

Title Model for Computer Networks Optimization

J. H. S. Pereira, L. M. Sato, P. F. Rosa and S. T. Kofuji

Abstract— The TCP/IP architecture was consolidated as a standard to the distributed systems. However, there are several researches and discussions about alternatives to the evolution of this architecture and, in this study area, this work presents the Title Model to contribute with the application needs support by the cross layer ontology use and the horizontal addressing, in a next generation Internet. For a practical viewpoint, is showed the network cost reduction for the distributed programming example, in networks with layer 2 connectivity. To prove the title model enhancement, it is presented the network analysis performed for the message passing interface, sending a vector of integers and returning its sum. By this analysis, it is confirmed that the current proposal allows, in this environment, a reduction of 15,23% over the total network traffic, in bytes.

Keywords— Message Passing Interface, Ontology, Optimization, Distributed Programming, Computer Networks.

I. INTRODUÇÃO

A comunicação distribuída permite associar o poder computacional de estações distintas e respectivos processadores, memórias e relógios. Há diversas tecnologias utilizadas para comunicação distribuída, mas em geral estas utilizam redes de computadores, na arquitetura TCP/IP, e fazem uso das camadas mais baixas (física e enlace) e intermediárias (rede e transporte), desta arquitetura. O uso destas camadas, associado ao controle e transferência de dados entre as estações, pode resultar, em certos casos, em elevada exigência de recursos de rede e, consequentemente, alto custo computacional [1].

Face à importância da comunicação distribuída para os sistemas computacionais, este trabalho apresenta o Modelo de Título como alternativa para a próxima geração de Internet e mostra o benefício do uso deste modelo para o exemplo da redução do custo de comunicação em rede, para a programação distribuída, quando utilizada a arquitetura TCP/IP com conectividade em camada 2.

A possibilidade de melhorias na comunicação em rede motiva esta pesquisa, sendo relevante posicionar que não é escopo deste trabalho expandir as discussões dos benefícios do

J. H. S. Pereira, Universidade de São Paulo, São Paulo, SP, Brasil, joaohs@usp.br

L. M. Sato, Universidade de São Paulo, São Paulo, SP, Brasil, liria.sato@poli.usp.br

S. T. Kofuji, Universidade de São Paulo, São Paulo, SP, Brasil, kofuji@pad.lsi.usp.br

P. F. Rosa, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil, frosi@facom.ufu.br

Modelo de Título a todos os campos da comunicação em rede, visto que, em decorrência da extensão do campo que abrange a área, não seria possível esgotar todas as informações neste artigo. Desta forma, a discussão aqui apresentada está limitada ao campo da programação distribuída.

Este trabalho está organizado em seis seções. A seção 2 apresenta os trabalhos relacionados na área de otimização de rede para a programação distribuída e os trabalhos na área de próxima geração de Internet. A seção 3 apresenta o Modelo de Título para a próxima geração de Internet. A seção 4 descreve a otimização do custo de rede para a programação distribuída e a seção 5 mostra a análise dos resultados e avaliação de desempenho. Por fim, a seção 6 apresenta as conclusões e sugestões para trabalhos futuros nesta área de pesquisa.

II. TRABALHOS RELACIONADOS À EVOLUÇÃO DA ARQUITETURA TCP/IP

A arquitetura TCP/IP é amplamente utilizada na área de redes e segue o Modelo de Referência OSI da ISO, porém, sem utilizar expressamente as camadas de sessão e apresentação. Os protocolos principais desta arquitetura surgiram há cerca de 3 décadas, por exemplo, o IP, TCP e UDP, publicados pelo IETF nas RFC 760, 761 e 768, em 1980 [2]-[4].

Conforme apresentado em [5] e [6], apesar da crescente evolução computacional, não houve melhorias expressivas nas camadas 3 e 4 da arquitetura TCP/IP, sendo que os novos requisitos das aplicações foram atendidos por novas especificações/adaptações em protocolos, sem que houvesse melhoria estrutural das camadas intermediárias, o que gerou lacunas no atendimento destes requisitos.

Alguns estudos avaliam que as melhorias nas camadas 3 e 4 da arquitetura Internet foram dificultadas pela expansão da base instalada, que passou de interesses militares e acadêmicos para o de uso comercial [7]. Com isto, as melhorias nas camadas 3 e 4 tornaram-se mais complexas, ao passar dos anos, e os novos requisitos, demandados pelas aplicações, passaram a ser atendidos por estratégias que não alterassem a estrutura original destas camadas.

Como exemplo, no início da década de 80, as redes não tinham necessidade de transmissão de voz e vídeo sobre as redes de computadores. Também não era necessário suportar redes de sensores, suas necessidades de tempo real e comunicação inter/intra *clusters* e *grids* computacionais. Alguns dos requisitos demandados pelas aplicações, que tiveram alterações nos últimos 30 anos, são: QoS; Segurança;

Mobilidade; Vazão; Tempo real; *Jitter*; Latência; Garantia de Entrega; e; Gerenciamento, dentre outros.

As alterações e evoluções destes requisitos, ao longo dos anos, nem sempre foram acompanhados por melhorias e otimizações nos modelos e arquiteturas das redes de computadores e da Internet. Nesta área, há intenso interesse atual de pesquisadores e alguns dos trabalhos recentes são desenvolvidos com perspectivas para a Internet do futuro.

A. Próxima geração de Internet

Dentre as razões para estudos na área de próxima geração de Internet há possibilidades de melhorias que motivam propostas de alterações na arquitetura atual e do IP, incluso o IPv4, que hoje é o principal protocolo de roteamento utilizado na Internet.

Desde a década de 90 o IETF trabalha em projeto de protocolos para substituir o IPv4. Houve propostas com o incremento da complexidade do IP e também com adaptações no protocolo OSI CLNS (*Connectionless Network Service*). Outro projeto procurou simplificar o IPv4 e aumentar a quantidade de endereços. Este projeto foi conhecido como SIP (*Simple IP*) e a extensão deste foi denominada SIPP (*Simple IP Plus*).

Nestas propostas, devido a uma confusão de versão, causada pelo IAB (*Internet Architecture Board*) ao publicar uma política com referência sobre a próxima versão do IP ser a de número 7, a proposta para nova versão do IP passou a ser chamada de IPng (*IP – Next Generation*), para evitar erros na referência da numeração das versões do IP.

Esta confusão ocorreu porque um dos documentos do IAB referenciavam erroneamente a versão corrente como sendo a 6 e a versão 5 havia sido designada para o protocolo ST [7]. Posteriormente a versão 6 consolidou-se como a proposta do IETF para a próxima geração do IP (IPng), cuja especificação foi publicada em 1995 como IPv6 [8]-[9].

O IPv6 apresenta algumas evoluções em comparação com o IPv4, como por exemplo o aumento da quantidade de endereços, o formato flexível do cabeçalho, incremento de informações opcionais, suporte a alocação de recursos e previsão para extensões no próprio protocolo. Por outro lado, a especificação do IPv6, realizada há mais de 1 década, não suporta parte das necessidades de segurança e mobilidade das aplicações atuais.

As discussões para a evolução da Internet permeiam a área dos sistemas distribuídos, há décadas, e possuem avanço constante e considerável para a camada de aplicação e as camadas inferiores. Porém, esta evolução não ocorre com a mesma expressividade nas camadas intermediárias, o que se comprova pelos estudos apresentados em [5].

Nesta área de pesquisa também há exemplos de tentativas de evolução do TCP para o SCTP (*Stream Control Transmission Protocol*), em 2000 [10]-[11]. Apesar de mais de 1 década de tentativa, o uso destas especificações permanece com baixa abrangência e impacto na Internet e a dificuldade de implantação de melhorias ocorre devido a diversos fatores.

Um destes fatores é a complexidade de alterar a rede legada e sua arquitetura, que funciona como infraestrutura para uma imensa quantidade de usuários que a utilizam diariamente. Também, uma alteração nas camadas intermediárias, como é o caso do IPv6, não ocorre transparentemente para a camada de aplicação, que precisará refazer códigos e recompilar uma grande quantidade de serviços, pois utilizam atualmente campos de endereços de 32 bits e precisarão alterar para 128 bits, exceto para os casos de mapeamento do Ipv4, previsto no projeto do IPv6.

Por outro lado, a arquitetura TCP/IP segue em constante incremento de sua complexidade para tentar atender necessidades de comunicação distribuída, não contempladas em seu projeto. Em paralelo, a possibilidade de uma próxima geração de Internet, que atenda aos novos requisitos dos sistemas distribuídos, tem motivado o aumento de interesse em pesquisas nesta área de estudos.

Um destes estudos apresenta propostas para FDR (*Future Domain Routing*), sendo conduzido pelo IRTF RRG (*Internet Research Task Force Routing Research Group*) com trabalhos sobre os problemas de escalabilidade realizados pelo RR-FS (*Scalability Research Subgroup*). Também há os trabalhos de Tsuchiya que introduz a proposta de “landmark” em seus estudos sobre hierarquia de roteamento em redes de grande extensão e os estudos sobre roteamento compacto feitos por Krioukov’s, também do IRTF RRG [12]-[13].

Outros trabalhos com propostas relacionadas à escalabilidade do sistema de roteamento e arquitetura de endereçamento para a Internet são realizados pelo *Network Working Group* do IETF, como o LISP (*Locator Identifier Separation Protocol*), que faz a separação dos endereços de Internet em “*Endpoint Identifiers*” (EIDs) e “*Routing Locators*” (RLOCs) sem a necessidade de alteração das pilhas de protocolo nas estações e pouco impacto na infra estrutura legada [14]. Este grupo também apresenta propostas para a interoperabilidade do LISP com o IPv4 e o IPv6 [15].

No LISP, para não haver elevado impacto na infraestrutura de roteadores e suas pilhas de protocolos, é colocada uma camada sobre as camadas de rede e transporte. Isto melhora a interoperabilidade com os sistemas legados e a escalabilidade, porém não reduz a crescente complexidade dos protocolos do TCP/IP. Ressalte-se que uma camada adicional impõe maior processamento nestes elementos de rede.

Algumas pesquisas relacionadas à evolução da Internet propõe uso de roteamento *flat* para a comunicação entre os elementos de rede. Nestas pesquisas há os trabalhos de *Landmark-based Flat Routing* e roteamento *flat* sobre um espaço de identidade binária [16]-[17], em que Pasquini apresenta propostas de melhorias sobre o IBR (*Identity Based Routing*), VRR (*Virtual Ring Routing*) e ROFL (*Routing on Flat Labels*) desenvolvidos por Caeser et. al [18]-[20].

Pasquini também apresenta estudos para identificadores de domínio em arquitetura de Internet de próxima geração, com trabalhos sobre roteamento *flat* nesta arquitetura [21]-[22].

Outra linha de estudos é desenvolvida por Bryan Ford, que propõe evoluções para uma nova Internet em seus trabalhos de

arquitetura para redes sociais, nomeada de UIA (*Unmanaged Internet Architecture*). Em suas propostas Ford especifica o UIP (*Unmanaged Internet Protocol*) [23]-[24].

Na área de arquitetura de Internet de próxima geração, para suporte a mobilidade e *multi-homing*, Wong apresenta estudos para melhorar a sobrecarga semântica do IP com a introdução de uma camada de identidade localizada entre as camadas de rede e transporte da arquitetura Internet [25]-[26].

Para a otimização do custo de comunicação em redes locais há a proposta do FINLAN (*Fast Integration of Network Layers*), que propõe uma alternativa para a comunicação distribuída sem utilização dos protocolos IP, TCP e UDP e possibilita a comunicação híbrida com estações que utilizam estes protocolos tradicionais das camadas de rede e transporte do TCP/IP [27]-[29].

O projeto Europeu 4WARD desenvolve trabalhos para arquitetura de Internet do futuro com a meta de desenvolver redes e aplicações distribuídas mais rápidas e fáceis tomando o resultado dos serviços de comunicação mais avançados e acessíveis às pessoas.

A proposta deste projeto apresenta o uso de caminhos genéricos (GP) definido no contexto do paradigma da rede de informação desenvolvida no 4WARD, habilitando conexões para objetos de informação e não apenas para estações [30]. Este grupo também colabora com as propostas de arquitetura de Internet de conteúdo centralizado (*Content-Centric*) feitas pelo FIA (*Future Internet Assembly*) e *Future Content Networks (FCN) Group* [31].

Incluindo o 4WARD, há dezenas de projetos no FIRE (*Future Internet Research and Experimentation*), como por exemplo ANA, AUTOI, BIONETS, CASCADAS, CHIANTI, ECODE, EFIPSANS, Euro-NF, Federica, HAGGLE, MOMENT, NADA, N4C, N-CRAVE, OneLab2, OPNEX, PERIMETER, PII, PSIRP, ResumeNet, Self-NET, SMART-Net, SmoothIT, TRILOGY, Vital++ e WISEBED.

Como contribuição aos estudos em Internet de próxima geração, este trabalho apresenta o uso do Modelo de Título para otimização do custo de rede na programação distribuída, com utilização de ontologia *cross layers* para aproximação semântica entre as camadas superiores e as inferiores e, consequentemente, melhor suporte às necessidades de comunicação em rede.

III. MODELO DE TÍTULO PARA A PRÓXIMA GERAÇÃO DE INTERNET

No endereçamento hierárquico do IP há o problema de a informação da rede/sub-rede ter forte acoplamento entre o endereço IP de uma estação e suas vizinhanças. Assim, caso uma estação desloque para outra rede o endereço desta, em geral, precisa ser alterado.

Este fato aumenta a complexidade da mobilidade na Internet, pois este problema normalmente precisa ser equacionado pelas camadas superiores (ou por links dedicados / VPN – *Virtual Private Network*), devido à mobilidade não ter sido uma premissa no projeto da camada de rede da arquitetura TCP/IP.

No endereçamento do IP também há o problema da segmentação das classes, implicando que ao crescer além de 254 estações, em uma classe C, é necessário alocar outro segmento para endereçamento, o que impacta no planejamento de crescimento de uma rede, pois este não poderá ser feito gradativo sem passar pelos saltos de alocação das faixas, ou sub-faixas.

Um terceiro problema no endereçamento com o IP ocorre em estações com 2 ou mais conexões com a Internet, pois precisam ter mais de um endereço IP. Isto implica no endereçamento, uma vez que uma única estação passa a comportar como entidades diferentes para a rede. Por exemplo, na Fig. 1, a estação 1 para falar com a estação 2 pode escolher os endereços 2A ou 2B via o roteador R [7].

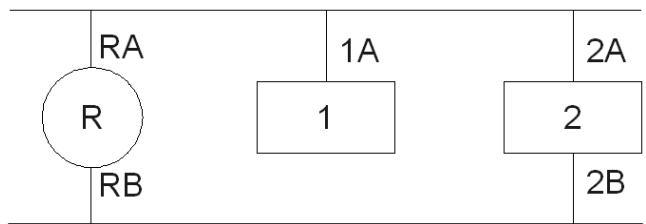


Figura 1. Problema de Comunicação no TCP/IP.

Caso um usuário, ou aplicação, da estação 1 conheça apenas o endereço 2A da estação 2 e esta conexão pare de funcionar haverá interrupção da comunicação, mesmo que a entidade 2 e a conexão 2B continuem em funcionamento e haja caminho disponível via o roteador R. Para contornar esta fragilidade na arquitetura TCP/IP é necessário o controle explícito de conexões distintas que usam o mesmo IP, uma vez que a identificação de diferentes conexões para uma mesma entidade não é nativa.

O endereçamento horizontal no Modelo de Título é uma alternativa para que a identificação e localização das entidades (elementos de rede) não tenha forte acoplamento com os vizinhos e a identificação destas, além de estação, também possa representar outros elementos da comunicação distribuída, como por exemplo, usuários, aplicações, conteúdo, sensores e nuvens computacionais.

Para a elaboração de uma proposta desta natureza é importante ter como ponto de partida o diagnóstico sobre as necessidades de comunicação com base em estudos e observações do mundo real, para então definir as metas e realizar a proposta de acordo com o suporte exigido para as necessidades de comunicação identificadas no diagnóstico.

A. Observações sobre as necessidades de comunicação

Os usuários dos sistemas de comunicação em rede normalmente possuem a necessidade de comunicar com outros usuários e também com outros elementos de rede, por exemplo, o uso de comunicação em voz sobre IP ou aplicações para visualizar páginas em um site.

Outras necessidades são as de comunicação disponível, móvel, instantânea e precisa, com boa qualidade na troca de sintaxes e semânticas. Além da transferência de semânticas e conceitos também há necessidade de transferência de dados e

informações de sons e imagens, dentre outros.

Há necessidades de poder optar por comunicação privada, sigilosa, segura, aberta entre grupos sociais privados/públicos ou coletiva, conforme o desejo momentâneo e/ou o assunto. Também, é desejável poder ajustar o volume, timbre, frequência, velocidade, tom, luminosidade, contraste, brilho e outras características durante a comunicação. Pode ser desejado interromper uma comunicação a qualquer instante e retomá-la quando melhor convier, como também pode ser intenção gravar para revisão ou distribuição posterior.

De uma forma geral, os atuais sistemas distribuídos atendem grande parte destas necessidades de comunicação, embora, na maioria, a camada de aplicação é a responsável por suportar tais necessidades, e as camadas intermediárias da arquitetura TCP/IP, por sua vez, não estão preparadas para compreender e suportar semanticamente várias destas diferentes necessidades de comunicação.

O que normalmente ocorre, na prática, é a escolha do protocolo de transporte como TCP ou UDP, quando da necessidade de comunicação confiável, ou não confiável, em situações de tempo real (como conversação de áudio e/ou vídeo) e esforços para configurações, em geral manuais, para ativar QoS nas redes por especificação explícita de endereços IP de origem e/ou destino, número de porta do TCP/UDP ou pela marcação do campo DS (*Differentiated Services*) no uso do DSCP (*Differentiated Services Code Point*), formado pelos 6 primeiros bits do DS [32]-[33].

Caso as camadas intermediárias da arquitetura Internet suportassem compreender semanticamente as comunicações e suas necessidades e/ou desejos, em curso pelo conteúdo dos pacotes trafegados, haveria a possibilidade de melhor atendimento das necessidades de comunicação e também da redução da complexidade da arquitetura atual.

B. Metas para o Modelo de Título

Como metas para o Modelo de Título há a necessidade de solucionar limitações da arquitetura atual que, de forma macro, são:

1. Melhor suporte às necessidades de comunicação;
2. Reduzir a complexidade da arquitetura atual;
3. Quantidade de endereços suficiente para necessidades atuais e futuras.

O problema em relação à quantidade de endereços é contornável no TCP/IP pelo uso do NAT e pelo aumento de endereços, que passa de 32bits no IPv4 para 128bits no IPv6. Por estas razões, este problema não é intransponível na arquitetura atual, porém é um ponto a considerar na definição de um novo modelo.

Quanto a melhorar o suporte das camadas intermediárias às necessidades de comunicação há os exemplos de mobilidade, segurança, privacidade, *multicast* e mudança de necessidades no decorrer de uma comunicação. Outra preocupação importante é a interoperabilidade com as redes legadas, que é fundamental para a comunicação com a infra estrutura atual.

Este modelo também precisa ser escalável com suporte ao aumento de ordens de grandeza computacionais por gerações,

sem impacto na complexidade da arquitetura.

Deve ser flexível para suportar futuras necessidades de comunicação sem impacto estrutural, ou que resulte em maior complexidade da arquitetura. Para isto é importante manter a independência entre as camadas por meio de suas interfaces. Com isto, uma entidade de comunicação não precisa tratar todos os detalhes de todos os níveis da comunicação, suas estruturas e protocolos envolvidos na troca de dados, informações e conhecimento.

C. Modelo de Título

A multiplicidade de endereços utilizados na arquitetura TCP/IP torna mais complexa a comunicação distribuída, quando do objetivo de atender as necessidades básicas de comunicação. Porém, os vários endereços utilizados são um meio para ter como consequência o estabelecimento da comunicação e isto é apenas uma parte do conjunto dos sistemas distribuídos como um todo.

Por exemplo, em uma comunicação, VoIP ou de mensagens texto via IM (*Instant Message*), o objetivo da comunicação é a troca de dados/informações/conhecimento entre dois, ou mais seres humanos, porém, para que esta comunicação seja possível na arquitetura TCP/IP são necessários vários endereços, como:

1. Endereço Físico;
2. Endereço Lógico;
3. Endereço de processos;
4. Endereço de AS (*Autonomous System*);
5. Endereço de usuário (*login*).

Esta quantidade de endereços controlada pelas diferentes camadas tende a aumentar a complexidade da comunicação distribuída, pois o objetivo fim é apenas o do usuário A comunicar com o usuário B.

Em outra situação mais simples, ao acessar um site para leitura, um usuário pode querer apenas receber texto de uma estação remota e, novamente, vários endereços serão necessários na arquitetura atual.

Pela análise desta observação e para reduzir a complexidade do endereçamento na arquitetura Internet, que envolve questões quanto à possibilidade de localizar não apenas estações, mas também aplicações e usuários, dentre outros, este trabalho propõe unificar as estruturas de endereçamento utilizadas no TCP/IP.

Para tanto, propõe-se que a utilização de títulos de aplicações, especificado na recomendação ISO-9545/X.207, seja estendido para as demais entidades de comunicação das redes de computadores. Conforme esta recomendação, os ASO-title (*Application Service Object-title*), que são utilizados para identificar sem ambiguidade os ASO, em um ambiente OSI, consistem de AP-title (*Application Process title*) que, por natureza, endereçam as aplicações horizontalmente [34].

Dessa forma, este trabalho busca ampliar o uso do título das aplicações com a unificação do endereçamento pelo uso do AP-title. Também é intenção que as camadas intermediárias suportem as necessidades das entidades de melhor forma.

Para não utilizar uma nomenclatura distinta para “título de usuário”, “título de estação” e “título de aplicação”, o que reduziria a flexibilidade de seu uso em outras necessidades de endereçamento (por exemplo, “título de *grid*”, “título de *cluster*” e “título de rede de sensor”), esta proposta apresenta a designação única de “título de entidade”, ou simplesmente “Título”, cujo objetivo é o de identificar uma entidade, independente de qual seja.

Assim, as entidades, independentemente de suas categorias, passam a ser suportadas por uma camada de serviços, capaz de compreender suas necessidades a cada instante no tempo. A Fig. 2 mostra o Modelo de Título, em camadas, comparado com o TCP/IP.

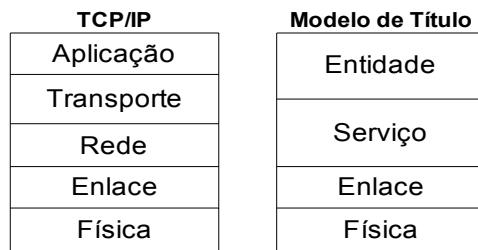


Figura 2. Comparação entre as camadas do TCP/IP e do Modelo de Título.

O Modelo de Título é independente da solução de endereçamento. Por exemplo, pode ser utilizado o “*Routing on Flat Labels*” ou o “*Locator Identifier Separation Protocol*”, porém estes precisam alterar suas estruturas para melhor suporte às necessidades das entidades unificadas em título e não apenas às de estações ou aplicações.

Para que a camada de serviços deste modelo ofereça suporte às necessidades das entidades, que podem alterar em função do contexto, é proposto utilizar ontologia, formalizada em OWL (*Web Ontology Language*), para representar explicitamente a interface entre estas camadas, como também representar o próprio Modelo de Título.

É proposto utilizar OWL por sua elevada expressividade semântica, por ter uso crescente na representação de ontologias na Internet e sua adoção pelo W3C (*World Wide Web Consortium*).

Pelo Modelo de Título, certas necessidades atuais das aplicações passam a ser atendidas de forma mais natural e menos complexa. Por exemplo, uma vez que o título endereça as entidades horizontalmente, a mobilidade é facilitada na Internet, pois deixa de existir a hierarquia dos segmentos de rede e sub-rede que ocorre no endereçamento com IP pelo uso das máscaras. Com isto é reduzido o acoplamento entre as vizinhanças e uma entidade e seus vizinhos podem estar naturalmente distribuídos em qualquer lugar no mundo.

Por exemplo, imagine o endereço IP nº. 200.225.100.170 estar na cidade de São Paulo e o endereço 200.225.100.171 estar em Tóquio, sem a necessidade de qualquer recurso adicional de rede como VPN (*Virtual Private Network*) ou conexões dedicadas. Imagine também estes endereços em deslocamento por diferentes redes de diferentes provedores sem a perda de comunicação ou necessidade de reconfiguração dos endereçamentos de rede. Estes são alguns

dos benefícios deste modelo que permite o endereçamento unificado de entidades, de forma horizontal.

Além de reduzir a complexidade dos múltiplos endereços utilizados na arquitetura atual o uso do título de entidade resolve o problema da quantidade de endereços possíveis, pois torna a quantidade de endereços ilimitada, visto que cada entidade tem uma identificação única e sem a delimitação da quantidade de caracteres, ou bits, possíveis.

D. Interoperabilidade entre o Modelo de Título e a Arquitetura Internet

A interoperabilidade deste modelo e a arquitetura Internet é semelhante ao LISP, mantendo as camadas atuais e seus diversos endereços (MAC, IP, Portas, *Login*, etc.) quando da comunicação com a rede legada. Desta forma, uma entidade no Modelo de Título precisará saber quais os endereços de destino na rede legada e precisará utilizar as mesmas pilhas de protocolos utilizadas atualmente. Para receber requisições da arquitetura TCP/IP precisa mapear endereços IP e número de porta diretamente para a identificação da entidade de comunicação.

Por outro lado, na comunicação entre entidades deste modelo, em redes também aptas a este, deixa de ser necessário o uso dos demais endereços, como o IP ou portas do TCP/UDP, sendo utilizado o título da entidade para o endereçamento.

IV. PROPOSTA DE OTIMIZAÇÃO DO CUSTO DE REDE PARA A PROGRAMAÇÃO DISTRIBUÍDA

O aumento da capacidade computacional dos elementos de rede permitiu o surgimento e a expansão de aplicações com maior complexidade e com demanda de requisitos distintos, como as aplicações com suporte a processamento distribuído. Embora esta evolução tenha sido expressiva para os elementos de rede, não foi acompanhada por melhorias expressivas nos protocolos das camadas de enlace, rede e transporte, da arquitetura TCP/IP, que manteve sua estrutura inalterada, em grande parte.

Por exemplo, as camadas de rede e transporte desta arquitetura são utilizadas pelos sistemas que utilizam a programação distribuída para suportar suas necessidades de comunicação, que neste caso são:

1. Endereçamento de estações;
2. Endereçamento de processos;
3. Controle do tamanho do pacote;
4. Garantia de entrega.

Uma solução que garanta o atendimento destas necessidades e que traga melhoria do custo de rede, pode ser uma alternativa para otimização da programação distribuída. Para tanto, este trabalho reduz redundâncias existentes nas camadas de enlace, rede e transporte, da arquitetura TCP/IP e traz uma alternativa pelo uso do Modelo de Título para a comunicação em rede.

A alteração do modelo em camadas impacta a compilação de aplicações e comunicação destas com os Sistemas

Operacionais Distribuídos (SOD). Não é objetivo deste trabalho discutir os impactos na compilação das aplicações, mas sim de apresentar uma alternativa que minimize o custo de rede a partir da otimização da estrutura dos protocolos de comunicação utilizados, alternativa esta baseada no Modelo de Título.

A. Ontologia para aproximação entre as camadas de rede

A comunicação das necessidades da camada de aplicação para as camadas intermediárias e inferiores, na arquitetura TCP/IP, é limitada pela escolha do protocolo de transporte (com ou sem garantia de entrega) e endereço de destino dos pacotes. A comunicação de outras necessidades é limitada, tais como, demandar segurança, mobilidade, multicast e QoS.

Para minimizar esta limitação de comunicação de necessidades entre as camadas, o Modelo de Título, apresentado neste trabalho, utiliza ontologia para aproximar semanticamente a camada superior das inferiores. O código abaixo, em OWL, exemplifica a formalização utilizada para a camada de aplicação para informar suas necessidades às camadas inferiores, em ambientes de programação distribuída.

```

<owl:ObjectProperty rdf:about="#Ontology1280888215552;Tem_Necessidade">
  <rdfs:domain rdf:resource="#Ontology1280888215552;Enderecamento_Estacao"/>
  <rdfs:domain
    rdf:resource="#Ontology1280888215552;Enderecamento_Processo"/>
  <rdfs:domain
    rdf:resource="#Ontology1280888215552;Controle_Tamanho_Pacote"/>
  <rdfs:domain rdf:resource="#Ontology1280888215552;Garantia_Entrega"/>
  <rdfs:subPropertyOf rdf:resource="#owl:topObjectProperty"/>
</owl:ObjectProperty>

<owl:Thing rdf:about="#Ontology1280888215552;ProgramacaoDistribuida">
  <rdf:type rdf:resource="#Ontology1280888215552;Aplicacao"/>
  <rdf:type rdf:resource="#owl:NamedIndividual"/>
  <TituloEstacao>PD_Mestre</TituloEstacao>
  <TituloEstacao>PD_Escravo_A</TituloEstacao>
  <TituloEstacao>PD_Escravo_B</TituloEstacao>
  <TituloProcesso>PD_LAM_MPI</TituloProcesso>
  <ControleTamanhoPacote>79</ControleTamanhoPacote>
  <GarantiaEntrega rdf:datatype="#xsd:boolean">Sim</GarantiaEntrega>
  <rdfs:comment>Programação Distribuída é uma instância da entidade Aplicação, que
    possui necessidade de comunicação entre mestre e escravos, com garantia de
    entrega e controle do tamanho do pacote.</rdfs:comment>
  <Tem_Necessidade
    rdf:resource="#Ontology1280888215552;ProgramacaoDistribuida"/>
</owl:Thing>

```

Enderecamento_Estacao, Enderecamento_Processo, Controle_Tamanho_Pacote e Garantia_Entrega são conceitos hierarquicamente abaixo do conceito necessidade na taxonomia do Modelo de Título, representado na ontologia por:

```

<owl:Class rdf:about="#Ontology1280888215552;Enderecamento_Estacao">
  <rdfs:subClassOf rdf:resource="#Ontology1280888215552;Necessidade"/>
</owl:Class>

<owl:Class rdf:about="#Ontology1280888215552;Enderecamento_Processo">
  <rdfs:subClassOf rdf:resource="#Ontology1280888215552;Necessidade"/>
</owl:Class>

<owl:Class rdf:about="#Ontology1280888215552;Controle_Tamanho_Pacote">
  <rdfs:subClassOf rdf:resource="#Ontology1280888215552;Necessidade"/>
</owl:Class>

<owl:Class rdf:about="#Ontology1280888215552;Garantia_Entrega">
  <rdfs:subClassOf rdf:resource="#Ontology1280888215552;Necessidade"/>
</owl:Class>

```

Este código é um exemplo de parte da formalização de necessidades de comunicação que podem variar em contextos

e ambientes distintos, porém representados formalmente pela ontologia do Modelo de Título, conforme a instância da entidade de comunicação e suas necessidades.

Esta comunicação de necessidades permite minimizar o custo de rede pela otimização da estrutura dos protocolos de comunicação utilizados na arquitetura TCP/IP. Assim, apresenta-se, a seguir, uma solução de rede para atender aos requisitos necessários à programação distribuída, possível a partir do uso da ontologia do Modelo de Título para as aplicações de programação distribuída informarem suas necessidades às camadas intermediárias da rede, em ambientes com conectividade em camada 2.

Estas necessidades são compreendidas a partir da semântica das necessidades formalizadas na ontologia para a comunicação entre a camada de aplicação e a de enlace. No Modelo de Título, esta formalização é respectiva à comunicação *cross layer* entre a camada de entidade e a de enlace, por meio da camada de serviço.

B. Considerações sobre o endereçamento de estações

O endereçamento de estações em redes de dados é realizado pelos endereços lógico e físico. Porém, de acordo com Hegering [37], para uma rede local que tenha conectividade em camada 2, o uso destes dois endereços por estação pode ser redundante em alguns casos. Em redes com conectividade em camada 2, o endereço lógico pode ser substituído pelo físico em alguns casos como, por exemplo, para estações sem endereçamento dinâmico.

Para manter a compatibilidade com os padrões atuais, as aplicações e seus usuários podem fazer uso do endereço lógico, e deixar o Sistema Operacional Distribuído resolvê-lo pelo uso do ARP (*Address Resolution Protocol*) [38]-[40]. Uma vez localizado o endereço físico pelo Reply do ARP (Eth Type: 0806 e ARP Opcode: 0002) no retorno do Request do endereço lógico (Eth Type: 0806 e ARP Opcode: 0001), não há mais necessidade de endereçar os pacotes com uso da camada 3, em uma rede com conectividade em camada 2.

Assim, os pacotes podem ser entregues diretamente pelo endereço físico, sem utilizar o endereço IP, garantindo a economia de 20 bytes, que é o tamanho mínimo do cabeçalho IP [41].

Apesar da exclusão da camada 3 não impactar nas conexões em camada 2, isto não resolve a necessidade de endereçamento de múltiplos processos, que para a arquitetura TCP/IP é feito pelo uso das portas na camada 4, tanto para o protocolo UDP quanto para o TCP [4] [42].

C. Considerações sobre o endereçamento de processos

Para otimizar o uso da camada de Transporte, é necessária uma alternativa para endereçar os processos. Este requisito pode ser atendido sem utilizar o UDP ou TCP, na camada 4 do TCP/IP. A proposta deste endereçamento é enviar 1 byte de controle no início do *payload* da camada de enlace.

Deste byte, os dois primeiros bits informam quantos bytes serão utilizados para o endereçamento de múltiplos processos, conforme a regra 2^N , sendo N o valor em decimal destes dois bits. Os bits complementares a estes dois serão utilizados para o

endereçamento de processos. Por exemplo, para o valor “00” são 6 bits complementares, que permitem até 64 endereços, conforme abaixo:

| | |
|---------------------------|-----------------------------|
| valor 00: $2^0 = 1$ byte | $2^6 = 64$ addresses |
| valor 01: $2^1 = 2$ bytes | $2^{14} = 16.384$ addresses |
| valor 10: $2^2 = 4$ bytes | $2^{30} = 1G$ addresses |
| valor 11: $2^3 = 8$ bytes | $2^{62} = 4E$ addresses |

O limite para a quantidade de portas para os protocolos TCP e UDP é de 2^{16} (64K) – menor que o desta proposta, que suporta 2^{62} . Para as atuais aplicações distribuídas, uma quantidade maior de endereços não é fundamentalmente necessária, mas futuramente poderá vir a ser, quando houver necessidade de comunicação entre estações com quantidades gigantescas de processadores e que suportem quantidades enormes de processos em HPC (*High Performance Computing*).

O TCP possui 20 bytes de cabeçalho e para endereçar os processos é proposto uma quantidade variável de 1 a 8 bytes, conforme a quantidade de endereços de processos utilizados. Assim, a redução da quantidade de bytes enviados em relação ao cabeçalho do TCP é:

| | |
|---------------------------|-----------------|
| valor 00: $2^0 = 1$ byte | $1/20 = 95,0\%$ |
| valor 01: $2^1 = 2$ bytes | $2/20 = 90,0\%$ |
| valor 10: $2^2 = 4$ bytes | $4/20 = 80,0\%$ |
| valor 11: $2^3 = 8$ bytes | $8/20 = 60,0\%$ |

O cabeçalho do UDP possui 8 bytes e nesta proposta, a redução da quantidade de bytes enviados em relação ao cabeçalho deste protocolo de transporte é:

| | |
|---------------------------|----------------|
| valor 00: $2^0 = 1$ byte | $1/8 = 87,5\%$ |
| valor 01: $2^1 = 2$ bytes | $2/8 = 75,0\%$ |
| valor 10: $2^2 = 4$ bytes | $4/8 = 50,0\%$ |
| valor 11: $2^3 = 8$ bytes | $8/8 = 0,0\%$ |

Desta forma, é aumentada a quantidade de endereçamento para processos com a redução do consumo de banda, exceto para o valor “11” em comparação ao UDP, que apresenta 0% de redução.

Para a arquitetura atual de comunicação distribuída, com valor máximo de porta igual a 2^{16} , o valor “10” seria suficiente, uma vez que permite o equivalente para endereçar até 1 Giga portas (processos). Para este valor, há uma redução do custo de rede de 80%, na camada de transporte, comparado ao uso do TCP e de 50% para o UDP.

Ao remover as camadas tradicionais de rede e transporte é necessário controlar o tamanho do pacote, pois esta informação é enviada, na arquitetura TCP/IP pelas camadas tradicionais (camadas 3 e 4), removidas nesta proposta. Para suportar esta necessidade é apresentada, a seguir, uma solução.

D. Considerações sobre o controle do tamanho do pacote

A quantidade de bytes do pacote é informada no Datagrama IP no Campo “Total Length”, de 16 bits e na camada de Transporte esta informação trafega no campo “Length” do TCP e do UDP, também de 16 bits. Na arquitetura TCP/IP, o uso do “Total Length” no IP e do

“Length” no TCP e UDP é redundante, visto que possuem uma relação próxima, definida por:

$$\text{Length} = \text{Total Length} - \text{IHL} \times 4$$

Obs.: IHL (*IP Header Length*) é o segundo campo do cabeçalho IP.

Para informar o tamanho do pacote, sem utilizar a camada 3 e 4, propõe-se o envio de um byte, onde os 2 primeiros bits indicam quantos bytes informarão o tamanho do pacote pela regra 2^N . Assim, o controle do tamanho do pacote é homogêneo com a solução anterior para endereçar processos e segue a mesma lógica de construção, onde o tamanho máximo dos dados de um pacote é:

| | |
|---------------------------|-----------------------------|
| valor 00: $2^0 = 1$ byte | $(2^6)-1 = 63$ bytes |
| valor 01: $2^1 = 2$ bytes | $(2^{14})-1 = 16.383$ bytes |
| valor 10: $2^2 = 4$ bytes | $(2^{30})-1 = 1G-1$ bytes |
| valor 11: $2^3 = 8$ bytes | $(2^{62})-1 = 4E-1$ bytes |

Para o IP, TCP e UDP, que utilizam 16 bits para informar o tamanho de um pacote, o tamanho máximo é de 64K bytes, sendo inferior à capacidade desta proposta, que permite aumentar o tamanho dos pacotes. Isto pode ser útil, em certos casos, na comunicação distribuída, onde for necessário o envio de uma quantidade grande de dados.

Por exemplo, na classificação e análise de alvos e vizinhanças de cubos espaciais utilizando UWB (*Ultra-Wideband*). Nesta situação, não seria necessário enviar múltiplos pacotes, o que reduz o *overhead* de rede gerado pelo envio do cabeçalho, em cada pacote.

Cabe ressaltar que, para algumas redes, há a fragmentação de pacotes, devido ao MTU (*Maximum Transmission Unit*), o que pode limitar esta possibilidade de melhoria. Nas redes Ethernet, por exemplo, é comum a utilização do MTU igual a 1500 bytes e também do Jumbo Frame, com 9000 bytes.

Para a programação distribuída, o envio de pacotes pode ser feito com uso do UDP ou TCP. Para a comunicação com UDP, não há garantia de entrega dos pacotes, porém, esta garantia ocorre no TCP. As considerações sobre esta garantia é abordada na próxima subseção.

E. Considerações sobre a garantia de entrega dos pacotes

Na arquitetura TCP/IP nem todos os pacotes precisam de garantia de entrega pela camada de transporte, pois algumas aplicações utilizam mecanismos próprios de controle e fazem uso do UDP para redução do custo de rede [43]. Com a evolução tecnológica das últimas décadas e o aumento gradual da qualidade dos elementos de rede, reduzindo perda de pacotes, a programação distribuída teve, consequentemente, melhorias em sua qualidade de comunicação.

Assim, é proposto que a garantia de entrega dos dados, entre estações, seja feita pela camada de enlace, a partir do uso de elementos de rede com qualidade suficiente para isto e utilizando o princípio discutido por Tanenbaum e Kurose [43]-[44], em que a camada 2 é responsável pela detecção e correção de erros no envio de dados entre estações.

Em especial, para ambientes em que a garantia de entrega tenha exigência reduzida, esta discussão possui menor impacto no contexto. Por exemplo, na programação

distribuída com o LAM MPI a exigência de garantia de entrega é parcial, conforme mostrado na seção 5 pela comparação de uso do UDP x TCP.

O endereçamento de estações/processos, controle do tamanho de pacotes e garantia de entrega em redes com conectividade em camada 2 podem ser utilizadas de forma a suportar a interoperabilidade com as arquiteturas atuais, questão esta que será discutida na próxima subseção.

F. Considerações sobre a interoperabilidade com as arquiteturas atuais

Para não haver elevado impacto nas estruturas de comunicação de rede atuais, esta proposta precisa ser implementada o mais transparentemente possível para as aplicações. Isto é possível a partir da alteração dos módulos dos Sistemas Operacionais Distribuídos (SOD's) que são responsáveis pela implementação da pilha de protocolo da arquitetura TCP/IP.

Nesta alteração, é proposto utilizar um novo Ether Type, em substituição ao 0x800 (decimal 2048). Este novo Ether Type é utilizado para identificar os pacotes com esta nova estrutura. Para utilização desse novo Ether Type, é sugerido uso do Ether Type 0x809 (decimal 2057), que está disponível para uso conforme a lista (parcial) abaixo, retirada do IANA:

| Ether type | Exp. | Ether. | Description | References |
|------------|-----------|----------|-------------------|------------|
| 2048 | 0800 | 513 1001 | Internet IPv4 | [IANA] |
| 2049 | 0801 | - - | X.75 Internet | [XEROX] |
| 2050 | 0802 | - - | NBS Internet | [XEROX] |
| 2051 | 0803 | - - | ECMA Internet | [XEROX] |
| 2052 | 0804 | - - | Chaosnet | [XEROX] |
| 2053 | 0805 | - - | X.25 Level 3 | [XEROX] |
| 2054 | 0806 | - - | ARP | [IANA] |
| 2055 | 0807 | - - | XNS Compatability | [XEROX] |
| 2056 | 0808 | - - | Frame Relay ARP | [RFC1701] |
| 2076 | 081C | - - | Symbolics Private | [DCP1] |
| 2184 | 0888-088A | - - | Xplex | [XEROX] |

Fonte: <http://www.iana.org/assignments/ethernet-numbers>

O SOD ao identificar a recepção de um pacote com o tipo 0x809, na camada de enlace, automaticamente entrega os dados para a aplicação com a correlação do endereço de processo recebido para a porta do UDP/TCP, de mesmo valor. Assim, para interoperar com as aplicações tradicionais, a informação do endereço de processo deve limitar-se ao valor 65.535, limite equivalente ao número de portas no UDP/TCP.

As aplicações construídas pelas definições desta proposta informam ao SOD suas necessidades, em OWL, com uso de Raw Socket. Desta forma elas informam que também estão preparadas para receber o fluxo de comunicação. Assim, o SOD encaminha os dados diretamente para estas, sem a necessidade de passar pela pilha tradicional do TCP/IP.

O SOD também fica responsável por traduzir os dados das aplicações tradicionais para as novas, quando os pacotes são enviados com a estrutura atual do TCP/IP. Nesta situação, o SOD encaminha os dados recebidos para as novas aplicações com a alteração das portas tradicionais para a nova forma de endereçamento, o que garante a interoperabilidade com as aplicações legadas.

A próxima seção apresenta a avaliação de desempenho desta proposta com a análise de um teste real de programação distribuída com uso do MPI (*Message Passing Interface*).

V. AVALIAÇÃO DE DESEMPENHO

Para medir, com precisão, o ganho em otimização pelo uso desta solução, é necessário verificar o impacto na comunicação de rede. Para tanto, é analisado nesta seção um caso real de programação distribuída com uso do MPI. Observe-se que outros ambientes com comportamento de rede semelhante ao da programação distribuída também apresentam ganhos similares. Por exemplo, algumas redes de voz/vídeo e de sensores que tenham conectividade em camada 2, pacotes pequenos e alternem a necessidade, ou não, de garantia de entrega dos pacotes.

A. Medida do custo de rede da programação distribuída com MPI

Nos testes realizados foram utilizadas 2 estações (1 mestre e 1 escravo), Sistema Operacional Linux versão 2.6.27.5-41.fc9.i686 e a implementação LAM para MPI, versão 7.1.4 /MPI 2, da Universidade de Indiana. Na captura dos pacotes foi utilizado o Wireshark, versão 1.0.3.

Para o envio e recepção dos dados foram utilizadas as chamadas MPI_Send e MPI_Recv, respectivamente.

No teste foi enviado um vetor de 100 posições para soma em cada escravo. Neste procedimento, foram enviados 56 pacotes entre a estação mestre e cada escravo. Todos os pacotes enviados tiveram 20 bytes de cabeçalho IP (tamanho mínimo do IHL), o que significa que não houve nenhuma informação no campo opcional. Cada comunicação ocorreu em aproximadamente 30,039 msec.

A Fig. 3 mostra o tamanho dos pacotes enviados para 1 comunicação, onde é verificada a relação entre o tamanho total do cabeçalho IP e o do UDP/TCP. A escala do eixo Y foi intencionalmente limitada em 300 para melhor visualização. Isto foi necessário porque os pacotes de 9 a 12 tiveram o tamanho muito superior à média dos demais, em decorrência de sua utilização para o envio dos dados do mestre para o escravo. De fato, estes 4 pacotes são fragmentos de 1 pacote único, com o tamanho total informado pelo UDP igual a 4.508 bytes.

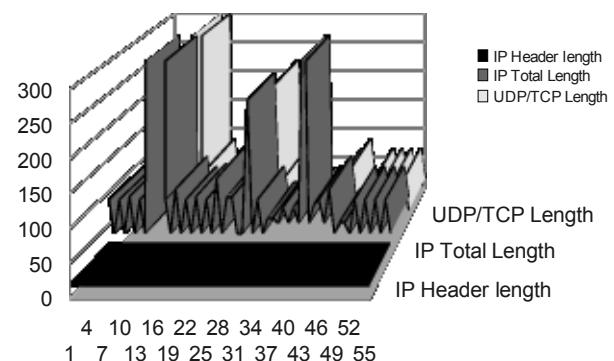


Figura 3. Tamanho dos pacotes enviados entre o mestre e o escravo.

Para os pacotes de 9 a 11 o tamanho total do IP foi de 1500 bytes, que é o valor do MTU no ambiente de teste utilizado. Cada um destes 3 pacotes possui 1480 bytes de dados e 20 bytes de cabeçalho IP. O pacote de número 12 possui 20 bytes

de cabeçalho IP acrescido de 8 bytes do cabeçalho UDP.

Exatamente 50% dos pacotes enviados foram originados pelo mestre e 50% pelo escravo. Do total de pacotes 76,8% (43) utilizaram o UDP como protocolo de transporte e os demais 23,2% (13) foram transmitidos com o TCP. Dos pacotes enviados pelo mestre, 82,1% (23) utilizaram o UDP e 17,9% (5) o TCP, enquanto que o escravo enviou 71,4% (20) pacotes com UDP e 28,6% (8) com TCP.

A Fig. 4 apresenta o gráfico da distribuição de uso do TCP e UDP entre o mestre e o escravo. Observa-se que os pacotes TCP possuem confirmação de entrega, assim, para cada pacote TCP enviado há outro de resposta.

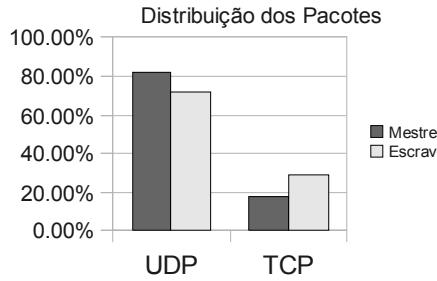


Figura 4. Distribuição dos pacotes entre o mestre e o escravo.

Por haver a confirmação de entrega no TCP, a diferença entre a quantidade enviada entre o mestre e o escravo é explicada pelo procedimento de conexão/desconexão do TCP, em 3 vias, onde o escravo iniciou e encerrou a conexão. No encerramento o escravo também enviou a confirmação do último pacote TCP recebido, ficando assim com 3 pacotes enviados a mais que o mestre.

B. Resultado da proposta de otimização do custo de rede

Para mostrar o impacto desta solução na redução do custo de rede apresenta-se, a seguir, a reconstrução do primeiro pacote, que foi enviado do mestre para o escravo. Também é apresentada a análise da otimização total para a comunicação completa da programação distribuída, realizada com o LAM MPI. Este primeiro pacote enviado possui 102 bytes, distribuídos conforme abaixo:

1) Protocolo Ethernet (14 bytes):

```
00 1a 4d a3 34 11 00 1b 24 f6 d8 14 08 00
```

2) Protocolo IP (20 bytes):

```
45 00 00 58 00 00 40 00 40 11 12 6d 0a 0a 0a 0a 0a 0b
```

3) Protocolo UDP (8 bytes):

```
82 a3 bb 29 00 44 55 92
```

4) Dados (60 bytes):

```
00 00 00 00 00 00 02 8f 00 00 00 01 40 00 00 0e 00 00 00 00
00 00 00 00 00 01 00 00 00 00 00 ff ff d9 1c 00 00 00 01
00 00 26 e4 00 00 00 05 00 00 00 05 00 00 00 05 bf 96 40 98
```

Nesta proposta há alterações nos protocolos conforme as estruturas descritas na seção 2. Para o protocolo Ethernet, há alteração do Ether Type de 0800 para 0809, conforme 2.5. Para informar a porta de destino deste pacote (bb 29 = 47913),

conforme a seção 4.C, é utilizado o valor “10” com 4 bytes para designar este número de porta. Com isto, tem-se os bytes “80 00 bb 29” pela construção binária:

```
1000 0000 0000 0000 1011 1011 0010 1001
```

Para o controle do tamanho do pacote tem-se o valor “00”, que permite informar o campo de dados com até 63 bytes (2^{6-1}). Assim, para este pacote, há 1 byte de informação, sendo este 3C:

```
0011 1100
```

Não há alteração no campo de dados e a reconstrução do pacote é feita conforme a sequência abaixo:

1) Protocolo Ethernet (14 bytes):

```
00 1a 4d a3 34 11 00 1b 24 f6 d8 14 08 09
```

2) Otimização Proposta (5 bytes):

```
80 00 bb 29 3C
```

3) Dados (60 bytes):

```
00 00 00 00 00 00 00 02 8f 00 00 00 01 40 00 00 0e 00 00 00 00
00 00 00 00 00 01 00 00 00 00 00 ff ff d9 1c 00 00 00 01
00 00 26 e4 00 00 00 05 00 00 00 05 00 00 00 05 bf 96 40 98
```

Para este pacote, há melhora no custo de rede de 102 para 79 bytes, o que representa uma redução de 22,5%, em seu tamanho. Em relação às camadas 3 e 4, esta proposta reduz 82,1% do *overhead*, já que os 28 bytes de cabeçalhos (20 do IP + 8 do UDP) são otimizados para 5 bytes. Para uso do TCP, a redução do custo de rede seria maior porque este protocolo possui cabeçalho com tamanho superior ao UDP e faz confirmação do recebimento dos pacotes.

Este cenário de programação distribuída avaliado possui estabilidade quanto às características de comunicação de rede do ambiente. Para demonstrar esta estabilidade o teste de soma de vetor foi repetido 20 vezes, com a respectiva análise da variação do custo de rede em todas as repetições do teste.

Em todos os testes, a comunicação ocorreu com a mesma quantidade de 56 pacotes e a Fig. 5 apresenta a variação do tamanho do quadro (*Frame Length*), em bytes, para os 20 testes (T1 a T20). Este gráfico comprova que não há variação significativa do custo de rede no ambiente de programação distribuída, para execuções consecutivas.

Variação do Tamanho do Quadro em Bytes

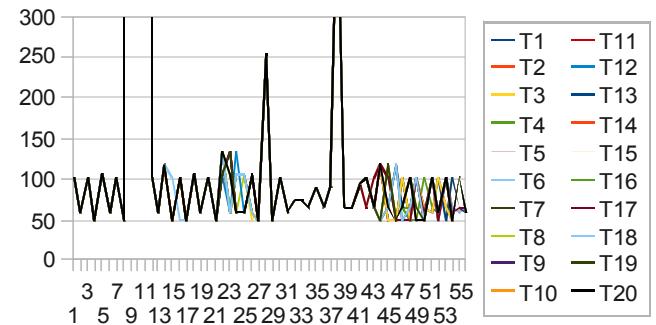


Figura 5. Variação do tamanho do quadro em 20 execuções.

O gráfico da Fig. 6 reforça esta comprovação, pois o tamanho total dos Frame Length, na somatória dos 56 pacotes,

totaliza 9.280 bytes, em todos os 20 testes realizados.

Em relação à análise sobre a inclusão de múltiplas estações na programação distribuída, também não há variação significativa no comportamento da rede, pois a comunicação de cada etapa na distribuição de tarefas normalmente é feita entre 2 estações, o mestre e o escravo.

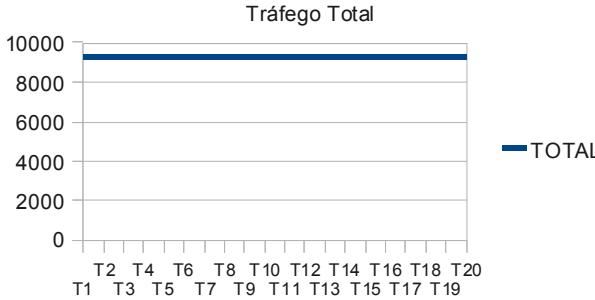


Figura 6: Somatória do tamanho do quadro na repetição de 20 testes.

Assim, a comunicação é replicada independentemente da quantidade de escravos participantes. Esta afirmação é comprovada pela análise da Fig. 7, que mostra o resultado na soma de um vetor de 300 posições de inteiros. Destes, 100 foram somados no mestre, 100 no escravo A e 100 no escravo B.

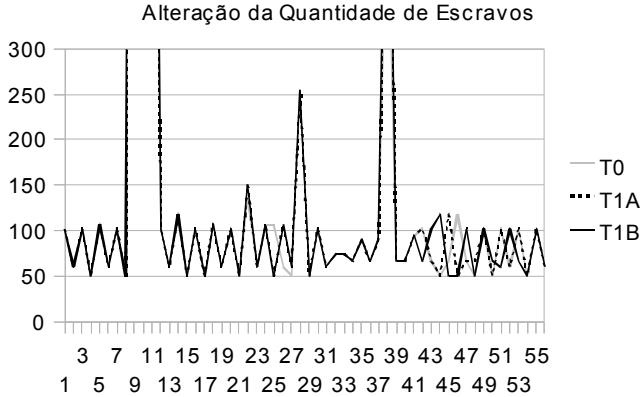


Figura 7. Análise da inclusão de novos escravos.

Neste gráfico o resultado T0 (teste 0) possui os valores do T1 (teste 1), da Fig. 5. T1A e T1B são os resultados do teste com a programação distribuída em 2 escravos (escravo A e escravo B). O comportamento dos testes T1A e T1B é próximo ao analisado para 1 escravo (T0), tanto para o tráfego em quantidade de pacotes quanto para o tamanho em bytes, o que comprova que não há variação significativa no custo de rede com o incremento de novas estações escravas.

Para esta análise também foram executados 20 testes. Observa-se que, ao incluir uma nova estação na programação distribuída, houve alteração de 2,33% sobre a quantidade total de bytes na somatória dos Frame Length, que passou de 9.280 bytes para 9.496 bytes. Esta alteração possui impacto reduzido na análise desta otimização, pois a quantidade de pacotes entre o mestre e cada um dos escravos manteve-se estável, com 56 pacotes trafegados entre estes. Consequentemente, a otimização dos cabeçalhos, em bytes, permaneceu estável.

A redução percentual do custo de rede de um pacote tem variações conforme seu tamanho. Para os testes realizados, em que o mestre enviou vetores de 100 posições para a soma em 1 escravo, a redução do custo de rede, por pacote, é apresentada na Fig. 8.

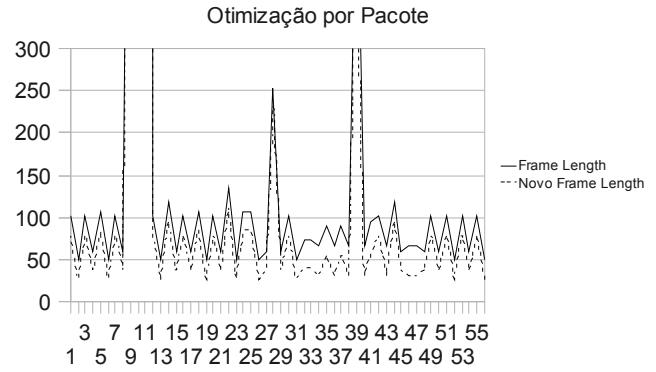


Figura 8: Redução do custo de rede por pacote.

Neste teste, com 1 escravo, os 56 pacotes enviados totalizam 9.280 bytes. Com a nova estrutura há redução para 7.867 bytes, o que representa uma melhoria de 15,23% para a quantidade de bytes enviados. Em relação ao uso dos cabeçalhos do IP e UDP/TCP, esta proposta reduz o *overhead* dos cabeçalhos de 1.724 para 291 bytes, em um total de 83,12% de otimização.

VI. CONCLUSÃO

A proposta apresentada neste artigo possibilita a redução do custo de rede para a programação distribuída em sistemas com conectividade em camada 2. Esta otimização é possível a partir do uso do Modelo de Título e de sua representação explícita com uso de ontologia, que suporta a aproximação entre a camada superior e as inferiores, na comunicação em redes de computadores.

Isto contribui para o suporte das necessidades de comunicação, que para a programação distribuída são: Endereçamento de estações; endereçamento de processos; controle do tamanho do pacote; e; garantia de entrega. Com o atendimento destas necessidades, este trabalho colabora com a melhoria de desempenho das aplicações de programação distribuída e amplia as perspectivas para uso da ontologia do Modelo de Título em outros ambientes.

Para o teste realizado com o LAM MPI, houve redução de 15,23% na quantidade de bytes trafegados, entre um mestre e um escravo, para a soma de um vetor de 100 posições. Em situações que demandam alto processamento, com envio de menor quantidade de dados por pacote, esta otimização percentual pode ser ainda maior, como no cálculo do PI com N casas decimais, por exemplo.

Para trabalhos futuros, sugere-se a expansão da proposta, testes e análise de desempenho do uso do Modelo de Título para ambientes diversos, como o de redes de sensores e *streams* de áudio e vídeo. Também sugere-se o envio de dados com tamanho gradativo, partindo de muito pequeno (1 byte)

até muito grande (Giga bytes). Outra sugestão é o estudo sobre a possibilidade de redução da quantidade de pacotes enviados, por exemplo, a retirada dos pacotes de confirmação do TCP e o uso de camadas de enlace com MTU maiores que 1500 para minimizar a fragmentação dos pacotes.

É pertinente também o estudo de um mecanismo mais eficiente para alocar as portas, para aplicações diversas comunicarem entre os mesmos mestres e escravos utilizados na programação distribuída. Por exemplo, usando o endereçamento horizontal por título de entidade e o Serviço de Domínio de Título (DTS), descrito em [36].

REFERÊNCIAS

- [1] A. S. Tanenbaum, "Sistemas Operacionais Modernos", 2^a ed, Rio de Janeiro, Prentice Hall Brasil, 2007.
- [2] J. Postel, "DoD Standard Internet Protocol", DARPA Information Processing Techniques Office, USC/Information Sciences Institute, RFC 760, 1980.
- [3] J. Postel, "DoD Standard Transmission Control Protocol", DARPA Information Processing Techniques Office, USC/Information Sciences Institute, RFC 761, 1980.
- [4] J. Postel, "User Datagram Protocol", STD 6, USC-ISI, RFC 768, 1980.
- [5] J. H. S. Pereira, S. T. Kofuji and P. F. Rosa, "Distributed Systems Ontology", New Technologies, Mobility and Security Conference – NTMS, IEEE Xplore, Cairo, 2009.
- [6] J. H. S. Pereira, S. T. Kofuji and P. F. Rosa, "Horizontal Address Ontology in Internet Architecture", New Technologies, Mobility and Security Conference – NTMS, IEEE Xplore, Cairo, 2009.
- [7] D. Comer, "Internetworking with TCP/IP Volume 1 – Principles, Protocols and Architecture", New Jersey, Prentice Hall, 1995.
- [8] S. Deering, and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", Network Working Group, RFC 1883, 1995.
- [9] S. Deering, and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", Network Working Group, RFC 2460, 1998.
- [10] R. Stewart, et. al, "Stream Control Transmission Protocol", Network Working Group, RFC 2960, 2000.
- [11] R. Stewart, Ed. "Stream Control Transmission Protocol", Network Working Group, RFC 4960, 2007.
- [12] P. F. Tsuchiya, "The Landmark Hierarchy: A new Hierarchy for Routing in Very Large Networks", ACM SIGCOMM 1988, Stanford, CA, USA, pages 35–42, 1988.
- [13] D. Krioukov, K. C. Claffy, K. Fall and A. Brady, "On Compact Routing for the Internet", ACM SIGCOMM Comput. Commun. Rev., 37(3):41–52, 2007.
- [14] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "Locator/ID Separation Protocol (LISP)", Network Working Group, draft-ietf-lisp-06, 2010.
- [15] D. Lewis, et. al, "Interworking LISP with IPv4 and IPv6", Network Working Group, draft-lewis-lisp-interworking-02, 2009.
- [16] R. Pasquini, F. L. Verdi and M. F. Magalhães, "Towards a Landmark-based Flat Routing", 27th Brazilian Symposium on Computer Networks and Distributed Systems - SBRC 2009. Recife - PE, Brazil, 2009.
- [17] R. Pasquini and M. F. Magalhães. "Flat Routing on a Binary Identity Space", Segundo Encontro dos Alunos e Docentes do Departamento de Engenharia de Computação e Automação Industrial - II EADCA. FEEC/Unicamp, 2009.
- [18] M. Caesar, "Identity-based routing", Technical Report No. UCB/EECS-2007-114 - University of California, Berkeley, USA, 2007.
- [19] M. Caesar, M. Castro, E. B. Nightingale and A. Rowstron, "Virtual Ring Routing: Net-work Routing Inspired by DHTs", In Proc. of ACM SIGCOMM, Pisa, Italy, pages 351 – 362, 2006.
- [20] M. Caesar, K. Lakshminarayanan, T. Condie, I. Stoica, J. Kannan and S. Shenker, "ROFL: Routing on Flat Labels", In Proc. of ACM SIGCOMM, Pisa, Italy, pages 363 – 374, 2006.
- [21] R. Pasquini, et. al, M. "Domain Identifiers in a Next Generation Internet Architecture", IEEE Wireless Communications & Networking Conference, 2009 (WCNC). Budapest, Hungary, 2009.
- [22] R. Pasquini e M. F. Magalhães. "Roteamento Flat para uma Arquitetura de Internet de Próxima Geração baseada em Identificadores de Domínios", Primeiro Encontro dos Alunos e Docentes do Departamento de Engenharia de Computação e Automação Industrial - I EADCA. FEEC/Unicamp, 2008.
- [23] B. Ford, UIA: A Global Connectivity Architecture for Mobile Personal Devices. Ph.D. Thesis. Massachusetts Institute of Technology (MIT), Dep. of Electrical Eng. and Comp. Sci, 2008.
- [24] B. Ford, Scalable Internet Routing on Topology-Independent Node Identities. MIT technical report conducted as part of the IRIS project (<http://project-iris.net>), 2003.
- [25] W. Wong, et. al, "An Architecture for Mobility Support in a Next Generation Internet", The 22nd IEEE International Conference on Advanced Information, Networking and Applications (AINA 2008). Okinawa, Japan, 2008.
- [26] W. Wong, F. L. Verdi and M. F. Magalhães, "A Next Generation Internet Architecture for Mobility and Multi-homing Support", 3rd International Conference on emerging Networking EXperiments and Technologies. Student Workshop. New York, NY, 2007.
- [27] E. S. Santos, F. S. F. Pereira, J. H. S. Pereira, P. F. Rosa e S. T. Kofuji, "Optimization Proposal for Communication Structure in Local Networks", International Conference on Networking and Services – ICNS, IEEE Xplore, Mexico, 2010.
- [28] F. S. F. Pereira, E. S. Santos, J. H. S. Pereira, P. F. Rosa e S. T. Kofuji, "FINLAN Packet Delivery Proposal in a Next Generation Internet", International Conference on Networking and Services – ICNS, IEEE Xplore, Mexico, 2010.
- [29] J. H. S. Pereira, E. S. Santos, F. S. F. Pereira, P. F. Rosa e S. T. Kofuji, "Layers Optimization Proposal in a Post-IP Network", International Journal on Advances in Networks and Services, vol 3, no 4 February, 2011, in Press.
- [30] L. M. Correia e L. Lundgren eds., "4WARD-Architecture and Design for the Future Internet", Newsletter Issue No. 6, 2009.
- [31] T. Zahariadis ed. "Why do we need a Content-Centric Future Internet? Proposals towards Content-Centric Internet Architectures" Future Content Networks Group, Prague, 2009.
- [32] P. Almquist, "Type of Service in the Internet Protocol Suite", Network Working Group, RFC 1349, 1992.
- [33] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", Network Working Group, RFC 2474, 1998.
- [34] ITU-T Recommendation X.207 | ISO/IEC 9545:1993, "Information Technology – Open Systems Interconnection – Application Layer Structure", 1993.
- [35] L. W. Lacy, "Owl: Representing Information Using the Web Ontology Language", Trafford, 2005.
- [36] J. H. S. Pereira, P. F. Rosa e S. T. Kofuji, "Horizontal Addressing by Title in a Next Generation Internet", International Conference on Networking and Services – ICNS, IEEE Xplore, Mexico, 2010.
- [37] H. Hegering, and A. Läpple, "Ethernet – Building a Communications Infrastructure", Addison-Wesley, Munich, 1993.
- [38] J. Arkko, and C. Pignataro, "IANA Allocation Guidelines for the Address Resolution Protocol (ARP)", RFC 5494, 2009.
- [39] S. Cheshire, "IPv4 Address Conflict Detection", RFC 5227, 2008.
- [40] D. Plummer, "Ethernet Address Resolution Protocol: Converting Network Protocol Addresses to 48-bit Ethernet Address for Transmission on Ethernet Hardware", RFC 826, 1982.
- [41] J. Postel, ed. "Internet Protocol", DARPA Internet Program Protocol Specification, USC/Information Sciences Institute, RFC 791, 1981.
- [42] J. Postel, ed. "DoD Standard Transmission Control Protocol", DARPA Information Processing Techniques Office, RFC 793, 1981.
- [43] A. S. Tanenbaum, and V. Maarten, "Sistemas Distribuídos", 2^a ed, São Paulo, Prentice Hall Brasil, 2007.
- [44] J. F. Kurose, K. W. Ross, "Computer Networking – A Top-Down Approach", 4th ed, Boston, Pearson Addison-Wesley, 2008.



João Henrique de Souza Pereira Possui graduação em Bacharelado em Ciência da Computação pelo Centro Universitário do Triângulo (2000), especialização em MBA Em Gestão Empresarial pela Fundação Getúlio Vargas - RJ (2003), especialização em MBA em Gestão de Pessoas pela Universidade Federal de Uberlândia (2006), mestrado em

Ciência da Computação pela Universidade Federal de Uberlândia (2004) e curso-técnico profissionalizante pela Escola Técnica de Eletrônica Francisco Moreira da Costa (1994). Atualmente é Coordenador de Configuração de Redes da Algar Telecom e Professor do Centro Universitário do Triângulo. Tem experiência na área de Ciência da Computação, com ênfase em Sistemas de Computação. Atuando principalmente nos seguintes temas: Rede de Próxima Geração, Redes Convergentes, Próxima Geração de Internet. Premiado com melhor artigo no Congresso da Sociedade Brasileira de Computação – *Workshop* em Sistemas Operacionais (CSBC-WSO), em 2009, e IEEE *best papers* no *New technologies, Mobility and Security* (NTMS), em 2009, e no *International Conference in Networking and Services* (ICNS), em 2010. *Silver paper* no *International Information and Telecommunication Technologies Symposium* (I2TS), em 2010.



Liria Matsumoto Sato Possui graduação em EESC pela Universidade de São Paulo (1977), mestrado em Engenharia Eletrônica e Computação pelo Instituto Tecnológico de Aeronáutica (1983) e doutorado em Engenharia Elétrica pela Universidade de São Paulo (1989). Atualmente é Professora Associada da Universidade de São Paulo. Tem experiência na área de Ciência da Computação, com ênfase em Sistemas de Computação. Atuando principalmente nos seguintes temas: Alto Desempenho, *cluster* de computadores, Linguagens de Programação Paralela. Premiada com *silver paper* no *International Information and Telecommunication Technologies Symposium* (I2TS), em 2010.



Pedro Frosi Rosa Possui Doutorado em Engenharia de Computação por um acordo de cooperação internacional entre Escola Politécnica da Universidade de São Paulo e o *Centre National de la Recherche Scientifique* (CNRS-France 1995) na área de sistemas distribuídos no *Laboratoire d'Analyse et d'Architecture des Systèmes* (LAAS)/Toulouse e Mestrado em Engenharia de Computação pela Escola Politécnica da Universidade de São Paulo (1990) na área de Arquitetura de Redes de Computadores. Atualmente é Professor Associado da Universidade Federal de Uberlândia na Faculdade de Computação. Tem experiência nas áreas de Ciência e Engenharia da Computação, com ênfase em Arquiteturas Altamente Escaláveis e de Alta Disponibilidade, Projeto e Validação de Protocolos para Computadores e Computação em Nuvem atuando principalmente nos seguintes temas: Telecomunicação (VoIP, NGN e IMS), Autômatos Escaláveis, Web Services, Protocolos, CMMI e CSCW. Premiado com IEEE *best papers* no *New technologies, Mobility and Security* (NTMS), em 2009, e no *International Conference in Networking and Services* (ICNS), em 2010. *Silver paper* no *International Information and Telecommunication Technologies Symposium* (I2TS), em 2010.



Sergio Takeo Kofuji Possui graduação em Bacharelado em Física pela Universidade de São Paulo (1985), Mestrado em Engenharia Elétrica pela Universidade de São Paulo (1988) e Doutorado em Engenharia Elétrica pela Universidade de São Paulo (1995). Atualmente é Professor Doutor RIDP da Escola Politécnica da Universidade de São Paulo. Tem experiência na área de Engenharia Elétrica e Ciência da Computação, com ênfase em Arquiteturas Avançadas de Computadores, atuando principalmente nos seguintes temas: Computação Pervasiva, Redes de Sensores Sem Fio, Grades Computacionais de Armazenamento (*DataGrids*, *GridServices*), Processamento Paralelo, Processadores (SMT, CMP, PIM), Simuladores de Processadores (Programação no Cell), Arquitetura Reconfigurável, Sistemas Ciber-Físicos e Embarcados, UWB e ondas milimétricas, Computação Móvel e Sem Fio, e Redes de Alta Velocidade. Premiado com IEEE *best papers* no *New technologies, Mobility and Security* (NTMS), em 2009, e no *International Conference in Networking and Services* (ICNS), em 2010. *Silver paper* no *International Information and Telecommunication Technologies Symposium* (I2TS), em 2010.