



## Trabalho de Graduação

# CRIAÇÃO DE UM AMBIENTE DE ESTUDO BASEADO NO SIMULADOR DE REDES NS-2

---

Eduardo da Silva Martins

Curso de Ciência da Computação

Santa Maria, RS, Brasil

2004

**CRIAÇÃO DE UM AMBIENTE DE ESTUDO  
BASEADO NO SIMULADOR DE REDES NS-2**

---

por

**Eduardo da Silva Martins**

Trabalho de Graduação apresentado ao Curso de Ciência da  
Computação – Bacharelado, da Universidade Federal de  
Santa Maria (UFSM, RS), como requisito parcial para  
obtenção do grau de  
**Bacharel em Ciência da Computação.**

**Curso de Ciência da Computação**

Trabalho de Graduação nº 181

Santa Maria, RS, Brasil

2004

**Universidade Federal de Santa Maria  
Centro de Tecnologia  
Curso de Ciência da Computação**

A Comissão Examinadora, abaixo assinada, aprova o Trabalho de  
Graduação

**CRIAÇÃO DE UM AMBIENTE DE ESTUDO  
BASEADO NO SIMULADOR DE REDES NS-2**

elaborado por  
**Eduardo da Silva Martins**

como requisito parcial para obtenção do grau de Bacharel em Ciência  
da Computação.

**COMISSÃO EXAMINADORA:**

---

**Raul Ceretta Nunes**  
(Orientador)

---

**Roseclea Duarte Medina**  
(Co-orientadora)

---

**Benhur de Oliveira Stein**

---

**Antônio Marcos de Oliveira Candia**

# Sumário

<b>Lista de Tabelas</b>	<b>1</b>
<b>Lista de Figuras</b>	<b>2</b>
<b>1 Introdução</b>	<b>3</b>
1.1 Motivação . . . . .	3
1.2 A Realidade da UFSM . . . . .	4
1.3 O <i>Network Simulator</i> (NS) . . . . .	4
1.4 Asterix . . . . .	5
1.5 Proposta e Objetivo deste Trabalho . . . . .	5
1.6 Organização do Texto . . . . .	6
<b>2 O NS e o Asterix</b>	<b>7</b>
2.1 NS - Network Simulator . . . . .	7
2.1.1 A Interface para o Interpretador . . . . .	7
2.1.2 Base da Simulação . . . . .	9
2.1.2.1 A Classe <i>Simulator</i> . . . . .	11
2.1.2.2 Nós e Repasse de Pacotes . . . . .	11
2.1.2.3 Ligações Simples . . . . .	12
2.1.2.4 Gerenciamento de Fila e Escalonamento de Pacote . . . . .	12
2.1.3 Agentes . . . . .	12
2.1.4 A Estrutura de Endereçamento no NS . . . . .	13
2.1.5 Suporte . . . . .	14
2.1.5.1 Matemático . . . . .	14

---

2.1.5.2	Rastreamento e Monitoração . . . . .	14
2.1.6	Roteamento . . . . .	15
2.1.6.1	Roteamento Simples . . . . .	15
2.1.6.2	Rede Dinâmica . . . . .	16
2.1.7	Aplicação . . . . .	16
2.1.8	Animador de redes . . . . .	17
2.2	Asterix . . . . .	20
<b>3</b>	<b>O NS na Disciplina de Comunicação de Dados</b>	<b>22</b>
3.1	Mapas Conceituais . . . . .	22
3.1.1	Método Relacional . . . . .	24
3.1.2	Método Estrutural . . . . .	24
3.2	Metodologia Utilizada na Disciplina . . . . .	25
3.2.1	Avaliar estágio cognitivo inicial do aluno - Mapa 1 . . . . .	25
3.2.2	Apresentação . . . . .	26
3.2.3	Utilização do NS . . . . .	27
3.2.4	Avaliar o crescimento cognitivo do aluno após a utilização do NS - Mapa 2 . . . . .	28
3.3	Metodologia Usada no Ambiente . . . . .	29
3.3.1	Disponibilidade do NS . . . . .	29
3.3.2	Tutorial . . . . .	29
3.3.3	Exemplos . . . . .	30
3.3.4	Exercícios Propostos . . . . .	30
3.3.5	Desafios . . . . .	30
3.3.6	Projeto de Simulação . . . . .	30
<b>4</b>	<b>Resultados</b>	<b>31</b>
4.1	NS com a turma . . . . .	31
4.1.1	Aspectos positivos encontrados . . . . .	32
4.1.2	Dificuldades encontradas . . . . .	32

4.2	Exercícios de Simulação . . . . .	33
4.2.1	Aspectos positivos ao analisar os scripts de simulação . . . . .	33
4.2.2	Dificuldades percebidas ao analisar os scripts de simulação . . . . .	34
4.3	Mapa Conceitual . . . . .	34
4.3.1	Avaliação quantitativa . . . . .	35
4.3.2	Análise qualitativa . . . . .	37
4.3.3	Comparação dos mapas iniciais e finais . . . . .	38
<b>5</b>	<b>Conclusão</b>	<b>40</b>
<b>A</b>	<b>Anexo 1 - Instalação e Tutorial</b>	<b>44</b>
A.1	Instalação do Simulador em Sistema Operacional Linux . . . . .	44
A.2	Tutorial . . . . .	48
<b>B</b>	<b>Anexo 2 - Exercícios de Simulação</b>	<b>52</b>
<b>C</b>	<b>Anexo 3 - Métodos de avaliação de Mapas Conceituais</b>	<b>59</b>

# Lista de Tabelas

4.1	Tabela de escores dos primeiros mapas. . . . .	36
4.2	Tabela de escores dos últimos mapas. . . . .	37

# Lista de Figuras

2.1	Script de Simulação . . . . .	8
2.2	Interface da Simulação . . . . .	9
2.3	Script de Simulação NAM . . . . .	18
2.4	Ligações entre nós . . . . .	18
2.5	Script - Envios de Pacotes . . . . .	19
2.6	Visualização - Envios de Pacotes . . . . .	19
2.7	Laboratório Virtual . . . . .	21
3.1	Exemplo de Mapa Conceitual . . . . .	23
3.2	<i>site</i> - ambiente digital criado para apoio . . . . .	28
4.1	Primeiro Mapa Conceitual (Mapa D) . . . . .	36
4.2	Segundo Mapa Conceitual . . . . .	39
C.1	Método Relacional [11] . . . . .	59
C.2	Método Estrutural [11] . . . . .	60



# Capítulo 1

## Introdução

Compreender o funcionamento de uma rede de computadores e realizar estudos de avaliação baseados em experimentos reais podem ser atividades de grande complexidade e alto custo. Por esse motivo, simulação é uma técnica utilizada com muita frequência, devido sua flexibilidade para testar cenários variados. Por exemplo, pode-se testar o comportamento de protocolos e novas tecnologias bem como o efeito de diferentes topologias.

O estudo de protocolos sob condições variadas é um elemento importante para compreender o seu comportamento e suas características. Em sala de aula, a compreensão dos conceitos pode se auxiliada por simuladores. Através da Simulação é possível avaliar protocolos de rede sobre uma grande variedade de condições, o que permite distinguir o benefício de cada tecnologia estudada.

Neste trabalho, discute-se a criação de um ambiente de estudo de comunicação de dados baseado no *Network Simulator-NS-versão 2* [1], um simulador que permite avaliar os diversos aspectos que compõem uma rede complexa, e no Asterix [4], um laboratório virtual de auxílio ao processo de ensino e aprendizagem da disciplina de redes de computadores.

### 1.1 Motivação

A motivação para este trabalho é saber como se pode melhorar a qualidade de ensino/aprendizagem das disciplinas ligadas a redes de computadores fazendo uso de

um simulador de rede. Entretanto, existem vários problemas, dificuldades e dúvidas que são muito comuns entre os que estão iniciando a utilizar um simulador de rede. Mesmo quem já tem algum tempo de uso num simulador de redes também pode encontrar algumas dificuldades de nível médio a avançado. Neste sentido é importante introduzir o uso de simuladores e avaliar o quanto ele auxilia na compreensão dos conhecimentos.

## 1.2 A Realidade da UFSM

A Universidade Federal de Santa Maria possui vários laboratórios informatizados para o estudante e pesquisa em diversas áreas. Especificamente, estes laboratórios dispõem de diversos computadores e equipamentos para seus alunos de graduação, pós-graduação e também para seus professores.

Por outro lado, na área de redes, os laboratórios não costumam utilizar de *softwares*, como ferramentas de apoio, para simulações em seus computadores, a fim de visualizar o comportamento de variadas redes, o que facilitaria o ensino e aprendizagem nas disciplinas da área de redes de computadores.

## 1.3 O *Network Simulator* (NS)

O NS [1] é um simulador de redes dirigido por eventos que simula vários tipos de redes IP. Ele pode trabalhar simulando protocolos de rede como o *Transport Control Protocol* (TCP) e o *User Datagram Protocol* (UDP), comportamento de tráfego em *File Transfer Protocol* (FTP), Telnet, Web, *Constant Bit Rate* (CBR) e VBR, mecanismos de gerenciamento de filas de roteadores como Drop Tail, RED e CBQ, algoritmos de roteamento como de *Dijkstra* e outros.

O NS também implementa multicast e alguns dos protocolos da camada MAC para simulação de LANs. O projeto NS é parte de projeto VINT, que desenvolve ferramentas para mostrar resultados de simulações, para realizar análises e para transformar topologias de redes, geradas no formato NS.

## 1.4 Asterix

O ASTERIX [4] é um laboratório virtual para auxiliar no processo de ensino e aprendizagem da matéria de redes de computadores. Visa suprir, em parte, a ausência que temos de laboratório físico de redes, mas não tem a pretensão de substituir um laboratório físico.

No ASTERIX temos a confluência de 3 áreas distintas: realidade virtual, simulação/animação e inteligência artificial. Todo o projeto, a seleção das tecnologias que fazem parte, a implementação e a utilização, tem como embasamento educacional a Aprendizagem Significativa [3], onde a aprendizagem parte de um conjunto de significados que são pontos de partida para a atribuição de outros significados.

O NS foi integrado ao laboratório virtual ASTERIX, executado em uma janela separada dentro do ambiente 3D. Ambos (NS e ASTERIX) hospedados no Teleduc, que é um ambiente virtual de aprendizagem, desenvolvido pela Unicamp e que está operacional no servidor AIA do CE/UFSM.

## 1.5 Proposta e Objetivo deste Trabalho

Primeiro, possibilitar que alunos de graduação, pós-graduação e de cursos de treinamento pratiquem o uso de simulador em seus experimentos curriculares, a fim de facilitar a construção e assimilação do conhecimento.

Segundo, disponibilizar conteúdos digitais, sobre as diversas metodologias e processos de utilização da ferramenta de simulação NS via Asterix, contribuindo assim de forma efetiva com outros projetos de ensino a distância.

Terceiro, criar exemplos práticos de utilização do NS e documentá-los, para a melhor compreensão das funcionalidades do simulador e para que os interessados possam mais facilmente criar suas próprias simulações utilizando ao máximo os seus recursos.

Quarto, melhorar a qualidade de ensino com a realização de experimentos, de simulação, na disciplina de Comunicação de Dados (ELC123) do curso de Ciência da Computação.

Quinto, integrar o NS no ASTERIX, enriquecendo o contexto do ambiente virtual para aprendizado de redes com o acréscimo das simulações de funcionamento dos protocolos.

Enfim, a integração do NS ao Asterix e a conseqüente simulação do funcionamento de protocolos em sala de aula, visando facilitar o aprendizado na área de redes de computadores e avaliar o impacto do uso de um simulador no aprendizado, compõem o objetivo deste trabalho.

## 1.6 Organização do Texto

O texto deste trabalho, nos capítulos que seguem, está dividido em três partes. No segundo capítulo estão descritos conceitos e detalhes de funcionamento do simulador de redes NS e o ASTERIX.

No terceiro capítulo está descrita a metodologia de como a ferramenta de simulação (NS) vai atuar numa turma da disciplina de comunicação de dados no sentido de melhora do aprendizado.

No quarto capítulo estão os resultados coletados tanto na forma de avaliação dos alunos, fazendo uso de Mapas Conceituais, como na análise do comportamento em sala de aula e nos scripts entregues como exercícios da disciplina. Por fim, a conclusão seguida dos anexos que são referenciados em várias partes do texto, terminam assim o texto desse trabalho.

# Capítulo 2

## O NS e o Asterix

Neste capítulo estão descritos as características e funcionalidades do simulador de redes NS - *Network Simulator*, e as características do ASTERIX, que é uma laboratório virtual para auxílio na aprendizagem nas disciplinas de redes de computadores.

### 2.1 NS - Network Simulator

O Network Simulator [1] disponibiliza aos seus usuários vários recursos de emulação para simular uma rede. Usando sua interface, é possível descrever a topologia e o comportamento de diferentes configurações de redes. Essa seção tem como objetivo descrever particularidades do simulador que serão de importância para o aluno no papel de usuário.

#### 2.1.1 A Interface para o Interpretador

NS é um simulador orientado a objetos, escrito em C++, com um interpretador OTcl [1] como interface para usuário. Suporta uma hierarquia de classes em C++ e uma hierarquia de classes similar no interpretador OTcl. A principal destas hierarquias é a classe TclObject. Usuários criam novos objetos de simulação através do interpretador; estes objetos são instanciados no interpretador; que finalmente são espelhados por um objeto compilado correspondente. A hierarquia de classes do interpretador é automaticamente estabelecida através de métodos definidos na

classe TclClass. Objetos instanciados pelo usuário são espelhados através de métodos definidos na classe TclObject. Há outras hierarquias nos códigos C++ e scripts OTcl, mas estas outras não são espelhadas.

Para exemplificar o uso da interface, o exemplo da Figura 2.1 descreve uma simulação na linguagem OTcl. A simulação descreve uma rede de quatro nós, onde há um nó central (n0) onde todos os outros estão ligados. A partir de um determinado instante de tempo os nós (n1,n3) iniciam uma transferência de dados com outro nó (n2) até o instante de tempo na qual a simulação termina. As partes desse exemplo serão explicadas no decorrer da seção.

```
set ns [new Simulator]

set tr [open trace.out w]

$ns trace-all $tr
$ns namtrace-all [open out.nam w]

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$ns duplex-link $n0 $n1 3Mb 1ms DropTail
$ns duplex-link $n0 $n2 3Mb 1ms DropTail
$ns duplex-link $n0 $n3 3Mb 1ms DropTail
$ns duplex-link $n2 $n3 3Mb 1ms DropTail

set udp1 [new Agent/UDP]
set udp2 [new Agent/UDP]
set udp3 [new Agent/UDP]

set cbr0 [new Application/Traffic/CBR]
set cbr1 [new Application/Traffic/CBR]

$ns attach-agent $n1 $udp1
$ns attach-agent $n2 $udp2
$ns attach-agent $n3 $udp3

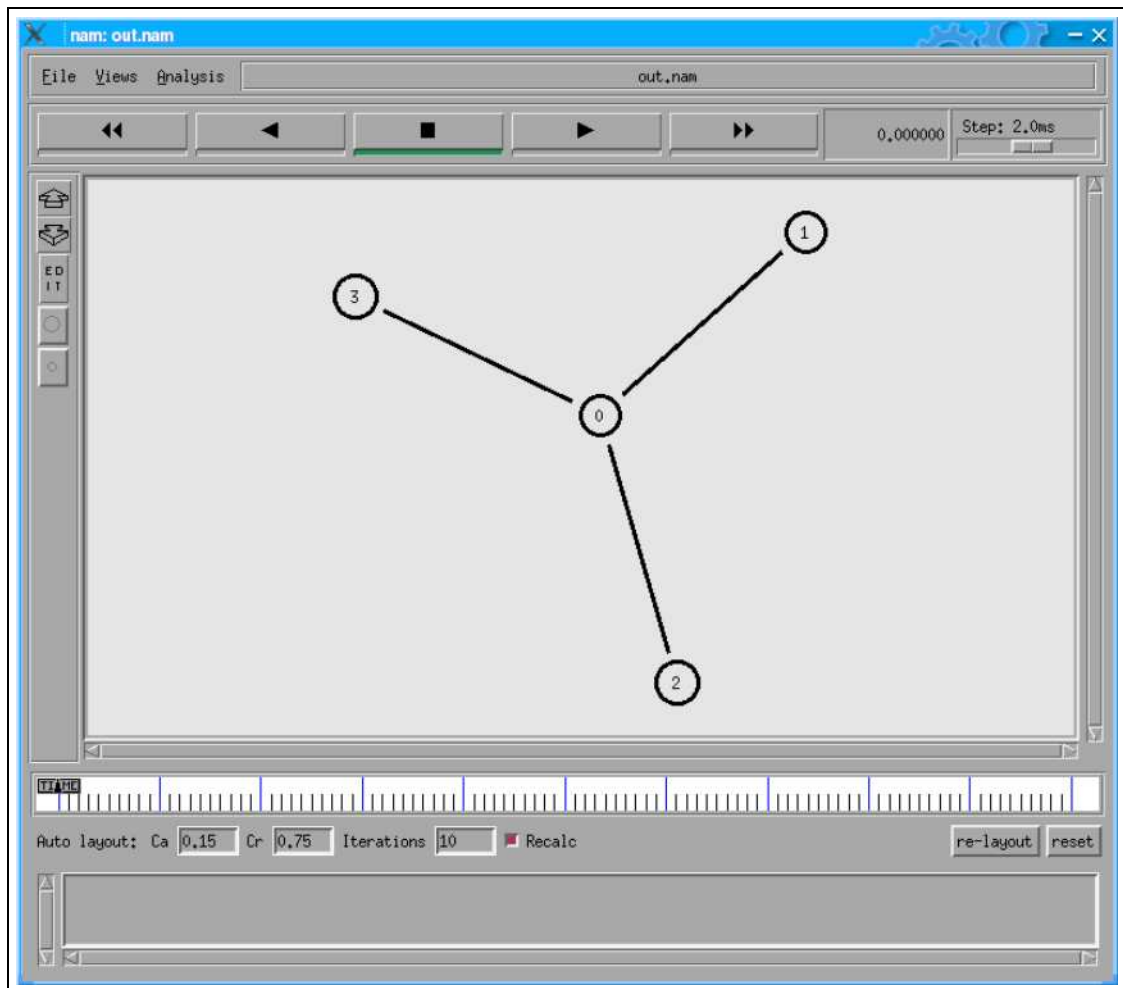
$cbr0 attach-agent $udp1
$cbr1 attach-agent $udp3

$ns connect $udp3 $udp2
$ns connect $udp1 $udp2

$ns at 1.0 "$cbr0 start"
$ns at 1.0 "$cbr1 start"
$ns at 15.0 "exec nam out.nam &"
$ns at 15.0 "exit 0"

$ns run
```

Figura 2.1: Script de Simulação



**Figura 2.2:** Interface da Simulação

A figura 2.1 mostra o exemplo anterior sendo visualizado no NAM que é a ferramenta de visualização de simulações para o NS. As características do NAM serão descritas mais adiante. Pode-se notar na figura os círculos simbolizando os quatro nós descritos no script e as linhas que simbolizam suas ligações.

### 2.1.2 Base da Simulação

O projeto do simulador levou em consideração o fato de que diferentes simulações requerem diferentes modelos de programação. Portanto, o NS adota duas linguagens de programação: C++ para o núcleo do simulador (*back-end*) e OTcl para a construção de *scripts* e modelagem da simulação (*front-end*). O objetivo é

prover flexibilidade sem prejudicar o desempenho.

O usuário do simulador precisará conhecer apenas os blocos básicos (comandos e as classes de objetos que implementam os protocolos na linguagem de scripts OTcl) para a construção de uma simulação. Contudo quando o usuário desejar incluir novas características a protocolos implementados, ou mesmo implementar novos protocolos, é necessária a implementação direta usando o C++.

Nó código a seguir a implementação da função de recebimento de pacotes TCP do objeto TCPSink, em C++.

```
void TcpSink::rcv(Packet* pkt, Handler*)
{
    hdr_tcp *th = (hdr_tcp*)pkt->access(off_tcp_);
    acker_->update(th->seqno());
    ack(pkt);
    Packet::free(pkt);
}
```

A seguir o código em OTcl que cria um um objeto do agente TCPSink.

```
set sink [new Agent/TCPSink]
```

Mecanismos e protocolos implementados em C++ tem um correspondente OTcl e vice-versa. Os *scripts* contêm chamadas a objetos OTcl ligados a objetos C++. Assim todos esses mecanismos e protocolos estão descritos em forma de classes. Quando o usuário desejar utilizar alguns desses mecanismos e classes, ele deve então instanciar esses objetos que estão descritos tanto em OTcl como em C++.

A seguir algumas descrições básicas de como esses mecanismos e protocolos se comportam numa simulação do NS. Essas descrições referem-se muitas vezes às classes que estão vinculados os mecanismos e protocolos.



### 2.1.2.1 A Classe *Simulator*

Tudo na simulação é descrita através de uma classe TCL chamada *Simulator*. Isto proporciona um conjunto de interfaces para se configurar uma simulação e para escolher o tipo de eventos do escalonador (início da simulação, início das aplicações, rompimento de uma ligação, ...) usado para conduzir a simulação. O script de simulação geralmente inicia criando uma instância de classes e chamando vários métodos para criar os nós, topologias, e configurar outros aspectos da simulação, como no exemplo abaixo.

```
set ns [new Simulator]
```

### 2.1.2.2 Nós e Repasse de Pacotes

A função de um nó, quando este recebe um pacote, é examinar os campos do pacote. Usualmente o endereço de destino e ocasionalmente o endereço de origem. Conhecendo os campos, o nó pode então mapear os valores para uma interface de saída a qual é o próximo receptor do canal de fluxo desse pacote. Abaixo os exemplos de como se define um nó.

```
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
```

No exemplo, está sendo criados quatro nós (n0,n1,n2,n3), e dizendo que eles são nós da simulação (\$ns).

### 2.1.2.3 Ligações Simples

A classe *SimpleLink* proporciona a habilidade de conectar dois nós como uma ligação ponto a ponto. O NS proporciona a instância de procedimento *simple-link* para formar uma ligação unidirecional de um nó a outro. No exemplo abaixo está sendo criada uma ligação simples (ligação de via única), onde está conectando o nó n0 com o nó n1 que é uma ligação de 3Mb/s e seu atraso é de 1ms e o tipo de fila que se está usando é DropTail.

```
$ns simple-link $n0 $n1 3Mb 1ms DropTail
```

### 2.1.2.4 Gerenciamento de Fila e Escalonamento de Pacote

Filas representam locais, em um nó, onde os pacotes podem ficar, para depois serem enviados a outro nó, ou serem descartados. Escalonamento de pacotes se refere a decisão de processamento usada para escolher quais pacotes serão servidos ou descartados. O exemplo a seguir é semelhante ao exemplo anterior, a única diferença é o tipo de ligação (duplex-link) que defini uma ligação em duas vias. Nesse exemplo a fila está exemplificada como "DropTail".

```
$ns duplex-link $n0 $n1 3Mb 1ms DropTail
```

## 2.1.3 Agentes

Como os nós não implementam diretamente nenhum protocolo da camada de transporte, esse papel é desempenhado por agentes, podendo-se dizer então que a camada de transporte são os próprios agentes. Os dois agentes definidos no NS correspondem aos protocolos TCP e UDP. Novos protocolos presentes em várias camadas da arquitetura de rede podem ser implementados a partir da definição da classe *agent*.

Agentes UDP aceitam dados de tamanhos variados de uma aplicação e segmentam os dados caso seja necessário.

Há dois tipos de agentes TCP: agentes de uma via e agente de duas vias. Agentes de uma via são subdivididos num conjunto de emissores e receptores. Agentes de duas vias representam simetricamente ambos, emissores e receptores. Abaixo um exemplo onde está sendo criados três agentes do tipo UDP para atuarem como camada de transporte.

```
set udp1 [new Agent/UDP]
set udp2 [new Agent/UDP]
set udp3 [new Agent/UDP]
```

O próximo passo está vinculando os agentes criados anteriormente com os nós da simulação.

```
$ns attach-agent $n1 $udp1
$ns attach-agent $n2 $udp2
$ns attach-agent $n3 $udp3
```

#### 2.1.4 A Estrutura de Endereçamento no NS

O espaço de endereçamento consiste de 2 partes, o nó-id e a porta-id. Os bits mais significativos assinalam o nó-id e os outros assinalam a porta-id ou a identificação do agente anexado ao nó. O bit mais significativo, se estiver assinalado com "1", indica o uso de multicast. O padrão são os 7 bits mais significativos para o nó-id, sendo o primeiro para multicast e os outros menos significativos para a porta-id. Naturalmente isto limita a simulação para 128 nós, Este espaço de endereçamento pode ser expandido para acomodar um grande número de nós na simulação. A porta-id

pode também ser expandida para suportar um grande número de agentes. Adicionalmente, o espaço de endereçamento também pode ser configurado num formato hierárquico, consistindo de múltiplas camadas de endereçamento.

### 2.1.5 Suporte

O NS disponibiliza algumas facilidades de apoio ao usuário, para melhorar as simulações no sentido de ter dados mais confiáveis. Funções matemáticas e de monitoramento são exemplos deste suporte. Deste modo, após uma simulação o usuário pode chegar a algumas conclusões mais facilmente.

#### 2.1.5.1 Matemático

O simulador inclui uma pequena coleção de funções matemáticas usadas para gerar variáveis aleatórias e integrações. Esta área do simulador está constantemente sofrendo modificações [1]. O exemplo a seguir cria uma variável com um valor aleatório.

```
set randRNG [new RNG]
```

#### 2.1.5.2 Rastreamento e Monitoração

Há várias maneiras de coletar a saída ou o caminho dos dados na simulação. Geralmente, cada caminho dos dados é mostrado diretamente durante a execução da simulação, ou (mais comumente) armazenadas num arquivo pós-processado e analisado.

Há duas maneiras primárias mas de diferentes tipos de monitoramento atualmente suportadas pelo simulador. A primeira, chamada *traces*, grava cada pacote individualmente. A segunda é chamada *monitors*, grava a contagem de varias quantidades de pacotes. O Monitoramento pode monitorar contagens associadas com todos os pacotes, ou de cada fluxo de dados. O exemplo que segue está definindo esse rastreamento num script, cria-se uma instância de rastreamento "tr" vinculando-a

a um arquivo no sistema de arquivos "trace.out".

```
set tr [open trace.out w]
```

A seguir o script está vinculando a instância de rastreamento a simulação. Logo depois está sendo criado um arquivo de rastreamento para a interface de visualização (NAM).

```
$ns trace-all $tr
```

```
$ns namtrace-all [open out.nam w]
```

## 2.1.6 Roteamento

Roteamento no NS proporciona ambientes próximos dos reais, simulando protocolos de roteamento usados em diversas redes. Os protocolos implementados para o NS, assim como seus comportamentos estão descritos a seguir.

### 2.1.6.1 Roteamento Simples

A estratégia de roteamento é um mecanismo pelo qual o NS vai computar as rotas de tráfego. Há três estratégias de roteamento no NS: Estática, Sessão, e Dinâmica. Assim, um protocolo de roteamento é a realização de um algoritmo específico. Atualmente, roteamento Estático e Sessão usam algoritmo de Dijkstra's [5]. Roteamento de estratégia dinâmica é atualmente implementado com o algoritmo distribuído Bellman-Ford [6]. No NS, não se enfatiza a distinção entre estratégias e protocolos para roteamento estático e de sessão, considerando-os protocolos simples. Como exemplo, o comando a seguir configura a simulação para roteamento do tipo vetor de distância.

```
$ns rtproto DV
```

### 2.1.6.2 Rede Dinâmica

O simulador NS tem também a capacidade de simular situações em redes dinâmicas. A interface para o usuário para redes dinâmicas é uma coleção de procedimentos instanciados da *class Simulator*, e um procedimento para rastreamento e *log* da atividade de rede dinâmica.

Quando ocorre da topologia sofrer algum tipo de alteração, quando uma nova ligação surge ou desaparece, a rede se adapta a esse novo estado, calculando novamente as rotas de roteamento e assim não há mais perda de pacotes ou se assume uma nova rota para eles. para que isso ocorra é necessário que a rede esteja configurada corretamente com o tipo vetor de distância.

Os comandos a seguir, são comandos de eventos que podem ser configurados na simulação a fim de se observar a rede quando uma determinada ligação se rompe (primeira linha) e após algum tempo se reestabelece (segunda linha).

```
$ns rtmodel-at 3.5 down $n0 $n1
```

```
$ns rtmodel-at 6.5 up $n0 $n1
```

### 2.1.7 Aplicação

Este elemento é responsável por gerar tráfego de dados para o simulador. Na verdade, o NS define alguns modelos de tráfego que são utilizados para simular aplicações.

Aplicações estão situadas no topo dos agentes de transporte no NS. Há dois tipos básicos de aplicações: geradoras de tráfego e aplicações simuladas.

Há atualmente duas classes de "aplicações simuladas" derivadas da classe *Application*: *Application/FTP* e *Application/Telnet*. Estas classes trabalham por uma contagem de pacotes disponível para ser enviadas por um agente de transporte TCP. A atual transmissão de pacotes disponível é controlada pelo controle de fluxo e congestio-

namento TCP. Abaixo um exemplo criando duas aplicações `cbr0` e `cbr1` que são do tipo *Traffic/CBR*, que é uma aplicação para gerar tráfego contínuo na simulação.

```
set cbr0 [new Application/Traffic/CBR]
set cbr1 [new Application/Traffic/CBR]
```

### 2.1.8 Animador de redes

O *Network Animator (NAM)* é um animador de redes que acompanha o NS e é utilizado para compreender o que ocorre durante a simulação. Através do *NAM* pode-se visualizar a topologia da rede, bem como acompanhar o fluxo de pacotes e seus conteúdos. Quando o *NAM* é ativado, apresenta um console que pode gerenciar várias atividades paralelamente, como animações e criação de simulações [2].

Durante a simulação, o NS gera um ou mais arquivos rastreamento que contém dados detalhados da simulação, para visualização posterior. A criação destes arquivos é opcional e depende da adição de alguns comandos no script de simulação OTcl, conforme explicado na seção 2.1.3.2. No final da simulação, o *NAM* pode ser acionado explicitamente no script para interpretar o arquivo de rastreamento e mostrar a animação da simulação.

Na figura 2.3 está descrito um script de simulação e na figura 2.4 está o resultado na interface de visualização *NAM*.

A figura 2.4 mostra uma situação de simulação sendo visualizada no *NAM*. É uma ligação ( linha fina ) entre dois nós (círculos). As linhas mais espessas simbolizam os pacotes enviados e recebidos pelos nós.

O script da figura 2.5 descreve uma simulação um pouco mais completa, utilizando o envio de muitos pacotes de dois dos nós, enchendo assim a fila do nó central, ocorrendo então o descarte de pacotes. A figura 2.6 mostra o resultado desse script no *NAM*.

```
#Criando a simulação
set ns [new Simulator]

#Criando e abrindo o arquivo de rastreamento NAM
set nf [open out.nam w]
$ns namtrace-all $nf

#Criando dois nós
set n0 [$ns node]
set n1 [$ns node]

#Criando uma ligação entre os nós.
$ns duplex-link $n0 $n1 1Mb 10ms DropTail

#Terminando após 5 segundos
$ns at 5.0 "exec nam out.nam &"
$ns at 5.0 "exit 0"

#Rodando a simulação
$ns run
```

Figura 2.3: Script de Simulação NAM

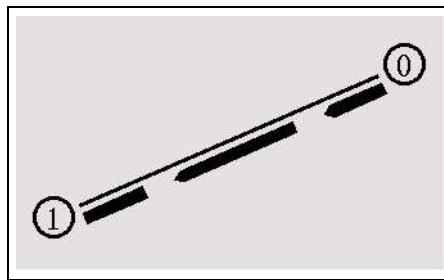


Figura 2.4: Ligações entre nós

A figura 2.6 mostra uma situação de simulação sendo visualizada no NAM, é uma pequena simulação de envio de pacotes dos nós 0 e 1 para o nó 3. Os pequenos pontos acima do nó 2 simbolizam a fila de pacotes e os que estão "caindo" simbolizam o descarte de pacotes.



```

#Criando a simulação
set ns [new Simulator]
#definindo diferentes cores para os pacotes de origens diferentes
$ns color 1 Blue
$ns color 2 Red
#Criando e abrindo o arquivo de rastreamento NAM
set nf [open out.nam w]
$ns namtrace-all $nf
#Criando quatro nós
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
#Criando os lnks entre os nós\
$ns duplex-link $n0 $n2 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n3 $n2 1Mb 10ms SFQ
#Criando um agente UDP e anexando-o no nó n0
set udp0 [new Agent/UDP]
$udp0 set class_ 1
$ns attach-agent $n0 $udp0
#Criando uma aplicação que vai gerar um tráfego constante e anexando no no
agente udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
#Criando um agente UDP e anexando-o no nó n1
set udpl [new Agent/UDP]
$udpl set class_ 2
$ns attach-agent $n1 $udpl
#Criando uma aplicação que vai gerar um tráfego constante e anexando no no
agente udpl
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udpl
#Criando um agente nulo, apenas para receber os pacotes, e anexando ao nó n3
set null0 [new Agent/Null]
$ns attach-agent $n3 $null0
#Conectando os agentes que vão se comunicar
$ns connect $udp0 $null0
$ns connect $udpl $null0
#Escalonando os eventos
$ns at 0.5 "$cbr0 start"
$ns at 1.0 "$cbr1 start"
$ns at 4.0 "$cbr1 stop"
$ns at 4.5 "$cbr0 stop"
#Terminando a simulação após 5 segundos
$ns at 5.0 "exec nam out.nam &"
$ns at 5.0 "exit 0"
$ns run

```

Figura 2.5: Script - Envios de Pacotes

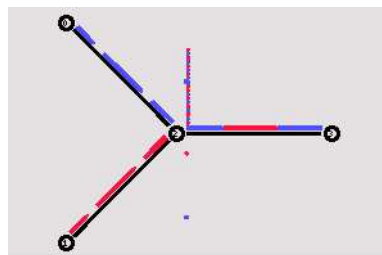


Figura 2.6: Visualização - Envios de Pacotes

## 2.2 Asterix

O ASTERIX é um laboratório virtual para auxiliar no processo de ensino e aprendizagem da disciplina de redes de computadores. Visa suprir, em parte, a ausência que temos de laboratório físico de redes, embora não possa substituir um laboratório físico. O nome ASTERIX vem de: Aprendizagem Significativa e Tecnologias aplicadas no Ensino de Redes de computadores: Integrando e eXplorando possibilidades.

A Realidade Virtual foi utilizada para desenvolver o ambiente do laboratório virtual de redes. O aluno "entra" numa sala/laboratório onde é apresentada uma bancada com dispositivos utilizados para extensão e segmentação de redes, como *hub*, *bridge*, *switch* e roteador. (Figura 2.4)

Ao utilizar a sala do laboratório (Figura 2.4), o aluno poderá, por exemplo, visualizar com detalhes os dispositivos de rede construídos com realidade virtual, como exemplo o switch. Todos os equipamentos podem ser visualizados em 3D.

O NS pode fazer parte deste conjunto, enriquecendo o contexto de redes com o acréscimo das simulações de funcionamento dos protocolos. O NS está integrado ao Asterix como uma contribuição externa, na forma de um *link* anexado dentro do laboratório virtual ASTERIX [4].

O projeto do laboratório virtual ASTERIX [4] tem como embasamento educacional a Aprendizagem Significativa, que é um conceito enfatizado por David Paul Ausubel [10] desde a década de 1960. Propondo uma explicação teórica do processo de aprendizagem, segundo um ponto de vista cognitivista, embora reconheça a importância da experiência afetiva.

Cognição é o processo através do qual o mundo de significados tem origem. A medida que o ser se situa no mundo, estabelece relações de significação, isto é, atribui significados à realidade em que se encontra. Esses significados não são entidades estáticas, mas pontos de partida para a atribuição de outros significados. Tem origem, então, a estrutura cognitiva (os primeiros significados), constituindo-se nos "pontos básicos de ancoragem" dos quais derivam outros significados.



**Figura 2.7:** Laboratório Virtual

Para haver aprendizagem significativa é preciso haver duas condições:

- o aluno precisa ter uma disposição para aprender: se o indivíduo quiser memorizar o material arbitrariamente e literalmente, então a aprendizagem será mecânica;
- o material a ser aprendido tem que ser potencialmente significativo, ou seja ele tem que ser logicamente e psicologicamente significativo. O significado lógico depende somente da natureza do material e o significado psicológico é uma experiência que cada indivíduo tem. Cada aprendiz faz uma filtragem dos materiais que têm significado ou não para si próprio.

## Capítulo 3

# O NS na Disciplina de Comunicação de Dados

O NS é um simulador para análise de redes. Ele possui uma interface simples (scripts OTcl), vários módulos especializados, bem como uma interface de visualização (NAM). Neste capítulo descreve-se utilização do simulador na primeira disciplina ligada a redes de computadores do Curso de Ciência da Computação da UFSM, no caso a disciplina de Comunicação de Dados.

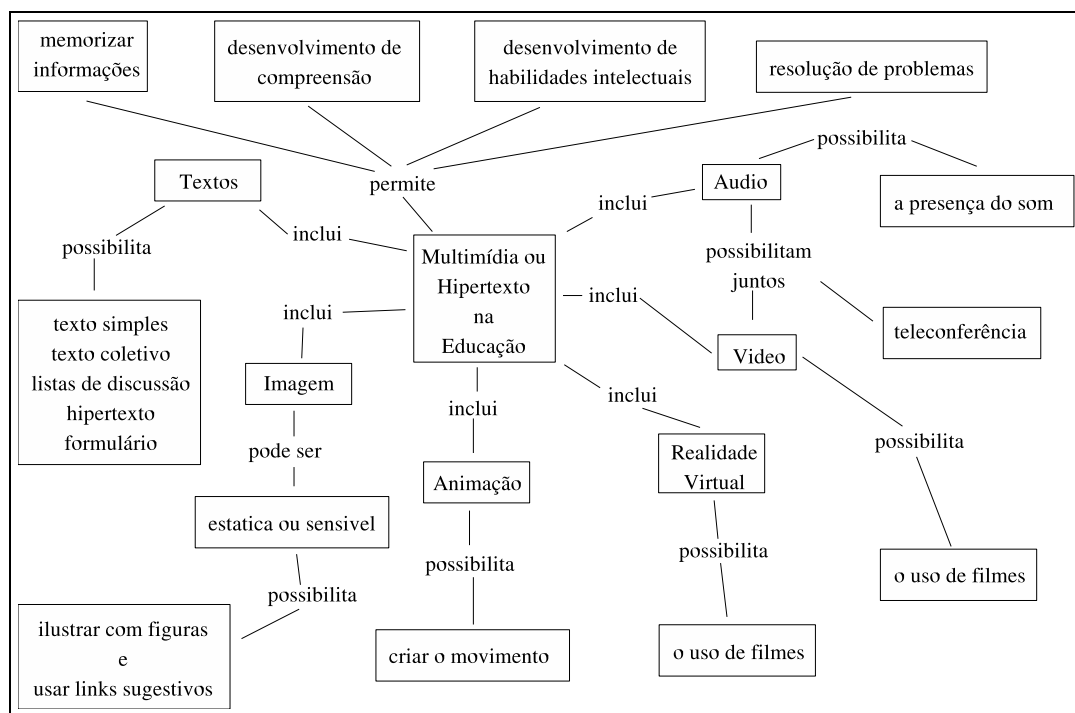
Esse capítulo irá descrever as metodologias usadas para a utilização do NS como apoio ao aprendizado na turma da disciplina de comunicação de dados, bem como as ferramentas de apoio e recursos que viabilizaram essas metodologias. Uma das ferramentas são os Mapas Conceituais, que nesse trabalho foram usados como avaliação e métrica da aprendizagem, e conceitualizados a seguir.

### 3.1 Mapas Conceituais

Mapas conceituais [9] são um recurso para a representação do conhecimento. Eles se constituem em uma rede de nós ou links representando conceitos ou objetos conectados por ligações com descritores das relações entre pares e nós. Neste trabalho os mapas foram utilizados para representar o estado cognitivo inicial (antes do uso do NS) e final (após o uso do NS) de cada grupo de alunos.

A figura 3.1 é de um mapa conceitual usado no curso de pós-graduação em informática na Educação na UFRGS. O assunto é sobre Multimídia na Educação

(caixa central). As linhas referem-se às relações entre conceitos, onde cada uma possui um rótulo que as levam a outro conceito.



**Figura 3.1:** Exemplo de Mapa Conceitual

Podem também ser conceitualizados como representações gráficas semelhantes a diagramas, que indicam relações entre conceitos ligados por palavras. Representam uma estrutura que vai desde os conceitos mais abrangentes até os menos inclusivos. São utilizados para auxiliar a ordenação e a seqüenciação hierarquizada dos conteúdos de ensino, de forma a oferecer estímulos adequados ao aluno. Dessa forma então será representado de forma abrangente o conhecimento adquirido de novos conceitos e as relações entre esses conceitos que estão presentes na simulações de rede e na disciplina de comunicação de dados.

Como os Mapas Conceituais são uma maneira de representar o conhecimento, podem então servir de métrica para avaliar o aprendizado, podendo também ser possível atribuir valores numéricos para os mapas com regras de pontuação de acordo com os métodos desenvolvidos para tal. Assim os mapas feitos pelos alunos foram avaliados e pontuados utilizando o método relacional e o método estrutural. Os

métodos, relacional e estrutural, são descritos a seguir.

### 3.1.1 Método Relacional

Neste método de atribuição de escores são realizadas atribuições individualmente para cada preposição identificada no mapa. Uma preposição é definida como dois conceitos conectados por uma seta e um rótulo, indicando o relacionamento entre esses conceitos. Para cada preposição pode ser atribuído um escore de zero a três de acordo com o grau de validade da relação. O escore final do mapa é encontrado somando-se os escores de todas as preposições do mapa. A seguir um exemplo desse método utilizando a Figura 3.1, um esquema detalhado desse método está no Anexo 3, Figura 8.1.

- a relação é válida? +1 ponto;
- o nome da relação está correto? +1 ponto;
- a sequência da relação está correta? +1 ponto;
- soma-se os pontos de cada preposição.

Segundo o esquema apresentado e aplicando-o na Figura 3.1, o mapa obteve segundo o método relacional um escore de 54 (18 x 3), pois verificou-se que todas as 18 relações cumpriram os itens anteriores.

### 3.1.2 Método Estrutural

Neste método é realizada uma contagem dos elementos do mapas. Conta-se as preposições, que são dois conceitos conectados por uma seta e um rótulo. Conta-se os níveis hierárquicos, que é o número de níveis que vai do conceito mais próximo do tema, até o conceito com o nível mais distante, e multiplica-se por cinco. Conta-se os *cross-links*, que são as relações entre preposições de origens diferentes, multiplicando-as por dez, E por último os exemplos de conceitos presentes no mapa. Somando-se esses cinco elementos, temos então o escore final. A seguir um exemplo

desse método, utilizando a Figura 3.1, e um esquema detalhado desse método está em no Anexo 3, Figura 8.2.

- 18 preposições: +18 pontos;
- 3 hierarquias: + 5 x 3 pontos;
- 1 cross-links: +10 x 1 pontos;
- 0 exemplos: +0 pontos.

Segundo o esquema apresentado e aplicando-o na Figura 3.1, o mapa obteve segundo o método relacional um escore de 43.

## 3.2 Metodologia Utilizada na Disciplina

Para inserir e avaliar a utilização do simulador na disciplina, aqui estão descritas as etapas que foram usadas numa interação direta com os alunos. Esta interação foi realizada em quatro etapas: avaliação inicial, apresentação do sistema para a turma, utilização do NS, segunda avaliação pós-utilização. A seguir cada uma é descrita com detalhes.

### 3.2.1 Avaliar estágio cognitivo inicial do aluno - Mapa 1

Como uma primeira avaliação aos alunos da disciplina, foi solicitado a eles a criação de um Mapa Conceitual, abordando o assunto TCP/IP. Os alunos já tinham conhecimento sobre mapas.

A avaliação foi executada sem aviso prévio, para detectar o conhecimento inicial do aluno antes da utilização do NS, evitando assim a "memorização pré-prova", e garantindo a validade do instrumento de coleta. Aqui é importante destacar que essa coleta de dados ocorreu após a turma ter encerrado o módulo TCP/IP na disciplina, ou seja, esse conteúdo já havia sido apresentado e concluído. A técnica de mapeamento conceitual utilizada no trabalho foi a "*low-directed*" (onde os mapas são construídos do zero).

### 3.2.2 Apresentação

Para uma introdução dos alunos no ambiente do simulador, é feita uma apresentação com o objetivo de preparar os alunos para uma nova carga de conhecimento e de informações sobre simulação e o NS. A apresentação foi composta dos seguintes tópicos:

- Simulações de Rede

Introduzir aos alunos o conceito de Simulação. Definir de onde vem a necessidade de se simular e quais benefícios irão lhe proporcionar. Deixá-los conscientes que depois terão capacidade de tomar decisões quanto à rede de computadores, aproximando-os de um ambiente real e preparando-os para enfrentar o mercado de trabalho.

- O que é o NS?

Apresentar aos alunos o simulador NS, mostrando onde ele se encaixa dentro da simulação.

- Por que usar o NS?

Apresentar um conjunto de qualidades do NS, tornando-o atrativo para sua utilização, mostrando possíveis situações em que terão a necessidade de utilizar o NS. Motivando e despertando a curiosidade de um primeiro contato, para com uma nova ferramenta.

- Características do NS.

Mostrar um conjunto de possibilidades e recursos disponíveis para as futuras simulações.

- Um pequeno passo-a-passo

Um pequeno passo-a-passo para iniciar uma primeira familiarização com a sintaxe.



- Exemplos.

Alguns exemplos para uma maior fixação da sintaxe e do ambiente de simulação NS.

Após a apresentação, é feita uma seção de exemplos práticos em conjunto, muito importante para observar as maiores dificuldades de quem está apenas começando a simular. Esta seção permite observar se houve o interesse esperado.

### 3.2.3 Utilização do NS

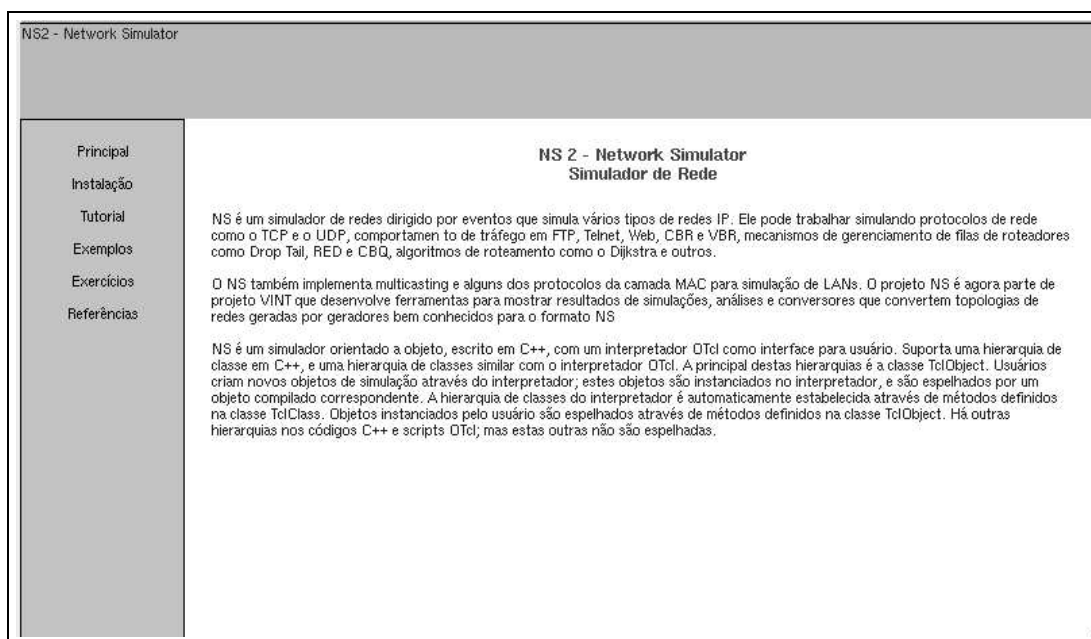
A utilização do simulador de rede NS é realizada pela primeira vez logo após a apresentação. Os alunos são colocados pela primeira vez em contato com o simulador de redes, num ambiente onde apenas possuem uma introdução de como seria simular com o NS. Essa utilização tem como principal objetivo deixar os alunos em contato com alguns conceitos de redes, conceitos de protocolos, comportamentos e camadas onde eles atuam.

Com a realização de um exemplo, com acompanhamento, é realizada uma descrição de topologia e comportamento de uma rede simples. Assim realiza-se uma simulação e uma observação de resultados, que são comparados com outros resultados gerados de pequenas modificações do comportamento ou topologia.

Os próximos passos da utilização devem seguir seu fluxo por meio do ambiente digital criado disponível na Internet (*site* de apoio), como mostra a figura 3.2, onde os conceitos já absorvidos irão ser agregados a outros.

A utilização prevê o uso do *NAM* que proporciona um ambiente mais atrativo, pois permite uma visualização gráfica dos resultados obtidos das simulações. O uso dessa interface tem como objetivo acelerar o processo de aprendizado, trazendo aos alunos algo muito mais compreensivo.

Os passos metodológicos para a utilização do simulador com apoio do ambiente serão descritos posteriormente.



**Figura 3.2:** *site* - ambiente digital criado para apoio

### 3.2.4 Avaliar o crescimento cognitivo do aluno após a utilização do NS - Mapa 2

Após a utilização prática do simulador é realizada uma nova avaliação utilizando Mapa Conceitual. Utilizou-se do mesmo tema da primeira avaliação. A avaliação é realizada num momento onde os alunos já estiveram em contato com conceitos de simulação de redes.

A nova avaliação tem como objetivo compará-la com a avaliação anterior (antes da utilização do NS), tendo assim provas do conhecimento dos alunos após a utilização do NS como apoio do aprendizado da disciplina. As diferenças que ocorreram entre os mapas iniciais e finais se devem exclusivamente a utilização do NS, visto que não houve retomada do assunto TCP/IP em sala de aula. A comparação entre as avaliações será pelo número de conceitos, relações entre esses conceitos e coesão lógica entre os mapas dentro de regras. Desta forma pode-se gerar escores de acordo com o método estrutural e relacional.

### 3.3 Metodologia Usada no Ambiente

A metodologia empregada no ambiente digital tem como objetivo traçar um fluxo de orientação para os alunos da disciplina. O fluxo leva os alunos a conhecer o ambiente, fixar a aprendizagem e motivarem-se a executar simulações de nível real.

Os alunos da disciplina poderão estar em contato com as informações do simulador através do ambiente digital criado, que nada mais é que um *site* contendo as informações citadas nesse capítulo e que pode ser acessado via Internet, no endereço <http://www.inf.ufsm.br/ns>. Onde quer que estejam, tendo um computador com Internet, os alunos podem assim acessar o ambiente.

#### 3.3.1 Disponibilidade do NS

Foi disponibilizado para uso o NS no Núcleo de Ciência da Computação (NCC), e também, disponibilizado para *download* no ambiente digital, dando oportunidade aos alunos realizarem seus experimentos e tarefas pertinentes a disciplina.

A instalação do simulador no NCC foi feita em conjunto com a administração de rede, não havendo maiores dificuldades. Seguindo o manual de instalação disponível no ambiente digital.

Foi disponibilizado também o *The Network Animator* (NAM), uma interface de visualização para o NS, para que os alunos visualizem seus exercícios de simulação. Algumas vezes por motivos didáticos, é de grande valor uma ferramenta visual.

#### 3.3.2 Tutorial

A disponibilidade de um tutorial é de importância para que os alunos interessados aprendam mais rapidamente sobre a ferramenta em questão. Assim foi feito um tutorial resumido.

O tutorial agrega as qualidades de outros tutoriais existentes, muitos estão escritos em uma língua estrangeira o que dificulta a rápida síntese do leitor, podendo muitas vezes confundir alguns detalhes, o que acarreta um atraso quando se quer desempenhar uma determinada função com o simulador.

### **3.3.3 Exemplos**

Diante das várias maneiras de aprendizagem, muitas pessoas buscam o seu aprendizado de diferentes formas, talvez para alguns os exemplos sejam uma das melhores e mais rápidas formas de aprendizagem. Neste sentido, vários exemplos de uso do simulador são criados e aplicados. Com os exemplos o aluno se sentirá mais a vontade no ambiente do simulador, facilitando uma primeira utilização do sistema.

### **3.3.4 Exercícios Propostos**

Os exercícios são uma forma de fixação do que se está aprendendo, e aqui o aluno poderá sentir a necessidade de voltar ao tutorial ou buscar uma informação mais completa. Além disso, os exercícios tem como objetivo preparar o indivíduo para problemas mais reais.

### **3.3.5 Desafios**

Nessa etapa, o aluno se sentirá desafiado a resolver problemas em questão e assim motivado para resolvê-los.

### **3.3.6 Projeto de Simulação**

Como projeto de simulação, o aluno será colocado num ambiente de um problema real a ser implementado, e sendo um profissional da área, então seria uma pessoa apta a resolvê-lo. A especificação do projeto está no anexo 2.

# Capítulo 4

## Resultados

Os resultados obtidos foram coletados segundo a metodologia apresentada no capítulo 3. Aqui estão apresentados os resultados que se puderam extrair diretamente de cada aluno, bem como da observação do comportamento dos alunos diante dos exercícios propostos em sala de aula. A análise dos scripts de simulações feitos extra classe, também contribuíram na geração de resultados.

A criação de mapas conceituais foi o instrumento utilizado para obter evidências de uma aprendizagem significativa decorrente da utilização do simulador NS. A análise dos resultados se fixou nos mapas conceituais, desenvolvidos pelos alunos, considerando um enfoque qualitativo e quantitativo dos mesmos.

### 4.1 NS com a turma

Foram usados alguns exemplos de acompanhamento diretos com a turma num estágio inicial de uso das simulações. Também foram propostos exemplos para se verificar dificuldades e avanços nos conceitos de uso da ferramenta de simulação.

O uso começou de forma mais lenta no início até se habituarem e se sentirem mais a vontade com a ferramenta de simulação, ponto em que estavam preparados para realizar os exercícios.

Os aspectos positivos e as dificuldades foram observadas enquanto os alunos realizavam as tarefas propostas e quando haviam perguntas de dúvidas decorrentes do objetivo da simulação. As dificuldades provenientes de erros foram resolvidas

por rápidas detecções dos problemas pelos próprios alunos, e esses próprios alunos cooperaram em ajudar os outros que se encontravam em situações de erros semelhantes.

#### 4.1.1 Aspectos positivos encontrados

Durante a execução das simulações pelos alunos, foram observados alguns aspectos positivos, como os interesses por parte da maioria em encontrar a solução de algum erro, chegando assim a realizar a simulação pedida.

No início do processo de implementação de uma ou outra simulação, o processo parecia estar lento, mas logo, com a ajuda mútua dos participantes, todos conseguiram desempenhar as simulações.

#### 4.1.2 Dificuldades encontradas

Aqui estão descritos as dificuldades encontradas pelos alunos em laboratório enquanto elaboravam suas simulações.

Comandos essenciais no script de simulação estão com a primeira letra em maiúscula, confundindo assim um usuário iniciante, sofrendo um pequeno atraso na detecção do problema.

Confusões quando for anexar os agentes nos nós e nas aplicação devendo ter de tomar certos cuidados, algumas questões devem ser cumpridas:

- anexar agentes diferentes em nós diferentes, quando necessário. Algumas vezes ocorria de estar usando por engano dois agentes num mesmo nó, quando a simulação necessitava que estivessem em diferentes nós;
- anexar agentes diferentes em aplicações diferentes, quando necessário. Semelhante ao anterior o mesmo acontecia com as aplicações;
- tomar o devido cuidado e atenção ao iniciar as aplicações, verificar em qual agente ela está anexada e se a conexão entre os agentes está correta. Ocorreu algumas vezes o engano de tentar conectar aplicações com agentes e vice-versa;

- fazer uma conexão equivocada de agentes, conectando agentes de nós diferentes dos necessitados que não participavam do objetivo da simulação ou conectar em agentes inexistentes;
- tentar conectar aplicações com agentes e vice-versa, inviabilizando assim a simulação, quando a simulação cresce de complexidade, esse tipo de erro é mais difícil de ser detectado. Perdendo assim muito tempo que poderia ser evitado com uma maior atenção na hora da implementação.

Num caso de engano, algumas simulações estavam tentando conectar agentes incompatíveis, inviabilizando assim a comunicação de dados entre os nós da simulação.

Anexar agentes em aplicação que não se encaixam. A aplicação necessitava de um tipo de agente, e tentou-se anexar um agente de outro tipo equivocadamente.

Com a ajuda do material de apoio (tutorial, exemplos) é possível minimizar esses problemas, pois lá estão presentes várias situações de casos onde as simulações funcionam.

## 4.2 Exercícios de Simulação

Foram propostos 6 exercícios de simulação que seguiram as características da metodologia. Os exercícios pedidos estão em anexo ao trabalho (Anexo 2).

As simulações 1, 2, 3 e 4 se encaixam na categoria de "Exercícios Propostos" do capítulo 3, servindo para treinar o uso dos scripts OTcl, bem como fixar a sua sintaxe. A simulação 5 se encaixa como "Desafio" pois é de uma maior complexidade e é apresentado um problema que se precisa resolver. A simulação 6 se encaixa como "Projeto de Simulação" pois é semelhante aos projetos de estruturação de redes que encontramos na vida real.

### 4.2.1 Aspectos positivos ao analisar os scripts de simulação

Dentro das simulações feitas e entregues pelos alunos, pode-se notar que houve interesse por parte deles na busca de maiores informações e recursos para um melhor

uso do sistema. como controlar o fluxo de pacotes que estão sendo enviados por um certo agente e classificar os nós dentro de um mesmo grupo com características semelhantes.

Uso de cores diferentes para nós e pacotes facilitando a análise da simulação.

As simulações quase em sua totalidade demonstraram uma exploração de recursos maior dos que mostrados e treinados em sala de aula.

Demonstram também uma boa organização dentro dos scripts, deixando eles fáceis de entender, auxiliando no processo de análise. Inclusive com diversos comentários que ocorreram com frequência no desenvolver do código.

#### **4.2.2 Dificuldades percebidas ao analisar os scripts de simulação**

Aconteceram algumas dificuldades dos alunos ao fazerem os scripts de simulação:

- não foi feito o uso de funções de controle, como por exemplo usar laços de repetição para a criação de nós, ligações, agentes, aplicações e eventos;
- visualizar uma rede real dentro da simulação, tendo a visão de que um equipamento como um HUB, pode ser representado por um nó;
- representar uma rede externa, que pode ser representado como um nó que teria ligações mais lentas, simulando assim os vários saltos que um certo pacotes teria que dar até o seu destino da rede externa.

### **4.3 Mapa Conceitual**

A criação de mapas conceituais foi outro instrumento utilizado para obter evidências de uma aprendizagem significativa decorrente da utilização do NS.

No início do semestre, antes da utilização do NS e após a apresentação do TCP/IP, foi solicitado aos alunos que elaborassem um mapa conceitual inicial para evidenciar o conhecimento prévio sobre redes de computadores (especificamente



TCP/IP). De acordo com a metodologia proposta no capítulo anterior, ao final do semestre foi solicitado um segundo mapa (mapa final) com o objetivo de verificar a influência da utilização do NS na consolidação dos conceitos de redes na estrutura cognitiva dos alunos. Para a construção dos mapas foi facultado aos alunos a utilização do software Cmap Tools, The Brain, editor de textos e até mesmo lápis/caneta e papel.

Conforme já citado nesse trabalho, os mapas conceituais são utilizados com muitos propósitos. Aqui estamos considerando sua aplicação na área educacional, e mais especificamente, como uma ferramenta para avaliar a aprendizagem significativa dos conceitos básicos de redes decorrente do uso do simulador de redes NS.

Vários autores (Novak e Gowin [12]; Wallace e Mintzes [13]; Markham [14]; Pearsall [15]) apresentaram pesquisas sobre a utilização de mapas conceituais como ferramenta de avaliação da aprendizagem, utilizando em seus trabalhos diferentes métodos para atribuir valor aos mapas. Nesse trabalho, de acordo com a metodologia proposta, foi adotada uma avaliação qualitativa analisando os mapas em termos de sua estrutura geral, baseando-se na adaptação do trabalho de autores como Amoretti [9]. Também foi realizado um levantamento quantitativo, baseado nas propostas de Novak [10] e McClure [11], resumidas no anexo 3.

### 4.3.1 Avaliação quantitativa

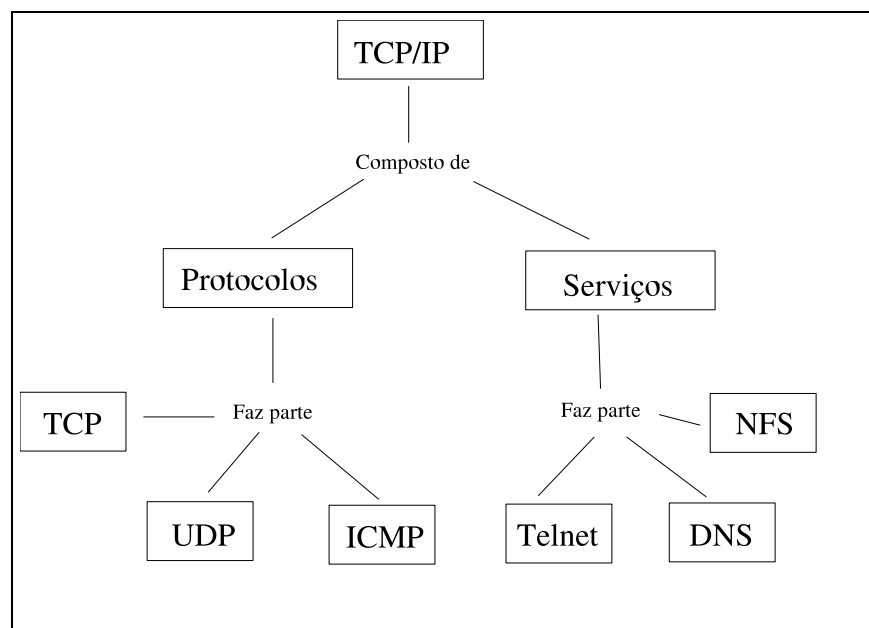
Um resumo da avaliação quantitativa pode ser observado nas tabelas 4.1 e 4.2. Nessas tabelas é apresentada a pontuação obtida pelos mapas desenvolvidos pelos alunos, no método estrutural proposto por Novak e relacional, de McClure [11], bem como a classificação obtida pelos mapas.

Cabe ressaltar que as tabelas 4.1 e 4.2 não tem a intenção de estabelecer uma comparação entre dois métodos de avaliação de mapas mas pontuar os mapas apresentados pelos alunos. A tarefa de atribuição de valores é difícil e demorada e os resultados variam de acordo com o método utilizado. A realização de uma análise pura e simples dos pontos obtidos pode levar a uma indistinção do que realmente o escore revela e do que pode ser efetivamente avaliado. Com relação a atribuição de

MAPA	Escores	
	Método estrutural	Método relacional
Mapa A	61	39
Mapa B	30	45
Mapa C	23	27
Mapa D	28	24
Mapa E	54	24

**Tabela 4.1:** Tabela de escores dos primeiros mapas.

escores, o próprio Novak afirma ser irrelevante pois o mais importante é observar as trocas qualitativas dos mapas conceituais dos alunos, mas ainda segundo o autor, a maioria dos alunos e professores "precisam" pontuar os mapas em função de vivermos numa sociedade "orientada a números".



**Figura 4.1:** Primeiro Mapa Conceitual (Mapa D)

Tanto nos escores apresentados, quanto na análise qualitativa dos mapas, foi possível observar uma relativa homogeneidade no grau de subsunção<sup>1</sup>, representado pelos níveis hierárquicos. Essa estrutura sugere a ocorrência de um processo de assimilação de conceitos ou proposições sob uma idéia mais inclusiva, no caso "Redes".

<sup>1</sup>Subsunção é o processo onde uma idéia, um conceito ou uma preposição, já existiu na estrutura cognitiva do aluno, serve de âncora a nova informação ou conceito, de modo que esse novo conceito adquira significado para o aluno.

MAPA	Escore	
	Método estrutural	Método relacional
Mapa 1	87	114
Mapa 2	143	96
Mapa 3	142	108
Mapa 4	105	84
Mapa 5	84	51
Mapa 6	85	75
Mapa 7	130	66
Mapa 8	80	57

**Tabela 4.2:** Tabela de escores dos últimos mapas.

### 4.3.2 Análise qualitativa

O grau de integração conceitual percebido nos mapas foi considerado muito bom, apesar do atributo *cross-link* apresentar uma discrepância entre os valores. Essa interpretação favorável é decorrente de uma análise conjunta que considerou a quantidade de *cross-links* ( $c$ ) nos mapas mas também a qualidade dos mesmos. Todos os mapas, em maior ou menor número, apresentaram integração de conexões entre os conceitos subordinados em diferentes ramificações dos mapas.

Mesmo apresentando diferentes escores, os mapas desenvolvidos foram coerentes, coesos, expressivos e lógicos. O "conflito cognitivo" ou "desvio cognitivo" (diferença entre os níveis de categorização) foi percebido em um mapa, que chamamos de "mapa exceção". Em nenhum momento, ou seja, nos mapas impressos e nas apresentações dos mesmos, houve a ocorrência de dois ou mais rótulos conceituais expressando o mesmo conteúdo ou então o mesmo rótulo utilizado para expressar mais de um conceito. Também não sucedeu a confusão com conceitos semelhantes, isso leva a crer que com a utilização do NS as diferenças conceituais ficaram bem explicitadas, transparecendo nos mapas que os conceitos foram percebidos e memorizados como distintos. A medida que os conceitos foram sendo apresentados no NS de várias formas, as diferenças começaram a ser explicitadas, ocorrendo o que Ausubel [10] chama de "princípio da diferenciação progressiva", onde os conceitos mais gerais vão sendo progressivamente individualizados em termos de detalhe e

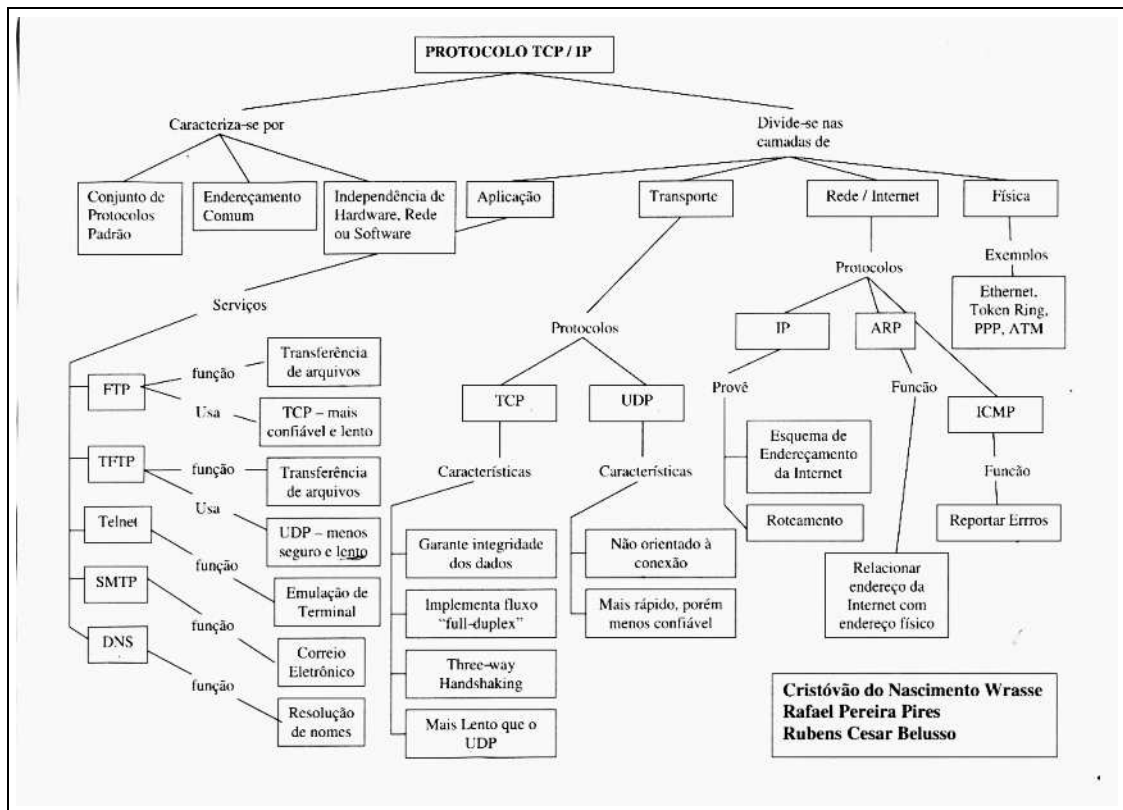
especificidade.

### 4.3.3 Comparação dos mapas iniciais e finais

Os mapas iniciais mostraram o conhecimento prévio dos alunos em relação aos conceitos de redes que seriam trabalhados e apresentaram um pequeno número de conceitos e relações (exemplo em Figura 4.1). Os mapas finais (exemplo em Figura 4.2) - mostram tanto a influência da utilização do NS, representada pelo aumento significativo de conceitos gerais e relações entre conceitos, como a consolidação do conhecimento adquirido, sugerindo a ocorrência de aprendizagem significativa e não apenas memorização temporária. Cabe destacar que nos mapas iniciais, desenvolvidos antes da utilização do NS, predominaram relações do tipo "é" e após a utilização do NS (onde a interação com as tecnologias disponibilizadas ocorreu de forma diversificada, auxiliando na integração de novos conceitos aos já existentes) foi observado um aumento no número e na qualidade das proposições, visto que nos mapas finais encontram-se mais relações como "possui", "suscetível a", "suporta", "reduz", "atinge", "possibilita", "permite", "amplia".

O resultado obtido na comparação entre os mapas iniciais e finais deve-se ao fato de que mapas conceituais são dinâmicos, estão constantemente mudando no curso da aprendizagem significativa. Isso foi observado no comportamento dos alunos com relação aos mapas pois as alterações continuaram ocorrendo inclusive no momento da apresentação dos mesmos e após também, sendo compartilhado no ambiente Teleduc a nova versão do mapa. Essas alterações, feitas de forma espontânea pelos alunos, demonstram que a estrutura cognitiva está continuamente se reorganizando, por diferenciação progressiva e reconciliação integrativa, o que sugere claramente a ocorrência de uma aprendizagem significativa dos conceitos envolvidos.

As análises qualitativas e quantitativas se mostraram consistentes com a literatura (McClure, Sonak e Suen [11]; Shavelson [8]) . A validade e a confiabilidade da utilização de mapas conceituais como ferramenta no processo de avaliação do aprendizado é defendida por vários autores como Ruiz-Primo [7], Shavelson e Hu-



**Figura 4.2:** Segundo Mapa Conceitual

ang (2003) [8]. Shavelson e Ruiz-Primo (2000) no seu artigo "Windows into the Mind" utilizam mapas conceituais para avaliar a estrutura do conhecimento e fornecem evidências da sua qualidade técnica. Eles mostram que os mapas proporcionam uma "janela dentro da estrutura de conhecimento" dos alunos, de forma diferenciada dos testes tradicionais. Para Amoretti (2000), os mapas são uma representação aberta do conhecimento e podem ser ferramentas capazes de mostrar as mudanças ocorridas na capacidade de representação prévia do aluno.

O resultado da análise realizado com a atribuição de escores aos mapas, comparando o primeiro mapa realizado com o segundo, mostra que houve um indiscutível aumento nos escores (tabela 4.2), nos mostrando uma melhora nos conceitos e em suas relações, após o uso do simulador de redes NS. Assim, comprova-se que houve, por parte dos alunos, uma melhora no aprendizado.

# Capítulo 5

## Conclusão

Os resultados obtidos mostram a evolução do aprendizado de conceitos e suas relações do conteúdo da disciplina de comunicação de dados, usando o NS como ferramenta de simulação e que apoiou o ensino.

Os resultados finais, quanto a evolução, se mostraram positivos com a utilização da ferramenta de apoio segundo os resultados da comparação dos mapas conceituais iniciais e finais dos alunos, obtendo assim a melhora do aprendizado.

Os aspectos positivos aqui notados também podem ser observados na execução de outros experimentos usando também semelhantes ferramentas de apoio no auxílio do ensino de dentro de sala de aula. As dificuldades podem ser evitadas usando de precauções quando o ensino ainda se encontrar em fase inicial.

Para o campo de pesquisa acadêmica, quando for de interesse este trabalho se torna um exemplo de caso de uso de uma ferramenta de ensino para apoio em sala de aula e para um caso de uso de Mapas Conceituais como métrica para comparação do estado de aprendizagem anterior e posterior ao uso do NS.

## Referências Bibliográficas

- [1] FALL, K; VARADHAN, K. The ns Manual (formerly ns Notes and Documentation) [online]. Disponível na Internet via URL <http://www.isi.edu/nsnam/ns/doc/>. Fevereiro de 2000.
- [2] KAMIENSKI, C; SADOK, D; CAVALCANTI; D. SOUSA, D; DIAS, K. **Simulando a Internet** [online]. Disponível na Internet via URL <http://www.dcc.ufmg.br/~fred/redes.pdf> . Arquivo capturado em 08.03.2004.
- [3] MEDINA, R. D; TAUROUCO, L. M. R. **Tecnologias Aplicadas no Ensino de Redes de Computadores: um Protótipo de Laboratório Virtual para Facilitar a Aprendizagem Significativa.** Cacic'2003. Argentina 2003.
- [4] MEDINA, R. D. **Tecnologias Aplicadas no Ensino de Redes de Computadores: ASTERIX - Aprendizagem Significativa e Tecnologias aplicadas no Ensino de Redes de computadores Integrando e eXplorando possibilidades.** Proposta de tese de doutorado. PEIE/UFRGS - Porto Alegre. 2004.
- [5] SHUTZ, F; WAGNER, D; WEIHE, K. **Dijkstra's Algorithm** [online]. Disponível na Internet via URL <http://i11www.ilkd.uni-karlsruhe.de/algo/people/dwagner/papers/sww-daole-00.pdf>. Arquivo capturado em 12.05.2004

- 
- [6] BLACK, P. **Bellman-Ford algorithm** [online]. Disponível na Internet via URL <http://www.nist.gov/dads/HTML/bellmanford.html>. Junho de 2004
- [7] RUIZ-PRIMO, M. A. Shavelson, R. J. Problems and issues in the use of concept maps in science assessment. *Journal of Research in Science Teaching*, 33(6), 569-600. 1996.
- [8] SHAVELSON, R.J., HUANG, Leta. **Responding responsibly to the frenzy to assess learning in higher education**. in: *Change*, V.35(1), 10-19. 2003.
- [9] AMORETTI, Maria Suzana Marc. **Protótipos e estereótipos: aprendizagem de conceitos - Mapas Conceituais: experiência em Educação à Distância**. In: *Revista Informática na Educação: Teoria Prática*. Porto Alegre: PGIE-UFRGS, v.4, n.2, dez./2001.
- [10] AUSUBEL, D.P.; NOVAK, J.D.; HANESIAN, H. **Educational psychology: A cognitive view**. 2.ed., Nova York, Holt: Rinehart and Winston Inc., 1978.
- [11] MCCLURE, J. R.; SONAK, B.; SUEN, H K. **Concept Map Assessment of Classroom Learning: Reliability, Validity and Logistical Practicality** Department of Education Psychology, Pennsylvania State University. Abril de 1998.
- [12] NOVAK, J.D.; GOWIN, D.B. (1984). **Learning how to learn**. New York, N.Y.: Cambridge University Press. (traduzido em Português pela Plátano,1999).
- [13] Wallace, J. and Mintzes, J. (1990). **The concept map as a research tool: Exploring conceptual change in biology**. *Journal of Research in Science Teaching*, 27(10): 1033-1052.



- 
- [14] Markham, K., Mintzes, J. and Jones, G. (1994). **The concept map as a research and evaluation tool: Further evidence of validity.** *Journal of Research in Science Teaching*, 31(1): 91-101.
- [15] Pearsall, R., Skipper, J., and Mintzes, J. **Knowledge restructuring in the life sciences: A longitudinal study of conceptual change in biology.** *Science Education*, 81, 193-215.

# Apêndice A

## Anexo 1 - Instalação e Tutorial

### A.1 Instalação do Simulador em Sistema Operacional Linux

Antes de começar a instalação deve-se ter em mãos os sistemas necessários para o funcionamento do simulador de redes NS, abaixo a listagem do arquivos que devem ser baixados, os links abaixo são opcionais, você pode encontrar (e possivelmente irá) outros locais para busca dos arquivos mais atualizados, e também poderá encontrar outros tutoriais de instalação.

- \* tcl8.4.5-src.tar.gz ( 3.3M )
- \* tk8.4.5-src.tar.gz ( 3.1M )
- \* otcl-1.8.tar.gz ( 294K )
- \* tclcl-src-1.15.tar.gz ( 140K )
- \* ns-src-2.27.tar.gz ( 43M )
- \* nam-src-1.10.tar.gz opcional ( 4.2M )

Existem comandos e aplicativos específicos utilizados para a instalação são em sistemas Linux. Além de ter o sistema instalado é necessário também um grau de conhecimento do sistema para efetuar os comandos e caminhar pela árvore do sistema de arquivos.

Inicialmente crie um diretório.

```
eduardo@simulador: $ mkdir netsim
```

Mude para esse diretório.

```
eduardo@simulador: $ cd netsim
```

Baixe os arquivos necessários para dentro dele.

```
eduardo@simulador: /netsim$
```

Execute os comandos a seguir que irá descompactar os arquivos para um diretório de nome semelhante.

```
eduardo@simulador: /netsim$ tar vfxz tcl8.4.5-src.tar.gz
```

```
eduardo@simulador: /netsim$ tar vfxz tk8.4.5-src.tar.gz
```

```
eduardo@simulador: /netsim$ tar vfxz otcl-1.8.tar.gz
```

```
eduardo@simulador: /netsim$ tar vfxz tclcl-src-1.15.tar.gz
```

```
eduardo@simulador: /netsim$ tar vfxz ns-src-2.27.tar.gz
```

A seguir a sequência de comandos que irá verificar se o sistema possui as bibliotecas necessárias para a instalação (./configure) e configurar a compilação que irá se suceder no próximo comando. Esses comandos devem ser executados para todos os diretórios originários da descompactação dos arquivos, um detalhe importante é a ordem, existe uma relação de dependência entre eles, a ordem errada implicará em um erro de compilação.

1) A biblioteca Tcl:

```
eduardo@simulador: /netsim$ cd tcl8.4.5
```

```
eduardo@simulador: /netsim/tcl8.4.5$ cd unix
```

```
eduardo@simulador: /netsim/tcl8.4.5/unix$ ./configure
```

```
eduardo@simulador: /netsim/tcl8.4.5/unix$ make
```

```
eduardo@simulador:/home/eduardo/netsim/tcl8.4.5/unix make install  
( Comando para instalar o tcl, execute como root )
```

```
eduardo@simulador: /netsim/tcl8.4.5/unix$ cd ../..
```

2) A biblioteca Tk:

Observação:

É necessários os cabeçalhos (.h's) da biblioteca X11 (libx11-dev) e é claro, a própria biblioteca X11 (libx11)

É Aconselhável instalar os pacotes.

```
eduardo@simulador: /netsim$ cd tk8.4.5
```

```
eduardo@simulador: /netsim/tk8.4.5$ cd unix
```

```
eduardo@simulador: /netsim/tk8.4.5/unix$ ./configure
```

```
eduardo@simulador: /netsim/tk8.4.5/unix$ make
```

```
eduardo@simulador:/home/eduardo/netsim/tk8.4.5/unix make install  
( Comando para instalar o tk, execute como root )
```

```
eduardo@simulador: /netsim/tk8.4.5/unix$ cd ../..
```

### 3) A biblioteca OTcl:

```
eduardo@simulador: /netsim$ cd otcl-1.8
```

```
eduardo@simulador: /netsim/otcl-1.8$ ./configure
```

```
eduardo@simulador: /netsim/otcl-1.8$ make
```

```
eduardo@simulador: /netsim/otcl-1.8$ cd ..
```

### 4) A biblioteca Tclcl:

```
eduardo@simulador: /netsim$ cd tclcl-1.15
```

```
eduardo@simulador: /netsim/tclcl-1.15$ ./configure
```

```
eduardo@simulador: /netsim/tclcl-1.15$ make
```

```
eduardo@simulador: /netsim/tclcl-1.15$ cd ..
```

3) Finalmente o NS:

Para todos usuários terem acesso ao simulador como root proceda:

```
eduardo@simulador:/home/eduardo/netsim/ns-2.27 make install
```

## A.2 Tutorial

Tutorial simples e rápido para quem está iniciando

Define um esquema para o script em alguns passos:

1. Criar instância de simulação;
2. Defina o arquivo que será a saída ou resultado da simulação;
3. Crie os nós que irão participar da simulação;
4. Defina a topologia;
5. Defina Agentes e Aplicações;
6. Relacione Agentes e Aplicações;
7. Escalone os Eventos;
8. Inicie a Simulação.

Para um melhor entendimento, abaixo o tutorial segue com um exemplo de acompanhamento.

Em seu console, edite ( com seu editor preferido ) um arquivo chamado "tutor.tcl". Note que o nome não importa muito, escolha outro se preferir.

```
set ns [new Simulator] # Criando a instancia do simulador
puts "Rodando pela primeira vez o simulador."
# O Comando "Puts"escreve seu parâmetro na saída padrão.
```

```
set tr [open trace.out w]
set trnam [open trace.nam w]
# Comando informando o arquivo de saída.
```

```
puts "Criando Trace."
```

```
$ns trace-all $tr
$ns namtrace-all $trnam
# relaciono o arquivo de saída à simulação.
```

```
puts "Criando nós"
```

```
set n0 [$ns node] # Criar um nó chamado "n0"e relaciono à
simulação "ns"
set n1 [$ns node] # Criar um nó chamado "n1"e relaciono à
simulação "ns"
```

```
puts "Definindo Topologia"
```

```
$ns duplex-link $n0 $n1 30Mb 1ms DropTail
# Aqui é definida uma ligação entr os nós dizendo o tipo de ligação, sua largura de
banda e seu delay.
```

```
puts "Definindo Agentes e aplicações"

set udp0 [new Agent/UDP]
set udp1 [new Agent/UDP]
# Defino alguns agentes que vão agir na simulação

set tel0 [new Application/Telnet]
# Defino uma aplicação, no caso "Telnet", que servirá de simulação.

puts "Relacionando Agentes"

$ns attach-agent $n3 $udp0
$ns attach-agent $n4 $udp1
# Relaciono os agentes aos nós

puts "Relacionando Aplicações"

$tel0 attach-agent $udp0
# Relaciono o agente à aplicação

puts "Escalonando"

$ns connect $udp0 $udp1
# Informo quais agentes devem se conectar

$ns at 3.0 "$tel0 start"
# Informo quando será iniciada a aplicação

$ns at 30.0 "exec nam trace.nam"
$ns at 30.0 "exit 0"
```



```
# Informo sobre o término da simulação
```

```
puts "Rodando"
```

```
$ns run
```

```
# Finalmente, inicia-se a simulação.
```

```
# Será criado um arquivo "trace.out" onde pode ser visto como foi o andamento da simulação.
```

```
# Para rodar a simulação utilize:
```

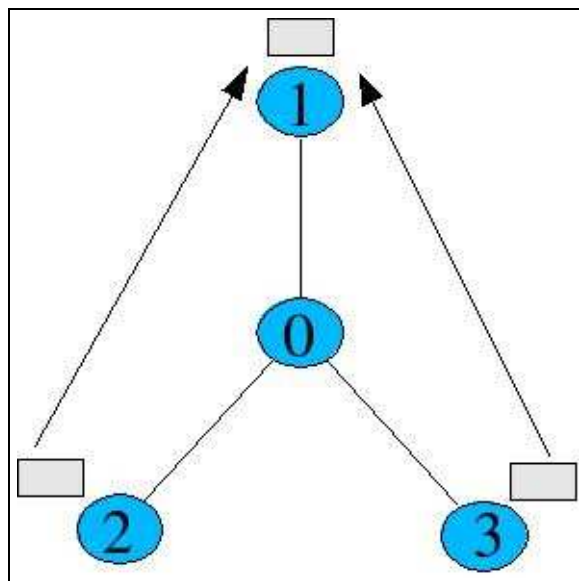
```
# $ ns tutor.tcl
```

# Apêndice B

## Anexo 2 - Exercícios de Simulação

Simulação (1)

Implemente a simulação como na figura.

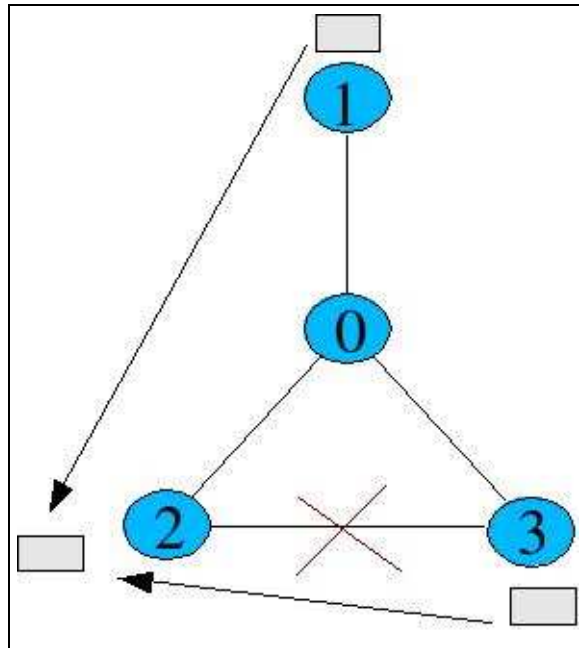


Informações:

- Nós 2 e 3 se conectam com nó 1.
- Utilize Agentes UDP.

## Simulação (2)

Implemente a simulação como descrito na figura.

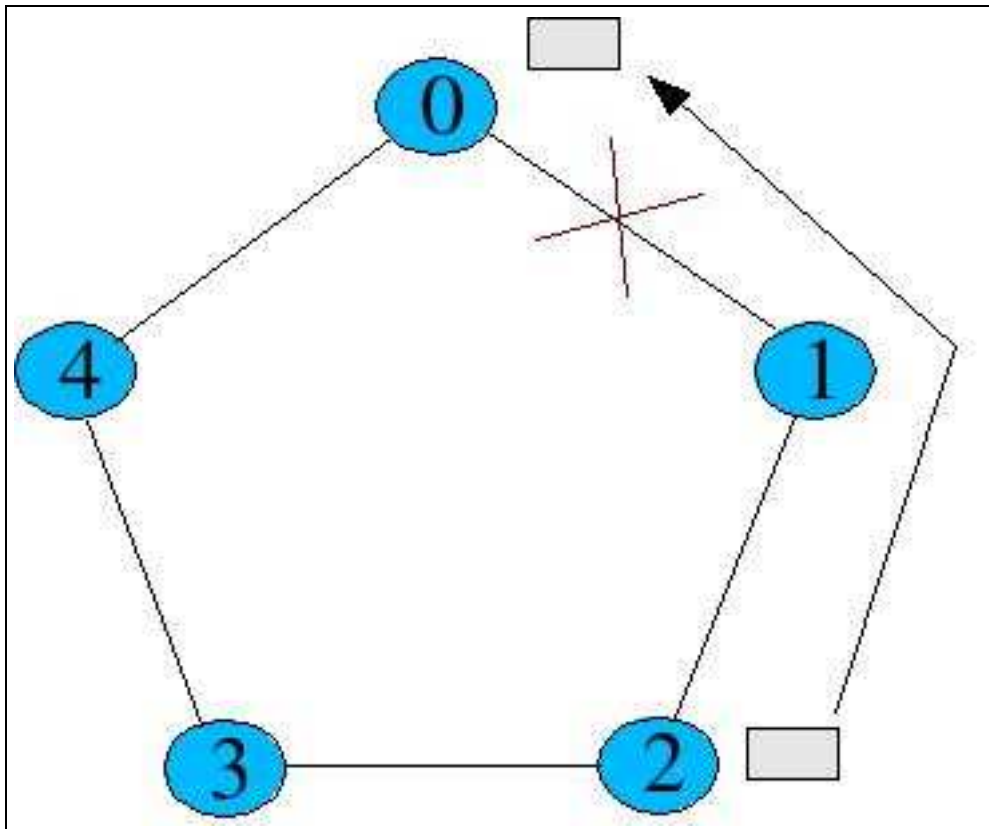


## Informações:

- Nós 1 e 3 se conectam com nó 2.
- Ligação 2-3 se rompe aos 10s
- Utilize Agentes UDP.
- Roteamento deve ser simples.

## Simulação (3)

Implemente a simulação como descrito na figura.

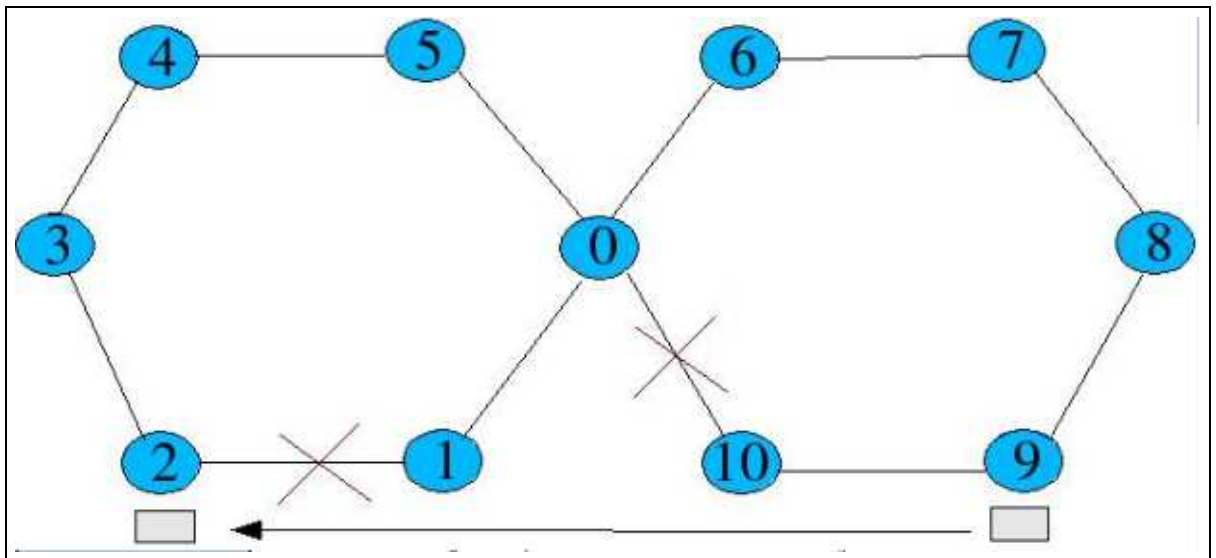


Informações:

- Nó 2 se conecta com nó 0.
- Ligação 0-1 se rompe aos 10s
- Ligação 0-1 se refaz aos 20s
- Utilize Agentes TCP.
- Roteamento deve ser dinâmico.

## Simulação (4)

Implemente a simulação como descrito na figura.



Informações:

- Nó 9 se conecta com nó 2.
- Ligação 2-1 se rompe aos 10s
- Ligação 0-10 se rompe aos 20s
- Ligações se refazem aos 30s
- Simule no mínimo até 40s
- Utilize Agentes UDP e TCP.
- Roteamento deve ser dinâmico.

## Simulação (5)

## Desafio

Implemente a Seguinte simulação:

Uma rede local com 6 computadores, um deles é o gateway da rede que conecta os outros na internet com uma conexão rápida.

Caso a ligação entre o gateway e os outros computadores se desfça, então um outro computador faz uma conexão discada e serve de gateway para os outros.

Simule uma situação em que os 5 computadores, com exceção do gateway, tenham uma conexão Telnet e uma FTP com um computador externo a essa rede.

Informações:

- Os 6 computadores estão através de um HUB.
- Considere o HUB como se fosse um nó.
- A ligação entre os 6 computadores é de 10Mb/s.
- A ligação do gateway e o computador externo é de 100Mb/s.
- A conexão discada é de 4.2kb/s
- Todos computadores iniciaram a tranferência ao mesmo tempo e 2s depois do início da simulação.
- O intervalo entre o envio de cada pacote é de 0.003 s.
- aos 20s a conexão com o gateway se rompe.
- a conexão discada inicia aos 22s.
- considere o delay de todos os nós de 10ms.

Depois de simular, relate o problema dessa rede e cite uma possível solução.

## Simulação (6)

## Projeto

Projete a rede interna de uma futura empresa.

A Empresa terá:

- 23 computadores distribuídos 3 andares.
- 1 servidor http.
- 2 servidores de email.
- 2 gateways.
- Dois pontos para vídeo conferência.

Atenção:

- Total de Computadores: 30
- Não utilize largura de banda desnecessária
- identifique os computadores especiais (servidores, gateways...) dentro do script.

Dica:

Utilize o comando de controle "for" para criar os nós e ligações.

Depois de Implementar a topologia, faça uma simulação envolvendo no mínimo 5 nós (5 conexões: FTP, Telnet).





## Apêndice C

### Anexo 3 - Métodos de avaliação de Mapas Conceituais

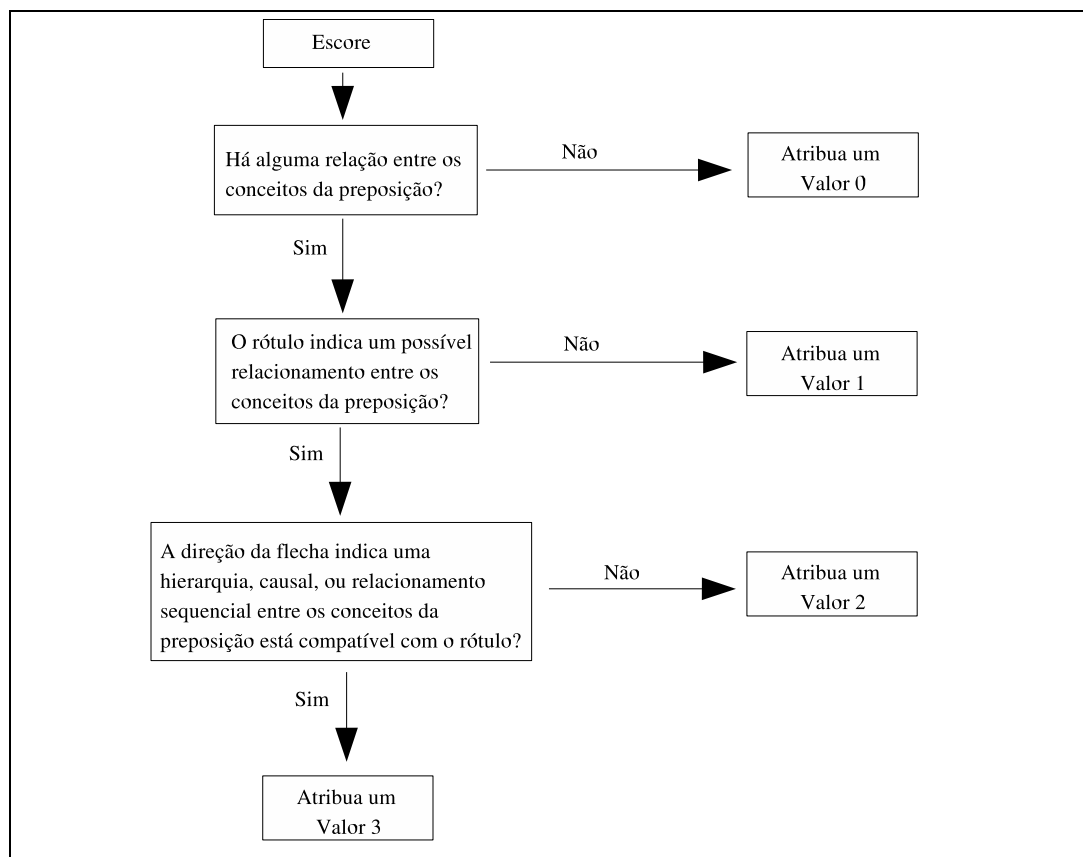


Figura C.1: Método Relacional [11]

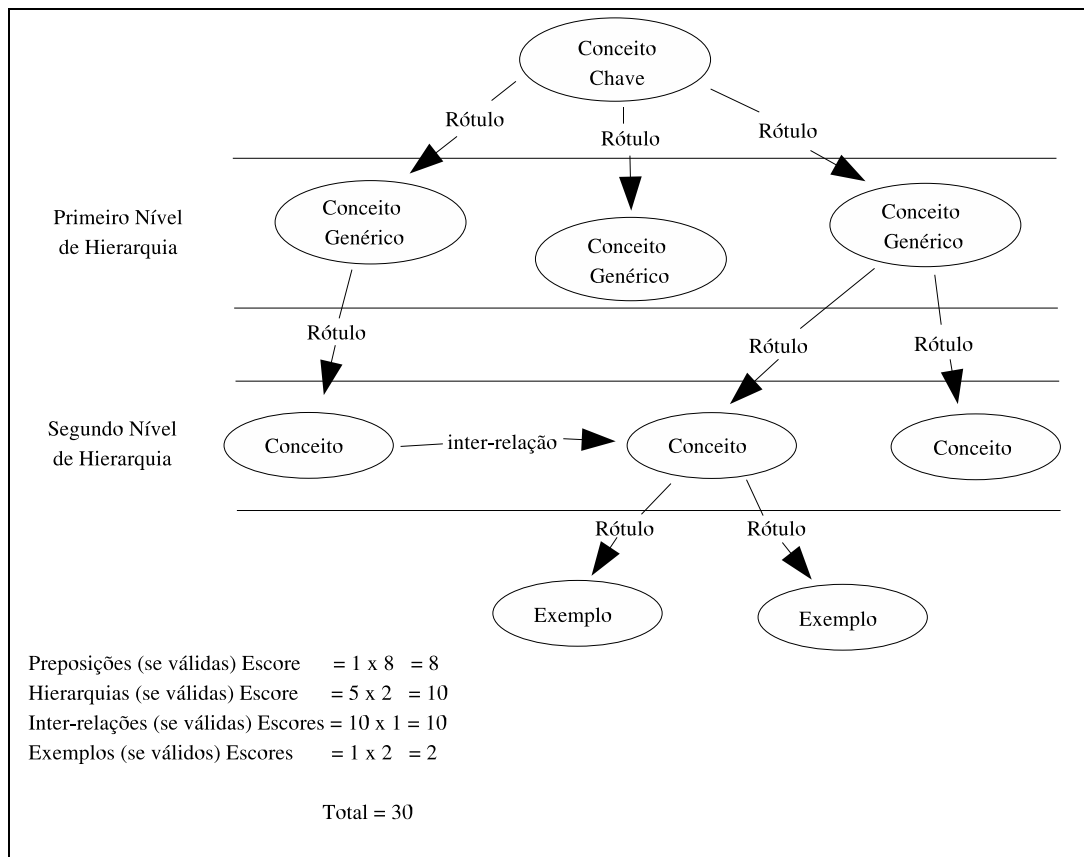


Figura C.2: Método Estrutural [11]