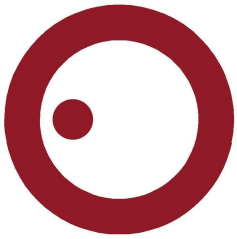


IPL

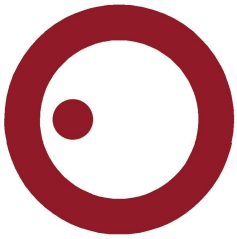
escola superior de tecnologia e gestão
instituto politécnico de leiria

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

ANÁLISE DE SEGURANÇA A PROTOCOLOS DE COMUNICAÇÃO DE ELEVADA LATÊNCIA

ESTUDANTE TIAGO ANDRÉ SANTOS GASPAR

Leiria, Março 2022



escola superior de tecnologia e gestão
instituto politécnico de leiria

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

ANÁLISE DE SEGURANÇA A PROTOCOLOS DE COMUNICAÇÃO DE ELEVADA LATÊNCIA

ESTUDANTE TIAGO ANDRÉ SANTOS GASPAR
Número: 2190227

Dissertação realizada sob orientação do Professor Doutor Paulo Jorge Gonçalves Loureiro (paulo.loureiro@ipleiria.pt).

Leiria, Março 2022

AGRADECIMENTOS

Tenho a agradecer à minha família pelo apoio e por estar sempre presente. Aos meus amigos pela preocupação e ajuda que forneceram ao longo destes anos. Agradeço ao meu orientador esta oportunidade, pela dedicação de todas as semanas estar lá e pela paciencia e ajuda que sempre teve. Por ultimo e de não menos importante quero agradecer à Helga, a minha psicologa por me meter sempre a cabeça no rumo certo e nunca me fazer desistir.

RESUMO

As redes de comunicação interplanetárias enfrentam desafios que as redes de comunicação terrestre não enfrentam. Com o evoluir da tecnologia é importante analisar as soluções de segurança e verificar se as mesmas têm impacto na performance da rede.

Com este trabalho o objectivo é verificar se a introdução de mecanismos de segurança nas comunicação interplanetárias tem impacto na performance da rede. Para isso foram implementados testes numa rede DTN utilizando a implementação ION com e sem segurança para se conhecer o seu impacto.

Com base nos resultados conseguidos foi concluído que o impacto dos mecanismo de segurança na performance da rede se altera consoante o tamanho dos ficheiros transmitidos e sendo que o mecanismo que mais teve influência foi o de verificação de confidencialidade.

ABSTRACT

Interplanetary communication networks face challenges that terrestrial communication networks do not. With the evolution of technology, it is important to analyze security solutions and verify if they have an impact on network performance.

With this work, the objective is to verify if the introduction of security mechanisms in interplanetary communication has an impact on the performance of the network. For this, tests were implemented in a DTN network using the ION implementation with and without security to know its impact.

Based on the results obtained, it was concluded that the impact of security mechanisms on network performance changes depending on the size of the transmitted files and that the mechanism that had the most influence was the confidentiality check.

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

AOS	Advanced Orbiting Systems.
BAB	Bundle Authentication Block.
BP	Bundle Protocol.
BPA	Bundle Protocol Agent.
BSP	Bundle Security Protocol.
CCSDS	Consultative Committee for Space Data Systems.
CGR	Contact Greph Routing.
CL	Convergence Layer.
CP	Checkpoint.
DOS	Denial of service.
DTN	Delay-Tolerant Network.
EOB	End of block.
EORP	End of red Part.
EPP	Encapsulation packet protocol.
FTP	File transfer protocol.
HTTP	Hypertext Transfer Protocol.
ION	Interplanetary overlay network.
LLC	Logical link control.

LTP	Licklider Transmission Protocol.
MAC	Media acess control.
NASA	National Aeronautics and Space Administration.
OSI	Open Systems Interconnection.
PCB	payload confidencialy block.
PIB	payload integrity block.
RFC	Request for Comments.
RTO	Retransmission TimeOut.
RTT	Road trip time.
TCP	Transmission Control Protocol/Internet Protocol.
TCP/IP	Transmission Control Protocol/Internet Protocol.
TLS	Transport Layer Security.
UDP	User Datagram Protocol .
VoIP	Voice Over Internet Protocol .

1

INTRODUÇÃO

Nos dias que correm as tecnologias de comunicação utilizadas são desenvolvidas para terem bom desempenho para valores de parâmetros de transmissão (taxa de transferência, latência, perdas, etc.) ajustados à dimensão do planeta terra. Para ações do dia-a-dia, como realizar chamadas de voz sobre IP (VoIP) ou assistir a vídeos pela Internet a latência não supera os 200 ms (Caini, 2018) e as taxas de erro são inferiores a 0.1%. Ambientes como este já não são considerados muito desafiadores porque não possuem elevadas taxas de erro nem alta latência e as tecnologias de comunicação utilizadas que seguem o modelo de referência TCP/IP (Mohr, 2009) já se encontram num estado de maturação muito elevado (Tselikis, 2013).

O modelo de comunicação de referência TCP/IP é baseado no modelo OSI desenvolvida em 1971 (Maathuis, 2003). O modelo OSI apresenta uma arquitetura baseada em sete camadas: Camada física, de Ligação de dados, Rede, Transporte, Sessão, Apresentação e Aplicação. A camada Física define as características associadas à transmissão, por exemplo, características de conectores, de interfaces e de modulação de sinais. A camada de Ligação de dados define soluções para o controlo de erros que ocorrem na camada física e para o controlo no acesso ao meio. Na camada de Rede definem-se os procedimentos para o encaminhamento de pacotes, e para o endereçamento de dispositivos. Os IPv4 e IPv6 são dois dos principais protocolos de endereçamento. A camada de Transporte faz o controlo de erros, o controlo de fluxo, a multiplexagem e a entrega confiável de mensagens. Os TCP e UDP são dois dois protocolos de transporte mais utilizados. A camada de Sessão gere as sessões que são criadas entre aplicações. A camada de Apresentação prepara os dados para a camada de Aplicação. A camada de Aplicação fornece ao utilizador as funções que permitem o acesso a serviços de rede, tais como, o serviço de email (SMTP, IMAP, POP3), web (HTTP, HTTPS), transferência de ficheiros (FTP), e outros.

O modelo de referência TCP/IP está estruturado em quatro camadas, as camadas Física, de Rede, de Transporte e de Aplicação (Ondeng, 2003). Este modelo é uma simplificação do modelo OSI pois algumas das camadas deste modelo agregam

funções de diferente camadas. A camada Física agrega as funções duas primeiras camadas do modelo OSI (Física e de ligação de dados). A camada de Aplicação agrega as funções das três últimas camadas do modelo OSI (Sessão, Apresentação e Aplicação). Nesta camada estão os serviços que dão suporte a aplicações.

Os protocolos TCP/IP são desenvolvidos e afinados para contextos de transmissão que apresentam valores de latência na ordem de entre 10 ms e 300 s. No caso de o protocolo de transporte utilizado ser o UDP, a latência não deve ser superior a 30 s. Caso o protocolo de transporte seja o TCP o valor padrão do parâmetro Timeout é de 300 s.

O valor da latência nas comunicações atuais é muito mais baixo relativamente a latências que podem ser encontradas em ambientes fora do planeta terra ou em comunicações interplanetárias, onde as latências podem ter valores variáveis entre 1 s e 18000 s (dentro do sistema solar) (NASA.gov, 2006). Neste caso a utilização de alguns protocolos TCP/IP não é viável. Para tal foi necessário desenvolver um conjunto de novos protocolos de comunicação que suportassem latências variáveis elevadas e altas taxas de erros.

A Delay-Tolerant Networking (DTN) é uma rede de comunicação desenvolvida para ligações interplanetárias, com suporte para elevadas latências. A arquitetura DTN segue o modelo TCP/IP e introduz uma nova camada designada por camada Bundle. Esta camada permite o armazenamento de informação que se encontra em trânsito nos nós intermédios da rede e efectua verificações de integridade da informação. Este modelo é detalhado no capítulo 3.

1.1 OBJETIVOS

Os objectivos do trabalho desenvolvidos são estudar o modelo de comunicação desenvolvido para as comunicações interplanetárias (DTN), os protocolos específicos deste tipo de rede e avaliar o impacto que os mecanismos de segurança desenvolvidos para as redes DTN têm no desempenho global da rede DTN.

O processo de avaliação dos protocolos DTN consistiu na implementação de testes de comunicação em cenários interplanetários que representam as condições de comunicações interplanetárias. Os testes são organizados em blocos, para condições de rede com e sem protocolos de segurança. Desta forma é possível comparar os resultados obtidos e avaliar o impacto da utilização dos protocolos de segurança no desempenho global da rede. As condições dos testes englobam o tipo de dados

(texto, imagens, vídeo), o volume de tráfego, a duração dos testes e a definição do cenário da rede (número de nós, tipo de nós e localização dos nós).

1.2 ORGANIZAÇÃO DO DOCUMENTO

Este documento está organizado em cinco capítulos. O primeiro capítulo é a Introdução e apresenta uma breve descrição dos modelos TCP/IP, OSI e DTN, e os objectivos do trabalho desenvolvido e dos testes realizados.

O capítulo dois apresenta uma revisão sobre a literatura sobre as redes DTN, ou seja, uma descrição de trabalhos publicados sobre redes DTN.

O capítulo três apresenta uma descrição detalhada da arquitectura DTN e dos protocolos específicos deste tipo de rede.

O capítulo quatro apresenta a descrição detalhada do trabalho realizado, ou seja, apresenta o cenário definido para a realização dos testes, a descrição dos testes realizados, a análise dos resultados obtidos e a identificação das ferramentas utilizadas para o tratamento dos dados obtidos com os testes.

O capítulo Conclusão apresenta o resultado global da análise aos resultados dos testes, bem como as conclusões sobre o trabalho desenvolvido.

2

TABALHO RELACIONADO

A arquitetura da Internet foi definida para condições de rede padrão, com poucas oscilações e com tempos de entrega de pacotes inferiores a 200 ms (Caini, 2018). Deste modo, os protocolos TCP/IP não incluem mecanismos que permitam o seu bom funcionamento em situações extremas, tais como, taxas de erro altas e latências elevadas.

As comunicações interplanetárias apresentam condições de rede consideradas extremas, onde os protocolos TCP/IP não podem ser aplicados com bom desempenho. Neste sentido novos protocolos foram desenvolvidos e enquadrados no modelo de rede Delay-Tolerant Networking (DTN).

Figura 1: Modelo TCP/IP

A arquitetura de rede DTN é semelhante á arquitetura TCP/IP na medida que é constituída por camadas e inclui alguns dos protocolos TCP/IP. No entanto, inclui diferenças substanciais com enorme impacto no desempenho da rede, como é o caso da introdução de uma nova camada entre a camada de aplicação e a camada de transporte denominada de Bundle.

O Bundle Protocol (BP) tem como objetivo interligar vários tipos de ligações diferentes, ou seja, pode receber dados por via do protocolo TCP e reenviar os

mesmos dados por via do protocolo UDP ou vice-versa (Burleigh, 2016). Em DTN, os dados são agrupados em Bundles, os quais têm o mesmo objetivos que os pacotes usados nas comunicações de Internet. Os Bundles transportam dados entre endpoints do Bundle Protocol.

O Licklider Transmission Protocol (LTP) é o protocolo definido para o transporte de Bundles. Este protocolo garante a entrega de dados em redes de comunicação com grandes latências, elevadas taxas de erro e estações que podem ser móveis, como é o caso das redes de comunicação interplanetárias. A utilização do protocolo LTP nas redes DTN não é obrigatório, pois nem sempre existem condições de rede com grande latência entre nós. Nestes casos o modelo de rede DTN suporta a utilização de outros protocolos de transporte, tais como, o TCP, o UDP ou outros protocolos de transporte. O modelo DTN define uma nova camada (Convergence Layer) para realizar as adaptações entre os segmentos de rede que usam o protocolo LTP e os segmentos da rede que usam qualquer outro protocolo de transporte.

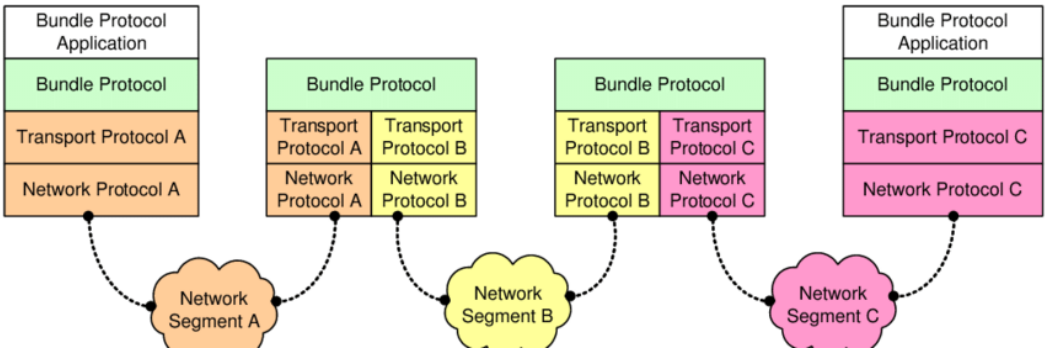


Figura 2: Arquitetura de rede com Bundle protocolo

A National Aeronautics and Space Administration (NASA) desenvolveu a interplanetary overlay network (ION) que é uma implementação da arquitetura e dos protocolos DTN e a a partir da qual é possível criar cenários de redes interplanetárias, realizar testes de comunicação e avaliar o desempenho dos seus protocolos.

A arquitetura DTN foi desenvolvida com o objetivo desta rede poder ser implementada em ambientes de condições de transmissão extremas e nas quais as questões de segurança da informação também fossem uma preocupação. O modelo TCP/IP define mecanismos de segurança, tal como o Transport Layer Security (TLS), através da adição de uma camada de segurança em cima de cada protocolo de comunicação. No caso do protocolo TLS é utilizada tanto encriptação simétrica

como encriptação assimétrica e, desse modo, são transmitidos pacotes privados com informação segura bem como autenticação dos pacotes recebidos.

O modelo DTN definiu um protocolo de segurança para com o objetivo de assegurar a transmissão de dados segura denominado de Bundle Security Protocol (BSP). Este protocolo foi especificado para garantir a confidencialidade entre todos os Nós da rede DTN.

2.1 ANÁLISE DE MECANISMOS DE SEGURANÇA PARA DTN

A implementação de mecanismos de segurança para a comunicação em rede deve ser flexível. Contudo, com a flexibilidade vem a complexidade, sendo que a complexidade depende de como a rede é configurada para ser segura. Para tal, em redes de comunicação usa-se políticas e chaves partilhadas para possibilitar a comunicação segura entre todos os nós (**Burleigh:2010**).

Numa rede segura as chaves têm políticas que as acompanham quando estas são distribuídas. Estas políticas são geridas por um mecanismo que é o que garante a fiabilidade das mesmas. Numa rede como a Internet a gestão de chaves é um processo crítico e complexo. Numa rede como a DTN, que é formada por nós que podem estar muito distanciados e nem sempre disponíveis, este processo é ainda mais complexo e não existe nenhum mecanismo que permita a gestão de chaves especifico para DTN Ivancic, 2010.

Neste artigo Ivancic, 2010 foram analisadas algumas das fraquezas das redes DTN, tendo sido implementados vários ataques para se avaliar se os mecanismos de segurança definidos eram realmente efetivos. Foram implementados testes com vários tipos de ataques por camadas do modelo DTN. Nestes testes foram realizados ataques do tipo Exploit através da camada de rede com o objetivo de criar situações de Denial of service (DO), ataques á camada de transporte, com o objetivo de criar DOS e também cancelamento da conexões com ataques como UDP Flooding e Man-In-the-Middle (MIM). Foi concluído neste artigo que para os ataques á camada de rede a DTN tem o protocolo Bundle Security Protocol (BSP) que inclui mecanismos de segurança como o Authentication Block (BAB), a Payload Integrity Block (PIB) e o Payload Confidentiality Block (PCB) para ajudar a combater as ameaças dos Exploits. Para a camada de transporte o LTP possui extensões de segurança que permitem combater o DOS e inclui opções para a implementação de Cookies.

Relativamente a desempenho, o artigo (Alessi, 2018) apresenta uma análise ao desempenho da rede DTN sem mecanismos de segurança. Este artigo remete para uma análise do desempenho da rede DTN para um cenário com condições de teste que representam a comunicação com o planeta Marte. Segundo este artigo a DTN possui 3 componentes essenciais, o BP, o LTP e o Contact Greph Routing (CGR) (**Alessi1:2019**). Neste artigo foi criado um cenário que integra os três elementos considerados cruciais da DTN: O BP, o LTP e o CGR. O cenário de teste implementado consiste num veículo a circular em solo marciano, com dois satélites na órbita de Marte, os quais enviam dados científicos para o planeta Terra, onde estão três estações terrestres que representam as antenas da DSN (Deep space network), uma base terrestre um satélite na órbita terrestre e uma terceira base terrestre que se ligada a uma das bases terrestre.

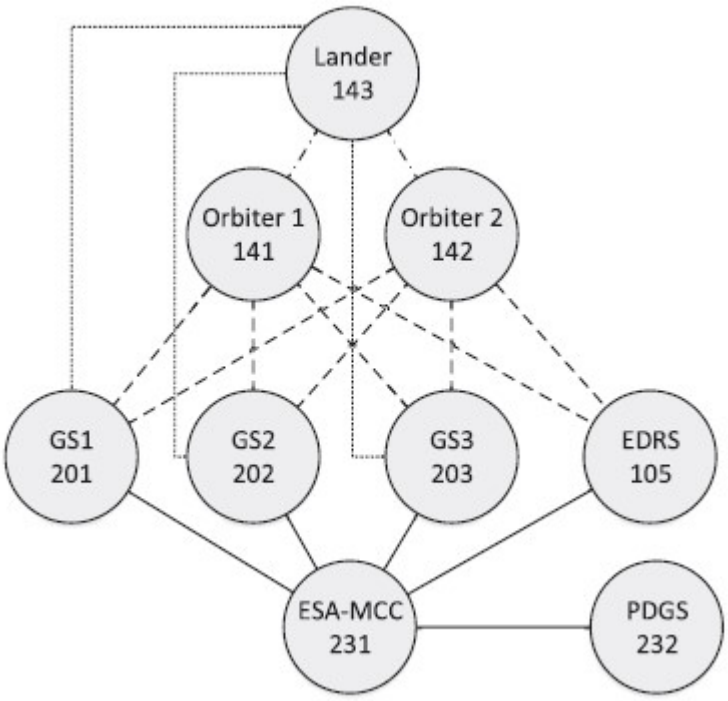


Figura 3: Cenário utilizado nos testes apresentados no artigo

Os testes realizados utilizaram uma máquina virtual com a implementação da DTN ION. Foi concluído que as comunicações no sentido Terra-Marte mostraram que o tempo de envio foi menor comparativamente ao sentido oposto da comunicação e que com o aumento da quantidade de informação enviada poderia ocorrer dispersão em termos de encaminhamento de bundles.

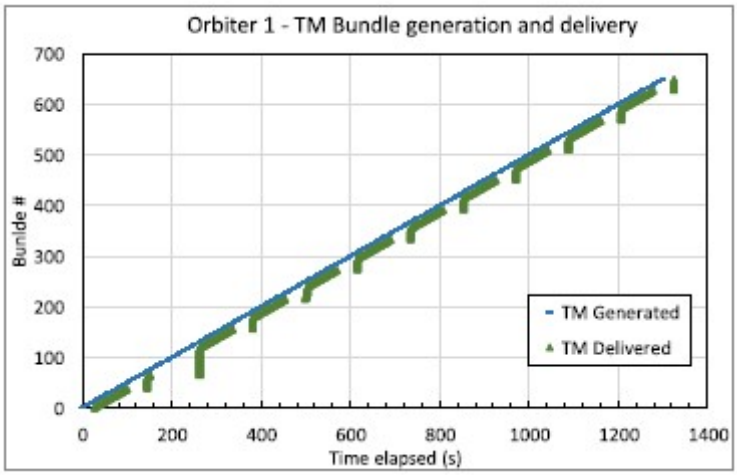


Figura 4: Resultados 1

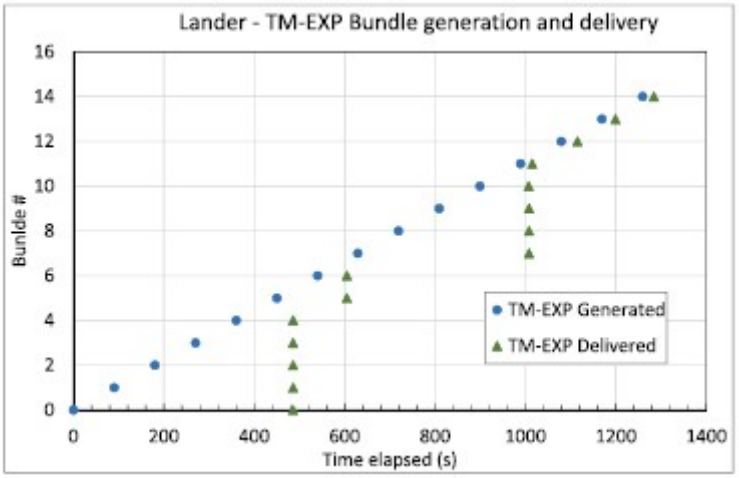


Figura 5: Resultados 2

Na Figura 4 é apresentado o envio e a recepção de bundles. Os pontos azuis representam os Bundles enviados sendo que durante o tempo decorrido, eixo dos X, foram emitidos 650 Bundles, os pontos a verde representam os Bundles que foram efetivamente entregues e os mesmos seus tempos de chegada. Da análise do gráfico, podemos concluir que a criação de Bundles é continua sendo que a receção já não é continua, visto que se trata de uma comunicação intermitente e apenas quando existe possibilidade de comunicação é que os Bundles podem ser entregues. Até ao tempo de 200 segundos foram criados 100 Bundles e foram entregues 60 aproximadamente. Os restantes 40 Bundles que estavam em falta foram entregues assim que houve possibilidade de comunicação.

No caso do gráfico representado na Figura 5 a forma de interpretação é a mesma, até ao tempo de 400 segundos o número de Bundles criados tinha sido de 4. Neste

caso a comunicação restabeleceu-se no tempo de 500 segundos e os 5 bundles criados até á altura (5) foram imediatamente recebidos.

2.2 SÍNTESE

Neste capítulo foram apresentados alguns trabalhos sobre a avaliação do desempenho da rede DTN com e sem mecanismos de segurança. No capítulo seguinte são apresentados a arquitectura da DTN bem como como os mecanismos de segurança que a mesma possui para proteger as suas ligações e os dados transmitidos.

3

DELAY TOLERANT NETWORK

A Delay Tolerant Network (DTN) foi desenvolvida pela Nasa com o objetivo de se criar um sistema de rede capaz de suportar comunicações interplanetárias. Este conceito de rede começou a ter o seu desenvolvimento a partir do inicio do século XXI Reinhart (2013).

O desenvolvimento do modelo DTN teve diversos requisitos, tais como, suportar latências elevadas e variáveis que podem ir desde alguns milisegundos até dias, suportar latências bastante superiores comparativamente a latências encontradas em redes terrestres e suportar taxas de erro altas comparativamente a taxas de erro ocorridas em ambiente terrestre.

A DTN utiliza uma solução de comunicação do tipo Store-to-Forward (NASA (2020)) (V.Samaras (2010)). Esta solução apresenta o seguinte modo de funcionamento: os dados partem de um nó fonte (emissor), estes dados são enviados em Bundles (explicado mais à frente) e vão saltando por vários nós até chegarem ao nó final. Quando um Bundle chega a um nó intermédio é armazenado em memória e enviado para o nó seguinte. Quando existe informação de que todas os Bundles de uma mensagem tenham sido enviados e recebidos pelo nó seguinte então a mensagem (formada por Bundles) é eliminada da memória desse nó, sendo este comportamento designado por Store-to-Forward.

A DTN é formada por nós localizados em ambientes desafiadores, como os ambientes interplanetários, mas também em ambientes mais controlados, como é o caso de redes DTN com nós em ambiente terrestre. As ligações entre nós localizados em ambientes interplanetários estão sujeitos a interrupções frequentes, a ligações unidireccionais, atrasos longos e altas taxas de erro. No caso dos ambientes controlados, como é o caso de redes DTN com nós num mesmo planeta os desafios são outros como no caso de planetas sem infraestruturas adequadas e com condições climáticas adversas. Os impactos de uma situação climática adversa são menos desafiadores do que o vácuo espacial.

A arquitetura da rede DTN foi desenvolvida para suportar diversos protocolos e infraestruturas de comunicação. Todas estas diferentes infraestruturas e protocolos

convergem na camada de Bundle que é o grande fator diferenciador do modelo DTN.

Esta nova camada inclui uma sub-camada denominada Camada de Convergência para servir de interface entre diferentes protocolos de transporte, como por exemplo, um segmento da rede pode estar a utilizar o protocolo TCP, o segmento seguinte da mesma rede pode estar a utilizar o protocolo LTP.

3.1 ORIGEM DA DTN

A primeira viagem tripulada a um objeto natural fora do planeta Terra foi feita em 1969 pela NASA. Nessa época já existiam métodos de comunicação com objetos fora do ambiente terrestre. Com o avançar das missões e com o avançar da tecnologia foi decidido a criação de uma Internet interplanetária e desse modo a NASA começou o respectivo desenvolvimento.

Em julho de 1998 e após uma série de sucessos com testes para uma nave espacial alojada em Marte, a NASA começou a desenvolver protocolos padrão com o objectivo de criar uma rede equivalente à Internet para comunicações com o espaço profundo. Esse desenvolvimento teve os primeiros progressos em dezembro de 2000 quando a NASA formou o grupo de investigação denominado de Interplanetary Network (IPN) e em maio de 2002 criou o modelo de rede DTN, onde se incluía o conceito de envio Store-to-Forward.

Em Novembro de 2007 é publicada a RFC5050, referência do Bundle Protocol (BP) e em setembro 2008 é publicada a RFC5326, referência do Licklider Transmission Protocol (LTP). Em fezembro de 2010 os dois protocolos são testados com sucesso, numa rede DTN que interligava a Terra e a nave espacial Earth Observer 1 (EO-1).

3.2 ARQUITETURA DA DTN

A DTN é composta por 6 camadas protocolares, cada uma com funções especificas. As camadas são; camada Física, camada de Ligação de Dados, camada de Rede, camada de Transporte, camada de Bundle, e camada de Aplicação (Khadyair, 2015). Os protocolos DTN utilizados nas várias camadas podem variar entre nós e entre interfaces do mesmo nó. Por exemplo, uma interface pode estar configurada para receber dados utilizando o protocolo de transporte UDP e a mesma interface pode estar configurada para enviar dados utilizando o protocolo LTP.

Nesta secção, as camadas DTN são apresentadas e os protocolos suportados são identificado e explicados.

3.2.1 *Bundle Protocol*

O Bundle Protocol (BP) está situado acima da camada de transporte e implementa o conceito de Store-to-Forward. O BP tem como tipo de dados suportados o Bundle. O BP tem as seguintes principais características:

- Retransmissão baseada em Store-to-Forward;
- Suportar ligações intermitentes;
- Suportar ligações calendarizadas, previstas ou oportunistas;
- Suportar valores elevados de latência.

O BP introduz também uma nova camada no modelo DTN identificada por Convergence Layer (CL) que permite dentro do mesmo nó a coexistência de vários protocolos de transporte agregados ao mesmo Bundle Protocol.

3.2.1.1 *Tecnologias do Bundle Protocol*

- Bundle
  - Método de Encapsulamento de dados do BP baseado no LTP e em Bundles.
- Bundle Payload
  - Bundles transmitidos para o nó seguinte.
- Nó
  - Estrutura física ou de software que permite a recepção e reenvio de Bundles;
  - Podem ser realizadas verificação aos dados de Bundles, podendo ser necessário executar as ações previstas sobre os mesmos;
- Agente da Bundle (BPA)
  - Componente do nó que executa os serviços do BP.
- Convergence layer Adaptapter

  - Componente do nó que envia e recebe Bundles a partir do BPA de acordo com os protocolos de transporte definidos.

- Application Agent
  - Componente do nó que utiliza os serviços do Bundle Protocol para algumas finalidades, tais como pedir a retransmissão de dados.

3.2.1.2 *Formato de Bundles*

Um Bundle é constituído por vários blocos (no mínimo dois) (Scott, 2007). O primeiro bloco é único e é denominado de primário. Todos os restantes blocos devem seguir o formato do bloco primário de modo a serem suportadas extensões. O último bloco do Bundle deverá ter a Flag de último bloco definida como 1.

Os blocos estão estruturados por campos aos quais estão associadas funções especificas. O primeiro campo do bloco primário é o campo da versão e indica a versão do Bundle Protocol. O segundo campo é o Control Flags, campo comum a todos os blocos de 21 bits e que é utilizado para identificar os processos que o Bundle vai desencadear. Os primeiros 6 bits são para caracterizar o Bundle, do 7º ao 13º definem o tipo de serviço e os restantes 7 bits são utilizados para definir os relatórios que é necessário gerar. A seguir o significado de cada bit é apresentado:

1. Fragmento.
2. Papel administrativo.
3. Não deve ser fragmentado.
4. Endpoint de destino único.
5. Conhecimento pela aplicação necessário.
6. vazio.
7. campo de 2 bits, define a prioridade (00,01,10,11).
8. Uso futuro.
9. Uso futuro.
10. Uso futuro.
11. Uso futuro.
12. Uso futuro.
13. Relatório de receção de Bundle.

14. Relatório de armazenamento aceite.
15. Relatório de seguimento de Bundle.
16. Relatório de receção de Bundle.
17. Relatório de eliminação de Bundle.
18. vazio
19. vazio

No caso do bloco não ser primário são adicionadas novas Flags de 7 bits com o objetivo de acrescentar informações sobre o Bundle, identificadas a seguir:

1. O bloco deve ser replicado em cada fragmento.
2. Transmitir um relatório de estado se o bloco não puder ser processado.
3. Eliminar o Bundle se o bloco não puder ser processado.
4. Último bloco.
5. Descartar o bloco se não puder ser processado.
6. Foi dado seguimento ao bloco sem ser processado.
7. O bloco contém campo com referência EID.

A Figura 6 apresenta os campos de um bloco primário de Bundle.

Version	Bundle Processing Control Flags *
Block Length *	
Destination scheme offset *	Destination SSP offset *
Source scheme offset *	Source SSP offset *
Report-to scheme offset *	Report-to SSP offset *
Custodian scheme offset *	Custodian SSP offset *
Creation Timestamp time *	
Creation Timestamp sequence number *	
Lifetime *	
Dictionary length *	
Dictionary byte array	
Fragment offset *	
Total application data unit length *	

Block Type	Block Processing Control Flags *	Block Length *
Bundle Payload (variable)		

Figura 6: Formato do bloco primário de Bundle

O terceiro campo é o campo do tamanho que indica o tamanho agregado de todos os campos do bloco. Os quarto e quinto campos servem para identificar o Endpoint de destino. Os campos seis e sete servem para identificar o Endepoint de origem. Os quatro campos seguintes referem-se a envio de relatórios e a pedidos de armazenamento. O décimo segundo é referente ao momento de criação do Bundle e o décimo terceiro é de Sequência, seguidamente o tempo de vida do Bundle. Neste bloco está presente um campo de Array que indica todos os nós por onde o Bundle tem de passar até chegar o destino final.

Caso o bloco não seja o primário, a estrutura do bloco é diferente. O primeiro campo é um inteiro de 8 bits que indica o tipo de bloco, isto é, indica se o bloco é um Payload. O segundo campo é o controlo das Flags, o terceiro campo é o tamanho do bloco e por último os dados enviados. Este último campo torna-se o Payload se o tipo de bloco for do tipo Payload.

Existem também blocos do tipo extensão, explicados mais à frente.

3.2.1.3 *Agente de Bundles*

O Agente de Bundle Protocol é um componente destinado a oferecer serviços de Bundle, tal como, a realização de pedidos de retransmissão de Bundles (Secretariat, 2015). O Agente fornece as seguintes opções:

- Cancelamento de transmissão;
- Verificação de receção de Bundle;
- Definição de Endpoint a enviar o Bundle.

3.2.1.4 *Convergence Layer Adapter*

O Converence Layer Adapter (CLA) é o mecanismo da DTN que permite a utilização de várias tecnologias de transporte na mesma ligação. Este mecanismo utiliza serviços nativos para fazer a conversão de dados recebidos para a tecnologia utilizada para a transmissão.

3.2.1.5 *Processamento de Bundles*

O processamento das bundles está a cargo do Agente da Bundle.

TRANSMISSÃO DE BUNDLE

Assim que o processo de transmissão é iniciado é criada um Bundle com os requisitos da transmissão. O Endpoint ID é inicialmente definido como Null porque só quando for iniciado o envio do Bundle é que é definido o Endpoint ID destino do Bundle. Nesta fase é introduzida uma limitação denominada envio pendente. Os campos referentes à identificação do nó de origem são preenchidos na totalidade.

Após a criação do Bundle é retirada a limitação introduzida anteriormente e introduzida uma nova denominada encaminhamento pendente. O agente do Bundle Protocol determina se o encaminhamento é ou não realizado, dependendo de vários fatores, por exemplo, o agente do Bundle Protocol tem de determinar o Endpoint para o qual vai ser encaminhado o Bundle. Após a escolha do Endpoint, o Bundle Protocol invoca os serviços da Convergence Layer com vista a proceder ao envio do Bundle.

RECEÇÃO DE BUNDLES

Assim que um Bundle é recebido, uma ação denominada de aguardar despacho é criada no nó. Caso no bloco primário do Bundle o 15º bit do bloco de controlo de Flags seja definido como 1 então este vai gerar um Report para enviar para o nó de origem para que este obtenha a informação que o Bundle foi entregue.

3.2.2 *Camada de Transporte*

Para o transporte de dados na DTN foi desenvolvido um novo protocolo designado por Licklider Transmission Protocol (LTP). Nesta secção são apresentados os componentes do protocolo Licklider Transmission Protocol (LTP) e o seu modo de funcionamento para vários cenários.

- Engine ID
  - Identificação do motor de funcionamento do LTP que possibilita a comunicação entre protocolos TLTP.
- Bloco
  - Array de bits com informações sobre a transmissão de Bundles a partir do LTP.
  - O bloco tem sufixos e prefixos, os sufixos são denominados de green-part e os prefixos de red-part.

- Red-Part
  - Prefixo do bloco que indica a informação que tem de ser enviada de modo confiável e a informação que caso se perca terá de ser reenviada.
- Green-Part
  - Sufixo do bloco que indica a informação que não tem de ser enviada de modo confiável e que não carece de reenvio, caso seja perdida.
- Sessão
  - Thread do protocolo LTP criada entre dois motores LTP com o objetivo de efetuar a transmissão de um bloco de dados. O fluxo de dados na sessão é unidirecional. Cada vez que é enviado um novo bloco de informação é criada uma nova sessão.
- Origem
  - Endpoint de origem da informação.
- Destino
  - Endpoint de destino da informação.
- Segmento
  - Um segmento pode ser: um bloco, um relatório, um relatório de conhecimento, um segmento de cancelamento ou um segmento de conhecimento de cancelamento.
- Objetivo
  - Identifica um subconjunto do bloco.
- EOB - Fim do bloco
  - Segmento do bloco que indica o tamanho do bloco em octetos.
- EORP - Fim da Red-Part
  - Indica o tamanho da Red-Part do bloco em octetos.
- Checkpoint
  - Segmento do bloco que solicita a criação do relatório para a origem.
- Report
  - Informação de blocos que foram recebidos pelo nó de modo a permitir a confirmação se o bloco foi todo recebido.

MODO DE FUNCIONAMENTO DO LTP

O LTP funciona por sessões (Ramadas, 2008). O processo iniciasse com o cliente a fazer um pedido ao LTP para transmitir um bloco de dados, o que corresponde a um conjunto de pacotes. Estes blocos têm dois tipos de dados denominados de Red-Part e Green-Part. Os dados Red-Part são os dados mais importantes e são enviados em ligações confiáveis, ou seja, caso os dados de Red-Part sejam perdidos eles são reenviados. Os dados Green-Part são enviados em ligações não confiáveis, ou seja, os dados perdidos não são reenviados.

O Endepoint origem especifica a identidade do Endepoint de destino para onde os dados vão ser transmitidos, bem como o tamanho total de dados transmitidos e o número de bytes presentes na componente de Red-Part. Nesta fase, a sessão é iniciada e o bloco de dados é transmitido para o nó seguinte.

A chegada do bloco de dados ao nó seguinte promove a criação de um Report com a informação dos Bundles que foram recebidos. Este Report é enviado para o nó anterior, o qual verifica se o bloco que enviou foi recebido na totalidade. Em caso positivo, o nó cria um Acknolage Report que envia para o nó seguinte, dando o processo de envio por concluído.

MODO DE FUNCIONAMENTO DO LTP EM CONDIÇÕES IDEAIS

Como o foco deste trabalho são comunicações espaciais vamos assumir a rede DTN esta a usar o Bundle Protocol nas suas ligações. O BP cria Bundles que são encapsuladas pelo LTP em blocos e deste modo evita a sobre carga da rede com múltiplos bundles. Com a utilização do BP os blocos são compostos por um ou mais Bundles e neste caso considera-se que são compostos apenas por Red-Part ou seja, dados que são importantes e que são retransmitidos, caso não cheguem ao destino. Se a comunicação é confiável, o bloco é guardado num Buffer Tx na origem até que a sessão seja concluída. Ao receber os dados, o Endpoint destino guarda-os num buffer Rx para possibilitar a reconstrução do bloco. Os dados podem não chegar ao destino ordenados porque os segmentos podem seguir ligações diferentes entre os Endpoints origem e destino.

Caso existam determinadas condições, ligações sem falhas e ausência de perdas de segmentos, Figura 7, uma sessão é iniciada e o Endepoint origem envia todos os segmentos do bloco LTP, incluindo o End of the red-Part (EORP), o Ende of block (EOB) e o CheckPoint (CP). O CP é informação de sinalização incluída no último

segmento. O período de tempo entre o envio do primeiro segmento e a chegada do último segmento do bloco é denominado Delivery Time. Em comunicações espaciais o Delivery Time pode exceder o tempo esperado devido a diversos motivos, tais como a falha de ligações ou o movimento de estações móveis. Em condições ideias o Delivery Time é aproximadamente igual ao atraso de propagação (CCSDSLTP, 2015).

Assim que o segmento referente ao CheckPoint (CP) é recebido o Endpoint destino cria um Report para devolver ao Endpoint origem com informação sobre os segmentos recebidos. O Report é enviado e assim que chega à origem, esta remove o conteúdo do Buffer Tx e a sessão na origem é terminada. O tempo de vida da sessão na origem é denominado de Export Session Life Time. Nesta fase, o Endpoint origem envia um Report de conhecimento para o destino para desencadear a finalização da sessão.

A Figura 7 apresenta o modo de transmissão de segmentos sem perdas de dados.

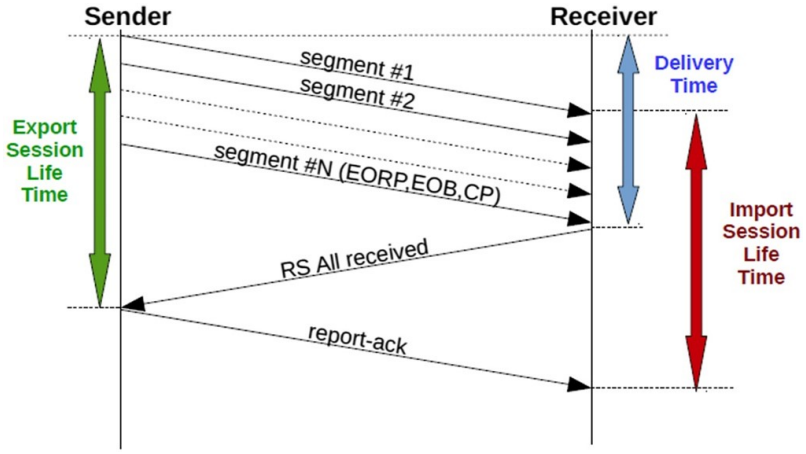


Figura 7: Modo de funcionamento do LTP sem perdas de dados

MODO DE FUNCIONAMENTO DO LTP COM PERDAS DE SEGMENTOS DE DADOS E DE CP

A sessão do LTP é criada e os segmentos são enviados. Alguns dos segmentos podem não chegar ao destino sendo que entre eles se pode incluir o segmento CP, segmento que promove a ativação do mecanismo que cria o Report no destino.

O Report é necessário para a conclusão da sessão e para a gestão dos recursos do LTP. Caso o segmento CP não seja entregue, o Report não é criado e a origem não o recebe. Para resolver este problema o LTP tem um mecanismo denominado

de Retransmission TimeOut (RTO). Quando o RTO é ultrapassado o segmento é enviado novamente com o respetivo CP.

A Figura 8 mostra o modo de transmissão com perdas de segmentos.

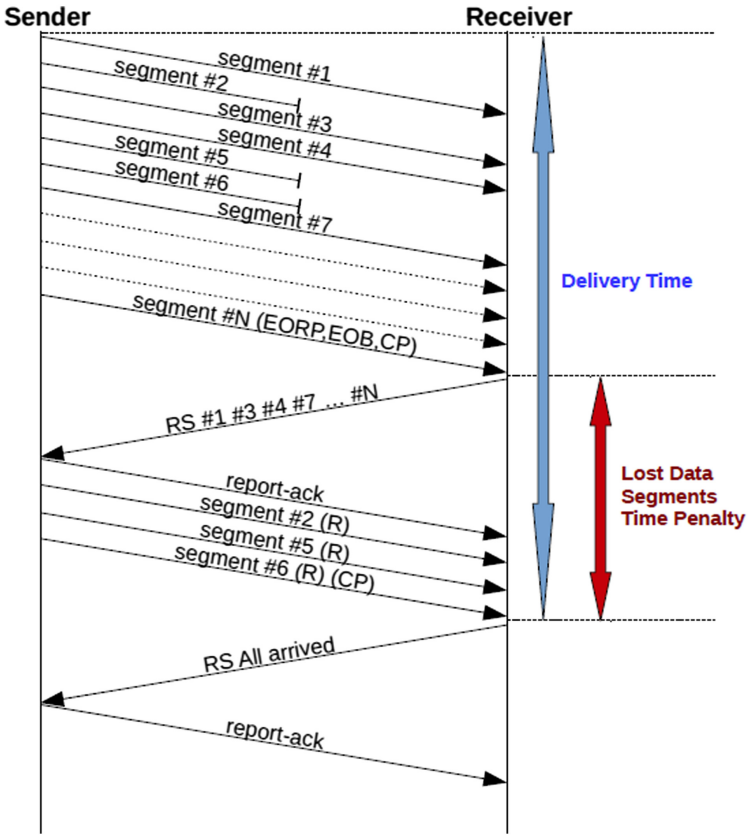


Figura 8: Modo de funcionamento do LTP com perdas de segmentos de dados e de CP

FUNCIONAMENTO DO LTP COM PERDAS DE REPORT

No caso de todos os segmentos serem enviados e recebidos com sucesso, o LTP cria um Report de conhecimento. Durante o caminho até ao nó de origem o Report pode perder-se. Neste caso, o Endepoint origem não é informado sobre a entrega do segmento de CP e reenvia novamente o segmento.

3.2.3 *Camada de Rede*

Na camada de rede, os Endpoints são identificados por um endereço IP, os Bundles encapsulados e enviados para a camada de ligação.

3.2.3.1 *Internet Protocol Extension*

O Internet Protocol Extension (IPE) é o uma extensão do protocolo IP, desenvolvida pela Consultative Committee for Space Data Systems (CCSDS). Esta extensão introduz funcionalidades para melhorar o encaminhamento de dados organizados em Bundles ao longo dos nós da rede. Nas redes tradicionais (Internet) os dados são organizados em pacotes. Na rede DTN a informação é organizada em Bundles. O endereço IP dos Endpoints é inserido nos Bundles.

A Figura 9 apresenta a estrutura dos campos do protocolo IPE.

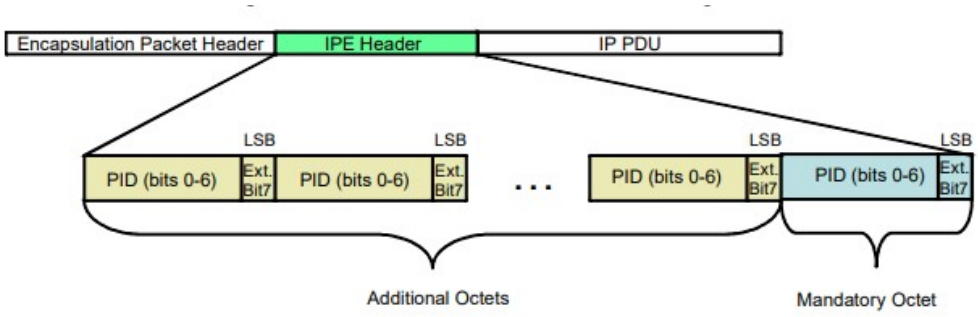


Figura 9: Estrutura do IPE

3.2.3.2 *Protocolo de Encapsulamento de Pacotes (EPP)*

O EPP é um protocolo usado para a transferência de unidades reconhecidas pelo CCSDS para comunicações entre a terra e objetos fora do planeta bem como em comunicações entre objetos situados no espaço. O objetivo do EPP é oferecer um mecanismo que permita transferir unidades de dados através de protocolos da camada de ligação específicos para comunicações espaciais.

A Figura 10 apresenta a estrutura de EPP.

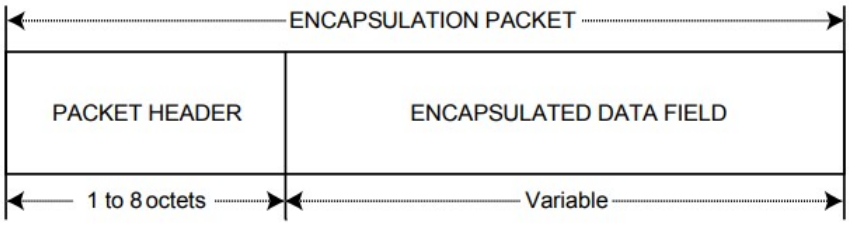


Figura 10: Estrutura do EPP

O EPP encapsula as unidades de dados em dois campos. O primeiro campo é o cabeçalho e o segundo o campo de dados. O primeiro campo é obrigatório e tem

entre 1 a 8 octetos. O segundo campo é opcional e tem entre 1 a 4294967287 bits (CCSDSEPP, 2020).

O cabeçalho possui até 8 octetos. Os primeiros 3 bits indicam o Packet Version Number (PVN), os seguintes 3 bits indicam o ID do protocolo de encapsulamento os seguintes 2 bits indicam o tamanho, de seguida são 4 bits que indicam o User Defined Field, o quinto conjunto de bits indicam o campo de extensão de protocolo de encapsulamento, o sexto é um campo destinado ao CCSDS e o sétimo é o tamanho do pacote. Depois do cabeçalho tem um campo de dados (CCSDSEPP, 2020).

A Figura 11 mostra as diversas opções de configuração de encapsulamento.

ENCAPSULATION PACKET HEADER

PACKET VERSION NUMBER 3 bits	ENCAPSULATION PROTOCOL ID 3 bits	LENGTH OF LENGTH 2 bits	USER DEFINED FIELD 0 or 4 bits	ENCAPSULATION PROTOCOL ID EXTENSION 0 or 4 bits	CCSDS DEFINED FIELD 0 or 2 octets	PACKET LENGTH 0 to 4 octets
'111'	'XXX'	'00'	0 bits	0 bits	0 octets	0 octets
'111'	'XXX'	'01'	0 bits	0 bits	0 octets	1 octet
'111'	'XXX'	'10'	4 bits	4 bits	0 octets	2 octets
'111'	'XXX'	'11'	4 bits	4 bits	2 octets	4 octets

Figura 11: Cabeçalho resultante do EPP

3.2.4 *Camada de Ligação*

A camada de ligação é responsável pela entrega de framas nó-a-nó no mesmo link. Esta camada define o controlo de acesso de media (MAC) e o controlo de ligação lógica (LLC) que garante que a ligação foi criada e organiza em Frames.

Para sistemas de comunicação no espaço a camada de ligação também define serviços de encriptação e desencriptação de dados. Nesta camada, o protocolo Advanced Orbiting Systems (AOS) Space Data Link Protocol foi desenvolvido pelo CCSDS para ligações espaciais.

3.2.5 *Camada Física*

A camada física consiste no hardware que transmite a informação que é pretendida enviar. A camada física define o que vai ser transmitido e de que maneira vai ser transmitido. os dados que chegam das camadas superiores têm de ser transformados

em sinais eléctricos para que sejam enviados para os nós de destino. No caso da informação ser enviada para fora da terra a Informação é enviada via antenas da DSN para satélites na órbita terrestre. Nesta camada, o protocolo Proximity-1 Space Link Protocol foi desenvolvido pelo CCSDS para ligações espaciais.

3.2.6 *Contact Graph Routing*

O Contact Graph Routing (CGR) é uma processo de encaminhamento adequado a ligações intermitentes ou planeadas desenvolvido para redes DTN. O CGR foi desenvolvido para escolher a melhor rota em redes DTN (Alessi, 2019).

A transmissão de dados neste tipo de ambiente é habitualmente planeada. O CGR aproveita este facto e analisa as várias possibilidades de ligação. Com esta informação, o CGR cria um gráfico de contactos ao qual aplica um algoritmo de otimização de rotas. As rotas com melhores condições de transmissão são escolhidas para o envio de Bundles. Este processo pode ser influenciado por fatores adicionais, tais como, a prioridade do Bundle, o tempo de expiração da ligação ou a tecnologia de transporte utilizada.

A Figura 12 monstra um exemplo de uma rede com vários nós. O CGR escolhe o melhor caminho para para o envio de Bundles. Neste caso, o caminho segue do nó1, nó3 e nó5. O nó5 reencaminha para o nó6.

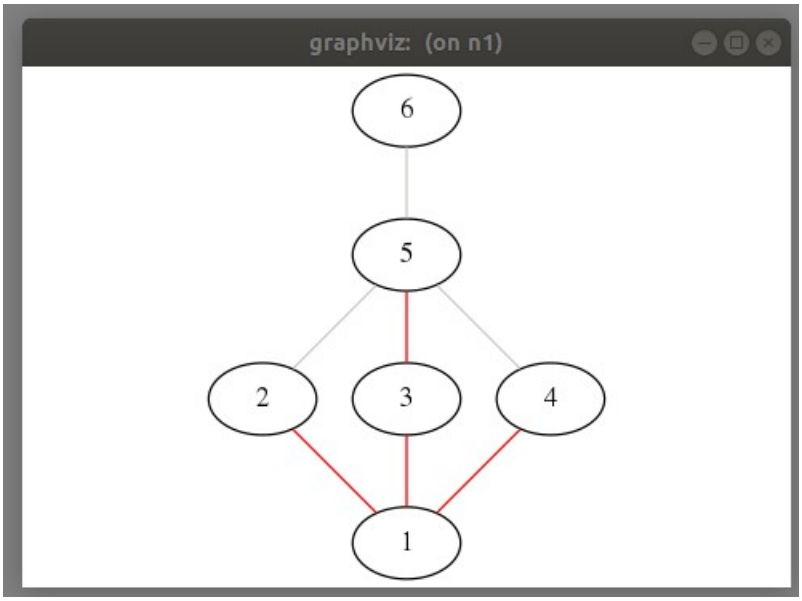


Figura 12: Gráfico de encaminhamento por CGR

3.3 SEGURANÇA EM DTN

Nesta secção são apresentados os problemas de segurança associados ao protocolo BP e as extensões de segurança desenvolvidas para o BP.

3.3.1 *Problemas de Segurança no Bundle Protocol*

O Bundle Protocol está definido na RFC5050. Aquando da sua criação não foram especificadas as extensões de segurança e isso foi um problema porque era possível aceder e manipular os Bundles em transito.

O BP sem extensões não tem mecanismos para detetar Bundles danificados ou corrompidos. Desta forma não é possível determinar se o Bundle entregue ao destino é integro ou não.

3.3.2 *Extensões de Segurança para o Bundle Protocol*

A segurança dos dados é importante para comunicações que usam o Bundle Protocol. Os ambientes sobre os quais o Bundle Protocol opera requerem que a DTN seja protegida de acesso indesejados. Porém estes ambientes possuem algumas barreiras que impõem desafios para a implementação de mecanismos de segurança, como por exemplo, a impossibilidade de gerir chaves em DTN.

Se a DTN em parte ou na totalidade for comprometida pode não assegurar os pilares de segurança de confidencialidade, integridade e disponibilidade.

3.3.2.1 *Fragmentação de Bloco*

Blocos dentro de um Bundle representam diferentes tipos de informação. O bloco primário contém a identificação do Bundle, nomeadamente a identificação dos Endpoints de origem e destino, informação necessária para o processo de encaminhamento. O bloco de Payload corresponde aos dados da aplicação.

Os Blocos de extensão contêm dados com informações adicionais para o processamento de Bundles ao logo do caminho. Podem ser aplicados mecanismos de segurança diversos em Bundles, pois os seus blocos constituintes podem representar informação diferente com necessidades de segurança também diferentes.

3.3.2.2 *Blocos de Segurança*

O Bundle Protocol (BP) foi definido para que os blocos de extensão possam ser adicionados a Bundles a qualquer momento na rede DTN. Caso o nó adicione blocos ao Bundle estes podem incluir extensões extra de segurança (ou outro tipo). Contudo, quando o nó adiciona blocos de segurança ao Bundle passa a existir uma origem do Bundle e uma origem do serviço de segurança. Por exemplo, um nó (N1) envia um Bundle para o nó (N3) passando pelo nó do meio (N2). A meio do caminho o nó N2 adiciona um serviço de segurança no Bundle. N1 é o nó de origem do Bundle e o N2 é o nó origem do serviço de segurança.

3.3.2.3 *Bundle Security Protocol*

A extensão de segurança para o BP é conhecida como Bundle Security Protocol (BSP). Esta extensão possui características interessantes porque a segurança pode ser tratada fim-a-fim ou salto-a-salto com soluções diferentes. Desta forma, os dados em trânsito podem ser mantidos seguros com chaves específicas. Outro tipo de dados adicionados aos nós podem ser seguros por outras soluções (Symington, 2011).

O BSP pode adicionar os seguintes blocos de segurança a Bundles:

1. Bundle Authentication Block (BAB): O BAB é utilizado para garantir a autenticidade e integridade do Bundle entre dois nós diretamente ligados. Ou seja, considerando o cenário com três nós (N1, N2, N3), o BAB só verifica a comunicação entre o N1 e N2 ou N2 e N3.
2. Payload Integrity Block (PIB): O PIB garante a autenticidade e integridade sobre vários nós. O PIB é equivalente ao BAB mas tem uma utilização mais global.
3. Payload Confidentiality Block (PCB): O PCB é Utilizado para garantir a confidencialidade dos dados do Bundle entre os Endpoints origem e destino.
4. Extension Security Block (ESB): O ESB foi Desenvolvido para garantir a segurança a blocos que não os de carga de dados, como por exemplo metadados. A ideia e de que as chaves do ESB, sendo diferentes de outros blocos de segurança, podem ser disponibilizadas para nós intermediários selecionados como encaminhadores DTN, sem comprometer a segurança fim-a-fim.

BLOCOS DE SEGURANÇA A adição de um mecanismo de segurança a um Bundle consiste na adição de um bloco de extensão. O bloco de extensão tem a seguinte estrutura:

- Tipo de bloco - identificação do bloco;
- Alvos de segurança - o numero de nós que este bloco vai atingir;
- Alvos de segurança 2 - o fragmento de dados onde estão identificados os alvos da operação de segurança;
- ID da cifra - utilizada para implementar os serviços de segurança;
- Pontos de controlo da cifra - campo que possui 8 bits em que cada um tem a sua função específica;
- Tamanho dos parâmetros da cifra - o tamanho do campo seguinte;
- Dados dos parâmetros da cifra - parâmetros utilizados com a cifra, por exemplo o identificador da chave;
- Tamanho dos resultados de segurança - tamanho do próximo campo;

BLOCO DE INTEGRIDADE DO BUNDLE (BIB)

O BIB é adicionado para o BAB ou o PIB pois ambos são blocos de integridade. O BAB é um mecanismo pouco utilizado devido ao facto de as comunicações deste tipo serem previstas e o BAB ter sido desenvolvido para comunicações não prevista. O BAB assina e encripta com base numa chave pública RSA (Tselikis, 2013).

Quando é recebido um Bundle que contém um bloco de integridade (BIB), o nó deve determinar se aceita as operações de segurança. Caso seja o nó de aceitação então este deve proceder ás operações de segurança e eliminar as mesmas do Bundle.

caso a execução das operações falhe é criado e enviado um relatório de erro para o nó de onde foi enviado e então o Bundle é eliminado do nó. Caso o nó que recebe o Bundle não for o nó de alvo dos mecanismos de segurança então o nó deve executar as verificações de segurança.

BLOCO DE CONFIDENCIALIDADE DE BUNDLE (BCB) A receção de um Bundle que contenha um bloco de confidencialidade (BCB) promove a determinação de se o nó é de aceitação das operações de segurança. Caso seja, o nó deve proceder ás operações de segurança e depois eliminar esse bloco do Bundle.

Se a execução das operações falhe é criado e enviado um relatório de erro para o nó de onde foi enviado e depois o Bundle é eliminado do nó. Caso o nó receptor for o destino do Bundle, o nó deve descriptografar os BCBs restantes que existam no bundle.

Caso um bloco de payload criptografado não puder ser descriptografado, o Bundle deve ser eliminado e não processado. Se um alvo de segurança criptografado diferente do bloco de carga útil não puder ser descriptografado, o alvo de segurança associado e todos os blocos de segurança associados a esse alvo devem ser eliminados e não processados. Em ambos os casos, os relatórios de estado solicitados podem ser gerados para refletirem a exclusão do pacote ou do bloco.

3.4 INTERPLANETARY OVERLAY NETWORK (ION)

A DTN é uma arquitetura de rede que foi criada com o objetivo de suportar comunicações interplanetárias. A ION é uma implementação dos protocolos da rede DTN.

3.4.1 *Design*

Esta implementação foi construída na Linguagem C para facilitar o processamento e para que os sistemas operativos, tais como as distribuições Linux, Windows, Solaris entre outros, a possam utilizar.

Esta implementação tem por princípios utilizar memoria partilhada, não copiar procedimentos e ser portável, o que permite aumentar a eficiência com que os dados são enviados entre Tasks de software, minimizar processos e economizar no processamento.

Para implementar a DTN, a ION possui módulos que suportam os protocolos de DTN. Nesse sentido, os seguintes módulos devem estar disponíveis: BP, LTP, Datagram Retransmission (DGR), Interplanetary Communication Infrastructure (ICI), CCSDS File Delivery Protocol (CFDP), Asynchronous Message Service (AMS) e Bundle Streaming Service (BSS).

3.5 DIFERENÇAS ENTRE DTN E TCP/IP

As redes DTN e TCP/IP são semelhantes na questão da arquitetura. A grande diferença entre as duas arquiteturas é a introdução da camada Bundle na DTN. Esta camada é uma solução para os desafios que as redes TCP/IP não conseguiam superar em ambientes interplanetários, como por exemplo as altas latências.

O tipo de ligação também é diferente, a rede TCP/IP utiliza o conceito de ligação constante, já no caso das redes DTN a comunicação utiliza o método de armazenar e reenviar, ou seja, sempre que não existe ligação entre um nó e o seguinte então a mensagem é armazenada até que a ligação com o próximo nó esteja disponível.

3.6 CENÁRIOS DE APLICAÇÃO

A DTN foi desenvolvida com o objetivo de proporcionar a comunicação com objetos fora da órbita terrestre ou até mesmo com outros planetas. A seguir estão identificados alguns cenários de aplicação:

1. Comunicação com objetos fora da órbita terrestre, como por exemplo a Estação espacial internacional;
2. Comunicação com objetos em outros planetas, como por exemplo a Rover Perseverance da NASA em Marte;
3. Comunicação com tripulações em missões espaciais

3.6.1 *Comunicações com objetos em outros planetas*

A comunicação com objetos em Marte requer pelo menos 4 nós, 2 nós presentes na Terra, 1 nó em órbita e 1 nó no objeto em Marte. A Figura 13 mostra um exemplo de como as ligações são estabelecidas.

Assumindo que a comunicação consiste no envio de informação do planeta Terra para Marte, o primeiro passo é criar os dados na base terrestre, depois os dados são enviados para a antena para proceder ao envio da informação para o satélite em órbita (linha vermelha). Após receber a informação, o satélite tem duas tarefas, a primeira é enviar um relatório de chegada para a antena na base terrestre, a segunda é enviar a informação que lhe chegou ao objeto em Marte. Os relatório de chegada da informação são assinalados pela linha a azul e o envio da informação a vermelho.

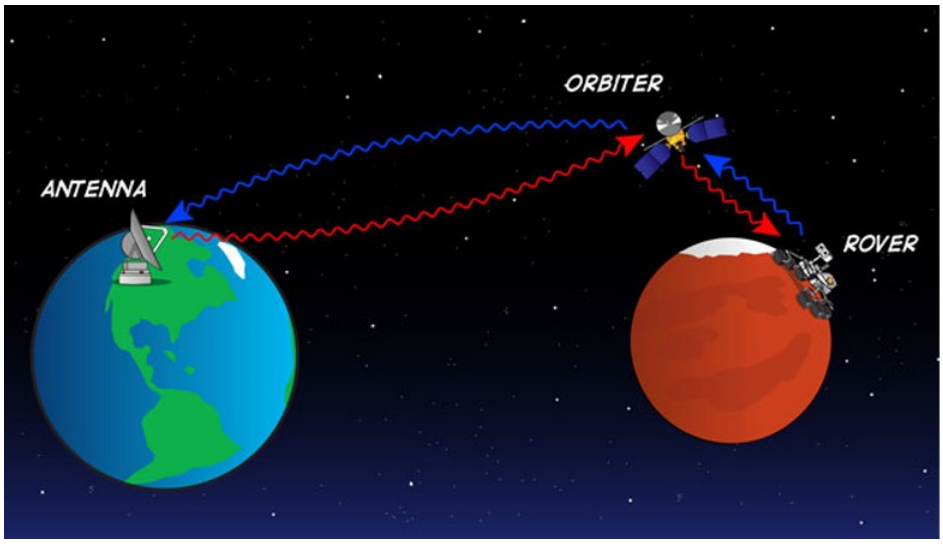


Figura 13: Comunicação Terra Marte

3.7 SÍNTESE

Neste capítulo foi explicada a estrutura e origem da DTN bem como os mecanismos de segurança que a mesma possui para proteger as suas ligações.

No próximo capítulo são apresentados os cenários de teste, as suas condições e apresentados os resultados realizados usando a DTN com e sem mecanismos de segurança para analisar o impacto dos mesmos.

4

TESTES SOBRE MECANISMOS DE SEGURANÇA EM COMUNICAÇÕES INTERPLANETÁRIAS

4.1 CENÁRIO DE TESTES

O cenário de testes foi definido com base no sistema de comunicação usadas durante as missões ao planeta Marte. Neste trabalho foi utilizado um cenário único porque este cenário representa as situações típicas a serem estudadas, ou seja, inclui latências elevadas entre a origem e o destino, taxas de erro de altas e variáveis e ligações intermitentes.

A construção do cenário começa com a definição dos componentes físicos da rede existentes entre a Terra e Marte. Para tal é necessário apresentar a estrutura da rede DTN bem como a estrutura da rede no local de chegada no planeta de destino, que neste caso é Marte. As comunicações realizadas com o planeta Terra representam muitos desafios, tais como, o facto de o planeta ter uma forma esferóide com mais de 6 300 km de raio. Para permitir que na Terra exista conectividade com os equipamentos em Marte foi necessário encontrar uma solução funcional. Este tipo de comunicação é planeada e desse modo é possível planear e enviar mensagens quando existe disponibilidade da rede. Contudo pode existir informação que não pode esperar para ser enviada e, por esta razão, o planeta tem de estar sempre contactável. Para resolver este caso foi criada Pela Jet Propulsion Laboratory (JPL) da NASA, a Deep Space Network (DSN), que consiste em ter o planeta coberto por 3 estações de antenas colocadas exatamente a 120º umas das outras para que qualquer mensagem que chegue á Terra tenha uma estação disponível para receber a mensagem. Estas estações de antenas estão situadas em Madrid, Camberra e Goldstone.

O cenário base representa a rede DSN e está apresentado na Figura 14. Neste cenário existem seis nós, quatro deles no planeta Terra um em Marte e um satélite presente na órbita marciana.

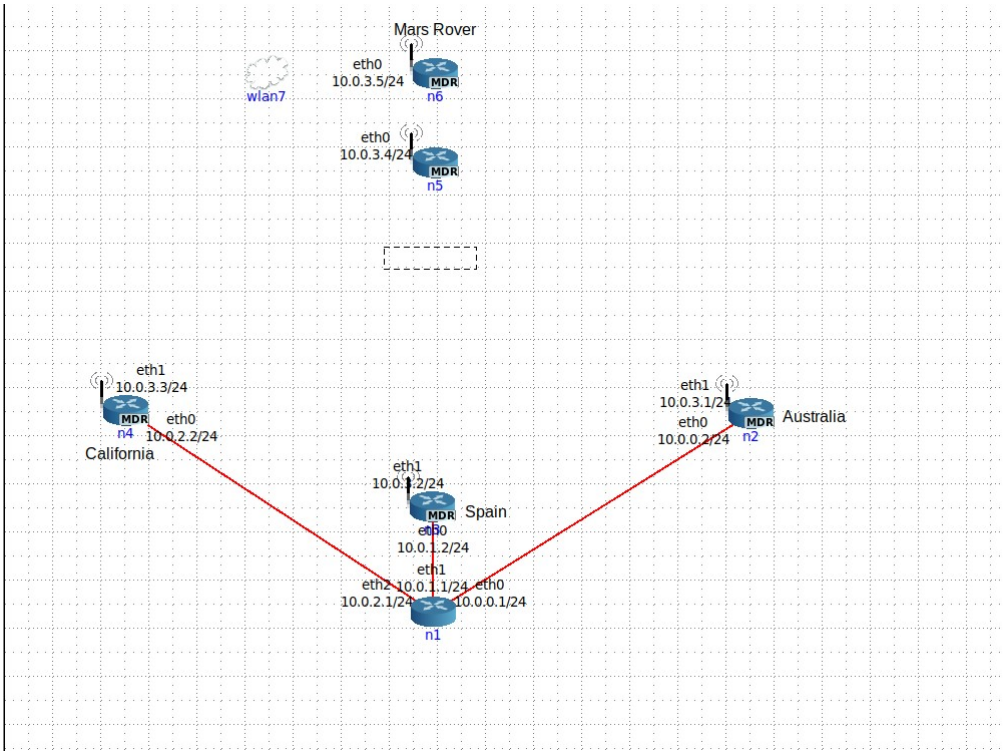


Figura 14: Cenário base para testes

4.1.1 *Estrutura do cenário de testes*

Começando pelos nós presentes no planeta Terra. O nó 1 (n1) representa a base terrestre de onde parte a informação para Marte e aonde chega a informação que vem de Marte. Esta base terrestre envia a informação para um dos nós da DSN (n2, n3 ou n4) dependendo de qual desses nós está mais perto do objetivo, ter conetividade com o nó n5. O nó que tiver a informação para enviar para Marte envia a informação para o satélite presente na órbita de Marte e este envia para o Rover que é o n6. Um processo semelhante, mas no com sentido inverso, é feito quando é o Rover a enviar a informação para a central na Terra.

4.1.2 *Ligações entre Nós*

O cenário de testes com as comunicações interplanetárias utiliza o Bundle Protocol e, desse modo, nem todas as comunicações utilizam os mesmos protocolos nas camadas de transporte, ligação ou física. Os nós n1, n2, n3 e n4 são nós presentes no planeta Terra, desse modo, não se justifica a utilização de protocolos de comunicação que

suportam alta latência ou altas taxas de erro, como por exemplo o LTP. Assim sendo temos como hipóteses mais viáveis os protocolos UDP ou o TCP. Neste cenário foi escolhido o protocolo TCP por garantir a entrega dados recebidos. Este tipo de comunicação pode ter latência variável, consoante o nó para o qual vai ser destinada a informação, sendo que, mesmo assim a latência neste caso é baixa e tipicamente varia entre 10 a 200 ms, o que no cenário global é pouco relevante. Relativamente a comunicação entre os nós da DSN e o n5 é utilizado o protocolo LTP para suportar altas taxas de erro bem como latências elevadas. O mesmo sucede na ligação entre os nós n5 e n6.

4.2 TESTES IMPLEMENTADOS

Os testes que foram realizados durante este estudo têm como objetivo analisar a performance da rede em resposta á introdução de mecanismos de segurança, para tal os testes foram separados por grupos. Os testes são divididos relativamente aos mecanismos de segurança utilizados, tendo sido criados 4 grupos de testes:

1. Sem mecanismo de segurança
2. Mecanismo PIB
3. Mecanismo PCB
4. Mecanismos PIB + PCB

O primeiro grupo de testes serve para obter a performance da rede quando não é utilizado qualquer mecanismo de segurança e, desse modo, permitir a análise comparativa sobre o impacto da introdução de mecanismos de segurança na rede. Para cada uma dos grupos são realizados vários testes com condições diferentes ao nível dos dados enviados, tais como:

1. Bundles de poucos bytes;
2. Ficheiro de texto;
3. Ficheiro de Vídeo.

Para implementar o envio de dados através da rede foi necessário ter uma vasta bateria de testes com características diferentes, desse modo foram criados testes com envios de ficheiros de vários tipos bem como envios de ficheiros de vários tamanhos com o objetivo de testar a maioria dos cenários possível. A Tabela 1 apresenta os testes realizados.

MEC. SEGURANÇA	TIPO DE TESTES
Sem	Envio constante de sinais de ping de 64 bytes Ficheiro de texto de comandos de 500kb Ficheiro de texto de comandos de 1000kb Ficheiro de vídeo do dispositivo fora da terra de forma única
PIB	Envio constante de sinais de ping de 64 bytes ficheiro de texto de comandos de 500kb ficheiro de texto de comandos de 1000kb ficheiro de vídeo do dispositivo fora da terra de forma única
PCB	Envio constante de sinais de ping de 64 bytes ficheiro de texto de comandos de 500kb ficheiro de texto de comandos de 1000kb ficheiro de vídeo do dispositivo fora da terra de forma única
PCB+PIB	Envio constante de sinais de ping de 64 bytes ficheiro de texto de comandos de 500kb ficheiro de texto de comandos de 1000kb ficheiro de vídeo do dispositivo fora da terra de forma única

Tabela 1: Grupo de testes

4.2.1 *Configuração dos nós da rede*

Antes da realização dos testes é necessário configurar o cenário de modo a que as condições que pretendemos sejam implementadas. Para isso é preciso entender como são configurados os nós na plataforma de testes. Cada nó tem 6 ficheiros de configuração cada um com o seu objetivo específico, descritos a seguir:

1. acsrc: este ficheiro tem como objetivo indicar ao ACS (administrador de sinais de custódia) quais são os comandos que vão ser executados. Neste ficheiro é inicializado o acsadmin, introduzindo o comando 1, seguido do nível de log e por último um argumento opcional que é as Heapwords que informa o ACS do numero de heap words indicado. A Figura 15 mostra a configuração do ficheiro acsrc.

```
# Aggregate Custody Signal configuration -- DZ 11/28/2014

# Initialization command (command 1).
1 7 262144
```

Figura 15: Ficheiro acsrc

2. bprc: O ficheiro bprc contém mais informação sobre o nó e a sua constituição. Neste ficheiro é indicado o esquema de ligação utilizado, os protocolos da camada de convergência, os Endpoints do nó, bem como os nós que este nó vai escutar e para onde vai enviar a informação. O comando 1 promove a inicialização, depois são introduzidos os esquemas utilizando o comando scheme. A seguir são adicionados os nós utilizando o comando endpoint, o respetivo endpoint seguido de um X se este endpoint não fizer nenhuma ação extra ao receber os bundles, ou seguido de um script se o endpoint tiver como objetivo fazer alguma ação sobre os Bundles. Em seguida são introduzidos os protocolos que o nó suporta e a capacidade em bytes para cada frame, seguido do número de bytes para o cabeçalho. por fim é introduzido os protocolos de receção suportados bem como os de envio. A Figura 16 mostra a configuração do ficheiro bprc.

```
# Initialization command (command 1).
1

# Add an EID scheme.
a scheme ipn 'ipnfw' 'ipnadminep'

# Add endpoints.
a endpoint ipn:1.1 x
a endpoint ipn:1.2 x
a endpoint ipn:1.3 x
a endpoint ipn:1.4 x
a endpoint ipn:1.5 x
a endpoint ipn:1.6 x
a endpoint ipn:1.7 x
a endpoint ipn:1.8 x
a endpoint ipn:1.9 x
a endpoint ipn:1.10 x
a endpoint ipn:1.64 x
a endpoint ipn:1.65 x

#------------------------------------------------------------------
# Add a protocol for external nodes.
#------------------------------------------------------------------
# Estimate transmission capacity assuming 1400 bytes of each frame
# for payload, and 100 bytes for overhead.
a protocol tcp 1400 100
a protocol udp 1400 100
a protocol ltp 1400 100

#------------------------------------------------------------------
# Add an induct. (listen)
#------------------------------------------------------------------
a induct  tcp 0.0.0.0 tcpcli
a induct  udp 0.0.0.0 udpcli
a induct  ltp 1 ltpcli

#------------------------------------------------------------------
# Add outducts.
#------------------------------------------------------------------
#a outduct tcp x.x.x.x tcpclo
a outduct udp 127.0.0.1 udpclo
a outduct tcp 10.0.0.2 tcpclo
a outduct tcp 10.0.1.2 tcpclo
a outduct tcp 10.0.2.2 tcpclo

#------------------------------------------------------------------
#a outduct ltp x ltpclo

#------------------------------------------------------------------
# Select level of BP watch activities - 0 = None; 1 = All
w 0

r 'ipnadmin n1.ipnrc'

# Start all declared schemes and protocols on the local node
s
```

Figura 16: Ficheiro bprc

3. ionconfig: Ficheiro de configuração que define a quantidade de memória alocada pelo ION naquele nó. A Figura 17 mostra a configuração presente neste ficheiro.

```
#wmKey      0
#sdrName        ion_sdr
#wmSize     5000000
#configFlags    1
#heapWords  5000000
pathName        /var/tmp/ion
```

Figura 17: Ficheiro ionconfig

4. ionsecrc: Ficheiro onde são identificados os mecanismos de segurança que são utilizados no nó. A Figura 18 mostra a configuração neste ficheiro.

```
# Initialization command (command 1).
1
# Select level of "echo control" activities
# 0 = None; 1 = print to both log and stdout
e 1

a key 'key2' /home/core/NASA_DTN_CORE_Scenarios/CORE_configs/mars/config/key2.key

#a bspbcbrule ipn:1.* ipn:2.* 1 'BCB-SHA256-AES128' key2
a bspbcbrule ipn:1.* ipn:6.* 1 'BCB-SHA256-AES128' key2
```

Figura 18: Ficheiro ionsecrc

5. ipnrc: é o ficheiro que indica os contactos possíveis que o nó pode ter. para isso, é utilizado o comando “a plan“ para adicionar contactos, seguido do protocolo de comunicação que vai se utilizado na comunicação bem como o endereço do nó de destino. A Figura 19 mostra a configuração neste ficheiro.

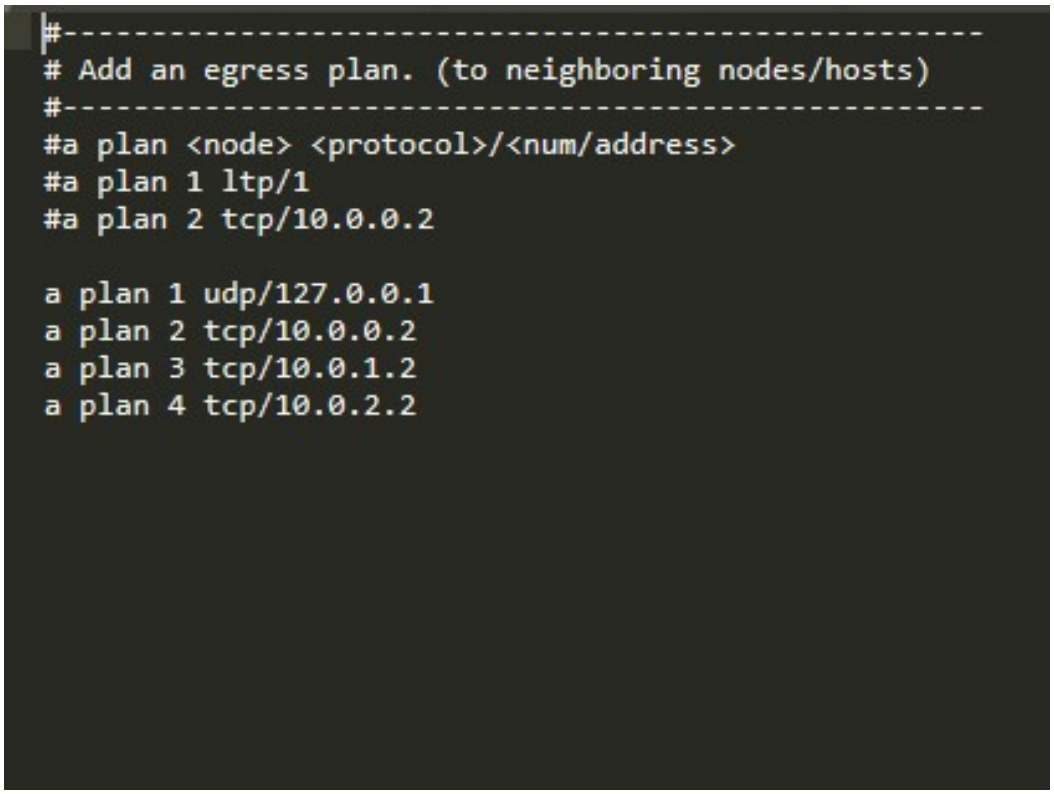


Figura 19: Ficheiro ipnrc

6. ltprc: Ficheiro de configuração do LTP. O comando 1 é seguido do número máximo de sessões suportadas bem como é possível indicar que o nó envia constantemente mensagens para um determinado nó e por último é possível indicar um nó para o qual este nó está sempre á escuta de mensagens. A Figura 20 mostra a configuração neste ficheiro.

```
#Initialization command (command 1).
#Establish the LTP retransmission window.
#A maximum of 64 sessions. 1 session ~ 1 second of transmission
#Set a block size limit of 1000000 bytes. (approx data sent per session)
####1 64 1000000
1 100

#------------------------------------------------------------------
#Add a span (a connection)
#       peer_engine_nbr
#       max_export_sessions
#       max_import_sessions
#       max_segment_size
#       aggregation_size_limit
#       aggregation_time_limit
#       LSO_command
#      [queuing_latency]

#------------------------------------------------------------------
# LTP Spans
#a span <num> 100 100 64000 100000 1 'udplso x.x.x.x:1113 40000000'

#------------------------------------------------------------------
# Listener on 0.0.0.0
s 'udplsi 0.0.0.0:1113'

w
```

Figura 20: Ficheiro ltprc

Para a execução de todos estes testes, os ficheiros foram configurados apenas uma vez porque o cenário não se altera. A única diferença entre os testes é a configuração do ficheiro de ionsecrc que contém a definição dos mecanismos de segurança do Bundle Protocol. No primeiro conjunto de testes o ficheiro ionsecrc tem o seguinte formato:

1. ”1“ - comando de inicialização;
2. ”e 1” - comando que indica que os logs vão ser registados.

Na execução dos testes com os mecanismos de segurança o ficheiro tem campos adicionais, no caso vai ser introduzida uma chave em modo de ficheiro e também a identificação do mecanismo de segurança utilizado. Os ficheiro tem o seguinte formato:

1. “1” - comando de inicialização;
2. “e 1“ - comando que indica que os logs vão ser registados;
3. ”a key chave11 chave.txt” - comando que adiciona uma chave;
4. “a a bspbcbrule ipn:1.* ipn:6.* 1 ’BCB-SHA256-AES128’ chave11” - comando que indica o mecanismo de segurança utilizado, neste exemplo o BSP, os nós

que vão ser afetados, neste caso, indica que vão ser todos os endpoints do nó 1 e no caminho para o Nó 6. Também é utilizado o algoritmo de encriptação BCB-SHA256-AES128 e a chave chave11. Esta chave tem um tamanho de 16 bytes ou 128 bits. Caso o tamanho fosse diferente a regra criada neste exemplo para verificar a confidencialidade do Bundle não seria possível de introduzir pois para este mecanismo é obrigatório que a chave tenha 128 bits.

4.3 FERRAMENTAS PARA CONSTRUÇÃO DE TESTES E EXTRAÇÃO DE RESULTADOS

Para a realização deste trabalho, foram testadas enumeras ferramentas para a obtenção e extração de resultados. Foram testadas ferramentas como wireshark, dtnPerf entre outras para fazer a leitura dos dados transmitidos pela rede, sendo que no final e após alguma análise, foi escolhida a ferramenta Wireshark devido ao facto de trazer mais fiabilidade aos resultados pretendidos.

Além da extração de dados na rede foi necessária encontrar uma ferramenta para a criação do cenário de testes e par tal foi utilizada uma máquina virtual configurada com a implementação ION, a qual continha o simulador de testes utilizado. A Figura 21 mostra a máquina virtual utilizada.

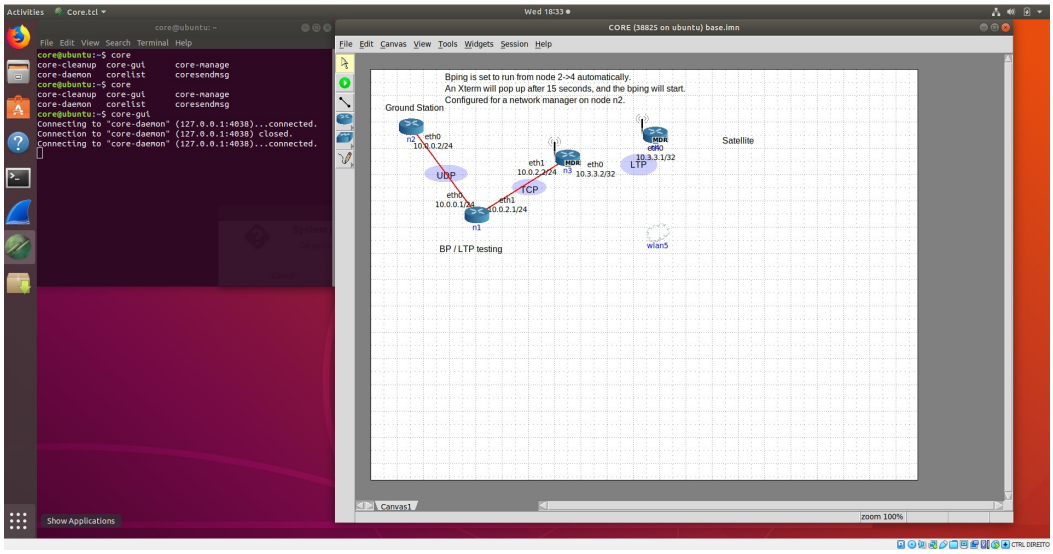


Figura 21: Máquina Virtual

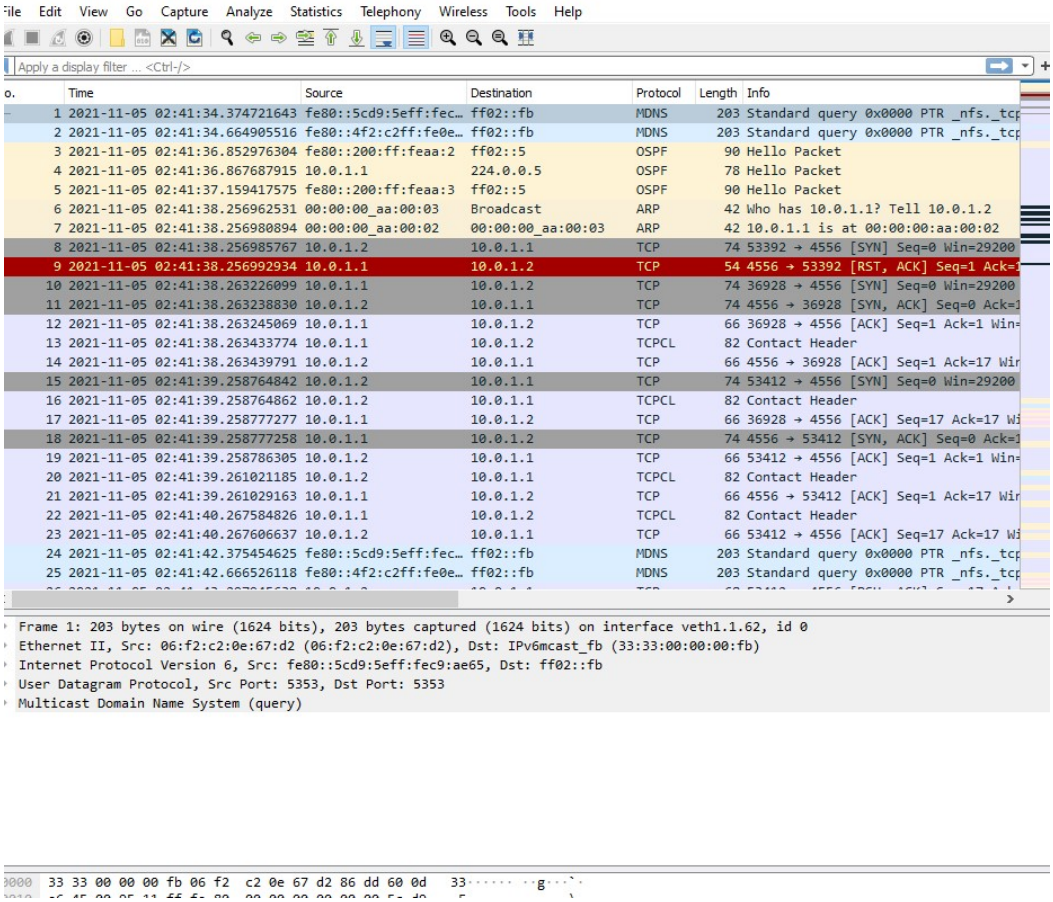


Figura 22: Wireshark

4.3.1 *Montagem dos componentes do cenário de testes*

O simulador permite a montagem do cenário de forma a que seja possível a sua visualização e a definição da posição relativa entre os nós. A montagem do cenário está definida num ficheiro denominado ”Nome do cenário“.imn. Este ficheiro tem a identificação de todos os nós bem como as ligações estabelecidas entre os nós. Depois disso é necessário configurar cada um dos nós, bem como criar o ficheiro que vai permitir que os nós móveis se desloquem durante o decorrer dos testes.

4.3.2 *Extração de resultados*

Quando os testes estão a ser executados, é possível abrir uma janela de Wireshark em cada um dos nós do cenário e recolher todos os dados trocados entre os nós para posterior análise dos resultados.

4.4 INTERPRETAÇÃO DE RESULTADOS

Nesta secção são apresentados todos os testes realizados e comentados os resultados obtidos.

4.4.1 *testes sem mecanismos de segurança*

ENVIO CONSTANTE DE PACOTES DE 64 BYTES

Este teste foi utilizado para testar a performance da rede quando são enviadas várias unidades de dados de pequenas dimensões, neste caso 64 bytes constantemente sem a influência de mecanismos de seguranças. O teste teve a duração de 240 segundos que corresponde a 2 voltas do cenário. Os resultados obtidos estão apresentados nos gráficos das figuras 23 e 24.

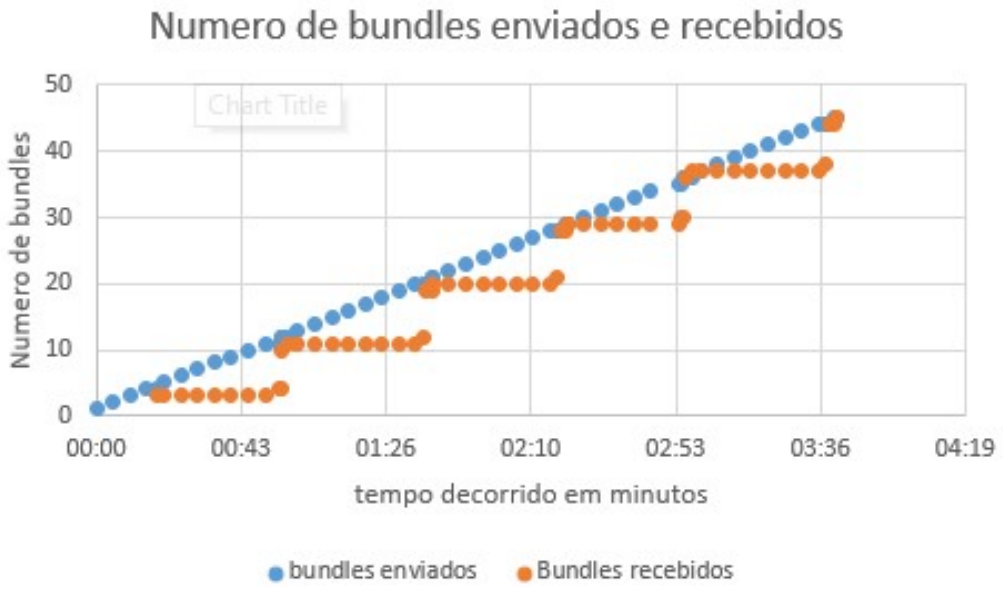


Figura 23: Testes sem segurança - Bundles enviados e recebidos

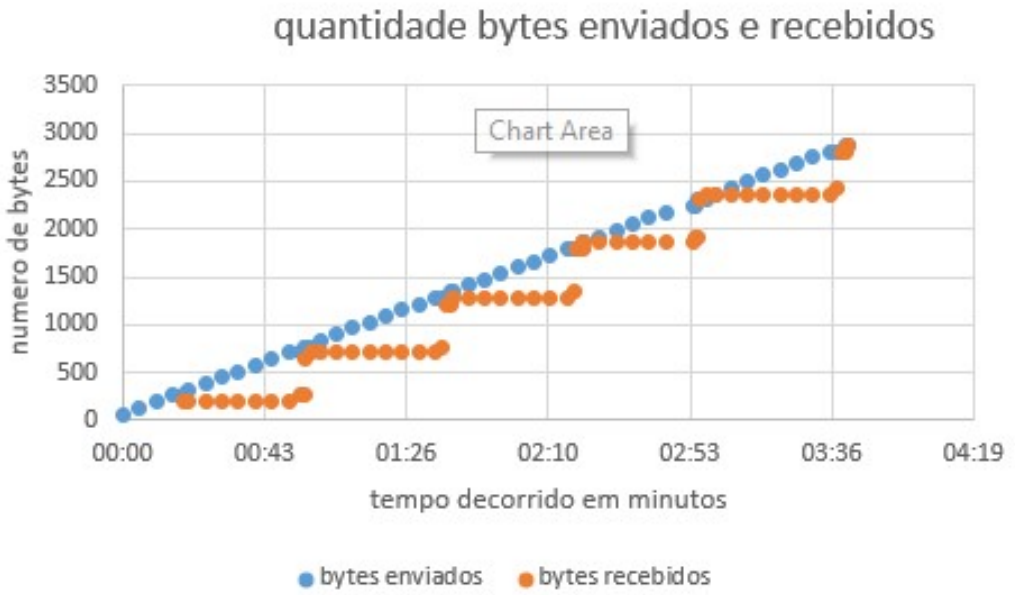


Figura 24: Testes sem segurança - bytes enviados e recebidos

No primeiro gráfico a sequência de pontos a azul representa os bundles que foram criados e enviados a partir do nó 1 para os nós intermédios e a sequência de pontos a laranja representa os bundles que chegam ao nó 6. Até á chegada dos primeiros bundles ao nó 6 foram criados e enviados 4 bundles no nó 1. Estes bundles foram entregues e depois de perder o contacto com o primeiro nó intermédio foram criados mais bundles. Entre a segunda e a 3ª entrega foram enviados 8 bundles tendo

no último intervalo sido enviados 9. No total do cenário de 240 minutos foram enviados 45 bundles e a receção do mesmo número, o que equivale a uma taxa de aproveitamento de 100%. Visto que os dados enviados foram de pequena dimensão, não houve necessidade de criar bundles extra para enviar a totalidade dos dados, desse modo e analisando a Figura 24 podemos concluir que foram enviados um total de 2880 bytes tendo cada entrega demorado cerca de 40 segundos a ser realizada.

ENVIO DE FICHEIROS DE TEXTO DE 500 KB

Este teste foi utilizado para testar a performance da rede quando é enviado um ficheiro de texto com 500 Kbytes sem a influência de mecanismos de segurança. O teste teve a duração de 120 segundos, o que corresponde a 1 volta no cenário. Para a execução deste teste foram enviados 2 ficheiros com 500Kb para simular o envio de ficheiros entre a estação terrestre e a base em Marte. Os resultados obtidos estão apresentados nos gráficos das figuras 25 e 26.

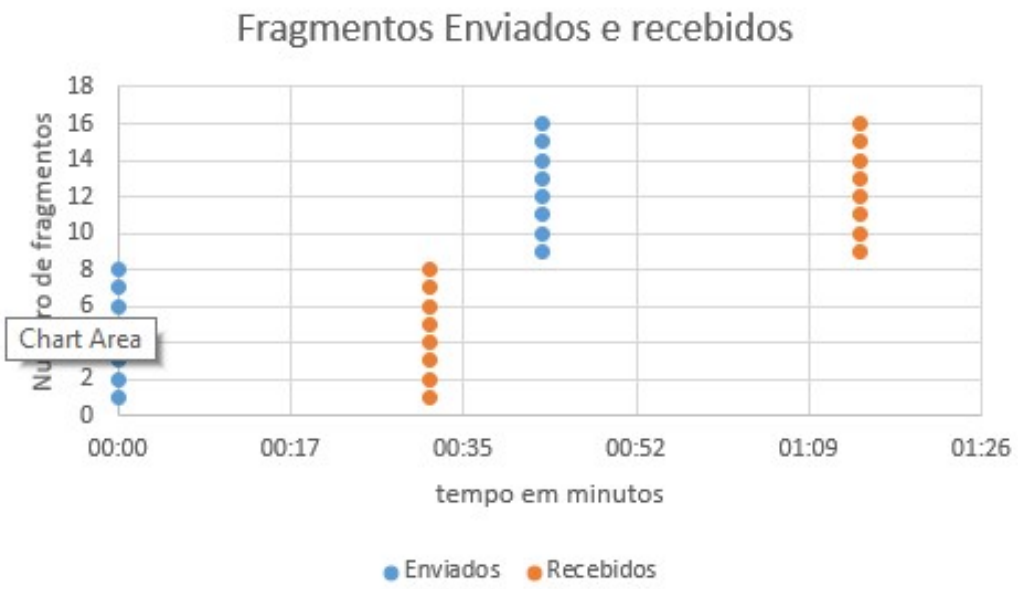


Figura 25: Testes sem segurança - fragmentos enviados e recebidos

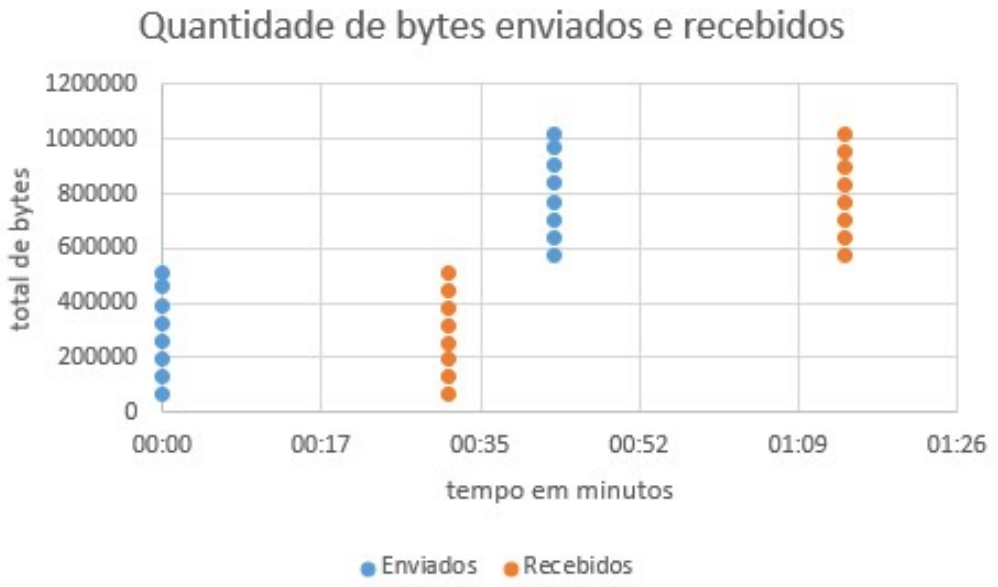


Figura 26: Testes sem segurança - bytes enviados e recebidos

Os ficheiros foram enviados em apenas 1 Bundle cada o que faz com que o Bundle tenha 500Kb o que é impossível de ser enviado por completo em apenas uma unidade de dados. Para tal o Bundle é fragmentado em varias unidades de dados que permite o seu envio e a que chamamos fragmentos. Os pontos azuis referem-se aos fragmentos que foram enviados a partir do nó 1 e os fragmentos a laranja referem-se aos fragmentos que foram recebidos no nó 6. A partir dos gráficos podemos concluir que o Bundle de 500Kb foi repartido em 8 fragmentos sendo que cada um deles tem um tamanho de 65536 bytes cada, tendo o último uma quantidade menor. O envio

destes ficheiros teve a duração de cerca de trinta segundos. Relativamente aos bytes recebidos podemos concluir que chegaram 1000Kb o que corresponde ao dobro do tamanho do ficheiro, ou seja ao tamanho de dois ficheiros, no gráfico é apresentado um valor superior a 1000kb o que diz respeito ao tamanho do cabeçalho do Bundle.

ENVIO DE FICHEIROS DE TEXTO DE 1000KB

Este teste foi utilizado para testar a performance da rede quando é enviado um ficheiro de 1000 Kbytes sem a influência de mecanismos de segurança e com a duração de 120 segundos, o que corresponde a 1 volta do cenário. Os resultados obtidos estão apresentados nos gráficos das figuras 27 e 28.

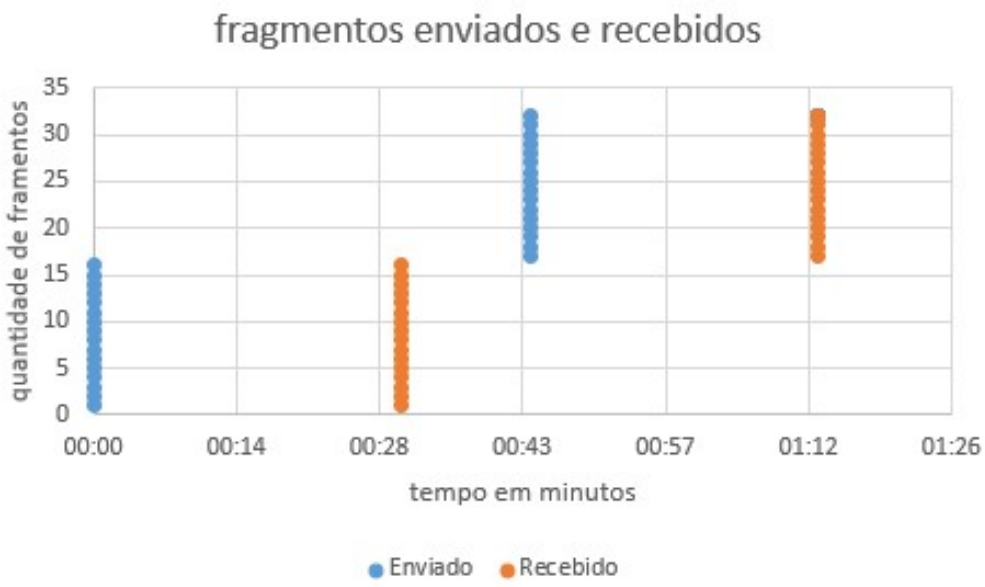


Figura 27: Testes sem segurança- fragmentos enviados e recebidos

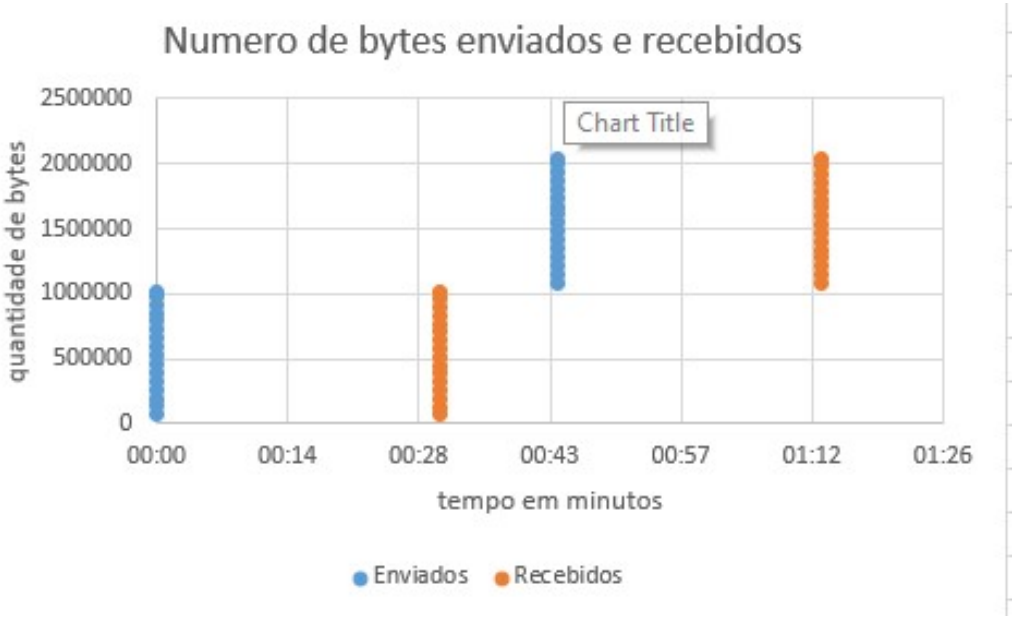


Figura 28: Testes sem segurança- bytes enviados e recebidos

Os ficheiros foram enviados em apenas 1 Bundle cada o que faz com que o Bundle tenha 1000Kb. O Bundle foi fragmentado em várias unidades de dados para permitir o seu envio e a que chamamos fragmentos. Os pontos azuis referem os fragmentos que foram enviados a partir do nó 1 e os fragmentos a laranja referem aos fragmentos que foram recebidos no nó 6. Podemos concluir que o Bundle de 1000Kb foi repartido em 16 fragmentos sendo que cada um deles tem um tamanho de 65536 bytes tendo o último uma quantidade menor. para proceder ao envio destes ficheiros foi necessário

cerca de trinta segundos. Relativamente aos bytes podemos concluir que foram recebidos 2000Kb o que corresponde ao tamanho de dois ficheiros. No gráfico é apresentado um valor superior a 2000kb o que diz respeito também ao tamanho do cabeçalho do Bundle.

ENVIO DE FICHEIRO DE VÍDEO DE MARTE PARA A ESTAÇÃO TERRESTRE

Este teste foi utilizado para testar a performance da rede quando é enviado um ficheiro de vídeo de 2.3MB a partir do nó 6 (o nó presente na superfície do planeta Marte). Neste teste pretendeu-se simular um cenário correspondente as envio de imagens de reconhecimento para a estação terrestre. O teste foi realizado sem a influência de mecanismos de segurança e teve a duração de 240 segundos para que fosse possível o envio na totalidade do vídeo. Os resultados obtidos estão apresentados nos gráficos das figuras 29 e 30.

Figura 29: Testes sem segurança- fragmentos enviados e recebidos

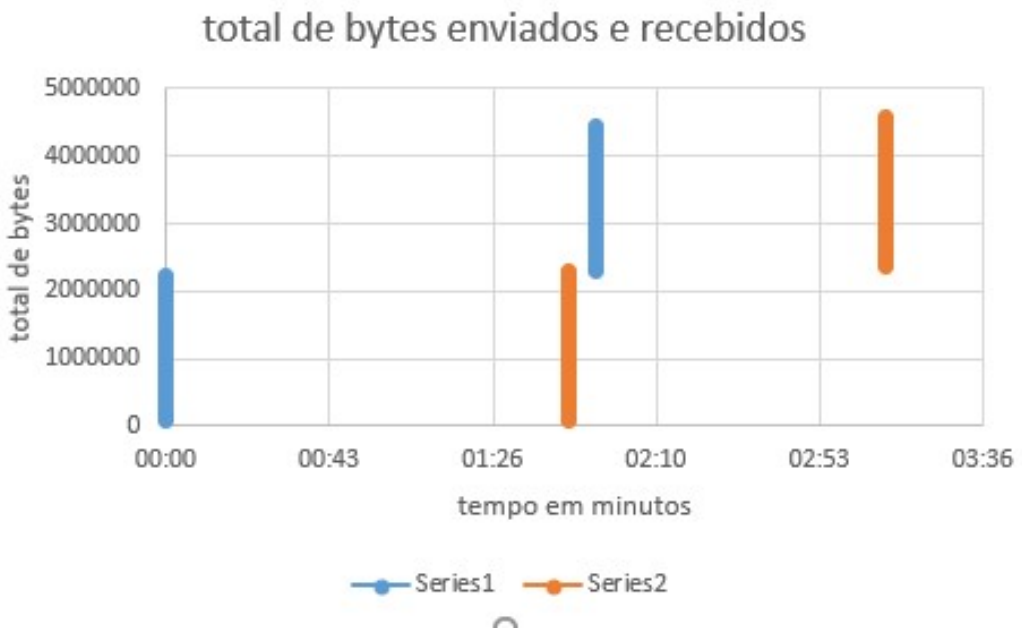


Figura 30: Testes sem segurança- bytes enviados e recebidos

O ficheiro de vídeo foi entregue no nó 1 ao fim de um minuto e 46 segundos. De seguida foi enviado novamente o ficheiro de vídeo para se verificar novamente o tempo que este demoraria a chegar. Verificou-se que enviando ao 1 minuto e 54 segundos do tempo decorrido, chegaria aos 3 minutos e 10 segundos o que faz com que o tempo de demora de entrega fosse de 85 segundos, ou seja, 1 minuto e 25 segundos. Sendo assim, no segundo teste o tempo médio de envio do ficheiro de vídeo foi de 1 minuto e 36 segundos. relativamente ao segundo gráfico pode-se confirmar que os dados enviados foram todos recebidos. Como foram enviados 2 ficheiros de 2.3 MBytes o valor total foi de 4.6 MBytes.

4.4.2 *Testes com mecanismos de segurança*

Nesta seção são apresentados os testes realizados com os diversos mecanismos de segurança.

4.4.2.1 *PIB*

ENVIO CONSTANTE DE SINAIS DE 64 BYTES

Este teste foi utilizado para testar a performance da rede quando são enviadas várias unidades de dados de pequenas dimensões, de 64 bytes, sobre a influência do

mecanismo de segurança PIB, durante 240 segundos que corresponde a 2 voltas do cenário. Para a configuração do PIB foi gerada uma chave de 128 bits, necessária para a execução do mecanismo de segurança bem como a configuração do ficheiro de segurança ionserc nos nós envolvidos. O nó 1 é o emissor de Bundles, o nó 6 é o recetor de Bundles.

Os resultados obtidos estão apresentados nos gráficos das figuras 31 e 32. É importante referir que no primeiro gráfico a sequência de pontos a azul representa os Bundles que foram criados e enviados a partir do nó 1 para os nós intermédios e a sequência de pontos a laranja representa os Bundles que chegam ao nó 6.

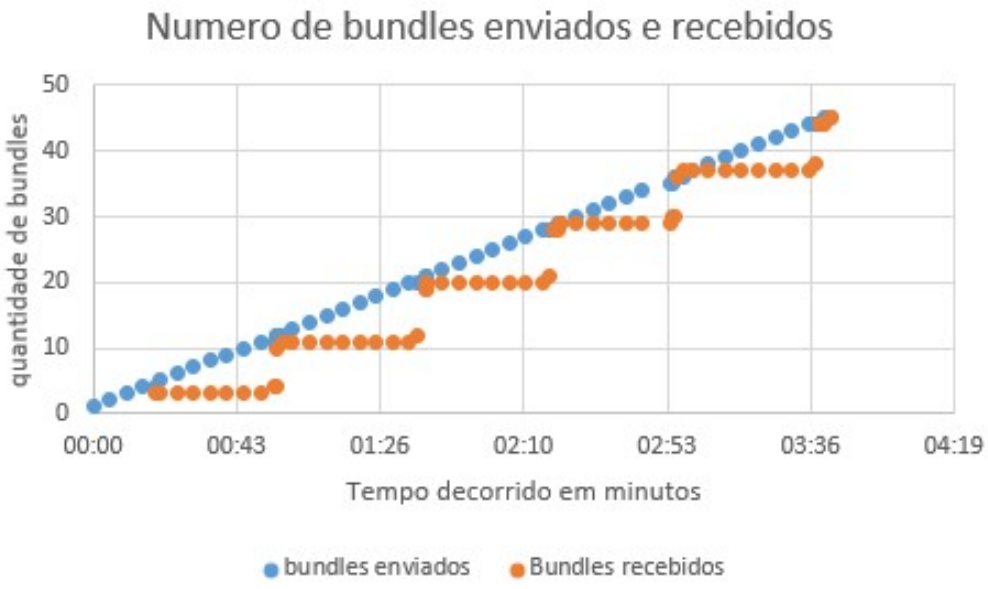


Figura 31: Bundles enviados e recebidos

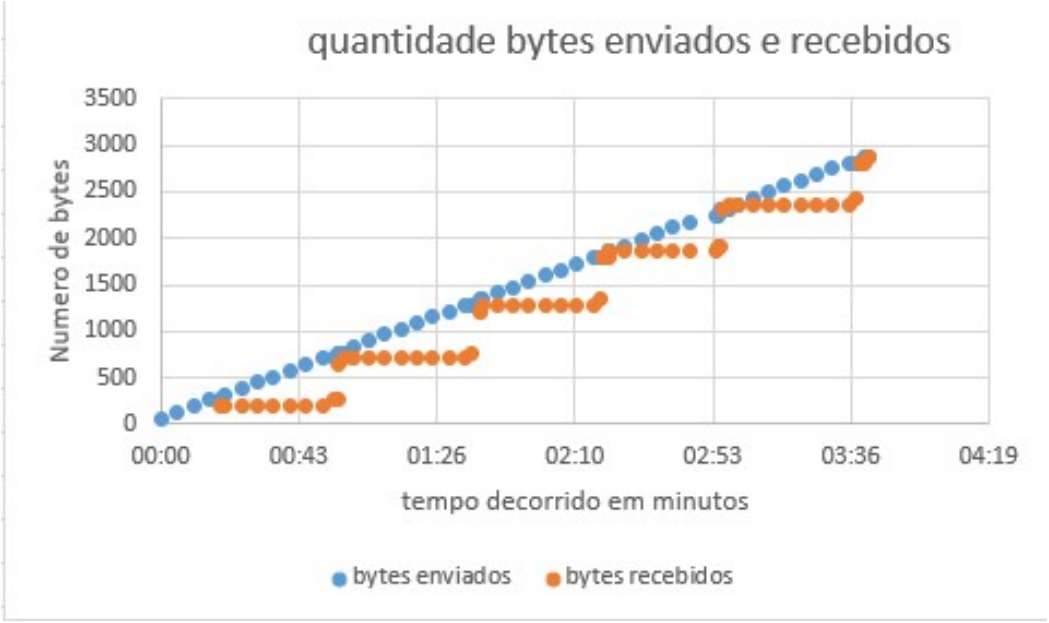


Figura 32: Bytes enviados e recebidos

Podemos concluir que em 240 segundos foram enviados em streams constante 45 Bundles de 64 bytes representado no gráfico pela sequência de pontos a azul. Todos os Bundles foram recebidos no nó 6. Podemos ainda concluir que os Bundles chegaram ao nó 6 40 segundos depois do envio, com isto podemos concluir que neste cenário para estas condições o tempo médio de trânsito não se alterou.

ENVIO DE FICHEIROS DE TEXTO DE 500 KBYTES

Este teste foi utilizado para testar a performance da rede quando é enviado um ficheiro de testo de 500 Kbytes sobre a influência do mecanismo de segurança PIB. O teste teve a duração de 120 segundos que corresponde a 1 volta do cenário. Para a execução deste teste foram enviados 2 ficheiros com 500kb cada para simular o envio de ficheiros entre a estação terrestre e a base em Marte, para analisar a performance da rede com um envio de um ficheiro de pequena ou média dimensão. Os resultados obtidos estão apresentados nos gráficos das figuras 33 e 34.

Figura 33: Bundles enviados e recebidos

Figura 34: Bytes enviados e recebidos

Os ficheiros foram enviados em apenas 1 Bundle cada o que faz com que o Bundle neste caso tenha 500kb. O Bundle foi fragmentado em varias unidades de dados. Nos gráficos, os pontos a azul referem aos fragmentos que foram enviados a partir do nó 1 e os fragmentos a laranja referem aos fragmentos que foram recebidos no nó 6. Analisando o primeiro gráfico concluísse que o nó de receção (n6) recebeu mais um fragmento do que os enviados pelo nó 1. Esta situação deve-se ao facto de no LTP os fragmentos terem um tamanho máximo menor e isso faz com que os

fragmentos enviados do nó 1 para os seus pontos intermédios, em que é utilizado o TCP sejam em menor número. Pode-se também concluir que neste teste o primeiro ficheiro chegou em 31 segundos após o envio e o segundo ficheiro chegou em 32 segundos após ser enviado. Relativamente ao segundo gráfico podemos concluir que os dois ficheiros chegaram na totalidade visto que o total de dados recebido foi de 1 MByte. Também podemos verificar que o envio teve mais 24 bytes do que o teste sem o mecanismo de segurança.

ENVIO DE FICHEIROS DE TEXTO DE 1000 KBYTES

Este teste foi utilizado para testar a performance da rede quando é enviado um ficheiro de 1 Mbytes sobre a influência do mecanismo de segurança PIB. O teste teve a duração de 120 segundos que corresponde a 1 volta do cenário. Para a execução deste teste foram enviados dois ficheiros com 1Mb cada para simular o envio de ficheiros entre a estação terrestre e a base em Marte, para analisar a performance da rede com um envio de um ficheiro de pequena ou média dimensão. Os resultados obtidos estão apresentados nos gráficos das figuras 35 e 36.

Figura 35: Bundles enviados e recebidos

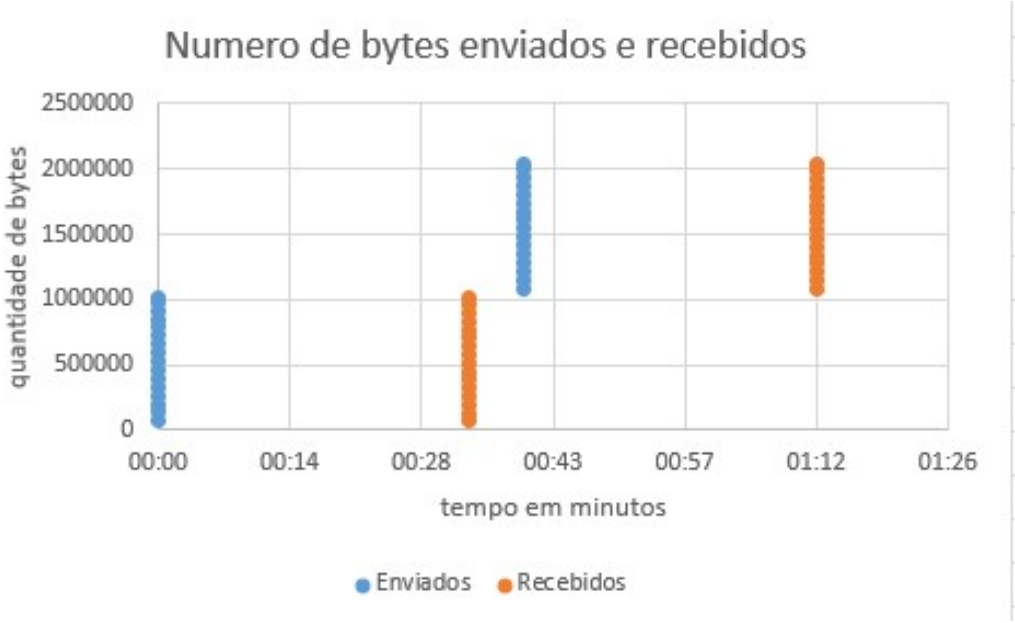


Figura 36: Bytes enviados e recebidos

Os ficheiros foram enviados em apenas 1 Bundle cada o que faz com que o Bundle neste caso tenha 1000kb. O Bundle foi fragmentado em várias unidades de dados. Nos gráficos, os pontos a azul referem aos fragmentos que foram enviados a partir do nó 1 e os fragmentos a laranja referem aos fragmentos que foram recebidos no nó 6.

Analisando o primeiro gráfico podemos concluir que como no teste anterior os fragmentos de entrega são mais do que os fragmentos de envio isto mais uma vez se deve ao facto dos fragmentos enviados pelo LTP serem em menor número do que os fragmentos enviados pelo TCP. Podemos verificar que a até à receção do ficheiro passaram 34 segundos, no segundo caso o ficheiro chegou em 33 segundos.

ENVIO DE FICHEIRO DE VÍDEO PARA A ESTAÇÃO NA TERRA

Este teste foi utilizado para testar a performance da rede quando é enviado um ficheiro de vídeo de 2.3 MBytes sendo que o nó 6 (o nó presente na superfície do planeta Marte) é o emissor. Com isto é previsto simular um cenário real em que são enviadas as imagens de reconhecimento para a estação terrestre com a influência do mecanismo de segurança PIB. O teste teve a duração de 4 minutos para que fosse possível o envio na totalidade do ficheiro de vídeo. Os resultados obtidos estão apresentados nos gráficos das figuras 37 e 38.

Figura 37: Bundles enviados e recebidos

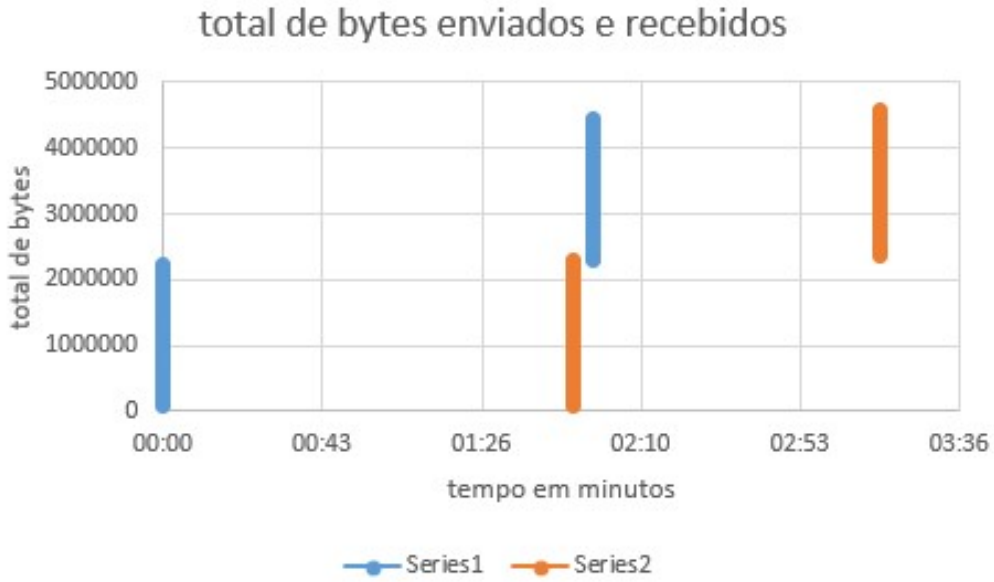


Figura 38: Bytes enviados e recebidos

No primeiro gráfico pudemos concluir que o tempo necessário para se proceder ao envio e receção do ficheiro de 2.3 MBytes foi de 1 minutos e 51 segundos. O tempo necessário para se proceder ao envio e receção do segundo ficheiro foi de 1 minuto e 18 segundos. Também se verifica que foram recebido mais fragmentos do que os que foram enviados inicialmente, o que se deve ao facto de os fragmentos rececionados pelo nó 6 terem um tamanho menor do que inicialmente foram enviados. No caso do envio os fragmentos têm cerca de 65000 bytes e na receção os mesmo

têm pouco mais de 63000 bytes. Neste teste o ficheiro de vídeo é fragmentado para permitir o envio de toda a informação de uma única vez num único Bundle. Os fragmentos são enviados como pacotes pela camada física e quando chegam á camada do Bundle Protocol são novamente agrupados num Bundle. No segundo gráfico podemos verificar que o número de bytes recebidos e enviados é igual. Em suma podemos concluir que com este mecanismo de segurança tempo de envio de ficheiros de 2MB pode ser elevado e que devido á introdução deste mecanismo pode ser necessário criar mais fragmentos do Bundle.

4.4.2.2 *PCB*

ENVIO CONSTANTE DE SINAIS DE 64 BYTES

Este teste foi utilizado para testar a performance da rede quando são enviadas várias unidades de dados de pequenas dimensões, de 64 bytes, sobre a influência do mecanismo de segurança PCB, durante 240 segundos que corresponde a 2 voltas do cenário. Para a configuração do PCB foi gerada uma chave de 128 bits, necessária para a execução do mecanismo de segurança bem como a configuração do ficheiro de segurança ionserc nos nós envolvidos. O nó 1 é o emissor de Bundles, o nó 6 é o recetor de bundles. Os resultados obtidos estão apresentados nos gráficos das figuras 39 e 40. É importante referir que no primeiro gráfico a sequência de pontos a azul representa os Bundles que foram criados e enviados a partir do nó 1 para os nós intermédios e a sequência de pontos a laranja representa os Bundles que chegam ao nó 6.

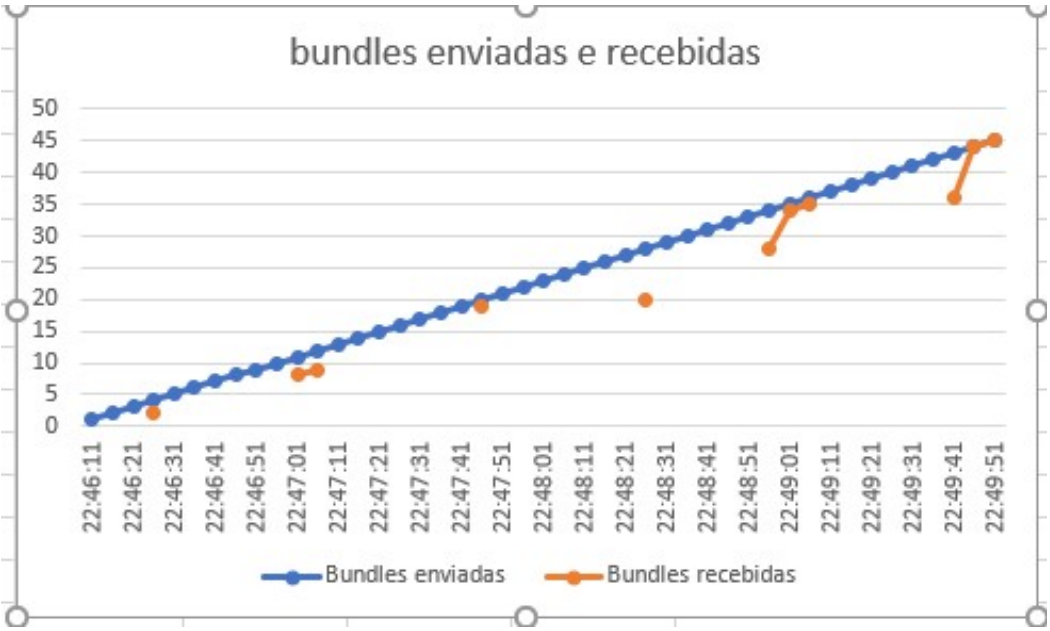


Figura 39: Bundles enviados e recebidos

Figura 40: Bytes enviados e recebidos

Analisando o primeiro gráfico podemos concluir que em 240 segundos foram enviados em Stream constante 45 bundles de 64 bytes representado no gráfico pela sequência de pontos a azul, tendo todos sido recebidos no nó 6. No segundo gráfico podemos concluir que foram enviados e recebidos 2880 bytes durante todo o processo. Podemos verificar que os Bundles foram enviados do nó 1 constantemente para os nós intermédios, mas que só chegaram ao nó 6 40 segundos depois, com isto

podemos concluir que neste cenário para estas condições o tempo médio de envio e chegada de um Bundle de 64 bytes foi de 40 segundos.

ENVIO DE FICHEIROS DE TEXTO DE 500 KBYTES

Este teste foi utilizado para testar a performance da rede quando é enviado um ficheiro de texto de 500 Kbytes sobre a influência do mecanismo de segurança PCB. O teste teve a duração de 120 segundos que corresponde a 1 volta do cenário. Para a execução deste teste foram enviados 2 ficheiros com 500kb cada para simular o envio de ficheiros entre a estação terrestre e a base em Marte, para analisar a performance da rede com um envio de um ficheiro de pequena ou média dimensão. Os resultados obtidos estão apresentados nos gráficos das figuras 41 e 42.

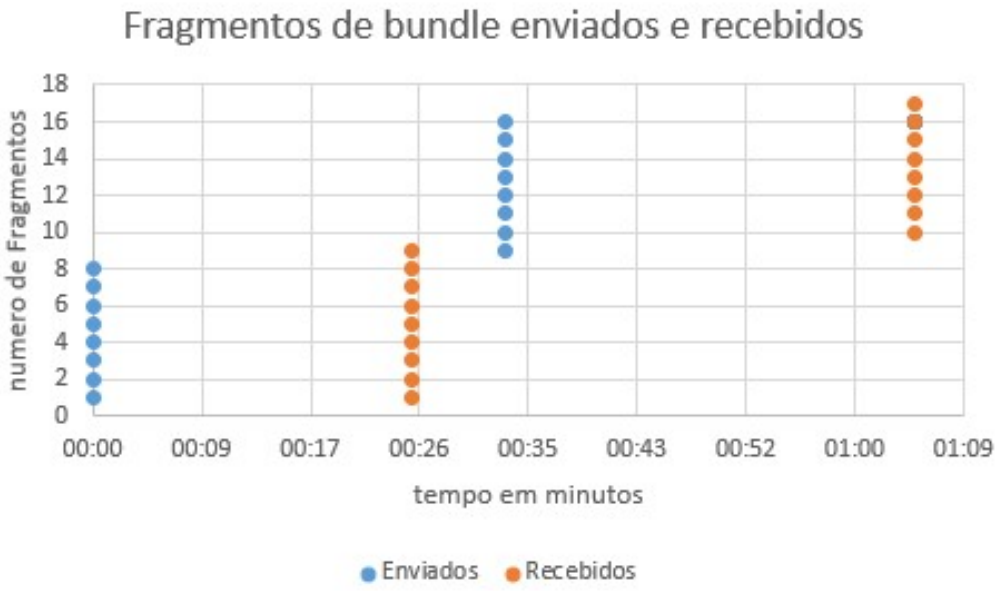


Figura 41: fragmentos enviados e recebidos

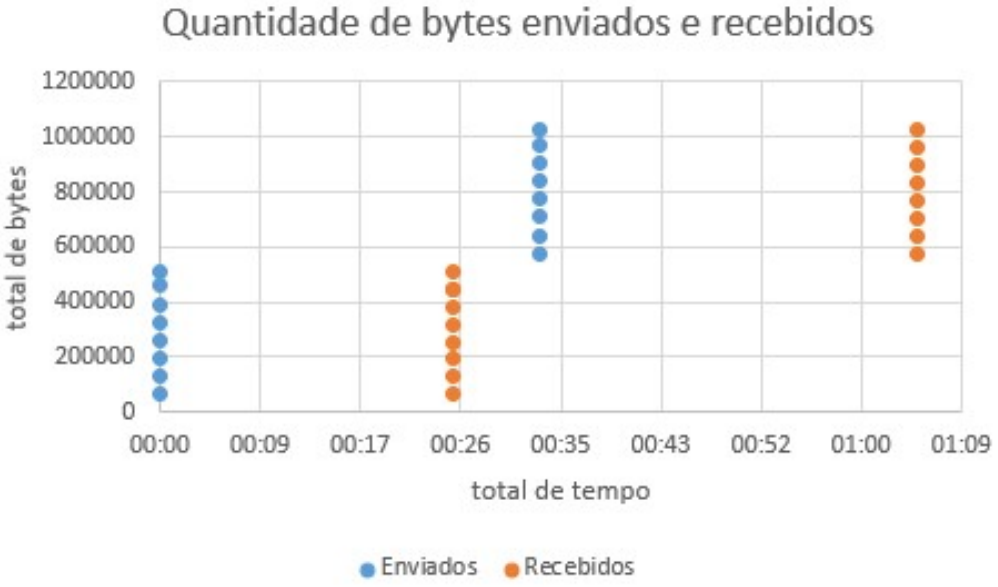


Figura 42: bytes enviados e recebidos

Os ficheiros foram enviados em apenas 1 Bundle cada o que faz com que o Bundle neste caso tenha 500kb. O Bundle foi fragmentado em varias unidades de dados. Nos gráficos, os pontos a azul referem aos fragmentos que foram enviados a partir do nó 1 e os fragmentos a laranja referem aos fragmentos que foram recebidos no nó 6. Analisando o primeiro gráfico concluísse que o nó de receção (n6) recebeu mais um fragmento do que os enviados pelo nó 1. Esta situação deve-se ao facto de no LTP os fragmentos terem um tamanho máximo menor e isso faz com que os fragmentos

enviados do nó 1 para os seus pontos intermédios, em que é utilizado o TCP sejam em menor número. Pode-se também concluir que neste teste o primeiro ficheiro chegou em 26 segundos após o envio e o 2 ficheiro chegou em 29 segundos após ser enviado. Relativamente ao segundo gráfico podemos concluir que os 2 ficheiros chegaram na totalidade visto que o total de dados recebido foi de 1 MByte.

ENVIO DE FICHEIROS DE TEXTO DE 1000 KBYTES

Este teste foi utilizado para testar a performance da rede quando é enviado um ficheiro de 1 Mbytes sobre a influência do mecanismo de segurança PCB. O teste teve a duração de 120 segundos que corresponde a 1 volta do cenário. Para a execução deste teste foram enviados dois ficheiros com 1Mb cada para simular o envio de ficheiros entre a estação terrestre e a base em Marte, para analisar a performance da rede com um envio de um ficheiro de pequena ou média dimensão. Os resultados obtidos estão apresentados nos gráficos das figuras 43 e 44.

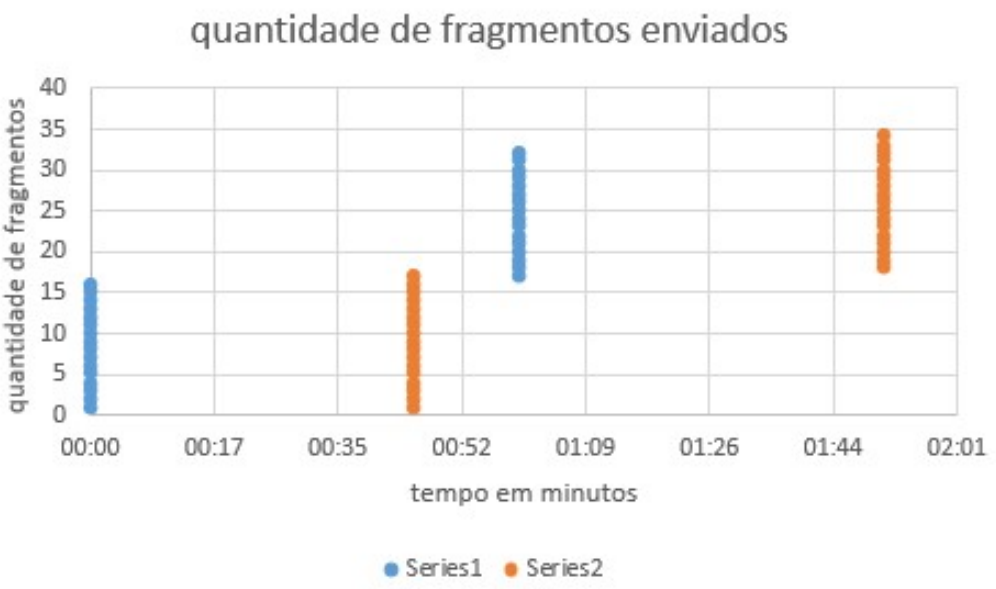


Figura 43: Fragmentos enviados e recebidos

Figura 44: Bytes enviados e recebidos

Os ficheiros foram enviados em apenas 1 Bundle cada, o que faz com que o Bundle neste caso tenha 1000Kb. O Bundle foi fragmentado em várias unidades de dados. Nos gráficos, os pontos a azul referem aos fragmentos que foram enviados a partir do nó 1 e os pontos a laranja referem aos fragmentos que foram recebidos no nó 6. Verifica-se que a receção do primeiro ficheiro demorou 45 segundos a chegar ao destino. O segundo ficheiro demorou 51 segundos durante a transmissão.

ENVIO DE FICHEIRO DE VÍDEO PARA A ESTAÇÃO NO PLANETA TERRA

Este teste foi utilizado para testar a performance da rede quando é enviado um ficheiro de vídeo de 2.3 MBytes sendo que o nó 6 (o nó presente na superfície do planeta Marte) é o emissor. Com isto é previsto simular um cenário real em que são enviadas imagens de reconhecimento para a estação terrestre com a influência do mecanismo de segurança PCB. O teste teve a duração de 4 minutos para que fosse possível o envio na totalidade do ficheiro de vídeo. Os resultados obtidos estão apresentados nos gráficos das figuras 45 e 46.

Figura 45: Fragmentos enviados e recebidos

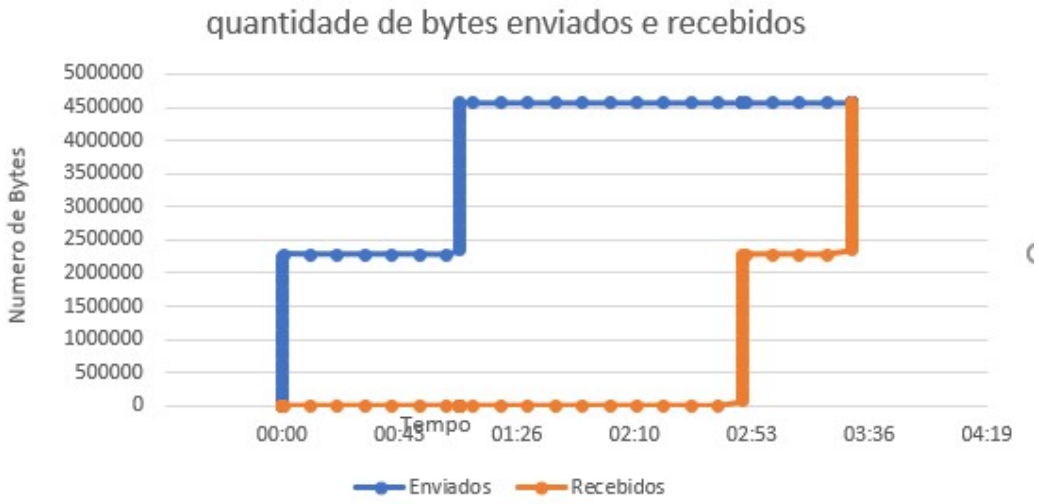


Figura 46: Bytes enviados e recebidos

No primeiro gráfico podemos concluir que o tempo necessário para se proceder ao envio e receção do ficheiros de 2.3 MBytes foi de 2 minutos e 53 segundos. O tempo necessário para se proceder ao envio e receção do segundo ficheiro foi de 3 minutos e 30 segundos. Também se verifica que foram recebido mais fragmentos do que os que foram enviados inicialmente, o que se deve ao facto de os fragmentos rececionados pelo nó 6 terem um tamanho menor do que inicialmente foram enviados. No caso do envio os fragmentos têm cerca de 65000 bytes e na receção os mesmo têm pouco mais de 63000 bytes. Neste teste o ficheiro de vídeo é fragmentado para permitir o envio de toda a informação de uma única vez num único Bundle. Os fragmentos são enviados como pacotes pela camada física e quando chegam á camada do Bundle Protocol são novamente agrupados num Bundle. No segundo gráfico podemos verificar que o número de bytes recebido e enviados é igual. Em

suma podemos concluir que com este mecanismo de segurança o tempo de envio de ficheiros de 2MB pode ser elevado e que devido á introdução deste mecanismo pode ser necessário criar mais fragmentos do Bundle.

4.4.2.3 *PCB+PIB*

ENVIO CONSTANTE DE SINAIS DE 64 BYTES

Este teste foi utilizado para testar a performance da rede quando são enviadas várias unidades de dados de pequenas dimensões, de 64 bytes, sobre a influência dos mecanismos de segurança PCB e PIB em simultâneo, durante 240 segundos que corresponde a 2 voltas do cenário. Para a configuração do PIB foi gerada uma chave de 128 bits, necessária para a execução do mecanismo de segurança bem como a configuração do ficheiro de segurança ionserc nos nós envolvidos. Para a configuração do PCB foram realizados os mesmos procedimentos que para o PIB mas com as especificações do PCB. O nó 1 é o emissor de Bundles, o nó 6 é o recetor de Bundles. Os resultados obtidos estão apresentados nos gráficos das figuras 47 e 48. É importante referir que no primeiro gráfico a sequência de pontos a azul representa os Bundles que foram criados e enviados a partir do nó 1 para os nós intermédios e a sequência de pontos a laranja representa os Bundles que chegam ao nó 6.

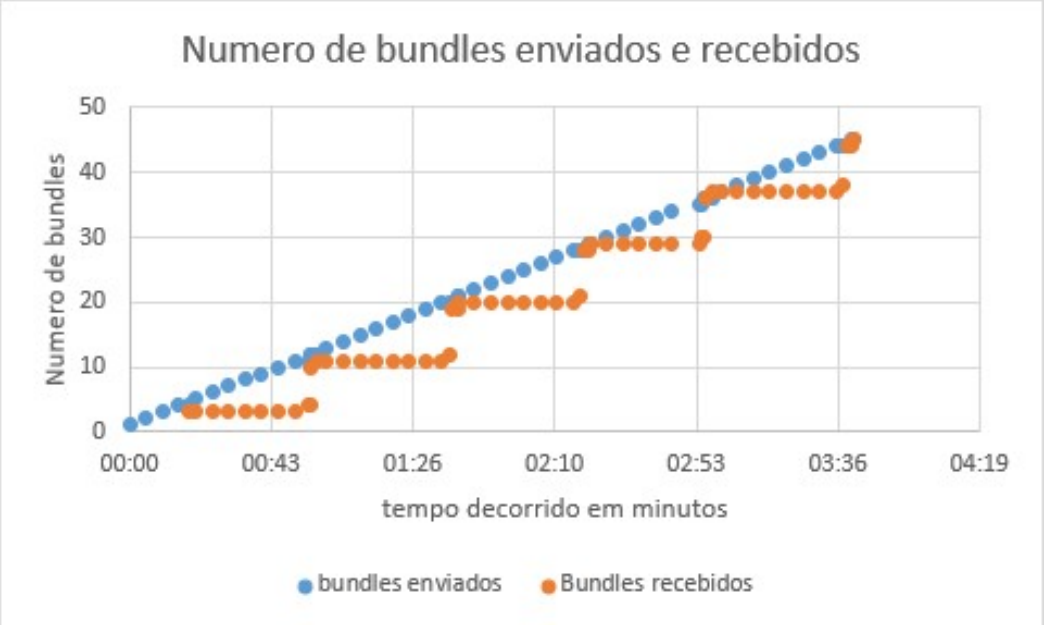


Figura 47: Bundles enviados e recebidos

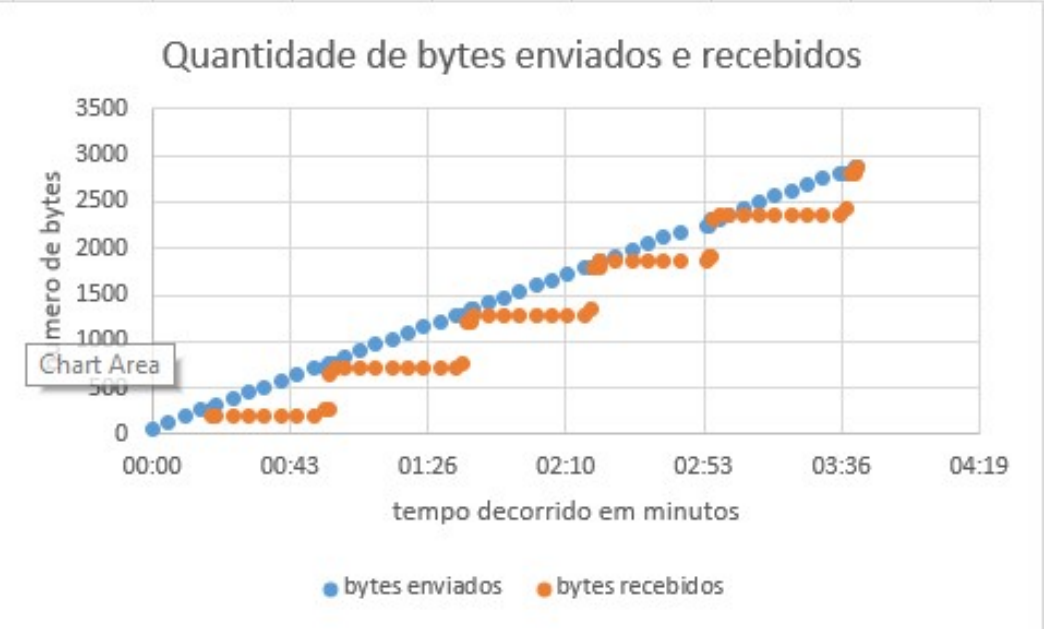


Figura 48: Bytes enviados e recebidos

Analisando o primeiro gráfico podemos concluir que em 240 segundos foram enviados em Stream constante 45 Bundles de 64 bytes representado no gráfico pela sequência de pontos a azul, tendo todos sido recebidos no nó 6. No segundo gráfico podemos concluir que foram enviados e recebidos 2880 bytes durante todo o processo. Podemos verificar que os Bundles foram enviados do nó 1 constantemente para os nós intermédios, mas que só chegaram ao nó 6 40 segundos depois, com isto

podemos concluir que neste cenário para estas condições o tempo médio de envio e chegada de um Bundle de 64 bytes foi de 40 segundos.

ENVIO DE FICHEIROS DE TEXTO DE 500 KB

Este teste foi utilizado para testar a performance da rede quando é enviado um ficheiro, neste caso 500 Kbytes sobre a influência dos mecanismos de segurança PCB e PIB. O teste teve a duração de 120 segundos, o que corresponde a 1 volta do cenário. Foi necessário configurar os mecanismos de segurança. Para o PCB foi gerada uma chave de 16 bytes ou 128 bits necessária á execução do mecanismo de segurança PCB. Para o PIB uma chave de 32 bytes, ou seja, 256 bits bem como a configuração do ficheiro de segurança dos nós envolvidos através dos ficheiros ionserc. O nó 1 foi configurado para o envio de Bundles e o nó 6 como recetor de Bundles. Para a execução deste teste foram enviados 2 ficheiros com 500kb cada um com o objetivo de simular o envio de ficheiros entre a estação terrestre e a base em Marte. Os resultados obtidos estão apresentados nos gráficos das figuras 49 e 50.

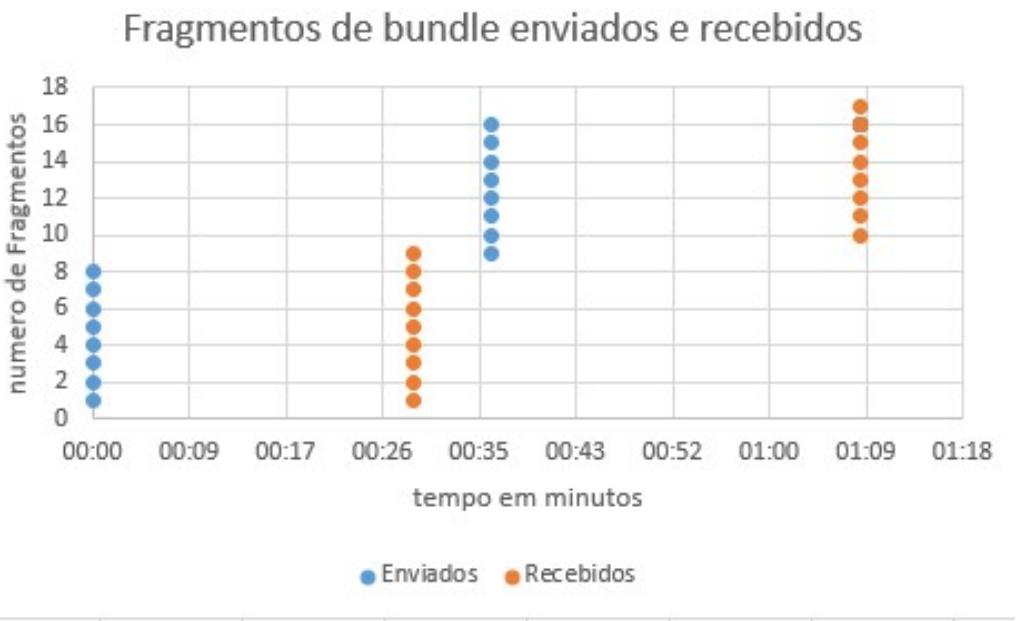


Figura 49: fragmentos enviados e recebidos

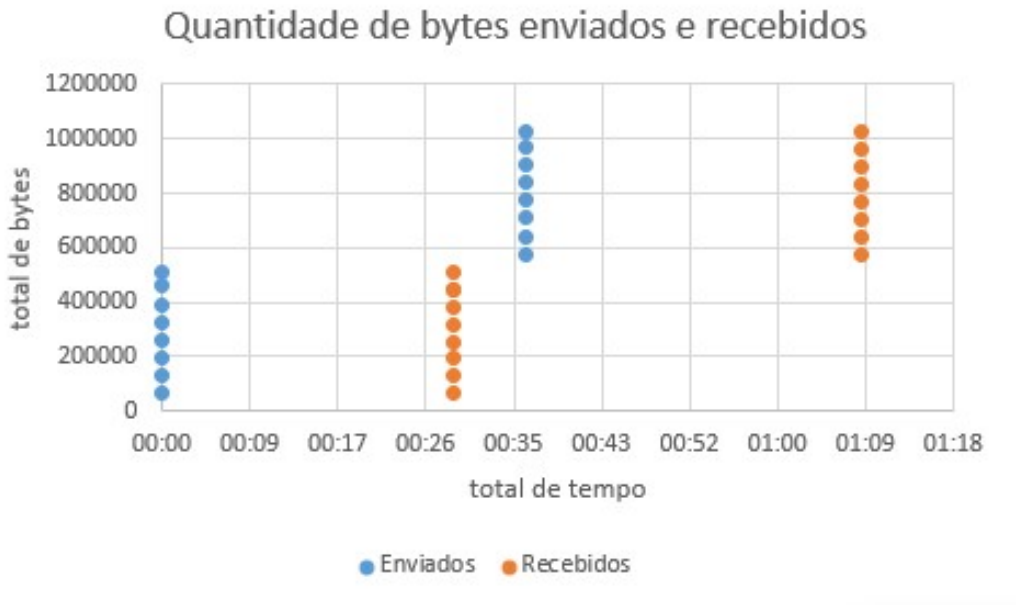


Figura 50: bytes enviados e recebidos

Os ficheiros foram enviados em apenas 1 Bundle cada com 500kb mas fragmentado em várias unidades de dados. Os pontos azuis referem aos fragmentos que foram enviados a partir do nó 1 e os fragmentos a laranja referem aos fragmentos que foram recebidos no nó 6. Da análise ao primeiro gráfico concluísse que o nó de receção nó 6 recebeu mais um fragmento do que foi enviado através do nó 1, isto acontece devido ao facto do LTP impor aos fragmentos um tamanho máximo menor

e isso faz com que os fragmentos enviados do nó 1 para os seus pontos intermédios, realizado por TCP, sejam menos visto que armazenam mais informação.

O primeiro ficheiro chegou em 28 segundos após o envio e o segundo ficheiro chegou a 1 minuto e 8 segundos após ter sido enviado aos 36 segundos o que faz com que o tempo de envio deste segundo ficheiro tenha sido de 32 segundos. Com isto concluímos que o tempo médio de envio neste teste tenha sido de 30 segundos. Relativamente ao segundo gráfico o podemos concluir que os 2 ficheiros chegaram na totalidade visto que o total de dados recebido foi de 1 MB.

ENVIO DE FICHEIROS DE TEXTO DE 1000 KB

Este teste foi utilizado para testar a performance da rede quando é enviado um ficheiro de 1000 Kbytes sobre a influência dos mecanismos de segurança PCB e PIB. O teste teve a duração de 120 segundos, o que corresponde a 1 volta do cenário. Foi necessário configurar os mecanismos de segurança. Para o PCB foi gerada uma chave de 16 bytes ou 128 bits necessária á execução do mecanismo de segurança PCB. Para o PIB uma chave de 32 bytes, ou seja, 256 bits bem como a configuração do ficheiro de segurança dos nós envolvidos através dos ficheiros ionserc. O nó 1 foi configurado para o envio de Bundles e o nó 6 como recetor de Bundles. Para a execução deste teste foram enviados 2 ficheiros com 1000kb cada um com o objetivo de simular o envio de ficheiros entre a estação terrestre e a base em Marte. Os resultados obtidos estão apresentados nos gráficos das figuras 51 e 52.

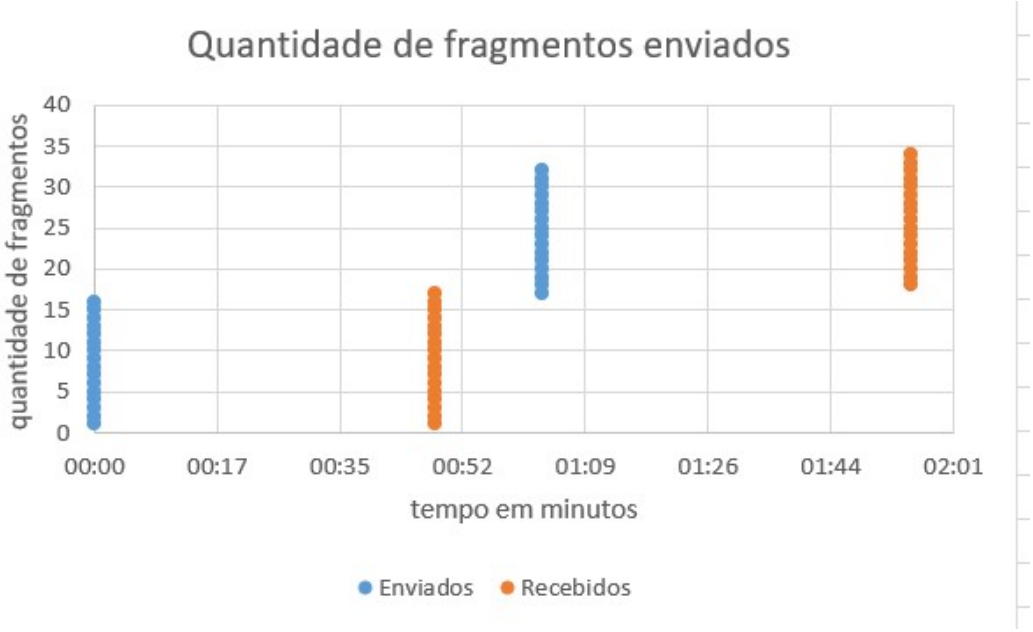


Figura 51: Fragmentos enviados e recebidos

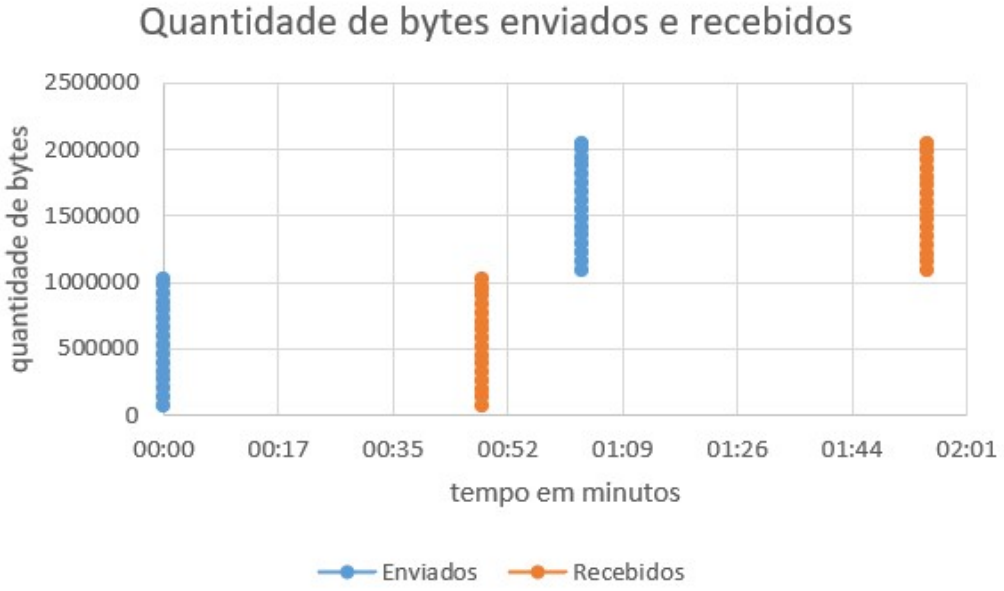


Figura 52: Bytes enviados e recebidos

Analisando o primeiro gráfico podemos concluir que os fragmentos na chegada são mais do que os fragmentos de envio devido ao facto do tamanho dos fragmentos enviados pelo LTP ser menor que os fragmentos enviados pelo TCP. Podemos verificar que a receção do ficheiro demorou 50 segundos. O segundo ficheiro demorou 52 segundos a ser transmitido. Neste teste em média o ficheiro demorou 50 segundos desde o seu envio até à sua chegada.

ENVIO DE FICHEIRO DE VÍDEO PARA A ESTAÇÃO TERRESTRE TERRA

Este teste foi utilizado para testar a performance da rede quando é enviado um ficheiro de vídeo de 2.3MB sendo que o nó 6 é o emissor. Com este teste pretende-se simular o envio de imagens de reconhecimento para a estação terrestre. O teste teve a duração de 7 minutos para que seja possível enviar a totalidade o ficheiro de vídeo. Os mecanismos de segurança configurados foram o PCB e o PIB. Os resultados obtidos estão apresentados nos gráficos das figuras 53 e 54.

Figura 53: Fragmentos enviados e recebidos

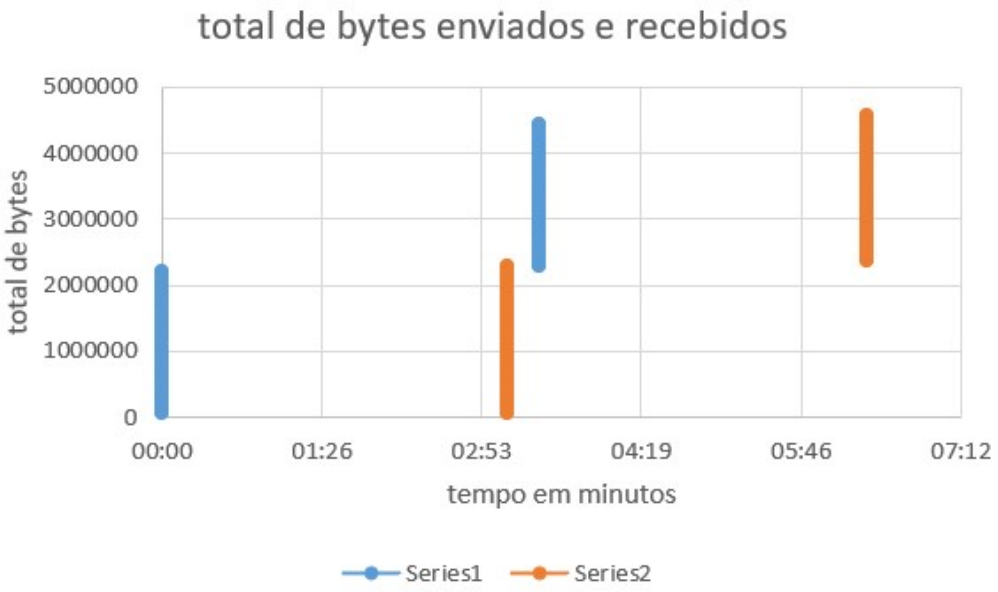


Figura 54: Bytes enviados e recebidos

Do primeiro gráfico pudemos concluir que para se proceder ao envio de ficheiros de 2.3 MB foram necessários 3 minutos e 7 segundos para o ficheiro 1 e 2 minutos e 57 segundos para o segundo ficheiro. Verifica-se ainda que foram recebido mais fragmentos do que os que foram enviados na origem. Os fragmentos são enviados como pacotes pela camada física e quando chegam á camada do Bundle Protocol são novamente agrupados num Bundle. No segundo gráfico podemos verificar que o

número de bytes recebido e enviados são iguais apesar de no gráfico anterior existir uma diferença nos fragmentos recebidos.

4.5 SÍNTESE

Neste capítulo foram detalhados os testes realizados para avaliar a performance da rede DTN com a introdução de mecanismos de segurança fornecidos pelo Bundle Protocol. O cenário desenvolvido foi apresentado, os testes realizados foram explicados bem como que ferramentas utilizadas para a extração e análise dos dados. Por último foram explicados e analisados os resultados dos testes.

5

CONCLUSÕES

Após a realização deste trabalho é possível chegar a algumas conclusões, face ao estudo da arquitetura da rede DTN, da componente de segurança para redes DTN e do resultado dos testes realizados utilizando a implementação ION.

Sobre a arquitetura DTN podemos afirmar que é uma arquitetura de rede bastante relevante, com diferenças significativas face á arquitetura TCP/IP e com uma grande utilidade face ao que se prevê que venha a ser a exploração espacial durante as próximas décadas.

A componente de segurança da arquitetura DTN está definida e possui protocolos específicos para a verificação da integridade e confidencialidade da informação enviada e recebida dentro da rede DTN. Esta componente ainda não está totalmente explorada, ainda existe algum trabalho a realizar como por exemplo a inclusão de soluções contra ataques do tipo Denial of service (DOS) bem como alternativas á utilização do Bundle Security Protocol, tendo sido esta a única solução de segurança encontrada.

Decorrente da análise ao resultado dos testes podemos tirar algumas conclusões sobre os mecanismos de segurança. Estes resultados estão apresentados na Tabela 2, a qual mostra os tempos médios que foram conseguidos através da realização de testes de performance á rede DTN com e sem mecanismos de segurança.

Resultados dos Testes				
	Sem segurança	PIB	PCB	PCB+PIB
Pings 64KB	00:40	00:40	00:40	00:40
Ficheiro de 500KB	00:31	00:31	00:31	00:31
Ficheiro de 1000KB	00:31	00:31	00:48	00:50
Ficheiro de Vídeo	01:31	01:35	02:53	03:02

Tabela 2: Resultados dos testes (valores em min:sec)

Podemos verificar na Tabela 2 os mecanismos de segurança não têm impacto significativo no resultado dos testes quando são enviados Bundles com pequenas quantidades de dados. Para os vários testes realizados, com e sem mecanismos de

segurança, o tempo de transmissão de Bundles de pequenas dimensões (Ping) foi de 40 segundos em todos os testes. Para os vários testes realizados, com e sem mecanismos de segurança, o tempo de transmissão de ficheiros de 500KB foi de 31 segundos em todos os testes.

Os resultados dos testes mostram que a transmissão de ficheiros de 1000KB com a introdução dos mecanismos de segurança já apresentam algum impacto no tempo de transmissão de dados. Este impacto é mais significativo nos mecanismos PCB e PCB+PIB.

Os resultados dos testes mostram que a transmissão de ficheiros de vídeo com a introdução de mecanismos de segurança apresentam um impacto significativo nos tempos de transmissão de dados. Com o mecanismo PIB a diferença é pequena mas com os mecanismos PCB e PCB+PIB o impacto é significativo.

A Figura 55 mostra o impacto do mecanismos de segurança no tempo de transmissão dos diversos tipos de dados utilizados nos testes.

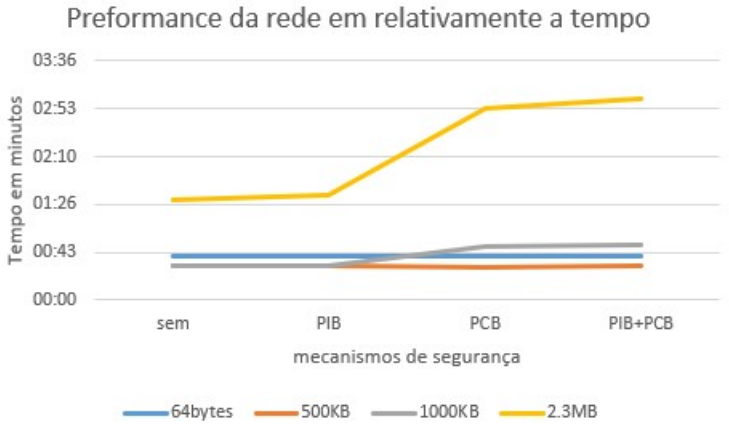


Figura 55: Impacto dos mecanismos de segurança na rede

BIBLIOGRAFIA

Alessi, Nicola (2018). «DTN Performance in Complex Deep-Space Networks». Em.

— (2019). «DTN performance analysis of multi-asset Mars-Earth communications». Em.

Caini, C. (2018). «Security Analysis of DTN Architecture and Bundle Protocol Specification for Space-Based Networks». Em.

CCSDSEPP (2020). «ENCAPSULATION PACKET PROTOCOL». Em: https://public.ccsds.org/Pubs/133x1b3e1.pdf.

CCSDSLTP (2015). «LICKLIDER TRANSMISSION PROTOCOL (LTP) FOR CCSDS». Em: https://public.ccsds.org/Pubs/734x1b1.pdf.

Ivancic, William D. (2010). «Security Analysis of DTN Architecture and Bundle Protocol Specification for Space-Based Networks». Em.

Khadyair, Kawakib (2015). «A Comprehensive Survey on Delay Tolerant Networks». Em: https://www.researchgate.net/publication/309547105_A_Comprehensive_Survey_on_Delay_Tolerant_Networks.

Maathuis, I.J.H. (2003). «The battle between standards: TCP/IP Vs OSI victory through path dependency or by quality?» Em: https://www.researchgate.net/publication/4047835_The_battle_between_standards_TCPIP_Vs_OSI_victory_through_path_dependency_or_by_quality.

NASA (2020). *Disruption Tolerant Networking*. Website. https://www.nasa.gov/archive/content/dtn.

NASA.gov (2006). «Earth Calling». Em.

Ondeng, Tonny Ochieng' (2003). «TCP/IP TECHNOLOGY». Em: https://www.researchgate.net/publication/318960971_TCPIP_TECHNOLOGY.

Ramadas, M. (2008). «Licklider Transmission Protocol - Specification RFC 5326». Em: https://datatracker.ietf.org/doc/html/rfc5326.

Reinhart (2013). «Space Communication and Navigation SDR Testbed, Overview and Opportunity for Experiments». Em: https://pdfs.semanticscholar.org/b845/a8426e43c77763f8da9f9f025a80a73dbf6f.pdf.

Scott, K. (2007). «Bundle Protocol Specification, RFC 5050». Em: https://www.researchgate.net/publication/309547105_A_Comprehensive_Survey_on_Delay_Tolerant_Networks.

Secretariat, CCSDS (2015). «CCSDS BUNDLE PROTOCOL SPECIFICATION». Em: https://public.ccsds.org/Pubs/734x2b1.pdf.

Symington, S. (2011). «Bundle Security Protocol Specification rfc 6257». Em: https://www.rfc-editor.org/rfc/rfc6257.html.

Tselikis, Christos (2013). «IMPLEMENTING DELAY-TOLERANT NETWORKING AT MOREHEAD STATE UNIVERSITY». Em.

V.Samaras, Christos (2010). «Design of Delay-TolerantTransportProtocol(DTTP) and its evaluation for Mars». Em: https://www.journals.elsevier.com/acta-astronautica.

APÊNDICES

A

APÊNCICE A

A.1 FICHEIROS DE CONFIRUAÇÃO DOS NÓS

A.1.1 *Nó 1*

A.1.1.1 *acsrsc*

1 7 262144

A.1.1.2 *ltprc*

"Initialization command (command 1). "Establish the LTP retransmission window. "A maximum of 64 sessions. 1 session 1 second of transmission "Set a block size limit of 1000000 bytes. (approx data sent per session) 1 64 1000000 1 100

"LTP Spans "a span <num> 100 100 64000 100000 1 'udplso x.x.x.x:1113 40000000'

-———————————————————————————————- "Listener on 0.0.0.0 s 'udplsi 0.0.0.0:1113'

w

A.1.1.3 *ipnrc*

"a plan <node> <protocol>/<num/address> "a plan 1 ltp/1 "a plan 2 tcp/10.0.0.2

a plan 1 udp/127.0.0.1 a plan 2 tcp/10.0.0.2 a plan 3 tcp/10.0.1.2 a plan 4 tcp/10.0.2.2

A.1.1.4 *ionconfig*

"wmKey 0 "sdrName ion$_sdr$"$wmSize$5000000"$configFlags$1"$heapWords$5000000$pathName/var/tmp/ion

A.1.1.5 *cfdprc*

"Initialize 1

"Add destinations a entity 1 bp ipn:1.0 10 0 0 a entity 2 bp ipn:2.0 10 0 0 a entity 3 bp ipn:3.0 10 0 0 a entity 4 bp ipn:4.0 10 0 0 a entity 5 bp ipn:5.0 10 0 0 a entity 6 bp ipn:6.0 10 0 0 a entity 7 bp ipn:7.0 10 0 0 a entity 8 bp ipn:8.0 10 0 0 a entity 9 bp ipn:9.0 10 0 0 a entity 10 bp ipn:10.0 10 0 0

e 1

w 1

m segsize 1000

m ckperiod 1 m maxtimeouts 2 m inactivity 300

"Discard partially received files upon cancellation of a file reception m discard 1

"Require CRCs on PDUs m requirecrc 1

s 'bputa'

A.1.1.6 *bprc*

1

a scheme ipn 'ipnfw' 'ipnadminep'

"Add endpoints. a endpoint ipn:1.1 x a endpoint ipn:1.2 x a endpoint ipn:1.3 x a endpoint ipn:1.4 x a endpoint ipn:1.5 x a endpoint ipn:1.6 x a endpoint ipn:1.7 x a endpoint ipn:1.8 x a endpoint ipn:1.9 x a endpoint ipn:1.10 x a endpoint ipn:1.64 x a endpoint ipn:1.65 x

-————————————————————————————- "Add a protocol for external nodes. -————————————————————————————-

a protocol tcp 1400 100 a protocol udp 1400 100 a protocol ltp 1400 100

-————————————————————————————- "Add an induct. (listen) -————————————————————————————- a induct tcp 0.0.0.0 tcpcli a induct udp 0.0.0.0 udpcli a induct ltp 1 ltpcli

-————————————————————————————- "Add outducts. -————————————————————————————- "a outduct tcp x.x.x.x tcpclo a outduct udp 127.0.0.1

udpclo a outduct tcp 10.0.0.2 tcpclo a outduct tcp 10.0.1.2 tcpclo a outduct tcp 10.0.2.2 tcpclo

-————————————————————————————- "a outduct ltp x ltpclo

-————————————————————————————- "Select level of BP watch activities - 0 = None; 1 = All w 0

r 'ipnadmin n1.ipnrc'

"Start all declared schemes and protocols on the local node s

A.1.1.7 *ionsecrc*

"Initialization command (command 1). 1 "Select level of "echo control"activities "0 = None; 1 = print to both log and stdout e 1

a key 'key2' /home/core/NASA$_DTN_CORE_Scenarios/CORE_configs/mars/config/key2.keyakey'key3

"a bspbcbrule ipn:1.* ipn:2.* 1 'BCB-AES128' key2 "a bspbibrule ipn:1.* ipn:6.* 1 'BIB-SHA256' key3

A.1.2 *Nó 2*

A.1.2.1 *acsrsc*

1 7 262144

A.1.2.2 *ltprc*

1 64 1000000 1 100

-————————————————————————————- "Add a span (a connection) "peer$_engine_br"max_exportsessions"max_importsessions"max_segmentsize"aggregationsizelimit"aggreg

-————————————————————————————- "LTP Spans "a span <num> 100 100 64000 100000 1 'udplso x.x.x.x:1113 40000000' a span 5 100 100 64000 100000 1 'udplso 10.0.3.4:1113 40000000'

-————————————————————————————- "Listener on 0.0.0.0 s 'udplsi 0.0.0.0:1113'

A.1.2.3 *ipnrc*

"a plan <node> <protocol>/<num/address> "a plan 1 ltp/1 "a plan 2 tcp/10.0.0.2

a plan 1 tcp/10.0.0.1 a plan 2 udp/127.0.0.1 a plan 5 ltp/5

A.1.2.4 *ionconfig*

"wmKey 0 "sdrName ion$_sdr$"$wmSize$5000000"$configFlags$1"$heapWords$5000000$pathName/var/tm

A.1.2.5 *cfdprc*

1

"Add destinations a entity 1 bp ipn:1.0 10 0 0 a entity 2 bp ipn:2.0 10 0 0 a entity 3 bp ipn:3.0 10 0 0 a entity 4 bp ipn:4.0 10 0 0 a entity 5 bp ipn:5.0 10 0 0 a entity 6 bp ipn:6.0 10 0 0 a entity 7 bp ipn:7.0 10 0 0 a entity 8 bp ipn:8.0 10 0 0 a entity 9 bp ipn:9.0 10 0 0 a entity 10 bp ipn:10.0 10 0 0

"Echo control (all output from cfdpadmin to be logged) e 1

"Turn on CFDP watch characters w 1

"Set max segment size "(Number of bytes of file data in each file data PDU transmitted "by the local CFDP entity) m segsize 1000

m ckperiod 1 m maxtimeouts 2 m inactivity 300

"Discard partially received files upon cancellation of a file reception m discard 1

"Require CRCs on PDUs m requirecrc 1

s 'bputa'

A.1.2.6 *bprc*

1

"Add an EID scheme. a scheme ipn 'ipnfw' 'ipnadminep'

"Add endpoints. a endpoint ipn:2.1 x a endpoint ipn:2.2 x a endpoint ipn:2.3 x a endpoint ipn:2.4 x a endpoint ipn:2.5 x a endpoint ipn:2.6 x a endpoint ipn:2.7 x a endpoint ipn:2.8 x a endpoint ipn:2.9 x a endpoint ipn:2.10 x

-————————————————————————————- "Add a protocol for external nodes. -————————————————————————————- "Estimate transmission

capacity assuming 1400 bytes of each frame "for payload, and 100 bytes for overhead. a protocol tcp 1400 100 a protocol udp 1400 100 a protocol ltp 1400 100

-————————————————————————- "Add an induct. (listen) -————————————————————————- a induct tcp 0.0.0.0 tcpcli a induct udp 0.0.0.0 udpcli a induct ltp 1 ltpcli

-————————————————————————- "Add outducts. -————————————————————————- "a outduct tcp x.x.x.x tcpclo a outduct udp 127.0.0.1 udpclo a outduct tcp 10.0.0.1 tcpclo

-————————————————————————- "a outduct ltp x ltpclo a outduct ltp 5 ltpclo

-————————————————————————- "Select level of BP watch activities - 0 = None; 1 = All w 0

r 'ipnadmin n2.ipnrc'

"Start all declared schemes and protocols on the local node s

A.1.2.7 *ionsecrc*

1 "Select level of "echo control"activities "0 = None; 1 = print to both log and stdout e 0

A.1.3 *Nó 3*

A.1.3.1 *acsrsc*

1 7 262144

A.1.3.2 *ltprc*

1 64 1000000 1 100

-———————————————————————- "Add a span (a connection)

"peer$_{e}ngine_{n}br$"$max_{e}xport_{s}essions$"$max_{i}mport_{s}essions$"$max_{s}egment_{s}ize$"$aggregation_{s}ize_{l}imit$"$a$

-———————————————————————- "LTP Spans "a span <num>

100 100 64000 100000 1 'udplso x.x.x.x:1113 40000000' a span 5 100 100 64000 100000 1 'udplso 10.0.3.4:1113 40000000'

-———————————————————————- "Listener on 0.0.0.0 s 'udplsi

0.0.0.0:1113'

A.1.3.3 *ipnrc*

"a plan <node> <protocol>/<num/address> "a plan 1 ltp/1 "a plan 2 tcp/10.0.0.2

a plan 1 tcp/10.0.1.1 a plan 3 udp/127.0.0.1 a plan 5 ltp/5

A.1.3.4 *ionconfig*

"wmKey 0 "sdrName ion$_{s}dr$"$wmSize$5000000"$configFlags$1"$heapWords$5000000$pathName/var/tm$

A.1.3.5 *cfdprc*

1

"Add destinations a entity 1 bp ipn:1.0 10 0 0 a entity 2 bp ipn:2.0 10 0 0 a entity 3 bp ipn:3.0 10 0 0 a entity 4 bp ipn:4.0 10 0 0 a entity 5 bp ipn:5.0 10 0 0 a entity 6 bp ipn:6.0 10 0 0 a entity 7 bp ipn:7.0 10 0 0 a entity 8 bp ipn:8.0 10 0 0 a entity 9 bp ipn:9.0 10 0 0 a entity 10 bp ipn:10.0 10 0 0

"Echo control (all output from cfdpadmin to be logged) e 1

"Turn on CFDP watch characters w 1

"Set max segment size "(Number of bytes of file data in each file data PDU transmitted "by the local CFDP entity) m segsize 1000

m ckperiod 1 m maxtimeouts 2 m inactivity 300

"Discard partially received files upon cancellation of a file reception m discard 1

"Require CRCs on PDUs m requirecrc 1

s ’bputa’

A.1.3.6 *bprc*

1

"Add an EID scheme. a scheme ipn ’ipnfw’ ’ipnadminep’

"Add endpoints. a endpoint ipn:3.1 x a endpoint ipn:3.2 x a endpoint ipn:3.3 x a endpoint ipn:3.4 x a endpoint ipn:3.5 x a endpoint ipn:3.6 x a endpoint ipn:3.7 x a endpoint ipn:3.8 x a endpoint ipn:3.9 x a endpoint ipn:3.10 x

-————————————————————————————- "Add a protocol for external nodes. -————————————————————————————- "Estimate transmission capacity assuming 1400 bytes of each frame "for payload, and 100 bytes for overhead. a protocol tcp 1400 100 a protocol udp 1400 100 a protocol ltp 1400 100

-————————————————————————————- "Add an induct. (listen) -———— ————————————————————————————- a induct tcp 0.0.0.0 tcpcli a induct udp 0.0.0.0 udpcli a induct ltp 1 ltpcli

-————————————————————————————- "Add outducts. -———————— ————————————————————————————- "a outduct tcp x.x.x.x tcpclo a outduct udp 127.0.0.1 udpclo a outduct tcp 10.0.1.1 tcpclo

-————————————————————————————- "a outduct ltp x ltpclo a outduct ltp 5 ltpclo

-————————————————————————————- "Select level of BP watch activities - 0 = None; 1 = All w 0

r ’ipnadmin n3.ipnrc’

"Start all declared schemes and protocols on the local node s

A.1.3.7 *ionsecrc*

1 "Select level of "echo control"activities "0 = None; 1 = print to both log and stdout e 0

A.1.4 *Nó 4*

A.1.4.1 *acsrsc*

1 7 262144

A.1.4.2 *ltprc*

1 64 1000000 1 100

-————————————————————————————- "Add a span (a connection)

"peer$_{e}ngine_{n}br$"$max_{e}xport_{s}essions$"$max_{i}mport_{s}essions$"$max_{s}egment_{s}ize$"$aggregation_{s}ize_{l}imit$"$a$

-————————————————————————————- "LTP Spans "a span <num>

100 100 64000 100000 1 'udplso x.x.x.x:1113 40000000' a span 5 100 100 64000 100000 1 'udplso 10.0.3.4:1113 40000000'

-————————————————————————————- "Listener on 0.0.0.0 s 'udplsi

0.0.0.0:1113'

w 0

A.1.4.3 *ipnrc*

"a plan <node> <protocol>/<num/address> "a plan 1 ltp/1 "a plan 2 tcp/10.0.0.2

a plan 1 tcp/10.0.2.1 a plan 4 udp/127.0.0.1 a plan 5 ltp/5

A.1.4.4 *ionconfig*

"wmKey 0 "sdrName ion$_{s}dr$"$wmSize$5000000"$configFlags$1"$heapWords$5000000$pathName/var/tm$

A.1.4.5 *cfdprc*

1

"Add destinations a entity 1 bp ipn:1.0 10 0 0 a entity 2 bp ipn:2.0 10 0 0 a entity 3 bp ipn:3.0 10 0 0 a entity 4 bp ipn:4.0 10 0 0 a entity 5 bp ipn:5.0 10 0 0 a entity 6 bp ipn:6.0 10 0 0 a entity 7 bp ipn:7.0 10 0 0 a entity 8 bp ipn:8.0 10 0 0 a entity 9 bp ipn:9.0 10 0 0 a entity 10 bp ipn:10.0 10 0 0

"Echo control (all output from cfdpadmin to be logged) e 1

"Turn on CFDP watch characters w 1

"Set max segment size "(Number of bytes of file data in each file data PDU transmitted "by the local CFDP entity) m segsize 1000

m ckperiod 1 m maxtimeouts 2 m inactivity 300

"Discard partially received files upon cancellation of a file reception m discard 1

"Require CRCs on PDUs m requirecrc 1

s 'bputa'

A.1.4.6 *bprc*

1

"Add an EID scheme. a scheme ipn 'ipnfw' 'ipnadminep'

"Add endpoints. a endpoint ipn:4.1 x a endpoint ipn:4.2 x a endpoint ipn:4.3 x a endpoint ipn:4.4 x a endpoint ipn:4.5 x a endpoint ipn:4.6 x a endpoint ipn:4.7 x a endpoint ipn:4.8 x a endpoint ipn:4.9 x a endpoint ipn:4.10 x a endpoint ipn:4.64 x a endpoint ipn:4.65 x

-————————————————————————————- "Add a protocol for external nodes. -————————————————————————————- "Estimate transmission capacity assuming 1400 bytes of each frame "for payload, and 100 bytes for overhead. a protocol tcp 1400 100 a protocol udp 1400 100 a protocol ltp 1400 100

-————————————————————————————- "Add an induct. (listen) -———— ————————————————————————————- a induct tcp 0.0.0.0 tcpcli a induct udp 0.0.0.0 udpcli a induct ltp 1 ltpcli

-————————————————————————————- "Add outducts. -———————— ————————————————————————————- "a outduct tcp x.x.x.x tcpclo a outduct udp 127.0.0.1 udpclo a outduct tcp 10.0.2.1 tcpclo

-————————————————————————————- "a outduct ltp x ltpclo a outduct ltp 5 ltpclo

-————————————————————————————- "Select level of BP watch activities - 0 = None; 1 = All w 0

r 'ipnadmin n4.ipnrc'

"Start all declared schemes and protocols on the local node s

D E C L A R A Ç Ã O

Declaro, sob compromisso de honra, que o trabalho apresentado nesta dissertação, com o título *"Análise de segurança a protocolos de comunicação de elevada latência"*, é original e foi realizado por Estudante Tiago André Santos Gaspar (2190227) sob orientação de Professor Doutor Paulo Jorge Gonçalves Loureiro (paulo.loureiro@ipleiria.pt).

Leiria, Março 2022

Estudante Tiago André Santos Gaspar