Temas:

O que é o quic(explicar por cima)

-Importância desse protocolo no âmbito do IOT

-Importância do projeto

-Sucessos e insucessos

O protocolo QUIC é um protocolo da camada de transporte orientado à conexão(rfc 9000) baseado no UDP que é criptografado de maneira nativa. O protocolo possui diversas features, sendo estas:

* Multiplexação de fluxos QUIC através do Id de conexão;
* Transmissão dos dados em uma via(0-RTT);
* Evita o Head-of-line Blocking(rfc 9000 parte 13.0)

O protocolo MQTT(Message Queue telemetry transport) é um protocolo de mensageria que permite que dispositivos remotos enviem mensagens com uma pequeno volume de código e com uma pequena largura de banda. O MQTT, atualmente é utilizado com o protocolo TCP.

====================================================================

Roteiro do relatório:

* Qual o objetivo do trabalho(introdução, ler o plano de atividades e o exemplo enviado por Ramon) ?
* Explicar o que é MQTT;
* Explicar o que é o QUIC;
* Como o QUIC se relaciona com o MQTT(possíveis benefícios);
* Como é a pesquisa(metodologia idealizada);
* Principais dificuldades e o que se alcançou;
* Parecer do orientador;
* Referências bibliográficas(RFC,livro de tanenbaum de SO)

Mudanças:

* Pesquisa sobre um repositório que usa QUIC com mqtt demorou um tempo;
* Aprendizado sobre as particularidades do funcionamento do NanoSDK em relação ao ambiente;

RFC’s de referencia:

* RFC do TLS 1.3
* RFC do QUIC
* RFC do MQTT

Artigos de referência:

* Artigo de Kumar
* Artigo do QUIC explicando o funcionamento dele(autoridade);
* Link do repositório do NanoSDK;

======================================================================

======================================================================

OBS:

* As referencias estão em vermelho
* As seções de comentários que podem ser retirados ou movidos estão em laranja

============================================================================================================================================

**Introdução ao tema estudado**

Esse relatório parcial tem como finalidade relatar os resultados do projeto de pesquisa “AVALIAÇÃO DE DESEMPENHO E SEGURANÇA DO PROTOCOLO QUIC EM DISPOSITIVOS DE INTERNET DAS COISAS ”(Cavalcanti,D..,2023). Portanto o projeto tem dois tópicos a serem estudados, o protocolo QUIC e o protocolo MQTT.

O protocolo QUIC é um protocolo da camada de transporte orientado à conexão que trabalha em cima do UDP(referência da RFC 9000 parte do overview). O protocolo QUIC é uma alternativa ao protocolo TCP por resolver problemas de head-of-line blocking além de implementar a segurança de forma nativa a partir do TLS 1.3, não necessitando de outro handshake(parte 7 da RFC 9000). Tendo em vista o aumento do tráfego de dados na internet aliado a uma crescente necessidade de abordagens de segurança mais eficazes, incluindo os dispositivos IOT, faz-se necessário estudar alternativas que incrementem a eficiência e segurança no tráfego desses dados.

O MQTT é um protocolo de mensageria com arquitetura Publish/subscribe entre um cliente e servidor. O mesmo é leve e projetado para ser fácil de implementar, sendo adequado para contextos Machine-to-Machine e IOT onde é necessário poucas linhas de código e/ou a largura de banda da rede é algo valioso(Abstract do <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html> ).

Os objetivos do trabalho é realizar uma análise de desempenho e segurança do protocolo QUIC em dispositivos IOT. O mesmo pode ser feito a partir de comunicações puramente QUIC ou com a utilização do MQTT sobre o QUIC, já que o MQTT tem se tornado um protocolo de aplicação de facto padrão para implementações de aplicações IoT(CAVALCANTI, D. J. M., CARVALHO, R. and ROSA, N. S., **Adaptive Middleware of Things**. *2021 IEEE; referencia retirada do* plano de atividades);

**Metodologia**

Primeiro foi feita um estudo do estado da arte do que é o protocolo QUIC a partir da leitura das RFC 8999(RFC 8999),para se estudar como o protocolo quic deve se comportar independentemente das versões futuras, como por exemplo: A estruturação dos pacotes de handshake, a negociação de versão e dentre outros. Além disso, também foi estudada a RFC 9000(sendo a versão 1 do protocolo para se estudar como é o padrão do protocolo).Além disso foi lido a RFC 8446 (para se informar como funciona o TLS 1.3). Logo após isso foi feita uma pesquisa das implementações existentes do protocolo MQTT sobre o QUIC(sendo encontrado as implementações feitas pelo EMQX). Também foi feita uma análise na implementação do projeto MsQUIC

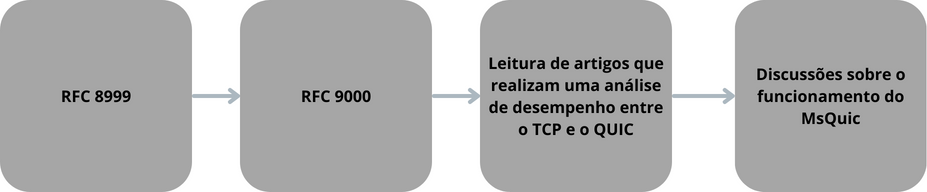
============================================

**descrição: Etapa de estudo do estado da arte(desenho/gráfico sequencial):**

RFC 8999→RFC 9000→RFC 8446→leitura de artigos que realizam uma análise de desempenho entre o TCP e o QUIC→ Discussões sobre o funcionamento do msquic(implementação da microsoft do protocolo QUIC no framework .NET)

============================================

IMAGEM:



Em paralelo a isso, também foram feitas reuniões semanais na forma de apresentação junto com algumas exemplificações no wireshark. A posteriori, foi feita uma palestra na SNCT no IFPE Igarassu.

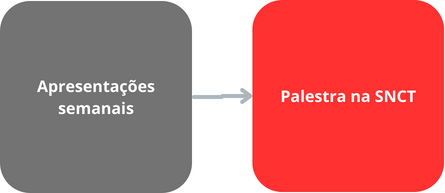
======================================================

**descrição: Apresentações(desenho/gráfico sequencial)**

Apresentações semanais->palestra na SNCT

======================================================

IMAGEM:



Logo após isso, foi feito um estudo de como funciona o protocolo a partir de tentativas de implementações de funções do protocolo QUIC em C(<https://github.com/davifurao/Dquic> ), por exemplo: Geração de ID de conexão, Estruturação dos cabeçalhos dos pacotes e criação de frames simples para entender melhor como o protocolo QUIC funciona. Também foi estudado implementações dos algoritmos de criptografia Hash(RFC 6234)

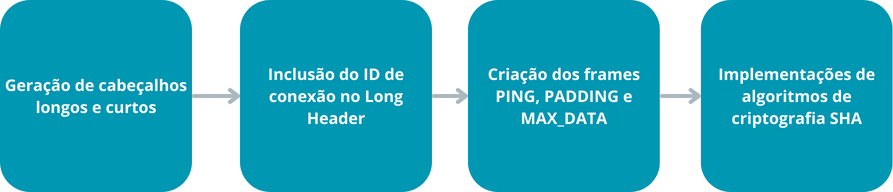
===========================================================

**descrição: Aprendizado com código(fluxograma)**

Geração de cabeçalhos longos e curtos→ Inclusão do ID de conexão no Long Header→Criação dos frames Ping, padding e Max\_Data →implementações de algoritmos de criptografia SHA

===========================================================

IMAGEM:



Sobre o ambiente de teste, foram analisadas várias alternativas, desde o arduino, ESP8266 até o Raspberry Pi. O que foi constatado é que o Raspberry Pi é a alternativa mais viável tendo em vista de que nem mesmo uma implementação do QUIC para a biblioteca do Arduíno está disponível na internet e que uma possível implementação levaria ao menos um ano, não sendo o objetivo do trabalho atualmente.

Atualmente o trabalho encontra-se na parte de implementações do MQTT sobre o QUIC no NanoSDK(<https://github.com/emqx/NanoSDK> ) numa máquina virtual ubuntu.

Depois da etapa de implementação do MQTT sobre o QUIC, será analisado os pacotes pelo Wireshark e a análise de tempo poderá ser feita em trechos do código em que

a mensagem do publish é um timestamp(Tpub), que será subtraído pelo valor obtido depois do recebimento da mensagem no subscriber(Tsub). O valor será um módulo, como é mostrado na imagem abaixo:

============================================================

**descrição: Coleta de dados em MQTT**

Latência = módulo(a mensagem que será enviada do publisher será um timestamp - tempo depois que a mensagem é recebida pelo subscriber )

Latência = | Tpub - Tsub |

============================================================

IMAGEM:



Caso a implementação do MQTT/QUIC tenha problemas(pois os projetos da EMQX ainda estão em Beta), será feito uma implementação em QUIC puro num dispositivo IOT(Raspberry Pi), com a análise de pacotes pelo wireshark e o armazenamento dos momentos em que cada método é chamado, desde o envio da informação do cliente até o recebimento da informação no servidor. O valor será um módulo, como é mostrado na imagem abaixo:

============================================================

**descrição: Coleta de dados em QUIC/TCP/UDP puro**

Latência = módulo(tempo atual depois da chamada de método que aloca a mensagem no buffer no cliente - tempo atual depois da leitura do buffer no servidor )

Latência= | Tsend - Treceive |

============================================================

IMAGEM:



O mesmo será feito com o TCP e UDP, após a atividade de implementação, será definido os testes incluindo outros detalhes

**Resultados**

Foi constatado que está sendo transmitida a mensagem MQTT sobre o QUIC pelo NanoSDK a partir do código de exemplo do projeto() na máquina virtual ubuntu, entretanto ainda não foi visualizado qualquer pacote QUIC ou MQTT pelo wireshark na porta UDP em que foi publicado a mensagem.

No decorrer dos primeiros meses da execução do plano de atividades vinculado a este relatório parcial as seguintes atividades foram executadas:

1. Revisão bibliográfica do protocolo QUIC e TLS 1.3
2. Estudo do MQTT
3. Apresentação no SNCT na forma de uma oficina.
4. Pesquisa sobre implementações MQTT/QUIC e QUIC puro
5. Considerações acerca do testbed(estudo do que seria o testbed)

**Discussões**

Foram encontrados diversos desafios até o momento, incluindo a escassez de projetos estáveis com o MQTT sobre o QUIC, podendo, a depender dos resultados futuros, alterar a análise de desempenho de MQTT/QUIC para o QUIC nativo tendo em vista o plano de atividade.

**Conclusão**

Esta pesquisa foi criada a partir

**Parecer do orientador**

Essa parte é exclusiva para ser feita pelo Orientador.

**Referências**

IYENGAR, J., THOMSON, M. **RFC 9000 - QUIC: A UDP-Based Multiplexed and Secure Transport.** IETF,2021 Disponível em: <https://datatracker.ietf.org/doc/rfc9000/>

In-text citation: (IYENGAR; THOMSON, 2021)

==========================================

CAVALCANTI, D. J. M., CARVALHO, R. and ROSA, N. S., Adaptive Middleware of Things. 2021 IEEE Symposium on Computers and Communications (ISCC), Athens, Greece, 2021, pp. 1-6, doi: 10.1109/ISCC53001.2021.9631408.

In-text citation: (CAVALCANTI; CARVALHO; ROSA, 2021)

===========================

MQTT Version 5.0. Disponível em: <<https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>>.

In-text citation: (“MQTT Version 5.0”, [s.d.])

==========================================

microsoft/msquic. Disponível em: <https://github.com/microsoft/msquic>. Acesso em: 10 mar. 2024.

‌In-text citation: (“microsoft/msquic”, 2024)

==========================================

LUNA, D. davifurao/Dquic. Disponível em: <https://github.com/davifurao/Dquic>. Acesso em: 10 mar. 2024.

‌

In-text citation: (LUNA, 2023)

==========================================

HANSEN, T.; EASTLAKE 3RD, D. E. US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF). Disponível em: <https://www.rfc-editor.org/rfc/rfc6234.html>. Acesso em: 10 mar. 2024.

‌

In-text citation: (HANSEN; EASTLAKE 3RD, 2011)

==========================================

THOMSON, M. RFC 8999: Version-Independent Properties of QUIC. Disponível em: <https://www.rfc-editor.org/rfc/rfc8999.html>.

‌

In-text citation: (THOMSON, [s.d.])

==================================

RESCORLA, E. rfc8446. Disponível em: <<https://datatracker.ietf.org/doc/html/rfc8446>>.

In-text citation:(RESCORLA, 2018)

============================================================================

EMQ Technologies. Disponível em: <https://github.com/emqx>. Acesso em: 10 mar. 2024.

In-text citation:(“EMQ Technologies”, [s.d.])

==================================

emqx/NanoSDK. Disponível em: <https://github.com/emqx/NanoSDK>. Acesso em: 10 mar. 2024.

‌

In-text citation: (“emqx/NanoSDK”, 2024)