Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Marciel de Lima Oliveira

SNMP Gateway CCN: Uma proposta de arquitetura para gerência de redes orientadas a conteúdo interoperável com sistemas legados

Campinas
2017

Marciel de Lima Oliveira

SNMP Gateway CCN: Uma proposta de arquitetura para gerência de redes orientadas a conteúdo interoperável com sistemas legados

Proposta de Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Christian Rodolfo Esteve Rothenberg

Este exemplar corresponde à versão final da dissertação defendida pelo aluno Marciel de Lima Oliveira e orientada pelo Prof. Dr. Christian Rodolfo Esteve Rothenberg

Campinas
2017

Agência(s) de fomento e nº(s) de processo(s): Não se aplica.
ORCID: http://orcid.org/http://orcid.org/ht

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

Oliveira, Marciel de Lima, 1981-
OL4s SNMP Gateway CCN : uma proposta de arquitetura para gerência de redes orientadas a conteúdo interoperável com sistemas legados / Marciel de Lima Oliveira. – Campinas, SP : [s.n.], 2017.

Orientador: Christian Rodolfo Esteve Rothenberg.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Redes de computadores. 2. Redes de computadores - Gerência. 3. Arquitetura de rede de computadores. 4. Telecomunicações - Redes de computação. I. Rothenberg, Christian Rodolfo Esteve. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: SNMP Gateway CCN : an information-centric networking management architectural proposal with legacy systems interoperability
Palavras-chave em inglês:
Computer network
Computer networks - Management
Computer network architecture
Telecommunications - Computer networks
Área de concentração: Engenharia de Computação
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Christian Rodolfo Esteve Rothenberg [Orientador]
Rodolfo da Silva Villaça
Maurício Ferreira Magalhães
Data de defesa: 17-04-2017
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

Candidato: Marciel de Lima Oliveira RA: 121625
Data da Defesa: 17 de Abril de 2017
Título da Tese: "SNMP Gateway CCN: Uma proposta de arquitetura para gerência de redes orientadas a conteúdo interoperável com sistemas legados"

Prof. Dr. Christian Rodolfo Esteve Rothenberg (Presidente, FEEC/UNICAMP)
Prof. Dr. Rodolfo da Silva Villaça (DTI/UFES) - Membro Titular
Prof. Dr. Maurício Ferreira Magalhães (FEEC/UNICAMP) - Membro Titular

Ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no processo de vida acadêmica do aluno.

Dedico este trabalho ao Senhor Jesus Cristo por me conceder oportunidade à vida e permitir compartilhar do seu infinito amor.

Agradecimentos

Inicialmente, agradeço ao Prof. Dr. Christian Rothenberg pela oportunidade concedida e orientações. Ao Prof. Dr. Mauricio Magalhães que proporcionou momentos de aprendizado de grande valor.

Aos meus pais: Paulo José de Oliveira e Marineide de Lima, que com muito amor carinho e dedicação ajudaram a moldar o meu caráter e ensinaram que os valores, o respeito e o amor estão acima de todas as outras coisas na vida.

À minha esposa: Ellen Dayanne Sanchez Oliveira, que sempre me apoio nesta jornada, com muita paciência e carinho compreendeu os momentos que precisei trocar descanso, por horas de trabalho no Mestrado.

Ao Caio de Moraes Elias, aluno de graduação da Unicamp, que contribuiu com importantes informações na fase de implementação da ferramenta.

Aos amigos em ocasião da Padtec que contribuíram com este projeto: Vinícius Geraldo Félix, Gustavo da Silva Souza Filho, Rodrigo Otávio Haydt, Lucas Martin Giacomini Mendes, Marcelo Mitsutoshi Uesono, Marcos Antonio de Siqueira, Josélis Pio de Oliveira e Rodrigo de Almeida Moreira.

Resumo

Pesquisas voltadas às Redes Orientadas a Conteúdo (ROCs) têm maior foco nos planos de dados e controle em comparação ao plano de gerência. Como contribuição para suprir essa carência, este trabalho apresenta alguns mecanismos de mapeamento e a ferramenta SNMP Gateway CCN, que permite o gerenciamento e monitoramento de nós CCN através de sistemas de gerência de redes legadas baseados no protocolo SNMP. Desta forma é possível explorar o conceito de gerencia orientada ao conteúdo (nomes/dados) em um ambiente emulado. A ferramenta exercita as principais operações do protocolo SNMP que comprovam seu funcionamento, neste processo cada mensagem SNMP é traduzida e encaminhada para a rede CCN, que retorna ao gateway em resposta a cada solicitação e novamente é traduzida no processo inverso. Para a prova de conceito a operação SNMP GET foi executada e teve seu funcionamento avaliado passo a passo, o trabalho também apresenta a coerência do consumo de banda durante a consulta de todos os objetos de dois elementos de rede CCN gerenciados.

Palavras-chaves: Redes Orientadas a Conteúdo (ROCs); Content-Centric Networking (CCN); Gerência de Redes; CCNx.

Abstract

Research efforts on Information-Centric Networking (ICN) mainly focus on the ”data and control plane”challenges compared to the efforts devoted so far on ”management plane”. Aiming at addressing this gap, this work presents some mapping mechanisms and SNMP Gateway CCN tool, in order to enable the management and monitoring of CCN nodes through legacy SNMP-based systems. Therefore it is possible to explore the concept of content-oriented management (name/data) in an emulated environment. The tool performs the main operations of the SNMP protocol that prove its operation, in this process each SNMP message is translated and forwarded to the CCN network, which returns to the gateway in response to each request then it is translated again in the opposite way. For the proof-of-concept SNMP GET operation was performed and had its operation evaluated step by step, the work also presents the consistency of the bandwidth consumption during the query of all the objects of two CCN network elements managed.

Keywords: Information-Centric Networking (ICN); Content-Centric Networking (CCN); Network Management; CCNx.

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

ASN.1	*Abstract Sintax Notation One*
BER	*Basic Encode Rules*
CCN	*Content-Centric Networking*
CLI	*Command Line Interface*
CMISE	*Common Management Service Element*
CMIP	*Common Management Information Protocol*
CS	*Content Store*
DWDM	*Dense Wavelength Division Multiplexing*
DSL	*Digital Subscriber Line*
DCN	*Data Communication Network*
DES	*Data Encryption Standard*
FIB	*Forward Information Base*
FTTH	*Fiber To The Home*
IP	*Internet Protocol*
IANA	*Internet Assigned Numbers Authority*
IETF	*Internet Engineering Task Force*
ICNRG	*Information-Centric Networking Research Group*
JSON	*JavaScript Object Notation*
LTE	*Long Term Evolution*
MPLS-TP	*Multiprotocol Label Switching - Transport Profile*
MIB	*Management Information Base*
MD5	*Message-Digest algorithm 5*
NOC	*Network Operation Center*
NMS	*Network Management System*
NONM	*Name-Oriented Network Management*
NETCONF	*Network Configuration*
NFV	*Network Functions Virtualization*
NLSR	*Named-data Link State Routing Protocol*
OPEX	*Operational Expenditure*
OSI	*Open System Interconnection*
OID	*Object identifier*
OS	*Operating System*
OSPFN	*Open Shortest Path First for Named Data Networking*
PDU	*Protocol Data Unit*

PIT	*Pending Interest Table*
REST	*Representational State Transfer*
RPC	*Remote Procedure Call*
ROCs	*Redes Orientadas a Conteúdo*
SOAP	*Simple Object Access Protocol*
SNMP	*Simple Network Management Protocol*
SHA	*Secure Hash Algorithm*
SDN	*Software Defined Networking*
SGMP	*Simple Gateway Monitoring Protocol*
SMI	*Structure of Management Information*
SCNT	*SNMP Content Network Translation*
UI	*User Interface*
VM	*Virtual Machine*
YANG	*Yet Another Next Generation*

Sumário

Capítulo 1

Introdução

Com o surgimento de grandes redes de transporte e equipamentos complexos, construídos para tratar diversos serviços, dentre eles, dados, voz, vídeo, surge também a preocupação em monitorar e otimizar o uso destas redes (equipamentos e serviços). Esse monitoramento é classificado como plano de gerência, que difere do plano de controle e plano de dados, que tratam dos serviços oferecidos ao cliente. Contudo, criou-se o centro de operações de redes (NOC) para cuidar do plano de gerência, que atua em regime 24/7, importante para a operação, manutenção e análise do desempenho das redes e equipamentos. O plano de gerência em grandes redes de transporte é tão importante quanto os planos de controle e dados, esses planos são tratados de forma separada, e exigem certo grau de independência, física e/ou lógica.

Geralmente o plano de gerência é definido muito antes dos planos de controle e dados tendo em vista a necessidade do controle e aprovisionamento dos serviços que serão fornecidos através do uso de mecanismos previamente adotados no plano de gerência. A escolha e a padronização de tais mecanismos tem sido um fator preocupante devido a grande diversidade de arquiteturas existentes atualmente. Tais arquiteturas foram definidas para tratar o controle, configuração, monitoramento e mapeamento de métricas de cada equipamento e serviços fornecidos pelas redes com o intuito de preservar a qualidade dos mesmos.

O plano de gerência pode contar com um Sistema de Gerência de Rede (NMS) central que geralmente atua ao mesmo tempo no controle das três grandes divisões das redes (plano de dados) de equipamentos de telecomunicações, NÚCLEO (ex.: DWDM), AGREGAÇÃO (ex.: MPSL-TP, Metro-Ethernet) e ACESSO (ex.: xDSL, LTE, FTTH). A comunicação entre o Sistema de Gerência de Rede e os equipamentos no plano de dados é feita através de uma rede dedicada chamada DCN (*Data Communication Network*) e os protocolos de rede TCP/IP são adotados como padrão para uso nos equipamentos que compões a DCN (Roteadores L3/IP/MPLS e Switches L2/Ethernet/Metro).

O surgimento de recentes trabalhos em Redes Orientadas a Conteúdo (ROCs) representa um novo paradigma onde o foco das redes é baseado no conteúdo e não mais

na sua localização (BRITO *et al.*, 2012). As ROCs propõem que o conteúdo seja o elemento central das redes, independente de sua localização, substituindo o foco de *onde* para *o quê*. Nas ROCs, a infraestrutura da rede participa ativamente no armazenamento (*caching*) e na distribuição dos conteúdos visando um aumento na eficiência da busca e na disponibilidade dos conteúdos na rede. As ROCs têm despertado grande interesse no meio acadêmico e dentre várias empresas e institutos relacionados às pesquisas na área das novas arquiteturas de rede vem abrindo espaço para novas aplicações, pesquisas e experimentos, tais como: CCN (JACOBSON *et al.*, 2009), que apresenta uma estrutura hierárquica para nomes semelhante às URLs; DONA (KOPONEN *et al.*, 2007), que utiliza o mecanismo de nomeação plana e funções de hash criptográfico e LIPSIN (JOKELA *et al.*, 2009) que possui uma arquitetura que identifica os enlaces pelo nome ao invés dos pares de endereços fim a fim. O novo paradigma proposto pelas ROCs traz consigo inúmeros desafios (XYLOMENOS *et al.*, 2009), tais como: foco em nomes/dados, roteamento/entrega dos dados, mobilidade, segurança, monitoramento e notificação de eventos.

Este trabalho foca no ponto de vista de gerência de redes orientadas a conteúdo levando em consideração a carência, tanto no nível de mecanismos adequados, como na definição de um plano de gerência para estas redes. Neste contexto, apresentamos uma proposta de arquitetura NONM (*Name-Oriented Network Management*) que tem como principais destaques a modelagem da MIB CCN para identificação dos objetos do nó CCN e um Gateway SNMP que converte as mensagens das redes legadas para interação com elementos nativos da rede CCN.

1.1 Motivação

A motivação principal deste trabalho deve-se à percepção da carência de paradigmas adequados à gerência de redes orientadas a conteúdo (KUTSCHER *et al.*, 2016). Consideramos a possibilidade de experimentar gerência de redes orientadas a conteúdo com o uso de protocolos e arquiteturas de gerência utilizados nas redes tradicionais, como por exemplo; TL1, REST, SNMP, CLI , WEB UI e NETCONF/YANG, transformando-as em ferramentas eficientes para a gerência de redes CCN. Servindo como um facilitador e primeiro passo para a interoperabilidade entre redes convencionais IP (*Internet Protocol*) e CCN, que permite aproveitar a infra-estrutura legada já existente.

1.2 Objetivo

A arquitetura CCN (*Content-Centric Networking*) adotada como referência nesse trabalho é reconhecidamente uma das propostas mais relevantes na literatura relativamente às redes orientadas ao conteúdo. As redes CCN utilizam uma estrutura de nomes hierárquicos e legíveis (formados por sequências de caracteres e números) para identificar os conteúdos. Tais nomes possuem características semânticas, ou seja, os componentes hierárquicos utilizados na identificação trazem algum tipo de informação sobre o conteúdo como, por exemplo, versão, formato ou propriedade.

Para tornar os sistemas de gerência compatíveis, esta proposta define como estratégia a utilização do *label*/nome como identificador único de um nó na rede CCN e o mapeamento de mensagens através de um gateway. Essa estratégia é denominada NONM (*Named-Oriented Network Management*), tem como principal tarefa compatibilizar a gerência tradicional baseada no protocolo SNMP (*Simple Network Management Protocol*), com o modelo CCN, dessa forma permitir a gerência de elementos de rede CCN nativos.

1.3 Escopo e Abordagem

Protocolo de gerência de redes. Por se tratar de um protocolo aberto e largamente utilizado em gerência de redes, optamos pelo protocolo SNMP como o primeiro passo para tornar os Sistemas de Gerência (NMS) legados compatíveis com redes CCN nativas. Teremos como tarefa propor o mapeamento das principais operações do protocolo (ex.: *GET*, *SET*, *TRAP*) para uso na gerência de redes CCN de forma totalmente transparente do ponto de vista do sistema de gerência legada.

Sistema Operacional e Plataforma de Emulação. Para prover a comunicação da rede legada com a rede CCN, utilizaremos o Linux O.S e a plataforma MiniCCNx (CABRAL *et al.*, 2013) como base para a implementação da prova de conceito e para o exercício de experimentos . A Plataforma MiniCCNx fornece um ambiente emulado completo para redes CCN, neste ambiente será possível observar o comportamento das mensagens SNMP capturadas e traduzi-las para interação com nós nativos da rede CCN através da ferramenta *SNMP Gateway CCN*.

1.4 Estrutura do trabalho

O trabalho está organizado da seguinte forma: no Capítulo 2 temos uma breve descrição das principais arquiteturas e protocolos para gerência de redes, uma descrição mais detalhada do protocolo SNMP e do modelo CCN, também a descrição de dois trabalhos relacionados; o Capítulo 3 aborda a proposta da arquitetura de gerência orientada a conteúdo; o Capítulo 4 apresenta as estratégias para a implementação da prova de conceito; o Capítulo 5 apresenta a metodologia de avaliação da proposta, experimentos e resultados obtidos; finalmente, o Capítulo 6 apresenta as conclusões, trabalhos futuros e contribuições.

Capítulo 2

Revisão Bibliográfica Resumida

Neste capítulo faremos uma breve descrição das principais arquiteturas/protocolos de gerência de redes para monitoramento e configuração de equipamentos, o protocolo SNMP e o modelo CCN.

2.1 Arquiteturas de Gerência e Configuração

TL1. O TL1 (*Transaction Language 1*) é um protoloco de gerência de redes de telecomunicações baseado em ASCII padronizado pelas normas *Bellcore Standard* GR-199/815/831/835-CORE. O protocolo padrão de transporte utilizado pelo TL1 é o TCP, é possível customizar o TL1 para utilizar outros protocolos de transporte independentes. O TL1 garante a comunicação entre elementos gerenciados e o sistema de gerência, um agente fornece acesso aos dados armazenados no dispositivo e o gerente utiliza este acesso para controlar e monitorar o dispositivo gerenciado. O TL1 é considerado um protocolo legado, porém continua sendo um importante protocolo usado no campo da monitoração pois ainda é muito popular entre os equipamentos SONET (*Synchronous Optical Network*, padrão Americano equivalente ao SDH), legados e outros equipamentos de rede (Webnms[1]).

REST. O REST (*Representational State Transfer*) é uma arquitetura que determina os elementos necessários para construir sistemas de informação distribuídos, como interface de comunicação e protocolos. O REST provê interação sem estado (*stateless*) entre agentes com uso do protocolo HTTP e seus métodos pré-definidos: GET, POST, PUT e DELETE. O modelo REST se difere do modelo SOAP (*Simple Object Access Protocol*), porém estudos apontam uma aproximação entre os dois modelos para implementação dos Serviços Web (NUNES; DAVID, 2013).

SNMP. O SNMP (*Simple Network Management Protocol*) é um protocolo aberto de gerência de redes desenvolvido para gerenciar e configurar elementos de rede, monitorar o comportamento dos elementos e a alteração de seus estados, que é feito através da observação de eventos gerados na rede. O SNMP será detalhado mais a frente em

[1]Webnms - http://www.webnms.com (acesso em 2014)

um tópico separado.

CLI. A CLI (*Command Line Interface*) é uma interface para o usuário interagir com sistemas operacionais de computadores ou equipamentos eletrônicos em geral. Essa interface é operada através de comandos em modo texto puro que obedecem a uma sintaxe padrão definida por cada equipamento ou fabricante. A interface CLI se tornou popular com a introdução de *display* de vídeo em computadores por volta da década de 1960. Atualmente muitos equipamentos ainda operam com esse tipo de interface, como por exemplo, equipamentos de rede *Switch L2* ou Roteadores L3 que possuem essa interface acessível através de interpretadores SHELL e protocolos de conexão (ex.: TELNET, SSH e etc) para o aprovisionamento de serviços e configurações em geral.

WEB UI. O WEB UI (*User Interface*) provê mecanismos que manipulam, inserem e removem configurações em dispositivos de redes. Esse modelo é baseado no protocolo HTTP e seus métodos: GET, POST, PUT e DELETE que utilizam um *Web Browser* como interface que interage com uma aplicação *Web Server* executada no equipamento de rede. A interface *Web UI* surgiu como uma opção à interface CLI, pois se trata de um ambiente intuitivo e agradável de fácil operação, que abstrai milhares de comandos utilizados para interagir diretamente com os equipamentos.

NETCONF/YANG. O NETCONF (*Network Configuration*) é um protocolo que provê mecanismos que instalam, manipulam e removem configurações em dispositivos de redes, o modelo cliente e servidor é utilizado como base desta comunicação. O cliente incia a comunicação com o servidor (sessão NETCONF) através de RPC (*Remote Procedure Call*) e utiliza camadas seguras de transporte como SSH, TLS, SOAP ou BEEP. Com a sessão estabelecida, é possível acessar ou alterar as configurações de um elemento de rede com uso de arquivos XML. É um protocolo que se difere dos modelos CLI e WEB UI, que foram concebidos para configuração manual, por outro lado o NETCONF tem como arquitetura básica prover uma interface de programação que permita gerenciar, configurar e monitorar elementos de rede. O NETCONF foi desenvolvido por um grupo da IETF e publicado incialmente em dezembro de 2006 (RFC4741), em junho de 2011 foi revisado e novamente publicado (RFC6242)(*Netconf Central*[2]). O YANG, é uma linguagem usada para modelar a configuração e manipular o estado dos dados através do NETCONF, tem forte abordagem no reuso pois foi projetada para ser estendida e permitir a criação de novas funcionalidades. Surgiu como uma proposta de modelagem com maior foco nas operações de configuração em grandes redes (*Tail-f*[3]) em relação aos outros modelos existentes, tais como, o SMI do SNMP, esquemas UML e XML. Os trabalhos para padronizar o modelo YANG iniciaram-se em 2007 com seu primeiro padrão em 2008, em outubro de 2010 (RFC6020) foi publicada a versão final pelo IETF (SCHONWALDER *et al.*, 2010). O padrão NETCONF/YANG Também possui suporte para notificação de eventos baseado em *event stream abstraction*. Clientes interessados em receber notificações, assinam (*subscribe*) um *event stream* e um filtro para recebimento de eventos pode ser aplicado antes da notificação ser enviada ao cliente. Isso permite que apenas eventos relevantes

sejam selecionados e também permite escalabilidade.

2.2 Protocolo SNMP

Até o final da década de 1980 o mundo estava carente de padrões para gerenciamento de redes e equipamentos, neste momento surgem os padrões OSI (*Open System Interconnection*) CMISE/CMIP (*Common Management Service Element/Common Management Information Protocol*) e o SNMP (*Simple Network Management Protocol*) da Internet. O SNMP foi amplamente aceito e se tornou o principal protocolo criado para ser independente de equipamentos e redes. A primeira versão do protocolo (SNMPv1) foi baseada no projeto SGMP (*Simple Gateway Monitoring Protocol*), (RFC 1028) e projetada em poucos meses por pesquisadores, usuários e estudantes universitários com experiência no SGMP. O protocolo evoluiu da versão SNMPv1 para as versões SNMPv2 e SNMPv3, lançada em 1999 e revisada em 2002.

O IETF (Internet Engineering Task Force) é responsável por definir os padrões e manter o protocolo SNMP através de publicações de RFCs (Requests for Comments). O SNMPv1 é a primeira versão do procotolo que foi definida na RFC 1157, a segunda versão SNMPv2 foi definida nas RFC 3416, RFC 3417 e RFC 3418, a atual versão SNMPv3 foi definida nos padrões RFC 3410, RFC 3411, RFC 3412, RFC 3413, RFC 3414, RFC 3415, RFC 3416, RFC 3417, RFC 3418, RFC 2576, RFC 2570 e RFC 2786, que acrescenta segurança criptográfica ao protocolo. A versão SNMPv3 utiliza os algorítimos MD5 ou SHA para autenticação de usuários (evita o envio de senha em texto claro) e o algorítmo DES para criptografar e descriptografar as mensagens SNMP, técnicas que buscam suprir a carência de mecanismos de segurança das versões anteriores.

O protocolo SNMP faz parte da infraestrutura que adota três componentes básicos; entidade gerenciadora, dispositivo gerenciado e o próprio protocolo de gerência, como mostra a Figura 2.1.

1- Entidade gerenciadora (Gerente). É uma aplicação que controla e coleta as informações de gerenciamento de uma rede, normalmente denominada NMS (*Network Management System*). Um NMS é responsável pelo *pooling* e recebimento de *TRAPs* dos agentes. Os *poolings* são requisições feitas para um agente por pedaços de informações. A *TRAP* é a forma que o agente se comunica com o gerente para notificar eventos que são relevantes para o bom funcionamento do elemento e da rede.

2- Dispositivo ou elemento gerenciado (Agente). É o elemento de rede

[2]Netconf Central - http://www.netconfcentral.org (acesso 2014)

[3]Tail-f - Grandes fabricantes já adotam o modelo NETCONF/YANG em suas soluções, atualmente algumas ferramentas comerciais já estão um passo a frente no sentido de unificar e automatizar a configuração, gerência e a operação das redes com base em novas tecnologias, tais como SDN (*Software Defined Networking*) e NFV (*Network Functions Virtualization*), com o principal objetivo de diminuir custos operacionais (OPEX). http://www.tail-f.com/wordpress/wp-content/uploads/2013/10/HR-Tail-f-NETCONF-WP-10-08-13.pdf (acesso em 2013)

que faz parte da rede gerenciada. Neste elemento podem existir diversos objetos gerenciados que são partes físicas ou lógicas do dispositivo, como uma interface de rede de um roteador, ou mesmo partes do software como, por exemplo, informações relativas a operação do protocolo de roteamento, número de pacotes recebidos, descartados, encaminhados e etc. Em cada dispositivo gerenciado existe um agente de gerenciamento que se comunica com a entidade gerenciadora/gerentes e executa ações específicas de acordo com solicitações dos gerentes. Uma base de informação denominada MIB (*Management Information Base*) é utilizada com o objetivo de apresentar dados quantitativos do dispositivo gerenciado à entidade gerenciadora.

3 - Protocolo de gerenciamento de rede. Atua entre o gerente e o agente, permitindo que o gerente consulte informações do dispositivo gerenciado e execute ações sobre eles mediante seus agentes, como alteração de valores (ROSS; KUROSE, 2005).

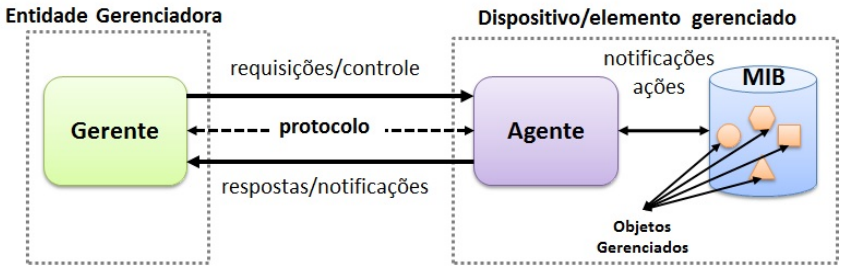

Figura 2.1: Arquitetura genérica de gerência SNMP.

As informações de um elemento gerenciado são apresentadas através de um conjunto de objetos, que formam a MIB. Os objetos da MIB são especificados na linguagem de definição de dados chamada SMI (*Structure of Management Information*), que é usada para modelar os detalhes do formato das informações. Essa linguagem é necessária para garantir um padrão na estrutura da MIB, existe 2 versões; SMIv1, definida nas RFCs 1155 e 1212; e SMIv2, definida nas RFCs 2578, 2579 e 2580. Do ponto de vista da organização do protocolo, podemos classificar a definição dos objetos gerenciados em três partes; nome ou identificador do objeto; tipo/sintaxe e codificação. **Nome ou identificador do Objeto (OID)**. Valor único e exclusivo para cada objeto, que é apresentado de forma numérica (numeração baseada na hierarquia ITU-T/ISO). **Tipo e sintax**. O *datatype* é definido pelo padrão ASN.1 (*Abstract Sintax Notation One*), é a sintaxe que define a estrutura correspondente dos tipos de objetos, ou seja, como os dados são representados e transmitidos entre os gerentes e agentes. **Codificação**. O objeto gerenciado é codificado em uma string de octetos que usa o BER (*Basic Encode Rules*), que define como os objetos são codificados e decodificados para serem transmitidos nas camadas de transporte inferiores, como Ethernet (MAURO; SCHMIDT, 2005) (WALSH, 2008).

Os objetos da MIB são nomeados e organizados de forma hierárquica de acordo com a estrutura de nomeação da ISO. Cada ramo da árvore possui um nome e um número, como apresentado na Figura 2.2. No topo da hierarquia estão a ISO e ITU-T e outro ramo com esforço em conjunto feito pelas duas organizações. Sob o ramo *Internet* da árvore (1.3.6.1) existem 7 categorias, sob o ramo *Private* (1.3.6.1.4) existe a lista Internet

Assigned Numbers Authority (IANA, 2009) com os nomes e códigos das empresas privadas, para livre consulta no site da instituição. Sob o ramo *Management* (1.3.6.1.2) e MIB-2 (1.3.6.1.2.1) finalmente estão as padronizações de códigos para a MIB.

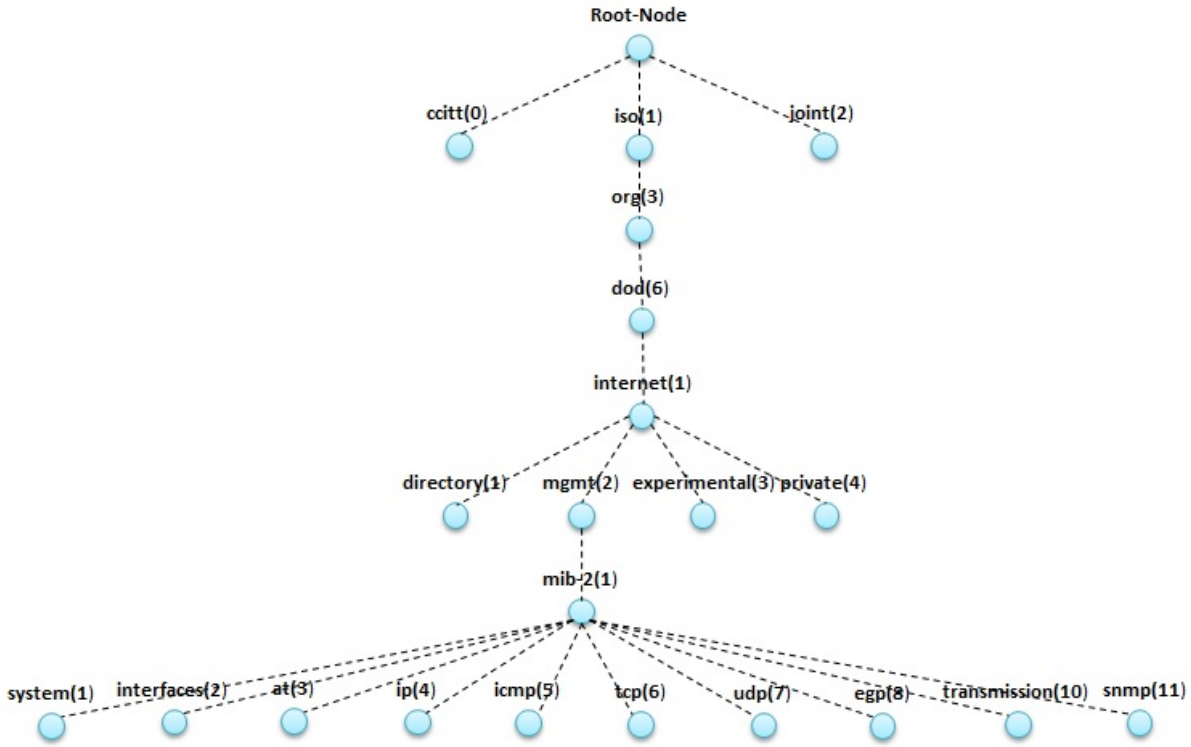

Figura 2.2: Sub-árvore da MIB-2

Sob o nó *mgmt* temos o nó MIB-2, onde estão os objetos usados para obter dados específicos dos dispositivos de rede. Esses objetos são divididos em 10 grupos, apresentados na Tabela 2.1.

Tabela 2.1: Grupos de objetos da MIB-2.

Grupo	**Informação**	**Identificador do Objeto (OID)**
system (1)	informações básicas do sistema	1.3.6.1.2.1.1
interfaces (2)	interfaces de rede	1.3.6.1.2.1.2
address translation (3)	tradução de endereços	1.3.6.1.2.1.3
ip (4)	protocolo ip	1.3.6.1.2.1.4
icmp (5)	protocolo icmp	1.3.6.1.2.1.5
tcp (6)	protocolo tcp	1.3.6.1.2.1.6
udp (7)	protocolo udp	1.3.6.1.2.1.7
egp (8)	protocolo egp	1.3.6.1.2.1.8
transmission (10)	meios de transmissão	1.3.6.1.2.10
snmp (11)	protocolo snmp	1.3.6.1.2.1.11

Operações do protocolo SNMP. O protocolo SNMP é utilizado para transportar informações da MIB entre Gerentes e Agentes. Neste contexto são permitidas ações que consultam e modificam valores de objetos da MIB associados a um elemento gerenciado. O SNMP também é utilizado para permitir que os agentes enviem mensagens (não solicitadas) caracterizadas como *eventos*. Essas mensagens são chamadas de *TRAPs*.

O SNMP apresenta algumas mensagens definidas como PDUs (Protocol Data Units), utilizadas para executar as principais operações do protocolo, conforme apresentados na Tabela 2.2.

Tabela 2.2: Descrição das operações do protocolo SNMP.

Tipo de PDU	**Remetente para receptor**	**Descrição**
GetRequest	gerente para agente	faz a leitura do valor de uma ou mais instâncias de objetos MIB
GetNextRequest	gerente para agente	faz a leitura do valor da próxima instância de objetos MIB na tabela
GetBulkRequest	gerente para agente	faz a leitura dos valores em grandes blocos de dados, valores em uma grande tabela
InformRequest	gerente para gerente	permite que um gerente envie informações relevantes diretamente a outros gerentes.
SetRequest	gerente para agente	define/altera valores de uma ou mais instâncias de objetos MIB
Response	agente para gerente ou gerente para gerente	gerada em resposta a todas as PDUs com exceção da Trap.
Trap	agente para gerente	informa um evento ao gerente

Segue a estrutura da troca de mensagens do protocolo SNMPv2, observe que a mensagem *trap* utiliza a porta UDP 162 e o restante das mensagens utilizam a porta 161.

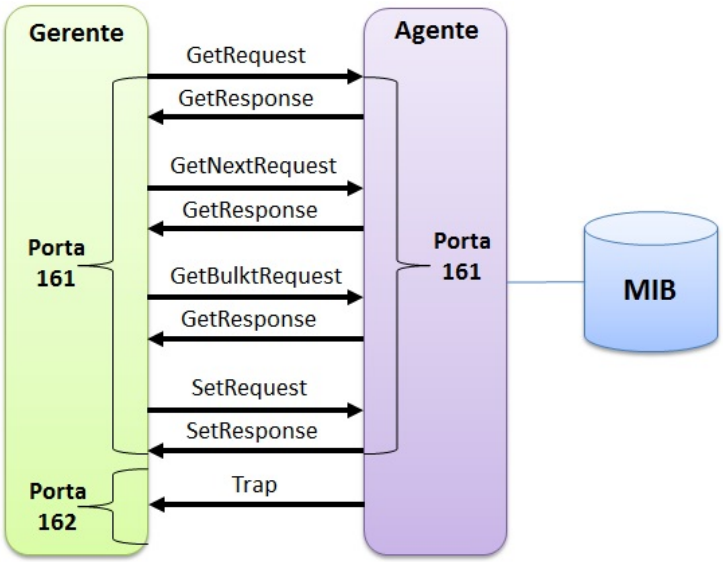

Figura 2.3: Troca de mensagens entre agentes e gerentes.

2.3 Modelo CCN

O modelo CCN (*Content-Centric Networking*) (JACOBSON *et al.*, 2009) utiliza basicamente dois tipos de pacotes: *Interest* e *Data*. Inicialmente o consumidor expressa seu interesse inserindo o nome do conteúdo desejado em uma mensagem do tipo *Interest* e a envia para rede. O produtor, ou algum caching no interior da rede, que possui tal conteúdo, receberá essa mensagem e enviará de volta ao consumidor uma mensagem do tipo *Data* como resposta. Ou seja, essas mensagens possuem uma relação um para um onde um pacote de dados satisfaz um de interesse se o nome de conteúdo em ambos os pacotes são equivalentes. A Figura 2.4 apresenta a estrutura das mensagens do modelo CCN. O pacote *Interest* conta com alguns parâmetros opcionais como seletores de escopo, preferência de ordem e filtro de exclusão. Embora pacotes *Data* não causem loop na rede CCN, pacotes *Interest* podem causar, para detectar e prevenir esse comportamento, os pacotes *Interest* possuem um valor aleatório *nonce* que quando recebidos de forma duplicada por caminhos diferentes podem ser descartados. O pacote *Data*, além do nome e do conteúdo, também carrega a assinatura e algumas informações opcionais como identificador do publicador e localização da chave para auxiliar na verificação da assinatura. O mecanismo de nomeação permite ao requisitante buscar o conteúdo posicionado em uma estrutura hierárquica. Esse mecanismo possui um controle de versionamento com marcadores que identificam a versão (v) do conteúdo e cada segmento (s). Esse conteúdo é acessado através do identificador hierárquico, por exemplo: *br.youtube/video/filme.avi/v1/s0*. Para acessar o próximo pedaço do conteúdo denominado *chunk* basta solicitar o próximo segmento,por exemplo: *br.youtube/video/filme.avi/v1/s1*.

Mensagens do modelo CCN:
- **Interest:** contem o *interesse* de um consumidor por um determinado conteúdo na rede.
- **Data:** são os conteúdos *satisfeitos* em resposta ao pacote de interesse.

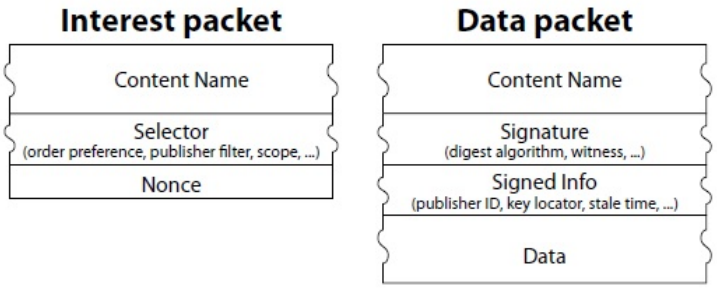

Figura 2.4: Mensagens CCN (reproduzido de (JACOBSON *et al.*, 2009)).

Cada elemento de rede CCN possui um buffer de memória para *cache*, que busca armazenar os pacotes de dados o maior tempo possível em uma estrutura denominada CS (*Content Store*), deste modo o mesmo conteúdo pode ser compartilhado por muitos consumidores. Quando um pacote *Interest* chega ao nó, se o conteúdo requisitado estiver armazenado no *cache* o pacote *Data* é imediatamente encaminhado na direção onde foi recebido o pacote de interesse. Caso contrário, o nó insere o nome do conteúdo desejado na PIT (*Pending Interest Table*) e a interface pela qual o pacote *Interest* foi recebido. Portanto, a PIT registra todos os interesses que passaram através do nó em busca do conteúdo. Quando o pacote de dados é recebido, logo na sequência é encaminhado corretamente em direção ao(s) consumidor(es). Apenas interesses são roteados no CCN; os pacotes de dados simplesmente seguem as entradas na PIT deixadas no caminho de volta ao consumidor. Estas entradas são apagadas assim que o pacote de dados correspondente é encaminhado ao consumidor ou por temporização, no caso em que o interesse não encontra o pacote de dados correspondente. Após registro na PIT, o pacote *Interest* é encaminhado pela FIB (*Forwarding Information Base*) do nó, através de uma busca de prefixo mais longo (*longest-prefix matching*) que indica por qual interface deve enviar o pacote *Interest*. Caso não haja uma entrada correspondente na FIB, o *Interest* é descartado. Da mesma forma que ocorre com a tabela de encaminhamento de roteadores IP convencionais, a FIB de um elemento de rede CCN também pode ser populada através de configuração manual ou de algum protocolo dinâmico de roteamento e descoberta de nomes, como por exemplo o OSFPN (WANG *et al.*, 2012) ou NSLR (HOQUE *et al.*, 2013).

Tabelas dos nós CCN são compostas por, CS, PIT e FIB:
- **Content Store (CS):** Buffer de memória para cache de dados.
- **Pending Interest Table (PIT):** registra pacotes de interesse não satisfeitos/pendentes.
- **Forward Information Base (FIB):** tabela de encaminhamento baseada em nomes hierárquicos.

A Figura 2.5 apresenta a estrutura do nó CCN e a dinâmica de encaminhamento.

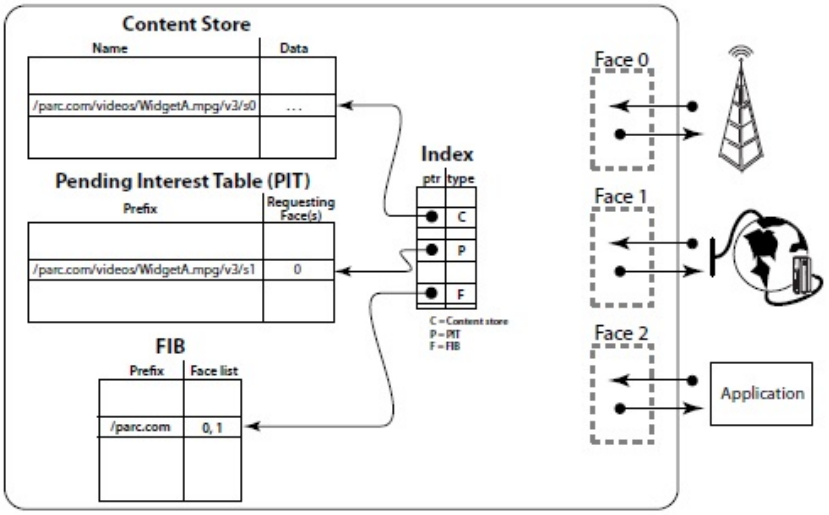

Figura 2.5: Arquitetura de roteamento do nó CCN (reproduzido de (JACOBSON *et al.*, 2009)).

2.4 Trabalhos relacionados

Nesta seção descreveremos sobre outros trabalhos que também apresentam propostas para gerência de redes orientadas a conteúdo.

A Networking Monitoring Tool for CCN

Como trabalho relacionado, a proposta definida em (KANG *et al.*, 2012) utiliza uma rede IP (*Internet Protocol*) convencional para transportar os pacotes *Interest* e *Data* entre os nós CCN. Neste cenário o protocolo IPFIX foi estendido para criar um agente IPFIX que captura os pacotes na rede IP (porta fixa UDP 9695) e converte para um formato XML (*Extensible Markup Language*) com os atributos relacionados ao CCN. Para os pacotes *Interest* são considerados como atributos; *message type*, *content name*, *chunk number*, *timestamp* e *address*. De forma similar para os pacotes *Data* são considerados como atributos; *content name* e informações de performance como; *bytes*, *packets* ou *data rate*. Com essas informações, o agente IPFIX cria um novo fluxo de dados que é encaminhado para um servidor central denominado *CCN Collector/Visualizer*. Esse servidor tem o papel de analisar as estatísticas do tráfego. Cada nó CCN também possui um agente SNMP que coleta diversas informações a respeito das características físicas do nó, tais como, CPU, memória, HDD, interfaces de rede e também informações a respeito das tabelas, como, CS, PIT e FIB. Uma MIB CCN é definida para coletar ou alterar dados dos objetos, e para o envio de mensagens de notificação ao NMS. As informações coletadas dos nós CCN pelos agentes IPFIX (tabelas de fluxos) e SNMP (tabelas de objetos monitorados) são exibidas em uma interface Web para o usuário.

Securing Building Management Systems Using Named Data Networking

O trabalho apresentado em (SHANG *et al.*, 2014) utiliza uma hierarquia de namespaces para posicionar cada objeto gerenciado, como feito na MIB, desse modo é possível identificar e consultar os objetos gerenciados, semelhante ao que ocorre no sistema de gerência SNMP. A proposta tem forte abordagem no aspecto de segurança, o sistema criptografa os pacotes IP e distribui as chaves de forma segura para múltiplos usuários, que promete ser mais eficiente e escalável em comparação às tradicionais soluções IP. Basicamente o sistema de gerenciamento foi desenvolvido para coletar dados de sensores em equipamentos industriais e distribuir estes dados e o controle de acesso para os usuários finais, aplicações e dispositivos de forma segura. Um Gateway atua entre a rede utilizada para a coleta de dados dos sensores (protocolo padrão BACnet sobre TCP/IP) e rede NDN. Este Gateway recebe os dados dos sensores e armazena em um repositório na rede NDN, o repositório é responsável por encaminhar pacotes de dados aos usuários em resposta aos pacotes de interesse, de acordo com cada *namespace/sensor*, essa intermediação evita ataques do tipo DOS/DDOS sobre os Gateways. O sistema NDN hierárquico é de fácil leitura, onde cada nome reflete exatamente a posição física de cada sensor na rede (ex.: /ndn/instituicao/predio/andar/sala/tipo-de-dado).

Conclusões comparativas

Ambas as ferramentas apresentam um modelo hierárquico de gerência, porém apenas o primeiro trabalho relacionado utiliza uma MIB CCN com o mesmo propósito da MIB definida para uso com a ferramenta SNMP Gateway CCN.

Este trabalho promete maior fidelidade em comparação com as outras ferramentas relacionadas, uma vez que surge com um agente CCN nativo e permite a tradução das operações básicas do protocolo SNMP para o modelo CCN. A Tabela 2.3 apresenta uma comparação de funcionalidades entre as propostas relacionadas e este trabalho.

Tabela 2.3: Comparativo das ferramentas para monitoramento com uso do modelo CCN.

Funcionalidades	**A Networking Monitoring Tool for CCN**	**Securing Building Management Systems Using Named Data Networking**	**SNMP Gateway CCN**
Modelo hierárquico para consulta de objetos	SIM	SIM	SIM
Define uma MIB CCN	SIM	NÃO	SIM
Agente e gerência de nós CCN nativos	NÃO	NÃO	SIM
Mapeamento das operações básicas do SNMP para CCN	NÃO	NÃO	SIM

Capítulo 3

Projeto e arquitetura para gerência de redes orientadas a conteúdo

Nesta seção apresentamos a proposta de arquitetura NONM (*Named-Oriented Network Management*) que define o SNMP como protocolo inicial a ser utilizado, bem como a criação de uma MIB exclusiva para tratar o modelo CCN. Por fim uma ferramenta para gerência de redes *in-band* denominada *SNMP Gateway CCN*.

3.1 Abordagem NONM: Name-Oriented Network Management

O objetivo deste trabalho é atender a carência de arquiteturas para gerência de redes orientadas a conteúdo e também permitir que as principais ferramentas de gerência de redes e de configuração existentes gerenciem elementos de rede nativos CCN com uso de um *gateway*. No modelo que chamamos de NONM (*Named-Oriented Network Management*) a proposta explora mecanismos de gerencia com orientação ao conteúdo (nomes/dados), que promete desempenho equivalente a arquiteturas tradicionais baseadas no endereçamento de interfaces dos nós.

3.2 Gerência de redes CCN baseada no protocolo SNMP

A modelagem de uma ferramenta SNMP Gateway CCN é o primeiro passo em direção à adoção de mecanismos para gerência de redes CCN, sejam nativas ou overlay. Para gerenciar elementos das redes orientadas a conteúdo, optamos pelo SNMP por se tratar de um protocolo largamente conhecido que se mostra mais eficiente do ponto de vista de monitoramento e alarmes em relação a outros protocolos que tem maior destaque na configuração, como por exemplo o protocolo NETCONF. A ferramenta SNMP Gateway CCN tem como uma de suas principais características o uso de uma arquitetura de gerência de redes *in-band*, que atua em conjunto com o plano de dados para monitoramento dos elementos de rede. Neste contexto o pacote de Interesse (requisição) é formado

pelo nome do conteúdo monitorado que deseja consultar e o *payload* do pacote de Dados (resposta) carrega o valor do objeto como resposta para a consulta realizada.

A construção de um modelo de mapeamento das funcionalidades mínimas de arquiteturas de gerência tradicionais para uso em CCN surge como uma proposta para a falta de arquiteturas de gerência de redes orientadas a conteúdo[4] que permitir também a coexistência de ferramentas gerentes de redes legadas interoperáveis com agentes em redes CCN nativas.

A Figura 3.1 apresenta de forma geral a ideia que busca a interoperabilidade de arquiteturas e protocolos convencionais de gerência com a rede de elementos CCN.

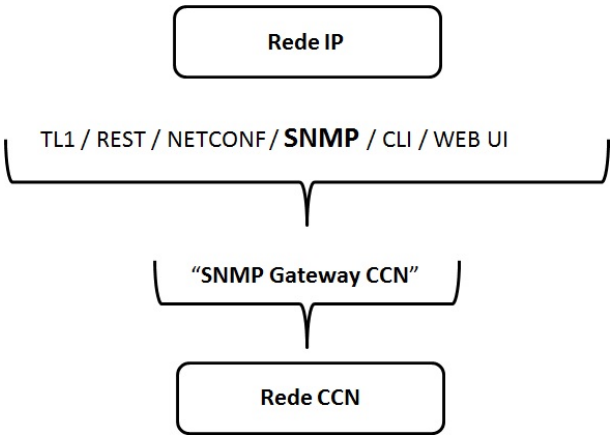

Figura 3.1: Modelo para gerência de redes CCN.

[4]A deficiência de arquiteturas de gerência de redes orientadas a conteúdo é apontada pelo grupo de pesquisa ICNRG (Information-Centric Networking Research Group) na **RFC7927** *ICN Challenges* tópico *4.8. Network Management.* https://datatracker.ietf.org/doc/rfc7927/ - July 2016.

3.3 Criação da MIB CCN

Neste trabalho definimos uma MIB para a rede CCN com o mesmo propósito da MIB apresentada em (KANG *et al.*, 2012), que se diferencia nos aspectos relativos à gerência de objetos refletidos em um agente CCN nativo, ao contrário de um agente SNMP convencional, e posicionamento sob o ramo MIB-2 (OID 100), ao invés do ramo *private*. Desta forma podemos utilizar a filosofia de extensibilidade da MIB padrão. A MIB CCN é apresentada de forma resumida na Figura 3.2.

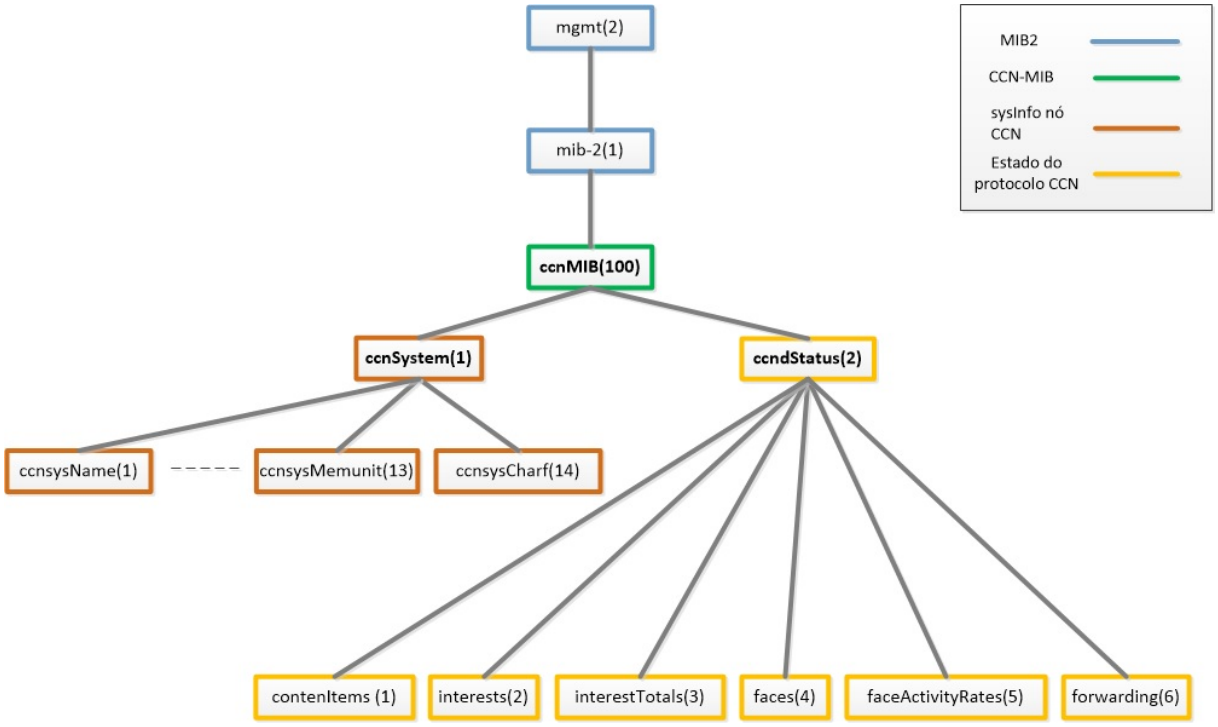

Figura 3.2: A MIB CCN e sua sub-árvore sob o ramo da MIB-2 estendida para suporte à gerência de elementos de redes CCN.

A MIB CCN proposta e modelada exclusivamente para este trabalho, tem como característica a criação de novos ramos na árvore que identificam objetos (OIDs) para o monitoramento de elementos de rede CCN em redes nativas. A MIB CCN fica sob hierarquia da MIB-2 e está classificada em duas partes, a primeira parte representa objetos *ccnSystem*, que tratam informações do próprio nó, a segunda parte *ccndStatus*, trata objetos específicos para monitoramento do protocolo CCNx através da coleta de dados da aplicação *ccndStatus*[5], que tem o mesmo nome. O nó CCN possui características que o torna único em relação à arquitetura de outros elementos da rede, pois ele possui tabelas de controle diferenciadas para tratamento e roteamento de pacotes/conteúdo.

[5]A *aplicação ccndStatus*, tem o papel de obter o estado da daemon CCND, definida no projeto Oficial CCNx. https://github.com/ProjectCCNx/ccnx/blob/master/doc/technical/CCNDStatus.txt.

As Tabelas 3.1 e 3.2 apresentam os objetos da MIB CCN classificados em 2 grupos principais *ccnSystem* e *ccndStatus* respectivamente, assim como a descrição de cada grupo e sub-grupo.

Tabela 3.1: Grupos de objetos da MIB CCN, ramo *ccnSystem*.

Grupos MIB com informações do nó CCN	**Informação**	**Identificador do Objeto (OID)**
ccnSystem (1)	Information on overall system statistics	1.3.6.1.2.100.1
ccnsysName (1.1)	Network Element System Name	1.3.6.1.2.100.1.1
ccnsysUptime (1.2)	Seconds since boot	1.3.6.1.2.100.1.2
ccnsysLoads (1.3)	1, 5, and 15 minute load averages	1.3.6.1.2.100.1.3
ccnsysTotalram(1.4)	Total usable main memory size	1.3.6.1.2.100.1.4
ccnsysFreeram (1.5)	Available memory size	1.3.6.1.2.100.1.5
ccnsysSharedram (1.6)	Amount of shared memory	1.3.6.1.2.100.1.6
ccnsysBufferram (1.7)	Memory used by buffers	1.3.6.1.2.100.1.7
ccnsysTotalswap (1.8)	Total swap space size	1.3.6.1.2.100.1.8
ccnsysFreeswap (1.9)	swap space still available	1.3.6.1.2.100.1.9
ccnsysProcs (1.10)	Number of current processes	1.3.6.1.2.100.1.10
ccnsysTotalhigh (1.11)	Total high memory size	1.3.6.1.2.100.1.11
ccnsysFreehigh (1.12)	Available high memory size	1.3.6.1.2.100.1.12
ccnsysMemunit (1.13)	Memory unit size in bytes	1.3.6.1.2.100.1.13
ccnsysCharf (1.14)	Padding to 64 bytes	1.3.6.1.2.100.1.14

Tabela 3.2: Grupos de objetos da MIB CCN, ramo *ccndStatus*.

Grupos MIB com informações do nó CCN	**Informação**	**Identificador do Objeto (OID)**
ccndStatus (2)	Informações sobre o estado do protocolo CCNx	1.3.6.1.2.100.2
contentItems (2.1)	Statistics related to CCNx Content Object flow	1.3.6.1.2.100.2.1
ciAccessioned (2.1.1)	Number of Content Objects accessioned	1.3.6.1.2.100.2.1.1
ciDuplicate(2.1.2)	Number of duplicate Content Objects	1.3.6.1.2.100.2.1.2
ciSent (2.1.3)	Number of Content Objects sent	1.3.6.1.2.100.2.1.3
ciSparse (2.1.4)	Number of Content Objects marked as sparse	1.3.6.1.2.100.2.1.4

Continua na próxima página

Tabela 3.2 – *Continuação da página anterior*

Grupos MIB com informações do nó CCN	**Informação**	**Identificador do Objeto (OID)**
ciStale (2.1.5)	Number of Content Objects marked as stale	1.3.6.1.2.100.2.1.5
ciStored(2.1.6)	Number of Content Objects stored	1.3.6.1.2.100.2.1.6
ciHost (2.1.7)	Host Name	1.3.6.1.2.100.2.1.7
ciTimestamp(2.1.8)	Time Stamp	1.3.6.1.2.100.2.1.8
interests (2.2)	Statistics related to Interest messages	1.3.6.1.2.100.2.2
iNames (2.2.1)	Number of Interest names	1.3.6.1.2.100.2.2.1
iNoted (2.2.2)	Number of noted Interests	1.3.6.1.2.100.2.2.2
iPending (2.2.3)	Number of pending Interests	1.3.6.1.2.100.2.2.3
iPropagating (2.2.4)	Number of propagating Interests	1.3.6.1.2.100.2.2.4
iHost (2.2.5)	Host Name	1.3.6.1.2.100.2.2.5
iTimestamp (2.2.6)	Time Stamp	1.3.6.1.2.100.2.2.6
interestTotals (2.3)	Statistics related to number of Interest messages	1.3.6.1.2.100.2.3
itAccepted (2.3.1)	Number of accepted Interests	1.3.6.1.2.100.2.3.1
itDropped (2.3.2)	Number of dropped Interests	1.3.6.1.2.100.2.3.2
itSent(2.3.3)	Number of sent Interests	1.3.6.1.2.100.2.3.3
itStuffed (2.3.4)	Number of stuffed Interests	1.3.6.1.2.100.2.3.4
iHost (2.3.5)	Host Name	1.3.6.1.2.100.2.3.5
iTimestamp (2.3.6) **faces** (2.4)	Time Stamp Contains the configured faces for this CCND node	1.3.6.1.2.100.2.3.6 1.3.6.1.2.100.2.4
fFace (2.4.1)	The face id	1.3.6.1.2.100.2.4.1
fFlags (2.4.2)	Hexidecimal value representing ccnd-private flags using names prefixed	1.3.6.1.2.100.2.4.2
fLocal (2.4.3)	Number of sent Interests	1.3.6.1.2.100.2.4.3
fPending (2.4.4)	The number of pending Interests on the face	1.3.6.1.2.100.2.4.4
fRemote (2.4.5)	Face connected to remote host	1.3.6.1.2.100.2.4.5
fHost (2.4.6)	Host Name	1.3.6.1.2.100.2.4.6
fTimestamp (2.4.7) **faceActivityRates** (2.5)	Time Stamp Statistics rates of face	1.3.6.1.2.100.2.4.7 1.3.6.1.2.100.2.5
farFace (2.5.1)	The current counter of face	1.3.6.1.2.100.2.5.1
farBytesIn (2.5.2)	Number of bytes in	1.3.6.1.2.100.2.5.2
farBytesOut (2.5.3)	Number of bytes out	1.3.6.1.2.100.2.5.3
farReceivedData (2.5.4)	Number of Content Objects in	1.3.6.1.2.100.2.5.4
farSentData (2.5.5)	Number of Content Objects out	1.3.6.1.2.100.2.5.5

Continua na próxima página

Tabela 3.2 – *Continuação da página anterior*

Grupos MIB com informações do nó CCN	**Informação**	**Identificador do Objeto (OID)**
farInterestsReceived (2.5.6)	Number of Interests in	1.3.6.1.2.100.2.5.6
farInterestsSent (2.5.7)	Number of Interests out	1.3.6.1.2.100.2.5.7
farHost (2.5.8)	Host Name	1.3.6.1.2.100.2.5.8
farTimestamp (2.5.9)	Time Stamp	1.3.6.1.2.100.2.5.9
forwarding (2.6)	Forwarded Entry	1.3.6.1.2.100.2.6
fwFace (2.6.1)	The current FIB Prefix Name	1.3.6.1.2.100.2.6.1
fwFlags (2.6.2)	The integer containing the inclusive OR of the Forwarding Flags	1.3.6.1.2.100.2.6.2
fwPath (2.6.3)	The current path	1.3.6.1.2.100.2.6.3
fwExpires (2.6.4)	Also known as Freshness Seconds, the remaining lifetime on the face	1.3.6.1.2.100.2.6.4
fwHost (2.6.5)	Host Name	1.3.6.1.2.100.2.6.5
fwTimestamp (2.6.6)	Time Stamp	1.3.6.1.2.100.2.6.6

3.4 Estratégias para mapeamento das operações básicas do SNMP

A Figura 3.3 apresenta a relação de comunicação entre *gerentes e agentes*, utilizada pelo protocolo SNMP com base na pilha TCP/IP.

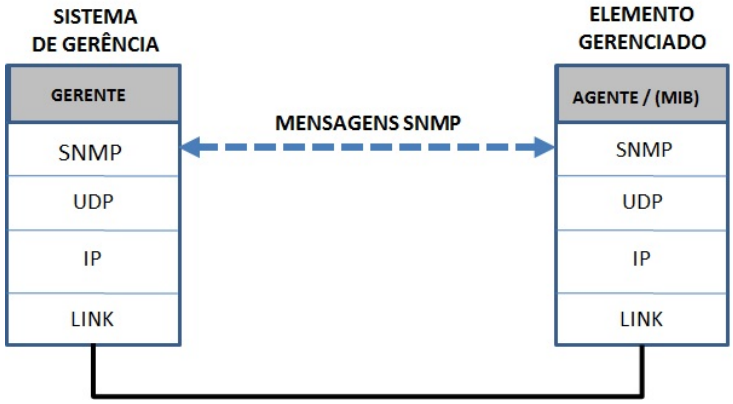

Figura 3.3: Relacionamento de gerente e agente baseado na pilha TCP/IP.

Levando em consideração que os elementos da rede CCN não suportam o protocolo SNMP, é necessário o uso de um gateway SNMP que permite o mapeamento das operações básicas do protocolo SNMP para monitoramento dos elementos nativos da rede CNN. Neste contexto um agente SNMP executado no gateway deve conhecer os objetos da MIB CCN para estabelecer a interface de comunicação entre as redes IP e CCN. Com esse propósito, desenvolvemnos um mecanismo de mapeamento denominado SCNT (*SNMP Content Network Translation*) que faz o papel de tradutor na arquitetura da ferramenta SNMP Gateway CCN, apresentada na Figura 3.4.

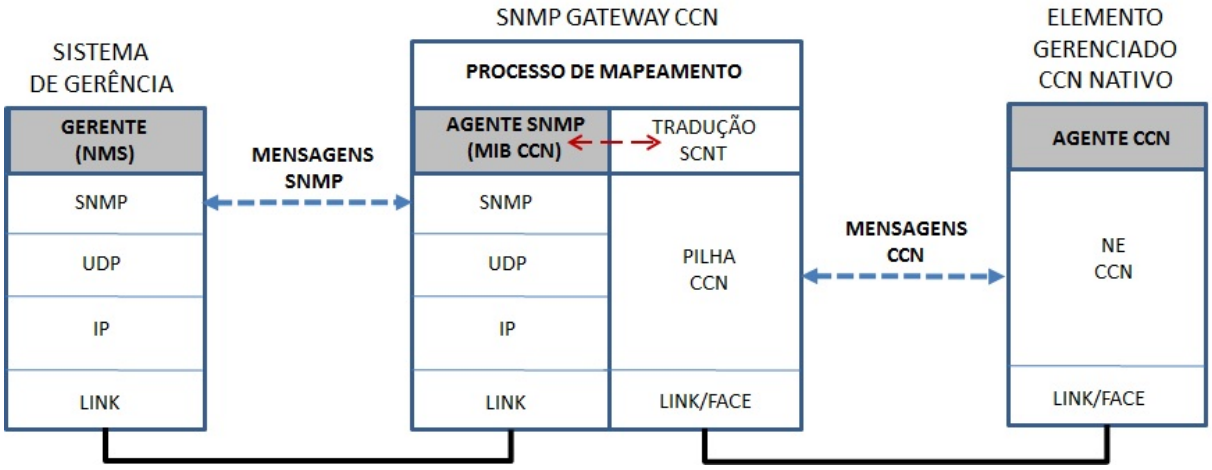

Figura 3.4: Arquitetura geral para mapeamento das mensagens SNMP para CCN.

A arquitetura utilizará dois tipos de agentes, um *agente SNMP* e um *agente CCN nativo*, como descritos a seguir.

Agente SNMP: Responsável por mapear as consultas SNMP de objetos (OIDs) da MIB CCN para pacotes *Interest* que serão encaminhados para a rede CCN.

Agente CCN: O agente CCN nativo, tem como principal objetivo fornecer conteúdo (Dados) em resposta às requisições de consultas (Interesse). O conteúdo fornecido pelo Agente CCN, representa um conjunto de objetos/informações de um elemento de rede CCN gerenciado, estes objetos estão mapeados em OIDs especificados na MIB CCN.

3.5 Mapeamento das operações básicas do protocolo SNMP através do SCNT

3.5.1 Nomeação e descoberta dos nós

Antes do processo de mapeamento das operações básicas do SNMP para CCN, é necessário conhecer o nome (prefixo mais longo e identificador único) de cada elemento na rede CCN (ex.: /<network>/site/<ne>/). A descoberta dos nomes destes elementos pode ser feita de forma automática através do processo de descoberta de topologia, geralmente realizado por um protocolo de roteamento dinâmico, como apresenta a proposta (HOQUE *et al.*, 2013).

Descoberta de nomes. O SNMP Gateway CCN poderá manter uma tabela dinâmica com os nomes dos nós conhecidos na rede, como mostra a Figura 3.5.

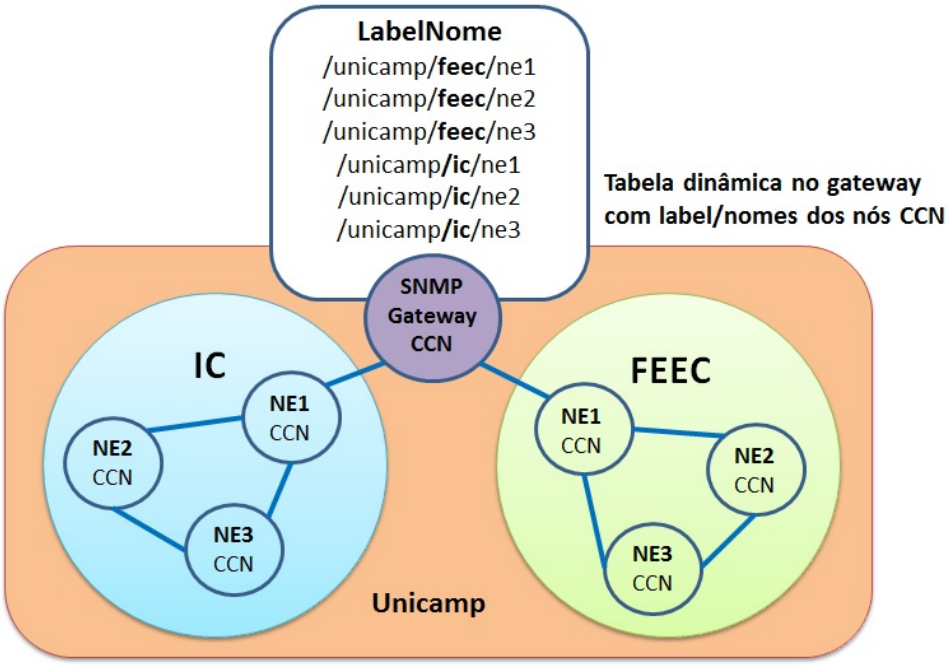

Figura 3.5: Arquitetura de nomeação e descoberta.

3.5.2 Mecanismo de tradução SCNT

O SNMP Gateway CCN deve executar um agente SNMP que tem com principal objetivo mapear a MIB CCN de modo que possa acessar os objetos definidos para gerenciamento dos nós CCN.

O campo *ContextName* (string formada em texto plano) da mensagem PDU do protocolo SNMPv3 é usado como passagem de parâmetro para informar ao gateway qual elemento deseja consultar na rede CCN. O campo *Community* existente nas versões SNMPv1 e SNMPv2 do protocolo também poderia ser usado para esse propósito, porém a funcionalidade do campo teria que ser re-definida. Portanto, a ferramenta SNMP Gateway CCN não possui suporte para as versões 1 e 2 do protocolo.

Consulta

O elemento de rede gateway executa um agente SNMP com todos os *OIDs* da MIB CCN mapeados. O agente SNMP utiliza o campo *contextName* da mensagem SNMPv3 como parâmetro para identificar o nó de destino na rede CCN, o campo *ObjectName* que representa o *OID* do objeto que deseja consultar é utilizado no processo para formar a mensagem de *Interesse* enviada para rede CCN.

O Gerente (NMS) envia uma mensagem GET formada pelos campos *contextName* e *OID* para o agente SNMP na porta padrão *161*, se o valor do *OID* faz parte da MIB CCN a ferramenta saberá que a mensagem recebida deve ser convertida e encaminhada para o mundo CCN.

O campo *Request ID* do protocolo SNMP é utilizado para identificar as mensagens de requisição geradas pelo processo gerente, uma vez que um gerente pode fazer múltiplas requisições SNMP para o mesmo agente. Múltiplas consultas de gerentes distintos para o mesmo objeto geram vários pacotes *Interest*, um para cada consulta.

A Figura 3.6 apresenta o formato da mensagem do protocolo SNMPv3, com destaque para os campos *ContextName* e *Object Name/OID*, utilizados no processo de mapeamento de consulta do mecanismo SCNT.

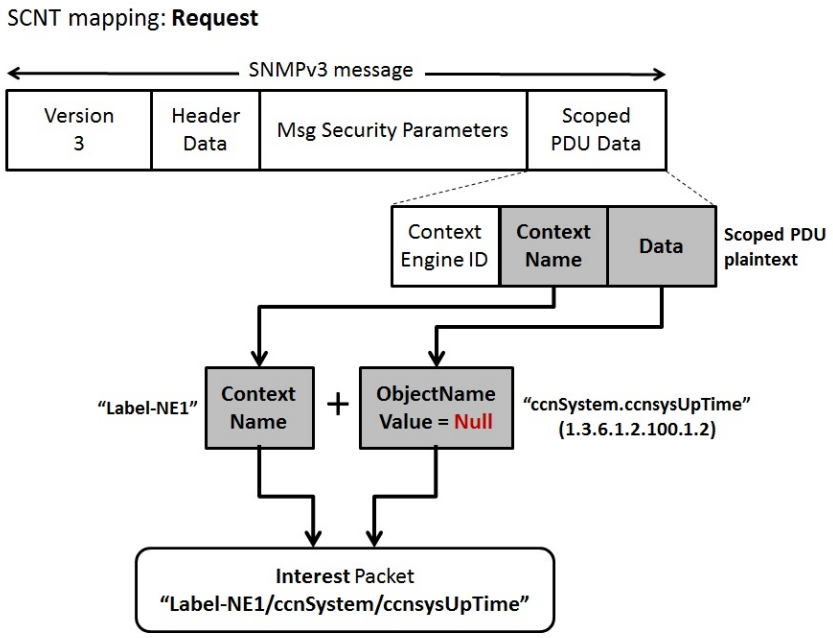

Figura 3.6: Formato geral da mensagem SNMPv3 utilizada no processo de mapeamento durante a consulta, com destaque para os campos *contextName* e *ObjectName*.

Resposta

Após a entrega da mensagem *Interest* do SNMP Gateway CCN para a rede CCN nativa formada pelo mapeamento descrito anteriormente, a rede deve retornar como resposta um pacote *Data* levando em consideração a arquitetura do modelo CCN. Quando o SNMP Gateway CCN recebe o pacote *Data* de volta como resposta, a mensagem *PDU Response* é formada de acordo com o conteúdo recebido e é encaminhada de volta para o Gerente (NMS) que fez a solicitação no início.

A Figura 3.7 apresenta o formato da mensagem do protocolo SNMPv3, com destaque para o campo *Data*, utilizado no processo de mapeamento de resposta do mecanismo SCNT.

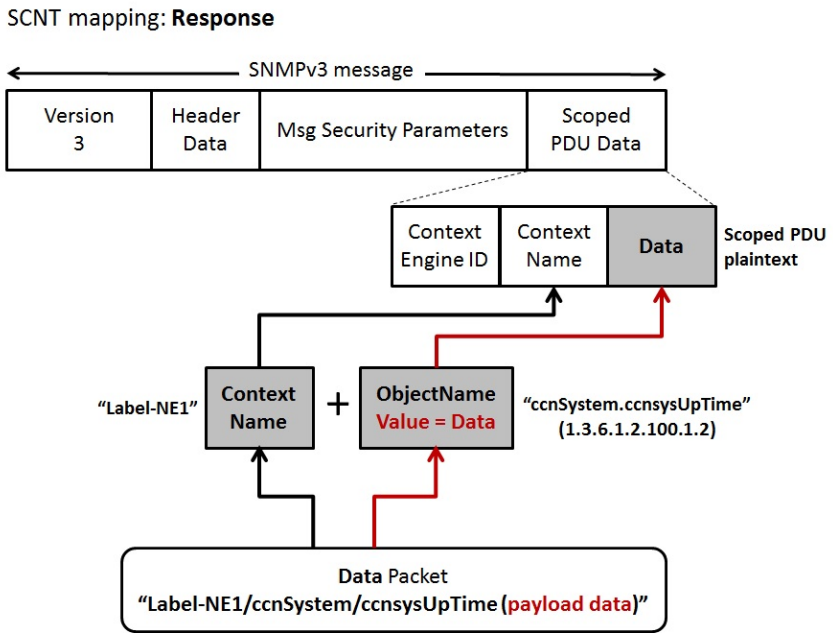

Figura 3.7: Formato geral da mensagem SNMPv3 utilizada no processo de mapeamento durante a resposta, com destaque para o campo *Data*, que tem o *payload* preenchido com informações de gerência do objeto consultado.

3.5.3 Mapeamento das operações básicas do SNMP para CCN

Operação GET

Com uso da arquitetura SCNT é possível iniciar uma consulta ao nó CCN nativo através da operação *GET* do SNMP. Como exemplo, iremos usar uma consulta feita para o elemento CCN com nome NE2 para mostrar como deve ser o fluxo das mensagens, NMS> GET Request> SNMP Gateway CCN> Interest> CCN Node, NMS <GET Response <SNMP Gateway CCN <Data <CCN Node, em uma topologia de 2 elementos, NE1 e NE2.

O sistema Gerente NMS inicia uma consulta SNMP *GET Request* ao OID *1.3.6.1.2.1.100.1.2* para o SNMP Gateway CCN, que reflete exatamente o objeto *ccnSystem/ccnsysUptime* (1). Para a formação da mensagem *SNMP GET*, o campo *contextName* deve ser preenchido com o Label/Nome do nó que deseja consultar, como exemplo, *NE2*. Com o nome *NE2*(contextName=NE2) e o OID *ccnSystem/ccnsysUptime*, o SNMP Gateway CCN converte a mensagem para o pacote *Interest NE2/ccnSystem/ccnsysUpTime* e encaminha o pacote para a rede de elementos CCN (2). Se o elemento *NE1* possui o conteúdo, ele responde a requisição imediatamente com a mensagem de dados. Neste caso, NE1 não possui o conteúdo em seu *cache*, como a consulta está direcionada ao elemento *NE2*, o *NE1* armazena o nome do Interesse em sua tabela *PIT* e encaminha a mensagem para os próximos nós na rede até que o conteúdo seja localizado (3), conforme característica de roteamento de um nó de rede CCN nativo. Quando o pacote *Interest* alcança o elemento alvo, o pacote *Data NE2/ccnSystem/ccnsysUpTime* é formado por *NE2* em resposta ao pacote *Interest*. O conteúdo localizado é armazenado no *cache*[6] do

nó *NE1* e em cada *NE* no caminho de volta (4). Em seguida a mensagem *Data* é entregue ao SNMP Gateway CCN (5) que a converte para uma mensagem padrão SNMP *GET Response* encaminhada de volta ao sistema Gerente NMS (6). Os passos descritos são apresentados na Figura 3.8.

Quando um determinado objeto fornece valores *dinâmicos* (ex.: contadores para pacotes *Interest* e outros), podemos utilizar o mecanismo de *versionamento*, ou seja, uma nova versão estará disponível com as informações mais recentes do objeto. Nestes casos o cliente poderá solicitar a versão mais recente do conteúdo e os valores dos objetos dinâmicos permanecerão em cache até que sejam atualizados, evitando assim que os valores antigos permanecem em cache.

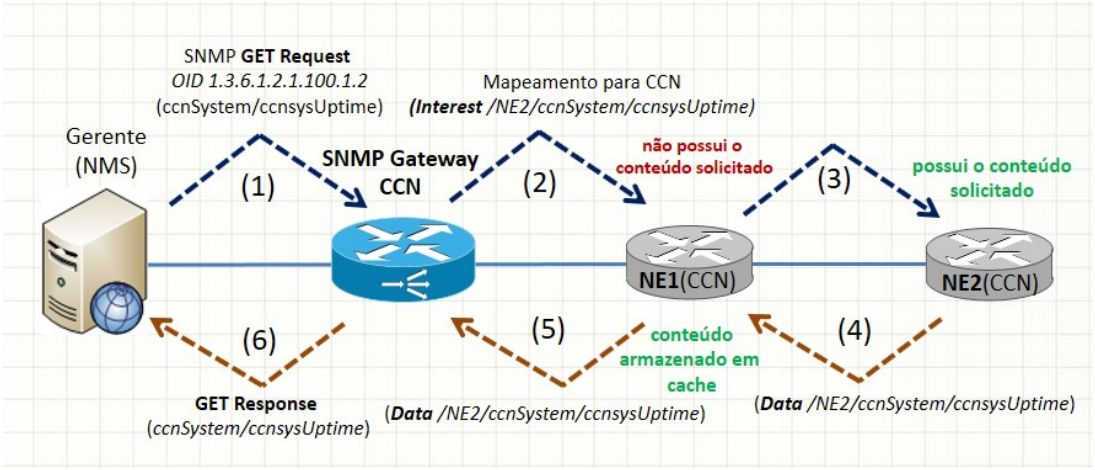

Figura 3.8: Passos para mapeamento da consulta da operação *GET*.

Operação GET-NEXT

O sistema Gerente NMS inicia a consulta SNMP *GET-NEXT Request* tendo como referência o objeto *1.3.6.1.2.100.1.1* **+1**, que reflete *ccnSystem/ccnsysName*, dessa forma o objeto *1.3.6.1.2.100.1.2*, que reflete *ccnSystem/ccnsysUpTime* seguinte na hierarquia sob o mesmo ramo da árvore, será o objeto efetivamente consultado. O SNMP Gateway CCN converte a mensagem para o pacote *Interest labelNEx/ccnSystem/ccnsysName*, o restante do processo de mapeamento é feito seguindo o mesmo princípio da operação *SNMP GET*.

Fica claro que, do ponto de vista do pacote *Interest*, não existe a operação *GET-NEXT*, ou seja, os tipos de operações do protocolo SNMP são irrelevantes do ponto de vista do mundo CCN.

[6]O *cache* em gerência de redes CCN tem maior benefício nos casos em que os conteúdos armazenados fazem referência a valores de *objetos estáticos* (ex.: número de interfaces de rede físicas de um elemento e etc), de outro modo é recomendado o uso do mecanismo de versionamento para a consulta de *objetos dinâmicos*.

Operação GET-BULK

O sistema Gerente NMS inicia a consulta SNMP *GET-BULK Request* de acordo com o número de objetos que deseja consultar seguintes na hierarquia sob o mesmo ramo da árvore, conforme defino no parâmetro *max-repetitions* da PDU SNMP *GET-BULK Request*, como exemplo, para objeto *1.3.6.1.2.100.1.1* **+3** *ccnSystem*, que reflete os objetos *1.3.6.1.2.100.1.1 ccnSystem/ccnsysName*, *1.3.6.1.2.100.1.2 ccnSystem/ccnsysUpTime* e *1.3.6.1.2.100.1.3 ccnSystem/ccnsysLoads*. O Gateway SNMP CCN converte a mensagem para os pacotes *Interest* relativos, o restante do processo é o mesmo feito para a operação SNMP GET.

Operação SET

Com a operação *SET* é possível alterar o valor de um objeto específico na rede de nós CCN. Como exemplo, podemos usar uma solicitação de alteração para o objeto *ccnsysName* do elemento *NE1*.

O sistema Gerente NMS solicita a alteração *SET Request* para o objeto com identificação *1.3.6.1.2.100.1.1*, que reflete o objeto *ccnSystem/ccnsysName*, com o valor de label/nome para *NE1* do tipo *OctetString* definido para o campo *Value* da PDU *SET Request* (1). O SNMP Gateway CCN converte a mensagem para um pacote de Interesse, exemplo *Interest NE1/ccnSystem/ccnsysName* e envia o pacote de Interesse para a rede de elementos CCN juntamente com o novo valor que deve ser substituído no elemento alvo (2). Se o elemento *NE1* é responsável pelo conteúdo *NE1/ccnSystem/ccnsysName*, ele altera imediatamente o valor do campo informado para o novo valor, em seguida um pacote de dados *Data NE1/ccnSystem/ccnsysName* é gerado e encaminhado de volta (3). O SNMP Gateway CCN recebe o pacote de dados, converte para um pacote SNMP *SET Response* e encaminha de volta para Gerente NMS indicando que o valor foi alterado com sucesso (4). Os passos descritos são apresentados na Figura 3.9.

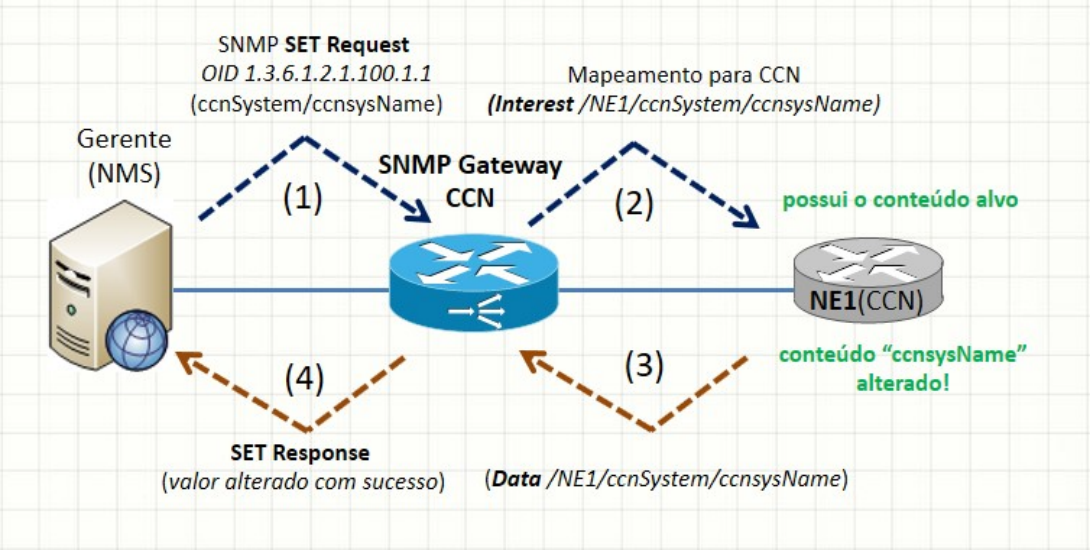

Figura 3.9: Passos para mapeamento da consulta da operação *SET*.

3.6 Mapeamento da operação de notificação de eventos (TRAP)

Como já mencionado, o protocolo SNMP utiliza a operação *TRAP* para notificar o gerente sobre alterações inesperadas de valores dos objetos gerenciados de um elemento de rede. Nas redes orientadas a conteúdo, temos como proposta a utilização da arquitetura *Publish/Subscribe Event Notification* para atender a necessidade do mecanismo para notificação de eventos.

Arquitetura Publish/Subscribe

A arquitetura *Publish/Subscribe* está baseada no princípio onde um publicador *publisher* produz uma informação específica que é consumida por um assinante *subscriber* que sinaliza o interesse prévio por essa informação. Neste sentido, o fluxo da informação é originado dos notificadores *(pub)* para os receptores *(sub)*, os receptores não recebem a informação diretamente do notificador, porém estão indiretamente relacionados de acordo com o tipo/conteúdo da notificação. Primeiro um assinante demonstra interesse por determinado conteúdo independente da sua publicação, de forma assíncrona um publicador notifica/publica todos os seus os conteúdos, em seguida cada receptor aceita/recebe o conteúdo o qual havia demonstrado interesse. Para evitar que cada publicador tenha conhecimento de todas as assinaturas para cada possível assinante, um elemento chamado *Notification Service* é utilizado para intermediar a comunicação. Ambos, publicador e assinante, mantêm comunicação apenas com o *Notification Service*, desta forma o publicador e o assinante trocam informações sem a necessidade de se conhecerem, esse anonimato é a principal característica da arquitetura *Publish/Subscribe* (VIRGILLITO, 2003). A Figura 3.10 apresenta a arquitetura Pub/Sub.

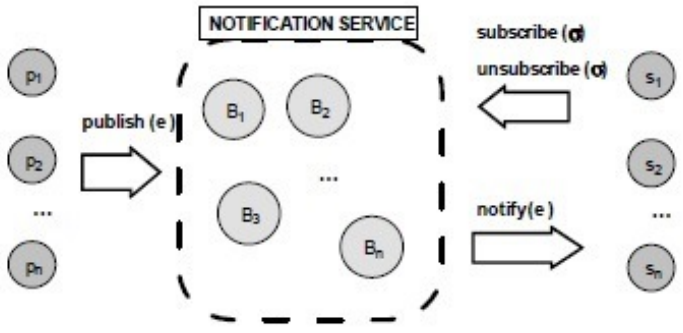

Figura 3.10: Visualização em alto nível da arquitetura *Publish/Subscribe* (reproduzido de (VIRGILLITO, 2003)).

Uso da arquitetura Publish/Subscribe Event Notification

Com base no modelo de comunicação *Publish/Subscribe* (VIRGILLITO, 2003) (EUGSTER *et al.*, 2003) (CARZANIGA *et al.*, 2011), temos como proposta inicial o uso do mecanismo *Publish/Subscribe Event Notification* para tratar as notificações de eventos na plataforma SNMP Gateway CCN. A ferramenta SNMP Gateway CCN deve implementar o processo *Forwarding Tables/Notification Service* que fará o gerenciamento das mensagens de publicação e assinaturas de eventos, também deverá implementar o processo *Sub-agente Consumer* que deve agir como *Consumer/Subscriber* do sistema. O Agente CCN deve implementar o processo *Producer/Publisher* que fornecerá a publicação dos eventos relacionados aos objetos gerenciados que podem mudar o seu estado (ex.: *linkDown*, *linkUP*).

Os processos necessários na ferramenta SNMP Gateway CCN e nó CCN para funcionamento do mecanismo de notificação de eventos estão classificados abaixo:

SNMP Gateway CCN
- Forwarding tables/Notification Service
- Consumer/Subscriber

Nó CCN
- Producer/Publisher

Mapeamento da operação TRAP do SNMP para CCN

O agente SNMP deve cadastrar os sistemas gerentes que receberão as *TRAPs*, como é feito no modo convencional (1). A ferramenta SNMP Gateway CCN deve implementar os processos *Consumer* e *Notification Service*, o processo *Consumer* expressará ao processo *Notification Service* o interesse em eventos específicos que deseja monitorar, como por exemplo; *Interest labelNE/ccnSystem/linkDown* e *Interest labelNE/ccnSystem/linkUP* (2). O processo *Producer* implementado no elemento CCN deve informar ao Gateway a publicação dos conteúdos *LabelNE/ccnSystem/linkDown* e *LabelNE/ccnSystem/linkUP* (3). Se o conteúdo/estado do *Interest labelNE/ccnSystem/linkUP* é alterado para *Interest labelNE/ccnSystem/linkDown*, o mesmo é imediatamente notificado ao Gateway (4). Em seguida o pacote é convertido para o formato de uma *TRAP* do SNMP e encaminhado para os sistemas gerentes cadastrados para receber as *TRAPs* (5). Os passos descritos são apresentados na Figura 3.11.

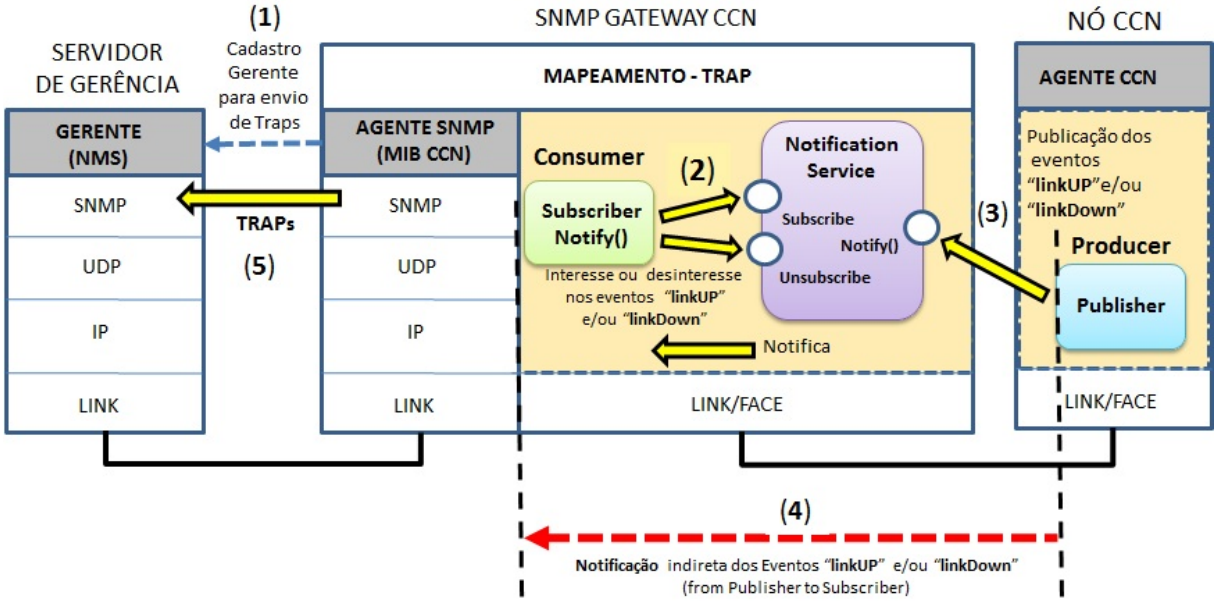

Figura 3.11: Arquitetura *Publish/Subcribe Event Notification* para o SNMP Gateway CCN.

Capítulo 4

Implementação da prova de conceito

4.1 Visão geral

Para a prova de conceito, a ferramenta *SNMP Gateway CCN* foi implementada tendo como foco a operação *GET*, levando em consideração as garantias das funcionalidades mínimas, como consulta e resposta de um determinado objeto da MIB CCN.

A plataforma *MiniCCNx* (CABRAL *et al.*, 2013) foi utilizada como ambiente para troca de mensagens entre o *gerente IP* e o *Agente CCN*, que também compõe a rede de elementos CCN. A plataforma *MiniCCNx* foi escolhida por ser considerada robusta e confiável para coleta de evidências que comprovem o funcionamento e eficácia dos experimentos. Do ponto de vista de um ambiente CCN nativo, a plataforma MiniCCNx possui características de um emulador que permite executar aplicações reais, necessárias para auxiliar na interação do *SNMP Gateway CCN* com o agente nativo do ambiente emulado.

A ferramenta SNMP Gateway CCN é composta basicamente por; Agente SNMP, mapeamento dos objetos da MIB CCN, ferramentas *ccnmanager* e *ccnagent* (Agente CCN nativo) que compõem a arquitetura *SCNT*.

Agente SNMP. O Agente SNMP é capaz de interpretar as consultas SNMP feitas para o objetos mapeados na MIB CCN da hierarquia, em seguida traduzir as mensagens que são encaminhadas para rede de elementos CCN nativos.

MIB CCN.A MIB CCN deve implementar todos os objetos (OIDs) definidos para gerência e monitoramento dos elementos CCN nativos. A implementação da MIB segue as especificações da RFC 3418, definida com base no mecanismo SMI padrão.

Agente CCN. O agente CCN nativo tem como principal objetivo fornecer conteúdo em resposta às requisições originadas no gateway, os conteúdos devem refletir exatamente cada objeto do elemento de rede gerenciado, diretamente relacionado com os OIDs especificados na MIB CCN.

Arquitetura SCNT. A arquitetura SCNT apresenta os mapeamentos necessários para a tradução das mensagens SNMP para CNN e CNN para SNMP, tendo como base os estados; consulta e resposta. Dois dos principais campos do protocolo SNMPv3 devem compor a mensagem mapeada para consultas, são; *contextName* e *ObjectName*. O conteúdo destes campos serão utilizados para formar o pacote *Interest*. O campo *Data* deve compor a mensagem de resposta, utilizado para formar o pacote *Data*, arquitetura descrita com maiores detalhes na Seção 3.5.2 (Mecanismo de tradução SCNT).

A ferramenta SNMP Gateway CCN é capaz de receber mensagens SNMP do tipo *Request* e traduzi-las para mensagens CCN do tipo *Interest*, o mesmo processo deve ocorrer no sentido inverso, ou seja, receber mensagens CCN do tipo *Data* e convertê-las para mensagens SNMP do tipo *Response*. O mecanismo de tradução atua de forma totalmente transparente do ponto de vista do sistema NMS. A Figura 4.1 apresenta uma visão geral da ferramenta.

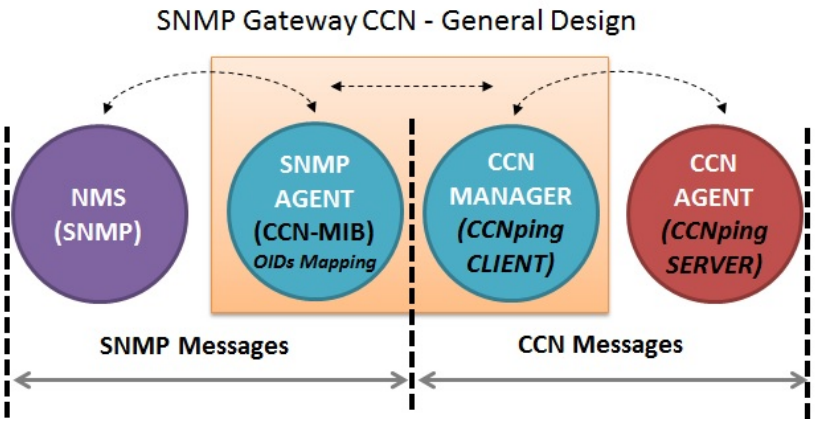

Figura 4.1: Visão geral da ferramenta SNMP Gateway CCN

4.2 Implementação

Para a implementação da ferramenta foram utilizadas as linguagens C e Python, presentes no projeto *MiniCCNx* e na ferramenta *CCNping* módulo *Server*, que teve parte do seu código fonte alterado com o objetivo da criação do ***Agente CCN***, uma das propostas deste trabalho. Também utilizamos o módulo *CCNping Client* como parte da implementação do ***Agente SNMP*** para compor a ferramenta *SNMP Gateway CCN*.

Como o ambiente MiniCCNx é Emulado com Base em Containers (EBC), foi possível adotar o reuso destas aplicações, fundamentais para o desenvolvimento da ferramenta.

4.2.1 Agente SNMP

No elemento de rede Gateway:

Agente SNMP. Criado para registrar os OIDs da MIB CCN e fazer o papel de gateway entre o mundo IP e CCN. Como base para desenvolvimento do agente SNMP utilizamos o projeto *Net-SNMP*[7] compilado na versão 5.7.2. O agente foi estendido para registrar todos os OIDs da MIB CCN, que tem o papel fundamental de traduzir mensagens *SNMP Request* e gerar as mensagens *Interest* para a rede CCN. Neste processo o Agente SNMP executa a aplicação *ccnmanager* e em uma linha de comando passa como parâmetro os dados mapeados dos campos *contextName* e *ObjectName/OID*. Exemplo, para uma consulta realizada ao elemento de rede com nome *r1* e *OID 1.3.6.1.2.1.100.1.4* (ccnSystem/Totalram), o Agente SNMP deve executar a linha de comando ***"ccnmanager -c 1 ccnx:/r1/ccnSystem/ccnsysTotalram"*** a partir do *container* gateway. O parâmetro *-c 1* nativo da ferramenta *ccnping* indica que uma única mensagem deve ser gerada no processo. Com esta linha de comando a aplicação *ccnmanager* irá formar o pacote *Interest* e o encaminhará para tratamento da instância *ccnd* (CCN daemon) do próprio container no ambiente MiniCCNx.

4.2.2 Agente CCN

No elemento de rede CCN nativo(gerenciado):

Agente CCN. A aplicação *CCNping Server* foi estendida para fornecer conteúdo com os valores das mensagens *Data* para gerência de objetos do próprio nó, em resposta ao pacote *Interest*. Cada valor fornecido pelo Agente CCN representa um objeto da MIB CCN. Estes valores são classificados conforme os dois grupos apresentados na MIB CCN, ***ccnSystem*** e ***ccndStatus***.

Grupo ***ccnSystem***. O Agente CCN consulta um conjunto de dados de sistema e hardware retornados pela função *sysInfo()* nativa do Linux como fonte de dados para consultas à objetos do grupo ***ccnSystem***.

Grupo ***ccndStatus***. Para o grupo *ccndStatus*, o Agente CCN consulta e utiliza o mesmo conjunto de métricas que são extraídas da saída da aplicação *daemon ccndStatus*, tratadas e armazenadas no banco de dados influxDB no ambiente MiniCCNx com tempo de ***polling*** (frequência da extração de dados) a cada 30 segundos. Para a obtenção dos dados, os valores são interceptados diretamente no espaço de código *Python* do módulo coletor de métricas definifo em (ELIAS, 2015) no formato JSON (*JavaScript Object Notation*) e armazenados no sistema de arquivos no espaço do Linux O.S na pasta ***/home/user/ccndStatus-ObjectValues***. Deste modo os arquivos de texto gerados no processo servem como fonte de dados para consultas a objetos do grupo ***ccndStatus***. A Figura 4.2 apresenta os blocos funcionais do processo.

[7] O projeto Net-SNMP é de código aberto, licenciado como Software Livre para desenvolvimentos de aplicações SNMP compatíveis com Unix O.S. http://www.net-snmp.org/

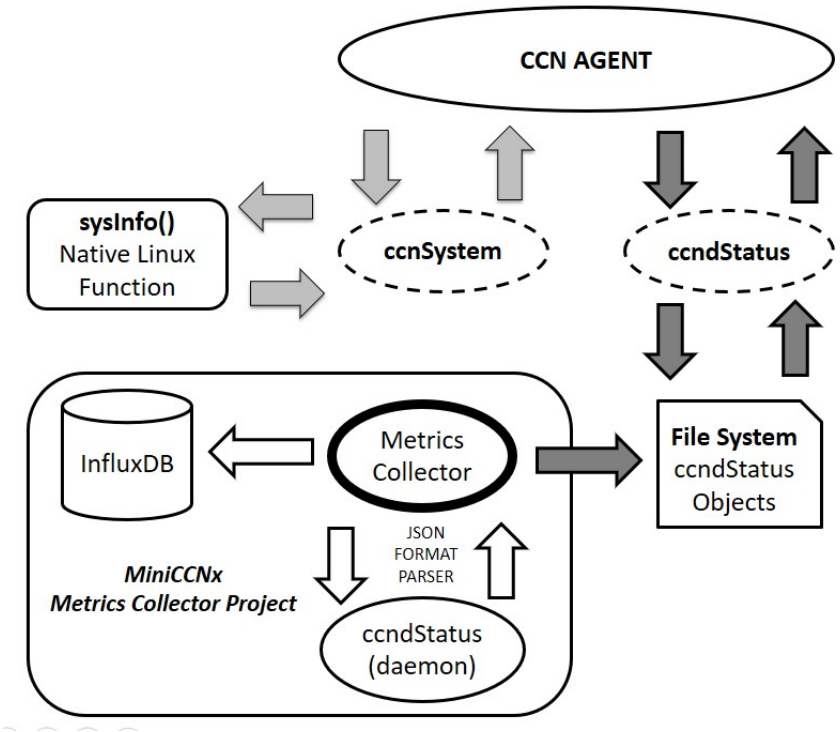

Figura 4.2: Blocos funcionais que apresentam as fontes de coleta de dados para o Agente CCN.

Para fins de experimentação e prova de conceito, utilizamos o MIB Browser *front-end* grafico ***SnmpB***, para carregar a MIB CCN e gerar as mensagens para consultas básicas do SNMP. O SnmpB é uma aplicações real que deve rodar em um *container* da plataforma *MiniCCNx* juntamente com as demais aplicações que compõem o *SNMP Gateway CCN*. Demais *containers* devem rodar apenas o ***Agente CCN*** nativo.
Toda a estrutura necessária para o funcionamento mínimo da ferramenta *SNMP Gateway CCN* foi implementada com uso de processos agentes (SNMP legado/ CCN nativo). A Figura 4.3 apresenta uma visão detalhada da ferramenta no ambiente MiniCCNx.

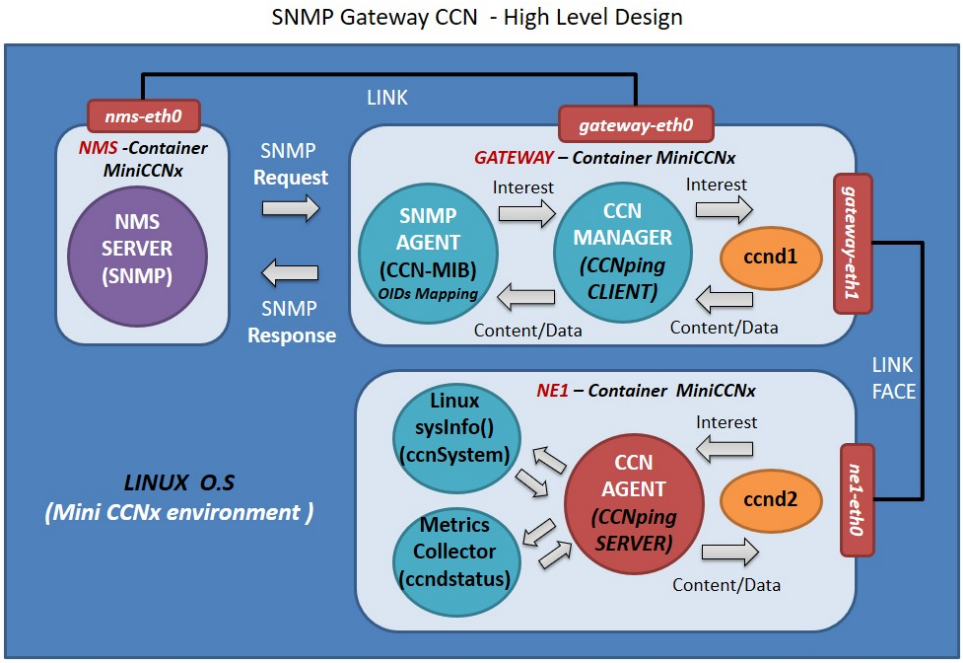

Figura 4.3: Visão detalhada da ferramenta SNMP Gateway CCN.

4.3 Uso da ferramenta SNMP Gateway CCN no ambiente MiniCCNx

Para iniciar a ferramenta SNMP Gateway CCN no ambiente MiniCCNx é necessário seguir alguns passos, descritos a seguir:

1. **Iniciando o Ambiente gráfico MiniCCNx**
   O ambiente gráfico MiniCCNx é executado através do comando abaixo no terminal da VM (*Virtual Machine*) do Linux O.S.

   ```
   # sudo miniccnxedit
   ```

   O comando fará com que a interface gráfica do MiniCCNx seja aberta, como mostra a Figura 4.4.

Figura 4.4: Interface gráfica MiniccnxEdit, para manipulação do ambiente MiniCCNx.

2. **Abrindo a topologia de referência**
   Para fins de experimentos, uma topologia de referência foi previamente criada e configurada com todas as opções necessárias para o funcionamento no ambiente, para abrir basta clicar em *File->Open* e selecionar o arquivo de configuração desejado. A topologia será detalhada mais a frente.

   Abrindo a topologia de referência através da interface gráfica MiniccnxEdit, como mostra a Figura 4.5.

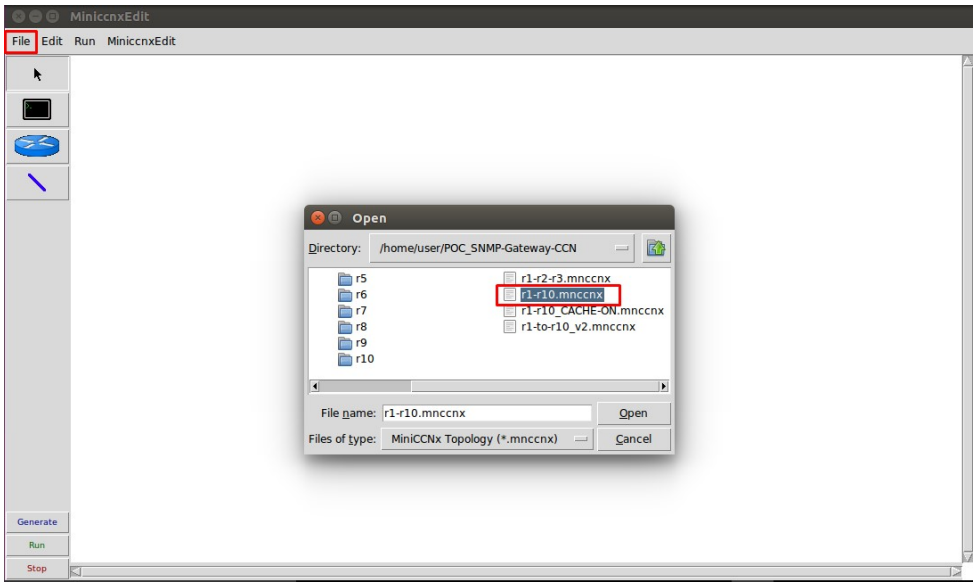

Figura 4.5: Abrindo a topologia de referência através de um arquivo pronto.

3. **Iniciando a topologia de referência**
   Para iniciar a topologia carregada, basta clicar no botão *Run*, posicionado no canto inferior esquerdo do menu.

Iniciando a topologia de referência através da interface gráfica MiniccnxEdit, como mostra a Figura 4.6.

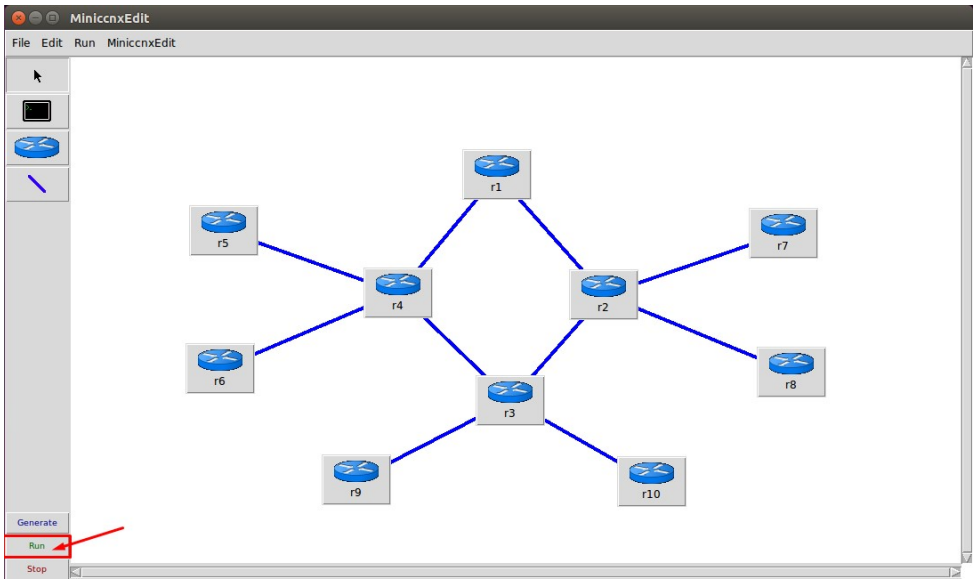

Figura 4.6: Iniciando a topologia de referência através da interface gráfica MiniccnxEdit.

Uma chamada para inicialização do Agente CCN foi introduzida na classe de CCNHost do projeto MiniCCNx, deste modo cada container CCN inicia automaticamente uma instância do Agente CCN durante o processo de inicialização da topologia. Como mostra a Figura 4.7.

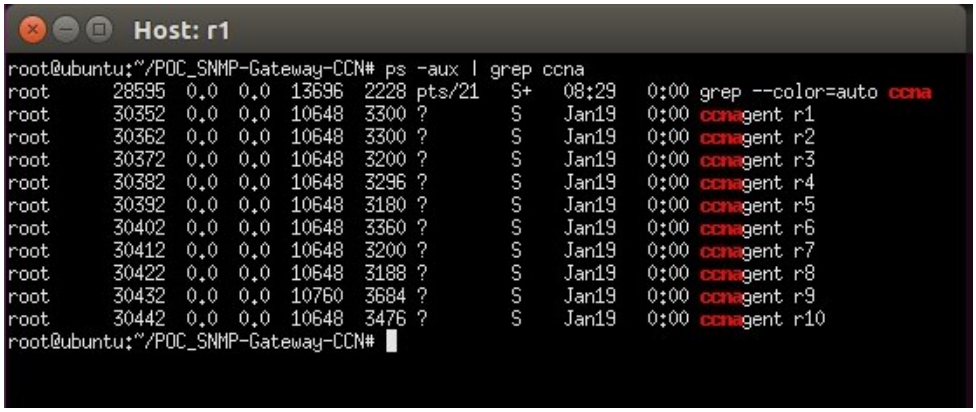

Figura 4.7: Agente CCN inicializado em cada host da topologia.

4. **Iniciando o Agente SNMP no elemento de rede gateway**

O Agente SNMP deve ser iniciado através do script ***snmp-gateway-ccn_start*** a partir de um elemento de rede Gateway. Como gateway para a rede CCN, selecionamos o elemento denominado ***r1*** da topologia de referência. Após início da topologia, é possível observar que os botões posicionados no canto superior esquerdo do menu, passam a ficar desabilitados (cor mais clara), deste modo, basta clicar com o botão

direito do mouse sobre o elemento gateway escolhido para acesso ao terminal e executar o comando abaixo na CLI do container.

```
# ./snmp-gateway-ccn_start
```

Conteúdo do script:

```
##################################
#!  /bin/sh
snmpd -Le -f -c /home/user/.snmp/snmpd.conf
##################################
```

Conteúdo do arquivo de configuração do agente SNMP *snmpd.conf*:

```
##################################
```

`rwuser user123456` (credenciais de escrita para acesso ao agente SNMPv3)
`rouser user123456` (credenciais de leitura para acesso ao agente SNMPv3)
`rwcommunity private` (credenciais de escrita para acesso ao agente SNMPv1 e SNMPv2)
`rocommunity public` (credenciais de leitura para acesso ao agente SNMPv1 e SNMPv2)
`injectHandler debug ccnMIB` (para fins de debug)

```
##################################
```

O comando invocará o Agente SNMP (daemon snmpd) com todos parâmetros necessários para funcionamento no gateway, como mostra a Figura 4.8.

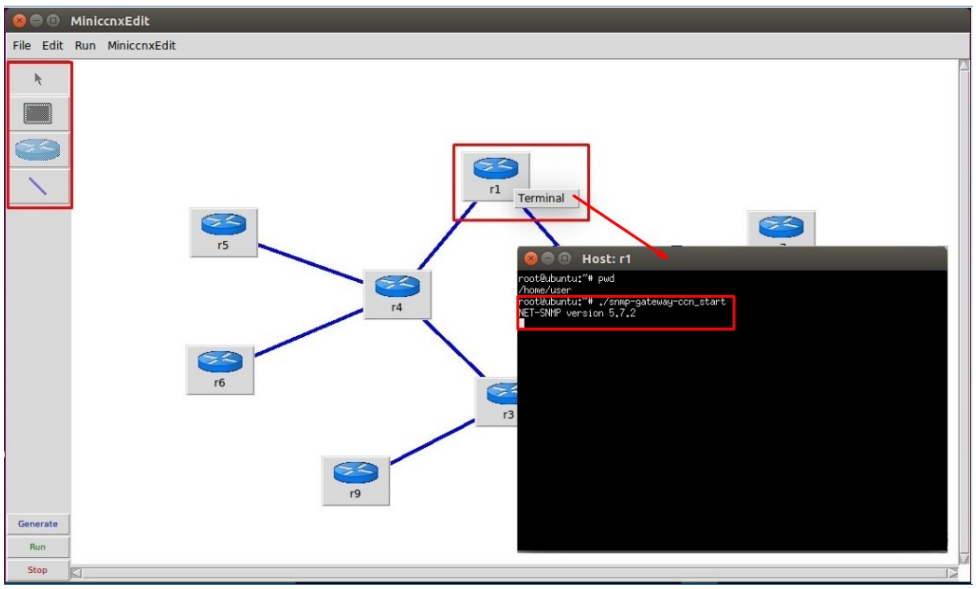

Figura 4.8: Iniciando o Agente SNMP a partir do terminal do elemento gateway.

5. **Iniciando o MIB Browser denominado SnmpB, front-end gráfico para realizar consultas à MIB CCN**

   O MIB Browser deve ser iniciado a partir de um elemento de rede Gateway, neste momento este mesmo elemento passa a funcionar também como um Servidor NMS, capaz de gerar requisições SNMP para o Agente SNMP executado no mesmo host no endereço de *localhost* porta 161. Para iniciar o MIB Browser, basta acessar um novo terminal do gateway e executar o comando abaixo:

   ```
   # sudo snmpb
   ```

   A Figura 4.9 exibe a janela do MIB Browser SnmpB.

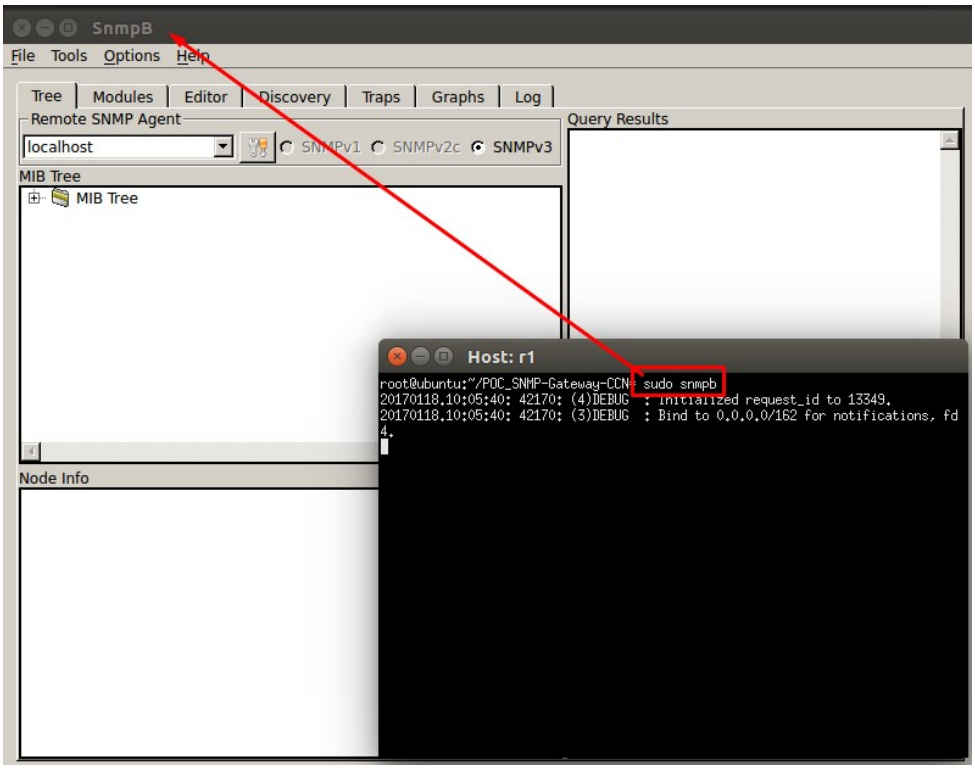

Figura 4.9: Iniciando o MIB Browser SnmpB a partir do terminal do elemento gateway.

6. **Configurando o MIB Browser e carregando a MIB CCN**

   O MIB Browser deve ser configurado com as credenciais necessárias para comunicação com o Agente SNMP executado no Gateway e deve carregar a MIB CCN para permitir acesso aos objetos sob os grupos *ccnSystem* e *ccndStatus*. Como destacado no capítulo anterior, o Agente SNMP tem suporte apenas à versão 3 do protocolo SNMP, conforme necessidade do campo *contextName* para identificação do elemento de rede CCN. Porém, as versões 1 e 2 também poderiam ser utilizadas pelo agente para responder consultas a objetos locais que representam informações do próprio gateway, caso compilado com a MIB-2 padrão ou qualquer outra MIB privada.

Agent Profiles: acesse *Options->Agent Profiles*.
Name: localhost (para acesso ao agente no próprio gateway)
Agent/Address Name: 127.0.0.1 (endereço de localhost do próprio gateway)
Retries: 1 (número de tentativas de consultas)
Timeout: (tempo de espera em segundos para abortar a operação)
Suported SNMP Version: SNMPv3 (para uso do protocolo SNMP versão 3)

Campos previamente preenchidos, como mostra a Figura 4.10.

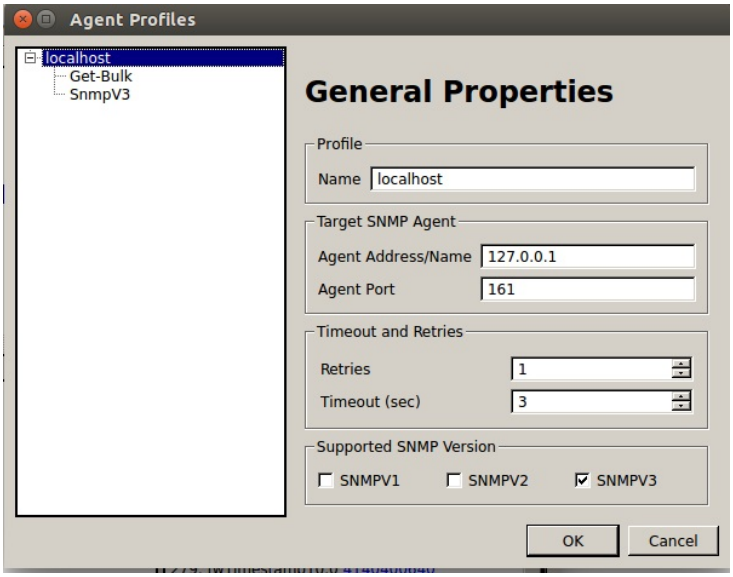

Figura 4.10: Configuração do parâmetro *Agent Profiles*.

Get-Bulk: acesse *Options->Agent Profiles->Get-Bulk*.
Non repeaters: 0 (número de repetições da operação)
Max repetitions: 7 (número máximo de ODIs consultados de cada vez)

Campos previamente preenchidos, como mostra a Figura 4.11.

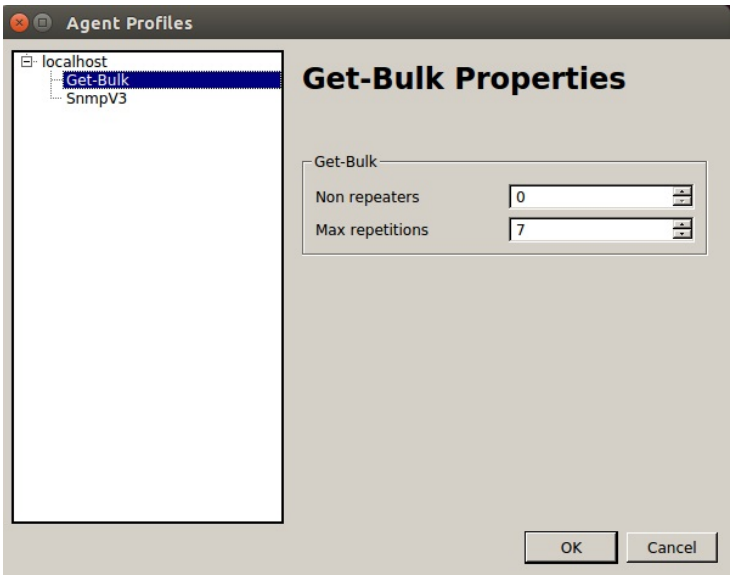

Figura 4.11: Configuração do parâmetro *Get-Bulk*.

SnmpV3: acesse *Options->Agent Profiles->SnmpV3*.

- SNMPv3 User (USM)
Security Name: user (nome de usuário)
Security Level: authNoPriv (com autenticação de usuário e senha, sem criptografia de mensagens)
- SNMPv3 context
Context Name: r + número do nó (nome do elemento de rede que deseja consultar)
Context Engine ID: (manter o campo em branco)

Campos previamente preenchidos, como mostra a Figura 4.12.

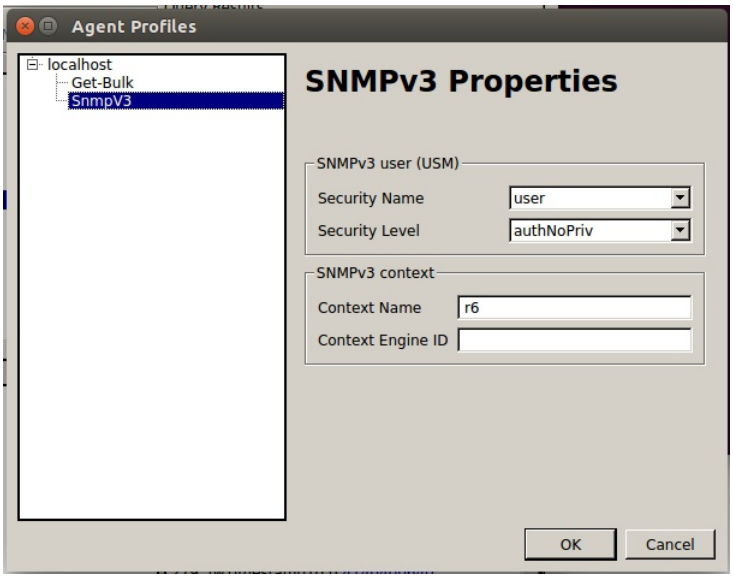

Figura 4.12: Configuração do parâmetro *SnmpV3*.

USM Profiles: acesse *Options->Manage SNMPv3 USM Profiles.*
- User
Security User Name: user (nome de usuário)
- Security
Authentication Protocol: MD5 (algoritmo para criptografar a autenticação do usuário)
Authentication Password: user (senha do usuário)
Privacy Protocol: none (criptografia de mensagens, desabilitada)
Privacy Password: (manter em branco)

Campos previamente preenchidos, como mostra a Figura 4.13.

Figura 4.13: Configuração de parâmetros de segurança do protocolo *SNMPv3*.

Modules: acesse *Options->Preferences->Modules.*
Clique no botão *Add* para adicionar o caminho da MIB CCN.

A Figura 4.14 apresenta a janela de configuração.

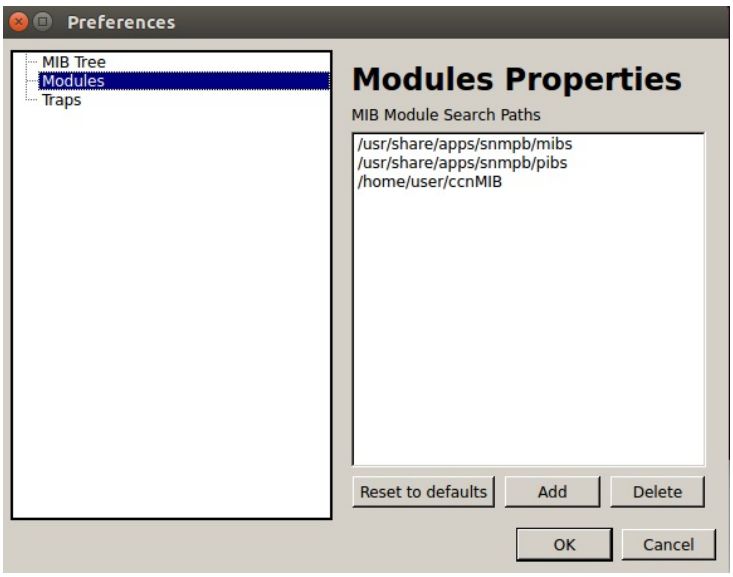

Figura 4.14: Configuração do caminho da MIB CCN.

Na janela principal acesse a aba *Modules* e localize a MIB CCN na janela *loaded MIB Modules*.
A Figura 4.15 a MIB CCN entre os módulos carregados no mib browser.

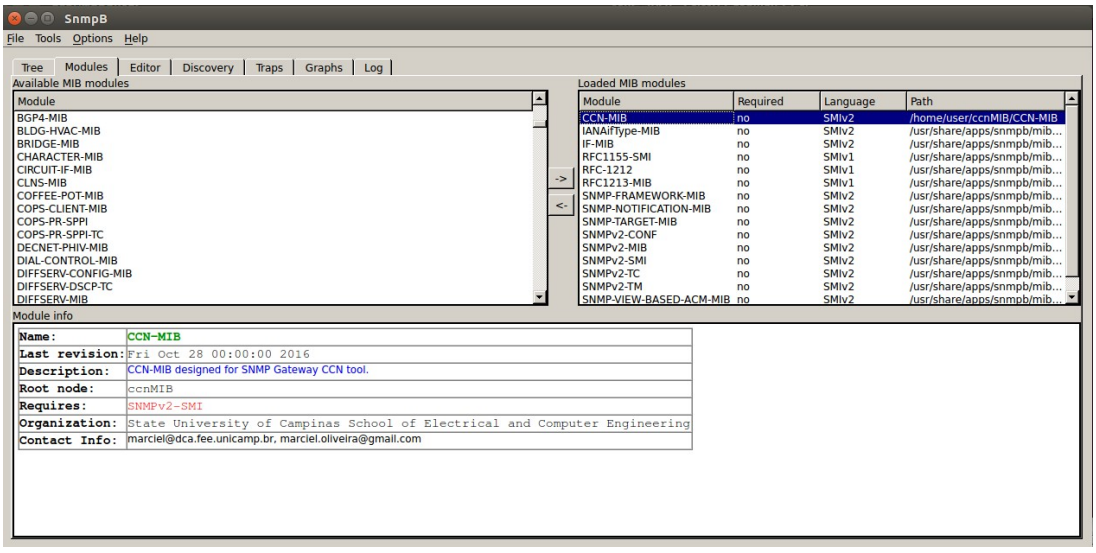

Figura 4.15: MIB CCN carregada com sucesso no mib browser SnmpB.

7. **Localizando a MIB CCN no MIB Browser SnmpB**

Com o MIB Browser iniciado, na janela MIB Tree percorra a hierarquia *MIB Tree->iso->org->dod->internet->mgmt->mib2->ccnMIB* para localizar os grupos *ccnSystem* e *ccndStatus* da MIB CCN.

A Figura 4.16 mostra a posição da MIB CCN sob o ramo da MIB-2 na hierarquia.

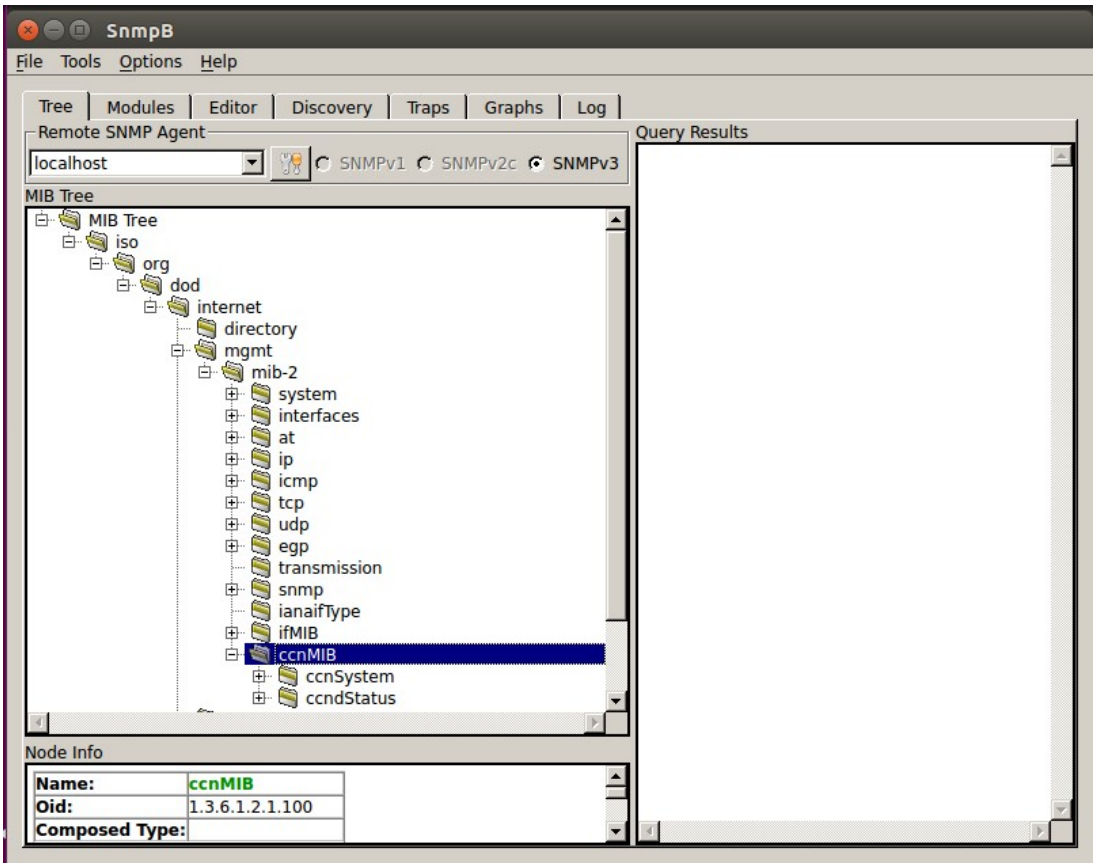

Figura 4.16: Seleção da MIB CCN através do MIB Browser SnmpB.

8. **Ferramenta SNMP Gateway CCN incializada**
   Por fim temos a ferramenta SNMP Gateway CCN em execução e pronta para ser utilizada.

   A Figura 4.17 mostra o ambiente totalmente funcional para início dos experimentos.

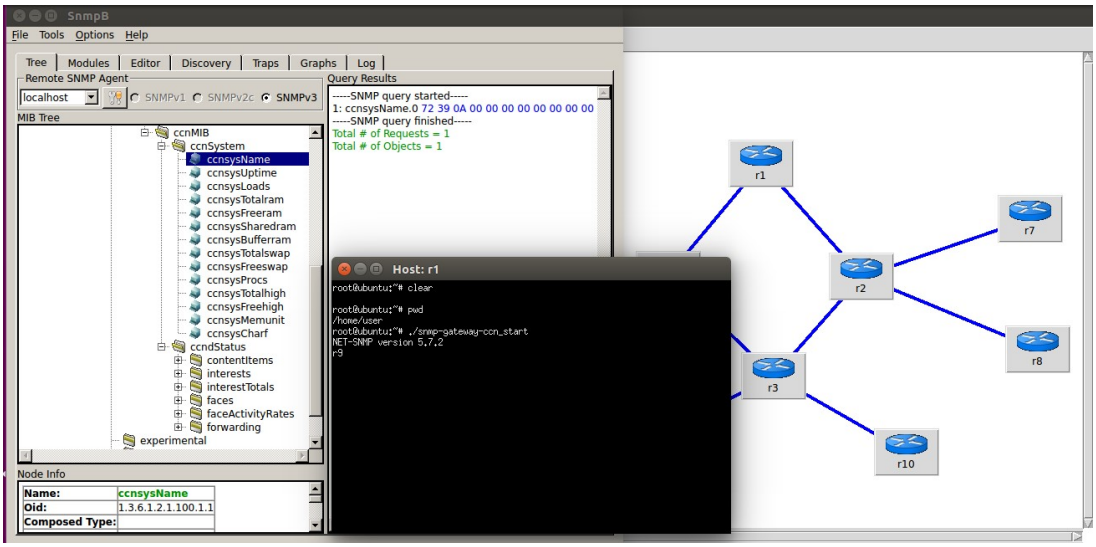

Figura 4.17: Ambiente pronto para utilização da ferramenta.

Capítulo 5

Metodologia, experimentos e resultados

5.1 Metodologia

A metodologia tem como objetivo descrever os critérios mínimos para avaliação experimental, coleta e análise de resultados da ferramenta SNMP Gateway CCN, com maior foco na validação funcional da ***operação GET*** do protocolo SNMP.

Pontos verificados: teste funcional da *operação GET*, avaliação da *operação WALK* quanto ao consumo de banda e comportamento em ambientes *com e sem cache habilitado* nos elementos de rede CCN.

5.1.1 Ambiente e versionamento

Para validação da proposta, o ambiente MiniCCNx foi pré-configurado juntamente com um conjunto de aplicações necessárias para o funcionamento da ferramenta SNMP Gateway CCN, conforme lista descrita abaixo.

Versões do ambiente:
- Linux Ubuntu 15.04 64bits (Vivid Vervet)
- InfluxDB v0.9.2
- Influxdb-python (0.1.12-1)
- Mininet 2.2.0b1
- CCNx v0.8.2
- Net-SNMP v5.7.2
- SnmpB v0.8
- Wireshark v1.6.2

5.1.2 Medidas

Para a verificar a troca de mensagens SNMP e CCN, adotamos o analisador de protocolos de rede *Wireshark* juntamente com um plugin[8] específico para suporte à análise de mensagens do protocolo CCNx. Com o auxílio do analisador é possível extrair algumas informações e métricas relevantes em cenário de gerência de redes, como por exemplo: o fluxo das mensagens enviadas (upstream) e recebidas (downstream), tempo de resposta, pico e consumo médio de banda para um conjunto de objetos consultados.

5.2 Experimentos e resultados

Os experimentos funcionais tem como objetivo principal validar a prova de conceito, que consiste em verificar que a ferramenta SNMP Gateway CCN cumpre o papel proposto de gerência SNMP em redes CCN.

5.2.1 Topologia de referência

Como topologia de referência para validação da prova de conceito, um cenário com 10 elementos de rede CCN foi configurado no ambiente MiniCCNx com a ajuda da ferramenta MiniccnxEdit. A topologia foi definida em anel com alguns elementos ramificados no formato linear. Por fim todas as configurações da topologia foram armazenadas no arquivo ***r1-r10.mnccnx*** para facilitar a restauração do ambiente quando necessário. Como nomeação de cada elemento, adotamos o prefixo ***r*** que representa a inicial da palavra ***router***, seguido de um número sequencial de 1 até 10.

Reforçando que o nome do elemento de rede é uma informação importante para identificar e localizar o nó que será monitorado na rede CCN, uma vez que o nome do nó mais o nome do conteúdo (prefixo mais longo) são utilizados no processo de encaminhamento das mensagens.

O elemento ***r1*** da topologia foi adotado para funcionar como *Servidor NMS* (gerar mensagens SNMP) e *Gateway de rede CCN* (tradutor das mensagens SNMP para CCN). Cada elemento de rede da topologia possui pelo menos uma interface de rede para comunicação com um elemento vizinho diretamente conectado. A seguir a Figura 5.1 apresenta a topologia de referência e a identificação das interfaces de rede de cada elemento.

[8] O plugin de *Wireshark* para suporte ao protocolo CCNx, foi compilado e instalado conforme instruções do projeto no link. https://github.com/ProjectCCNx/ccnx/blob/master/apps/wireshark/README-wireshark-1.6.txt

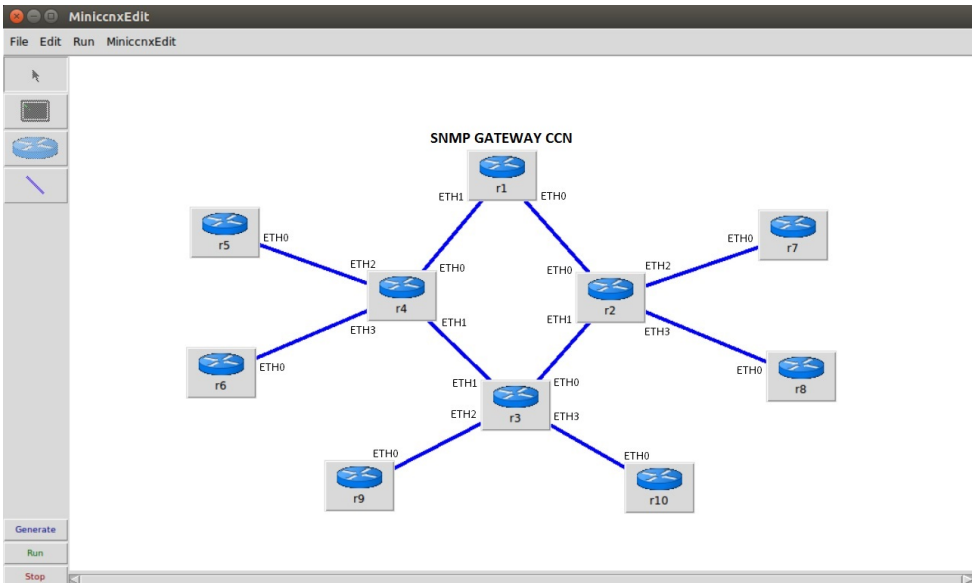

Figura 5.1: Topologia de referência para execução dos experimentos.

5.2.2 Teste funcional: *operação SNMP GET*

O teste funcional tem como objetivo mostrar que a ferramenta SNMP Gateway CCN permite o gerenciamento de elementos de rede CCN nativos a partir de sistemas de gerências legadas, baseadas no protocolo SNMP. Para validação da prova de conceito uma mensagem *SNMP GET Request* será gerada com uso do *MIB Browser SnmpB* que fará o papel de um sistema NMS. O ambiente MiniCCNx usará de broadcast nativo para encaminhamento das mensagens, uma vez que não é objetivo habilitar protocolo de roteamento para a execução dos experimentos.

Tendo como base a topologia de referência, o elemento de rede CCN ***r1*** foi escolhido como gateway de rede. O objetivo inicial é consultar o valor de um objeto qualquer em um elemento de rede distante do gateway, deste modo, uma mensagem de consulta *SNMP GET Request* deve partir do elemento ***r1*** e alcançar o elemento alvo, uma mensagem *SNMP GET Response* deve retornar ao gateway em resposta à requisição. A interface de rede ***lo*** (loopback/IP 127.0.0.1) será monitorada no elemento de rede gateway para captura das mensagens SNMP. As interfaces ***eth0*** e ***eth1*** serão monitoradas para captura das mensagens CCN. Abaixo seguem os passos e o comportamento esperado como resultado para o teste funcional da operação SNMP GET:

Passos:

- O *MIB Browser SnmpB* deve ser configurado para preencher o campo *contextName* do protocolo SNMPv3 conforme o nome do elemento de rede que deseja consultar, como primeiro experimento exercitamos a consulta do elemento ***r6*** da topologia.
- Executar o *Wireshark* para monitorar as interfaces de rede ***lo***, ***eth0*** e ***eth1*** do elemento gateway ***r1***.
- A partir do *Mib Browser*, iniciar uma consulta *SNMP GET Request* para uma OID da MIB CCN.

- Na captura realizada na interface *lo* do gateway r1, deve ser possível verificar a existência de uma mensagem de requisição *SNMP GET* na versão 3, o campo *contextName* preenchido com o label definido no *Mib Browser* e na sequência uma mensagem de resposta *SNMP GET Response* com o valor solicitado.
- Nas capturas realizadas nas interfaces *eth0* e *eth1* do gateway, deve ser possível verificar a existência de uma mensagem CCN de requisição *Interest* e na sequência uma mensagem de resposta *Data* com o valor solicitado.

Como o ambiente não tem protocolo de roteamento habilitado, as mensagens CCN serão encaminhadas via *broadcast*, por este motivo, ambas as interfaces do gatetway *eth0* e *eth1* podem ser monitoradas pelo analisador de protocolo.

Neste primeiro experimento, optamos por capturar pacotes CCN apenas da interface *eth1* que na topologia apresenta o menor caminho (número de saltos) para alcançar o elemento alvo r6.

Resultado:
A partir do gateway de nome ***r1*** iniciou-se a consulta ***SNMP GET Resquest*** ao elemento de rede com nome ***r6***, foi possível observar que a mensagem SNMP GET é gerada com sucesso para o OID *1.3.6.1.2.1.100.2.1* (ccnSystem/ccnsysUpTime) com *contextName r6* e recebida na interface de rede*lo* (loopback) do gateway. Em seguida o Agente SNMP converte a mensagem SNMP GET para uma mensagem ***Interest*** (ccnx://r6/ccnSystem/ccnsysUpTime/id) e a encaminha para a rede, uma mensagem ***Data*** (ContentObject) retorna com o conteúdo solicitado ***r6*** em resposta à mensagem *Interest*, por fim, a mensagem *Data* é convertida de volta para uma mensagem ***SNMP GET Response*** e entregue ao Servidor NMS em resposta à consulta SNMP GET. As evidências podem ser observadas nas Figuras 5.2, 5.3, 5.4, 5.5 e 5.6.

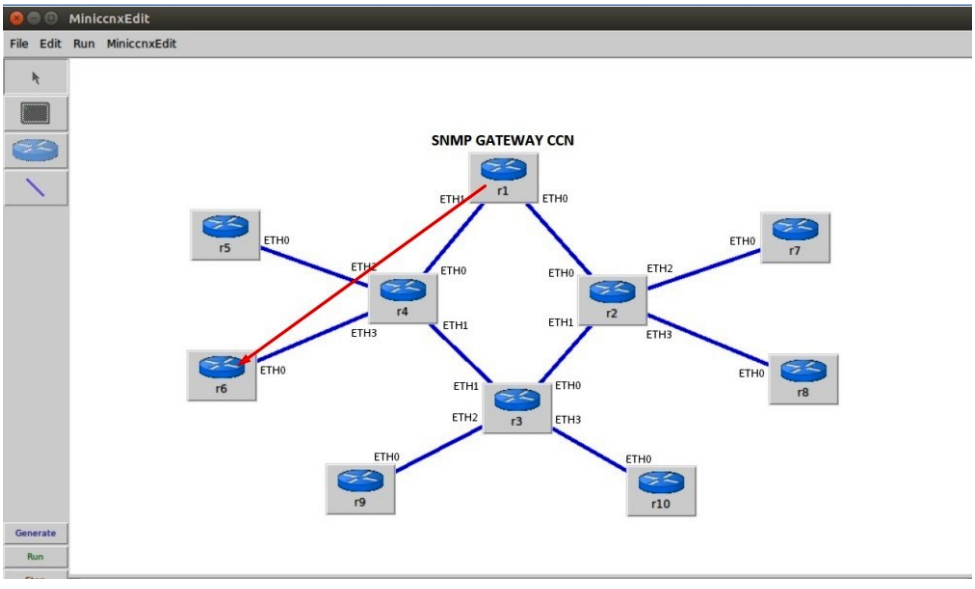

Figura 5.2: Gateway da rede *r1* e elemento de rede consultado *r6*.

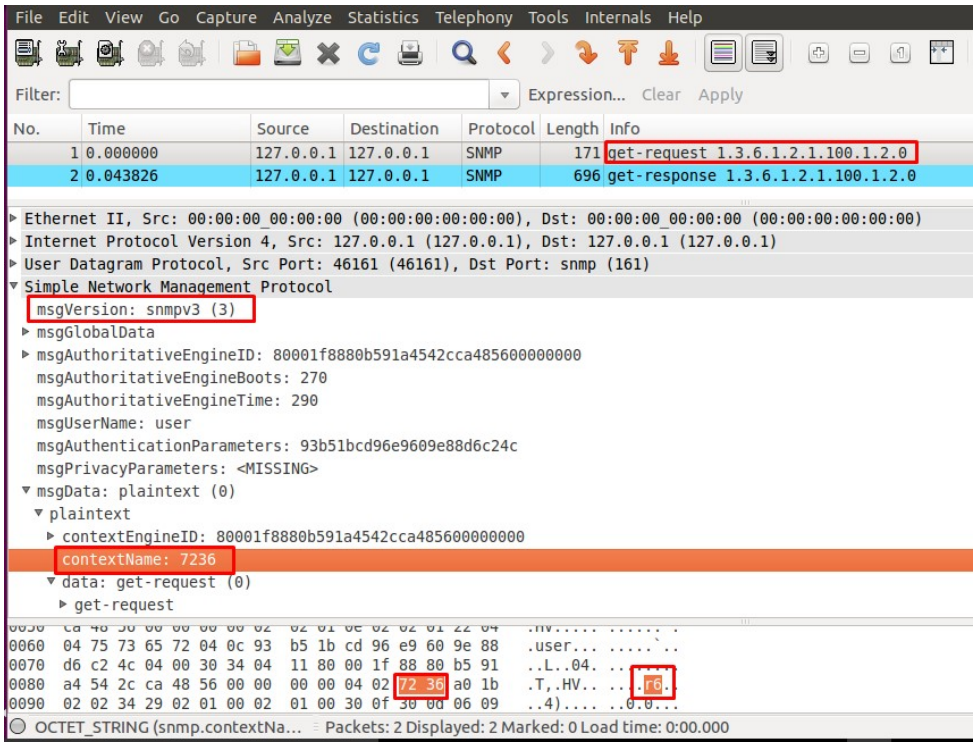

Figura 5.3: Captura da mensagem de consulta SNMP GET Request.

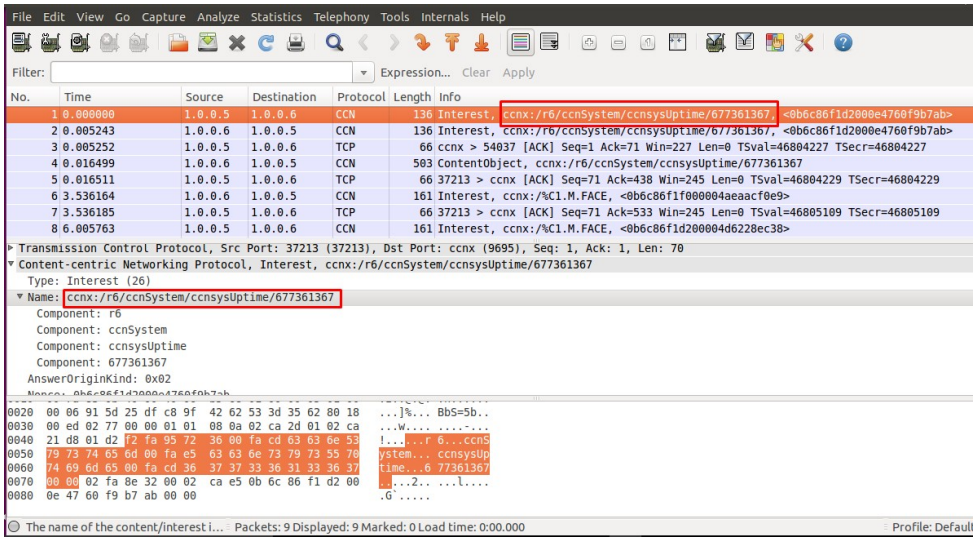

Figura 5.4: Captura da mensagem de consulta *Interest*.

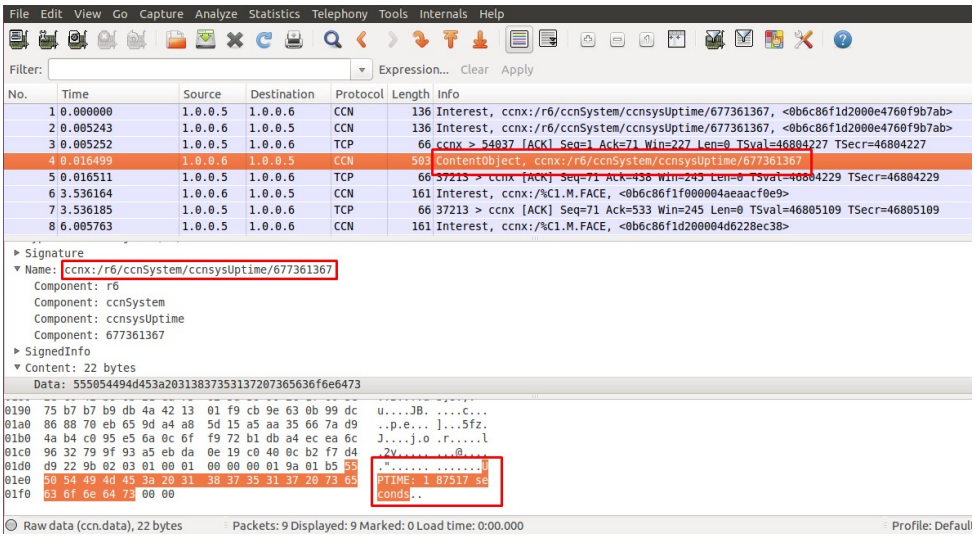

Figura 5.5: Captura da mensagem de resposta *ContentObject/Data*.

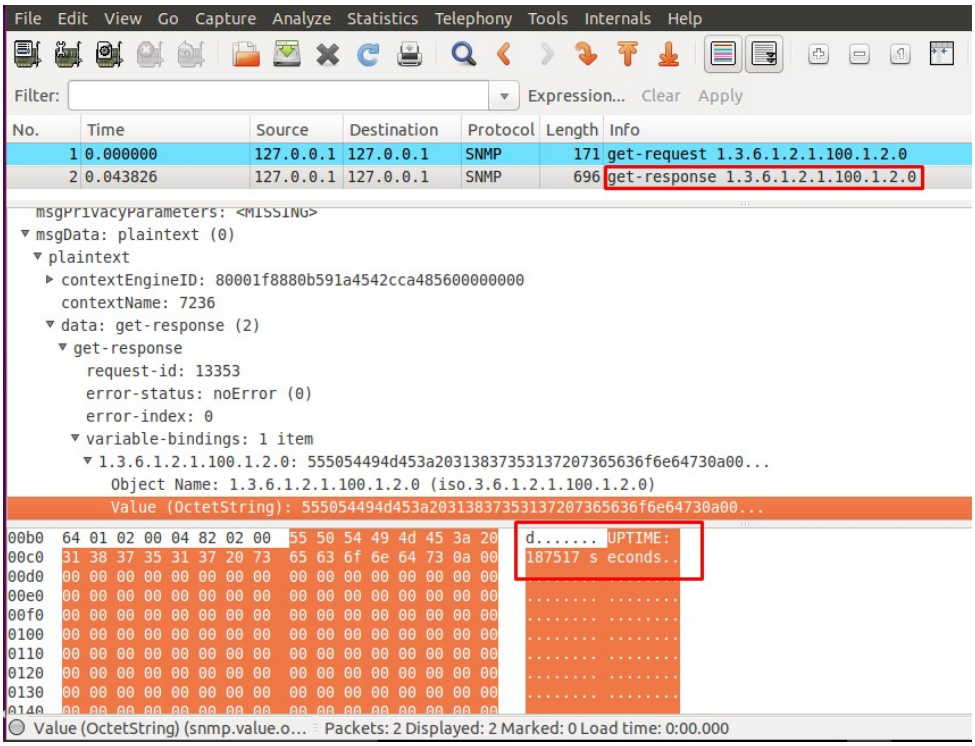

Figura 5.6: Captura da mensagem de resposta convertida para SNMP GET Response.

5.2.3 Teste de múltiplas consultas: *operação SNMP WALK*

A realização de múltiplas consultas a objetos em elementos de rede CCN, que são executadas em passos sequenciais via operação SNMP WALK, nos permitirá examinar o comportamento do sistema em um ambiente CCN nativo. Deve-se experimentar requisições em um elemento mais próximo e outro elemento mais distante, levando em consideração o número de saltos. Tendo como base a topologia de referência, o elemento de rede CCN ***r1*** continua como gateway de rede, e os elementos ***r6*** e ***r9*** serão consultados neste experimento.

Um dos principais objetivos deste experimento é percorrer todos os 280 objetos da MIB CCN e verificar o consumo de banda nas interfaces de rede ***eth0*** e ***eth1*** do gateway. Mensagens *Interest* de requisição devem partir do elemento ***r1*** e alcançar os elementos alvo, uma mensagem de resposta ***Data*** deve retornar ao gateway em resposta à cada requisição. As interfaces ***eth0*** e ***eth1*** serão monitoradas para captura das mensagens CCN. Abaixo seguem os passos e o comportamento esperado como resultado para o teste funcional da operação SNMP WALK:

Passos:
- O *Mib Browser SnmpB* deve ser configurado com o *contextName* do elemento de rede que deseja consultar, no exemplo exercitamos a consulta dos elementos ***r6*** e ***r9*** da topologia.
- Executar o *Wireshark* para monitorar apenas as interfaces de rede ***eth0*** e ***eth1*** do elemento gateway ***r1***.
- A partir do *Mib Browser*, iniciar uma consulta *SNMP WALK Request* a partir do OID principal da MIB CCN.
- Na captura realizada nas interfaces *eth0* e *eth1* do gateway r1, deve ser possível verificar a existência de uma mensagem de requisição *Interest* na sequência uma mensagem de resposta *Data* para cada valor solicitado.
- Nas capturas realizadas nas interfaces *eth0* e *eth1* do gateway, deve ser possível verificar a existência de uma mensagem CCN de requisição *Interest* e na sequência uma mensagem de resposta *Data* para cada valor solicitado.

Os pacotes capturados nas interfaces permitem analisar o consumo de banda durante o processo de consulta e o tempo decorrido em segundos.

Resultado:
A partir do gateway de nome ***r1*** iniciou-se a consulta ***SNMP WALK Resquest*** ao elemento de rede com nome ***r6***, foi possível observar que a operação SNMP WALK gera uma série de mensagens *GET Bulk Request* com um conjunto de 5 objetos em cada solicitação com *contextName r6* a partir do OID principal da MIB CCN (*1.3.6.1.2.1.100*/ccnMib). Em seguida o Agente SNMP converte as mensagens da operação SNMP WALK para mensagens ***Interest*** e as encaminha para a rede CCN. As mensagem ***Data*** (ContentObject) retornam com os conteúdos solicitados em resposta a cada mensagem *Interest*, por fim,

as mensagens *Data* são convertidas de volta para mensagens ***SNMP GET Response*** e entregues ao Servidor NMS em resposta às consultas da operação SNMP WALK. Os passos foram repetidos com sucesso para consulta do elemento *r9*. A Figura 5.7 indica os elementos de rede CCN consultados e as Figuras 5.8 e 5.9 apresentam a ferramenta em funcionamento durante o processo de consulta dos elementos *r6* e *r9* respectivamente.

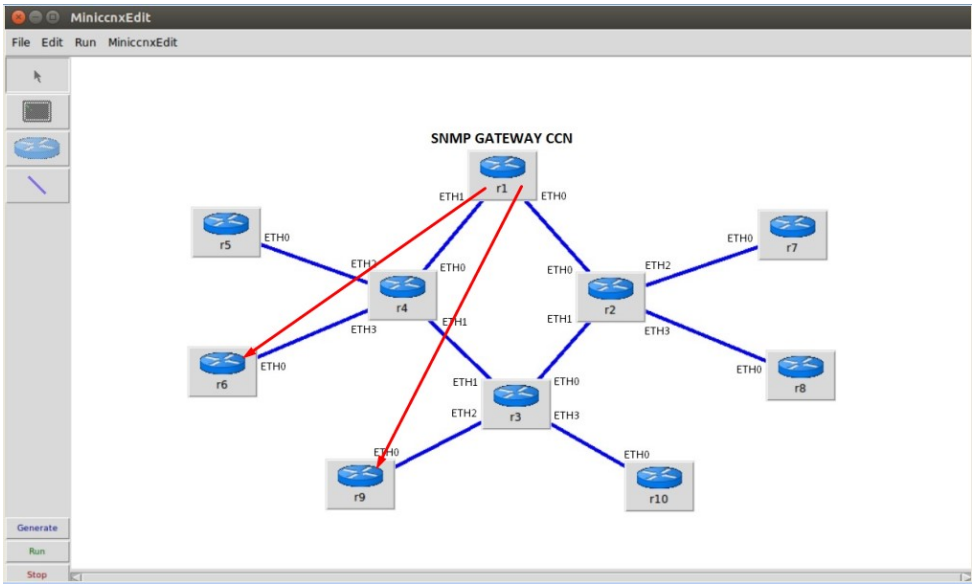

Figura 5.7: Gateway da rede *r1* e elementos de rede consultados, *r6* e *r9*.

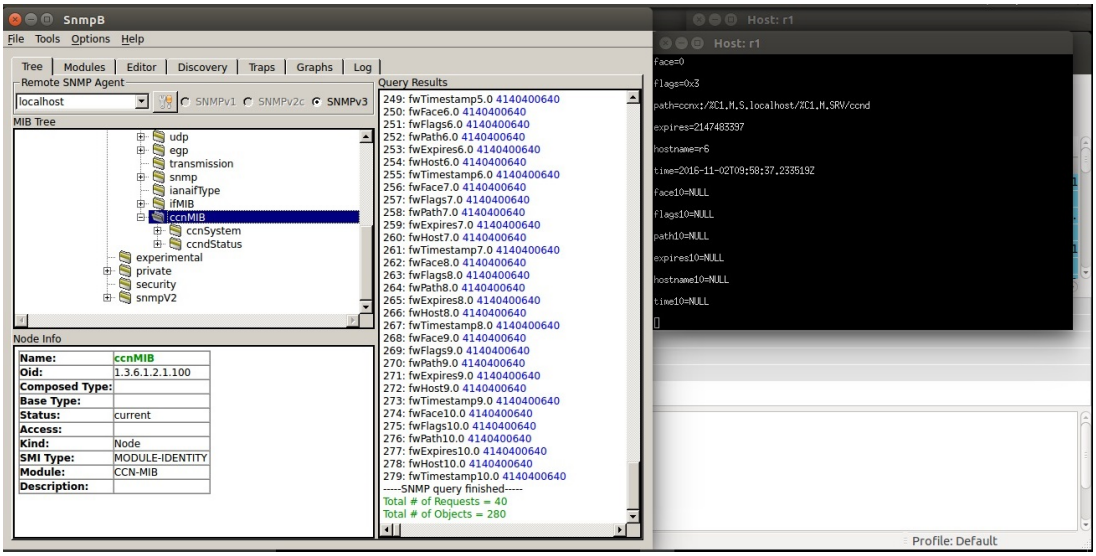

Figura 5.8: Ambiente durante consulta de todos os objetos do elemento de rede r6.

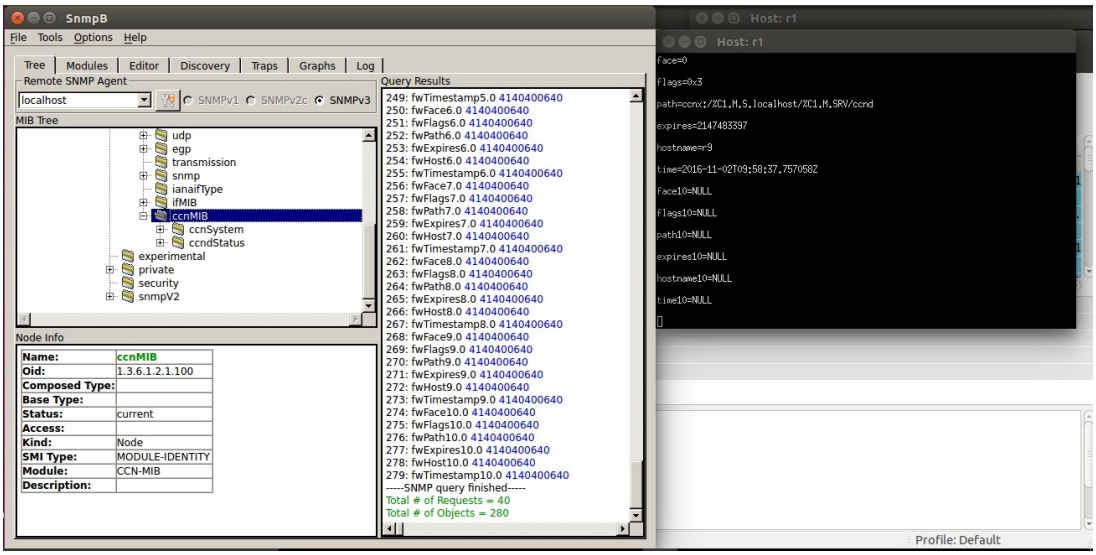

Figura 5.9: Ambiente durante consulta de todos os objetos do elemento de rede r9.

5.2.4 Consumo de banda e coerência

O uso da operação SNMP WALK, nos permite consultar a MIB CCN por completo e gerar tráfego constante, necessário para a coleta de medidas relacionadas com o consumo de banda. Foram coletadas três amostras do tráfego médio apresentado pela operação SNMP WALK, o experimento foi realizado em dois momentos, no primeiro momento o ambiente MiniCCNx foi configurado com suporte a *cache* e no segundo momento foi configurado sem suporte a *cache*, para cada conjunto de 3 amostras uma nova média foi calculada como resultado da análise final. As amostras foram coletadas durante consultas realizadas em ambos os elementos de rede da topologia de referência, r6 e r9, devemos observar que o elemento gateway r1 apresenta 2 caminhos distintos para alcançar os elementos gerenciados. A partir do elemento gateway, para alcançar o elemento r6 existem 4 saltos via interface eth0 e 2 saltos via interface eth1. Já para alcançar o elemento r9 existem exatamente 2 saltos em ambas as direções, via interface eth0 e interface eth1.

Com todos os dados coletados e compilados, é possível observar e destacar alguns resultados, entre eles, o tempo total de consulta e resposta durante a operação SNMP WALK, que apresentou tempo médio de 11 e 12 segundos para os elementos r6 e r9 respectivamente. As consultas realizas ao elemento de rede r6 apresentaram maior consumo médio de banda e pico máximo na interface eth1 do gateway, uma vez que se trata da interface de rede com menor caminho (2 saltos) entre o gateway e o elemento gerenciado r6, portanto justamente o caminho mais utilizado entre os elementos envolvidos na operação. As consultas realizadas ao elemento r9 apresentam consumo de banda equilibrado entre as interfaces eth0 e eth1 do gateway. Deste modo, é possível concluir que de forma geral o consumo de banda monitorado nas interfaces eth0 e eth1 do gateway se mostraram coerentes, tendo em vista a posição de cada elemento na topologia. As Tabelas 5.1 e 5.2 evidenciam os dados relatados.

Tabela 5.1: Consultas aos elementos r6 e r9, cenário sem cache.

Cache-Off	**SNMP WALK - r6**		**SNMPWALK - r9**	
(Kbps)	gw-eth0	gw-eth1	gw-eth0	gw-eth1
avg	38	88	71	72
peak	90	180	160	150
Elapsed time (sec)	11		12	

Tabela 5.2: Consultas aos elementos r6 e r9, cenário com cache.

Cache-On	**SNMP WALK - r6**		**SNMPWALK - r9**	
(Kbps)	gw-eth0	gw-eth1	gw-eth0	gw-eth1
avg	34	82	71	72
peak	90	185	140	145
Elapsed time (sec)	11		12	

Para melhor visualização e efeito de comparação do consumo de banda dos experimentos de gerência de redes CCN nos cenário com e sem cache, a Figura 5.10 apresenta apenas uma amostragem de cada consulta realizada, neste caso é possível observar que ambos os cenários apresentam consumo de banda equivalente sem grandes divergências.

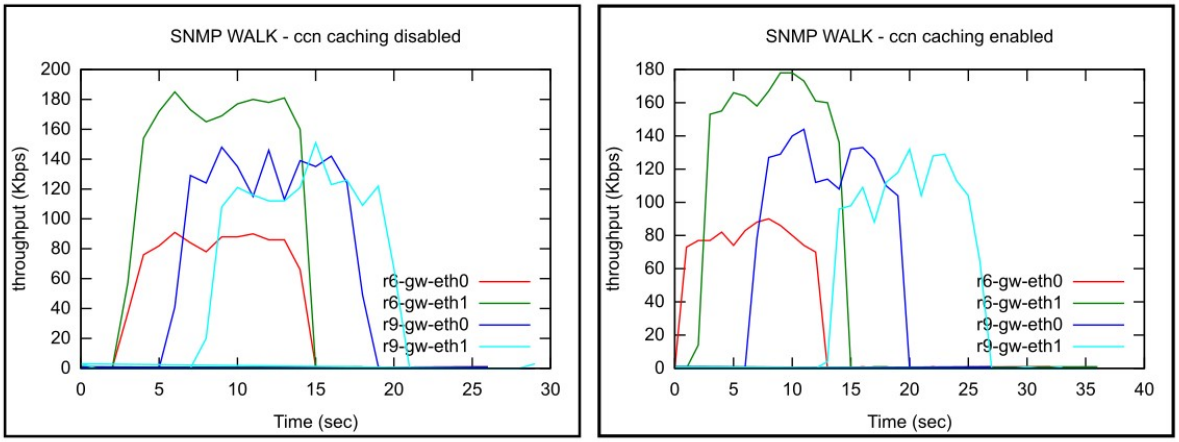

Figura 5.10: Consumo de banda SNMP WALK r6 e r9, cenário com e sem cache habilitado.

Capítulo 6

Conclusões, Trabalhos Futuros e Contribuições

6.1 Conclusões

As grandes redes com centenas ou milhares de elementos exigem soluções e arquiteturas robustas para monitoramento, operação, manutenção e configuração de seus dispositivos e serviços. Dentre alguns padrões utilizados, adotamos o protocolo SNMP como base para experimentar gerência de redes orientadas a conteúdo (ROCs), por se tratar de um protocolo largamente utilizado, tendo em vista a carência de soluções adequadas e eficientes. Dentre algumas arquiteturas de gerência e configuração, destacamos o protocolo SNMP, que surgiu como uma proposta melhor estruturada com base na relação gerente e agente que apresenta um conjunto de objetos representados por recursos gerenciados em um determinado elemento. A identificação de elementos e conteúdos com base em uma estrutura de nomes hierárquicos, permitiu a elaboração da arquitetura NONM que disponibiliza um mecanismo de tradução SCNT utilizado pela ferramente SNMP Gateway CCN. Esse mecanismo torna possível a gerência de redes orientadas a conteúdo a partir de um sistema tradicional que utiliza o protocolo SNMP. A ferramenta surge como primeiro passo para a criação de novas arquiteturas rumo a gerência de redes orientadas a conteúdo.

6.2 Trabalhos futuros

A ferramenta SNMP Gateway CCN atualmente se limita a algumas funcionalidades, dentre elas, as operações básicas do protocolo SNMP (*GET*, *GET-NEXT*, *GET-BULK* e *WALK*) e apresenta uma topologia de referência com 10 elementos de rede, que tem o objetivo principal experimentar a gerência de redes CCN e demonstração da prova de conceito. Tais limitações abrem espaço para a evolução funcional da ferramenta, com destaque para; a operação *SET*, responsável por alterar a configuração de elementos de rede e a operação de notificação de eventos *TRAP*, ambas mapeadas neste trabalho, porém não presentes na ferramenta. A implementação da funcionalidade de notificação de eventos permitiria por exemplo analisar o comportamento das mensagens em um ambiente CCN congestionado tendo em vista a necessidade da garantia de entrega desse tipo de mensagem devido a alta criticidade em momentos de falhas, como por exemplo, o rompimento de um enlace de rede ou limiar de temperatura acima do esperado. A funcionalidade de descoberta de topologia não menos importante, seria uma facilitador quanto a identificação dos nomes dos elementos de rede em um cenário com centenas ou milhares de elementos.

Novas pesquisas podem ser conduzidas para a criação de um *Gerente CCN* nativo além do *Agente CCN* já proposto neste trabalho, que permitiria por exemplo desacoplar o ambiente legado (NMS SNMP/IP) do ambiente CCN e experimentar a relação *gerente/agente* em um cenário baseado apenas no modelo CCN. A especificação de uma MIB exclusiva para tratar conteúdo, poderia ser mais um componente importante para experimentar com maior fidelidade o conceito de gerência de redes orientadas a conteúdo. Outros aspectos também poderiam ser explorados, como por exemplo, segurança e robustez, temas que contribuem para confiabilidade de um sistema de gerência completo.

6.3 Contribuições

Como destaques deste trabalho apresentamos uma proposta de arquitetura *NONM* (Name-Oriented Network Management) e o mecanismo de mapeamento *SCNT* (SNMP Content Network Translation) utilizados como base para o desenvolvimento da ferramenta SNMP Gateway CCN que permite o gerenciamento e monitoramento de nós de rede CCN nativas através de sistemas de gerência de redes SNMP legadas.
Destacamos também a modelagem da *MIB CCN* composta por novos ramos posicionados abaixo da MIB-2 e um *Agente CCN* nativo que permite consultar valores específicos de objetos em elementos de rede CCN.
O código fonte aberto da ferramenta SNMP Gateway CCN está disponível no link `https://github.com/marcieloliveira/snmp-gateway-ccn`.

Bibliografia

BRITO, G. M. de; VELLOSO, P. B.; MORAES, I. M. Redes orientadas a conteúdo: Um novo paradígma para a internet. 2012. Minicursos do XXX Simpósio Brasileiro de Redes de Computadores (SBRC 2012).

CABRAL, C. M. S.; ROTHENBERG, C. E.; MAGALHAES, M. F. Mini-ccnx: prototipagem rápida para redes orientadas a conteúdo baseadas em ccn. 2013. Salão de Ferramentas do XXXI Simpósio Brasileiro de Redes de Computadores (SBRC 2013).

CARZANIGA, A.; PAPALINI, M.; WOLF, A. L. Content-based publish/subscribe networking and information-centric networking. *ACM Computing Surveys (CSUR)*, 2011.

ELIAS, C. d. M. Levantamento de métricas em redes orientadas a conteúdo com ferramenta de emulação de redes. Projeto de Iniciação Científica, Faculdade de Engenharia Elétrica e de Computação - FEEC da Universidade Estadual de Campinas - UNICAMP. 2015.

EUGSTER, P. T.; FELBER, P. A.; GUERRAOUI, R.; KERMARREC, A.-M. The many faces of publish/subscribe. *ACM Computing Surveys (CSUR)*, 2003.

HOQUE, A.; AMIN, S. O.; ALYYAN, A.; ZHANG, B.; ZHANG, L.; WANG, L. Nlsr: Named-data link state routing protocol. *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking, ACM*, 2013.

JACOBSON, V.; SMETTERS, D. K.; THORNTON, J. D.; PLASS, M. F.; BRIGGS, N. H.; BRAYNARD, R. L. Networking named content. *Proceedings of the 5th international conference on Emerging networking experiments and technologies, ACM*, 2009.

JOKELA, P.; ZAHEMSZKY, A.; ROTHENBERG, C. E.; ARIANFAR, S.; NIKANDER, P. Lipsin: line speed publish/subscribe inter-networking. *ACM SIGCOMM Computer Communication Review*, 2009.

KANG, W.; SIM, B.; KIM, J.; PAIK, E.; LEE, Y. A network monitoring tool for ccn. *World Telecommunications Congress (WTC), IEEE*, 2012.

KOPONEN, T.; CHAWLA, M.; CHUN, B.-G.; ERMOLINSKIY, A.; KIM, K. H.; SHENKER, S.; STOICA, I. A data-oriented (and beyond) network architecture. *ACM SIGCOMM Computer Communication Review*, 2007.

KUTSCHER, D.; EUM, S.; PENTIKOUSIS, K.; PSARAS, I.; CORUJO, D.; SAUCEZ, D.; SCHMIDT, T.; WAEHLISCH, M. *Information-Centric Networking (ICN) Research Challenges*. [S.l.], 2016. 27-29 p. <https://www.rfc-editor.org/rfc/rfc7927.txt>. Disponível em: <https://www.rfc-editor.org/rfc/rfc7927.txt>.

MAURO, D.; SCHMIDT, K. *Essential SNMP*. [S.l.]: O'Reilly Media, Inc. USA., 2005.

NUNES, S.; DAVID, G. Uma arquitectura web para servicos web. *XATA 2005-XML: Aplicações e Tecnologias Associadas, Portugal*, 2013.

OLIVEIRA, M.; ROTHENBERG, C. E. Snmp proxy ccn: Uma proposta de arquitetura para gerência de redes orientadas a conteúdo interoperável com sistemas legados. Workshop de Redes P2P, Dinâmicas, Sociais e Orientadas a Conteúdo do XXXII Simpósio Brasileiro de Redes de Computadores (SBRC 2014), 2014.

OLIVEIRA, M.; ROTHENBERG, C. E. Snmp gateway ccn: Software de gerência de redes orientadas a conteúdo interoperável com sistemas legados. Salão de Ferramentas do XXXV Simpósio Brasileiro de Redes de Computadores (SBRC 2017), 2017.

ROSS, K. W.; KUROSE, J. F. *Redes de Computadores e a Internet - Uma abordagem top-down*. [S.l.]: Person Educartion do Brasil. Brasil., 2005.

SCHONWALDER, J.; BJORKLUND, M.; SHAFER, P. Network configuration management using netconf and yang. *IEEE Communications Magazine*, 2010.

SHANG, W.; DING, Q.; MARIANANTONI, A.; BURKE, J.; ZHANG, L. Securing building management systems using named data networking. *Network, IEEE*, 2014.

VIRGILLITO, A. *Publish/subscribe communication systems: from models to applications*. Tese (Doutorado) — Universita La Sapienza, 2003.

WALSH, L. *SNMP MIB Handbook - Essential Guide to MIB Development, Use, and Diagnosis*. [S.l.]: Wynham Press. USA., 2008.

WANG, L.; HOQUE, A. K. M. M.; YI, C.; ALYYAN, A.; ZHANG, B. Ospfn: An ospf based routing protocol for named data networking. *Technical Report NDN-0003*, 2012. Disponível em: <https://www.named-data.net/techreport/TR003-OSPFN.pdf>.

XYLOMENOS, G.; VERVERIDIS, C.; SIRIS, V.; FOTIOU, N.; TSILOPOULOS, C.; VASILAKOS, X.; KATSAROS, K.; POLYZOS, G. A survey of information-centric networking research. *IEEE*, 2009.

Apêndice A

Publicações

Este trabalho gerou 2 publicações:

1. Artigo no Wp2p+ 2014 - Workshop de Redes P2P, Dinâmicas, Sociais e Orientadas a Conteúdo, na XXXII edição do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), realizado em Maio de 2014 em Florianópolis-SC, com o título *SNMP Proxy CCN: Uma proposta de arquitetura para gerência de redes orientadas a conteúdo interoperável com sistemas legados* (OLIVEIRA; ROTHENBERG, 2014).

2. Artigo no Salão de Ferramentas, na XXXV edição do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), realizado em Maio de 2017 em Belém-PA, com o título *SNMP Gateway CCN: Software de gerência de redes orientadas a conteúdo interoperável com sistemas legados* (OLIVEIRA; ROTHENBERG, 2017).

Apêndice B

MIB CCN

Objetos da MIB CCN sob o ramo ccnSystem

A *Figura B.1* refelete objetos relativos ao estado e informações do Sistema de um elemento de rede CCN.

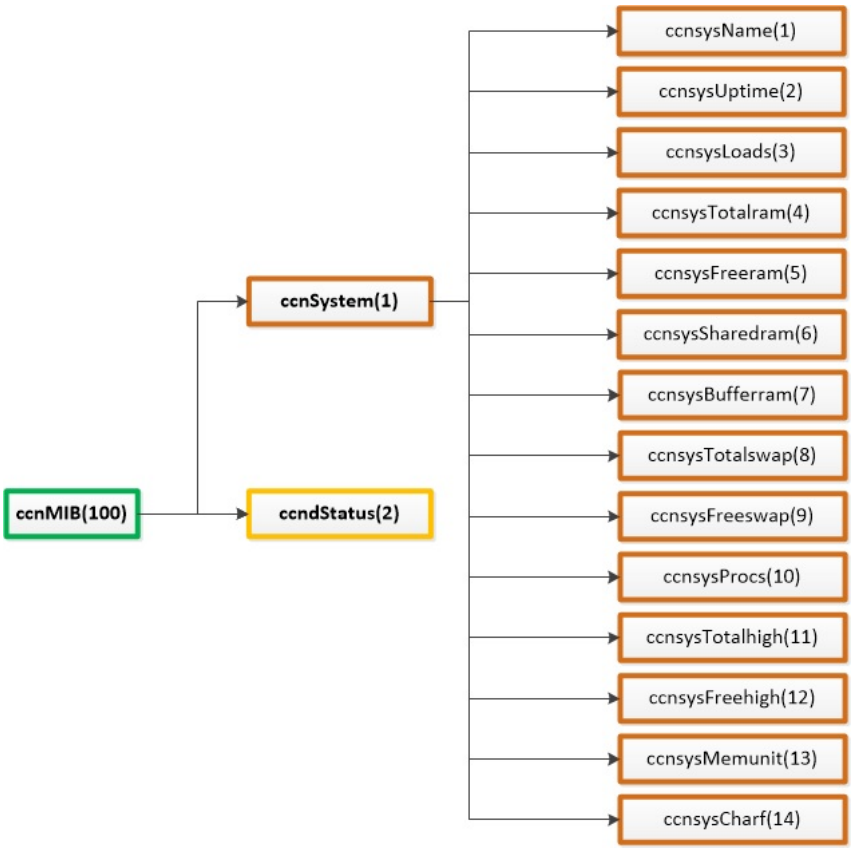

Figura B.1: Ramo *ccnSystem*.

Objetos da MIB CCN sob o ramo ccndStatus

A *Figura B.2* reflete objetos relativos ao estado do protocolo CCN coletados através da ferramenta **ccndStatus**.

Esquemas em formato XML foram utilizados para definir as informações extraídas através da ferramenta CCND status, conforme link abaixo.
http://www.ccnx.org/releases/ccnx-0.8.2/doc/technical/CCNDStatus.html.

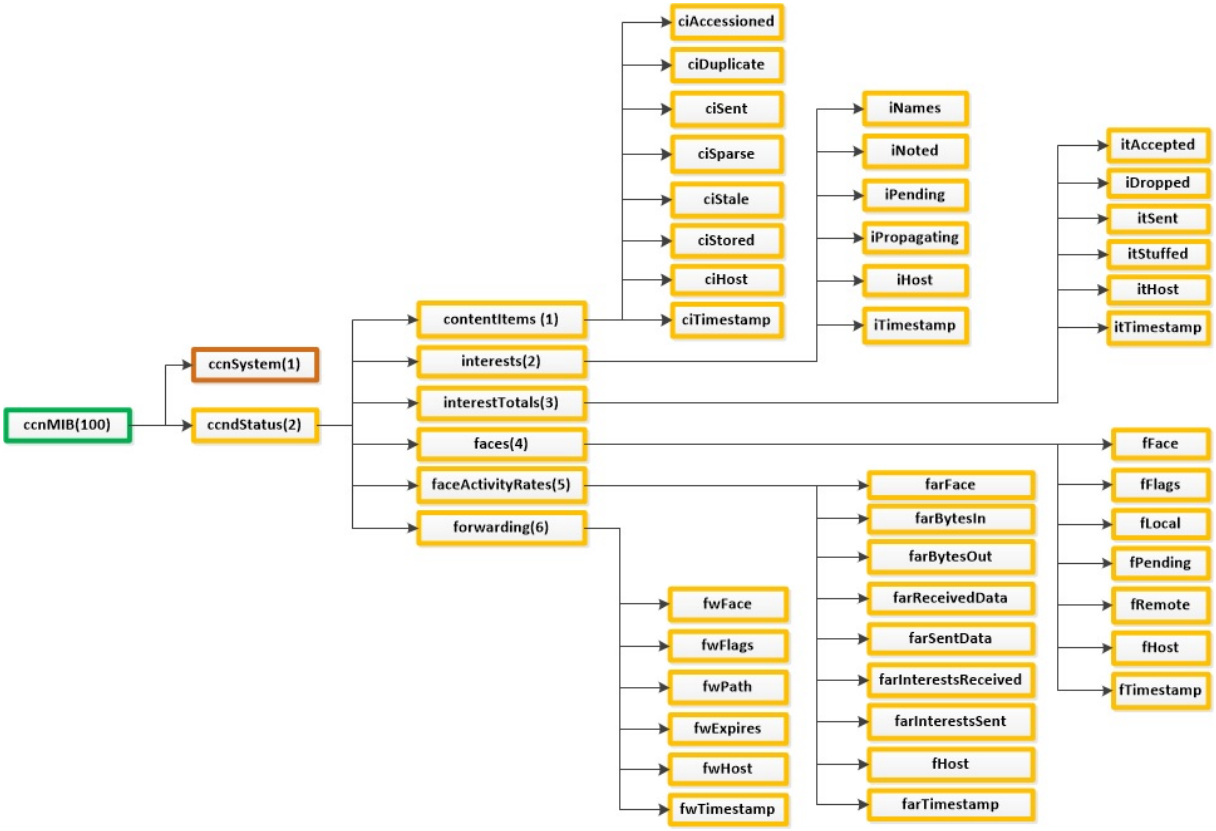

Figura B.2: Ramo *ccndStatus*.