

## Nota de Aplicação NAP165

# Comunicação OPC UA com Controladores ALTUS

## Sumário

|            |                                                                                  |           |
|------------|----------------------------------------------------------------------------------|-----------|
| <b>1.</b>  | <b>Introdução .....</b>                                                          | <b>2</b>  |
| <b>2.</b>  | <b>Revisões de Software e Firmware Utilizadas .....</b>                          | <b>2</b>  |
| <b>3.</b>  | <b>Configurações do Servidor OPC UA no Controlador .....</b>                     | <b>2</b>  |
| <b>3.1</b> | <b>Configuração das Variáveis OPC UA .....</b>                                   | <b>2</b>  |
| 3.1.1      | Tipos de Variáveis Suportadas .....                                              | 3         |
| 3.1.2      | Quantidade Máxima de Variáveis .....                                             | 4         |
| <b>3.2</b> | <b>Configurações de Segurança Cibernética .....</b>                              | <b>5</b>  |
| 3.2.1      | Configuração de Criptografia nos Controladores Nexto .....                       | 5         |
| 3.2.2      | Exemplo de Configuração de Criptografia num Cliente OPC UA (UaExpert) .....      | 5         |
| 3.2.3      | Remoção de Criptografia nos Controladores Nexto .....                            | 6         |
| <b>4.</b>  | <b>Conceitos Básicos do Protocolo OPC UA.....</b>                                | <b>6</b>  |
| 4.1        | Subconjunto Implementado nos Controladores Nexto .....                           | 7         |
| 4.2        | Subscriptions (Variáveis Reportadas por Exceção) .....                           | 7         |
| 4.3        | Grupos de Variáveis Reportadas por Polling .....                                 | 8         |
| <b>5.</b>  | <b>Principais Parâmetros de Comunicação Configurados em Clientes OPC UA.....</b> | <b>8</b>  |
| 5.1        | Endpoint URL .....                                                               | 9         |
| 5.2        | Publishing Interval (ms) e Sampling Interval (ms) .....                          | 9         |
| 5.3        | Lifetime Count e Keep-Alive Count .....                                          | 12        |
| 5.4        | Queue Size e Discard Oldest .....                                                | 13        |
| 5.5        | Filter Type e Deadband Type .....                                                | 13        |
| 5.6        | PublishingEnabled, MaxNotificationsPerPublish e Priority .....                   | 14        |
| <b>6.</b>  | <b>Análise de Desempenho.....</b>                                                | <b>14</b> |
| 6.1        | Consumo de Banda de Comunicação .....                                            | 15        |
| 6.2        | Consumo de Processamento .....                                                   | 16        |
| 6.2.1      | Resultados dos Testes dos Cenários .....                                         | 17        |
| 6.2.2      | Conclusões Qualitativas sobre os Testes de Cenários .....                        | 19        |
| <b>7.</b>  | <b>Revisões .....</b>                                                            | <b>21</b> |

## 1. Introdução

---

Esta nota de aplicação aborda diversos aspectos relacionados à comunicação OPC UA com controladores ALTUS que suportam um servidor OPC UA, ou seja, controladores das séries Nexto.

Entre os aspectos descritos encontram-se os seguintes:

- Configurações do servidor OPC UA nos controladores ALTUS compatíveis;
- Configurações de segurança cibernética (criptografia);
- Principais parâmetros de configuração negociados com clientes OPC UA, e seus efeitos sobre o comportamento e desempenho da comunicação;
- Análise de desempenho da comunicação OPC UA, considerando o consumo de CPU e consumo de banda de comunicação Ethernet. Os objetivos são recomendar técnicas de configuração que otimizem o desempenho, e prever a viabilidade de aplicações em conformidade com o desempenho esperado.

## 2. Revisões de Software e Firmware Utilizadas

---

Para elaboração desta nota de aplicação foram executados testes com os seguintes controladores, programador e cliente OPC UA, utilizando as versões de software e firmware indicadas a seguir:

- Controlador NX3030: V1.7.0.4
- Controlador XP325: V1.7.3.0
- Controlador NX3003: V1.7.2.0
- Programador Mastertool IEC XE (MT8500): V3.11
- Cliente OPC UA UaExpert: V1.5.0

## 3. Configurações do Servidor OPC UA no Controlador

---

Esta seção trata das configurações OPC UA nos controladores Nexto, usando o Mastertool.

### 3.1 Configuração das Variáveis OPC UA

---

As variáveis que serão disponibilizadas pelo servidor OPC UA são um subconjunto de variáveis definidas em GVLs ou POU's do projeto criado através do programador Mastertool.

Depois de criar as GVLs e POU's com estas variáveis, deve-se executar os seguintes passos no programador Mastertool:

1. Adicionar o objeto *Symbol Configuration* abaixo do tópico *Application* na árvore de dispositivos;
2. No objeto *Symbol Configuration*, na aba *Configurações*, marcar o checkbox *Suporte a características OPC UA*;
3. Compilar o projeto sem que haja erros reportados;

4. Selecionar as GVLs e POUs que contém variáveis OPC UA, marcando os correspondentes checkboxes no objeto *Symbol Configuration*;
5. Expandindo os checkboxes de GVLs ou POUs específicas é possível:
  - a. Selecionar um subconjunto das variáveis declaradas dentro destas GVLs ou POUs para comunicação OPC UA;
  - b. Configurar direitos de acesso para cada variável (read-write, read-only, write-only). O default é read-write.

A figura seguinte mostra um exemplo em que todas as variáveis da GVL\_STRING foram disponibilizadas, e em que parte das variáveis da POU UserPrg foram disponibilizadas (somente a variável StartTime não foi disponibilizada nesta POU).

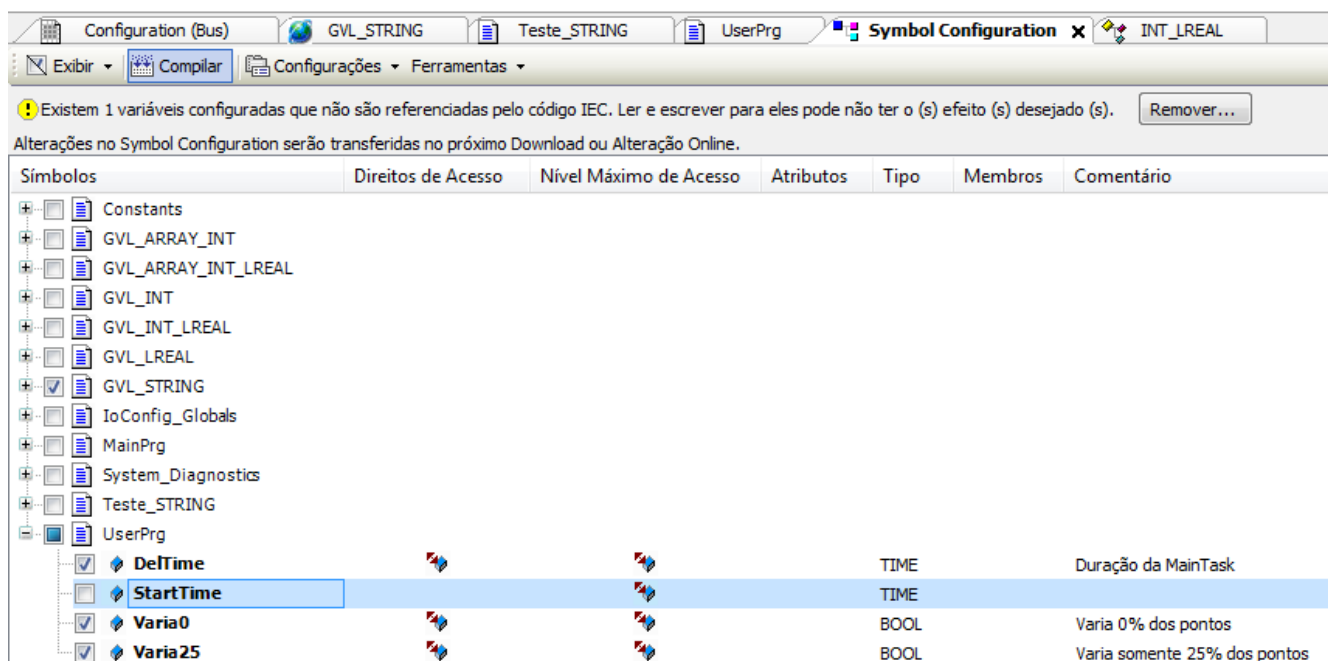


Figura 3-1. Configuração das variáveis disponibilizadas pelo servidor OPC UA

### 3.1.1 Tipos de Variáveis Suportadas

A comunicação OPC UA pode envolver variáveis simples ou estruturadas (STRUCTs ou function blocks). Além disso, também é possível definir arrays de variáveis simples ou estruturadas.

A tabela seguinte mostra os tipos básicos de variáveis simples, e os respectivos tamanhos destas variáveis (em bytes) alocados na transmissão das mesmas no frame OPC UA. Esta informação de tamanho é importante para estimar o consumo de banda de comunicação Ethernet (ver seção 6.1).

| Tipo         | Tamanho (bytes) |
|--------------|-----------------|
| BIT          | Nota 1          |
| BOOL         | 1               |
| SINT         | 1               |
| USINT / BYTE | 1               |
| INT          | 2               |

|               |        |
|---------------|--------|
| UINT / WORD   | 2      |
| DINT          | 4      |
| UDINT / DWORD | 4      |
| LINT          | 8      |
| ULINT / LWORD | 8      |
| REAL          | 4      |
| LREAL         | 8      |
| STRING        | Nota 2 |
| TIME          | 4      |
| LTIME         | 8      |

**Tabela 3-2. Tipos de variáveis simples suportadas**

**Nota 1:**

O tipo BIT na verdade não pode ser declarado como tipo simples, mas pode ser declarado dentro de uma STRUCT. Na memória do controlador este tipo aloca 1 bit, mas no frame OPC UA ele aloca 1 byte.

**Nota 2:**

O tipo STRING aloca N bytes na memória do controlador se for declarado com N posições. Entretanto, no frame OPC UA o tamanho alocado varia entre 4 e N+4 bytes dependendo do valor atualmente atribuído ao string (mínimo de 4 quando o string estiver vazio). Os 4 bytes adicionais são utilizados para indicar o tamanho atual do string.

O tamanho de uma variável estruturada é a soma dos tamanhos das variáveis que a compõem. Por exemplo, considere o tipo INT\_REAL definido a seguir:

```
TYPE INT_LREAL :  
STRUCT  
    VI : INT;  
    VL : LREAL;  
END_STRUCT  
END_TYPE
```

O tamanho de uma variável estruturada do tipo INT\_REAL será 10 bytes (2 bytes do INT + 8 bytes do LREAL).

Considere um array do tipo INT\_REAL, por exemplo:

```
ARRAY_INT_REAL : ARRAY [1 ... 100] OF INT_REAL;
```

Este array, como um todo, alocará 1000 bytes (100 posições \* 10 bytes por posição).

### 3.1.2 Quantidade Máxima de Variáveis

Até 5000 variáveis podem ser disponibilizadas via servidor OPC UA. O compilador do Mastertool indicará erro se mais do que 5000 variáveis forem disponibilizadas no servidor OPC UA no objeto *Symbol Configuration*.

É importante salientar como tais variáveis são contadas:

- Cada variável simples conta como uma variável (exemplo: tipos BOOL, INT, REAL, STRING, etc.);
- Cada variável estruturada é contada como N variáveis, onde N é o número de variáveis que compõem a estrutura. Considerando o exemplo anterior da estrutura INT\_REAL (ver seção 3.1.1), cada variável deste tipo estruturado conta como 2 variáveis;
- Cada elemento de array conta como uma variável para tipos simples, ou como N variáveis para tipos estruturados. Por exemplo:
  - Considerando um array com 100 elementos do tipo INT, este array conta como 100 variáveis;
  - Considerando um array com 100 elementos do tipo estruturado INT\_REAL, este array conta como 200 variáveis.

## 3.2 Configurações de Segurança Cibernética

---

Se desejado, o usuário pode configurar criptografia para a comunicação OPC UA usando o profile Basic256SHA256. A seguir, definem-se os passos para configurar esta funcionalidade no servidor, e também um exemplo das configurações no cliente UaExpert.

### 3.2.1 Configuração de Criptografia nos Controladores Nexto

---

Para configurar a criptografia num servidor OPC UA deve-se criar um certificado para o mesmo, executando os seguintes passos no programador Mastertool:

1. Definir um caminho ativo (*active path*) para comunicação com o controlador (não é necessário fazer login);
2. No menu Visualizar, selecionar *Tela de Segurança*;
3. Clicar na aba *Devices* no lado esquerdo desta tela;
4. Clicar no ícone  para executar um refresh;
5. Clicar no ícone *Device*, abaixo do qual se abrirão diversas pastas de certificados (Own Certificates, Trusted Certificates, Untrusted Certificates, Quarantined Certificates);
6. Clicar no ícone  para gerar um certificado e selecione os seguintes parâmetros:
  - a. Key length (bit): 3072
  - b. Validity period (days): 365 (pode ser modificado se desejado)
7. Aguarde enquanto o certificado é calculado e transferido para o controlador (isso pode levar alguns minutos);
8. Reinicialize (desligue e religue) o controlador.

### 3.2.2 Exemplo de Configuração de Criptografia num Cliente OPC UA (UaExpert)

---

Depois de gerar um certificado para criptografia num controlador Nexto, é necessário executar um procedimento para validar o certificado de um cliente que deseja se conectar ao controlador.

O procedimento seguinte exemplifica o que deve ser feito no cliente UaExpert (para outros clientes, consultar o manual dos mesmos):

1. Adicione uma conexão para o controlador, e selecione o tipo de conexão “Basic256Sha256 - Sign & Encrypt (uatcp-uasc-uabinary)”;
2. Selecione o menu *Server/Connect*. Uma tela de erro se abrirá. Marque o checkbox “Accept the server certificate temporarily for this session”, e depois pressione o botão “Continue”;
3. Volte ao programador Mastertool (com caminho ativo definido corretamente para o controlador, sem necessidade de fazer login) e clique no ícone  da *Tela de Segurança* para executar um refresh;
4. Na *Tela de Segurança*, selecione a pasta “Quarantined Certificates” abaixo do *Device*. No painel direito deve-se observar um certificado solicitado pelo cliente UaExpert;
5. Arraste este certificado para a pasta “Trusted Certificates”;
6. Volte para o cliente UaExpert e selecione o menu *Server/Connect*. Uma tela de erro se abrirá. Marque o checkbox “Accept the server certificate temporarily for this session”, e depois pressione o botão “Continue”. Caso apareça uma message box de “Connect Error”, aperte o botão “Ignore”;
7. Depois disso, abre-se um “Adress Space” onde é possível navegar e selecionar as variáveis da subscription. A partir deste momento, tudo funciona de maneira idêntica a conexões não criptografadas.

### 3.2.3 Remoção de Criptografia nos Controladores Nexto

Caso o servidor possua criptografia configurada e deseja-se removê-la, deve-se seguir o seguinte procedimento:

1. Definir um caminho ativo (*active path*) para comunicação com o controlador (não é necessário fazer login);
2. No menu *Visualizar*, selecionar *Tela de Segurança*;
3. Clicar na aba *Devices* no lado esquerdo desta tela;
4. Clicar no ícone  para executar um refresh;
5. Clicar no *Device*, abaixo do qual se abrem diversas pastas de certificados (Own Certificates, Trusted Certificates, Unstruted Certificates, Quarantined Certificates);
6. Clicar na pasta “Own Certificates”, e no painel direito selecionar o certificado (OPC UA Server);
7. Clicar no ícone  para remover este certificado do projeto e do controlador;
8. Reinicialize (desligue e religue) o controlador.

## 4. Conceitos Básicos do Protocolo OPC UA

Antes de prosseguir, é importante ter o conhecimento de alguns conceitos básicos sobre o protocolo OPC UA.

## 4.1 Subconjunto Implementado nos Controladores Nexto

A especificação OPC UA é extremamente extensa, abrangendo diversas funcionalidades, que tipicamente não são encontradas em um mesmo servidor ou cliente.

O servidor implementado nos controladores da série Nexto possui um subconjunto das funcionalidades disponíveis na especificação:

- Codificação de dados: UA binary (ver seção 3.1.1 que define os dados suportados);
- Nível de transporte: protocolo UA TCP, usando a porta TCP 4840;
- Conexão segura criptografada (opcional) com profile Basic256SHA256;
- Capacidade de “browse” das variáveis (importação on-line das variáveis para o cliente). Alguns clientes também suportam configuração off-line das variáveis (ex: arquivo CSV, etc), de forma que a etapa de “browse” é desnecessária;
- Suporta múltiplos clientes conectados simultaneamente. No entanto, é importante salientar que o desempenho da comunicação é reduzido à medida que mais clientes são conectados (ver seção 6);
- As variáveis podem ser reportadas por exceção (somente ao modificar seu valor) ou por polling (ciclicamente). O método de polling não é recomendado devido ao alto consumo de banda de comunicação e processamento), e não será abordado nesta nota de aplicação (alguns clientes nem suportam o método de polling);
- Possibilidade de agrupar as variáveis reportadas por exceção de um mesmo cliente em diferentes subscriptions (ver seção 4.2) para obter melhor desempenho;

## 4.2 Subscriptions (Variáveis Reportadas por Exceção)

Ao configurar um cliente OPC UA, é possível agrupar subconjuntos das variáveis disponibilizadas por um servidor OPC UA em diferentes subscriptions, para as variáveis reportadas por exceção.

Observar que as variáveis contidas dentro de cada subscription são definidas no cliente, ou seja, diferentes clientes conectados a um mesmo servidor podem ter diferentes configurações de subscriptions.

Uma subscription é usada para enviar notificações para o cliente. Notificações trafegam em pacotes *PublishResponse* do protocolo OPC UA, que transportam apenas as variáveis da subscription que sofreram modificações. Desta forma, economiza-se banda de comunicação.

Para visualizar os diversos pacotes OPC UA, tais como *PublishRequest* e *PublishResponse* e outros, pode-se utilizar o analisador de protocolo Wireshark com filtro “opcua”.

Sobre as notificações de uma subscription transmitidas em pacotes *PublishResponse*, é importante ressaltar:

- O intervalo mínimo entre dois pacotes *PublishResponse* é limitado por um parâmetro (Publishing Interval) definido na seção 5.2. Desta forma, é possível limitar a banda de comunicação;
- Cada pacote *PublishResponse* transmite apenas as variáveis da subscription cujos valores foram modificados, o que também promove economia da banda de comunicação;

- Caso nenhuma variável da subscription tenha se modificado, o intervalo entre pacotes PublishResponse será ainda maior que o parâmetro Publishing Interval, o que também promove economia da banda de comunicação.

Ao configurar a comunicação num cliente, a opção mais simples é colocar todas as variáveis do servidor numa única subscription. No entanto, alocar as variáveis em múltiplas subscriptions pode ter vantagens significativas do ponto de vista de desempenho (banda de comunicação e consumo de CPU no servidor).

Uma sugestão de alocação de variáveis em diferentes subscriptions poderia ser, por exemplo:

- Subscription 1: variáveis digitais que precisam ser reportadas rapidamente quando se modificam (configurar baixo valor de Publishing Interval). Resulta numa baixa banda de comunicação caso poucas destas variáveis se modifiquem frequentemente.
- Subscription 2: variáveis digitais que podem ser reportadas mais lentamente quando se modificam (configurar valor maior de Publishing Interval). Resulta numa baixa banda de comunicação por ter um valor maior de Publishing Interval.
- Subscription 3: variáveis analógicas que se modificam frequentemente, mas que podem ser reportadas mais lentamente (configurar valor maior de Publishing Interval). Resulta numa baixa banda de comunicação por ter um valor maior de Publishing Interval. Técnica de banda morta (deadband) podem ser aplicadas para impedir que pequenas modificações sejam transmitidas (ver seção 5.4 sobre deadbands).
- Subscription 4: parâmetros da aplicação escritos pelo sistema SCADA (ex: setpoints, níveis de alarme) que raramente são modificados. Resulta numa baixa banda de comunicação porque raramente se modificam.

### 4.3 Grupos de Variáveis Reportadas por Polling

Quando as variáveis são reportadas pelo método de polling, elas são lidas ciclicamente através de pacotes ReadRequest e ReadResponse. Reportar pelo método de polling não é recomendado, pois demanda uma banda de comunicação Ethernet muito maior.

#### Nota 1:

Este documento não analisa a configuração e o desempenho do método de polling, pois não existe aparente motivação para utilizar este método.

#### Nota 2:

Alguns clientes podem usar pacotes ReadRequest e ReadResponse apenas para testar a conexão com o servidor (não para ler variáveis). Se você visualizar tais pacotes num log do Wireshark, perceba que não se trata de polling. Estes pacotes são pequenos e transmitidos com intervalos relativamente altos, sem causar impacto relevante no consumo de banda Ethernet.

## 5. Principais Parâmetros de Comunicação Configurados em Clientes OPC UA

A configuração do servidor OPC UA no Mastertool é simples, conforme descrito na seção 3.

Além disso, existem alguns parâmetros de comunicação que são configurados no cliente OPC UA e negociados com o servidor por ocasião do estabelecimento da conexão. O entendimento e configuração



correta destes parâmetros são importantes para que o desempenho da comunicação OPC UA atinja objetivos tais como os seguintes:

1. Limitar o retardo entre a modificação de uma variável no controlador e sua notificação para o cliente OPC UA;
2. Limitar a banda de comunicação necessária na rede Ethernet;
3. Adequar-se à capacidade de processamento do controlador Nexto;

O protocolo OPC UA pode consumir grandes quantidades de banda de comunicação e consumir um percentual relativamente alto da capacidade de processamento do controlador ALTUS.

A tarefa de comunicação OPC UA executa no controlador com prioridade menor que tarefas essenciais como a MainTask. Se houver grande quantidade de variáveis que devem ser reportadas frequentemente, e o consumo de CPU pela MainTask for muito alto, então é possível que a comunicação OPC UA tenha desempenho insatisfatório. Outras tarefas que executam com a mesma prioridade que o servidor OPC UA, tais como protocolos de comunicação MODBUS, também podem reduzir o desempenho da comunicação OPC UA.

Uma configuração inadequada de parâmetros no cliente OPC UA também pode reduzir o desempenho da comunicação OPC UA. A seção 6 contém uma análise de desempenho que mostra a influência de alguns destes parâmetros.

As subseções seguintes descrevem alguns dos parâmetros que tem maior impacto no desempenho, e que deveriam ser configuráveis em qualquer cliente OPC UA.

**Nota:**

Alguns clientes OPC UA poderão dar nomes diferentes para estes parâmetros. Os nomes empregados neste documento estão de acordo com a norma OPC UA.

## 5.1 Endpoint URL

Deve-se informar o endereço IP e porta TCP do servidor da seguinte maneira:

`opc.tcp://192.168.17.2:4840`

Neste exemplo, o endereço IP do controlador é 192.168.17.2.

A porta TCP sempre deve ser 4840.

## 5.2 Publishing Interval (ms) e Sampling Interval (ms)

O parâmetro Publishing Interval deve ser definido para cada subscription e define o intervalo mínimo entre pacotes PublishResponse usados para reportar notificações de mudanças de variáveis configuradas na subscription.

O parâmetro Sampling Interval deve ser definido para cada variável (monitored item) configurada numa subscription, de acordo com a especificação OPC UA. No entanto, muitos clientes só permitirão configurar um único valor de Sampling Interval comum para todas as variáveis configuradas numa mesma subscription. Este parâmetro define o período utilizado no controlador (servidor) para detectar mudanças na variável associada.

O retardo máximo entre uma modificação de variável no servidor e sua transmissão num pacote PublishResponse é igual à soma dos parâmetros Sampling Interval (atraso máximo para detectar

mudança) e Publishing Interval (atraso máximo para transmitir a mudança depois de detectá-la). Obviamente este é o pior caso. O retardo médio é a metade deste valor.

O parâmetro Publishing Interval permite limitar a banda de comunicação da rede Ethernet, pois quanto maior for seu valor, menos banda será consumida.

Quanto ao parâmetro Sampling Interval, é importante não colocar valores muito pequenos no mesmo, pois a tarefa de detecção de modificações consome um elevado processamento do controlador. Se o valor for muito pequeno, a comunicação OPC UA poderá ficar muito lenta.

#### Nota:

Caso o servidor não suporte os valores configurados pelo cliente para Publishing Interval e Sampling Interval, ele poderá revisar estes valores para um valor suportado imediatamente superior, e informar isso para o cliente. Alguns clientes exibem os valores revisados pelo servidor e outros não. Uma análise de logs via Wireshark nos pacotes CreateSubscriptionResponse e CreateMonitoredItemsResponse permite observar os valores revisados destes parâmetros pelo servidor.

O seguinte log do Wireshark mostra a comunicação de uma subscription cujas variáveis estão mudando mais rapidamente que o Sampling Interval, e onde Publishing Interval = 1000 ms e Sampling Interval = 500 ms.

| No.  | Time      | Source         | Destination    | Protocol | Length | Info                                            |
|------|-----------|----------------|----------------|----------|--------|-------------------------------------------------|
| 379  | *REF*     | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 453  | 1.005771  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 456  | 1.006074  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 522  | 2.005969  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 131    | UA Secure Conversation Message: ReadRequest     |
| 523  | 2.007767  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 526  | 2.008056  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 528  | 2.009590  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 128    | UA Secure Conversation Message: ReadResponse    |
| 586  | 3.009714  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 589  | 3.010059  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 650  | 4.012693  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 653  | 4.012951  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 711  | 5.013936  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 714  | 5.014124  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 789  | 6.015854  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 792  | 6.016156  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 855  | 7.016051  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 131    | UA Secure Conversation Message: ReadRequest     |
| 856  | 7.017164  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 859  | 7.017377  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 860  | 7.019567  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 128    | UA Secure Conversation Message: ReadResponse    |
| 942  | 8.021881  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 945  | 8.022198  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 1002 | 9.024634  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 1005 | 9.024837  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 1108 | 10.027561 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 1111 | 10.027810 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 1198 | 11.031552 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 1201 | 11.031891 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |

**Figura 5-1. Notificações com variáveis mudando mais rapidamente que o Sampling Interval**

No pacote 379 ( $t = 0$  s), ocorre um PublishRequest com o qual o cliente solicita que o servidor transmita uma PublishResponse notificando as variáveis da subscription cujos valores foram modificados. O momento em que o servidor transmitirá o pacote PublishResponse depende de alguns fatores:

- Do parâmetro Publishing Interval, que define o intervalo mínimo entre dois pacotes PublishResponse consecutivos da mesma subscription;

- De ter ocorrido modificação de valor em pelo menos uma variável da subscription. Caso nenhuma variável da subscription tenha se modificado, o intervalo entre dois pacotes PublishResponse consecutivos da mesma subscription será maior que o Publishing Interval;
- Do parâmetro Keep-Alive Count, que será descrito na seção 5.3. Ele define um intervalo máximo entre pacotes PublishResponse, quando nenhuma variável da subscription muda seu valor por um longo tempo.

No pacote 453 ( $t = 1$  s) o PublishResponse é transmitido, 1 segundo depois do PublishRequest que o solicitou. Isto ocorre porque um PublishResponse anterior havia sido transmitido poucos milissegundos antes do PublishRequest do pacote 379 (não mostrado na figura), e é necessário manter um intervalo mínimo de 1000 ms (Publishing Interval) entre pacotes PublishResponse consecutivos.

Observa-se que 1 ms depois do PublishResponse no pacote 453, no pacote 456, um novo PublishRequest foi transmitido, solicitando um novo PublishResponse, que só poderá chegar 1 segundo mais tarde.

O seguinte log do Wireshark mostra a comunicação de uma subscription cujas variáveis estão mudando mais lentamente que o Sampling Interval, e onde Publishing Interval = 1000 ms e Sampling Interval = 500 ms. Neste caso, as variáveis estão mudando com intervalo de 3 segundos.

| No.  | Time      | Source         | Destination    | Protocol | Length | Info                                            |
|------|-----------|----------------|----------------|----------|--------|-------------------------------------------------|
| 28   | *REF*     | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 176  | 3.005423  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 179  | 3.005776  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 305  | 5.005510  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 131    | UA Secure Conversation Message: ReadRequest     |
| 307  | 5.008039  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 128    | UA Secure Conversation Message: ReadResponse    |
| 354  | 6.012090  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 357  | 6.012400  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 493  | 9.018028  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 496  | 9.018362  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 563  | 10.017564 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 131    | UA Secure Conversation Message: ReadRequest     |
| 567  | 10.019962 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 128    | UA Secure Conversation Message: ReadResponse    |
| 691  | 12.025974 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 694  | 12.026318 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 859  | 15.025676 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 131    | UA Secure Conversation Message: ReadRequest     |
| 861  | 15.027941 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 128    | UA Secure Conversation Message: ReadResponse    |
| 862  | 15.032934 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 866  | 15.033260 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 985  | 18.039701 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 988  | 18.040040 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 1076 | 20.039713 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 131    | UA Secure Conversation Message: ReadRequest     |
| 1079 | 20.042696 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 128    | UA Secure Conversation Message: ReadResponse    |
| 1126 | 21.044040 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 1129 | 21.044309 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 1243 | 24.048793 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 404    | UA Secure Conversation Message: PublishResponse |
| 1246 | 24.049129 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 109    | UA Secure Conversation Message: PublishRequest  |
| 1285 | 25.048777 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 131    | UA Secure Conversation Message: ReadRequest     |
| 1287 | 25.051086 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 128    | UA Secure Conversation Message: ReadResponse    |

**Figura 5-2. Notificações com variáveis mudando a cada 3 segundos**

Observa-se neste log que o intervalo entre as PublishResponses é de 3 segundos, conforme o ritmo de modificação das variáveis da subscription.

Ou seja, quando poucas variáveis da subscription estão se modificando, o consumo de banda de comunicação é reduzido.

Além disso, o tamanho de cada pacote PublishResponse depende da quantidade de variáveis da subscription que efetivamente se modificaram. Desta forma, o consumo de banda é reduzido quando menos variáveis da subscription se modificam. No exemplo anterior isto não é visível (todos os pacotes

PublishResponse tem o mesmo tamanho) pois todas as variáveis da subscription estão sendo modificadas no teste realizado.

Os parâmetros Publishing Interval e Sampling Interval de cada subscription devem ser configurados para atender os seguintes objetivos:

- Limitar os retardos máximo e médio entre a modificação de uma variável da subscription e sua transmissão num pacote PublishReponse:
  - Retardo Máximo = Publishing Interval + Sampling Interval
  - Retardo Médio = Retardo Máximo / 2
- Limitar o consumo de banda de comunicação Ethernet (ver seção 6.1);
- Limitar o consumo de processamento requerido da CPU do controlador (ver seção 6.2).

### **5.3 Lifetime Count e Keep-Alive Count**

---

Estes dois parâmetros devem ser configurados para cada subscription.

O objetivo destes parâmetros é criar um mecanismo de desativação de uma subscription por iniciativa do servidor, caso o cliente parar de transmitir pacotes PublishRequest por muito tempo para esta subscription.

Caso o servidor não receba pacotes PublishRequest por um tempo superior a Lifetime Count multiplicado por Publishing Interval, o servidor desativa a subscription, que deverá ser criada novamente pelo cliente no futuro se assim o desejar.

Em situações onde as variáveis de uma subscription não se modificam, poderia passar um longo tempo sem a transmissão de PublishResponses e consequentemente de PublishRequests que o sucedem, provocando uma desativação indesejada da subscription. Para evitar que isso aconteça, foi criado o parâmetro Keep-Alive Count. Caso não haja modificações de dados na subscription por um tempo igual a Keep-Alive Count multiplicado por Publishing Interval, o servidor enviará um pacote PublishResponse pequeno e vazio indicando que nenhuma variável mudou. Este PublishResponse vazio autorizará que o cliente mande imediatamente o próximo PublishRequest.

O valor de Keep-Alive Count deve ser menor do que o valor de Lifetime Count para evitar uma desativação indesejada da subscription. Sugere-se que LifeTime Count seja no mínimo 3 vezes maior que Keep-Alive Count.

O seguinte log do Wireshark mostra um caso em que LifeTime Count = 60, Keep-Alive Count = 5 e Publishing Interval = 1000 ms, e onde nenhuma variável da subscription está sendo modificada.

| (opcua) |           |                |                |          |        |                                                 |
|---------|-----------|----------------|----------------|----------|--------|-------------------------------------------------|
| No.     | Time      | Source         | Destination    | Protocol | Length | Info                                            |
| 5123    | *REF*     | 192.168.16.125 | 192.168.17.2   | OpcUa    | 101    | UA Secure Conversation Message: PublishRequest  |
| 5308    | 2.999760  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 131    | UA Secure Conversation Message: ReadRequest     |
| 5311    | 3.001969  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 128    | UA Secure Conversation Message: ReadResponse    |
| 5457    | 5.007576  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 143    | UA Secure Conversation Message: PublishResponse |
| 5460    | 5.007863  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 101    | UA Secure Conversation Message: PublishRequest  |
| 5630    | 8.007854  | 192.168.16.125 | 192.168.17.2   | OpcUa    | 131    | UA Secure Conversation Message: ReadRequest     |
| 5632    | 8.009980  | 192.168.17.2   | 192.168.16.125 | OpcUa    | 128    | UA Secure Conversation Message: ReadResponse    |
| 5740    | 10.026448 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 139    | UA Secure Conversation Message: PublishResponse |
| 5743    | 10.026729 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 101    | UA Secure Conversation Message: PublishRequest  |
| 5937    | 13.025915 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 131    | UA Secure Conversation Message: ReadRequest     |
| 5939    | 13.028735 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 128    | UA Secure Conversation Message: ReadResponse    |
| 6060    | 15.048236 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 139    | UA Secure Conversation Message: PublishResponse |
| 6063    | 15.048517 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 101    | UA Secure Conversation Message: PublishRequest  |
| 6232    | 18.047977 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 131    | UA Secure Conversation Message: ReadRequest     |
| 6234    | 18.050649 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 128    | UA Secure Conversation Message: ReadResponse    |
| 6343    | 20.059995 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 139    | UA Secure Conversation Message: PublishResponse |
| 6346    | 20.060269 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 101    | UA Secure Conversation Message: PublishRequest  |
| 6520    | 23.060040 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 131    | UA Secure Conversation Message: ReadRequest     |
| 6522    | 23.062141 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 128    | UA Secure Conversation Message: ReadResponse    |
| 6654    | 25.075913 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 139    | UA Secure Conversation Message: PublishResponse |
| 6657    | 25.076194 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 101    | UA Secure Conversation Message: PublishRequest  |
| 6817    | 28.076122 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 131    | UA Secure Conversation Message: ReadRequest     |
| 6819    | 28.078346 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 128    | UA Secure Conversation Message: ReadResponse    |
| 6951    | 30.090858 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 139    | UA Secure Conversation Message: PublishResponse |
| 6954    | 30.091135 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 101    | UA Secure Conversation Message: PublishRequest  |
| 7137    | 33.091190 | 192.168.16.125 | 192.168.17.2   | OpcUa    | 131    | UA Secure Conversation Message: ReadRequest     |
| 7139    | 33.094458 | 192.168.17.2   | 192.168.16.125 | OpcUa    | 128    | UA Secure Conversation Message: ReadResponse    |

**Figura 5-3. Pacotes PublishRequest e PublishResponse com variáveis não mudando**

Observa-se que existe um tempo de 5 segundos ( $\text{Keep-Alive Count} = 5 * \text{Publishing Interval} = 1000$  ms) entre um PublishRequest e o próximo PublishResponse. Este PublishResponse é um pacote pequeno, pois não reporta nenhuma variável. Poucos milissegundos depois de um PublishResponse, um novo PublishRequest é transmitido pelo cliente.

## 5.4 Queue Size e Discard Oldest

Estes parâmetros devem ser mantidos com os seguintes valores fixos, que normalmente são os valores default nos clientes:

- Queue Size: 1
- Discard Oldest: enable

De acordo com a norma OPC UA, é possível definir estes parâmetros para cada variável. No entanto, muitos clientes apenas permitem definir valores comuns para todas as variáveis configuradas numa subscription.

Queue Size deve ser mantido com o valor 1 pois não existe suporte a eventos nesta implementação do servidor OPC UA e, portanto, é desnecessário criar filas de eventos. Aumentar o valor de Queue Size pode implicar em aumento na banda de comunicação e processamento da CPU, portanto isso deve ser evitado.

Discard Oldest deve ser mantido com o valor “enable”, para que o pacote PublishResponse sempre reporte a mudança de valor mais recente detectada para cada variável.

## 5.5 Filter Type e Deadband Type

Estes parâmetros devem ser mantidos com os seguintes valores fixos, que normalmente são os valores default nos clientes:

- Filter Type: DataChangeFilter
- Deadband Type: none

De acordo com a norma OPC UA, é possível definir estes parâmetros para cada variável. No entanto, muitos clientes apenas permitem definir valores comuns para todas as variáveis configuradas numa subscription.

O parâmetro Filter Type deve valer DataChangeFilter, indicando que mudanças de valores nas variáveis devem provocar sua transmissão num pacote PublishResponse.

Deadband Type deve ser mantido em “none” porque não existe implementação de deadbands para variáveis analógicas ou numéricas. Desta forma, qualquer alteração da variável analógica, por mínima que seja, provoca sua transmissão num pacote PublishResponse.

Para reduzir consumo de processamento e banda de comunicação Ethernet, o usuário poderá implantar deadbands por sua conta da seguinte forma:

- Não incluir a variável analógica na subscription;
- Ao invés disso, incluir na subscription uma variável auxiliar vinculada à variável analógica;
- Copiar a variável analógica para a variável auxiliar somente quando o deadband gerenciado pelo usuário for extrapolado.

## **5.6 PublishingEnabled, MaxNotificationsPerPublish e Priority**

Sugere-se que os seguintes parâmetros sejam mantidos com os seguintes valores, que normalmente são os valores default nos clientes:

- PublishingEnabled: true
- MaxNotificationsPerPublish: 0
- Priority: 0

Estes parâmetros devem ser configurados para cada subscription.

PublishingEnable deve valer “true” para que as variáveis da subscription sejam reportadas em caso de mudança de valor.

MaxNotificationsPerPublish indica quantas das variáveis que mudaram de valor podem ser incluídas num mesmo pacote PublishResponse. O valor especial “0” indica que não existe um limite para isso, e recomenda-se utilizar este valor para que todas as variáveis que mudaram sejam reportadas num mesmo pacote PublishResponse.

Priority indica a prioridade relativa desta subscription em relação a outras. Caso em determinado momento o servidor deva enviar múltiplos pacotes PublishResponse de subscriptions diferentes, priorizará aquele com o maior valor de priority. Se todas subscriptions tiverem a mesma prioridade, os pacotes PublishResponse serão transmitidos numa sequência fixa.

## **6. Análise de Desempenho**

Nesta seção é analisado o desempenho da comunicação OPC UA sob os seguintes pontos de vista:

- Consumo de banda de comunicação Ethernet, considerando diversos fatores de influência;

- Consumo de processamento necessário para o atendimento aos requisitos de retardo da comunicação especificados através dos parâmetros Publishing Interval e Sampling Interval, considerando diversos fatores de influência.

## 6.1 Consumo de Banda de Comunicação

Para estimar a banda de comunicação consumida por uma subscription, diversos fatores devem ser considerados:

- O número e tipo das variáveis incluídas na subscription, a partir do qual se pode calcular o número máximo de bytes transmitidos associados a estas variáveis;
- O parâmetro Publishing Interval;
- Uma estimativa da percentagem média das variáveis que são transmitidas em cada Publishing Interval.

Esta seção descreve como calcular a banda de comunicação em regime, depois da fase de inicialização, e sem a presença de falhas de comunicação.

Considerar os seguintes parâmetros para o cálculo:

- N: número de variáveis configuradas na subscription (a seção 3.1.2 descreve como contar estas variáveis);
- TM: tamanho médio das variáveis em bytes. Por exemplo, se a subscription tiver 3000 variáveis INT (3000 \* 2 bytes) e 2000 variáveis LREAL (2000 \* 8 bytes),  $TM = (3000 * 2 + 2000 * 8) / (2000 + 3000) = 4.4$  bytes;
- PI: parâmetro Publishing Interval (milissegundos) configurado para a subscription;
- PMV: percentual médio do número de variáveis da subscription que serão transmitidas em cada Publishing Interval. No pior caso, assumir que 100% das variáveis estão se modificando mais rápido que o Sampling Interval, e que o Sampling Interval é menor ou igual ao Publishing Interval.

Cada variável com tamanho T bytes (ex: 8 bytes para um LREAL) é transmitida usando 22 + T bytes (ex: 30 bytes para um LREAL), pois além do valor, existem outras informações transmitidas (timestamps, client handle, tipo, etc.).

Desta forma, a banda de comunicação em kbits/segundo será de no mínimo:

$$\text{Banda Calculada (kbps)} = (8 * N * (TM + 22) * PMV / (PI * 100))$$

Por exemplo, para 5000 variáveis LREAL variando continuamente (PMV = 100%) e com Publish Interval de 1000 ms:

$$\text{Banda Calculada (kbps)} = (8 * 5000 * (8 + 22) * 100 / (1000 * 100)) = 1.200 \text{ kbps}$$

Além disso, existem alguns overheads. Mediu-se no Wireshark os seguintes tráfegos durante um minuto:

- Upstream (cliente para servidor): com o filtro “tcp.port = 4840 and ip.dst == <endereço IP servidor>”
- Downstream (servidor para cliente): com o filtro “tcp.port = 4840 and ip.src == <endereço IP servidor>”

A tabela seguinte resume o resultado de alguns testes com 1000, 2000, 3000, 4000 e 5000 variáveis do tipo LREAL, sendo que 100% destas variáveis estão sendo modificadas mais rapidamente que o Sampling Interval que é menor que o Publishing Interval:

| N (LREAL) | TM (bytes) | PMV (%) | PI (ms) | Banda calculada (kbps) | Banda real upstream (kbps) | Banda real downstream (kbps) |
|-----------|------------|---------|---------|------------------------|----------------------------|------------------------------|
| 1000      | 8          | 100,00% | 1000    | 240                    | 6                          | 253                          |
| 2000      | 8          | 100,00% | 1000    | 480                    | 10                         | 500                          |
| 3000      | 8          | 100,00% | 1000    | 720                    | 14                         | 753                          |
| 4000      | 8          | 100,00% | 1000    | 960                    | 19                         | 1007                         |
| 5000      | 8          | 100,00% | 1000    | 1200                   | 24                         | 1258                         |

**Tabela 6-1. Bandas de comunicação medidas com Wireshark**

É possível estimar as bandas reais pelas seguintes equações:

- **Banda Real Upstream = 0.03 \* Banda Calculada**
- **Banda Real Downstream = 1.06 \* Banda Calculada**

Nota:

Utilizando criptografia não existem diferenças relevantes na banda de comunicação consumida. As mesmas equações podem ser utilizadas.

## 6.2 Consumo de Processamento

A comunicação OPC UA precisa de um percentual mínimo da capacidade de processamento da CPU do controlador para atingir os objetivos de performance, tais como máximo retardo de atualização de variáveis no cliente.

Nesta seção, mostram-se os resultados de testes que foram executados para diversos cenários, que determinaram o percentual mínimo de processamento necessário para cada um destes cenários. Cada cenário é caracterizado por diversos fatores (modelo de CPU, número de clientes, quantidade e tipo de variáveis da subscription, e outros fatores que serão descritos nesta seção).

Deve-se observar que a tarefa de comunicação OPC UA executa com prioridade menor que a tarefa MainTask, e compete com outras tarefas de mesma prioridade (comunicações MODBUS, diagnósticos, etc).

Para executar cada cenário de teste, utilizou-se uma ferramenta especial para medir o consumo de processamento do servidor OPC UA, e além disso confirmou-se através de um analisador de protocolo Ethernet (Wireshark) que os pacotes PublishResponse estavam sendo transmitidos com a frequência configurada no parâmetro Publishing Interval.

Os seguintes fatores de influência sobre os testes foram utilizados para definir cenários:

1. Modelo de CPU, visto que diferentes modelos têm capacidades de processamento diferentes. Foram utilizados os modelos NX3030, NX3003 e XP325. Os modelos NX3010, NX3020, NX3004 e NX3005 não foram testados porque tem capacidade de processamento similar ao NX3030;
2. O uso ou não de criptografia (segurança cibernética);
3. Quantidades e tipos de variáveis incluídas na subscription;
4. Percentual das variáveis da subscription que estão se modificando com ciclo menor que o Sampling Interval;
5. Publishing Interval;



6. Sampling Interval;
7. Número de clientes OPC UA conectados;
8. Tempo de ciclo da MainTask;
9. Número de subscriptions.

Além de apresentar os resultados para diversos cenários, esta seção também apresenta algumas conclusões qualitativas sobre o desempenho, relacionadas com os fatores de influência listados anteriormente.

### 6.2.1 Resultados dos Testes dos Cenários

A tabela seguinte mostra os resultados dos testes de cenários. Alguns testes (T47 até T52) ocupam duas linhas na tabela, pois envolvem duas subscriptions.

| Teste | Fatores de Influência sobre os Testes |              |                                  |                     |                          |                        |                            |                     | Resultados              |
|-------|---------------------------------------|--------------|----------------------------------|---------------------|--------------------------|------------------------|----------------------------|---------------------|-------------------------|
|       | Modelo de CPU                         | Criptografia | Variáveis OPC UA na subscription | % Variáveis mudando | Publishing Interval (ms) | Sampling Interval (ms) | Clientes OPC UA conectados | Ciclo MainTask (ms) | Consumo CPU para OPC UA |
| T01   | NX3030                                | Não          | 1000 INT                         | 0%                  | 1000                     | 100                    | 1                          | 100                 | 51,0%                   |
| T02   | NX3030                                | Não          | 1000 INT                         | 0%                  | 1000                     | 300                    | 1                          | 100                 | 22,8%                   |
| T03   | NX3030                                | Não          | 1000 INT                         | 0%                  | 1000                     | 500                    | 1                          | 100                 | 16,8%                   |
| T04   | NX3030                                | Não          | 1000 INT                         | 0%                  | 1000                     | 1000                   | 1                          | 100                 | 12,2%                   |
| T05   | NX3030                                | Não          | 1000 INT                         | 0%                  | 1000                     | 1000                   | 1                          | 10                  | 12,2%                   |
| T06   | NX3030                                | Não          | 500 STRUCT(INT,LREAL)            | 0%                  | 1000                     | 1000                   | 1                          | 100                 | 13,5%                   |
| T07   | NX3030                                | Não          | 1000 INT                         | 100%                | 1000                     | 100                    | 1                          | 100                 | 63,0%                   |
| T08   | NX3030                                | Não          | 1000 INT                         | 100%                | 1000                     | 300                    | 1                          | 100                 | 31,0%                   |
| T09   | NX3030                                | Não          | 1000 INT                         | 100%                | 1000                     | 500                    | 1                          | 100                 | 22,8%                   |
| T10   | NX3030                                | Não          | 1000 INT                         | 100%                | 1000                     | 1000                   | 1                          | 100                 | 16,3%                   |
| T11   | NX3030                                | Não          | 1000 INT                         | 100%                | 1000                     | 1000                   | 1                          | 10                  | 15,5%                   |
| T12   | NX3030                                | Não          | 500 STRUCT(INT,LREAL)            | 100%                | 1000                     | 1000                   | 1                          | 100                 | 16,0%                   |
| T13   | NX3030                                | Não          | 1000 INT                         | 0%                  | 1000                     | 100                    | 2                          | 100                 | 67,5%                   |
| T14   | NX3030                                | Não          | 1000 INT                         | 0%                  | 1000                     | 300                    | 2                          | 100                 | 41,0%                   |
| T15   | NX3030                                | Não          | 1000 INT                         | 0%                  | 1000                     | 500                    | 2                          | 100                 | 30,0%                   |
| T16   | NX3030                                | Não          | 1000 INT                         | 0%                  | 1000                     | 1000                   | 2                          | 100                 | 21,0%                   |
| T17   | NX3030                                | Não          | 1000 INT                         | 0%                  | 1000                     | 1000                   | 2                          | 10                  | 21,0%                   |
| T18   | NX3030                                | Não          | 500 STRUCT(INT,LREAL)            | 0%                  | 1000                     | 1000                   | 2                          | 100                 | 22,0%                   |
| T19   | NX3030                                | Não          | 1000 INT                         | 100%                | 1000                     | 100                    | 2                          | 100                 | 70,0%                   |
| T20   | NX3030                                | Não          | 1000 INT                         | 100%                | 1000                     | 300                    | 2                          | 100                 | 54,0%                   |
| T21   | NX3030                                | Não          | 1000 INT                         | 100%                | 1000                     | 500                    | 2                          | 100                 | 38,5%                   |
| T22   | NX3030                                | Não          | 1000 INT                         | 100%                | 1000                     | 1000                   | 2                          | 100                 | 26,5%                   |
| T23   | NX3030                                | Não          | 1000 INT                         | 100%                | 1000                     | 1000                   | 2                          | 10                  | 27,0%                   |
| T24   | NX3030                                | Não          | 500 STRUCT(INT,LREAL)            | 100%                | 1000                     | 1000                   | 2                          | 100                 | 27,5%                   |
| T25   | NX3030                                | Não          | 2000 LREAL                       | 0%                  | 1000                     | 1000                   | 1                          | 100                 | 22,0%                   |
| T26   | NX3030                                | Não          | 2000 LREAL                       | 100%                | 1000                     | 1000                   | 1                          | 100                 | 27,7%                   |
| T27   | NX3030                                | Não          | 2000 LREAL                       | 0%                  | 1000                     | 1000                   | 2                          | 100                 | 37,2%                   |
| T28   | NX3030                                | Não          | 2000 LREAL                       | 100%                | 1000                     | 1000                   | 2                          | 100                 | 48,0%                   |
| T29   | NX3030                                | Não          | 2000 INT                         | 0%                  | 1000                     | 1000                   | 1                          | 100                 | 21,0%                   |
| T30   | NX3030                                | Não          | 2000 INT                         | 100%                | 1000                     | 1000                   | 1                          | 100                 | 27,5%                   |
| T31   | NX3030                                | Não          | 2000 INT                         | 0%                  | 1000                     | 1000                   | 2                          | 100                 | 34,3%                   |
| T32   | NX3030                                | Não          | 2000 INT                         | 100%                | 1000                     | 1000                   | 2                          | 100                 | 46,4%                   |
| T33   | NX3030                                | Não          | 2000 INT                         | 100%                | 2000                     | 1000                   | 1                          | 100                 | 26,4%                   |
| T34   | NX3030                                | Não          | 2000 INT                         | 100%                | 4000                     | 1000                   | 1                          | 100                 | 25,5%                   |

| Teste | Fatores de Influência sobre os Testes |               |                                  |                     |                          |                        |                            |                     | Resultados              |
|-------|---------------------------------------|---------------|----------------------------------|---------------------|--------------------------|------------------------|----------------------------|---------------------|-------------------------|
|       | Modelo de CPU                         | Cripto-grafia | Variáveis OPC UA na subscription | % Variáveis mudando | Publishing Interval (ms) | Sampling Interval (ms) | Clientes OPC UA conectados | Ciclo MainTask (ms) | Consumo CPU para OPC UA |
| T35   | NX3030                                | Não           | 2000 INT                         | 100%                | 2000                     | 1000                   | 2                          | 100                 | 44,5%                   |
| T36   | NX3030                                | Não           | 2000 INT                         | 100%                | 4000                     | 1000                   | 2                          | 100                 | 43,5%                   |
| T37   | NX3030                                | Não           | 5000 INT                         | 0%                  | 1000                     | 1000                   | 1                          | 100                 | 43,0%                   |
| T38   | NX3030                                | Não           | 5000 INT                         | 50%                 | 1000                     | 1000                   | 1                          | 100                 | 49,6%                   |
| T39   | NX3030                                | Não           | 5000 INT                         | 100%                | 1000                     | 1000                   | 1                          | 100                 | 55,0%                   |
| T40   | NX3030                                | Não           | 5000 INT                         | 100%                | 2000                     | 1000                   | 1                          | 100                 | 51,4%                   |
| T41   | NX3030                                | Não           | 2500 INT                         | 0%                  | 1000                     | 1000                   | 1                          | 100                 | 24,0%                   |
| T42   | NX3030                                | Não           | 2500 INT                         | 100%                | 1000                     | 1000                   | 1                          | 100                 | 32,5%                   |
| T43   | NX3030                                | Não           | 2500 INT                         | 0%                  | 1000                     | 1000                   | 2                          | 100                 | 39,7%                   |
| T44   | NX3030                                | Não           | 2500 INT                         | 100%                | 1000                     | 1000                   | 2                          | 100                 | 54,1%                   |
| T45   | NX3030                                | Não           | 1 INT                            | X                   | 1000                     | 1000                   | 1                          | 100                 | 4,7%                    |
| T46   | NX3030                                | Não           | 1 INT                            | X                   | 1000                     | 1000                   | 2                          | 100                 | 4,8%                    |
| T47   | NX3030                                | Não           | 1000 INT                         | 0%                  | 1000                     | 1000                   | 1                          | 100                 | 21,6%                   |
|       |                                       |               | 1000 LREAL                       |                     | 1000                     | 1000                   |                            |                     |                         |
| T48   | NX3030                                | Não           | 1000 INT                         | 100%                | 1000                     | 1000                   | 1                          | 100                 | 28,0%                   |
|       |                                       |               | 1000 LREAL                       |                     | 1000                     | 1000                   |                            |                     |                         |
| T49   | NX3030                                | Não           | 1000 INT                         | 0%                  | 1000                     | 500                    | 1                          | 100                 | 32,0%                   |
|       |                                       |               | 1000 LREAL                       |                     | 1000                     | 500                    |                            |                     |                         |
| T50   | NX3030                                | Não           | 1000 INT                         | 100%                | 1000                     | 500                    | 1                          | 100                 | 42,0%                   |
|       |                                       |               | 1000 LREAL                       |                     | 1000                     | 500                    |                            |                     |                         |
| T51   | NX3030                                | Não           | 1000 INT                         | 0%                  | 1000                     | 500                    | 1                          | 100                 | 27,1%                   |
|       |                                       |               | 1000 LREAL                       |                     | 2000                     | 1000                   |                            |                     |                         |
| T52   | NX3030                                | Não           | 1000 INT                         | 100%                | 1000                     | 500                    | 1                          | 100                 | 34,5%                   |
|       |                                       |               | 1000 LREAL                       |                     | 2000                     | 1000                   |                            |                     |                         |
| T53   | XP325                                 | Não           | 5000 INT                         | 0%                  | 1000                     | 1000                   | 1                          | 100                 | 44,6%                   |
| T54   | XP325                                 | Não           | 5000 INT                         | 100%                | 1000                     | 1000                   | 1                          | 100                 | 58,2%                   |
| T55   | XP325                                 | Não           | 2500 INT                         | 0%                  | 1000                     | 1000                   | 1                          | 100                 | 28,0%                   |
| T56   | XP325                                 | Não           | 2500 INT                         | 100%                | 1000                     | 1000                   | 1                          | 100                 | 37,0%                   |
| T57   | XP325                                 | Não           | 2500 INT                         | 0%                  | 1000                     | 1000                   | 2                          | 100                 | 45,2%                   |
| T58   | XP325                                 | Não           | 2500 INT                         | 100%                | 1000                     | 1000                   | 2                          | 100                 | 58,3%                   |
| T59   | XP325                                 | Não           | 1 INT                            | X                   | 1000                     | 1000                   | 1                          | 100                 | 6,0%                    |
| T60   | XP325                                 | Não           | 1 INT                            | X                   | 1000                     | 1000                   | 2                          | 100                 | 6,1%                    |
| T61   | NX3003                                | Não           | 5000 INT                         | 0%                  | 1000                     | 1000                   | 1                          | 100                 | não ok                  |
| T62   | NX3003                                | Não           | 5000 INT                         | 100%                | 1000                     | 1000                   | 1                          | 100                 | não ok                  |

| Teste | Fatores de Influência sobre os Testes |              |                                  |                     |                          |                        |                            |                     | Resultados              |
|-------|---------------------------------------|--------------|----------------------------------|---------------------|--------------------------|------------------------|----------------------------|---------------------|-------------------------|
|       | Modelo de CPU                         | Criptografia | Variáveis OPC UA na subscription | % Variáveis mudando | Publishing Interval (ms) | Sampling Interval (ms) | Clientes OPC UA conectados | Ciclo MainTask (ms) | Consumo CPU para OPC UA |
| T63   | NX3003                                | Não          | 2500 INT                         | 0%                  | 1000                     | 1000                   | 1                          | 100                 | 41,2%                   |
| T64   | NX3003                                | Não          | 2500 INT                         | 100%                | 1000                     | 1000                   | 1                          | 100                 | 45,8%                   |
| T65   | NX3003                                | Não          | 2500 INT                         | 0%                  | 1000                     | 1000                   | 2                          | 100                 | 54,4%                   |
| T66   | NX3003                                | Não          | 2500 INT                         | 100%                | 1000                     | 1000                   | 2                          | 100                 | 58,4%                   |
| T67   | NX3003                                | Não          | 1 INT                            | X                   | 1000                     | 1000                   | 1                          | 100                 | 19,9%                   |
| T68   | NX3003                                | Não          | 1 INT                            | X                   | 1000                     | 1000                   | 2                          | 100                 | 20,6%                   |
| T69   | NX3030                                | Sim          | 2000 INT                         | 0%                  | 1000                     | 1000                   | 1                          | 100                 | 20,5%                   |
| T70   | NX3030                                | Sim          | 2000 INT                         | 100%                | 1000                     | 1000                   | 1                          | 100                 | 28,2%                   |
| T71   | NX3030                                | Sim          | 2000 INT                         | 0%                  | 1000                     | 1000                   | 2                          | 100                 | 34,0%                   |
| T72   | NX3030                                | Sim          | 2000 INT                         | 100%                | 1000                     | 1000                   | 2                          | 100                 | 49,1%                   |
| T73   | NX3030                                | Sim          | 5000 INT                         | 0%                  | 1000                     | 1000                   | 1                          | 100                 | 39,9%                   |
| T74   | NX3030                                | Sim          | 5000 INT                         | 100%                | 1000                     | 1000                   | 1                          | 100                 | 57,8%                   |
| T75   | NX3030                                | Sim          | 2500 INT                         | 0%                  | 1000                     | 1000                   | 1                          | 100                 | 24,3%                   |
| T76   | NX3030                                | Sim          | 2500 INT                         | 100%                | 1000                     | 1000                   | 1                          | 100                 | 34,0%                   |
| T77   | NX3030                                | Sim          | 2500 INT                         | 0%                  | 1000                     | 1000                   | 2                          | 100                 | 40,5%                   |
| T78   | NX3030                                | Sim          | 2500 INT                         | 100%                | 1000                     | 1000                   | 2                          | 100                 | 57,0%                   |
| T79   | NX3030                                | Sim          | 1 INT                            | X                   | 1000                     | 1000                   | 1                          | 100                 | 5,0%                    |
| T80   | NX3030                                | Sim          | 1 INT                            | X                   | 1000                     | 1000                   | 2                          | 100                 | 5,0%                    |

**Tabela 6-2. Resultados dos Testes de Cenários de Consumo de Processamento**

## 6.2.2 Conclusões Qualitativas sobre os Testes de Cenários

Nesta seção, os resultados dos testes registrados na tabela anterior são utilizados para a obtenção de conclusões qualitativas relacionadas à capacidade de processamento demandada pela tarefa do servidor OPC UA. Os testes são referenciados pelos seus identificadores (T01, T02, etc) que aparecem na primeira coluna da tabela.

### Influência do Modelo de CPU

Alguns modelos de CPU são mais rápidos do que outros e, portanto, consomem menos processamento. As CPUs foram divididas em três grupos com desempenho similar:

- NX3030: NX3030, NX3010, NX3020, NX3004, NX3005
- XP325: XP325, XP315, XP300
- NX3003

Considerando testes com 2500 variáveis do tipo INT e outros fatores de influência idênticos, a seguinte comparação pode ser feita, desde o grupo mais rápido (NX3030) até o mais lento (NX3003):

- NX3030 – Teste T42: 33.8 %
- XP325 – Teste T56: 37.0%
- NX3003 – Teste T64: 45.8%

### Influência da Criptografia (Segurança Cibernética)

Comparando alguns testes, como T42 x T76, e T44 x T78, observa-se um pequeno acréscimo de processamento devido ao uso de criptografia:

- 2500 INT, 1 cliente: T42 – sem criptografia: 32.5% x T76 – com criptografia: 34.0%
- 2500 INT, 2 clientes: T44 – sem criptografia: 54.1% x T786 – com criptografia: 57.0%

### **Influência dos Tipos de Variáveis**

Um fator que tem influência muito pequena ou desprezível são os tipos das variáveis, bem como o fato de serem estruturas, arrays ou arrays de estruturas. O que realmente importa é a contagem de variáveis, conforme descrito na seção 3.1.2.

Por exemplo, os seguintes testes onde a contagem foi 1000 variáveis, e os demais fatores de influência são idênticos, reportaram resultados muito similares:

- T10: 1000 INT: 16.3%
- T12: 500 STRUCTS(INT, LREAL): 16.0%

### **Influência do Sampling Interval**

Este parâmetro tem alto impacto no consumo de processamento. Um Sampling Interval baixo demanda alta capacidade de processamento.

Comparando os testes T07, T08, T09 e T10, apenas o Sampling Interval mudou, e observa-se que o processamento mínimo demandado variou de 63% com Sampling Interval de 100 ms até 16.3% com Sampling Interval de 1000 ms.

### **Influência do Publishing Interval**

Este parâmetro tem impacto significativo no consumo de processamento. Um Publishing Interval baixo demanda maior capacidade de processamento.

Isso fica demonstrado pelos testes T30, T33 e T34, onde apenas o Publishing Interval mudou, e observa-se que o processamento mínimo demandado variou de 27.5% com Publishing Interval de 1000 ms até 25.5% com Publishing Interval de 4000 ms.

### **Influência do Percentual de Variáveis se Modificando**

Variáveis que se modificam mais rapidamente que o Sampling Interval consomem processamento relativo à inserção num fila de transmissão (em cada Sampling Interval) e à sua transmissão no pacote PublishResponse (em cada Publishing Interval).

Por exemplo, os testes T37, T38 e T39 (para 5000 INTs) demonstram um aumento de 43.0% para 55.0% de consumo de processamento, quando o percentual de variáveis que se modificam mais rapidamente que o Sampling Interval muda de 0% para 100% dos 5000 INTs.

### **Influência do Número de Clientes OPC UA Conectados**

Nestes testes todos os clientes tem a mesma configuração de subscriptions.

Descontando uma quantidade básica de processamento comum a todos os clientes, o resto do processamento é praticamente proporcional ao número de clientes conectados.

Por exemplo, o teste T45 (com apenas 1 variável INT) consome 4.7%, o que pode ser considerado a quantidade básica de processamento comum a todos os clientes.

Os testes T30 (1 cliente: 27.5%) e T32 (2 clientes: 46.4%) só diferem quanto ao número de clientes. Observa-se que:

$(27.5\% - 4.7\%)$  é aproximadamente a metade de  $(46.4\% - 4.7\%)$ .

### **Influência do Ciclo da MainTask**

O tempo de ciclo da MainTask tem influência muito pequena ou desprezível nos resultados.

Isso fica demonstrado comparando os testes T10 (ciclo de 100 ms: 16.3%) e T11 (ciclo de 10 ms: 15.5%), que possuem resultados similares.

## Influência do Número de Subscriptions

Dividir as variáveis em múltiplas subscriptions, mantendo iguais os demais fatores de influência, tem influência muito pequena ou desprezível nos resultados. Isso é demonstrado comparando os testes T30 (2000 variáveis na mesma subscription: 27.5%) e T48 (duas subscriptions cada uma com 1000 variáveis: 28%), que apresentam resultados semelhantes.

Entretanto, é vantajoso dividir as variáveis em múltiplas subscriptions para que seja possível selecionar valores diferentes de Publishing Interval e Sampling Interval para cada subscription. Comparando os testes T50 e T52, observa-se que foi possível reduzir o consumo de processamento de 42.0% para 34.5% aumentando o Publishing Interval e Sampling Interval de somente uma das duas subscriptions. Além de reduzir o consumo de processamento, esta estratégia também permite reduzir a banda de comunicação Ethernet.

## 7. Revisões

---

A revisão deste documento é mostrada na margem superior, indicando alterações no conteúdo ou melhorias no formato.

O histórico a seguir apresenta observações correspondentes a cada revisão:

**Revisão:** A  
**Data:** 30/11/2018  
**Autoria:** Osmar Brune  
**Revisor:** Roberto Martiny  
**Aprovação:** Rafael Lima

Natureza das alterações:

- Primeira emissão do documento.

**Revisão:** B  
**Data:** 10/12/2018  
**Autoria:** Osmar Brune  
**Revisor:** Roberto Martiny  
**Aprovação:** Rafael Lima

Natureza das alterações:

- Correções e inclusão de novos parâmetros na seção 5.