یک موجودیت پروتکل و فقط به درخواست موجودیت برنامه SNMP تولید می شود. پس از دریافت GetNextRequest-PDU، موجودیت پروتکل دریافت با توجه به هر قانون قابل اجرا در لیست زیر پاسخ می دهد:

(1) اگر به هر نام شی در فیلد متغیر-مقید (variable-bindings field)، که نام آن به ترتیب دیکشنری (قاموسی) (lexicographically) قبل از نام برخی از اشیاء موجود برای عملیات get در دید MIB مربوطه نباشد، پس موجودیت دریافت کننده به سازنده ی پیام دریافتی، پیامی مشابه GetResponse-PDU می فرستد، جز اینکه ارزش فیلد وضعیت خطا (error-status) noSuchName است، و ارزش فیلد شاخص خطا (error-index) شاخص مولفه نام شی ذکر شده در پیام دریافتی است.

(2) اگر اندازه ی GetResponse-PDU تولید شده به شرح زیر از محدودیت های محلی تجاوز کند، سپس موجودیت دریافت کننده به سازنده پیام دریافتی پیامی مشابه GetResponse-PDU می فرستد، جزاینکه ارزش فیلد وضعیت خطا (tooBig) است و ارزش فیلد شاخص خطا (خطا) صفر است.

(3) اگر برای هر نام شی در فیلد متغیر اتصالات (، ارزش نام شی بعدی در ترتیب قاموسی نمی تواند به دلایلی توسط هر یک از قوانین فوق را پوشش داده نمی بازیابی، سپس موجودیت دریافت کننده به سازنده پیام

دریافتی پیامی مشابه PDU-GetResponse می فرستد، جز اینکه ارزش فیلد وضعیت خطا (genErr) و

ارزش فیلد شاخص خطا شاخص مولفه نام شی ذکر شده در پیام دریافتی است.

اگر هیچ یک از قوانین فوق اعمال نشود، پس از آن موجودیت پروتکل دریافت کننده به سازنده پیام دریافتی پیام PDU-GetResponse را چنین می فرستد که، برای هر نام در فیلد متغیر-مقید (variable-bindings field) پیام دریافت شده، جزء (مولفه) متناظر در پیام PDU-GetResponse نام و مقدار آن شی ای را نشان می دهد که نامش در ترتیب قاموسی نام تمام اشیاء موجود برای انجام عملیات get از دید MIB مربوطه است، همراه با مقدار فیلد نام جزء داده شده، بلافاصله بعدی آن ارزش. ارزش فیلد وضعیت خطای (پیام PDU-GetResponse، noError است و ارزش فیلد شاخص خطا (errorindex) صفر است. ارزش فیلد شناسه درخواست (پیام PDU-GetResponse همان مقدار از پیام دریافتی است.

4.2.2.1. مثالی از پیمایش جدول

یک استفاده از PDU-GetNextRequest در پیمایش جداول مفهومی اطلاعات درون یک MIB است. مفهوم این نوع از درخواست به همراه روش شناسایی انفرادی نمونه های اشیای درون MIB، دسترسی به اشیای مرتبط درون MIB را تهیه میکند مثل اینکه آنها از آرایش جدولی لذت میبرند.

در تعویض پروتکل که در زیر شرح داده شده، یک برنامه آدرس فیزیکی وابسته به media و نوع نگاشت آدرس برای هر entry در جدول net-to-media برای یک المان شبکه مشخص را بازیابی میکند. همچنین مقدار sysUpTime که نگاشت در آن وجود دارد را برمیگرداند. فرض کنید که دستور پاسخ دهنده ی جدول net-to-media entry3 دارد:

Interface-Number	Network-Address	Physical-Address	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

SNMP entity که از برنامه تولید کننده دستورات پشتیبانی میکند با فرستادن یک GetNextRequest-PDU ای که مقدار OBJECT IDENTIFIER مشخص شده را به عنوان متغیر درخواست شده شامل میشود, شروع میکند:

```
GetNextRequest ( sysUpTime,  
ipNetToMediaPhysAddress,  
ipNetToMediaType )
```

SNMP entity که از برنامه پاسخ دهنده دستورات پشتیبانی میکند با یک Response-PDU پاسخ میدهد:

```
Response (( sysUpTime.0 = "123456" ),  
( ipNetToMediaPhysAddress.1.9.2.3.4 = "000010543210" ),  
( ipNetToMediaType.1.9.2.3.4 = "dynamic" ))
```

SNMP entity که از برنامه تولید کننده دستورات پشتیبانی میکند ادامه میدهد:

```
GetNextRequest ( sysUpTime,  
ipNetToMediaPhysAddress.1.9.2.3.4,  
ipNetToMediaType.1.9.2.3.4 )
```

SNMP entity که از برنامه پاسخ دهنده دستورات پشتیبانی میکند پاسخ میدهد:

```
Response (( sysUpTime.0 = "123461" ),  
( ipNetToMediaPhysAddress.1.10.0.0.51 = "000010012345" ),  
( ipNetToMediaType.1.10.0.0.51 = "static" ))
```

SNMP entity که از برنامه تولید کننده دستورات پشتیبانی میکند ادامه میدهد:

```
GetNextRequest ( sysUpTime,  
ipNetToMediaPhysAddress.1.10.0.0.51,  
ipNetToMediaType.1.10.0.0.51 )
```

SNMP entity که از برنامه پاسخ دهنده دستورات پشتیبانی میکند پاسخ میدهد:

```
Response (( sysUpTime.0 = "123466" ),  
( ipNetToMediaPhysAddress.2.10.0.0.15 = "000010987654" ),  
( ipNetToMediaType.2.10.0.0.15 = "dynamic" ))
```

SNMP entity که از برنامه تولید کننده دستورات پشتیبانی میکند ادامه میدهد:

```
GetNextRequest ( sysUpTime,  
ipNetToMediaPhysAddress.2.10.0.0.15,  
ipNetToMediaType.2.10.0.0.15 )
```

از آنجایی که هیچ entry دیگری در جدول نیست SNMP entity که از برنامه پاسخ دهنده به دستورات پشتیبانی میکند با متغیری که مقدار بعدی در ترتیب قاموسی نام اشیا قابل دسترس است پاسخ میدهد. مثلاً:

```
Response (( sysUpTime.0 = "123471" ),  
( ipNetToMediaNetAddress.1.9.2.3.4 = "9.2.3.4" ),  
( ipRoutingDiscards.0 = "2" ))
```

در نظر داشته باشید چگونه با رسیدن به انتهای ستون ipNetToMediaPhysAddress دومین variable binding از برنامه پاسخ دهنده به دستورات هم اکنون به اولین در ستون بعدی "wrapped" شده است. در نظر داشته باشید چگونه با رسیدن به انتهای ستون ipNetToMediaPhysAddress برای سومین variable binding برنامه پاسخ دهنده به دستوران با شیء در دسترس بعدی که بیرون از جدول است جواب داده است. این جواب نشان دهنده انتهای جدول برای برنامه تولید کننده دستورات است.

4.2.3. GetBulkRequest-PDU

یک GetBulkRequest-PDU به خاطر درخواست یک برنامه تولید و فرستاده میشود. هدف -GetBulkRequest PDU برای درخواست انتقال مقدار بالنسبه زیادی از داده شامل ولی نه محدود به کارایی و سرعت در بازیابی جداول بزرگ میشود.

در دریافت یک GetBulkRequest-PDU , SNMP entity گیرنده برای تولید یک Response-PDU هر variable binding در لیست variable binding را پردازش میکند و فیلد request-id را همان مقدار در دستور request میگذارد.

برای نوع GetBulkRequest-PDU پردازش موفق هر variable binding در request باعث تولید صفر یا بیشتر variable binding در Response-PDU میشود. بطور دقیق تر, نگاشت یک به یک بین variable binding های GetRequest-PDU, GetNextRequest-PDU, SetRequest-PDU و Response-PDU حاصل نگاشت بین variable binding های یک GetBulkRequest-PDU و Response-PDU حاصل را اجرا نمیکند. مقدار فیلدهای "non-repeaters" و "max-repetitions" در request پردازش درخواست شده را مشخص میکند. یک variable binding در Response-PDU برای اولین N variable binding در request درخواست شده و M variable binding برای هر R variable binding باقی مانده در request. در نتیجه تعداد variable binding هایی که توسط request فرستاده شده برابر با $N + (M * R)$ خواهد بود که N مینیمم :

- مقدار فیلد "non-repeaters" در request
- تعداد variable binding ها در request

و M مقدار فیلد "max-repetitions" در request و R ماکزیمم:

• (تعداد variable binding ها در request) - N

• 0

خواهند بود.

SNMP entity گیرنده یک Response-PDU با حداکثر به تعداد variable binding های درخواست شده که با request فرستاده شده اند را تولید میکند. Request-id باید همان مقدار GetBulkRequest-PDU دریافت شده را بگیرد.

اگر N از 0 بیشتر بود اولین تا N امین variable binding در Response-PDU به طریق زیر تولید میشوند:

1. متغیری که به ترتیب قاموسی نام همه متغیرها که با این درخواست در دسترس است و نامش اولین جانشین قاموسی در نام variable binding در GetBulkRequest-PDU, تعیین میشود. فیلد نام و مقدار variable binding متناظر در Response-PDU با نام و مقدار متغیر تعیین شده, قرار داده میشود.
2. اگر نام variable binding درخواست شده به لحاظ قاموسی قبل از نام هیچ متغیر در دسترس با این درخواست نبود, یعنی هیچ جانشین قاموسی نبود آنگاه variable binding متناظر تولید شده در Response-PDU مقدار فیلد value را مساوی "endOfMibView" قرار میدهد و فیلد نام را با نام Variable binding در درخواست مساوی قرار میدهد.

اگر M و R غیر صفر باشند, (N+1) امین variable binding و متغیرهای بعدی Response-PDU هرکدام به روش مشابهی تولید میشوند. برای هر تکرار i (i بزرگتر از 0 و کوچکتر مساوی M) و برای هر متغیر تکرار شده r (r بزرگتر از 0 و کوچکتر مساوی R), $(N + ((i-1) * R) + r)$ امین variable binding در Response-PDU به ترتیب زیر تولید میشود:

1. متغیری که در لیست ترتیب قاموسی نام های همه متغیرهای در دسترس با این درخواست که نامش i امین جانشین قاموسی (N + r) امین نام variable binding در GetBulkRequest-PDU دریافتی تعیین میشود و فیلد نام و مقدار variable binding با نام و مقدار متغیر تعیین شده برابر میشود.
2. اگر هیچ i امین جانشینی نبود variable binding متناظر تولید شده در Response-PDU فیلد مقدارش با "endOfMibView" برابر و فیلد نامش یا با آخرین جانشین یا اگر هیچ جانشینی نباشد با نام (N+r) امین variable binding در request برابر میشود.

با اینکه حداکثر تعداد variable binding ها در Response-PDU توسط $N + (M * R)$ محدود میشود ممکن است پاسخ به یکی از سه دلیل زیر تعداد کمتری variable binding (حتی 0 تا) تولید کند:

1. اگر ساین پیامی که قرار است در Response-PDU کپسول شود و حاوی تعداد درخواست شده variable binding است از یک محدودیت محلی یا "max message size" تولید کننده بیشتر باشد، در آن صورت پاسخ با تعداد کمتری از variable binding ها تولید میشود. این تعداد کمتر مجموعه مرتب شده variable binding ها با حذف برخی از آنها از انتهای مجموعه است به طوری که ساین پیام کپسول شده تقریباً برابر و نه بزرگتر از محدودیت محلی یا "max message size" تولید کننده باشد. توجه کنید که تعداد variable binding های حذف شده هیچ ربطی به مقادیر N,M,R ندارد.
 2. پاسخ ممکن است با تعداد کمتری از variable binding ها تولید شود اگر به ازای برخی مقادیر تکرار i بزرگتر از 0 و کوچکتر از M تمام variable binding های تولید شده مقدار "endOfMibView" را داشته باشند. در این مواقع ممکن است variable binding ها بعد از $(N+(i*R))$ امین variable binbing بریده شوند.
 3. در صورتی که پردازش یک درخواست با تعداد زیادی تکرار مقدار قابل توجهی زمان برای پردازش نسبت به یک درخواست معمول بگیرد در آن صورت ممکن است برنامه پاسخ دهنده به دستورات ممکن است به آن درخواست با تعداد کمتری از تکرارها خاتمه بدهد (با اتمام حداقل 1 تکرار).
- اگر پردازش یکی از variable binding ها به دلیل دیگری بجز موارد بالا با مشکل روبه رو شد، Response-PDU با مقدار مشابه request-id و فیلدهای variable binding که در GetBulkRequest-PDU, که مقدار error-status آن "genErr" و مقدار error-index آن اندیس متغیری که در پیام اصلی در variable binding دچار مشکل شد، دوباره قالب بندی میشود.
- در غیر این صورت مقدار فیلد error-status در Response-PDU "noError" و مقدار فیلد error-index 0 خواهد بود.
- Response-PDU تولید شده (احتمالاً با فیلد خالی variable binding) در یک پیام کپسول می شود. اگر ساین پیام حاصله کوچکتر یا مساوی هر دو محدودیت های محلی و "max message size" تولید کننده بود به تولید کننده GetBulkRequest-PDU ارسال میشود در غیر این صورت شمارنده "snmpSilentDrops" یکی زیاد شده و پیام حاصله درو انداخته میشود.

4.2.3.1. یک مثال دیگر از پیمایش جدول

این مثال نشان می‌دهد چگونه GetBulkRequest-PDU میتواند جانشینی برای GetNextRequest-PDU باشد. همان پیمایش جدول net-to-media که در بخش 4.2.2.1 نشان داده شد با تبدلات کمتر به دست آمده است. SNMP entity که از برنامه تولید کننده دستورات پشتیبانی میکند با ارسال یک GetBulkRequest-PDU با مقدار نسبتاً کم "max-repetition" = 2 و دربرداشتن مقدار OBJECT IDENTIFIER متغیر درخواست شده، شروع میکند

```
GetBulkRequest [ non-repeaters = 1, max-repetitions = 2 ]
( sysUpTime,
  ipNetToMediaPhysAddress,
  ipNetToMediaType )
```

SNMP entity که از برنامه پاسخ دهنده دستورات پشتیبانی میکند با یک Response-PDU پاسخ میدهد:

```
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" ))
```

SNMP entity که از برنامه تولید کننده دستورات پشتیبانی میکند ادامه میدهد:

```
GetBulkRequest [ non-repeaters = 1, max-repetitions = 2 ]
( sysUpTime,
  ipNetToMediaPhysAddress.1.10.0.0.51,
  ipNetToMediaType.1.10.0.0.51 )
```

SNMP entity که از برنامه پاسخ دهنده دستورات پشتیبانی میکند پاسخ میدهد:

```
Response (( sysUpTime.0 = "123466" ),
  ( 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" ))
```

توجه کنید چگونه همانند اولین مثال variable binding در پاسخ مشخص میکند به انتهای جدول رسیده ایم. چهارمین variable binding با برگرداندن اطلاعات از ستون در دسترس بعدی همین کار را می‌کند. پنجمین variable binding با برگرداندن اطلاعات در مورد اولین متغیر در دسترس که به ترتیب فاموسی در جدول قرار دارد، همین کار را می‌کند. این پاسخ انتهای جدول را برای برنامه تولید کننده دستورات نشان میدهد.