

TESTE DE GESTAO DE REDES

CURSO MEI

CARLOS MANUEL DA SILVA ALVES

PG41840

Data:9-01-2020

Questão 1

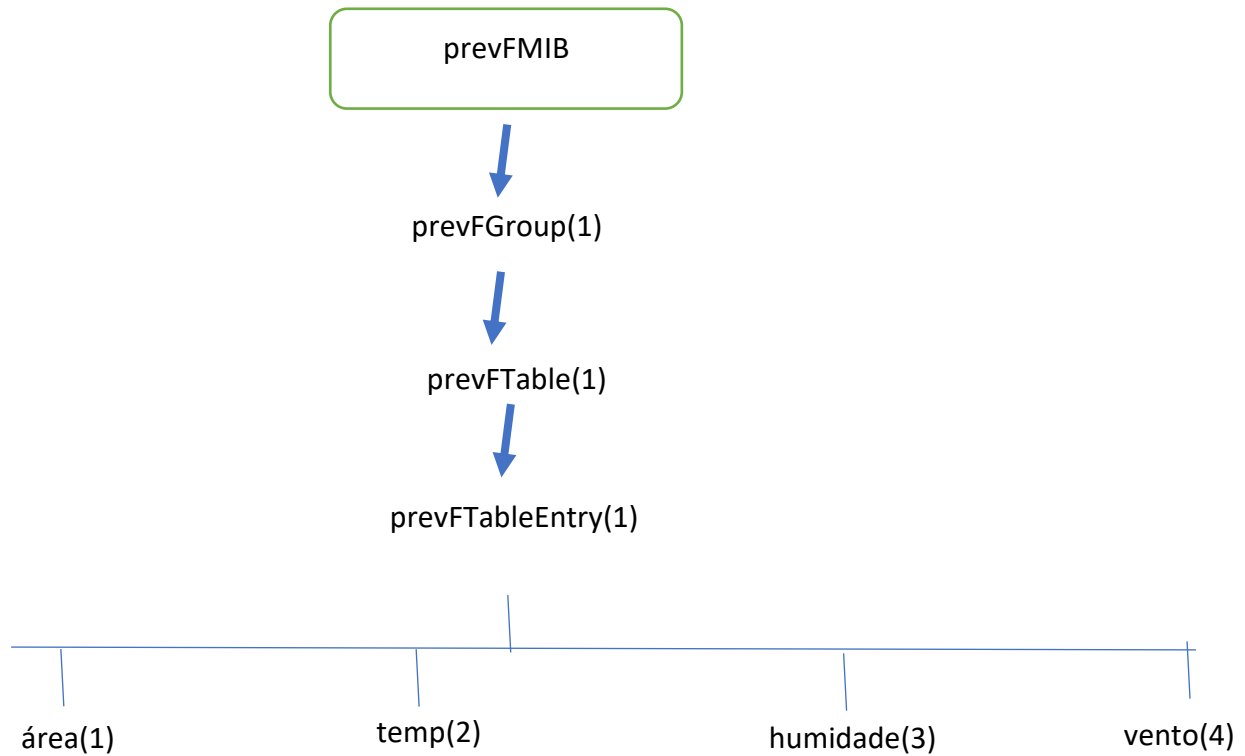
A arquitetura INMF/SNMP pode ser usada na gestão de redes de controladores, tendo máquinas gestoras a receberem informações de todos os agentes colocados na fábrica, onde cada agente é responsável pelo processamento dos dados emitidos por cada sensor instalado na respectiva zona em que se encontra o agente (1 agente por área). As informações coletadas ficariam armazenadas em bases de dados (MIB) que posteriormente seriam requeridas pelo gestor. Mas como numa fábrica seria necessário interligar as diferentes e diversas redes locais, pois esta pode estar a funcionar bem mas como não consegue comunicar com as outras, e claro não consegue ligar ao gestor. Então a solução baseava-se na implementação numa máquina, numa rede local, um protocolo de gerenciamento que permitisse ser gerenciada ou pelo menos ter acesso às informações de gerenciamento coletadas.

Em relação aos problemas na utilização deste tipo de arquitetura que pode vir apresentar, temos a falta de segurança, isto é, como se trata de um sistema de fabrico industrial, a perda de alguns pacotes (mensagem) ou até mesmo um agente que devido algum problema não esteja a comunicar corretamente com o manager, pode causar numerosos acidentes como prejuízos para a empresa em questão. Além disso, a sensorização e o tempo real podem causar falhas na veracidade dos dados captados. Estes são alguns dos problemas que podem ocorrer, isto sem especular muito.

Algo que poderia ser utilizado para contornar alguns problemas, seria optar pelo uso de TCP em algumas comunicações “sensíveis” em vez do UDP. A construção cuidadosa dos agentes snmp como das MIBs, visto que as mibs possui a característica de estruturas fixas.

Questão 2

ALINEA A



-- The prevF Group

prevFTable OBJECT-TYPE

SYNTAX SEQUENCE OF PrevFTableEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION "coiso"

::={prevFMIB 1}

prevFTableEntry OBJECT-TYPE

SYNTAX PrevFTableEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION "Tabela coiso"

::={prevFTable 1}

```
prevFTableEntry ::= SEQUENCE{  
    area    INTEGER,  
    temp    INTEGER,  
    humidade INTEGER,  
    vento    INTEGER  
}  
  
area            OBJET-TYPE  
SYNTAX          INTEGER  
ACCESS          read-only  
STATUS          mandatory  
DESCRIPTION     "Index da area dos valores coletados"  
::={prevFTableEntry 1}
```

```
temp            OBJET-TYPE  
SYNTAX          INTEGER  
ACCESS          read-only  
STATUS          current  
DESCRIPTION     "Valores Temperatura registada"  
::={prevFTableEntry 2}
```

```
humidade        OBJET-TYPE  
SYNTAX          INTEGER  
ACCESS          read-only  
STATUS          current  
DESCRIPTION     "Valores de Humidade registados"  
::={prevFTableEntry 3}
```

```
vento            OBJET-TYPE  
SYNTAX          INTEGER  
ACCESS          read-only
```

STATUS current

DESCRIPTION “Valores de vento registados”

::={prevFTableEntry 4}

prevFTableEntry

área(1)	temp(2)	humidade(3)	vento(4)
1	18	30	10
2	50	5	70
3	20	20	20

ALINEA B

ou

Snmppget -v 2c 192.168.1.1:161 scan temp.X humidade.x vento.x

Os valores obtidos serão individuais, desta forma:

Temp.x = 18

Humidade.x = 30

Vento.x = 10

ALINEA C

- 1) Fazer pull das variáveis;
- 2) Armazenar na BaseDados(MIB)
- 3) No Gestor percorrer a base de dados
- 4) Verifica os valores de Temperatura, Humidade e Vento.
- 5) Compara com os valores pré-definidos(Tmax, Hmax e Vmax)
- 6) Enquanto existe comunicação
 - a. Se os valores forem superiores aos pre-definidos, em simultâneo
 - i. Inicia Timer e compara o com o TempoMax
 - ii. Chamar Função ALARME (Função Alarme pode ser pedido aos bombeiros.)
 - b. Se não continua a verificar.
- 7) Verificar quando agente ficou inoperável
- 8) Se tempo for superior ao pré-definido
- 9) Emitir Alarme de Assistência

Em código seria algo do tipo

```
INT Tmax = x, Hmax = x, Vmax = x, TempoMax = y;
```

```
If(snmppget(temp.entry)>Tmax && snmppget(temp.entry>Hmax &&  
snmppget(temp.entry)>Vmax)){
```

```
    tempo = startTime();
```

```
    If(tempo>TempoMax)){
```

```
        FuncAlarm();
```

```
    }
```

```
    Else{
```

```
        Return False;
```

```
    }
```

```
}
```

```
FuncAlarme(){}
```