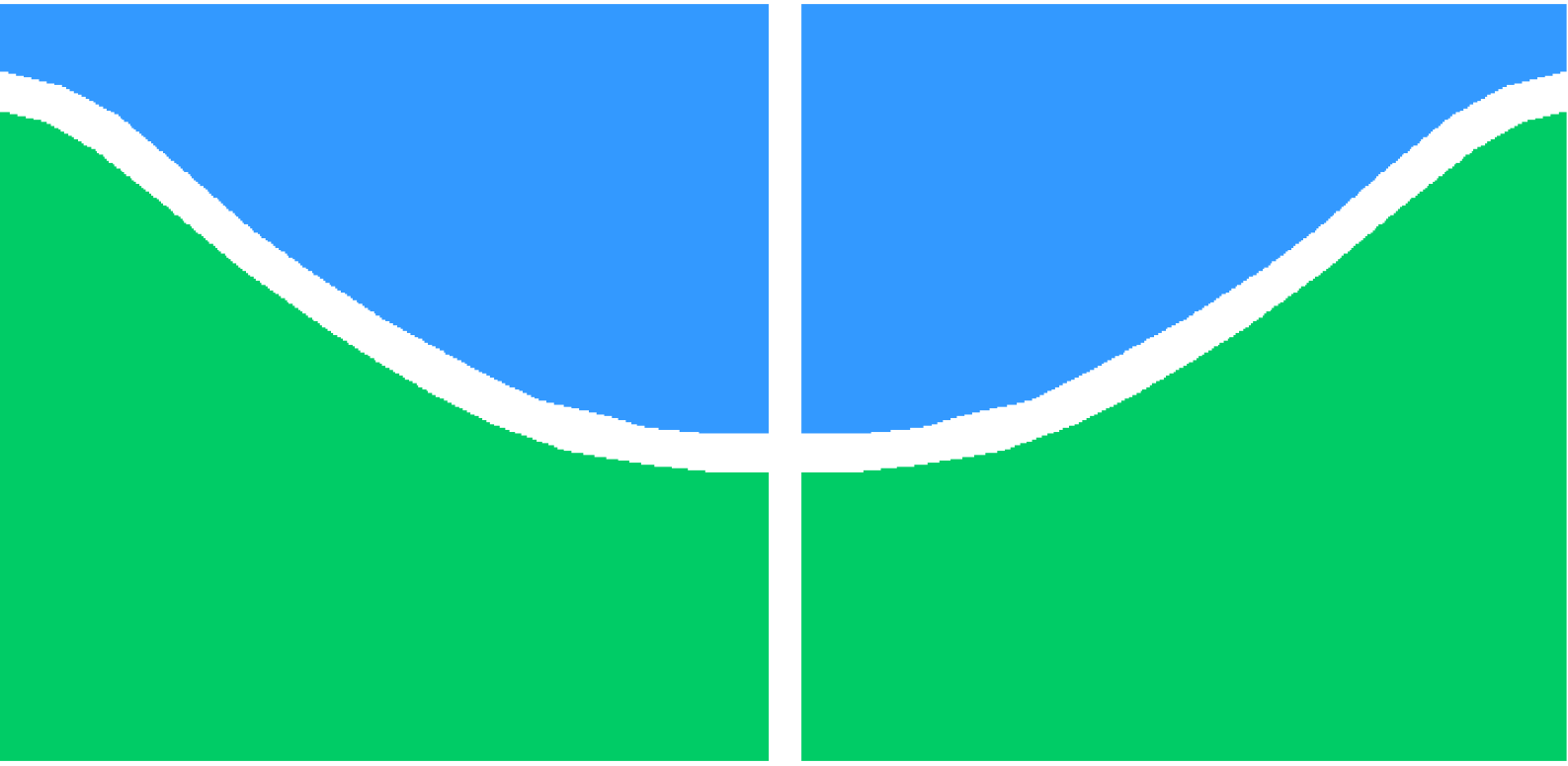

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Implementação de protocolo de federação para redes sociais

Autor: Gabriel dos Santos Silva
Orientador: Prof. Dr. Paulo Roberto Miranda Meirelles

Brasília, DF
2016

Gabriel dos Santos Silva

Implementação de protocolo de federação para redes sociais

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Paulo Roberto Miranda Meirelles

Brasília, DF

2016

Gabriel dos Santos Silva
Implementação de protocolo de federação para redes sociais/ Gabriel dos Santos Silva. – Brasília, DF, 2016-
47 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Paulo Roberto Miranda Meirelles

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2016.

1. Federação. 2. Redes Sociais. I. Prof. Dr. Paulo Roberto Miranda Meirelles. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Implementação de protocolo de federação para redes sociais

CDU 02:141:005.6

Gabriel dos Santos Silva

Implementação de protocolo de federação para redes sociais

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Trabalho aprovado. Brasília, DF, 08 de Dezembro de 2016:

Prof. Dr. Paulo Roberto Miranda Meirelles
Orientador

Dr. Antonio Soares de Azevedo Terceiro
Convidado 1

Dr. Tiago Alves da Fonseca
Convidado 2

Brasília, DF
2016

Resumo

A federação de redes sociais propõe a integração de usuários através de uma estrutura descentralizada, possibilitando a interoperabilidade entre serviços distintos de maneira transparente usando de protocolos de comunicação. Apesar da iniciativa da comunidade em federar redes sociais livres, não existe adoção a uma especificação padrão, o que dificulta o surgimento de sistemas federados. De forma a entender as dificuldades envolvidas na construção e padronização de serviços federados, foi realizado um estudo a respeito das especificações e implementações de interoperabilidade existentes no cenário de redes sociais livres. Em segundo lugar, foi desenvolvida uma prova de conceito no caso específico da plataforma Noosfero, onde foi implementada uma parte do protocolo Diaspora para a federação de usuários e conteúdos públicos, o que inclui a utilização de especificações intermediárias, como o WebFinger e o Salmon. Como resultado, foi possível implementar parte do protocolo Diaspora no Noosfero, evidenciando as medidas a serem tomadas na continuidade do desenvolvimento.

Palavras-chaves: federação. redes sociais. Noosfero. Diaspora.

Abstract

The federation of social networks proposes to integrate users through a decentralized structure, making the interoperability between distinct services possible in a transparent way by using communication protocols. Despite the initiative in the community to federate open social networks, there is no adoption of any standard, which hinders the emergence of new federated systems. In order to understand the difficulties in the development and standardization of federated services, it was conducted a research regarding the existing specifications and implementations of interoperability between social networks. Moreover, it was developed a proof of concept of federation within the Noosfero platform, implementing a subset of the Diaspora protocol to federate users and public content, in addition to complementary specifications, such as Salmon and WebFinger. As results, it was possible to federate Noosfero with Diaspora networks, pointing the steps to be taken before further development.

Key-words: federation. social networks. Noosfero. Diaspora.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

IETF	*Internet Engineering Task Force*
TCP	*Transmission Control Protocol*
IP	*Internet Protocol*
PubHubSub	PubHubSubbub
HTTP	*Hypertext Transfer Protocol*
JSON	*Javascript Object Notation*
LRDD	*Link-based Resource Description Discovery*
HTML	*Hyper Text Markup Language*
API	*Application Programming Interface*
POP	*Post Office Protocol*
SMTP	*Simple Mail Transfer Protocol*
SSL	*Secure Sockets Layer*

Sumário

1 INTRODUÇÃO

Pode-se definir mídias sociais como aplicações da Internet que permitem que indivíduos mantenham um conjunto de conexões com outros usuários, e compartilhem informações pessoais e conteúdos gerados através de redes digitais (BOYD; ELLISON, 2007). Tais redes sociais são baseadas no fluxo de informações que nem sempre são públicas, e geralmente acontece através de infraestruturas privadas de entidades interessadas em prover o serviço, como por exemplo o Facebook e o Google.

Em segurança computacional, pode-se definir privacidade como a capacidade de um indivíduo controlar quais e como as informações relacionadas a ele podem ser consumidas ou armazenadas, e para quais indivíduos estes dados podem ser divulgados. (STALLINGS, 2010). Além de garantir o direito à privacidade de seus usuários, as redes sociais devem garantir a confidencialidade das informações que hospeda, assegurando que dados privados são sejam expostos a indivíduos não autorizados.

Argumenta-se que o poder de um único grande provedor sobre um fluxo de dados privados coloca em risco a privacidade de seus usuários, já que não garante a confidencialidade das informações privadas. As críticas a estes provedores geralmente é acentuada pela falta de transparência da utilização dessas informações, que geralmente assume um papel comercial, como na identificação de padrões de comportamento em suporte à publicidade.

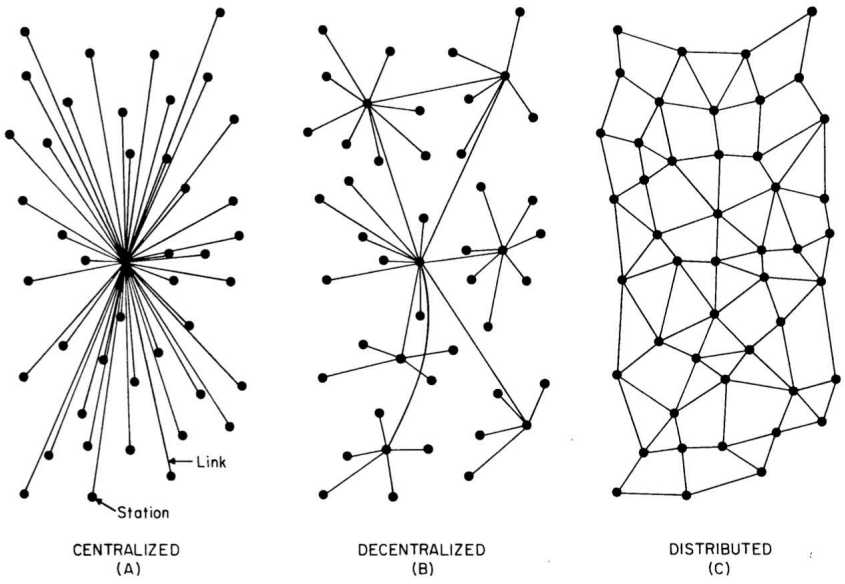

Figura 1 – Tipos de redes de comunicação (BARAN, 1964)

Uma alternativa à centralização do fluxo de dados privados é baseada no conceito de redes descentralizadas, um dos padrões de organização de redes de comunicação. Indica-

se que em relação à organização de seus nós, redes de comunicação podem assumir uma organização centralizada ou distribuída (BARAN, 1964). Como pode ser visto na Figura 1, em contraste a redes centralizadas, uma rede distribuída não depende de um único servidor central.

Redes sociais privadas se assemelham à configuração de redes centralizadas, onde o nó central representa o servidor, e cada um dos conectados um dos milhões de clientes. Uma alternativa comum a esta configuração é a organização descentralizada, que substitui um único nó central com servidores intermediários, conectados entre si para o estabelecimento intercomunicação.

Alguns autores também definem esta organização como uma rede federada, que é similar à definição de redes descentralizadas de (BARAN, 1964), mas também é definida na literatura como um conjunto de implementações interoperáveis que respeitam o modelo cliente-servidor (BAROCAS et al., 2012). Esta definição é importante por que generaliza o conceito de federação para outros tipos de sistemas de comunicação que não redes literais de computadores, e indica a propriedade de extensibilidade — qualquer entidade capaz que garanta a interoperabilidade pode ser parte da federação.

A distribuição dos serviços contrapõe a existência de um único provedor, comum em redes sociais fechadas como o Facebook e Twitter. A descentralização garante que o armazenamento das informações não está restrito a apenas um proprietário. Mais importante, a independência de extensão permite que qualquer indivíduo inclua um novo servidor na federação, desde que respeite os critérios de interoperabilidade, hospedando seus próprios dados. A descentralização do fluxo de dados privados distribui a responsabilidade pela manutenção de confidencialidade, o que passa a não depender de uma única entidade com intenções inexplícitas.

Por outro lado, a federação é fundada na capacidade de interoperabilidade entre sistemas. Visto que o cenário provável envolve a comunicação entre sistemas distintos em vários aspectos, é essencial considerar protocolos e padrões.

O objetivo deste trabalho é estudar a federação no contexto de redes sociais, investigando quais os aspectos envolvidos na interoperabilidade destas mídias, e o estado do esforço de padronização da interoperabilidade. Um segundo objetivo é aprofundar esta análise no caso específico do Noosfero, uma rede social livre, projetando e implementando uma prova de conceito de federação.

A utilização e padronização de protocolos de comunicação na interoperabilidade de sistemas é introduzida no Capítulo 2, que aborda a utilização de especificações na federação de sistemas, apresentando as iniciativas de padronização nesse contexto. O Capítulo 3 descreve o Noosfero e retrata o estado da federação até a publicação deste trabalho, propondo a evolução da federação com outras redes através de implementação

com base no protocolo Diaspora. Por fim, o Capítulo 4 apresenta os resultados de parte dessa implementação, enquanto o Capítulo 5 conclui o trabalho apontando os próximos passos para o desenvolvimento do projeto.

2 PROTOCOLOS DE COMUNICAÇÃO

A comunicação envolve a troca de uma série de mensagens entre duas entidades. Parte fundamental para a comunicação bem sucedida é a premissa a respeito das informações transmitidas e recebidas, como por exemplo a linguagem e o meio de transmissão (COMER, 2000). Caso as entidades envolvidas na comunicação não concordem em relação a estas premissas, não será possível estabelecer um diálogo adequada.

Restrições a respeito do formato, meios de transmissão, e ações a serem tomadas no envio e recebimento de mensagens são definidas por meio de protocolos. A partir do momento em que duas entidades sigam o mesmo protocolo, pode-se garantir que a comunicação será estabelecida. Assim como qualquer outra entidade, componentes de *hardware* e *software* também estão sujeitos a protocolos de comunicação (KUROSE; ROSS, 2012).

Antes de abordar os protocolos de federação, é necessário entender a necessidade de protocolos de comunicação a partir da utilização em redes de computadores. Também se faz necessário discutir o processo de padronização desses protocolos, o que deve ajudar na discussão a respeito da definição e utilização de padrões na construção de sistemas federados.

2.1 SUÍTES DE PROTOCOLOS

No contexto da computação, redes de comunicação são construídas com diferentes tecnologias de acordo com necessidades e restrições específicas, o que prejudica a capacidade de intercomunicação entre dispositivos (COMER, 2000). A Internet, por exemplo, é uma coleção de redes menores que eventualmente utilizam tecnologias diferentes, como é o caso de linhas telefônicas, e transmissão a rádio (TANENBAUM; WETHERALL, 2010). Um desafio diferente é alcançar a intercomunicação em um sistema complexo como tal, onde as redes que o compõem utilizam seus próprios protocolos, desta vez específicos ao meio de transmissão.

De acordo com (COMER, 2000), existem duas observações fundamentais ao projeto de redes de comunicação:

1. Não existe nenhuma tecnologia de rede capaz de satisfazer às restrições de todos os possíveis contextos;

2. Usuários desejam intercomunicação universal.

A primeira observação sugere que a necessidade de interoperabilidade eventualmente pode surgir entre sistemas incompatíveis. Ainda assim, a discrepância deve ser invisível ao usuário, que de qualquer forma espera a interoperabilidade, ponto reafirmado pela segunda observação.

Esse tipo de interoperabilidade pode ser implementada tanto no nível das aplicações quanto no nível da rede. Enquanto a primeira estratégia supõe que as aplicações prevejam explicitamente suporte a cada uma das tecnologias, a segunda estratégia é mapeada desde o *hardware*, providenciando uma camada de abstração para os componentes utilitários.

A intercomunicação no nível de rede (ou simplesmente internet (COMER, 2000)) também pode empregar um modelo de camadas. Neste caso, os protocolos são organizados em uma hierarquia vertical, de acordo com seus objetivos. A interface entre cada camada é bem definida, o que concede modularidade ao sistema (KUROSE; ROSS, 2012).

Neste tipo de hierarquia as mensagens são enviadas sucessivamente da camada mais superior até a camada mais inferior durante a transmissão, o inverso do que acontece no recebimento. Apenas as camadas mais inferiores do transmissor e destinatário são conectadas por meio de um meio de transmissão arbitrário, o que garante abstração do meio para as camadas superiores.

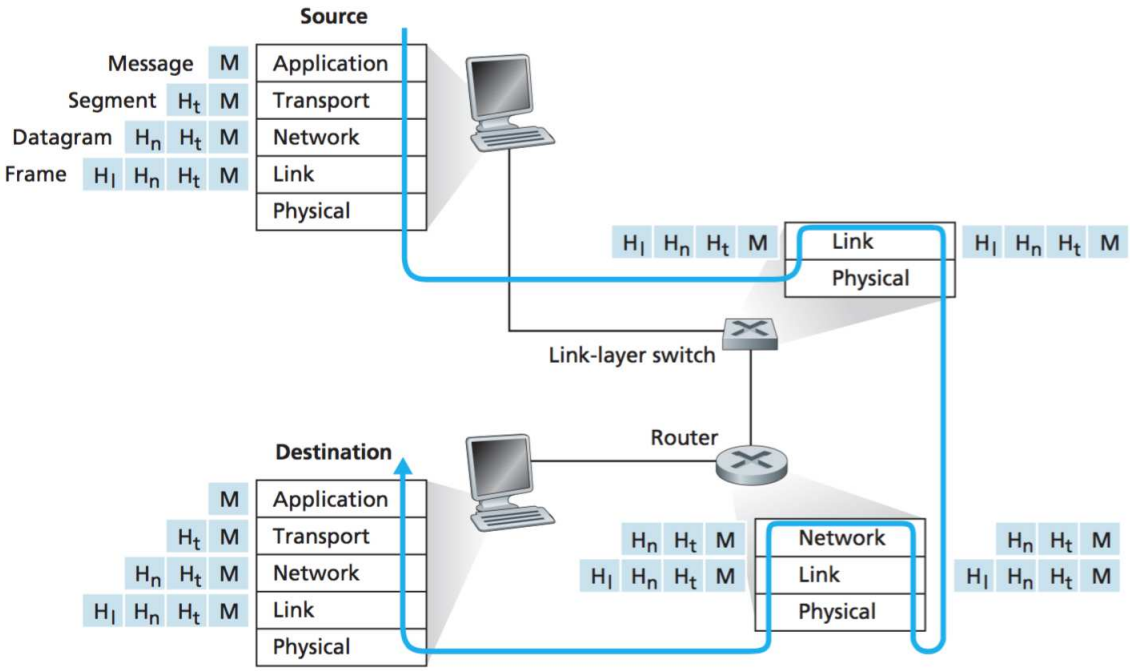

Figura 2 – Encapsulamento de mensagens em um modelo de camadas (KUROSE; ROSS, 2012)

A Figura 2 mostra este processo no contexto do TCP/IP, também conhecido como a suíte de protocolos da Internet. Neste caso, os protocolos de cada camada adicionam um

cabeçalho às mensagens recebidas do nível anterior, o que permite a adição de informações adicionais destinadas ao mesmo protocolo implementado no comunicante.

O OStatus é outro exemplo de uma suíte de protocolos, por sua vez proposto para a federação de aplicações, que envolve alguns padrões intermediários na construção de implementações independentes e interoperáveis. Apesar de não adotar explicitamente um modelo de camadas, também exemplifica a importância dos conceitos de hierarquia e encapsulamento na garantia de interoperabilidade.

2.2 PADRONIZAÇÃO DE PROTOCOLOS

A definição de protocolos é apenas o primeiro passo para a intercomunicação de sistemas. Se não existir um acordo a respeito da especificação utilizada será mais difícil estabelecer a comunicação entre serviços (KUROSE; ROSS, 2012). A padronização garante o estabelecimento deste acordo.

O conceito de efeito de rede indica que a adoção de um protocolo se torna mais valiosa à medida que um maior número de entidades também o utilize (LIEBOWITZ; MARGOLIS, 1998). Justifica-se portanto o interesse em incentivar a padronização de protocolos.

A partir do momento em que uma série de entidades entra em consenso a respeito da especificação de um protocolo, estabelece-se um padrão *de facto*. A iniciativa de padronização também pode surgir através de entidades regulamentadoras, como a Organização Internacional para a Padronização (ISO), ou a *Internet Engineering Task Force* (IETF), o que leva ao estabelecimento padrões *de jure* (TANENBAUM; WETHERALL, 2010).

O processo de definição de novos padrões *de jure* depende da entidade regulamentadora relacionada, e ocasionalmente parte de padrões *de facto* já utilizados na comunidade. Geralmente uma especificação é proposta, discutida, e revisada pela entidade antes de se tornar um padrão, o que no caso da ISO pode levar de seis meses a alguns anos (TANENBAUM; WETHERALL, 2010).

Propostas e padrões estabelecidos devem ser documentados de alguma forma. A IETF adota o formato de *Request for Comments* (RFC), publicações que descrevem completamente uma especificação, e estão disponíveis a consulta pela comunidade.

Uma proposta deve cumprir uma série de requisitos antes de ser endossado por uma organização. No caso da IETF, cada proposta passa por vários níveis de maturidade até alcançar a categoria de padrão. Cada um destes níveis pode ser alcançado ao satisfazer as recomendações de determinados grupo da comunidade. Um exemplo destes requisitos é a exigência de uma prova de conceito de interoperabilidade, como uma implementação de referência entre duas ou mais entidades distintas (POSTEL, 1992).

Protocolos de federação também estão sujeitos ao efeito de rede, e apresentam as mesmas necessidades de padronização. O *Simple Mail Transfer Protocol* é um exemplo de especificação utilizado na interoperabilidade de sistemas federados que passou por um esforço oficial de padronização, tornando-se um caso interessante na análise deste processo.

2.2.1 Simple Mail Transfer Protocol

Um sistema de *e-mails* pode ser considerado federado, já que respeita a definição de implementações interoperáveis no modelo cliente servidor apresentada por (BAROCAS et al., 2012). O SMTP é um dos protocolos utilizados na implementação de interoperabilidade entre serviços distintos.

Trata-se de um protocolo para o transporte e entrega de mensagens de e-mail entre processos. A especificação garante que a troca de mensagens aconteça entre clientes que se localizam em redes diferentes, o que permite a construção de um serviço que funcione de maneira confiável sobre a internet (KLENSIN, 2001).

Caracterizado como um protocolo orientado a conexões entre clientes e servidores, ou transmissores e receptores, o SMTP é guiado por uma série de comandos predefinidos. Os servidores também são responsáveis por retransmitir mensagens, caso não sejam os destinatários finais (KUROSE; ROSS, 2012).

A troca de mensagens geralmente acontece em um único salto após o estabelecimento de uma conexão orientada entre o remetente e o destinatário. A retransmissão de mensagens é uma alternativa, utilizada por exemplo em casos em que um usuário moveu sua caixa de e-mails de um servidor para outro e deseja receber as mensagens no seu novo endereço.

Como pode ser visto na Figura 3, o SMTP é um protocolo intermediário entre servidores de e-mail que alternam entre os papéis de transmissor e receptor. Cada um destes servidores fornece a seus próprios clientes, constituindo sistemas menores onde a interação não é necessariamente coberta pela especificação do SMTP.

Cada um destes sistemas intermediários não é necessariamente compatível, visto que a comunicação entre o servidor e o usuário depende das aplicações envolvidas, e eventualmente utiliza outros padrões, como por exemplo POP3 ou IMAP no gerenciamento de caixas de e-mail pessoais (TANENBAUM; WETHERALL, 2010).

Por se tratar de uma especificação amplamente utilizada na Internet, o SMTP se tornou foco de grupos de trabalho da IETF, passando pelo processo de padronização formal desta entidade, incluindo a publicação em formato de RFC. Trata-se de um padrão *de jure* aberto, definido por uma organização antes de ser adotado pelo mercado.

Considerando a descentralização da arquitetura e a padronização do protocolo,

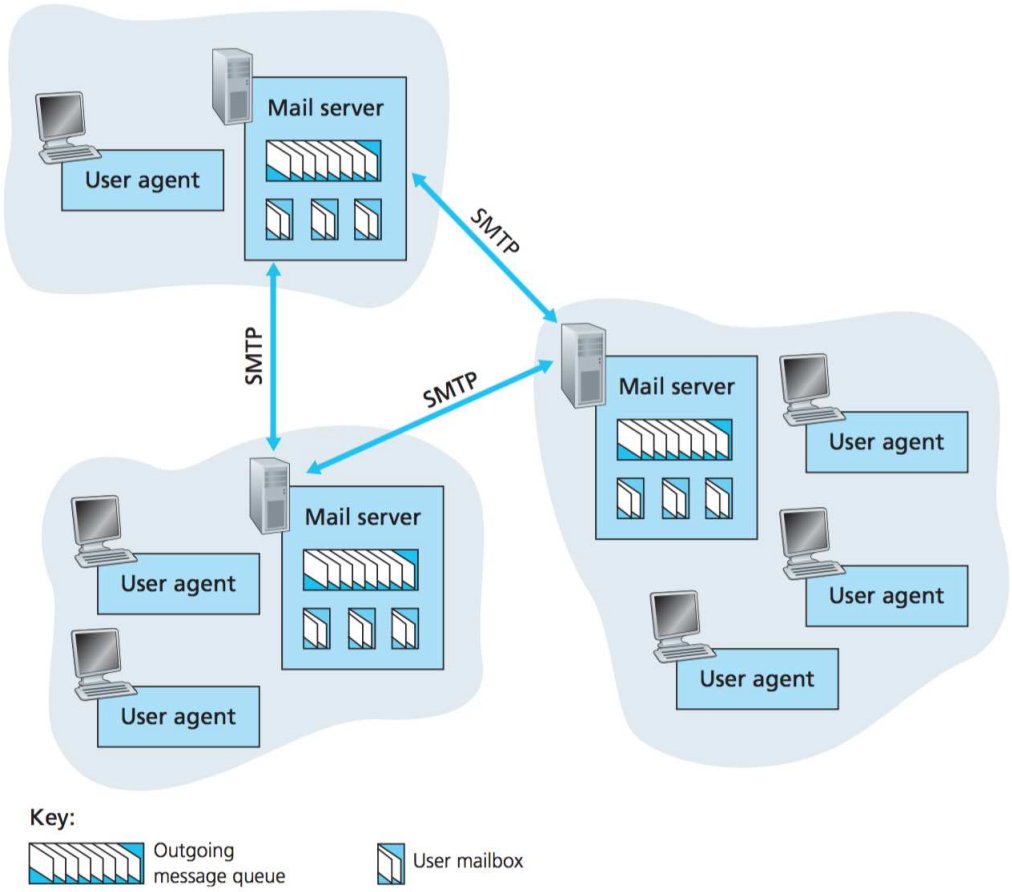

Figura 3 – Uma visão geral de um sistema de e-mails (KUROSE; ROSS, 2012)

qualquer novo sistema é capaz de implementar e se tornar parte da federação com relativa facilidade. Nestes casos a heterogeneidade entre os sistemas intermediários é completamente transparente para os usuários do serviço, o que o indica como um protocolo de federação bem sucedido.

2.3 PROTOCOLOS DE FEDERAÇÃO

A inexistência de um protocolo padrão para a federação de mídias sociais dificulta o desenvolvimento de aplicações interoperáveis. São enfrentados os mesmos problemas identificados na intercomunicação de redes heterogêneas, já que os desenvolvedores solucionam os problemas com tecnologias diferentes, de acordo com as próprias restrições e necessidades.

Ainda que exista uma tendência da comunidade em adotar propostas de protocolos, o que pôde ser observado no caso do projetos OStatus e Diaspora, abordagens alternativas naturalmente serão desenvolvidas. A ausência de um protocolo bem difundido segmenta a adoção por parte dos desenvolvedores, o que prejudica o efeito de rede e impede o surgimento de um padrão *de facto*.

Fazendo um paralelo com o SMTP, o mais próximo a uma entidade como IETF no caso dos protocolos de federação para redes sociais é o *World Web Consortium* (W3C), que também endorsa padrões oficialmente por meio de relatórios técnicos. O processo de padronização tem início com a criação de um grupo de trabalho que deve produzir rascunhos da especificação, sujeitos à revisão da comunidade pública e dos membros do W3C. Se um rascunho é revisado o suficiente e cumpre os critérios da comunidade, pode eventualmente se tornar uma recomendação do consórcio (NEVILE; JACOBS, 2015).

Ao discutir a padronização da federação é interessante não só analisar o esforço da comunidade em discutir e adotar propostas ainda não completamente estabelecidas, como também identificar as questões que impediram o acordo. Não existem referências na literatura que tratam das dificuldades no estabelecimento de padrões para a federação, e a discussão da comunidade está espalhada por listas de discussão e repositórios de projetos individuais. Antes de abordar este debate, é importante apresentar os projetos de federação que ganharam maior visibilidade, o OStatus e o Diaspora.

2.3.1 OStatus

O OStatus[1] é um conjunto de protocolos que permite interação em tempo real entre redes redes sociais. Foi inicialmente proposto por Evan Prodromou para a implementação do StatusNet, que posteriormente deu origem ao projeto GNU Social[2].

Até certo ponto, tecnologias como o Atom e RSS permitem a interação entre sistemas, mais especificamente no compartilhamento de conteúdo. No entanto, são restritas no que diz respeito a interação em tempo real, problema que o OStatus propõe resolver combinando *feeds* Atom com uma série de outros mecanismos.

A especificação do OStatus propõe a utilização das tecnologias listadas a seguir.

- PubSubHubbub: mecanismo de inscrição e submissão
- WebFinger: descoberta de identidade
- Activity Streams: representação de atividades de usuários em redes sociais
- Salmon: descentralização de trocas de mensagens

O projeto obteve visibilidade na comunidade, contando com uma das maiores iniciativas de padronização. O OStatus chegou a se tornar alvo de um grupo de trabalho do W3C em 2012, uma das maiores organizações de padronização para a *web*. Apesar disto, o grupo não apresentou avanços desde então, e o desenvolvimento do projeto está estagnado.

[1] <https://www.w3.org/community/ostatus>
[2] <https://gnu.io/social>

2.3.1.1 PubSubHubbub

O PubHubSub[3] especifica um sistema distribuído para publicação e assinatura de conteúdos. A ideia é que serviços possam se inscrever em diretórios centrais (ou *hubs*), expressando o interesse em receber as atualizações em tempo real. Este padrão complementa tecnologias como o RSS, que baseia a atualização apenas na solicitação dos interessados.

Serviços inscritos devem identificar o tópico desejado através de URIs, e oferecer um servidor disponível pela internet para que a notificação seja realizada. É de responsabilidade de cada serviço notificar os *hub* conhecidos a cada nova publicação. Cada *hub* é responsável por replicar a mensagem a todos os serviços inscritos.

2.3.1.2 WebFinger

O WebFinger é um protocolo de descoberta de identidade que soluciona o problema do compartilhamento de informações de usuários entre servidores remotos (JONES G. SALGUEIRO; SMARR, 2013). A proposta é que a partir de um atributo identificador e do endereço do servidor de origem, seja possível garantir existência de um usuário e recuperar suas informações públicas.

A especificação do WebFinger propõe que todos os recursos possam ser identificados por uma URL, e que todas as solicitações e respostas sejam realizadas através de requisições HTTP. Servidores que forneçam informações através do WebFinger devem responder aos *endpoints* definidos no protocolo com objetos JSON. Servidores que pretendam consumir tais informações precisam conhecer apenas a URL do recurso de interesse, o que pode ser encontrado a partir do identificador do usuário e do domínio do servidor de origem (em um processo que pode ser classificado como LRDD (HAMMER-LAHAV, 2010)).

2.3.1.3 Activity Streams

O Activity Streams[4] é uma especificação que propõe um formato para a representação das atividades dos usuários em uma rede social. O objetivo é facilitar o consumo destas ações para o resto da rede e serviços externos.

A especificação mais atual do Activity Streams propõe a representação em formato JSON, definindo propriedades que identificam a ação, o usuário responsável, e a entidade alvo. Especificações mais antigas utilizam o padrão Atom, com restrições semelhantes em relação ao conteúdo das mensagens. O OStatus adota a especificação baseada em Atom, utilizando mensagens em formato XML.

[3] <https://github.com/pubsubhubbub/PubSubHubbub>
[4] <http://activitystrea.ms/>

O padrão trata apenas da representação das mensagens, dependendo de outros mecanismos para a distribuição das atividades. Geralmente são utilizadas APIs ou *feeds* em conjunto com mecanismos de inscrição.

2.3.1.4 Salmon

Combinar soluções como Atom e PubHubSub permite que conteúdos sejam publicados e atualizados em tempo real. No entanto, a partir do ponto em que um conteúdo pode ser consumido e atualizado a partir de um número arbitrário de serviços, também deve-se investir esforço para garantir que o seu estado seja o mesmo em toda a rede.

O objetivo do Salmon[5] é descentralizar a capacidade de contribuir com o estado de um conteúdo, mas mantendo sua consistência por toda a federação. Para isto, especifica a troca de mensagens entre o servidor de origem e todos os outros que podem contribuir com o estado de um conteúdo, definindo o formato das mensagens, e as diretrizes necessárias para agregar as atividades.

Um servidor que implemente o protocolo deve incluir a URL de um *endpoint* capaz de processar mensagens Salmon, enviadas para este endereço por qualquer serviço interessado em modificar o estado de uma publicação. Uma mensagem Salmon se trata de uma nova entrada no *feed*, codificada em *base64* e envolta por outra estrutura envelope (geralmente XML) assinada digitalmente. Cabe ao servidor processar as solicitações de acordo com suas próprias políticas.

2.3.1.5 Estado do padrão

O projeto OStatus deu um passo importante em direção à padronização com a criação do grupo de trabalho no W3C. No entanto, a comunidade não conseguiu entrar em consenso em relação às decisões do protocolo, principalmente no que diz respeito à privacidade.

Não exitem avanços no projeto nos últimos anos, e o grupo de trabalho do W3C não produziu nenhum relatório desde que foi criado. Em 2012 o fundador do projeto, Evan Prodromou, anunciou o desenvolvimento do *pump.io*, outro protocolo de federação, abandonando o desenvolvimento do OStatus.

Ao mesmo tempo em que não existia comunidade ativa para a evolução do projeto, parte das tecnologias utilizadas continuou a ser mantida, o que contribui para o seu estado de obsolescência. O Activity Streams, por exemplo, já conta com uma segunda especificação que suporte tecnologia JSON, enquanto o OStatus ainda considera a especificação baseada em Atom.

[5] <http://www.salmon-protocol.org>

Apesar de defasado, o OStatus ainda é usando em novas iniciativas de redes sociais livres e federadas. O Mastodon[6] é um exemplo de projeto moderno, atualmente em desenvolvimento, e que optou pela compatibilidade com o GNU Social através da suíte do OStatus.

2.3.2 Diaspora

O projeto Diaspora[7] surgiu com a intenção de implementar o conceito de redes sociais descentralizadas em resposta aos problemas de liberdade e privacidade encontrados em plataformas sociais privadas. Sua primeira versão foi lançada em Setembro de 2010 como fruto de uma campanha de financiamento coletivo, passando a ser completamente governado pela comunidade a partir de Agosto de 2012.

A proposta dos desenvolvedores é evitar a centralização de conteúdo, portanto a falta de controle sobre as informações, construindo uma rede de instâncias pessoais da plataforma, ou *pods*. Cada servidor Diaspora reúne apenas as informações dos seus próprios usuários, mas em cooperação com outros *pods* possibilita a interação em uma rede federada.

O protocolo de federação do Diaspora foi definido para convergir com o OStatus assim que o último passe a suportar o conceito de privacidade limitada, o que nunca aconteceu. A implementação leva em consideração a troca de mensagens em formato XML, respeitando alguns conceitos básicos como a existência de usuários remotos e atualização remota e retransmissão.

2.3.2.1 Usuários Remotos

Um conceito fundamental para a implementação de redes federadas é considerar a existência de usuários remotos. A maioria das aplicações só possui o conceito de usuários locais, que estão diretamente autenticados no serviço e possuem todas as informações na base de dados local. No entanto, ao possibilitar a interação com usuários de outras redes, usuários externos ao sistema precisam ser explicitamente considerados na implementação das funcionalidades.

O Diaspora conceitualiza seus usuários em locais e externos. Enquanto usuários locais respeitam a definição tradicional, os usuários externos interagem com a aplicação através dos mecanismos de federação. É importante prever a existência de usuários externos na modelagem de sistemas. Por este motivo, implementar federação em aplicações já consolidadas pode exigir um certo esforço de refatoração.

[6] <https://github.com/Gargron/mastodon>
[7] <https://diasporafoundation.org/>

2.3.2.2 Capacidade de Retransmissão

A retransmissão é essencial em sistemas federados, visto que interações em uma rede eventualmente devem afetar *pods* relacionados. A restrição implementada pelo Diaspora indica que todas as notificações neste contexto sejam entregues tanto aos usuários locais quanto aos usuários remotos. Adicionalmente, a notificação de usuários locais não deve depender da resposta dos demais *pods*.

Em configurações de integração mais complexas, a capacidade de retransmissão passa a ser um requisito essencial para a troca de mensagens. Considere uma situação hipotética em que *pods* **A** e **B** são federados com o *pod* **C**, mas não entre si. Qualquer modificação em um conteúdo de **C** compartilhado com **A** e **B** deve afetar os três *pods*. No entanto, se a modificação partir de **A**, há uma dificuldade em notificar **B**, visto que o *pod* em questão só reconhece a existência de **C**. A solução defendida pela implementação do Diaspora é que **C** retransmita a notificação para todos os *pods* com os quais o conteúdo seja compartilhado. Isso garante que todos os sistemas federados envolvidos em uma interação sejam notificados, contribuindo com consistência das informações.

2.3.2.3 Troca de Mensagens

O Diaspora define um conjunto de mensagens que delimitam as possíveis interações entre *pods*.

- Compartilhamento de informações
- Publicações de conteúdo
- Comentários e reações a publicações
- Mensagens privadas

A troca de mensagens segue a definição do protocolo Diaspora, que utiliza um subconjunto do protocolo Salmon. De modo geral, restringe como a mensagem deve ser construída e enviada para o *endpoint* Salmon do *pod* de destino.

2.3.2.4 Estado do protocolo Diaspora

A partir de 2012 o projeto passou a ser completamente mantido pela comunidade, logo após que os fundadores abandonaram o projeto. Apesar disto, ainda conta com desenvolvimento ativo. O protocolo também continua a ser mantido, sendo que a especificação mais recente foi lançada em Julho de 2016.

O projeto ganhou certa visibilidade pouco após a sua proposta, mas o protocolo não foi capaz de adquirir adoção o suficiente para desencadear uma tentativa formal

de padronização. Ainda assim, é oficialmente suportado por outras redes sociais como o Friendica, o Hubzilla e o Loomio.

O protocolo ainda não foi capaz de padronizar a comunicação entre quaisquer redes sociais genéricas, tendo servido apenas ao propósito de permitir a integração de qualquer outra rede com o Diaspora.

2.3.3 Padronização de protocolos de federação

Apesar de não existirem produções acadêmicas que explorem a inexistência de um protocolo padrão para a federação de redes sociais, todo o debate da comunidade está documentado em lista de discussões e grupos de desenvolvimento. Esta seção é fruto de uma análise subjetiva baseada no conteúdo destes repositórios, e pretende discutir os fatores que levaram à segmentação dos projetos de federação e no insucesso de alcançar um padrão.

A comunidade destaca duas estratégias diferentes para alcançar federação[8]. Primeiro, a manutenção de um protocolo ou sistema que deva ser suportado por todas as aplicações interessadas em ingressar na federação, um método caracterizado por uma entidade que é o denominador comum entre todas as redes.

A segunda estratégia é que aplicações implementem explicitamente o protocolo de cada rede com a qual se deseja integrar, o que descarta a necessidade imediata de um protocolo padrão, configurando uma técnica poliglota.

A estratégia poliglota apresenta restrições à federação de sistemas, já que depende que cada aplicação responda aos protocolos de todas as outras aplicações da rede. Esta estratégia parece incentivar a segmentação de especificações, ao contrário do que aconteceria com a utilização de um único protocolo como denominador comum. Por outro lado, a utilização de um denominador comum depende de um único protocolo capaz de cobrir as restrições de todas as possíveis redes, visto que cada aplicação propõe seus próprios conceitos a respeito de relacionamento de usuários, publicação e compartilhamento de conteúdos, e privacidade.

As críticas direcionadas ao OStatus em relação a privacidade dos conteúdos[9] é um exemplo das barreiras que podem ser causadas por estas diferenças conceituais. Visto que o projeto suporta apenas conteúdos públicos, redes que possuam uma definição de privacidade mais complexa não são atendidas. Ao mesmo tempo em que o OStatus falhou em satisfazer estas necessidades, projetos que deram maior atenção à privacidade emergiram

[8] Parte da discussão está registrada em listas de *e-mail*, como por exemplo em uma das *threads* da lista de redes sociais federadas do W3C, que pode ser visualizada em <https://lists.w3.org/Archives/Public/public-fedsocweb/2013May/0058.html>

[9] As críticas também pode ser vista na lista de *e-mails* do W3C <https://lists.w3.org/Archives/Public/public-fedsocweb/2013May/0061.html>

e ganharam visibilidade, como o Diaspora e o Friendica.

Uma especificação que atenda universalmente todas as redes sociais exigira um grande esforço de projeto, desenvolvimento e manutenção. Uma alternativa seria um protocolo que definisse apenas um subconjunto de políticas que serviria como base para todas as implementações. Todavia, estes conceitos precisariam ser elementares o suficiente para cobrir políticas específicas que eventualmente são opostas, fazendo com que as comunidades não fossem capazes de chegar a um acordo por causa de seus próprios objetivos ou filosofias.

Apesar do limite que ser introduzido por um protocolo que especifique apenas um conjunto de diretrizes para interações entre usuários, cada aplicação poderia resolver suas necessidades específicas com base nas restrições do denominador comum, formando uma suíte de protocolos organizada em um modelo de camadas. A distribuição de responsabilidade nesta hierarquia de protocolos possibilitaria a utilização de um denominador comum diminuindo o impacto das restrições sobre os objetivos e decisões arquiteturais de aplicações que o adotassem. No entanto, a adoção não só dependeria de um acordo por parte das comunidades como envolveria a reformulação de todos as redes sociais existentes, o que não contribui para a viabilidade desta estratégia.

A adoção de uma estratégia de certa forma contradiz a necessidade de padronização observada em redes de comunicação. Neste caso, parece claro que a dificuldade em encontrar um acordo na integração das redes sociais ultrapassa os benefícios da interoperabilidade que seriam resultado da adoção de um padrão. Por outro lado, se mais aplicações implementam a integração individualmente projetos específicos podem se destacar, o que desencadearia um efeito de rede, e conduziria a um padrão válido para um subconjunto de aplicações.

Por enquanto o caminho para a federação é adotar a especificação de um dos projetos existentes, que geralmente é definida a partir de uma suíte de padrões já estabelecidos. A escolha deve ser feita levando em consideração a atividade da comunidade e a aderência da especificação às necessidades particulares. A comunidade do Noosfero identificou que esta especificação é a proposta pelo Diaspora.

Partindo dessa definição, deve-se prosseguir com o projeto da federação no Noosfero através do protocolo Diaspora, o que cobre a implementação dos conceitos necessários em sua arquitetura e a utilização de um conjunto de padrões já estabelecidos na promoção da interoperabilidade.

3 NOOSFERO

O Noosfero é um *software* livre para a construção de redes sociais e colaborativas. Desenvolvido em Ruby on Rails e licenciado sob AGPL versão 3, o projeto conta com desenvolvimento ativo.

Além dos mecanismos de interação social, o Noosfero também conta com um sistema de gerenciamento de conteúdo, o que possibilita a criação de *blogs* e o compartilhamento de arquivos. A plataforma também pode ser estendida por *plugins* desenvolvidos pela comunidade, e conta com o conceito de ambientes, que permitem a criação de diversas redes isoladas funcionando sobre uma mesma instância da aplicação.

As informações e publicações de pessoas e organizações podem ser públicas ou privadas. Já os relacionamentos entre estas entidades podem ser tanto simétricos como assimétricos.

Enquanto um relacionamento simétrico depende da concordância de ambas as partes para o compartilhamento das informações privadas (como por exemplo amizades ou filiações), um relacionamento assimétrico depende apenas do interesse de uma das entidades em acompanhar as informações públicas de algum perfil (como no caso da funcionalidade de seguidores).

3.1 SUPORTE À FEDERAÇÃO

Já existe uma iniciativa de federação no Noosfero em desenvolvimento por parte da comunidade, tendo o autor deste trabalho colaborado desde então. O objetivo é possibilitar a integração tanto com outras instâncias do Noosfero como com outras redes sociais, o que exige a adoção de especificações que tenham o mínimo de aderência na comunidade. A utilização de padrões difundidos amplia as possibilidades de integração dentre outras redes sociais federadas.

Antes da execução deste trabalho, os protocolos Diaspora e OStatus já haviam sido escolhidos como referência para a implementação da federação no Noosfero, resultado de uma observação das discussões e execução de projetos como o Hubzilla, Friendica e o próprio Diaspora.

As primeiras contribuições com a federação no Noosfero tiveram início antes deste trabalho, e estão descritas na subseção 3.1.1. Os resultados relativos à integração com outras redes, apresentados na subseção 3.1.2, são produtos deste projeto.

3.1.1 Federação entre redes Noosfero

As atividades de implementação de federação já desenvolvidas podem ser separadas em quatro fases, que cumpriram objetivos distintos de integração, que cobrem desde as funcionalidades até a reestruturação da arquitetura da aplicação.

Foi definido que um usuário de uma rede Noosfero pode acessar qualquer outra instância com as credenciais de sua rede de origem. Um usuário federado deve ser capaz de visualizar conteúdos públicos, comentar publicações, seguir usuários, e deixar mensagens em murais. As notificações destas interações devem ser entregues tanto aos usuários na rede local, quanto ao usuário na rede de origem.

O protocolo construído entre redes Noosfero é baseado nas especificações do WebFinger e OAuth para a descoberta de identidade e autorização de perfis, respectivamente. Em relação à comunicação entre as redes, o protocolo Diaspora foi definido como referência.

3.1.1.1 Fase 1: preparação

Até a versão 1.5 do Noosfero, todos os relacionamentos entre as entidades da rede eram baseados no conceito de relacionamento simétrico. No entanto, as demais redes federadas, e a maioria dos padrões mais implementados, trabalham apenas com o conceito de relacionamentos assimétricos, o que incentivou o desenvolvimento da funcionalidade de seguidores no Noosfero.

Na fase de preparação foram introduzidos os relacionamentos assimétricos através desta funcionalidade. Os seguidores são notificados a respeito de atividades públicas de perfis seguidos. No Noosfero, cada perfil pode permitir ou não que usuários o sigam. Usuários por sua vez organizam seus seguidores em círculos, categorizando suas relações.

3.1.1.2 Fase 2: intercomunicações

Durante a fase de intercomunicações foi construída a infraestrutura básica para a integração entre redes Noosfero. Ambientes e usuários externos foram introduzidos à arquitetura do Noosfero, que passa a considerar a ação de usuários que não possuem perfis locais sobre a aplicação.

O conceito de usuário externo, introduzido nesta fase, é importante para toda a implementação da federação do Noosfero. Um usuário local do Noosfero é definido por basicamente dois objetos de negócio — um usuário, que armazena as credenciais de acesso, e um perfil, que armazena suas demais informações na aplicação, sendo o que de fato se relaciona com o restante do domínio.

Já o usuário externo não possui credenciais de acesso na instância visitada, apenas um objeto que representa o seu perfil externo, e que do ponto de vista da implementação

deve ser capaz de reproduzir o comportamento de um perfil comum. A implementação alcançada faz uso de métodos *stub* e relações polimórficas.

Nesta fase também foi implementada a especificação do WebFinger, que já está sendo utilizada para a descoberta de usuários na autenticação entre redes Noosfero.

Inicialmente, apenas as redes listadas no diretório central do Noosfero [1] podem ser habilitadas no painel de administração da federação. A descentralização desta lista ou a automatização do processo de descoberta não fizeram parte do planejamento inicial.

3.1.1.3 Fase 3: integração externa

A fase de integração externa teve como objetivo aproveitar a infraestrutura de usuários externos para autenticar usuários de outros serviços sem a necessidade de perfis locais. Com isto, usuários de sistemas que suportem OAuth podem acessar o Noosfero, consumir conteúdo, e executar um conjunto limitado de ações.

Durante esta etapa, o *plugin* que torna o Noosfero em um cliente OAuth foi evoluído para permitir que usuários possam tanto criar um perfil local a partir das informações da rede de origem, como também apenas acessar o Noosfero com um perfil temporário. Por enquanto, os únicos fornecedores OAuth suportados são o Google, Facebook, Twitter, GitHub, e o próprio Noosfero. No entanto, novos fornecedores podem ser facilmente adicionados.

3.1.1.4 Fase 4: inter-relações

A última fase de desenvolvimento da federação de redes Noosfero foi implementar o relacionamento entre usuários de instâncias diferentes. De modo geral, esta fase consistiu em permitir que usuários externos sejam capazes de seguir perfis, comentar conteúdos, e publicar em murais de outros usuários.

De forma a permitir relações entre usuários federados foi necessário refatorar a funcionalidade de seguidores, adicionando o suporte a perfis externos. Neste ponto, usuários federados podem tanto seguir usuários locais, quanto serem seguidos por eles. A necessidade de simetria se dá pela contabilização e exibição dos perfis seguidores e seguidos.

Essa fase também envolve a implementação da infraestrutura de troca de mensagens, que seria utilizada nas notificações e interações entre os usuários. Até a execução deste trabalho esse mecanismo não foi completamente definido.

[1] <directory.noosfero.org>

3.1.2 Federação com outras redes sociais

A federação com redes não Noosfero deve usar a infraestrutura desenvolvida para a integração entre redes Noosfero, principalmente os mecanismos de usuários externos. Todas as atividades de implementação propostas neste trabalho foram desenvolvidas com base no protocolo Diaspora, respeitando a conclusão alcançada pela comunidade.

A longo prazo, o ideal é que as mesmas funcionalidades implementadas na federação de redes Noosfero sejam suportadas. É essencial que o Noosfero responda ao protocolo de uma forma que permita a integração bidirecional, permitindo que outras redes também sejam capazes de descobrir usuários e consumir publicações do Noosfero.

Ainda que o protocolo deva ser suportado com completude, neste primeiro momento é interessante implementar apenas uma parte da especificação. O desenvolvimento de um conjunto básico de funcionalidades, além de garantir um nível limitado de federação, deve cobrir parte da reestruturação arquitetural necessária, ajudando na identificação das modificações que devem ser introduzidas no Noosfero, o que vai ser útil para discussões futuras.

As funcionalidades a seguir foram definidas para a primeira interação de desenvolvimento definindo as contribuições deste trabalho.

1. Possibilitar um usuário possa acessar outras redes com as credenciais de sua rede de origem, sem a necessidade de um novo cadastro. Inicialmente, deve-se desenvolver um *plugin* para autorização com OpenID, visto que é o único padrão implementado pelo Diaspora;

2. Permitir que usuários de outras redes possam ser encontrados através da busca do Noosfero, o que é o primeiro passo para as inter-relações. A busca deve respeitar o padrão de descoberta do Diaspora, baseado no WebFinger;

3. Implementar relações assimétricas entre os usuários do Noosfero e de redes que respondam ao protocolo Diaspora. O protocolo deve ser usado para que as duas redes estejam cientes da relação;

4. Permitir que o Noosfero receba as publicações enviadas por redes que implementem o protocolo Diaspora.

3.1.2.1 Implementação do Protocolo Diaspora

O protocolo Diaspora está implementado no formato de uma *gem*, que pode ser facilmente incorporado como dependência em projetos desenvolvidos em linguagem Ruby, como o Noosfero. A *gem* é mantida pela mesma comunidade responsável pelo projeto original, e sempre acompanha a última especificação adotada.

Segundo os mantenedores, o protocolo Diaspora ainda não está estável, e alterações capazes de introduzir incompatibilidades podem ser introduzidas na *gem*. A implementação inicial no Noosfero deve ser executada sobre a última versão estável disponível no momento de sua adição ao projeto.

O primeiro elemento necessário para a implementação da interoperabilidade é um mecanismo de descoberta de informações entre servidores, no caso do Diaspora o WebFinger. A implementação base já está disponível no Noosfero, sendo necessário testar a integração com outro sistema que implemente o protocolo. Neste ponto a especificação do Diaspora também exige que a resposta WebFinger inclua um hCard[2], que por sua vez contém informações pessoais de cada usuário, e também deve ser implementado no Noosfero.

O segundo elemento é a comunicação entre os servidores, realizada através da troca de mensagens contendo entidades, que representam as interações entre os usuários e conteúdos. É importante que o Noosfero reconheça as entidades listadas a seguir.

- Perfil de usuário e atualizações
- Publicações e respostas
- Participação (inscrição em publicações)
- Contato entre usuários (seguir ou deixar de seguir)

As mensagens são transferidas entre os servidores por meio do protocolo Salmon. A *gem diaspora_federation* provê as funcionalidades de criptografia e serialização necessárias para a comunicação sobre este padrão, e será adicionada como dependência do Noosfero para auxiliar a implementação do protocolo Diaspora.

A implementação destes recursos também foi uma das contribuições deste trabalho, e sua realização está descrita no Capítulo 4.

[2] O hCard é um formato para a representação de informações de uma entidade, como por exemplo uma pessoa. <http://microformats.org/wiki/hcard>

4 IMPLEMENTAÇÃO

Com a contribuição deste trabalho, deve ser possível que um usuário do Noosfero consiga encontrar e seguir a atividade de usuários de alguma instância do Diaspora. Os usuários remotos descobertos devem possuir um perfil limitado no Diaspora, para que as publicações na rede de origem sejam replicadas no Noosfero.

A implementação destas funcionalidades deve cobrir os pontos a seguir.

- Descoberta de usuários em um *pod* do Diaspora.
- Resposta às consultas das informações do servidor Noosfero e da identidade ou informações do perfil de usuários locais.
- Envio de mensagens Salmon públicas e privadas para transportar as entidades que representam a interação social ou publicações.
- Recebimento de mensagens públicas que transportam entidades que representam novas publicações, criando publicações locais respectivas.

A documentação da biblioteca utilizada para a implementação do protocolo não cobre todos os aspectos da especificação, portanto foi necessário instanciar dois *pods* locais do Diaspora para analisar a comunicação. A comunicação do Diaspora depende que cada um dos *pods* responda a HTTPS e seja capaz de resolver os nomes dos demais servidores.

Para os testes desse trabalho foram utilizadas duas máquinas virtuais com Debian 8 em rede privada, com os nomes registrados no arquivo de *hosts* para a resolução local. Em cada uma das máquinas, o Diaspora foi servido por um servidor NGINX com SSL configurado com um certificado auto-assinado. Para que a comunicação não fosse prejudicada pela verificação dos certificados, eles foram manualmente adicionados aos certificados reconhecidos em cada uma das máquinas.

4.1 AUTENTICAÇÃO COM O DIASPORA

A federação através do Diaspora foi planejada com base na implementação já existente no Noosfero, que conta com um mecanismo de autenticação entre as redes. Por esse motivo inicialmente se julgou necessário permitir a autenticação de usuários com credenciais de redes Diaspora.

O Diaspora não disponibiliza nenhum *endpoint* de *login* em sua API, mas implementa o papel de fornecedor de identidades OpenID Connect[1]. Portanto, a fim de utilizar as credenciais do Diaspora, é necessário que o Noosfero possa desempenhar o papel de cliente OpenID.

Mesmo que o Noosfero também forneça identidades OpenID, ainda não seria possível usar credenciais locais para a autenticação em *pods* Diaspora, que utiliza apenas estratégias de autenticação local. Isso reforçou a decisão de implementar apenas a função de cliente OpenID por enquanto.

Ainda que o *plugi-in* tenha sido implementado como parte deste trabalho, o restante da federação não depende deste mecanismo de autenticação, já que os usuários só interagem com o servidor remoto através de sua rede de origem.

4.1.1 Desenvolvimento do plugin OpenID Client

Apesar do Noosfero já oferecer suporte à autorização com OAuth 1.0 através de um *plugin*, foi necessário implementar o suporte ao OpenID. A decisão foi criar um novo *plugin* que transforme o Noosfero em um cliente OpenID.

A *gem* openid_connect foi utilizada na implementação do consumidor. A biblioteca já oferece as diretrizes de descoberta, registro de clientes e autenticação, todas interações que envolvem requisições HTTP ao servidor fornecedor. Um perfil externo também é criado no Noosfero para armazenar algumas informações do perfil remoto, como *link* para o seu perfil, ou sua imagem do avatar.

Os passos realizados pelo *plugin* na autenticação de um usuários são:

1. O usuário digita o endereço do fornecedor OpenID de sua escolha;
2. O Noosfero tenta descobrir informações a respeito do provedor, solicitando informações do emissor de identidades;
3. Caso a resposta do provedor seja válida, o Noosfero solicita o registro como um novo cliente, solicitando acesso a informações necessárias para a criação de um perfil externo;
4. A requisição é redirecionada ao fornecedor OpenID, onde o usuário deve se autenticar, e revisar a solicitação enviada pelo Noosfero;
5. Se o usuário se autenticar no seu provedor e aprovar as informações solicitadas, a resposta do servidor será usada na criação de um perfil externo, e o usuário será autenticado no Noosfero

[1] O OpenID Connect é um padrão de autenticação construído sobre a segunda versão do OAuth. <http://openid.net/specs/openid-connect-core-1_0.html>

4.2 DESCOBERTA DE USUÁRIOS

A especificação do Diaspora propõe a descoberta de usuários através do WebFinger para a consulta de identidades, e do hCard para o compartilhamento das informações do perfil. O protocolo ainda segue a implementação do WebFinger que responde em formato XML, considerada legada.

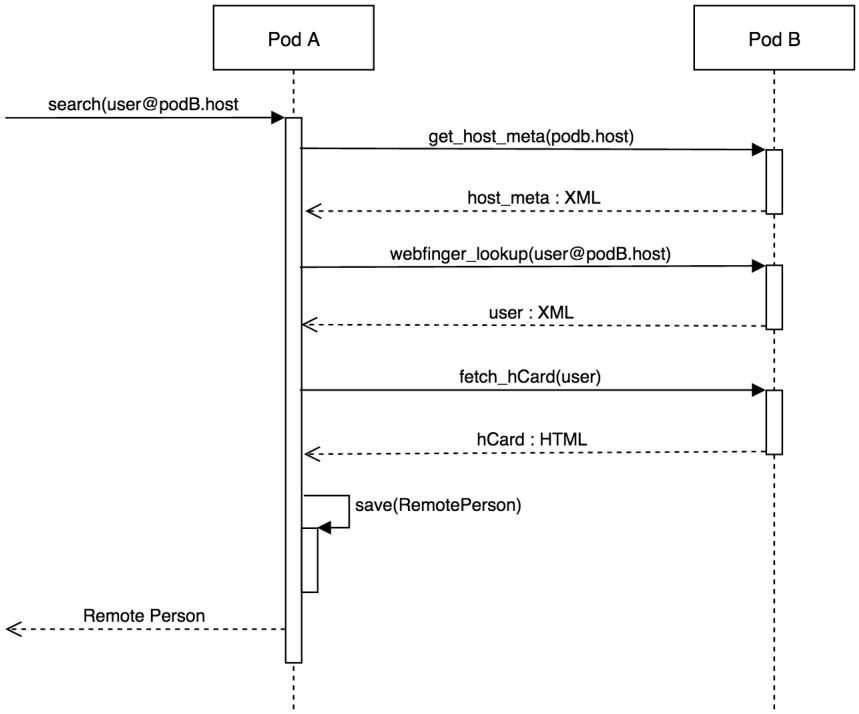

Figura 4 – Diagrama de sequência do processo de descoberta de usuários

Como pode ser visto no diagrama da Figura 4, o servidor remoto é descoberto a partir de um identificador no formato *nome do usuário@servidor diaspora*. O *endpoint* de descoberta de usuários deve ser identificado através dos metadados do servidor remoto, também obtido através do WebFinger, o qual será utilizado na consulta de identidade. A resposta às duas requisições é formatada em XML. Após a descoberta de identidade, o servidor remoto é novamente consultado pelo hCard do usuário, por sua vez em formato HTML, contendo as informações do perfil público.

No Noosfero, a descoberta de usuários remotos foi implementada a partir da busca de pessoas. Se a *string* de busca conter o símbolo "@", a *string* é quebrada no formato *nome@host*, e o processo descrito anteriormente é executado para os atributos extraídos. Visto que trata-se de uma busca e os resultados são imediatamente renderizados pelo servidor, por enquanto toda a operação é realizada em *foreground*.

Já que a implementação de descoberta é baseada no padrão WebFinger, apesar de seguir o protocolo do Diaspora, é possível encontrar usuários em qualquer aplicação que

o implemente neste formato. Os *endpoints* que respondem às requisições do WebFinger e hCard foram implementadas como parte deste trabalho, e pode substituir aos mecanismos atualmente implementados para a federação de redes Noosfero.

4.3 TROCA DE MENSAGENS

A primeira mudança introduzida no Noosfero foi a criação de páginas de perfil para usuários externos, para conter as publicações resgatadas da rede de origem, e as opções de interação. Anteriormente, referências a usuários remotos redirecionavam a sua rede de origem. A criação de páginas locais exigiu alterações no mecanismo de carregamento dos perfis no Noosfero.

Para seguir um externo, é necessário enviar uma mensagem Salmon privada para o seu *endpoint* de recebimento. Ao receber esta mensagem, o Diaspora cria um perfil remoto para o usuário do Noosfero que solicita a ação — o que envolve a obtenção de identidade e informações públicas na rede de origem, e registra as informações a respeito do servidor. A Figura 5 descreve o processo, que ocorre de maneira similar à executada durante a descoberta de usuários.

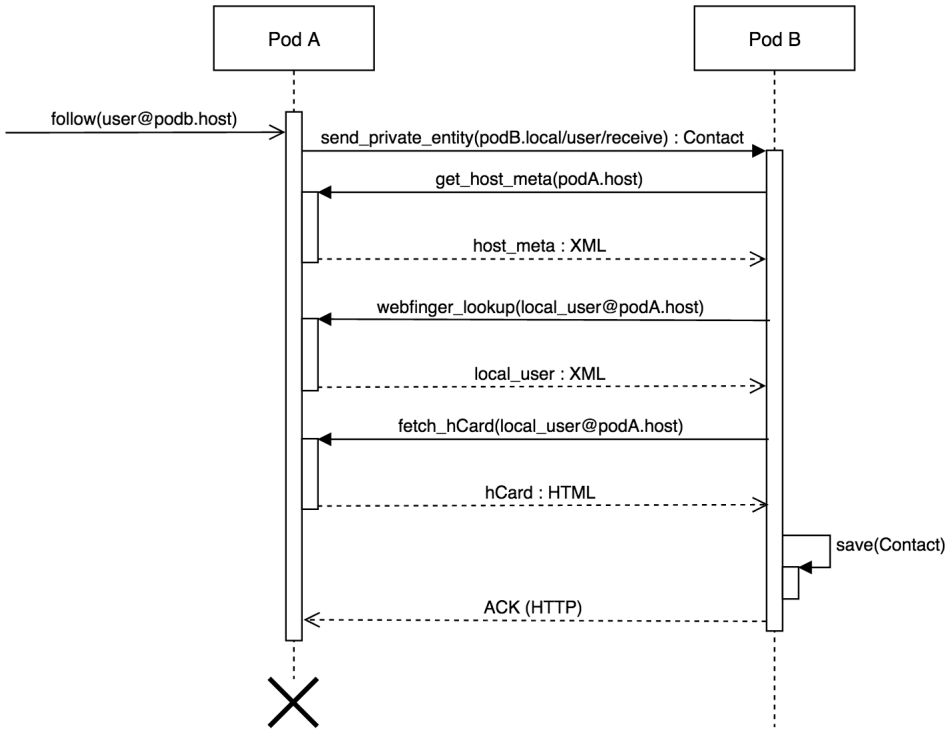

Figura 5 – Diagrama de sequência do processo de compartilhamento de contatos

O envio da entidade de contato por meio do Salmon é realizado em *background* durante a execução da funcionalidade de seguir. É importante garantir que a mensagem foi entregue, caso contrário o usuário é seguido no Noosfero mas a interação não foi corre-

tamente federada. A implementação executada programa outra execução da tarefa após 10 minutos caso a mensagem não seja entregue. É necessário considerar um parâmetro para o cancelamento da requisição, visto que a tarefa vai continuar a ser programada infinitamente se o servidor de destino for desativado.

Mensagens privadas do Salmon devem ser criptografadas com RSA, portanto também é necessário que cada usuário do Noosfero possua um par de chaves individual. As chaves de um usuário são geradas apenas quando necessárias, o que evita o aumento do esforço computacional na criação de usuários, ou em maiores dificuldades em uma instalação que já possua vários usuários criados.

O par de chaves é gerado com a implementação em Ruby do OpenSSL, e inicialmente os valores são armazenados serializados como texto em claro no banco de dados. Também é importante encontrar uma alternativa para o armazenamento inadequado das chaves privadas.

A biblioteca que implementa o protocolo providencia funções para a construção e assinatura de mensagens Salmon, o que foi suficiente para o envio de contatos. O Diaspora já foi capaz de reconhecer o servidor do Noosfero, criar os perfis remotos, e exibir as notificações da interação.

O último passo foi consumir as publicações dos usuários do Diaspora. Como pode ser visto na Figura 6, novos conteúdos são enviados a todos os os servidores assinados naquela interação, no caso redes de origem de usuários que sigam o autor. Priorizou-se a federação de conteúdos públicas, e neste caso uma mensagem Salmon pública é enviada para o servidor Noosfero.

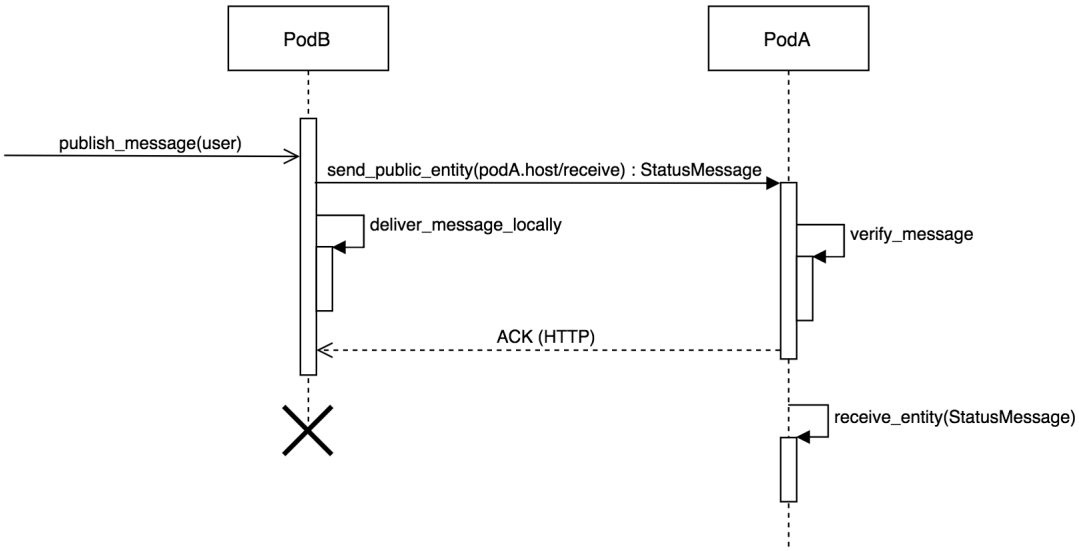

Figura 6 – Diagrama de sequência do envio de publicações entre servidores

As publicações do Diaspora foram representadas no Noosfero como *scraps*, que originalmente eram publicadas apenas por perfis locais. Foi necessário evoluir as relações da camada de domínio, permitindo que usuários externos também publicassem novos *scraps*, e adicionar um atributo GUID, um identificador universal, obrigatórios em todas as entidades federadas através do protocolo Diaspora.

Com essas modificações, é possível criar um *scrap* para o usuário externo a partir da requisição do Diaspora. A publicação é visível apenas no perfil do usuário externo, já que o Noosfero ainda não conta com um mecanismo de *feed* ou notificação de publicações de perfis seguidos.

Todas as modificações descritas nesta seção foram efetuadas no código *core* do Noosfero[2]. Algumas necessidades, como por exemplo montar as rotas das ações da federação na raiz do *host*, podem dificultar a construção do código em um *plug-in*. Ainda assim, a federação poderia ser desenvolvida em um destes componentes com modificações menores no *code*, caso a comunidade julgue mais adequado.

[2] O código desenvolvido pode ser visualizado na *branch* diaspora_federation, disponível no repositório do autor. <https://gitlab.com/gabrielssilva/noosfero>

5 CONSIDERAÇÕES PRELIMINARES

Através da implementação descrita no Capítulo 4 é possível obter um conjunto de funções que demonstra as capacidades de federação através do protocolo Diaspora. Todas as funções listadas abaixo foram resultado da contribuição realizada durante a execução deste trabalho.

- Conhecendo o identificador do usuário e o endereço de sua rede de origem, é possível encontrá-lo a partir da busca de pessoas;
- A partir da rede local, é possível seguir o perfil de um usuário em sua rede de origem, expressando a intenção de receber suas publicações;
- As publicações de usuários remotos podem ser visualizadas em seu perfil local do Noosfero assim que seu servidor de origem expeça as notificações.

Essas funcionalidades tornam o Noosfero federado com qualquer aplicação que implemente o protocolo Diaspora. Mais precisamente, por enquanto o Noosfero só depende de uma aplicação que respeite a especificação no suporte à descoberta de usuários, recebimento de contatos, e envio notificações de publicação.

Algumas destas funções também dependem do suporte no Noosfero aos padrões WebFinger e hCard, o que já possibilita que usuários do Noosfero possam ser descobertos por qualquer outra rede que implemente o mesmo protocolo, inclusive outras instâncias do Noosfero. Isso permite que a implementação através do Diaspora substitua a federação entre redes Noosfero implementada anteriormente através da API, o que não seria custoso, já que o código ainda não foi integrado na *branch* primária.

Para implementar este subconjunto do protocolo Diaspora, foi necessário introduzir algumas modificações que afetaram a arquitetura do Noosfero. A lista abaixo descreve as mudanças que exerceram maior impacto sobre o código.

- A busca passou a retornar perfis de usuários externos;
- Usuários remotos possuem uma página de perfil que pode ser visualizada na instância local, e devem ser capazes de publicar *scraps*;
- Todas as classes de domínio envolvidas na federação devem possuir um identificador universal (GUID). Por enquanto, o novo campo foi adicionado nas classes que representam usuários remotos e *scraps*;

- Cada um dos usuários do Noosfero possui um par de chaves RSA próprio para o envio de mensagens privadas através do Salmon;
- Usuários remotos podem ser seguidos e adicionados em círculos por usuários locais;[1]

Tendo sumarizado as contribuições deste trabalho, é possível propor um cronograma de referência para a continuidade do projeto. A Tabela 1 exibe um cronograma mensal contendo os marcos para a finalização da federação com o Diaspora. Também propõe uma última atividade para a integração entre duas instâncias do Noosfero — o Rede Comunidade UnB e o Stoa Social da USP, oferecendo um caso de federação entre dois sistemas em produção.

Mês	**Atividades**
Janeiro	implementar retrações das entidades já implementadas, permitindo que o Noosfero remova publicações apagadas na rede original. Também é necessário tratar atualizações de perfis.
Fevereiro	Implementar o suporte às entidades de Contato no Noosfero, fazendo com que seus usuários também possam ser seguidos a partir de outras redes. Nesse ponto, é necessário estabelecer o mecanismo que envia novas publicações a todos os inscritos.
Março	Federar conteúdos privados, como publicações para círculos específicos, ou mensagens diretas.
Abril e Maio	Analisar a integração do código do protocolo Diaspora na *branch* de federação do *core* do Noosfero, considerando a substituição da implementação atual.
Junho	Atualizar o Noosfero do Comunidade UnB e do Stoa para uma versão federada, e realizar a integração entre as plataformas.

Tabela 1 – Cronograma de referência para a continuidade do projeto

[1] Essa alteração em específico já havia sido desenvolvida como parte da federação entre redes Noosfero. No entanto, até este ponto a contribuição ainda está sob revisão, e por esse motivo foi integrado manualmente à *branch* em que a federação com o Diaspora foi desenvolvida.

Referências

BARAN, P. On distributed communications networks. *IEEE Transactions of the Professional Technical Group on Communications Systems*, January 1964. Citado 3 vezes nas páginas 9, 17 e 18.

BAROCAS, S. et al. A critical look at decentralized personal data architectures. *CoRR*, abs/1202.4503, 2012. Disponível em: <http://arxiv.org/abs/1202.4503>. Citado 2 vezes nas páginas 18 e 24.

BOYD, D. m.; ELLISON, N. B. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, Blackwell Publishing Inc, v. 13, n. 1, p. 210–230, 2007. ISSN 1083-6101. Disponível em: <http://dx.doi.org/10.1111/j.1083-6101.2007.00393.x>. Citado na página 17.

COMER, D. E. *Internetworking with TCP/IP, Volume 1: Principles, Protocols, and Architectures, Fourth Edition*. 4th. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000. ISBN 0130183806. Citado 2 vezes nas páginas 21 e 22.

HAMMER-LAHAV, E. *LRDD: Link-based Resource Descriptor Discovery*. [S.l.], 2010. 1-16 p. Disponível em: <https://tools.ietf.org/id/draft-hammer-discovery-06.txt>. Citado na página 27.

JONES G. SALGUEIRO, M. J. P.; SMARR, J. *WebFinger*. [S.l.], 2013. 1-28 p. Disponível em: <https://tools.ietf.org/rfc/rfc7033.txt>. Citado na página 27.

KLENSIN, J. *Simple Mail Transfer Protocol*. [S.l.], 2001. 1-79 p. Disponível em: <https://tools.ietf.org/rfc/rfc1280.txt>. Citado na página 24.

KUROSE, J. F.; ROSS, K. W. *Computer Networking: A Top-Down Approach (6th Edition)*. 6th. ed. [S.l.]: Pearson, 2012. ISBN 0132856204, 9780132856201. Citado 6 vezes nas páginas 9, 21, 22, 23, 24 e 25.

LIEBOWITZ, S. J.; MARGOLIS, S. E. *Network Externalities (Effects)*. 1998. Disponível em: <https://web.archive.org/web/20160410041613/http://www.utdallas.edu/~liebowit/palgrave/network.html>. Citado na página 23.

NEVILE, C. M.; JACOBS, I. *World Wide Web Consortium Process Document*. [S.l.], 2015. Disponível em: <https://www.w3.org/2015/Process-20150901/>. Citado na página 26.

POSTEL, J. *IAB Official Protocol Standards*. [S.l.], 1992. 1-32 p. Disponível em: <https://tools.ietf.org/rfc/rfc1280.txt>. Citado na página 23.

STALLINGS, W. *Cryptography and Network Security: Principles and Practice*. 5th. ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2010. ISBN 0136097049, 9780136097044. Citado na página 17.

TANENBAUM, A. S.; WETHERALL, D. J. *Computer Networks*. 5th. ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2010. ISBN 0132126958, 9780132126953. Citado 3 vezes nas páginas 21, 23 e 24.