


PROJETO AUTOMÁTICO DE REDES DE TV A CABO

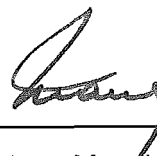
José Carlos Cabrera Zúñiga

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

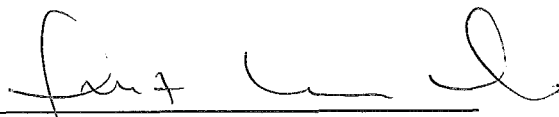
Aprovada por:



Prof. Inês de Castro Dutra, Ph.D.



Prof. Nelson Maculan Filho, D.Habil.



Prof. Luiz Satoru Ochi, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

JUNHO DE 2001

CABRERA ZÚÑIGA, JOSÉ CARLOS

Projeto Automático de Redes de
TV a Cabo [Rio de Janeiro] 2001.

VII, 122p. 29,7 cm (COPPE/UFRJ,
M.Sc., Engenharia de Sistemas e
Computação. 2001)

Tese – Universidade Federal do
Rio de Janeiro, COPPE

1. CATV
 2. Projeto de CATV
 3. Redes de telecomunicações
- I. COPPE/UFRJ II. Título (série)

Agradecimentos

À CAPES, pelo auxílio financeiro.

Eu não poderia ter desenvolvido esta tese sem ajuda de minha professora e orientadora Inês de Castro Dutra que me sugeriu o tema da tese. Gostaria também de agradecer a seu irmão Renato que me forneceu a informação detalhada de como se faz o projeto de redes CATV atualmente (e que tornou a minha vida impossível!). A eles meu infinito agradecimento.

Aos meus professores da COPPE.

Também agradeço a minha família e amigos. Com seu apoio eu me senti tranqüilo para trabalhar.

Por último, agradeço a Deus que me deu a oportunidade de vir estudar nesta universidade.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.).

PROJETO AUTOMÁTICO DE REDES DE TV A CABO

José Carlos Cabrera Zúñiga

Junho/2001

Orientadora: Inês de Castro Dutra

Programa: Engenharia de Sistemas e Computação

Esta tese tem por objetivo o desenvolvimento de algoritmos para projeto automático de redes de TV a cabo (CATV).

Para o desenvolvimento deste trabalho, foram feitas consultas a um especialista em projeto de redes CATV no Rio de Janeiro. Com base nas informações sobre as características precisas do problema definimos sua topologia, estruturas de dados e algoritmos.

Procuramos soluções a problemas similares ao nosso e avaliamos programas de computadores atualmente utilizados para redes CATV. Não encontramos nenhuma ferramenta automática para resolver o projeto das redes e encontramos pouca bibliografia técnica e científica sobre o assunto. Desta forma assumimos que nossas soluções são originais.

Observamos que as diversas soluções existentes para problemas similares têm uma complexidade exponencial. A nossa solução possui a mesma complexidade, mas pelas características dos grafos, ela se reduz drasticamente.

Aplicamos nossos algoritmos a situações da vida real, extraídas diretamente de um mapa da cidade do Rio de Janeiro e comprovamos que nossas soluções são equivalentes ou melhores que os resultados encontrados manualmente.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.).

CABLE TV NETWORK AUTOMATIC DESIGN

José Carlos Cabrera Zúñiga

June/2001

Advisor: Inês de Castro Dutra

Department: Systems Engineering and Computer Science

In this work we develop a formal description of cable TV (CATV) network planning, and present the algorithms and data structures necessary to do automatic planning.

In order to obtain precise information and develop special heuristics suitable to solve the problem, we interviewed a cable TV network planning expert.

Most literature and software available on CATV network planning is commercial. To the best of our knowledge, our algorithms are a first step on making CATV network planning fully automatic.

We applied our algorithms to actual regions of Rio de Janeiro city, and obtained solutions better or equivalent to a manual planning.

Conteúdo

<i>Agradecimentos</i>	III
<i>Conteúdo</i>	VI
<i>Introdução</i>	1
Apresentação	1
Motivação e Objetivos	2
Contribuições	3
Organização da Tese	3
<i>Capítulo 2</i>	4
<i>Revisão Bibliográfica e Metodologia</i>	4
<i>Capítulo 3</i>	7
<i>Descrição Detalhada do Problema</i>	7
Componentes Ativos	9
Amplificador	9
Componentes Passivos e a Atenuação	10
Componentes Passivos	11
O Processo de Instalação dos Distribuidores num Poste	28
Mínimo Requerimento de Sinal de um Poste	29
A Atenuação e a Freqüência do Sinal	31
A Entrada de Dados do Problema	32
Observações	32
<i>Capítulo 4</i>	34
<i>Modelagem</i>	34
Estruturas de Dados Básicas	35
Estruturas de Dados para os Distribuidores de Sinal	35
Representação do Grafo $G = (P, L)$	41
Representação de um Caminho de Postes	42
Uma Estrutura de Dados Auxiliar	44
Requerimentos Mínimos para Ligar um Poste Q a outro P Usando Apenas Cabo	45
Algoritmos Principais	46
O algoritmo do Melhor Caminho a partir de um Poste Inicial (<i>backbone</i>)	46
Indicando um Caminho PLUSCS no Grafo $G = (P, L)$	51
Percorrendo o Backbone	52

Escolha de um Ligador de Cabo para Instalar um novo Distribuidor Binário de Sinal	54
Heurística PP1	56
Heurística PP2	58
Melhor Conector de Postes para uma Intensidade de Sinal	59
O Melhor Caminho PLUSCS para um Distribuidor Binário de Sinal num Conjunto de Conectores de Postes	60
O Melhor Distribuidor Binário de Sinal num Conjunto de Conectores de Postes, para Instalar num Ligador de Cabo de Outro Distribuidor de Sinal	62
Procura de Postes Ilhados no Grafo $G = (P, L)$	63
Ligando os Postes Vizinhos de um Poste	64
Ligando os Postes de um Grafo a partir de um Backbone Inicial (algoritmo de ligação de CATV chamado JCCZ)	72
Postes Obrigatórios a Conectar	76
Algoritmo de Correção	83
Exemplos	84
Melhoras ao ALGORITMO de JCCZ	106
<i>Capítulo 5</i>	<i>108</i>
<i>Outras Formas de Modelagem</i>	<i>108</i>
Usando Restrições e Constraint Logic Programming (CLP)	108
Usando o Minimal Spanning Tree	111
<i>Capítulo 6</i>	<i>112</i>
<i>Conclusões</i>	<i>112</i>
<i>Referências Bibliográficas</i>	<i>114</i>

Introdução

Apresentação

Projeto de redes de telecomunicações abrange uma gama muito grande de aplicações, entre elas, redes de telefonia, sistemas de satélite, problemas de localização e transmissão de sinais de televisão a cabo [13]. Uma rede de telecomunicações normalmente é representada por um grafo onde as arestas correspondem às linhas de transmissão que carregam a informação (sinal) e os nós correspondem aos usuários. Associado a cada aresta pode existir um peso para a ligação entre dois usuários. As soluções para resolver o problema se baseiam em particionamentos do grafo de forma hierárquica, para que a complexidade do projeto seja reduzida. Vários algoritmos e métodos de modelagem matemática inclusive utilizando funções heurísticas, busca tabu, simulated annealing e meta-heurísticas têm sido utilizados para resolver problemas nestas áreas [13]. Nosso trabalho concentra-se em uma área de redes de telecomunicações que é pouco divulgada cientificamente: o projeto automático de transmissão de sinais de televisão por cabo. A maioria da literatura encontrada sobre o problema é proveniente de empresas de telecomunicações que não divulgam algoritmos e estratégias de projeto das suas redes.

O problema geral de transmissão de sinais de TV por cabo (CATV) envolve vários tópicos, entre eles o projeto da rede que consiste em atender o máximo número de residências tentando minorar o custo de equipamentos. Este tipo de problema é de complexidade exponencial, porém utilizando-se a idéia do particionamento do grafo

mencionado anteriormente pode-se encontrar heurísticas interessantes para resolver o problema inteiro.

Uma abordagem geralmente utilizada pelos projetistas de redes CATV considera regiões de *residências*. Cada uma destas regiões é chamada *nó*. O nó é dividido em sub-regiões chamadas *células*. Cada uma destas células vai se dividir em *microcélulas*. Os postes da rua servem como base para a instalação do cabo. Alguns postes possuem um grupo de residências a serem atendidas. O número de residências por poste não necessariamente é constante. Este tipo de abordagem é chamada de HFC (Hybrid Fiber Coaxial) [12].

Nesta tese propomos uma solução para o problema do projeto automático de transmissão de sinal de TV a cabo utilizando heurísticas desenvolvidas especificamente através das características de dados reais de uma região da cidade do Rio de Janeiro. Embora estas heurísticas tenham sido desenvolvidas em função das características destes dados reais, estas últimas são similares para outras regiões, o que generaliza a nossa abordagem.

Nesta tese desenvolvemos algoritmos para resolver o problema de transmissão de sinais para uma microcélula. Estes mesmos algoritmos podem ser utilizados e/ou estendidos para resolver o problema geral que considera m residências em um nó. A topologia da microcélula é simples porque o número de residências é reduzido e na maioria dos casos é comum que um poste receba sinal de um único poste e envie sinal a um único poste. Raramente teremos casos em que postes enviam sinal a mais de 1 poste.

Motivação e Objetivos

Todas as ferramentas utilizadas para CATV são comerciais e não resolvem o problema de forma totalmente automática [13]. Desta forma, há uma carência muito grande de ferramentas de baixo custo e totalmente automáticas para a solução deste problema.

Procuramos então algoritmos que possam resolver este problema de forma totalmente automática. Apresentamos assim um novo algoritmo de ligação de CATV

chamado **JCCZ**. Procuraremos encontrar algumas expressões matemáticas que sirvam como base para a criação dos algoritmos e heurísticas.

Contribuições

Atualmente, de nosso conhecimento, as soluções para este problema de transmissão de sinal por cabo na microcélula se enquadram em três categorias:

- 1) Solução totalmente manual.
- 2) Solução manual com auxílio de programas primitivos de alto custo (ex. LODE DATA e MAPINFO [19][18]).
- 3) Solução semi-automática com software de altíssimo custo [19];

Neste trabalho, além de modelar o problema formalmente, desenvolvemos algoritmos simples e com potencial para resolver o problema de forma totalmente automática. Neste sentido, nossas contribuições são:

- 1) Modelagem matemática e topológica, e definição das características do problema.
- 2) Desenvolvimento de algoritmos eficazes para a solução automática do problema.
- 3) Solução de baixo custo financeiro para os usuários potenciais de sistemas deste tipo.

Organização da Tese

Este trabalho está dividido em seis capítulos. O capítulo 2 apresenta uma revisão bibliográfica na área de planejamento de CATV. O capítulo 3 descreve formalmente o problema de CATV descrevendo seus aspectos topológicos e físicos. No capítulo 4, apresentamos os algoritmos e heurísticas baseados na modelagem apresentada no capítulo 3, limitações do algoritmo JCCZ, estimativas de complexidade dos algoritmos e exemplos de aplicação dos algoritmos num caso real. No capítulo 5, apresentamos formas alternativas de modelagem para o problema de CATV. Finalmente, no capítulo 6, apresentamos nossas conclusões e perspectivas de trabalhos futuros.

Revisão Bibliográfica e Metodologia

Há poucas referências bibliográficas sobre o problema de CATV. A maior parte delas é comercial e com pouco conteúdo técnico e científico. A literatura existente sobre redes de telecomunicações em geral, aborda problemas diferentes dos abordados nesta tese. Desta forma buscou-se informações com um especialista do problema [11]. Foram feitas várias entrevistas para a obtenção de informações relevantes e bibliografia adequada sobre os equipamentos [3], [7], [8], [12]. Esta interação com o especialista foi bastante importante, porque obtivemos informações precisas sobre as características do problema o que facilitou a criação de algoritmos e heurísticas mais adequados.

A partir das informações obtidas do especialista, começamos a fazer o planejamento topológico do problema, fazendo definições baseadas no livro de topologia de Schwartz [15]. As estruturas de dados do problema foram construídas com base nas estruturas de dados descritas em [16] e [17]. Para a modelagem do problema observamos que teríamos que usar algoritmos distribuídos para lidar com determinadas situações tais como a procura de postes *ilhados* [2].

Paralelamente, pesquisamos soluções para problemas similares de planejamento de redes de telecomunicações. Em [13], [14], [9] e [1] encontramos diferentes abordagens que utilizam princípios de programação com restrições e programação linear, além de algoritmos em grafos.

Todos os artigos encontrados na literatura se concentram em problemas de localização, fluxo máximo (max flow) e qualidade de sinal e a maior parte deles enfoca redes de telefonia e de computadores. Poucos artigos abordam o problema do planejamento e projeto da rede propriamente dito e se fala muito pouco sobre o problema de planejamento e projeto de CATV [10].

Pesquisamos também programas existentes para projeto de CATV. LODE DATA [11] é um software bastante utilizado em projeto de CATV e é vendido em dois módulos. O primeiro módulo é o *design assistant*, cujo custo chega a US\$ 4,000 dólares, e que tem chave de hardware. Este módulo aceita como entrada as distâncias entre postes e número de assinantes por poste. A interface é no estilo Microsoft Excel e exibe como saída os dispositivos instalados pelo projetista em cada poste e um relatório de custo. O usuário deve definir manualmente a localização do amplificador inicial. A partir daí, o software coloca amplificadores automaticamente quando o sinal fica muito fraco para atender as residências. Este módulo não possui interface gráfica.

O segundo módulo do LODE DATA é o *draft assistant* que possui interface com AUTOCAD para pegar a entrada gráfica do mapa. O usuário desenha o mapa, mas o projeto ainda é feito de forma primitiva. As especificações dizem que este módulo faz também o roteamento automático, mas não é conhecido nenhum caso de projetos com resultados bem sucedidos.

Quando o amplificador inicial é colocado num poste e o sinal não consegue alcançar o caminho inteiro da rede, o usuário precisa refazer todo o cabeamento, ou seja, este software não implementa *backtracking*.

Um outro software conhecido para projeto de CATV é o MAPINFO [18]. Este é similar ao AUTOCAD e consegue ler o mapa de entrada. O usuário realiza todo o cabeamento selecionando os postes manualmente para colocar os taps. Quando o sinal é insuficiente para alcançar as residências de um determinado poste, o software automaticamente inclui um novo amplificador.

Este tipo de software impõe uma série de limitações ao projeto de CATV. Primeiro, as soluções encontradas com a utilização deste softwares é feita de forma manual assistida por computador o que pode gerar projetos custosos e com equipamentos redundantes. Além disto, como o usuário deve controlar o processo de instalação do amplificador inicial e da instalação dos taps, o tamanho do problema não pode ser muito grande. Uma outra limitação destes softwares é que se o tamanho do problema for de médio a grande, o projetista pode cometer erros significativos.

Desta forma, este trabalho se propõe a contornar as limitações dos softwares existentes, através de uma abordagem original totalmente automática para a solução do projeto de redes CATV.

No capítulo seguinte apresentaremos a topologia do problema e suas características principais.

Descrição Detalhada do Problema

Para realizar a transmissão de sinais de CATV, as empresas de CATV utilizam a infra-estrutura já existente: os postes de rua e sua instalação elétrica. Inicialmente faz-se um levantamento do número de residências por poste e um levantamento da topologia da rede elétrica. Com estas informações normalmente o problema é resolvido manualmente com a escolha de áreas de maior densidade demográfica para instalação dos amplificadores de sinal.

Nossa abordagem para a solução deste problema classifica os postes em dois grupos distintos e principais: (1) postes que não possuem residências, mas recebem cabo e (2) postes que possuem residências e recebem cabo. Por residências entendemos casas. No caso dos prédios, basta traduzi-los para um grafo de postes.

Dependendo das condições de vizinhança e do tipo de poste poderemos ter que instalar equipamentos de distribuição de sinal, normalmente chamados de *taps*, *divisores*, *acopladores direcionais* e *amplificadores de sinal* que veremos mais a frente.

O objetivo da solução é atender o número máximo de residências de um conjunto P de postes que compõem a microcélula ao mesmo tempo que minimizando a quantidade de equipamentos. Os equipamentos a serem utilizados são os dispositivos de distribuição de sinal, um único *amplificador* e o cabo.

Para descrever o problema e sua modelagem usaremos as condições e definições a seguir.

Definição. A *intensidade de um sinal de CATV* é uma grandeza física definida pela tensão (não pela potência), que vai nos permitir realizar medições em uma rede de CATV. É também conhecida como *tensão de rádio frequência*. A unidade de medida da intensidade de sinal é o decibel-milivolt (dBm).

A intensidade de um sinal de TV vai nos permitir projetar a distribuição de sinal, mantendo um nível mínimo de qualidade¹.

Definição. Identificaremos por $|P|$ o *número de postes* de um conjunto de postes P .

Definição. Identificaremos por $\|P\|$ o *número de residências* que podem ser atendidas por um conjunto de postes P .

Definição. No conjunto P se define a relação L como:

$$L(P, Q) \Leftrightarrow Q \text{ pode ser ligado a } P \text{ ou vice-versa;}$$

onde P, Q são postes de P .

Definição. Para todo $P \in P$, o conjunto

$$V(P) = \{ Q \in P / L(P, Q) \}$$

será chamado *conjunto dos postes vizinhos de P* ou simplesmente *vizinhos de P*.

Restrição do Problema. Seja P um conjunto de postes a ligar. Para todo $P \in P$, $|V(P)| \leq 6$.

¹ Segundo a lei 8.977 de 06 de janeiro de 1995, regulamentado pelo decreto nº 1.718 de 28 de dezembro de 1995.

Definição. Sejam $P, Q \in \mathcal{P}$, $P \neq Q$. Um *caminho de P a Q* segundo L é uma sucessão de postes P_1, P_2, \dots, P_m onde

a) $P_1 = P$ e $P_m = Q$, $m \geq 2$, m inteiro e

b) Para todo P_i, P_{i+1} se cumpre $L(P_i, P_{i+1})$, $1 \leq i \leq m-1$, $m > 2$.

Definição. Um conjunto de postes \mathcal{P} é *conexo* se para todo par (P, Q) , com $P \neq Q$, do produto cartesiano \mathcal{P}^2 , existe um caminho segundo L de P a Q .

Condição do Problema. Aceitaremos que o conjunto de postes \mathcal{P} a analisar é conexo.

Precisamos conhecer os dispositivos para distribuição de sinal que utilizaremos. Além disso, mostraremos definições para a sua utilização. Estes componentes podem ser ativos ou passivos.

Componentes Ativos

São equipamentos cuja tarefa básica é melhorar o sinal de chegada, precisando de eletricidade para tal. Nesta tese, utilizaremos apenas o amplificador.

Amplificador

É um equipamento que aumenta a intensidade do sinal de entrada, preservando sua qualidade. Possui uma única entrada e dependendo do tipo pode ter até 4 saídas. Nesta tese só utilizaremos amplificadores de uma saída. A seguinte figura apresenta o esquema de um amplificador de uma saída.

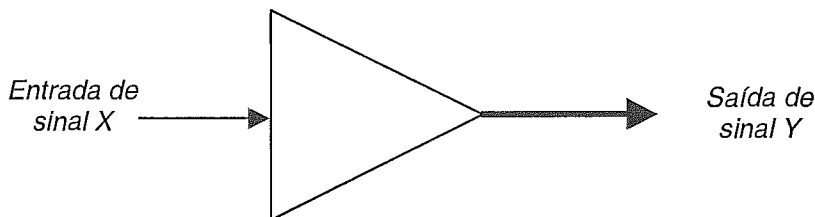


Figura 3.1. Esquema de um amplificador.

Assim, se X é a intensidade do sinal que chega ao amplificador e Y é a intensidade na sua saída, então:

$$X < Y$$

Observações.

1. X tem um valor mínimo especificado pelo fabricante para que o amplificador possa trabalhar. Y também tem um valor segundo o fabricante.
2. Geralmente, um sinal de 47 dBm em Y consegue atender 80 residências manualmente (ou doze postes aproximadamente).

Condição do Problema. No conjunto P da microcélula, somente um único poste P cumprirá a proposição: *no poste se instalou um amplificador.*

Existem limitações físicas para a instalação do amplificador em uma microcélula P . Assim, somente será possível instalar o amplificador em alguns dos postes de um subconjunto H de P . Usando o algoritmo de JCCZ que veremos mais a frente (pág 71), vamos precisar gerar redes de CATV a partir de cada um dos postes de H retornando como reposta a rede que tenha o maior número de residências.

Componentes Passivos e a Atenuação

A finalidade de um *componente passivo* é transmitir um sinal. Quando o sinal de TV passa por um destes componentes, o sinal que se tem nas saídas do componente é de *menor tensão de rádio freqüência*. Este fenômeno é conhecido como *atenuação*.

A *atenuação* é uma grandeza física medida em decibel-milivolt (dBm) que indica quantitativamente se um valor diminui durante um processo qualquer.

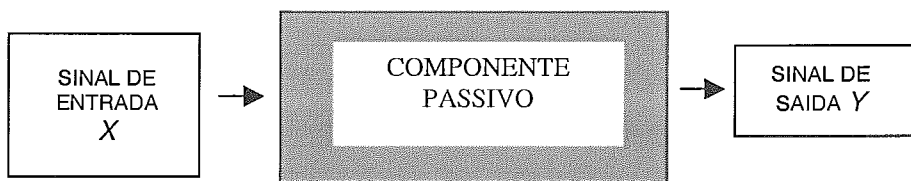


Figura 3.2. Esquema de um componente passivo.

Na figura 3.2, o componente passivo recebe um sinal de tensão de rádio frequência X . Assim, se o sinal de saída tem uma intensidade Y então:

$$X > Y$$

Componentes Passivos

Os componentes passivos são construídos para atenuar o mínimo possível o sinal em um determinado ponto.

Os componentes passivos que vamos utilizar são:

Cabo. Conecta a distribuição. Só usaremos cabo coaxial CC5¹. Vamos representá-lo por:

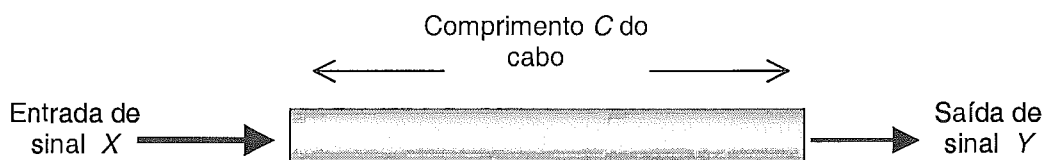


Figura 3.3. Representação do Cabo CC5.

Nesta figura, X e Y são as intensidades de sinal de entrada e saída do cabo, respectivamente. Se A é a atenuação do cabo por metro, então por construção:

¹ Em nossa modelagem para utilizar outro cabo que não seja CC5, basta modificar o valor da atenuação.

$$Y = X - A \times C$$

O valor de A é fornecido pelo fabricante. Um exemplo de atenuação é $A = 0,0512$ dBm/m.

Divisor de 2 saídas (*splitter 2 way*). É um dispositivo que tem uma entrada de sinal e duas saídas. Em cada saída se consegue a mesma intensidade atenuada de sinal. Assim, o divisor divide o sinal de entrada em dois caminhos. Observemos a seguinte figura:

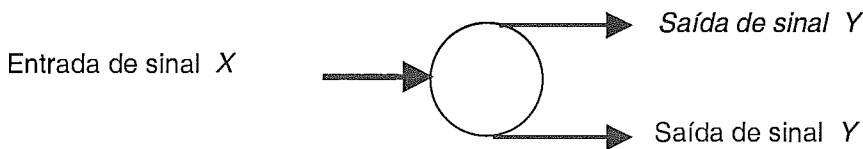


Figura 3.4. Representação de um divisor.

Nesta figura, se a intensidade na entrada é X e a intensidade em cada saída é Y então

$$Y = X - \text{Atenuação}$$

onde *Atenuação* é a atenuação do divisor (que é a atenuação em cada uma de suas saídas). Como exemplo, um valor para *Atenuação* é 3,9 dBm. O valor da atenuação do divisor é subministrada pelo fabricante. Representaremos este dispositivo com



Divisor de 3 saídas não balanceado (*splitter 3 way not balanced*). É um dispositivo que tem uma entrada de sinal e três saídas. Por construção uma das saídas tem sinal maior que os outros dois que são iguais. Vamos representá-lo na seguinte figura:

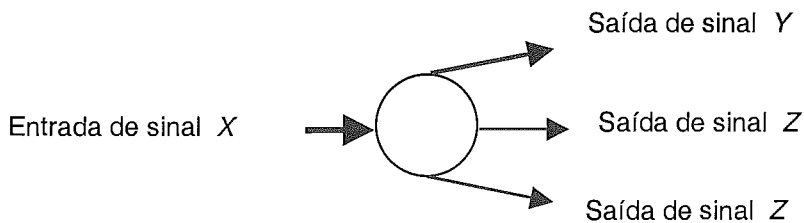


Figura 3.5. Representação de um divisor de três saídas.

onde X é a intensidade do sinal de entrada, Y é a intensidade de sinal na saída mais forte e Z é a intensidade de sinal nas outras duas saídas mais fracas. As equações que relacionam estas variáveis são:

$$Y = X - \text{Atenuação}_Y$$

$$Z = X - \text{Atenuação}_Z$$

onde Atenuação_Y e Atenuação_Z são as atenuações para Y e Z respectivamente. Além disso, por construção,

$$\text{Atenuação}_Y < \text{Atenuação}_Z$$

e portanto, $Y > Z$.

É conhecido que um divisor de três saídas é equivalente a dois divisores iguais de duas saídas, pois pode-se obter

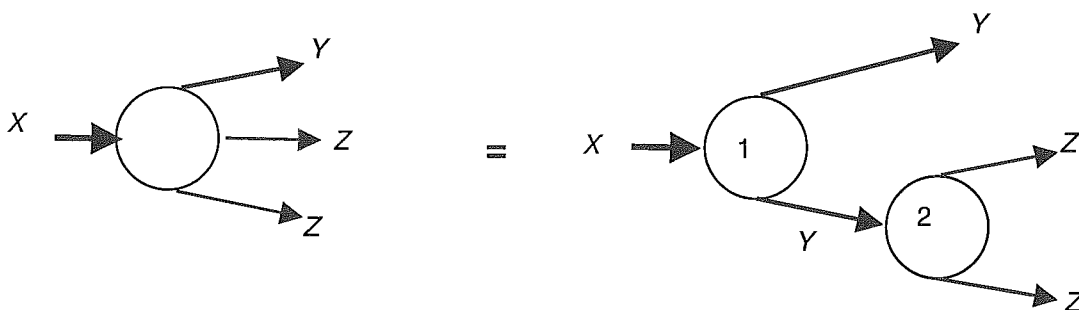


Figura 3.6. Equivalência de um divisor de três saídas com dois divisores de duas saídas.

onde $Atenuação_Y$ é a atenuação para a saída de intensidade Y no divisor 1 e $Atenuação_Z$ é a atenuação para as saídas do divisor 2. Portanto, temos:

$$Atenuação_Z = 2 \times Atenuação_Y, \text{ e}$$

$$Z = X - Atenuação_Z$$

Acoplador direcional (*directional coupler*). Tem uma entrada de sinal e duas saídas, uma bem mais atenuada que a outra, permitindo um direcionamento de sinal mais forte para um lado, quando for necessário. Existem três tipos diferentes de acopladores direcionais. A seguinte tabela é uma mostra de informação subministrada pelo fabricante sobre acopladores direcionais.



Acoplador direcional	Atenuação nas saídas		Representação
	Sinal Fraco	Sinal Forte	
Direct Coupler 8	8 dBm	1,5 dBm	
Direct Coupler 12	12 dBm	1,2 dBm	

Tabela 3.1. Típicos valores de atenuação para os acopladores direcionais

Observação: A atenuação produzida por um divisor ou acoplador direcional é superior à atenuação do cabo. Por esta razão precisamos procurar minimizar o número destes dispositivos na rede.

Tap. Equipamento de distribuição de sinal às residências. Além da entrada e saída para o cabo CC5, tem saídas atenuadas para alimentar as residências. A residência precisa de uma intensidade R_s na saída do tap correspondente para que possa obter um sinal de boa qualidade (definido pela lei de CATV). A residência precisa de uma das saídas do tap e uma saída do tap só pode atender a uma residência. Existem três tipos de tap: de 2, 4 e 8 saídas para residências. Na seguinte figura vamos mostrar um esquema de um tap de S saídas:

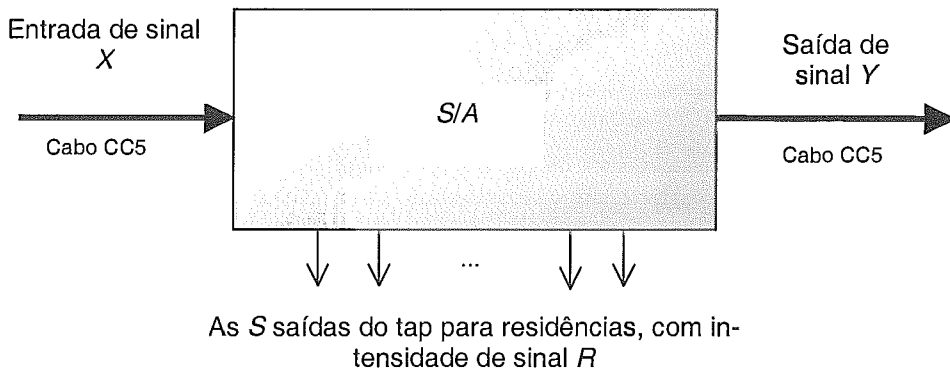


Figura 3.7. Representação de um tap.

Nesta figura, S/A indica um tap de S saídas ($S = 2, 4, 8$). A é um valor que representa a atenuação para cada uma das saídas para residências do tap. Os valores que A pode assumir para cada tipo de tap são proporcionados pelo fabricante. X é a intensidade do sinal de entrada do tap e Y a intensidade da saída para o cabo CC5. R é a intensidade do sinal que sai do tap para a residência.

Por construção temos a relação:

$$X - A = R$$

Quando na saída do tap o sinal Y é maior que zero, este se calcula com a fórmula

$$Y = X - \Delta_{S/A}$$

onde $\Delta_{S/A}$ é a atenuação do sinal de entrada ao sair do tap pela saída de Y . Os valores de $\Delta_{S/A}$ para cada tipo de tap são proporcionados pelo fabricante. Nas seguintes tabelas podemos ver exemplos destes dados [11]. Estas tabelas foram construídas convenientemente com os taps ordenados de maior a menor atenuação para residências, pois nossos algoritmos precisam desta ordenação para escolher o tap adequado.

<i>índice</i>	A	$\Delta_{2/A}$
1	26	0,8
2	23	0,8
3	20	0,9
4	17	1,0
5	14	1,1
6	11	1,4
7	8	1,6
8	4	gb_LNP

Tabela 3.2. Típicos valores de atenuação para os taps de duas saídas. O último não tem continuação para cabo CC5 e a linha não prossegue.

<i>índice</i>	A	$\Delta_{4/A}$
1	26	0,8
2	23	0,8
3	20	0,9
4	17	1,0
5	14	1,1
6	11	1,6
7	8	gb_LNP
8	4	

Tabela 3.3. Típicos valores de atenuação para os taps de quatro saídas. Não existe tap de 4 saídas com atenuação de 4 dBm para residência. O último não tem continuação para cabo CC5 e a linha não prossegue.

<i>índice</i>	<i>A</i>	$\Delta_{8/A}$
1	26	0,8
2	23	0,8
3	20	0,9
4	17	1,0
5	14	1,6
6	11	gb_LNP
7	8	
8	4	

Tabela 3.4. Típicos valores de atenuação para os taps de oito saídas. Não existe tap de 8 saídas com atenuações de 8 e 4 dBm para residências. O último não tem continuação para cabo CC5 e a linha não prossegue.

Nestas tabelas, $A \in \{26, 23, 20, 17, 14, 11, 8, 4\}$. Uma característica relativa aos últimos valores de atenuação dos taps é que são terminados¹, ou seja não têm saída para cabo CC5 e a linha não prossegue (gb_LNP = linha não prossegue). Chamaremos estes taps de *terminais* e assumiremos que o sinal na saída CC5 de um tap terminal é zero.

Definição. Cada dispositivo tap, divisor ou acoplador direcional será chamado distribuidor de sinal (ou *dispositivo de distribuição de sinal*). Em alguns casos somente utilizaremos a palavra distribuidor para nos referir a qualquer um destes dispositivos.

Definição. Os divisores e os acopladores direcionais serão chamados *distribuidores binários de sinal*.

Observação. Assumiremos que os distribuidores binários de sinal formam uma lista.

Condição do Problema. Os distribuidores e o cabo se ligam conectando uma saída de sinal de um à entrada de sinal do outro.

Condição do Problema. Distribuidores somente podem ser instalados em postes.

¹ Existe um casamento de impedância característica Z_0 da linha de transmissão, com o objetivo de tornar a linha infinita e evitar a reflexão do sinal.

Condição do Problema. É dado do problema o número de residências em cada poste.

Para atender as residências de um poste vamos assumir dois casos distintos: (1) número de residências 2, 4 e 8; (2) número de residências qualquer.

Para escolher o tap adequado para uma intensidade de sinal X e um número de residências $S = 2, 4, 8$, vamos percorrer a tabela respectiva começando pela linha de menor índice. O primeiro tap encontrado na tabela cujo valor de A satisfaz:

$$X - A \geq R_s - Tol$$

onde R_s é a intensidade que normalmente deve receber uma residência e Tol é a tolerância da mesma, será o tap de S saídas escolhido. Se depois de passar pela última linha da tabela não encontramos aquele valor de A , então diremos que não existe tap de S saídas para aquela intensidade de sinal X .

Exemplo. Vamos supor que precisamos procurar um tap de 8 saídas para uma intensidade de sinal $X = 37$ dBm. Assumindo que $R_s = 17$ dBm e $Tol = 0$, então na tabela dos valores de atenuação para os taps de oito saídas (tabela 3.4), a linha 3 que tem o valor $A = 20$ indica o valor de atenuação do tap adequado, pois $37 - 20 = 17$. Observamos que linha 2 não pode ser solução porque $37 - 23 = 14 < R_s$. Se usássemos o tap da linha 4, aumentaríamos sem necessidade a atenuação na saída para o cabo CC5.

Vamos construir uma versão mais detalhada deste algoritmo para um tap de duas saídas.

O tipo de dado que representará um tap deve ter os seguintes campos:

- Saídas** ; que indicará o número de saídas 8 ou 4 ou 2 do tap,
- A** ; que indica a atenuação de cada saída para a residência,
- D_SA** ; que indica a atenuação $\Delta_{S/A}$.

Assim, para escolher o tap adequado para uma intensidade de sinal X e um número de residências $S = 2$, vamos supor que temos o array unidimensional TAP2[], cujos elementos possuem os campos mencionados acima, e são inicializados com os dados de todos os taps de 2 existentes (ou seja, vamos representar a tabela 3.2). Desta forma:

```

TAP2[1].Saídas = 2, TAP2[1].A = 26, TAP2[1].D_SA = 0.8
TAP2[2].Saídas = 2, TAP2[2].A = 23, TAP2[2].D_SA = 0.8
      ⋮
TAP2[8].Saídas = 2, TAP2[8].A = 4, TAP2[8].D_SA = gb_LNP

```

Procuraremos dentre todos os tipos de tap de 2 que existem, aquele cujo campo *A* seja o primeiro que satisfaz:

$$(X - \text{TAP2}[i].A) \geq \text{gb_Rs} - \text{gb_TOL}$$

onde **gb_Rs** representa a variável *Rs* e **gb_TOL** representa a tolerância *Tol*.

Observação: Nos algoritmos, **BYREF** indicará que o parâmetro está sendo passado por referência.

Então o algoritmo é o seguinte:

```

Procurar_um_tap_de_2_para(intensidade de Sinal que chega ao tap      X,
                          BYREF tap a encontrar                      TAP,
                          BYREF intensidade de Sinal que sai do tap  Y)
{
  Seja TAP2[ ] um array cujo tipo de dado é um tap, e tem a coleção de todos
                os taps de dois que existem.

```

// assim, por Ex:

```

//   TAP2[1].Saídas = 2, TAP2[1].A = 26, TAP2[1].D_SA = 0.8
//   TAP2[2].Saídas = 2, TAP2[2].A = 23, TAP2[2].D_SA = 0.8
//           ...
//   TAP2[8].Saídas = 2, TAP2[8].A = 4, TAP2[8].D_SA = gb_LNP

```

i = 0;

Encontrado = FALSE;

Repetir

```
{ i = i + 1;
```

```
  Se ((X - TAP2[i].A) >= gb_Rs - gb_TOL) então
```

```
  {
```

```
    TAP = TAP2[i];
```

```
    Y = X - TAP.D_SA;
```

```
    Encontrado = TRUE;
```

```

    }
}
até (TAP2[i].D_SA = gb_LNP) ou (Encontrado = TRUE);

Se (Encontrado = FALSE) então
    { // indicaremos que não existe tap de 2 colocando um
      // valor inconsistente em Y e no campo Saídas de TAP.
      TAP.Saídas = 0;
      Y = -1;
    }
}

```

Evidentemente, no caso dos taps de 4 e 8 os algoritmos são análogos. Chamaremos estes dois algoritmos de:

Procurar_um_tap_de_4_para(X, TAP, Y)
Procurar_um_tap_de_8_para(X, TAP, Y)

Há situações em que o poste não atende a 2, 4 ou 8 residências exatamente. Neste caso devemos ligar vários taps com a finalidade de poder atender a todas as residências.

Definição. Chamaremos de *supertap* a qualquer combinação de taps, conectados.

Sejam nt_8 , nt_4 e nt_2 o número de taps de 8, 4 e 2 que formam um supertap. Para minorar o número de taps a usar vamos estabelecer que nt_4 e nt_2 só podem assumir valores 0 ou 1. Para as seguintes fórmulas, assumiremos que temos a intensidade de sinal suficiente para todos os taps que formam o supertap. Podemos imaginar um supertap como:

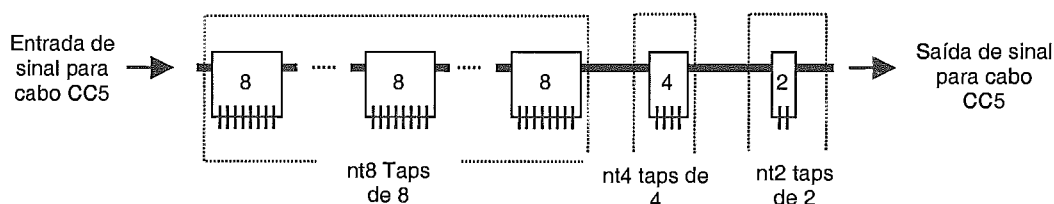


Figura 3.8. Representação de um Supertap.

Assim, para representar um supertap, vamos utilizar uma estrutura que tenha os seguintes campos:

nt8; número de taps de 8 que formam o supertap;
nt4; número de taps de 4 que formam o supertap;
nt2; número de taps de 2 que formam o supertap;

Para instalar um supertap, para uma quantidade menor de 8 residências seguiremos as seguintes regras:

- RT1. Se o número de residências é 7 então tentar usar somente um tap de 8.
- RT2. Se o número de residências é 5 ou 6 então, se existe intensidade de sinal suficiente para conectar um tap de 8, tentar usá-lo. Senão, tentar usar um tap de 4 e um tap de 2.
- RT3. Se o número de residências é 3 ou 4, então tentar usar um tap de 4.
- RT4. Se o número de residências é 1 ou 2 então tentar usar um tap de 2.

Estas regras têm a finalidade de minimizar o número de taps a usar, além de tentar utilizar taps mais adequados à quantidade de residências, por motivos de custos. Não estamos assumindo que a rede possa se expandir no futuro, mas na prática é fácil e viável trocar, por exemplo, um tap de 2 e $A = 14$, por um tap de 8 e $A = 14$, sem alterar de forma significativa os sinais na saída para cabo CC5 do tap. Com as regras vamos fazer o procedimento:

```
Tentar_Definir_STAP7( número_de_residências RESD,  
                    intensidade_de_sinal X,  
                    BYREF Supertap STAP,  
                    BYREF intensidade_de_sinal Y  
                    )  
{  
    variável TAP;  
    // o tipo de dado do TAP tem três campos:  
    // Saídas ; que indicará o número de saídas 8 ou 4 ou 2 do tap  
    // A ; que indica a atenuação para cada saída a residência  
    // D_SA ; que indica a atenuação  $\Delta_{S/A}$   
  
    // TAP é válido se TAP.Saídas > 0  
    // inicializaremos STAP. Se fica no seu valor inicial, então não
```

```

// conseguimos encontrar um supertap e os campos de STAP
// vão nos indicar isto.
STAP.nt8 = 0;
STAP.nt4 = 0;
STAP.nt2 = 0;
Se (RESID = 7) então
{
    Procurar_um_tap_de_8_para(X, TAP, Y);
    Se é_válido(TAP) então
        {
            STAP.nt8 = 1;
        }
}
Se ((RESID = 5) ou (RESID = 6)) então
{
    Procurar_um_tap_de_8_para(X, TAP, Y)
    Se é_válido(TAP) então
        {
            STAP.nt8 = 1;
        }
    Senão
        {
            Procurar_um_tap_de_4_para(X, TAP, Y)
            Se é_válido(TAP) então
                {
                    STAP.nt4 = 1;
                    Procurar_um_tap_de_2_para(Y, TAP, Y)
                    Se é_válido(TAP) então
                        {
                            STAP.nt2 = 1;
                        }
                    Senão
                        { // indicaremos que não existe supertap
                            STAP.nt4 = 0;
                            STAP.nt2 = 0;
                        }
                }
        }
}
}
}
}

```

Se o número de residências for maior que 8, então dividiremos este número por 8. O quociente inteiro da divisão vai nos indicar o número inicial de taps de 8 a usar. Seja $nt8$ aquele quociente. Para a instalação sucessiva de cada um daqueles

nt8 taps de 8, vamos verificar se o sinal que chega a cada um deles é suficiente. Se não existe sinal suficiente para a instalação de apenas um tap daqueles **nt8** (ou seja, não existe tap de 8 que possa se instalar para aquele sinal), então o supertap não poderá ser instalado. Se o resto da divisão for diferente de zero e existir sinal depois da instalação dos **nt8** taps, então utilizaremos o procedimento:

Tentar_Definir_STAP7

com seu parâmetro **RESD** igual ao resto da divisão.

O procedimento para encontrar o supertap para qualquer número de residências é o seguinte:

```
Indica_o_Supertap( número_de_residências RESD,
                   intensidade_de_sinal X que chega ao supertap,
                   BYREF Supertap STAP,
                   BYREF intensidade_de_sinal Y)
{
  nt8 = Quociente inteiro da divisão entre RESD e 8;
  nt07 = Resto da divisão entre RESD e 8;
  k = 1;
  Válido = TRUE;      // será FALSE quando não
                      // existe um supertap.
  variável STAP7 do tipo de um supertap;
  STAP.nt8 = 0;
  STAP.nt4 = 0;
  STAP.nt2 = 0;

  Enquanto ((k <= nt8) e (Válido = TRUE)) fazer
  {
    Procurar_um_tap_de_8_para (X, TAP, Y)
    Se é_válido(TAP) então
      { X = Y;      // a saída de sinal de um tap será a entrada
                // do seguinte tap.
        STAP.nt8 = STAP.nt8 + 1;
        k = k + 1;
      }
    Senão
      {
        Válido = FALSE;
      }
  }
}
```

```

Se (Válido = TRUE) então
{
    Tentar_Definir_STAP7(nt07, X, STAP7, Y)
    // uma variável T do tipo supertap não é válido se
    // T.nt8 = 0; T.nt4 = 0; T.nt2 = 0;
    Se é_válido_supertap(STAP7) então
        {
            STAP.nt8 = STAP.nt8 + STAP7.nt8;
            STAP.nt4 = STAP.nt4 + STAP7.nt4;
            STAP.nt2 = STAP.nt2 + STAP7.nt2;
        }
    Senão
        {
            Válido = FALSE;
        }
}

Se não (Válido = TRUE) então
{ // devolveremos um supertap não válido
    STAP.nt8 = 0;
    STAP.nt4 = 0;
    STAP.nt2 = 0;
}
}

```

Os algoritmos que definem os supertaps não levam em consideração os tipos de taps que serão instalados nos postes, e sim, a quantidade de cada tap de 8, 4 e 2 saídas. Isto é feito desta forma, porque no processo de busca pelo planejamento mais adequado que será discutido mais tarde, é custoso manter as estruturas de dados para cada tipo de tap definido pelo supertap. Os tipos de taps de 8, 4 e 2 que constituem o supertap podem ser facilmente encontrados após identificarmos os valores finais de **nt8**, **nt4** e **nt2** e sabermos o valor final do sinal de entrada do poste.

Para representar um supertap usaremos o símbolo



onde r indica o número de residências do dispositivo. A saída à direita do círculo representará a saída do supertap para cabo CC5.

Definição. Um poste $P \in \mathcal{P}$ é dito *ligado*, se:

1. Nele se instalou o amplificador, ou
2. Recebe sinal de outro poste já ligado, pela instalação de cabo entre eles.

Definição. Identificaremos por $\overline{\mathcal{P}}$ ao conjunto de todos os postes ligados de \mathcal{P} .

É importante notar que $\overline{\mathcal{P}}$ não precisa ter tamanho máximo para atender o maior número de residências.

Definição. No conjunto \mathcal{P} se define a relação Γ , P envia sinal a Q , como

$$\begin{aligned}\Gamma(P, Q) &\Leftrightarrow \text{de } P \text{ se envia sinal a } Q \text{ usando somente cabo} \\ &\Leftrightarrow Q \text{ está ligado a } P.\end{aligned}$$

onde P, Q são postes de $\overline{\mathcal{P}}$.

Definição. Diremos que um poste Q de \mathcal{P} está *desligado* se não existe um poste P de \mathcal{P} que cumpra $\Gamma(P, Q)$.

Definição. Sejam $P, Q \in \mathcal{P}$ dois postes ligados. Um *caminho de postes ligados* de P a Q é uma sucessão de postes ligados P_1, P_2, \dots, P_m onde

- a) $P_1 = P$ e $P_m = Q$, $m \geq 2$, m inteiro e
- b) Para todo P_i, P_{i+1} se cumpre $\Gamma(P_i, P_{i+1})$, $1 \leq i \leq m-1$, $m > 2$.

Condição do Problema. Se ligamos os postes P e Q , enviando sinal do primeiro ao segundo, então não poderemos enviar sinal de Q a P passando cabo do segundo ao primeiro e tampouco poderemos usar o cabo utilizado para emitir sinal de P a Q . Usando a definição anterior, expressaremos esta condição com:

$$\Gamma(P, Q) \Rightarrow \neg \Gamma(Q, P)$$

Definição. Sejam P e T dois conjuntos de postes conexos com $P \subset T$ e $P \neq T$.

1. Um poste P de P está na fronteira de P se existe um vizinho $Q \in T$ de P tal que $Q \notin P$. A fronteira de P será representada por $Fr(P)$.
2. Um poste que está na fronteira de um conjunto de postes P será chamado *fronteiriço*.

Exemplo. Seja o conjunto de postes $T = \{A, B, C, D, E, F, G, H, I, J\}$. Na seguinte figura, se existe uma aresta ----- entre dois postes, então assumiremos que existe a relação L entre eles.

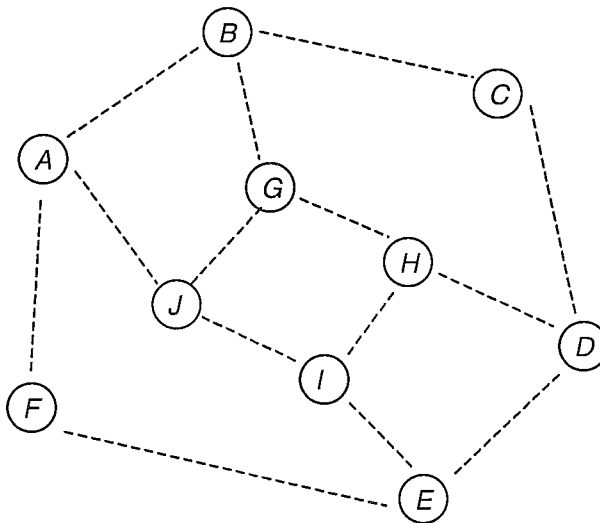


Figura 3.8. A fronteira de um conjunto.

Se P é o conjunto formado pelos postes G, H, I, J ; então a fronteira de P está formada por todos seus postes pois, para o poste G , existe o poste B de T , vizinho de G que não está em P . Para o poste H , existe o poste D de T , vizinho de G que não está em P . E assim por diante.

Se $P = \{G, H, I, J, E\}$, então a fronteira de P será o conjunto $\{G, H, J, E\}$. O poste E está na fronteira de P , porque existe pelo menos o poste D de T que é vizinho de E e não está em P .

Se $P = \emptyset$ então, evidentemente, $Fr(\emptyset) = \emptyset$.

Definição. Seja P um conjunto conexo de postes. Se I é um conjunto não necessariamente conexo de postes não ligados tal que $I \subset P$, $I \neq P$, então I será chamado *ilhado* se:

1. Para todo poste P de I , P não está na fronteira de P .
2. Se Q é vizinho de P , então, Q está ligado ou Q está em I .

Um poste de um conjunto ilhado também será chamado de ilhado.

Exemplo. Seja o conjunto de postes $T = \{A, B, C, D, E, F, G, H, I, J, K, L\}$. Na seguinte figura, se existe uma aresta dirigida \longrightarrow entre dois postes X e Y , então assumiremos que o poste X envia sinal ao poste Y .

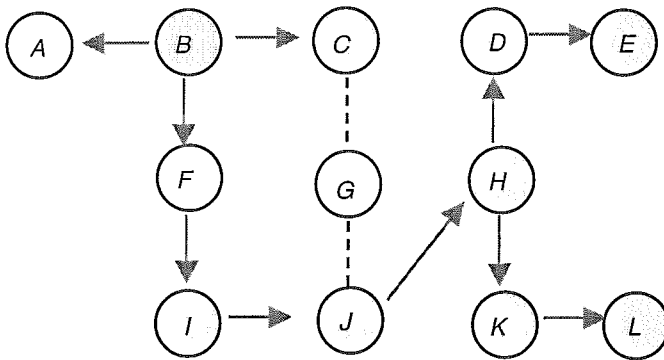


Figura 3.9. O poste G é ilhado

Se a fronteira de T é $\{E, L, C\}$ então o conjunto $\{G\}$ é ilhado.

Definição. Um poste Q de P é dito *terminal*, se existe um único poste P de P que cumpre $L(P, Q)$ e para todo conjunto T que contenha propriamente P , não existe um poste T de T , tal que $L(P, T)$.

O Processo de Instalação dos Distribuidores num Poste

Definição. Uma *árvore binária de distribuição de sinal*, é um conjunto T de distribuidores binários onde apenas um distribuidor recebe sinal de um cabo CC5 ou da saída de um supertap. Ele será a raiz da árvore. Para cada um dos outros distribuidores D daquele conjunto, existe um outro distribuidor D' de T que tem uma de suas saídas de sinal ligada à entrada de sinal de D .

Condição do Problema. Se um poste P a ser ligado tem residências, então instala-se o supertap respectivo, e para atender a seus vizinhos, na saída do supertap instala-se uma árvore binária de distribuição de sinal. Mas se P não tem residências, então instala-se apenas uma árvore binária de distribuição de sinal que receberá sinal do cabo que transmite sinal ao poste P .

É interessante observar que como $|V(P)| \leq 6$ então o número de saídas de sinal a ligar na árvore binária de distribuição de sinal deve ser menor ou igual que 6.

Assim num poste com 4 vizinhos e que não tem residências, teremos uma situação como a mostrada na seguinte figura:

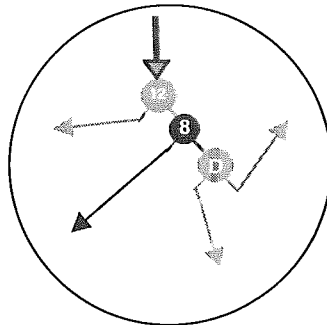


Figura 3.10. A árvore binária de distribuição de sinal.

O círculo maior representa um poste P . Neste caso, a árvore binária de distribuição de sinal está formada por dois tipos diferentes de acopladores direcionais e um divisor. A raiz da árvore é o acoplador direcional de 12. A árvore tem quatro saídas para transmitir sinal a quatro postes vizinhos. Cada seta que sai de um distribuidor binário servirá para transmitir sinal a um único vizinho. O único distribuidor que recebe sinal de um

cabo CC5 de entrada é o acoplador direcional de 12. Isto é representado pela seta que aponta ao acoplador direcional de 12.

Definição. Um subconjunto de postes D não necessariamente conexo de P , onde alguns postes de P podem estar ligados, é chamado *dispensável*, se cada um de seus postes é ilhado ou terminal e não tem residências para atender.

Definição. Um poste P de P é chamado *crítico*, se é ilhado ou terminal e tem residências para atender.

Definição. Um poste P de P é chamado *inalcançável* nas atuais ligações feitas naquele conjunto, se não existe outro poste Q de P , tal que com as condições dadas para ligar postes, possa se cumprir $\Gamma(P, Q)$.

Mínimo Requerimento de Sinal de um Poste

É desejável obter alguma expressão matemática que nos ajude a intuir os mínimos requerimentos de sinal para um poste.

Se um poste P tem residências para atender e ainda não está ligado, então a mínima intensidade de sinal que precisa é a necessária para atender a seu supertap respectivo se P não tem vizinhos que possam ser alimentados por ele. Se P tem vizinhos que possam ser ligados a ele, então a intensidade mínima de sinal será a necessária para atender a seu supertap respectivo de forma que na saída do supertap exista intensidade de sinal suficiente para atender o mínimo dos mínimos requerimentos de sinal de cada um dos vizinhos de P .

Se o poste P é dispensável ou já está ligado, então não precisamos considerá-lo para calcular a intensidade mínima de sinal. Por definição, vamos assumir que a intensidade mínima de sinal necessária para um poste de este tipo, será igual a uma constante inteira **gb_Intensidad_insatisfativel** que terá o valor 10000 dBm. Assim, este poste nunca estará no caminho da busca.

Sendo assim, nossa fórmula matemática para a intensidade mínima de sinal é definida a seguir.

Definição. Seja H um subconjunto de postes de P .

Seja P um poste de P .

A *intensidade mínima de sinal* necessária a P não levando em consideração os postes de H , é representada por $\text{minsc}(P, H)$ onde:

$\text{minsc}(P, H) =$ intensidade mínima necessária apenas a seu supertap, se P tem residências para atender, ainda não está ligado, não está em H e, se existir um Q vizinho de P tal que, Q está em H , ou Q já está ligado.

$\text{minsc}(P, H) =$ **gb_Intensidad_insatisfativo**, se P é dispensável, ou está ligado, ou *está* em H .

$\text{minsc}(P, H) =$ intensidade mínima necessária ao seu supertap de forma que o sinal na saída daquele dispositivo, seja maior ou igual que o mínimo, dos mínimos requerimentos de sinal de entrada necessário aos vizinhos. Ou seja, o sinal na saída seja maior o igual a

$$\min\{ \text{minsc}(Q_1^P, H \cup \{P\}) + \text{atenuação}(P, Q_1^P), \dots, \\ \text{minsc}(Q_v^P, H \cup \{P\}) + \text{atenuação}(P, Q_v^P) \},$$

onde Q_1^P, \dots, Q_v^P são os v vizinhos ($1 \leq v \leq |V(P)|$) de P que não estão em H , e $\text{atenuação}(P, Q_k^P)$ é a atenuação do cabo desde o poste P ate o poste Q_k^P , para todo $1 \leq k \leq v$, k inteiro.

$$\text{minsc}(P, H) = \min\{ \text{minsc}(Q_1^P, H \cup \{P\}) + \text{atenuação}(P, Q_1^P), \dots, \\ \text{minsc}(Q_v^P, H \cup \{P\}) + \text{atenuação}(P, Q_v^P) \},$$

caso contrário. Onde Q_1^P, \dots, Q_v^P são os v vizinhos ($1 \leq v \leq |V(P)|$) de P que não estão em H , e $\text{atenuação}(P, Q_k^P)$ é a atenuação do cabo desde o poste P ate o poste Q_k^P , para todo $1 \leq k \leq v$, k inteiro.

Observação. Nesta definição, a sentença *não levando em consideração o conjunto H de postes* é equivalente a dizer que os postes de H não serão usados para ligar P ou para enviar sinal a eles.

Definição. A *intensidade mínima de sinal* para que um poste P de \mathbf{P} seja ligado usando só *cabo*, será representada por $\text{min_sinal_SC}(P)$ e será:

$$\text{min_sinal_SC}(P) = \text{minsc}(P, \emptyset).$$

onde \emptyset é o conjunto vazio.

Infelizmente implementar um algoritmo para estas expressões pode levar a uma execução de custo muito alto.

A Atenuação e a Freqüência do Sinal

Na TV por cabo existem duas faixas de freqüências distintas chamadas: grupo de freqüência direta (Canal Direto - que varia de 50 a 500 MHz) e grupo de freqüência reversa (Canal Reverso - que varia de 5 a 45 MHz). A atenuação depende da freqüência do sinal medida em Mhz. Por exemplo, para freqüências altas (500Mhz) o cabo CC5 tem uma atenuação de 0,0512 dBm para cada metro, enquanto que para freqüências baixas (50Mhz) sua atenuação é 0,0144 dBm para cada metro. Isto acontece para todos os dispositivos apresentados até agora.

Nosso trabalho levará em consideração apenas o Canal Direto e os valores de atenuação para freqüências altas (neste caso, somente o valor de atenuação da freqüência mais alta - 500 MHz) nos diferentes dispositivos. A razão para isto é que a solução encontrada para altas freqüências (onde se tem a maior atenuação) também será solução para baixas freqüências.

Se X_H e X_L são, respectivamente, as intensidades de sinal em alta e baixa freqüências que entram num cabo CC5; Y_H e Y_L , são respectivamente, as intensidades que saem do cabo e chegam a um poste P ; e C é o comprimento do cabo, então, por construção do equipamento,

$$Y_H = X_H - A_H \times C$$

$$Y_L = Y_H - A_L \times C$$

onde A_H e A_L são as atenuações para altas e baixas frequências do cabo CC5 por cada metro. Observamos que Y_L depende de Y_H .

Para fins de relatório devemos calcular a intensidade de sinal em altas e baixas frequências e nos Canais Direto e Reverso que chega a um poste.

A Entrada de Dados do Problema

Para que possamos resolver o problema, é necessário conhecer a topologia do problema que será representada por um grafo. Neste grafo os postes são os nós e as arestas indicam os vizinhos de cada poste. Além disso cada aresta tem um peso que corresponde ao comprimento de cabo necessário para ligar dois postes vizinhos. É necessário também saber o número de residências de cada poste.

Neste caso, podemos resumir a informação de entrada como:

1. Os identificadores dos postes (nós de um grafo).
2. Os vizinhos de cada poste (arestas).
3. Se P e Q são postes, onde Q é vizinho de P , então precisamos conhecer o comprimento do cabo necessário para ligar P a Q , para todo P e todo vizinho Q de P .
4. O número de residências para cada poste.
5. Os postes da fronteira.
6. Os postes terminais que estão na microcélula.

Observações

O mais importante neste problema é ligar a maior quantidade de residências, procurando evitar utilizar divisores ou acopladores direcionais por sua forte atenuação e pelo custo. Resumindo, queremos atingir o maior número de residências possível com menor custo de equipamentos.

O uso do amplificador é obrigatório, porque só através dele pode-se transmitir sinal ao conjunto de postes a ligar.

A complexidade de nosso problema aumenta muito somente se o número de vizinhos por poste for grande. A condição que limita o número de vizinhos de um poste a ser menor ou igual que 6 tem por finalidade evitar isso. Porém, as heurísticas apresentadas no capítulo seguinte são insensíveis a esta limitação.

As definições apresentadas neste trabalho tentam contemplar o conjunto máximo de situações possíveis após alguns postes estarem ligados. A situação em que postes ficam ilhados é um exemplo disso.

Modelagem

Vamos definir com a relação L apresentada anteriormente e o conjunto de postes P , o grafo $G = (P, L)$, onde L definido como

$$L = \{ (P, Q) \in P^2 \mid L(P, Q) \}$$

é o conjunto de *arestas de G*.

Nas figuras representaremos a relação $(P, Q) \in L$ com ---- . Na figura 4.1, o conjunto de postes é $\{A, B, C, D, E, F, G, H, I, J, K, L\}$.

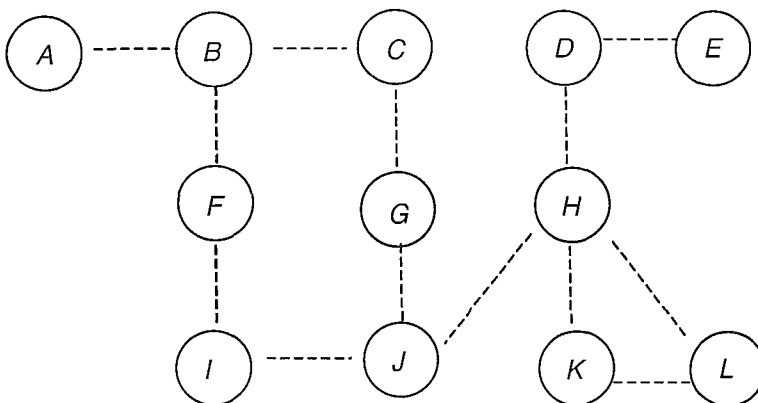


Figura 4.1. Um grafo de postes G .

Na figura 4.1, poderemos dizer que:

$(A, B) \in L$: o poste B pode ser ligado ao poste A e vice-versa.

$(B, C) \in L$: o poste C pode ser ligado ao poste B e vice-versa.

$(C, G) \in L$: o poste G pode ser ligado ao poste C e vice-versa.

etc. Devemos observar que $(P, Q) \in L$ não implica, $\Gamma(P, Q)$ ou $\Gamma(Q, P)$.

Para representar num grafo G que o poste Q está ligado ao poste P usaremos uma seta \longrightarrow que parte de P e aponta para Q como em

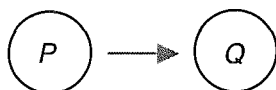


Figura 4.2. Representação de postes ligados.

Estruturas de Dados Básicas

Podemos observar que para cada poste precisamos conhecer basicamente:

1. Um identificador.
2. Uma lista circular de estruturas para *apontar* a seus vizinhos.
3. Um ponteiro para um tap ou um árvore binária de distribuição de sinal.
4. Seu número de residências.
5. Se o poste é fronteiroço.
7. Se o poste ficou ilhado.
8. Se o poste é terminal.

Estruturas de Dados para os Distribuidores de Sinal

Para projetar 2. e 3. definiremos para cada poste

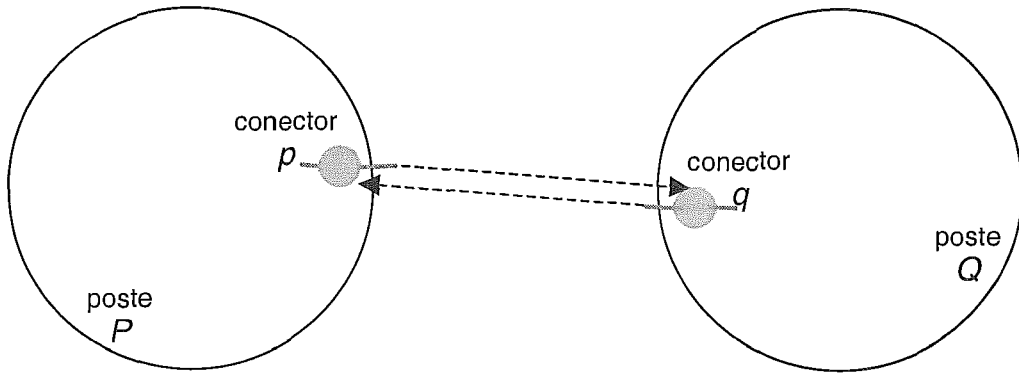



Figura 4.3. Representação da relação $L(P, Q)$

o *conector de postes*, representado pelo símbolo  como mostra a figura 4.3. Este conector tem duas saídas. Uma que aponta para um outro conector externo e a outra que somente é utilizada quando o conector recebe sinal. Cada uma das saídas é chamada *ligador de cabo*.

Na figura 4.3, o conector p do poste P , *apontará* ao conector q do poste Q e vice-versa, obedecendo a relação $L(P, Q)$. Se cada conector tem uma propriedade que indica o poste onde está, então poderemos representar a relação L da seguinte maneira:

- $L(P, Q) \Leftrightarrow$ o poste Q pode ser ligado ao poste P ou vice-versa
- \Leftrightarrow existem conectores de postes p em P e q em Q tal que p aponta para q ou vice-versa.

Assim, podemos imaginar um conector de postes da seguinte forma:

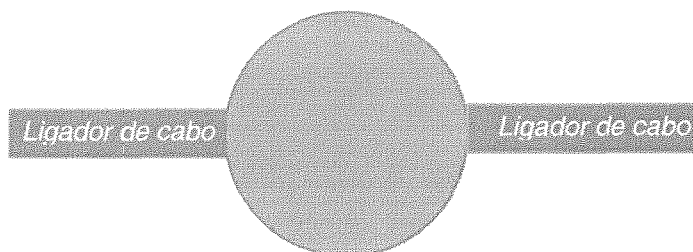


Figura 4.4. Conector de postes.

Como cada conector de postes precisa ter um *apontador ao poste onde está*, então o conector p deve ter um apontador ao poste P onde se encontra. De modo similar para o conector q do poste Q .

A estrutura do ligador de cabo será utilizada como campo na definição da estrutura do conector de postes. Portanto, o conector de postes poderá apontar a outro poste usando a sua estrutura do ligador de cabo.

Para simular o envio de sinal do poste P ao poste Q pelos conectores de postes p e q , precisaremos de um campo na estrutura do ligador de cabo que chamaremos **Cabo_Ligado**. Se o valor deste campo for TRUE então existirá um direcionamento de sinal do conector p ao conector q (e simularemos que existe cabo conectado entre os dois postes P e Q). Isto não acontecerá se aquele campo tiver o valor FALSE. Na figura 4.5, a seta



indica que o conector p envia sinal ao conector q .

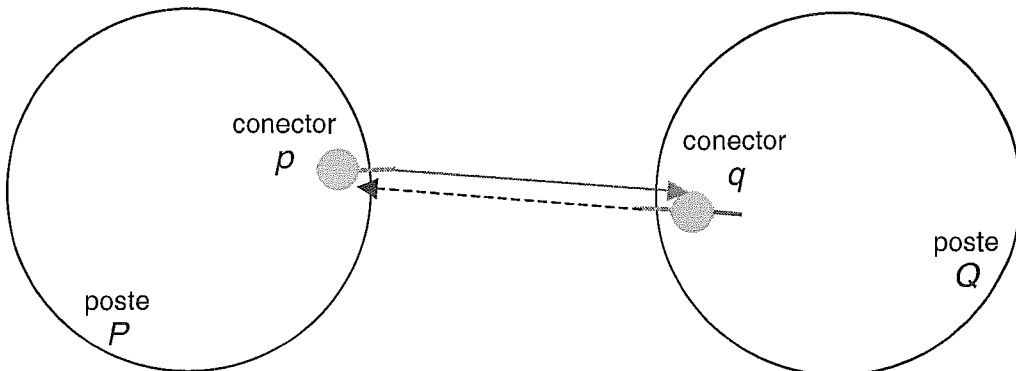


Figura 4.5. Representação do envio de sinal de um poste a outro.

Aquele ligador de cabo que não for utilizado para apontar a outro conector de postes, será utilizado para transmitir sinal internamente no seu poste. Usando aquele ligador de cabo vamos tentar instalar um supertap ou um distribuidor binário de sinal.

Todo ligador de cabo vai ter uma propriedade que vai nos indicar o sinal que sai do conector de postes. Esta propriedade vamos chamar de **Sinal_de_Saida**.

O ligador de cabo também vai ter um campo que chamaremos **Atenuação**. Ele, no caso do conector de postes deve ter o valor ZERO, pois o conector de postes é um dispositivo fictício. A estrutura do ligador de cabo, deverá ter um apontador ao conector de postes em que está definido.

Um conector de postes vai poder também receber sinal de um distribuidor de sinal. Neste caso, precisaremos também conhecer a intensidade do sinal que sai pelo seu ligador de cabo de saída, quando isso acontece. Na figura:

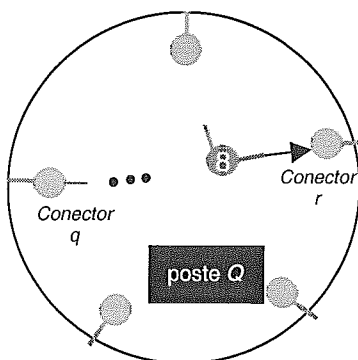


Figura 4.6. Envio de sinal de um distribuidor de sinal.

o acoplador direcional de 8, envia sinal ao conector de postes r do poste Q . Para ligar o vizinho de Q apontado por r , deveremos conhecer o sinal que sai do ligador de cabo. Todo conector de postes r deverá ter uma propriedade que indique qual ligador de cabo de um distribuidor de sinal está enviando sinal a ele.

Para armazenar a distância dos postes P e Q , utilizaremos os ligadores de cabo dos conectores de postes p e q . Todo ligador de cabo deverá ter um campo chamado **dist_a_Vizinho**. Ele será utilizado nos ligadores de cabo que apontam para um conector de postes de outro poste. Assim, no caso de p e q , o campo **dist_a_Vizinho** terá o valor da distância do poste P ao poste Q .

Para um poste, lembremos que precisamos especificar uma lista circular para *apontar* a seus vizinhos. Podemos imaginar um poste com sua lista de vizinhos através da seguinte figura:

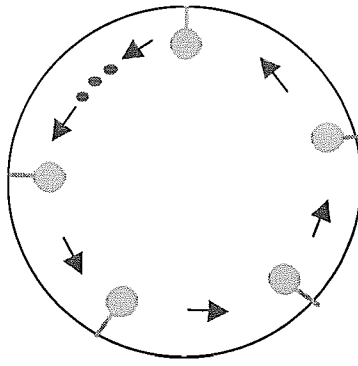


Figura 4.7. Representação da lista dos conectores de postes.

Nesta figura, cada seta → indica o elemento seguinte da lista circular dos conectores de postes. Assim, precisamos definir na estrutura do conector de postes, um campo que é um apontador para outro conector de postes para poder formar a lista circular respectiva. Por definição, cada um destes conectores tem o endereço do poste em que está *instalado* (lembrar que um conector de postes é um dispositivo imaginário) e se apontamos a um deles então facilmente poderemos conhecer o poste em que está definido.

Os distribuidores binários de sinal são similares aos conectores de postes. Eles seguem o esquema:

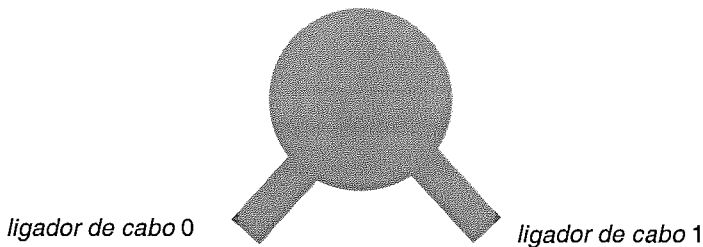


Figura 4.8. Representação de um distribuidor binário de sinal.

Podemos observar a existência de dois ligadores de cabo, 0 e 1. Estes ligadores devem ter atenuações diferentes de zero para representar os diferentes distribuidores binários que podem existir. Por exemplo, se desejamos representar um divisor, os campos **Atenuação** nos ligadores 0 e 1 devem ser iguais.

A estrutura que representará um supertap é também similar. Cada supertap segue o esquema:

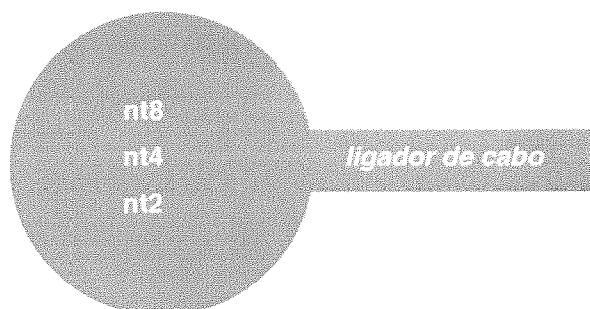


Figura 4.9. Representação de um supertap.

nt8, **nt4** e **nt2** são respectivamente, o número de taps de 8, 4 e 2 que formam o supertap. Como já foi observado anteriormente, o ligador de cabo servirá para transmitir sinal para um único distribuidor binário ou para um único conector de postes.

Podemos observar que um conector de postes poderá receber sinal de um *ligador de cabo* de um supertap.

Observação: Um conector de postes também será chamado distribuidor de sinal. É um dispositivo fictício.

Por último, o amplificador será representado com um conector de postes. Quando simulamos a instalação do amplificador num poste, inserimos na sua lista circular dos conectores de postes outro conector que não apontará para ninguém, porque o poste onde se instalou o amplificador somente pode receber sinal. Vamos assumir, entretanto, que o conector de postes que representa o amplificador, recebe sinal de outro poste que não é da microcélula. Um dos ligadores de cabo daquele novo conector passará o sinal do amplificador ao interior do poste.

Podemos imaginar um poste com um amplificador na seguinte figura:

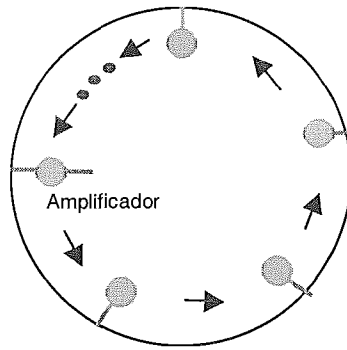



Figura 4.10. Representação da instalação de um amplificador.

O símbolo  indica o conector de postes associado ao amplificador instalado no poste.

É importante representar as estruturas de dados dos diferentes distribuidores de sinal em uma *única estrutura de dados*. Quando for relevante, indicaremos outras propriedades necessárias às estruturas de dados.

Afirmção. O número de postes de um conjunto de postes ligados é igual ao número de conectores de postes que *recebem* sinal de outro poste.

Demonstração. Desde que um poste só pode receber sinal de um único outro poste, o número total de conectores de postes que recebem sinal e que pertencem a esse conjunto vai ser igual ao número de postes ligados.

Representação do Grafo $G = (P, L)$

Para conhecer o grafo G precisamos formar uma lista encadeada com os postes.

Representação de um Caminho de Postes

Seja o grafo $G = (P, L)$ onde P é um conjunto de postes e L o conjunto das arestas segundo a relação L , e P é um poste de P .

Definição. Um *caminho de postes ligados usando somente cabo e supertaps*, chamado também *caminho PLUSCS*, é um caminho de postes $\{P_1, \dots, P_m\}$ de P (para algum inteiro $m > 0$, e menor ou igual ao número de postes), com $P_1 = P$, e que para $1 \leq k \leq m$ se cumpra que o sinal que sai do poste P_k é o que sai do supertap respectivo, se este poste tiver residências. Caso contrário, o sinal que sai de P_k será o mesmo que entra pelo cabo que chega a ele.

Para representar um caminho de postes ligados usando somente cabo e supertaps usando os conectores de postes, sejam E_k e S_k os conectores de postes do poste P_k por onde entra e sai o sinal no poste P_k , para $1 \leq k \leq m$. Assim, um caminho PLUSCS será representado pelo conjunto de conectores de postes:

$$\{ E_1, S_1, E_2, S_2, \dots, E_k, S_k, \dots, E_m, S_m \}$$

Se do último poste não sair sinal para nenhum outro poste, então o poste P_m só poderá ter um conector de postes por onde entra sinal E_m . E aquele caminho será representado por:

$$\{ E_1, S_1, E_2, S_2, \dots, E_k, S_k, \dots, E_m \}$$

Na figura 4.11 mostramos um caminho de postes e os conectores de postes que estabelecem aquele caminho.

Em algumas situações poderemos ter conjuntos de conectores de postes onde o primeiro elemento S_0 é um conector que envia sinal para um conector E_1 de outro poste P_1 (portanto S_0 aponta a E_1), como em:

$$\{ S_0, E_1, S_1, E_2, S_2, \dots, E_k, S_k, \dots, E_m, S_m \}$$

ou

$$\{ S_0, E_1, S_1, E_2, S_2, \dots, E_k, S_k, \dots, E_m \}$$

Neste caso, diremos que S_0 aponta ao caminho de postes que inicia no poste P_1 , mas que o poste P_0 que contém S_0 não pertence àquele caminho.

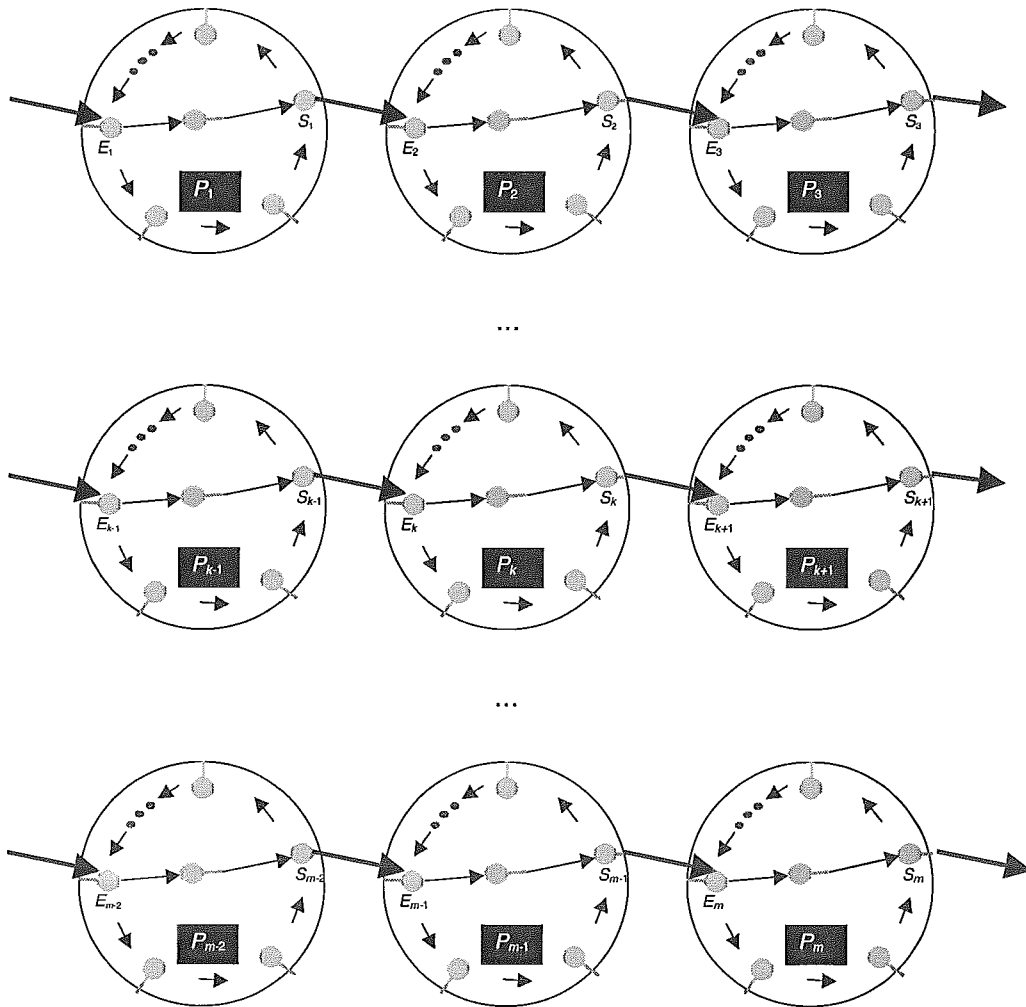


Figura 4.11. Caminho de postes ligados usando somente cabo e supertaps, e sua representação usando conectores de postes.

A função que nos indicará o poste onde está instalado um conector de postes é definida por:

```

Poste_em_que_esta_o_conector(conector de postes Cp)
{
    Devolver
    um identificador do poste em que está o conector de postes Cp;
}
    
```

Com esta função será fácil avaliar, a partir dos conectores de postes E_k , os postes do caminho e portanto o número total de residências daquele conjunto.

Definição. Um conjunto S de conectores de postes que representa um caminho de postes ligados usando só cabo e supertaps, será chamado *caminho de conectores de postes ligados usando somente cabo e supertaps* o também *caminho C_PLUSCS*.

Observação. Por um abuso de linguagem, diremos também que um caminho C_PLUSCS representa um caminho de postes ligados usando somente cabo e supertaps.

Sejam S e T dois *caminhos de conectores de postes ligados usando somente cabo e supertaps*. Então:

Definição. O número de postes de T será identificado por $|T|$.

Definição. O número de residências de T será identificado por $\|T\|$.

Definição. Diremos que S é *melhor que* T se:

$$\|S\| > \|T\| \quad \text{ou,} \quad \|S\| = \|T\| \text{ e } |S| < |T|$$

em qualquer outro caso diremos que S é *indiferente* a T .

Uma Estrutura de Dados Auxiliar

Em nossos algoritmos, vamos utilizar o conjunto:

{ (X, Y, Z) tal que:

X, indica um conector de postes

Y, indica o sinal que chegaria ao conector de postes

Z, indica se o conector X vai receber sinal de outro poste }

para representar os caminhos de conectores de postes ligados usando só cabo e supertaps. Este tipo de dado vamos chamar de

Tipo_do_Caminho_de_Conectores

Não é difícil ver que a partir desta estrutura podemos (procedendo como na explicação do caminho de conectores de postes) identificar um caminho PLUSCS. Portanto, estenderemos as quatro últimas definições a esta estrutura.

Requerimentos Mínimos para Ligar um Poste Q a outro P Usando Apenas Cabo

Seja o grafo $G = (P, L)$ onde P é um conjunto de postes, e L o conjunto das arestas de G segundo a relação L . Se P e Q são dois postes de P , onde P está ligado, e Sp é o conector de postes que aponta ao conector de postes Eq do poste Q , então ligaremos o poste Q ao poste P usando cabo, se qualquer das seguintes condições se cumprir:

1. Se o poste Q tem residências, e existe um supertap para Q que suporta o sinal que vai chegar de P através do cabo que vai chegar ao conector Eq .
2. Se o poste Q não tem residências, e o sinal que vai chegar a ele através do conector Eq , é maior que Rs , onde Rs é o sinal mínimo necessário para uma residência.

Com estas condições vamos construir um algoritmo, que irá dizer se podemos ligar um poste a outro.

```
Sao_Satisfeitos_os_Min_Req_Conx(Conector de postes          CNTOR_Sp,
                                Conector de postes          CNTOR_Eq,
                                Sinal que sai de conector Sp  I_Sp)
```

```
Variável booleana São_Satisfeitos;
```

```
P = Poste_em_que_esta_o_conector (CNTOR_Ap);
```

```
Q = Poste_em_que_esta_o_conector (CNTOR_Eq);
```

```
I_Eq = I_Sp - Atenuação_do_cabo(P, Q); /* sinal que chegaria a Eq */
```

```
São_Satisfeitos = FALSE;
```

```
Se o poste  $Q$  não está ligado então
```

```
  Se (as_Residências_de(Q) > 0) então
```

```
    {
```

```
      Indica_o_Supertap( Q.Residências, I_Eq, STAP, ISaidaSTAP)
```

```
      Se é_válido(STAP) então São_Satisfeitos = TRUE;
```

```
    }
```

```
  Senão
```

```
    {
```

```
      Se (I_Eq > gb_Rs) então São_Satisfeitos = TRUE
```

```
    }
```

```
  devolver São_Satisfeitos;
```

```
}
```

Algoritmos Principais

Nossa abordagem para resolver o problema tenta inicialmente estabelecer o maior caminho possível que parte do poste em que se instalou o amplificador gerando uma lista encadeada de postes para representar um caminho PLUSCS. Uma vez tendo estabelecido este caminho principal (*backbone*) da rede, percorremos este caminho principal procurando vizinhos dos postes do backbone que ainda não tenham sido ligados, gerando para isso novos caminhos PLUSCS que partam dos vizinhos dos postes do backbone. A medida que percorremos os postes do backbone, instalamos os supertaps definitivos em cada poste que tenha residências e os distribuidores de sinal (se necessário), e recalculamos a intensidade de sinal até o final do caminho (observe que esta é uma operação linear no número de postes do backbone). Se o sinal não for suficiente para atender todas as residências até o final do caminho, dispensamos postes fronteirços ou que ficam *perto da fronteira e não estão ilhados*, passando os mesmos a pertencer a uma nova microcélula. Os novos caminhos PLUSCS vão ser novos backbones e repetiremos em cada um destes o processo descrito acima. Como o sinal e o número de postes são finitos, existirá um momento em que não aparecerão novos backbones e o processo finalizará.

Os algoritmos necessários para obter a solução são apresentados a seguir.

O algoritmo do Melhor Caminho a partir de um Poste Inicial (*backbone*)

A partir de um poste P do grafo $G = (P, L)$, podem-se fazer vários caminhos de postes ligados usando só cabo e supertaps. Dentre eles, vamos escolher aquele que seja o *melhor*.

O algoritmo, usando recursividade e os conectores de postes, procura o melhor dos melhores caminhos PLUSCS que podem existir a partir dos vizinhos de P . Na seguinte figura:

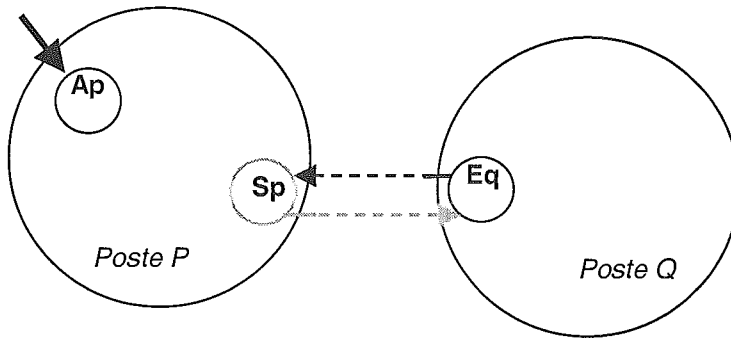


Figura 4.12. Procura do melhor caminho de postes ligados usando somente cabo e supertaps a partir de **Ap**.

mostra-se uma situação onde **Ap** representa um conector de postes do poste *P* que *pode receber sinal de outro poste*. **Sp** representa um conector de postes do poste *P* que aponta ao conector **Eq** do poste *Q*. Também **Eq** aponta a **Sp**.

Para procurar o melhor *caminho de conectores de postes ligados usando somente cabo e supertaps* a partir de **Ap**, e com uma intensidade de sinal **I** que poderia sair deste conector, vamos calcular primeiro a intensidade de sinal **ISaidaSTAP** na saída do supertap respectivo de *P*, se ele tiver residências. Se *P* não tiver residências, a variável **ISaidaSTAP** será **I**. Com o valor final de **ISaidaSTAP**, para cada um dos outros conectores de postes **Sp** em *P*, vamos procurar o melhor *caminho de conectores de postes ligados usando somente cabo e supertaps* que parte do conector de postes apontado por **Sp**.

A figura:

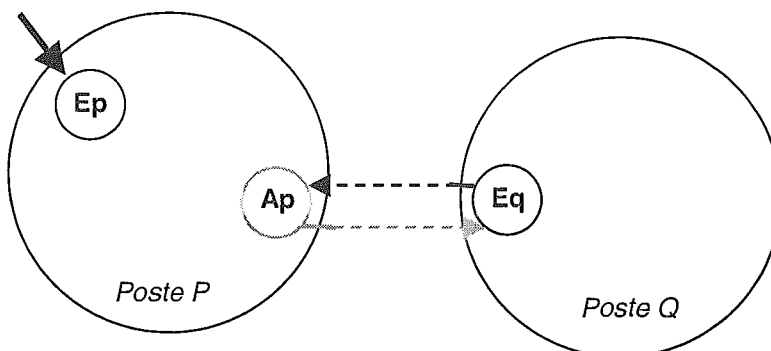


Figura 4.13. Procura do melhor caminho de postes ligados usando somente cabo e supertaps a partir de **Ap**.

mostra o segundo caso onde o conector **Ap** não vai receber sinal de outro poste. Ou seja receberá sinal de um conector interno. Neste caso, o melhor caminho desde **Ap** será procurado passando por **Eq**.

Como para comparar caminhos precisamos conhecer seu número de residências, postes e comprimento do caminho, então precisaremos de uma estrutura de dados com as seguintes características:

1. Campo que indique o conjunto de conectores de postes que representam o caminho. Vamos chamá-lo de **CJ**.

Cada elemento deste conjunto, deve ter três características, que indiquem:

1. Conector de poste.
 2. Sinal que poderia receber.
 3. Se vai receber o sinal de outro poste.
2. Número de residências do caminho. Chamaremos de **NRC**.
 3. Número de postes do caminho. Chamaremos de **NPC**.
 4. Comprimento do caminho. Chamaremos de **CPC**.

Chamaremos este tipo de dado de **Tipo_de_dado_dum_Caminho**.

Após definir as estruturas de dados necessárias para buscar o melhor caminho, apresentaremos nosso algoritmo. Seleciona-se um dos postes do grafo para iniciar a busca. Assume-se um sinal de entrada **I** para este poste. Em seguida, procuramos em profundidade o melhor caminho a partir daquele poste. O algoritmo é recursivo e termina quando já não há sinal suficiente para passar a outro poste ou quando todos os postes já estiverem ligados. A cada caminho parcial visitado, este caminho é comparado com o melhor caminho encontrado até o momento. Desta forma este algoritmo se comporta como um *branch and bound*.

```
Da_Melhor_Caminho_Desde( Conector_de_postes          CNTOR_Ap,  
                        var_booleana                CN_Receber_sinal_de_outro_Poste;  
                        Sinal_que_receberia_o_conector I );
```

/ o tipo de dado desta função é Tipo_de_dado_dum_Caminho */*

```
{  
  /* Esta função procura o melhor caminho de postes, representado por uma sucessão de */  
  /* conectores de postes, a partir do conector Ap que receberia um sinal I. */  
  /* Se recebe o sinal de outro poste então o segundo parâmetro é igual a TRUE. */  
  /* Senão é FALSE. */  
  /* Lembrar que um conector de postes tem atenuação 0 para cada uma de suas saídas. */
```

```
ptr_Cx;          /* Apontador para percorrer os outros */  
                /* conectores de postes na lista circular do */  
P;              /* poste P. */  
C0, C1, Cm_Eq, Cap, CMax; /* Representam caminhos de postes usando */  
                /* só cabo e supertaps. Portanto, seu tipo */  
                /* deve ser Tipo_de_dado_dum_Caminho. */  
                /* Em particular, CAp indicará o caminho de */  
                /* de postes ligados usando somente cabo */  
                /* e supertaps que parte do conector Ap. */  
ISaidaSTAP;    /* Sinal na saída do supertap. */  
I_Sp;          /* Sinal num conector de postes Sp. */  
I_Eq;          /* Sinal que chega ao conector de postes Eq. */  
NRSOP;        /* Uma variável auxiliar */
```

P = Poste_em_que_esta_o_conector (CNTOR_Ap);

/ temos o poste P que contém o conector Ap */*

Se CN_Receber_sinal_de_outro_Poste

```
  /* se o conector Ap receber sinal de outro poste ... */  
  {  
    /* quando um poste se liga a outro é porque os requerimentos */  
    /* mínimos de sinal do segundo estão satisfeitos. */  
    CAp.CJ = { (CNTOR_Ap, I, TRUE) }  
    /* Cada elemento do conjunto CAp.CJ tem três campos. */  
    /* O primeiro, indica um conector de postes */  
    /* e com ele poderemos conhecer seu poste respectivo. */  
    /* O segundo, indica o sinal que pode chegar ao conector de postes */  
    /* O terceiro, indicará se o conector vai receber sinal de outro poste */
```

```
    CAp.NPC = 1; /* CAp tem apenas um poste. */  
    CAp.CPC = 0; /* O comprimento de cabo usado a partir do */  
                /* poste P é 0. */
```

Se (as_Residências_de(P) > 0)

```
  { /* veremos o possível supertap */  
    Indica_o_Supertap( P.Residências, I, STAP, ISaidaSTAP)  
    CAp.NRC = P.Residências;  
  }
```

Senão

```
  ISaidaSTAP = I ;
```



```

/* CMax será o nome do melhor caminho de conectores de postes */
/* a encontrar. */
CMax = CAp;
/* por enquanto o melhor caminho tem o conector Ap */

/* Precisamos do seguinte conector a Ap na lista circular dos */
/* conectores de postes de P. */
ptr_Cx = seguinte_conector(CNTOR_Ap);

Enquanto (ptr_Cx <> CNTOR_Ap)
{
    CNTOR_Sp = ptr_Cx;
    /* temos o conector Sp */

    CNTOR_Eq = Conector_do_poste_Q_apontado_por(CNTOR_Sp)
    /* temos conector Eq */

    /* a intensidade de sinal que chega ao conector Sp */
    /* é a que sai do tap se este existe. senão, */
    /* IsaidaSTAP = I pois a atenuação é zero */

    I_Sp = IsaidaSTAP;

    Se Sao_Satisfeitos_os_Min_Req_Conx(CNTOR_Sp, CNTOR_Eq, I_Sp) Então
    { /* se são satisfeitos os mínimos requerimentos de conexão entre o */
    /* conector Sp e o conector Eq com o sinal I_Sp ... */

        C1.CJ = { (CNTOR_Sp, I_Sp, FALSE) };
        C1.NPC = 0;
        C1.NRC = 0;
        C1.CPC = 0;
        C1 = UNIR(CAp, C1);
        /* UNIR é uma função do tipo Tipo_de_dado_dum_Caminho */
        /* que une os conjuntos CAp.CJ e C1.CJ. O dado retornado pela */
        /* função, no seu campo NPC, possui a soma dos postes da união.*/
        /* No seu campo NRC, possui a soma das residências da união. */
        /* No seu campo CPC, possui a soma dos comprimentos dos */
        /* caminhos da união. */

        I_Eq = I_Sp - Atenuacao_do_Cabo_desde(CNTOR_Sp);
        /* a expressão Atenuação_do_Cabo_desde fornecerá a */
        /* atenuação desde o poste P ao poste Q. */

        Cm_Eq = Da_Melhor_Caminho_Desde( CNTOR_Eq,
                                         TRUE,
                                         I_Eq);

        C1 = UNIR(C1, Cm_Eq);
    }
}

```

```

        Se É_Melhor(C1, CMax)
        {
            Eliminar_Conjunto(CMax);
            CMax = C1;
        }
        Senão
        {
            Eliminar_Conjunto(C1);
        }
    }
    ptr_Cx = seguinte_conector(ptr_Cx);
}
}
Senão /* Ap não recebe sinal de outro poste ... */
{
    CNTOR_Eq = Conector_do_poste_Q_apontado_por(CNTOR_Ap)
    /* temos o conector Eq */
    I_Ap = I;

    Se São_Satisfeitos_os_Min_Req_Conx(CNTOR_Ap, CNTOR_Eq, I_Ap);
    {
        I_Eq = I_Ap - Atenuacao_do_Cabo_de(CNTOR_Ap);
        C0.CJ = { (CNTOR_Ap, I_Ap, FALSE) };
        C0.NPC = 0;
        C0.NRC = 0;
        C0.CPC = 0;
        C1 = Da_Melhor_Caminho_Desde(CNTOR_Eq,
                                    TRUE,
                                    I_Eq);

        CMax = UNIR(C0, C1);
    }
    Senão
        CMax = ∅; /* CMax assume o conjunto vazio */
}
/* Retornará o melhor caminho */
Devolver CMax;
}

```

Indicando um Caminho PLUSCS no Grafo $G = (P, L)$

Para todo poste precisamos de duas propriedades, (1) uma que indica por qual conector de postes aquele poste está recebendo sinal e a intensidade de sinal que entra, e (2) uma que indica o primeiro conector de postes pelo qual deverá sair sinal do poste, e a intensidade de sinal que sai, se existir sinal suficiente.

O algoritmo anterior, retorna uma variável **CMax** da qual podemos obter um caminho PLUSCS. Para indicar aquele caminho no grafo $G = (P, L)$, vamos utilizar suas propriedades (ou campos). Vamos percorrer os conectores dos elementos do campo **CJ** (que é um conjunto), localizando-os nos seus postes do grafo G , indicando em cada conector de postes pelo qual sai o sinal: *o número de residências do caminho que inicia no poste apontado pelo conector, o número de postes e o comprimento daquele caminho mais o comprimento do cabo que parte daquele conector até o próximo poste daquele caminho*. Em um conector de postes que recebe sinal, *indicaremos o número de residências atendidas no poste em que está instalado* e nas duas propriedades mencionadas acima, indicaremos este conector como aquele pelo qual entra o sinal no poste. O elemento seguinte a este conector no conjunto **CJ**, indicará o primeiro conector de postes pelo qual sairá o sinal, se existir. Na medida em que percorremos o conjunto e encontramos um poste que não tenha residências, indicamos no grafo como *ligado* e continuamos a percorrer os postes seguintes até o final. Se, no decorrer do conjunto **CJ**, chegamos a um poste que, assim como todos os seus sucessores, não possui residências, então desligamos todos estes postes no grafo G .

Percorrendo o Backbone

Ao percorrermos o backbone, ocorrerão situações onde será necessário atender um caminho de postes ligados cujo poste inicial não é aquele em que se instalou o amplificador no grafo $G = (P, L)$, e cuja intensidade de sinal é diferente daquela que formou aquele caminho (por exemplo, adicionamos um vizinho não ligado e isto fez reduzir o sinal de alimentação do caminho restante no backbone).

Suponhamos que um backbone inicialmente esteja formado pelos postes P_1, P_2, \dots, P_m de P . Também vamos supor que existe um poste P_k ($1 \leq k \leq m-1$) que através de um de seus conectores de postes, digamos **Sp**, envia sinal a P_{k+1} , e que por alguma circunstância o sinal que sai por **Sp** é diferente daquele que originalmente se enviava a P_{k+1} . Pode acontecer que alguns dos últimos postes do backbone fiquem sem sinal suficiente e portanto precisem ser desligados. Desta forma se faz necessário recalcular para os postes que estão no caminho PLUSCS que inicia em P_{k+1} , o novo número de residências que podem ser atendidas e o novo comprimento deste caminho.

Para fazer isto, devemos iniciar uma busca a partir do conector **Sp**, parando somente quando não pudermos passar para outro poste daquele caminho (porque o sinal não é suficiente) ou quando chegarmos ao poste final. Além disso, deveremos calcular os novos supertaps de cada poste.

Para o algoritmo que vamos apresentar, precisamos da seguinte estrutura de dados que chamaremos **Tipo_Dados_Dum_Caminho**, cujos campos são:

1. Residências do Novo Caminho: **RNC**;
2. Postes do caminho: **NPC**;
3. Comprimento do novo caminho: **CNC**;

O algoritmo que implementa a função de recálculo do backbone a partir de **Sp** está descrito a seguir.

```

Função Pesquisa_Caminho_Sinalado_por(Conector de postes      CNTR_Sp,
                                     Sinal que sai do Sp      I_Sp)

/* esta função é do tipo de dado Tipo_Dados_Dum_Caminho */

{ /* suponhamos que CNTR_Sp transmite sinal */
  P = poste onde o conector de postes CNTR_Sp está instalado;
  C = CNTR_Sp;
  Variável Rptas do tipo Tipo_Dados_Dum_Caminho;
  RNC = 0;      /* Residências do Novo Caminho.                */
  NPnDPL = 0;  /* Número de Postes não dispensáveis que Podem ser Ligados.*/
  CdNC = 0;    /* Comprimento do Novo Caminho.                */
  NPSR = 0;    /* Número de Postes sem Residências depois do último poste */
               /* com Residências.                                     */
  CCPSR = 0;   /* comprimento do caminho de postes sem residências após  */
               /* o último poste com residências.                       */
  pode_se_continuar = TRUE;
  Seja CNTR_Eq o conector de postes apontado por CNTR_Sp;
  Enquanto (Sao_Satisfeitos_os_Min_Req_Conx(CNTR_Sp, CNTR_Eq, I_Sp) e
            pode_se_continuar) fazer
  {
    Seja Q o poste do conector CNTR_Eq;
    Se (o poste Q tem residências) Então
    {
      Indica_o_Supertap( Q.RESD, Iq, STAP, Y);
      /* Y é o sinal que poderia sair do supertap no poste Q */
      RNC = RNC + Q.RESD;
    }
  }
}

```

```

        NPnDPL = NPnDPL + 1;
        CdNC = CdNC + distancia_entre_os_postes(P, Q);
        /* como o ultimo poste tem residências ....*/
        NPSR = 0;
        CCPSR = 0;
    }
    Senão
    {
        NPSR = NPSR + 1;
        CCPSR = CCPSR + distancia_entre_os_postes(P, Q);
    }
    Se (o poste Q não tem conector de postes para envio de sinal) então
    {
        pode_se_continuar = FALSE;
    }
    Senão
    {
        CNTR_Sp = Conector de postes pelo que se envia sinal de Q;
        P = Q;
        CNTR_Eq = o conector de postes apontado por CNTR_Sp;
    }
}
/* se existe um caminho de postes sem residências depois do ultimo poste */
/* com residências, então devemos retirar aquele número de postes do */
/* número de postes calculado. Analogamente com o comprimento. */
NPnDPL = NPnDPL - NPSR;
CdNC = CdNC - CCPSR;

Rptas.RNC = RNC;
Rptas.NPC = NPnDPL;
Rptas.CNC = CdNc;
}

```

Escolha de um Ligador de Cabo para Instalar um novo Distribuidor Binário de Sinal

Para um poste com residências, lembremos que para distribuir o sinal que chega a ele, primeiro instalamos o supertap respectivo, o qual vai receber sinal do ligador de cabo de um conector de postes. O ligador de cabo que está na saída do supertap, será usado para instalar, um por um, os distribuidores binários de sinal, dependendo do número de vizinhos do poste.

Para um poste sem residências, para distribuir o sinal que chega, utilizaremos o ligador de cabo do conector de postes que recebe sinal para instalar, um por um, os distribuidores binários de sinal.

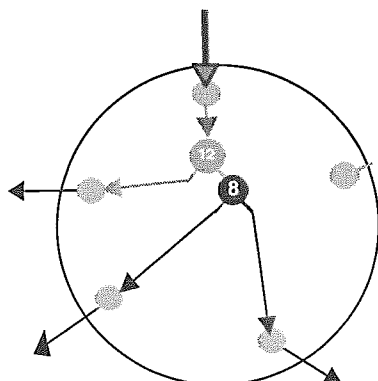


Figura 4.14. Dentre os ligadores de cabo dos acopladores direcionais de 12 e 8 que apontam a um conector de postes vamos escolher um para instalar outro distribuidor binário e assim atender ao conector de postes que falta ligar.

Pode acontecer o caso em que um ligador de cabo envia sinal a um conector de postes, mas a intensidade de sinal que chega no conector não pode ser alterada (por exemplo, este poste tem vizinhos que ficaram ilhados e neste caso não podemos tornar o sinal mais fraco). Neste caso, não instalaremos nenhum distribuidor binário de sinal naquele ligador de cabo.

Definição. Chamaremos de *invariante* a um conector de postes que recebe sinal e que por algum motivo não se possa variar o sinal recebido.

Depois de ter instalado vários distribuidores binários de sinal, que ligador de cabo de um distribuidor binário devemos escolher para a instalação de um novo distribuidor binário? (ver figura 4.14).

Podem-se usar várias heurísticas para responder esta pergunta. Discutiremos algumas mais interessantes.

Heurística PP1

A primeira heurística possível é procurar dentre todos os ligadores de cabo dos distribuidores binários atualmente instalados na árvore de distribuição de sinal e que transmitam sinal aos conectores de postes, aquele que tenha a maior intensidade de sinal de saída na esperança de que este seja suficiente para atender os outros postes não ligados a serem adicionados ao caminho, desde que é importante primeiro ligar os seus vizinhos (figura 4.15).

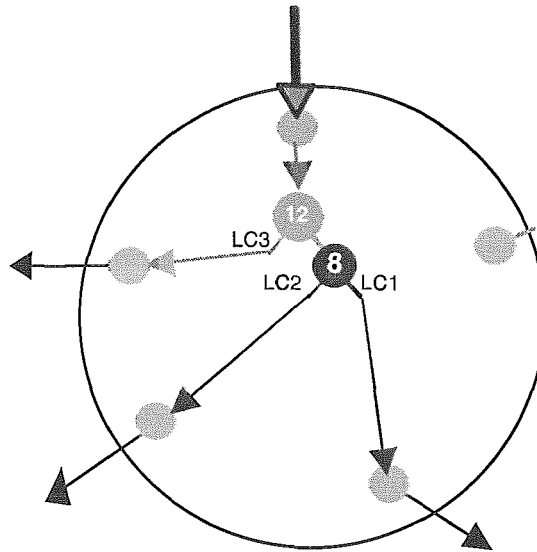


Figura 4.15. Sejam LC1, LC2 e LC3 os ligadores de cabo dos acopladores direcionais de 12 e 8 que enviam sinal para postes vizinhos através dos conectores de postes. Se por LC1 sai a intensidade maior de sinal, então a heurística PP1 escolherá este ligador para instalar o novo distribuidor binário.

Se dois ligadores têm a mesma máxima intensidade de sinal, vamos escolher aquele que transmite a um conector de postes que atende ao menor número de residências. Assim procuramos manter o maior número de residências.

Se existem dois ligadores que têm a mesma intensidade de sinal, e atendem o mesmo número de residências, então vamos escolher aquele que transmite sinal ao conector de postes que atende o caminho PLUSCS de menor comprimento. Isto, porque o caminho de menor comprimento vai precisar normalmente de menor sinal para existir.

Não vamos escolher ligadores de cabo que apontam para conectores *invariantes*.

```
E_Melhor_o_Lgdor_de_Cabo(Ligador de cabo LCaboA,
                          Ligador de cabo LCaboB)
{
  /* esta função indicará se o ligador de cabo LCaboA  */
  /* é melhor que o ligador de cabo LCaboB           */

  É_Melhor_LC = TRUE;                /* variável booleana */
  (Sa, Ra, Ca, AiA) = Sinal entregue, residências, Comprimento do caminho PLUSCS
                          apontado, e propriedade que indica se o conector de postes que
                          recebe sinal de LCaboA é invariante;

  (Sb, Rb, Cb, AiB) = Sinal entregue, residências, Comprimento do caminho PLUSCS
                          apontado, e propriedade que indica se o conector de postes que
                          recebe sinal de LCaboB é invariante;

  Se (AiB = TRUE) então           /* Se o conector de postes que recebe sinal do ligador  */
                                  /* de cabo LCaboB é invariante                       */
      É_Melhor_LC = TRUE
  Senão
      Se (Sa <> Sb) então
          Se (Sa > Sb) então É_Melhor_LC = TRUE
          Senão É_Melhor_LC = FALSE;
      Senão
          {
              Se (Ra <> Rb) então
                  Se (Ra < Rb) então É_Melhor_LC = TRUE
                  Senão É_Melhor_LC = FALSE
              Senão
                  Se (Ca <> Cb) então
                      Se (Ca < Cb) então É_Melhor_LC = TRUE
                      Senão É_Melhor_LC = FALSE
                  Senão É_Melhor_LC = TRUE
          }
  Devolver É_Melhor_LC;
}
```

Para poder fazer a busca, podemos estabelecer que todos os conectores de postes tenham uma propriedade que nos permita saber que ligador de cabo de um distribuidor binário de sinal está transmitindo sinal àquele conector. Desta forma, bas-

tará percorrer a lista circular dos conectores de postes e procurar o ligador de cabo desejado. O algoritmo PP1 é apresentado a seguir:

```
Escolha_dum_Ligador_de_Cabo_para_instalar_um_novo_DS_PP1(poste P)
{
    /* binário de sinal e que o número de vizinhos de P é maior o igual à 3.      */
    ptr_Cx;                               /* Apontador para percorrer os outros      */
                                           /* conectores de postes na lista circular. */
    MMLCBO;                               /* identificará o melhor ligador de      */
                                           /* cabo dos distribuidores binários.     */
    Seja CNTOR_Ep o conector de postes do poste P pelo que entra sinal nele.

    ptr_Cx = seguinte_conector(CNTOR_Ep);

    enquanto (E_Invariante(ptr_Cx) ou Não Recebe_Sinal(ptr_Cx)) fazer
    {
        ptr_Cx = seguinte_conector(ptr_Cx);
    }
    MMLCBO = Ligador_de_cabo_que_da_sinal_a(ptr_Cx);

    Enquanto (ptr_cx <> CNTOR_Ep) fazer
    {
        ptr_Cx = seguinte_conector(ptr_Cx);
        LCabo = Ligador_de_cabo_que_da_sinal_a(ptr_Cx);
        Se E_Melhor_o_Lgdor_de_Cabo(LCabo, MMLCBO) então
        {
            MMLCBO = LCabo;
        }
    }
    devolver MMLCBO;
}
```

Heurística PP2

Para uma segunda heurística, seja **CNTOR_Sp** o conector de postes pelo qual o sinal sai inicialmente. Se **LCabo** é o ligador de cabo que transmite sinal àquele conector de postes, então vamos usá-lo sempre para instalar o novo distribuidor binário de sinal. O seguinte é o algoritmo PP2:

Escolha_dum_Ligador_de_Cabo_para_instalar_um_novo_DS_PP2(poste P)

```
{  
  
    Seja CNTOR_Sp o primeiro conector de postes do poste P pelo que se  
    transmitia sinal para outro poste vizinho.  
  
    ptr_Cx = CNTOR_Sp;  
    LCabo = Ligador_de_cabo_que_da_sinal_a(ptr_Cx);  
    devolver LCabo;  
}
```

A vantagem deste algoritmo é que a escolha é feita rapidamente. Como ao indicar um caminho PLUSCS no grafo G , em cada poste vai se indicar o primeiro conector de postes pelo qual há saída de sinal, então este algoritmo vai ter uma complexidade $O(1)$. Porém sua utilização frequente pode levar a uma perda de sinal significativa fazendo com que o sinal recebido pelo conector **CNTOR_Sp** fique muito fraco. Para um poste que tem poucos vizinhos, este algoritmo é suficiente, e isto acontece normalmente.

Não acreditamos que uma escolha aleatória do ligador de cabo seja uma boa opção. Porém, uma boa opção poderia ser a escolha aleatória do ligador de cabo em um subconjunto de postes vizinhos ligados.

Se um poste fosse considerado como um autômato, o processo de escolha poderia ser bastante interessante, pois cada poste poderia escolher o seu conector segundo suas necessidades e utilizando algoritmos distribuídos. Mas esta discussão está fora do escopo desta tese.

Melhor Conector de Postes para uma Intensidade de Sinal

Seja **CJCP** um conjunto de conectores de postes de um poste P e **CNTOR_Ap** um elemento daquele conjunto. Lembremos que com o algoritmo **Da_Melhor_Caminho_Desde** podemos pesquisar o melhor caminho PLUSCS que se inicie no poste apontado por **CNTOR_Ap**, com uma determinada intensidade de sinal, digamos I que receberia aquele conector. Assim, para cada conector de postes de **CJCP**, vamos poder conhecer o melhor caminho PLUSCS que parte do poste que ele aponta, com uma intensidade de sinal I que iria sair de aquele conector. Dentre todos

aqueles conectores, diremos que o melhor é aquele que dá o melhor caminho de PLUSCS. O algoritmo que vai fazer esta tarefa é descrito a seguir.

```

Melhor_Cntor_de_postes_e_Cmho( Sinal que recebe o conector Ap      I,
                                Conectores de postes                CJCP,
                                BYREF Conector de postes           MConector,
                                BYREF Caminho apontado por MConector Cmax_Bp)
{
  Seja CNTOR_Ap um elemento de CJCP;
  MELHOR_CNTOR = CNTOR_Ap;
  Para cada CNTOR_Bp de CJCP diferente de CNTOR_Ap seja
  {
    Cmax_Bp = Da_Melhor_Caminho_Desde( CNTOR_Bp,
                                        FALSE;
                                        I);
    Se o caminho de postes ligados C apontado por CNTOR_Bp é melhor que o
    caminho de postes apontado por MELHOR_CNTOR, então
    {
      MELHOR_CNTOR = CNTOR_Bp
    }
  }
  MConector = MELHOR_CNTOR;
}

```

O Melhor Caminho PLUSCS para um Distribuidor Binário de Sinal num Conjunto de Conectores de Postes

Seja **CJCP** um conjunto de conectores de postes de um poste *P*. Vamos supor que conhecemos o ligador de cabo **LCabo** onde vamos instalar um novo distribuidor binário e **CNTOR_Sp** é o conector de postes apontado por aquele ligador de cabo antes de instalar nele o distribuidor. Seja **I** o sinal que sai de **LCabo**.

Vamos supor que um distribuidor binário vai receber o sinal **I**. Então, com o sinal mais fraco do distribuidor, vamos procurar o melhor conector de postes de **CJCP** usando o algoritmo anterior. A saída mais forte do distribuidor ficará reservada para o caminho PLUSCS apontado por **CNTOR_Sp**. Neste caminho, deveremos conhecer os novos números de residências, número de postes, e seu comprimento. O procedimento que faz esta pesquisa para um distribuidor binário é:

```

Melhores_ligações_dum_DB( Distribuidor binário DB,
                          Sinal que receberia o distribuidor I,
                          Conector de postes CNTOR_Sp
                          Conjunto de conectores de postes CJCP,
                          BYREF var. do Tipo_de_dado_dum_Caminho MCF,
                          BYREF var. do Tipo_Dados_Dum_Caminho DCM)
{
  I_fraca = Sinal na saída mais atenuada de DB;
  I_forte = Sinal na saída mais forte de DB;
  Variável Dados_CM_Sp do tipo Tipo_Dados_Dum_Caminho;
  CTOR = primer conector do conjunto CJCP;
  MC_CTOR = Da_Melhor_Caminho_Desde( CTOR,
                                     Receberia_sinal_de_outro_Poste;
                                     I_fraca);

  Melhor_Caminho_para_sinal_fraca = MC_CTOR;
  Melhor_Conector_de_Postes;
  Receberia_sinal_de_outro_Poste = FALSE; /* CTOR receberia sinal de um */
                                           /* conector de postes de P */

  Para cada conector de postes CN <> CNTOR fazer
  {
    MC_CN = Da_Melhor_Caminho_Desde( CN,
                                     Receberia_sinal_de_outro_Poste;
                                     I_fraca);

    Se o caminho em MC_CN é melhor que o caminho em MC_CTOR então
    {
      Melhor_Caminho_para_sinal_fraca = MC_CN;
      Melhor_Conector_de_Postes = CN;
    }
  }

  /* pesquisaremos o caminho apontado por Sp */
  Dados_CM_Sp = Pesquisa_Caminho_Sinalado_por(CNTR_Sp, I_forte);

  MCF = MC_CN;
  DCM = Dados_CM_Sp;
}

```

O Melhor Distribuidor Binário de Sinal num Conjunto de Conectores de Postes, para Instalar num Ligador de Cabo de Outro Distribuidor de Sinal

Seja **CJCP** um conjunto de conectores de postes de um poste *P*. Vamos supor que conhecemos o ligador de cabo **LC** onde vamos instalar um novo distribuidor binário e **Sp** é o conector de postes apontado por aquele ligador de cabo antes de instalar nele um distribuidor binário. Para cada distribuidor binário de sinal, vamos procurar os melhores resultados das ligações que poderíamos fazer instalando ele, utilizando para isso o algoritmo **Melhores_ligações_dum_DB**.

Definição. Diremos que um distribuidor binário é *elegível para ser instalado em LC*, se ao ser instalado em **LC** ligando sua saída mais forte a **Sp**, então o caminho apontado por **Sp** tem um número de residências diferente de 0. No caso de um divisor, bastará que a saída ligada a **Sp** cumpra com esta condição.

Definição. Sejam DB1 e DB2 dois distribuidores binários de sinal elegíveis para ser instalados em **LC**. Diremos que DB1 é melhor que DB2, se com DB1 poderemos enviar sinal a maior número de residências que com DB2 ao instalar cada um em **LC**. Se os dois distribuidores binários empatam neste valor, diremos que é *melhor* aquele que resulte no maior número de residências no caminho apontado pelo conector de postes **Sp**. Se ainda assim temos empate, então diremos que são *indiferentes*.

Usaremos a seguinte estrutura de dados com dois campos que chamaremos:

Tipo_Dst_Cmho_Rs:

1. Campo que representa um Distribuidor binário de sinal **Ds**;
2. Campo do tipo Tipo_de_dado_dum_Caminho **Caminho**;

O segundo campo é para indicar o melhor caminho que pode ser ligado usando a saída mais fraca do distribuidor Ds.

O nome da função que retorna este tipo de dado é:

Melhor_DSTRB_e_Cmho_para_conec_em (
Ligador de cabo onde instalaremos **LC**,
conjunto de conectores de postes a ligar **CJCP**,
BYREF indicará se existe o melhor dist. Bin. **Existe_DB)**

```

{ Existe_DB = FALSE;
  I = Sinal em LC;
  DB1 = primeiro distribuidor de sinal;
  CNTOR_Sp = Conector apontado por o ligador de cabo LC;

  /* procuraremos o primeiro distribuidor binário de sinal que seja elegível */
  Repetir
  {
    Melhores_ligações_dum_DB(DB1, I, CNTOR_Sp, CJCP, MCF1, DCM1 );
    /* os argumetos trabalham como os parâmetros do procedimento */
    Se DB é elegível então Existe_DB = TRUE;
    DB1 = seguinte distribuidor de sinal;
  }
  até ( Existe_DB ou (não se tem mais distribuidores a probar));
  Melhor_DB = DB1; /* o melhor dist. Bin. é DB1 por enquanto. */
  Se Existe_DB Então
  {
    Enquanto (fique um distribuidor binário por probar)
    {
      Seja DB2 o seguinte distribuidor binário a DB1;
      Melhores_ligações_dum_DB(DB2, I, CNTOR_Sp, CJCP, MCF2, DCM2 );
      Se DB2 é melhor que DB1 então Melhor_DB = DB2;
    }
  }
  senão
  {
    devolveremos um valor para o campo DS qualquer;
    devolveremos o campo Caminho igual a  $\emptyset$ ;
  }
}

```

Procura de Postes Ilhados no Grafo $G = (P, L)$

Seja o grafo $G = (P, L)$, com P seu conjunto de postes e L sua coleção de arestas. Se P e Q são dois postes de P , então vamos supor que existe um canal de comunicação bidirecional entre P e Q se e somente se é certo, $L(P, Q)$ ou $L(Q, P)$. A coleção daqueles canais de comunicação vamos chamar de \bar{L} .

Assim vamos definir um novo grafo $G' = (P, \bar{L})$. Neste grafo, para procurar os postes ilhados, vamos executar o algoritmo distribuído que faça o seguinte:

1. Todos os postes não ligados que estão na fronteira de P e os terminais, vão enviar uma mensagem m a todos os seus vizinhos através dos canais respectivos do grafo G' . Após o envio da mensagem m cada poste enviará seu identificador (os canais devem ser FIFO).
2. Se um poste está ligado e recebe m então não re-envia m , mas envia seu identificador a todos os seus vizinhos.
3. Se um poste está ligado e a primeira mensagem recebida é um identificador, então ele envia seu identificador a todos os seus vizinhos.
4. Se o poste não está ligado e recebe a mensagem m pela primeira vez, então re-enviará esta mensagem a seus vizinhos através dos canais de G' e ignorará as próximas chegadas de m . Após receber a primeira mensagem m enviará seu identificador a todos os seus vizinhos.
5. Se o poste não está ligado e a primeira mensagem que recebe é um identificador, então enviará seu identificador a todos os seus vizinhos.

O processo termina quando todos os postes receberem os identificadores de todos os seus vizinhos. Os postes ilhados serão aqueles que receberam somente os identificadores de seus vizinhos.

Ligando os Postes Vizinhos de um Poste

Seja P um poste que recebe sinal por um de seus conectores de postes Ep . Seja LC o ligador de cabo de Ep que tem sinal para distribuir (o outro, lembremos, vai apontar ao conector de postes do poste que envia sinal a P).

Vamos supor que existe sinal suficiente que sai de P através de um de seus conectores de postes Sp . Por enquanto nenhum outro conector de P recebe sinal.

Se P não tem residências, então o ligador de cabo a usar para a instalação da árvore binária de distribuição de sinal, será LC . Também vamos supor que inicialmente, LC aponta a Sp .

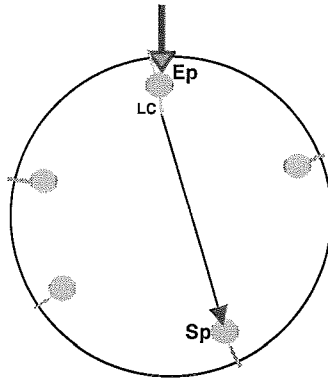


Figura 4.16. No ligador de cabo LC vai se instalar a árvore binária de distribuição de sinal.

Se P tem residências, então vamos instalar o supertap para que receba sinal de **LC**. O ligador de cabo da saída do supertap, que será representado agora por **LC**, vai ser então usado para a instalação da árvore de distribuição de sinal, se existe em **LC** intensidade suficiente de sinal. Mas antes de instalar qualquer distribuidor binário, vamos supor que **LC** aponta para **Sp**, simulando o envio de sinal àquele conector de postes. Assim, usando **Sp** poderemos conhecer o ligador de cabo que envia sinal a ele.

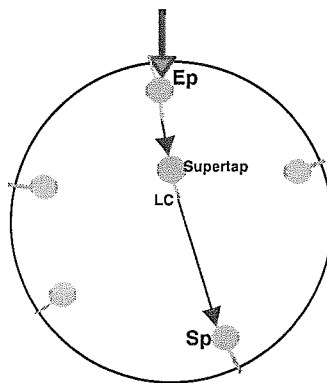


Figura 4.17. No ligador de cabo LC vai se instalar a árvore binária de distribuição de sinal. Agora ele é a saída do supertap.

Só utilizaremos a árvore de distribuição de sinal se o número de vizinhos de P segundo L é maior que dois.

Alguns dos conectores de postes podem estar apontando a postes ilhados, a esta coleção de conectores vamos chamar **C_CP_ap_Ilhados**. Se isto acontece, é importante ligá-los com prioridade, pois sempre vamos procurar que os postes que ficam perto de *P* fiquem ligados. Vamos procurar o melhor ligador de cabo para instalar um distribuidor binário usando o algoritmo

Escolha_dum_Ligador_de_Cabo_para_instalar_um_novo_DS_PP1(poste P)

ou o algoritmo

Escolha_dum_Ligador_de_Cabo_para_instalar_um_novo_DS_PP2(poste P)

Este ligador de cabo vamos chamar de **LC**. Ligaremos a este conector o distribuidor binário que melhor distribua o sinal de **LC**, e para encontrá-lo usaremos o algoritmo **Melhor_DSTRB_e_Cmho_para_conec_em**.

Depois de escolher aquele distribuidor binário, vamos tirar do conjunto **C_CP_ap_Ilhados** aquele conector de postes que agora está ligado e indicaremos ele como invariante.

Repetiremos o processo de escolher o ligador de cabo, selecionar um distribuidor binário para ligar um conector de postes do conjunto de conectores de postes **C_CP_ap_Ilhados**, e tirar aquele conector deste conjunto (indicando que é invariante), até que ele fique vazio ou não tenhamos sinal suficiente para ligar outro conector de **C_CP_ap_Ilhados** (isto acontecerá quando seja certo que o melhor distribuidor binário a usar somente pode atender na sua saída mais atenuada um conjunto vazio de postes).

```
procedimento Ligar_um_Conj_de_DISTR_ap_Ilhados( Ligador_de_Cabo LC,
                                                BYREF Conjunto CJCP_ap_Ilhados )
{
  /* CJCP_ap_Ilhados. Este é o conjunto dos Conectores de Postes que apontam a */
  /* postes ilhados. LC é o ligador de cabo onde se instalará um novo          */
  /* distribuidor binário.                                                    */

  variável booleana  Pode_se_ligar_Outro_Poste ;
  /* Indicaré se ainda podemos ligar outro poste. Pode ser FALSE por exemplo, se */
  /* não temos intensidade suficiente de sinal.                                */

  variável booleana  Existe_DB;
  /* indicará se existe um distribuidor binário elegível para instalar          */
}
```

variável MDSTRB_CMNHO que é do tipo de dado Tipo_Dst_Cmho_Rs;

I = Intensidade de sinal no ligador de cabo LC;

Pode_se_ligar_Outro_Poste = TRUE;

P = poste onde vamos a fazer as conexões;

Enquanto ((CJCP_ap_Ilhados <> \emptyset) && Pode_se_ligar_Outro_Poste) fazer

```
/* enquanto não este vazio o conjunto CJCP_ap_Ilhados */
/* e Pode-se ligar outro poste fazer... */
{
    /* procurar o distribuidor binário com o que obteremos o melhor */
    /* caminho PLUSCS usando só o ligador de cabo de sua saída */
    /* mais atenuada (fraca), e que vai nos permitir atender ao maior número */
    /* de residências no caminho apontado pelo conector de postes */
    /* inicialmente apontado por LC. */
}
```

MDSTRB_CMNHO = Melhor_DSTRB_e_Cmho_para_conec_em (LC,
C_DIST_ap_Ilhados, Existe_DB);

```
/* se o melhor caminho encontrado é o conjunto vazio para os conectores que */
/* estão no conjunto CJCP_ap_Ilhados, então não poderemos ligar mais */
/* postes apontados por os elementos de aquele conjunto. */
/* Também se */
```

Se (o caminho de MDSTRB_CMNHO é igual que \emptyset) ou Não Existe_DB então

```
{
    Pode_se_ligar_Outro_Poste = FALSE;
}
```

Senão

```
{
    instala_o_distribuidor_no_ligador_de_cabo(LC, MDSTRB_CMNHO);
    /* Seja Sp o Conector de postes apontado por LC antes da instalação */
    /* do distribuidor binário de sinal. */
    /* Este procedimento: instala_o_distribuidor_no_ligador_de_cabo, */
    /* tem que marcar os postes de MDSTRB_CMNHO como que já */
    /* recebem sinal, no grafo G = (P, L). Também deverá asignar um */
    /* mesmo número a cada poste de aquele caminho, que é o seguinte do */
    /* número dado a cada um dos postes do caminho anteriormente */
    /* marcado no grafo G =(P, L). Além disso, deverá atualizar o */
    /* caminho de postes apontado por Sp. */
}
```

```
/* Já utilizamos o ligador LC. Agoura deveremos procurar outro */
/* ligador de cabo de um dos distribuidores já instalados e que */
/* aponte a um conector de postes dando para ele sinal. */
```

LC = Escolha_dum_Ligador_de_Cabo_para_instalar_um_novo_DS_PP1(poste P);

I = Intensidade de sinal no ligador de cabo LC;

```
/* CPUS representará o conector do poste P utilizado pelo */
/* distribuidor binário depois de instalado */
```

CPUS = Conector de postes de P que aponta ao caminho de postes definido em MDSTRB_CMNHO

```
/* o conector de poste CPUS não deve ser tomado em conta para */
/* procurar novos caminhos de postes ligados usando só cabo. */
/* deve ficar como invariante. */
CPUS.Invariante = TRUE;

C_DIST_ap_Ilhados = CJCP_ap_Ilhados - { CPUS };
gb_Existente_um_Poste_Expandible = TRUE;
/* indicará que existe pelo menos outro poste no grafo G= (P, L) */
/* que devermos visitar a fim de ligar seus vizinhos. */
}
}
Se (CJCP_ap_Ilhados <> Ø )
{
    Eliminar_Camino(CJCP_ap_Ilhados);
    /* para livrar a memória. */
}
}
```

Depois de ligar os conectores que apontam a postes ilhados, passamos a ligar os conectores de postes que apontam a postes *livres*. Estes conectores são aqueles que apontam a postes que não são ilhados e não estão ligados.

Para isto, selecionaremos o ligador de cabo para instalar o novo distribuidor binário do mesmo jeito como foi feito no caso dos postes ilhados. Vamos chamá-lo de **LC**. Seja **Sp** o conector de postes que recebe sinal de **LC**, e **CJCP_ap_Livres** o conjunto de conectores de postes que apontam a postes livres. Vamos escolher o distribuidor binário que melhor possa distribuir o sinal de **LC**, usando o seu sinal mais atenuado para enviar sinal a um dos conectores do conjunto **CJCP_ap_Livres**, e o outro para enviar sinal ao conector de postes apontado por **LC**. Uma vez encontrado aquele distribuidor, vamos marcar o novo caminho de postes encontrado no grafo $G = (P, L)$, e revisar o caminho apontado por **Sp**.

Pode acontecer que apareçam conectores de postes de *P* que apontam a postes ilhados mas que antes apontavam a postes livres. Para isto, deveremos procurar os postes ilhados. Após, ligaremos aqueles conectores de postes com o algoritmo anterior e continuaremos com os conectores de postes que apontam a postes livres. Repetiremos este processo até que o conjunto de postes **CJCP_ap_Livres** fique vazio.

O seguinte algoritmo tentará ligar um conjunto de distribuidores que apontam a postes livres:

```

Procedimento Ligar_um_Conj_de_DISTR_ap_Livres(
    Ligador_de_Cabo LC,
    BYREF Conjunto CJCP_ap_Livres
)
{
    variável MDSTRB_CMNHO que é do tipo de dado Tipo_Dst_Cmho_Rs;
    /* Indicaré se ainda podemos ligar outro poste. Pode ser FALSE por exemplo, se */
    /* não temos intensidade suficiente de sinal. */

    Pode_se_ligar_Outro_Poste = TRUE;

    variável booleana Pode_se_ligar_Outro_Poste ;
    /* Indicaré se ainda podemos ligar outro poste. Pode ser FALSE por exemplo, se */
    /* não temos intensidade suficiente de sinal. */

    variável booleana Existe_DB;
    /* indicaré se existe um distribuidor binário elegível para instalar */

    I = Intensidade de sinal no ligador de cabo LC;
    P = poste onde vamos a fazer as conexões;
    Enquanto ((CJCP_ap_Livres <> Ø) && Pode_se_ligar_Outro_Poste) fazer
    {
        MDSTRB_CMNHO = MIhor_DSTRB_e_Cmho (LC, CJCP_ap_Livres, Existe_DB);
        Se ( (o caminho em MDSTRB_CMNHO é igual que Ø) ou não Existe_DB) então
        {
            Pode_se_ligar_Outro_Poste = FALSE;
        }
        Senão
        {
            instala_o_distribuidor_no_Ligador_de_cabo(LC, MDSTRB_CMNHO);
            /* Seja Sp o Conector de postes apontado por LC antes da instalação */
            /* do distribuidor binário de sinal. */
            /* Este procedimento: instala_o_distribuidor_no_Ligador_de_cabo, */
            /* tem que marcar os postes do caminho de MDSTRB_CMNHO como */
            /* que já recebem sinal, no grafo G = (P, L). Também deverá assinar */
            /* um número a cada poste de aquele caminho, que é o seguinte */
            /* do número dado a cada um dos postes do caminho anteriormente */
            /* marcado no grafo G = (P, L). Além disso, deverá atualizar o */
            /* caminho de postes apontado por Sp. */

            /* CPUS representará o conector do poste P utilizado pelo */
            /* distribuidor binário depois de instalado. */

            CPUS = Conector de postes de P que aponta ao caminho de postes definido
                em MDSTRB_CMNHO;
        }
    }
}

```

```

/* o conector de poste CPUS não deve ser tomado em conta para */
/* procurar novos caminhos de postes ligados usando só cabo. */

```

```

C_DIST_ap_Livres = CJCP_ap_Livres - { CPUS };
gb_Existem_um_Postes_Expandible = TRUE;
/* indicará que existe pelo menos outro poste no grafo G = (P, L) */
/* que devermos visitar a fim de ligar seus vizinhos. */
/* Já utilizamos o ligador LC. Agora devermos procurar outro */
/* ligador de cabo de um dos distribuidores já instalados e que */
/* aponte a um conector de postes dando para ele sinal. */

```

```

LC = Escolha_dum_Ligador_de_Cabo_para_instalar_um_novo_DS_PP1(poste P);
I = Intensidade de sinal no ligador de cabo LC;
/* A aparição de um caminho PLUSCS, */
/* pode implicar que apareçam novos postes ilhados . Logo, */
/* precisamos executar o nosso algoritmo de busca de postes */
/* ilhados no grafo G = (P, L). */

```

```

Pesquisar_Postes_Ilhados no grafo G = (P, L);

```

```

/* Procuraremos os conectores de postes de CJCP_ap_Livres */
/* que apontem a postes ilhados. */
C_DIST_ap_Ilhados = Procurar_Conectores_ap_Ilhados_em(CJCP_ap_Livres);
Se (C_DIST_ap_Ilhados <> ∅ ) então
{
    Ligar_um_Conj_de_DISTR_ap_Ilhados(LC, C_DIST_ap_Ilhados);
    CJCP_ap_Livres = CJCP_ap_Livres - C_DIST_ap_Ilhados;
}
}
}

```

Se a entrada de P não tem sinal suficiente para atender nenhum conector de postes, mas tem residências e o sinal é suficiente para atender estas residências, então só há necessidade de instalar o supertap respectivo. Se não tem residências, então desligamos P do caminho, retrocedendo ao poste que enviava sinal a ele e modificando o conector de postes respectivo no seu campo **Cabo_Ligado** fazendo com que ele seja FALSE. Assim, o poste não pertencerá ao caminho.

Se o número de vizinhos de P é maior que 2, P não tem residências e qualquer tentativa de ligar um vizinho de P fracassou, então devermos desligar P da rede, retrocedendo ao poste que envia sinal a ele e modificando o conector de postes respectivo.

Para ligar os vizinhos de P procurando ligar o maior número de residências, vamos usar o seguinte algoritmo, onde E_p , é o conector de postes de P que recebe sinal e S_p é o primeiro conector de postes pelo que sai sinal de P .

Procedimento Ligar_os_Conectores_de_Postes_dum_Poste(P)

```

{
  C_CP_ap_Ilhados;
    /* Conjunto de Conectores de Postes que apontam a postes ilhados */
  C_CP_ap_Livres;
    /* Conjunto de Conectores de Postes que apontam a postes livres. */

  Variável booleana Llivr = FALSE; /* vai nos indicar ligo-se outro poste. */
  Variável booleana Lilha = FALSE; /* vai nos indicar ligo-se outro poste. */

  /* Precisaremos que o poste P tenha duas propriedades: uma que indique */
  /* o conector de postes pelo que recebe sinal, e outro que indique o primeiro */
  /* conector de postes pelo que sai sinal. */
  CNTOR_Ep = Conector de postes pelo que entra sinal em P;
  CNTOR_Sp = Primeiro conector de postes pelo que sai sinal de P;
  LC = Ligador de cabo do CNTOR_Ep; /* este conector já deve apontar a Sp. */

  Se (CNTOR_Sp = NULL) então
    /* Se P inicialmente não tem saída de sinal, então */
    {
      Se (P.Residências > 0) então
        {
          Indica_o_Supertap(P.Residências, I, STAP, Y)
          /* Y assumirá o sinal na saída do supertap */
          Instalar_SuperTap(LC, STAP, Y);
          /* Instalar_SuperTap botará o supertap definitivo no ligador LC. */
          /* especificando os diversos tipos de taps que formam o supertap. */
          /* Assim, a saída do supertap irá ter um sinal Y. */

        }
      Senão
        {
          CNTOR_Sq = Conector de postes que envia sinal a CNTOR_Ep;
          Estabelecer que CNTOR_Sq não enviará sinal ao CNTOR_Ep;
        }
    }
  Senão
    {
      Pesquisar_Postes_Ilhados no grafo G = (P, L);

      /* pesquisar se P tem TAP <=> seu número de residências é positivo */
      Poste_Inhabilitado = FALSE;
    }
}

```

```

Se (P.Residências > 0) então /* se o poste tem várias residências para atender */
{
    Indica_o_Supertap(P.Residências, I, STAP, Y)
    Instalar_SuperTap(LC, STAP, Y);
    /* Instalar_SuperTap botará o supertap definitivo no ligador LC. */
    /* Assim, a saída do supertap irá ter um sinal Y. */
}

/* procura os conectores de postes que apontam a postes ilhados */
C_CP_ap_Ilhados = Procura_Conect_de_Postes_ap_Ilhados_do_Poste(P);

/* procura os conectores de postes que apontam a postes livres */
C_CP_ap_Livres = Procura_Conect_de_Postes_ap_livres_do_Poste(P);

/* Vamos tentar ligar os conectores de postes, */
/* que apontam a postes ilhados. */
LC = Escolha_dum_Ligador_de_Cabo_para_instalar_um_novo_DS_PP1(poste P);
/* LC irá a ser o ligador do cabo escolhido para inserir */
/* um novo distribuidor se for necessário. */

/* Vamos tentar ligar os conectores de postes que apontam a postes ilhados. */
Ligar_um_Conj_de_CNTP_ap_pIlhados(LC, C_CP_ap_Ilhados);

/* Vamos tentar ligar os conectores de postes que apontam a postes livres. */
Ligar_um_Conj_de_CNTP_ap_pLivres(LC, C_CP_ap_Livres);
/* Marcar o poste para indicar que nele foi aplicada a função */
/* Ligar_os_Conectores_de_Postes_dum_Poste */
/* conhecida também como LCPP */

P.foi_aplicado_o_procedimento_LCPP_no_poste = TRUE;

Se não existe um conector de postes de P que envie sinal, e
o poste não tem residências Então
{
    CNTOR_Sq = Conector de postes que envia sinal a CNTOR_Ep;
    Estabelecer que CNTOR_Sq não enviará sinal ao CNTOR_Ep;
}
}

```

Ligando os Postes de um Grafo a partir de um Backbone Inicial (algoritmo de ligação de CATV chamado JCCZ)

A seguir apresentamos o algoritmo principal do nosso trabalho que utiliza os outros algoritmos apresentados até então.

Lembremos o grafo $G = (P, L)$. Para ligar todos (ou quase todos) os postes de um grafo a partir de um poste P de P , vamos instalar em P o amplificador e depois procurar o melhor caminho de postes ligados usando só cabo e supertaps a partir daquele poste. Indicaremos no grafo G aquele caminho (e assinalaremos para cada poste o número 1, para indicar que pertencem ao primeiro caminho indicado em G). Seja G_P o nome do grafo formado pelos postes ligados a partir de P e as arestas dirigidas que indicam ligações. Por enquanto, G_P é aquele caminho.

O melhor caminho de postes que parte de B seria

BCGJHDE

se tem o maior número de residências. Assumindo certo isto, as setas



e os postes mencionados, indicaram o grafo G_P .

Foi assumido que tivemos sinal suficiente para fazer estas ligações.

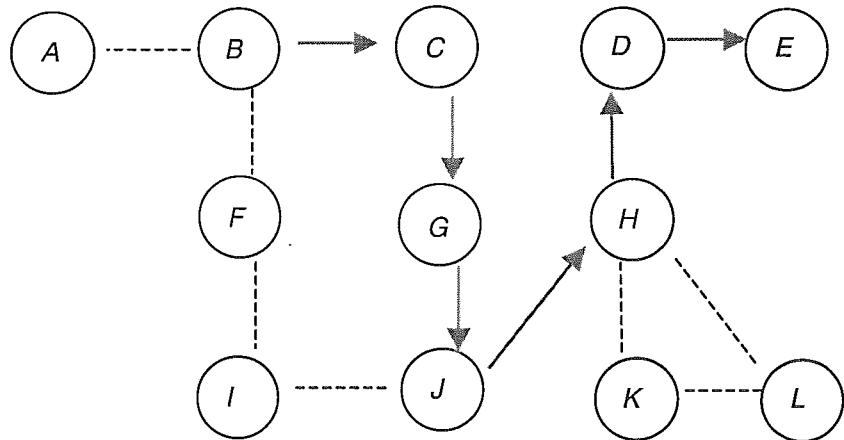


Figura 4.18. Em procura do melhor caminho de postes que parte de B .

A partir do poste P , vamos percorrer o grafo G_P em pré-ordem, aplicando os seguintes passos.

P1. Quando visitamos um poste Q de G_P , e neste ainda não aplicamos o procedimento:

Ligar_os_Conectores_de_Postes_dum_Poste(P)

então aplicaremos este procedimento àquele poste e seguiremos com o poste seguinte apontado pelo conector de postes S_q , se este conector existir. Se S_q não existir, então o algoritmo termina com um caminho de postes ligados usando somente cabo e supertaps.

Se no poste Q já se aplicou aquele procedimento, então vamos continuar com os outros postes não apontados nem por S_q , nem pelo conector E_q do poste Q que recebia sinal.

Continuaremos fazendo isto, até que todos os caminhos de postes ligados usando só cabo e supertaps que existiam antes de iniciar a busca em P , sejam visitados.

P2. Seja G_P o novo grafo de postes ligados depois de aplicar P1. Se existe apenas um novo caminho de postes ligados usando só cabo e supertaps, então aplicaremos novamente P1. Senão existe, então simplesmente o processo termina com as ligações necessárias no grafo G .

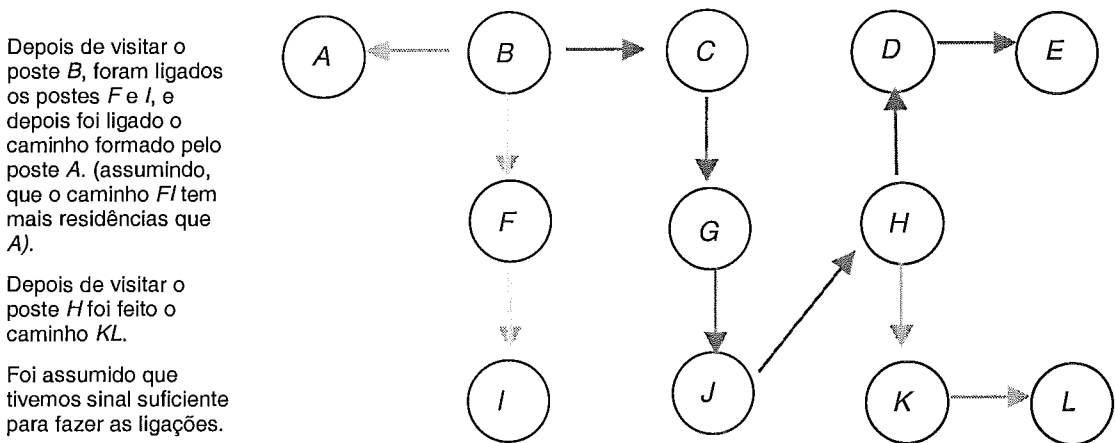


Figura 4.19. Ligando os vizinhos de um poste.

Os seguintes, formam o algoritmo apresentado nos passos P1 e P2 de uma forma mais detalhada:

Ligar_Postes_com_os_Conectores_de_Postes_a_partir_do_Poste(Poste P)

```

Se (não foi_aplicado_o_procedimento_LCPP_no_poste(P))
  /* se não foi aplicado o procedimento */
  /* Ligar_os_Conectores_de_Postes_dum_Poste(P) */
  {
    Ligar_os_Conectores_de_Postes_dum_Poste(P);
  }
senão
  {
    /* ligar segundo cada um dos caminhos que partem pelos */
    /* conectores de postes... */
    pt_CNTOR = seguinte conector de CNTOR_Ep, na lista dos conectores de P;
  }

```



```

gb_Exista_um_Poste_Expandible = TRUE;
Enquanto gb_Existe_um_Poste_Expandible
{
  gb_Existe_um_Poste_Expandible = FALSE;
  /* ligaremos os postes com os conectores de postes a partir */
  /* do poste P. */
  Ligar_Postes_com_os_Conectores_de_Postes_a_partir_do_Poste(P);
}
}

```

Postes Obrigatórios a Conectar

Os postes terminais são um tipo especial de postes que devem receber sinal de qualquer forma ao fazer as ligações da microcélula. Fazem parte do conjunto de postes críticos.

Se depois de aplicar o algoritmo JCCZ, restar postes terminais que não recebem sinal, então deveremos abandonar a solução produzida pelo algoritmo JCCZ e executá-lo novamente instalando o amplificador em outro poste.

Repetiremos isto até que as ligações feitas a partir de um poste, sejam satisfatórias, ou seja, de forma que o sinal do amplificador seja suficiente para todos os postes terminais do grafo.

Poderíamos tentar implementar uma heurística que ao ligar os postes vizinhos de um poste P , primeiro ligue os vizinhos terminais deste poste e após continue com seus vizinhos ilhados e livres. Se com o sinal que chega a P , não pode-se ligar todos os vizinhos terminais de P , então o processo de ligações iniciadas no poste onde se instalou o amplificador deverá culminar em *fracasso* e deveremos instalar o amplificador em outro poste.

Deve-se observar que isto não garante que todos os postes terminais serão ligados.

O seguinte algoritmo **Ligar_os_vizinhos_terminais_do**, ligará os vizinhos terminais de um poste P , a partir de um ligador de cabo **LC**. Ele é muito similar ao utilizado para ligar postes ilhados.

Seja **CJCP_ap_Terminais** o conjunto dos conectores de postes apontando a postes terminais que têm residências e não estão ligados.

Seja **CNTOR_Sp** o conector de postes pelo qual inicialmente sai sinal do poste *P*.

Para ligar todos os conectores deste conjunto, vamos procurar o melhor ligador de cabo para instalar um distribuidor binário usando o algoritmo:

Escolha_dum_Ligador_de_Cabo_para_instalar_um_novo_DS_PP1(poste P)

ou o algoritmo:

Escolha_dum_Ligador_de_Cabo_para_instalar_um_novo_DS_PP2(poste P)

Este ligador de cabo vamos chamar de **LC**. Ligaremos a este ligador o distribuidor binário que melhor distribua o sinal de **LC**, e para encontrá-lo usaremos o algoritmo **Melhor_DSTRB_e_Cmho_para_conec_em**.

Depois de escolher aquele distribuidor binário, vamos tirar do conjunto **CJCP_ap_Terminais** aquele conector de postes que agora está ligado.

Repetiremos o processo de escolher o ligador de cabo, seleccionar um distribuidor binário para ligar um conector de postes do conjunto de conectores de postes **CJCP_ap_Terminais**, e tirar aquele conector deste conjunto, até que ele fique vazio ou não possamos instalar outro distribuidor binário *elegível* para ligar um dos conectores do conjunto **C_CP_ap_Terminais**.

Se ao final o conjunto não é vazio, deveremos avaliar os seguintes casos:

Se (o número de elementos de **CJCP_ap_Terminais** é 1) Então

Se **CNTOR_Sp** não aponta para um poste terminal, então

LC = ligador de cabo que envia sinal ao **CNTOR_Sp**;

Cx = elemento de **CJCP_ap_Terminais**;

Se o sinal que sai de **LC** é suficiente para enviar sinal ao poste apontado por **Cx**, então desligar **CNTOR_Sp** e ligar o conector **Cx** para que receba sinal de **LC**. Assim, o poste terminal apontado por **Cx** será ligado.

Se o sinal que sai de **LC** não é suficiente para atender ao poste apontado por **Cx**, então as ligações feitas a partir

do poste onde se instalou o amplificador deverão ser paradas e se instalará o amplificador em outro poste;

Senão

As ligações feitas a partir do poste onde se instalou o amplificador deverão ser paradas e se instalará o amplificador em outro poste;

Senão

As ligações feitas a partir do poste onde se instalou o amplificador deverão ser paradas e se instalará o amplificador em outro poste;

A seguir, o algoritmo em uma versão mais detalhada:

```
procedimento Ligar_um_Conj_de_DISTR_ap_Terminalis(
                                Ligador_de_Cabo LC,
                                BYREF Conjunto CJCP_ap_Terminalis ,
                                BYREF var. booleana Parar_Ligações_G)
{
    /* CJCP_ap_Terminalis. Este é o conjunto dos Conectores de Postes que apontam a */
    /* postes terminais com residências e não estão conectados.                */
    /* LC é o ligador de cabo onde se instalara um                             */
    /* novo distribuidor binário.                                               */

    variável booleana Pode_se_ligar_Outro_Poste ;
    /* Indicará se ainda podemos ligar outro poste. Pode ser FALSE por exemplo, se */
    /* não temos intensidade suficiente de sinal.                               */

    variável booleana Existe_DB;
    /* indicará se existe um distribuidor binário elegível para instalar        */

    variável MDSTRB_CMNHO que é do tipo de dado Tipo_Dst_Cmho_Rs;

    I = Intensidade de sinal no ligador de cabo LC;
    Pode_se_ligar_Outro_Poste = TRUE;
    P = poste onde vamos fazer as conexões;
    CNTOR_Sp = o primeiro conector de postes de P pelo que sai sinal;
    /* Se o conector Sp aponta a um poste terminal, então não estará no */
    /* conjunto CJCP_ap_Terminalis .                                        */
}
```

/ Asumiremos que não fracassaremos */*

Parar_Ligações_G = FALSE;

Enquanto ((CJCP_ap_Terminais <> ∅) && Pode_se_ligar_Outro_Poste) Fazer

/ enquanto não está vazio o conjunto CJCP_ap_Ilhados */*

/ e Pode-se ligar outro poste fazer... */*

{

/ procurar o distribuidor binário com o qual obteremos o melhor */*

/ caminho PLUSCS (que neste caso só pode ter um poste, o terminal escolhido)*/*

/ usando só o ligador de cabo de sua saída */*

/ mais atenuada (fraca) e que garanta que o número de residências total */*

/ que pode ser atendido por este distribuidor é o maior, ligando a saída mais */*

/ forte do distribuidor ao conector inicialmente apontado por LC. */*

MDSTRB_CMNHO = Mlhor_DSTRB_e_Cmho(LC, CJCP_ap_Terminais, Existe_DB);

/ se o melhor caminho encontrado é o conjunto vazio para os conectores que */*

/ estão no conjunto CJCP_ap_Terminais, então não poderemos ligar mais */*

/ postes apontados pelos elementos daquele conjunto. */*

/ Também se */*

Se (o caminho de MDSTRB_CMNHO é igual a ∅) ou Não Existe_DB então

{

Pode_se_ligar_Outro_Poste = FALSE;

}

Senão

{

instala_o_distribuidor_no_ligador_de_cabo(LC, MDSTRB_CMNHO);

/ Seja Sp o Conector de postes apontado por LC antes da instalação */*

/ do distribuidor binário de sinal. */*

/ Este procedimento: instala_o_distribuidor_no_ligador_de_cabo, */*

/ tem que marcar os postes de MDSTRB_CMNHO como que já */*

/ recebem sinal, no grafo G = (P, L). Também deverá atribuir um */*

/ mesmo número a cada poste daquele caminho, que é o seguinte do */*

/ número dado a cada um dos postes do caminho anteriormente */*

/ marcado no grafo G =(P, L). Além disso, deverá atualizar o */*

/ caminho de postes apontado por Sp. */*

/ Já utilizamos o ligador LC. Agora deveremos procurar outro */*

/ ligador de cabo dentre aqueles que enviam sinal a um conector de */*

/ postes para fazer a instalação de um novo distribuidor binário. */*

LC = Escolha_dum_Ligador_de_Cabo_para_instalar_um_novo_DS_PP1(poste P);

I = Intensidade de sinal no ligador de cabo LC;

/ CPUS representará o conector de postes de P que recebe sinal da saída */*

/ mais fraca do novo distribuidor binário instalado. */*

CPUS = Conector de postes de P que aponta ao caminho de postes definido em MDSTRB_CMNHO com a saída mais fraca do distri. Bin.;

```

    /* o conector de poste CPUS não deve ser tomado em conta para */
    /* procurar novos caminhos de postes ligados usando só cabo. */

    CJCP_ap_Terminais = CJCP_ap_Terminais - { CPUS };
    gb_Existem_um_Poste_Expandible = TRUE;
    /* indicará que existe pelo menos outro poste no grafo G = (P, L) */
    /* que devermos visitar a fim de ligar seus vizinhos. */
}
}

Se (CJCP_ap_Terminais <> Ø )
{
    Se (o número de elementos de CJCP_ap_Terminais é 1 ) Então
        /* se houver mais de um, não poderemos fazer nada. */
        {
            Se CNTOR_Sp não aponta para um poste terminal Então
                {
                    Seja Cx o conector de CJCP_ap_Terminais;
                    Seja Ey o conector de postes apontado por Cx;
                    I = o sinal que recebe o conector CNTOR_Sp;
                    LC = ligador de cabo que envia sinal para CNTOR_Sp;
                    Se São_Satisfeitos_os_Min_Req_Conx(Cx, Ey, I) Então
                        {
                            Fazer que LC envie sinal a Cx para que o poste de Ey receba sinal;
                            Indicar o conector Cx como o primeiro conector de
                                postes de P pelo qual sai sinal;
                            /* isto só é para arrumar as ligações de P */
                        }
                    Senão
                        {
                            /* se avisará que não pode-se continuar com as ligações */
                            /* iniciadas no poste onde se instalou o amplificador. */
                            Parar_Ligações_G = TRUE;
                        }
                }
            Senão
                {
                    /* se avisará que não pode-se continuar com as ligações iniciadas no */
                    /* poste onde se instalou o amplificador. */
                    Parar_Ligações_G = TRUE;
                }
        }
    Senão
        {
            /* se avisará que não pode-se continuar com as ligações iniciadas no */
            /* poste onde se instalou o amplificador. */
            Parar_Ligações_G = TRUE;
        }
}
}
}

```

Usando este procedimento, o algoritmo para conectar os vizinhos de um poste P , onde primeiro ligaremos seus vizinhos terminais ficará:

```

Procedimento Ligar_os_Conectores_de_Postes_dum_Poste(P)
{
    CJCP_ap_Terminais;
        /* Conjunto de Conectores de Postes que apontam a postes terminais */
        /* que tem residências. */
    C_CP_ap_Ilhados;
        /* Conjunto de Conectores de Postes que apontam a postes ilhados */
    C_CP_ap_Livres;
        /* Conjunto de Conectores de Postes que apontam a postes livres. */

    Variável booleana Livr = FALSE; /* vai nos indicar ligou-se outro poste. */
    Variável booleana Lilha = FALSE; /* vai nos indicar ligou-se outro poste. */

    /* Precisaremos que o poste P tenha duas propriedades: uma que indique */
    /* o conector de postes pelo qual recebe sinal, e outra que indique o primeiro */
    /* conector de postes pelo qual sai sinal. */
    CNTOR_Ep = Conector de postes pelo qual entra sinal em P;
    CNTOR_Sp = Primeiro conector de postes pelo qual sai sinal de P;
    LC = Ligador de cabo do CNTOR_Ep;

    Se (CNTOR_Sp = NULL) então
        /* Se P inicialmente não tem saída de sinal, então */
        {
            Se (P.Residências > 0) então
                {
                    /* não é necessário verificar requerimentos mínimos de sinal, porque */
                    /* foi atualizado na instalação dos distribuidores do poste fonte. */
                    Indica_o_Supertap(P.Residências, I, STAP, Y)
                    /* Y assumirá o sinal na saída do supertap */
                    Instalar_SuperTap(LC, STAP, Y);
                    /* Instalar_SuperTap botará o supertap definitivo no ligador LC. */
                    /* especificando os diversos tipos de taps que formam o supertap. */
                    /* Assim, a saída do supertap irá ter um sinal Y. */
                }
            Senão
                {
                    CNTOR_Sq = Conector de postes que envia sinal a CNTOR_Ep;
                    Estabelecer que CNTOR_Sq não enviará sinal ao CNTOR_Ep;
                }
        }
    Senão
        {

```



```

/* pesquisar se P tem TAP <=> seu número de residências é positivo */
Poste_Inhabilitado = FALSE;
Se (P.Residências > 0) então /* se o poste tem várias residências para atender */
{
    Indica_o_Supertap(P.Residências, I, STAP, Y)
    Instalar_SuperTap(LC, STAP, Y);
    /* Instalar_SuperTap botará o supertap definitivo no ligador LC. */
    /* Assim, a saída do supertap irá ter um sinal Y. */
}

```

LC = Escolha_dum_Ligador_de_Cabo_para_instalar_um_novo_DS_PP1(poste P);

```

/* procura os conectores de postes que apontam a postes terminais */
/* e não estão ligados. */
CJCP_ap_Terminais = Procura_Conect_de_Postes_ap_terminais_do_Poste(P);
Ligar_um_Conj_de_DISTR_ap_Terminais( LC, Conjunto CJCP_ap_Terminais,
                                     Parar_Ligações_G);

```

Se Parar_Ligações_G Então

```

{
    PARAR AS LIGAÇÕES INICIADAS NO POSTE ONDE SE INSTALOU
    O AMPLIFICADOR E INSTALAR AQUELE DISPOSITIVO EM
    OUTRO POSTE;
}

```

Pesquisar_Postes_Ilhados no grafo $G = (P, L)$;

```

/* procura os conectores de postes que apontam a postes ilhados */
C_CP_ap_Ilhados = Procura_Conect_de_Postes_ap_Ilhados_do_Poste(P);

```

```

/* Vamos tentar ligar os conectores de postes, */
/* que apontam a postes ilhados. */
LC = Escolha_dum_Ligador_de_Cabo_para_instalar_um_novo_DS_PP1(poste P);
/* LC irá a ser o ligador do cabo escolhido para inserir */
/* um novo distribuidor se for necessário. */

```

```

/* Vamos tentar ligar os conectores de postes que apontam a postes ilhados. */
Ligar_um_Conj_de_CNTP_ap_pIlhados(LC, C_CP_ap_Ilhados);

```

```

/* Vamos tentar ligar os conectores de postes que apontam a postes livres. */
/* procura os conectores de postes que apontam a postes livres */
C_CP_ap_Livres = Procura_Conect_de_Postes_ap_livres_do_Poste(P);

```

```

Ligar_um_Conj_de_CNTP_ap_pLivres(LC, C_CP_ap_Livres);
/* Marcar o poste para indicar que nele foi aplicada a função */
/* Ligar_os_Conectores_de_Postes_dum_Poste */

```

```

/* conhecida também como LCPP                                     */
P.foi_aplicado_o_procedimento_LCPP_no_poste = TRUE;
Se não existe um conector de postes de P que envie sinal, e
  o poste não tem residências Então
  {
    CNTOR_Sq = Conector de postes que envia sinal a CNTOR_Ep;
    Estabelecer que CNTOR_Sq não enviará sinal ao CNTOR_Ep;
  }
}

```

As modificações feitas mostram que o algoritmo JCCZ, pode evoluir para com ele obter melhores resultados.

Algoritmo de Correção

Infelizmente, não pode-se garantir a ausência de postes ilhados, mesmo lhes dando prioridade para ligação. Assim, ao final devemos procurar os postes ilhados que tenham residências e de algum modo, retirá-los do isolamento.

Definição. Um poste *ilhado* que tem residências será chamado *poste crítico* ou *poste problemático*.

Seja Ξ o conjunto de todos os postes críticos de um grafo $G = (P, L)$. Se $\Xi \neq \Phi$, aplicaremos os seguintes passos para eliminá-los de G . Lembremos que a cada caminho de postes usando só cabo e supertaps foi atribuído um número inteiro. Chamaremos este número de *idade do poste*.

P1. Repetir P2 e P3 até que Ξ seja vazio.

P2. Seja:

P um poste de Ξ .

F um poste da fronteira de G .

$\zeta(P, F)$ representa o conjunto dos caminhos de postes segundo a relação L , de P a F (evidentemente, pode existir mais de um).

Como conhecemos o algoritmo distribuído da menor distância de um poste P para outro F num grafo (ver [4]), poderemos então, baseados neste algoritmo,

encontrar o caminho de postes de P a F com o menor número de postes. Denotaremos aquele caminho por $\overline{\overline{PF}}$.

Como para todos os postes F da fronteira de G , existe $\overline{\overline{PF}}$, dentre todos aqueles caminhos, vamos escolher aquele que tenha *os postes de maior idade*. Denotaremos este caminho com $\langle \overline{\overline{PF}} \rangle$.

Para todo poste de $\langle \overline{\overline{PF}} \rangle$, vamos desligar estes e as ligações que partem destes.

P3. Tirar o poste P do conjunto Ξ . Pesquisar os postes de Ξ que ficarem livres com nosso algoritmo para pesquisar os postes ilhados e retirá-los de Ξ .

Se este algoritmo for aplicado, nunca vamos encontrar um caminho de um poste ilhado até um poste terminal, pois o algoritmo procura caminhos de postes aos fronteirigos.

Infelizmente, a complexidade deste algoritmo é exponencial. Mas, para nossa boa sorte, normalmente cada poste só tem dois vizinhos segundo L .

O melhor é tentar escolher grafos que pela sua topologia não permitam a aparição de postes ilhados. A construção destes grafos não é tema de discussão desta tese.

Exemplos

Os seguintes exemplos, tentam mostrar como se comportam os algoritmos apresentados em situações reais. O primeiro deles, foi retirado de uma região da cidade do Rio de Janeiro. Os exemplos são pequenos (exemplos maiores requereriam um esforço bastante grande para serem resolvidos de forma manual com aplicação dos algoritmos), mas ilustram cada passo do algoritmo de forma precisa.

Os valores dos distribuidores binários e taps são os dos exemplos feitos ao mostrar cada um de aqueles distribuidores.

Vamos supor que a lista dos distribuidores binários está ordenada assim:

Divisor, acoplador direcional de 8, acoplador direcional de 12.

EXEMPLO 1. Seja o seguinte grafo G com os postes A, B, C, D :

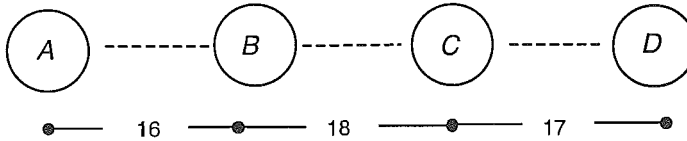


Figura 4.20. Exemplo de um grafo de postes a ligar.

onde as distâncias entre os postes são:

$$d(A, B) = 16\text{m}; \quad d(B, C) = 18\text{m}; \quad d(C, D) = 17\text{m};$$

As linhas ----- indicam a relação L .

As residências para cada poste são dadas pela seguinte tabela:

Poste	Residências
A	8
B	6
C	5
D	8

Tabela 4.1.

A atenuação para uma residência será $gb_Res = 17$ dBm com uma tolerância $gt_Tol = 0$. O sinal do amplificador será 37 dBm.

Ao aplicar o procedimento **Ligar_Postes_a_partir_de(A)**, vamos instalar o amplificador no poste **A**. Portanto, ele vai ficar com dois conectores de postes **Ea** e **Sa** para entrada e saída de sinal respectivamente.

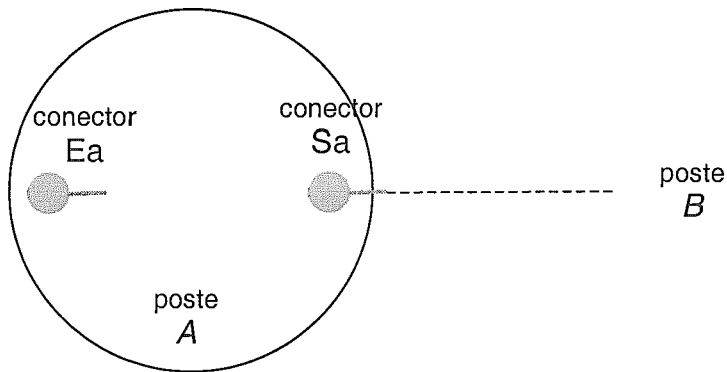


Figura 4.21. Instalação do amplificador **Ea**.

Após vamos procurar o melhor caminho de postes usando somente cabo e supertaps, com poste inicial em **A**, usando o procedimento

Da_Melhor_Caminho_Desde(Ea, TRUE, 37)

Como **A** tem 8 residências, precisamos procurar o supertap adequado para este poste e um sinal de 37 dBm. O supertap adequado é o de 8 saídas e atenuação 20 para a residência.

O seguinte conector de **Ea** é **Sa**. O conector **Sa** portanto vai ter o seguinte sinal:

$$37 - 0,9 = 36,1$$

Sejam **Eb** e **Sb** os conectores de postes de entrada e saída do poste **B**. Então, o sinal em **Eb** será:

$$36,1 - \text{atenuação do cabo de } A \text{ a } B = 36,1 - 16 * 0,0512 = 35,2808$$

e aplicaremos o algoritmo

Da_Melhor_Caminho_Desde(Eb, TRUE, 35,2808)

Como *B* tem residências, precisamos procurar o supertap adequado para este poste e um sinal de 35,2808 dBm. O supertap adequado é o de 8 saídas e atenuação 17 para residência.

O seguinte conector de **Eb** é **Sb**. O conector **Sb** portanto, vai ter o seguinte sinal:

$$35,2808 - 1 = 34,2808$$

Sejam **Ec** e **Sc** os conectores de postes de entrada e saída do poste *C*. Então, o sinal em **Ec** será:

$$\begin{aligned} 34,2808 - \text{atenuação do cabo de } B \text{ a } C \\ = 34,2808 - 18 * 0,0512 = 33,3592 \end{aligned}$$

e aplicaremos o algoritmo

Da_Melhor_Caminho_Desde(Ec, TRUE, 33,3592)

Como *C* tem residências, precisamos procurar o supertap adequado para este poste e um sinal de 33,3592 dBm. O supertap é aquele de 8 saídas e atenuação 14 para residência.

O seguinte conector de **Ec** é **Sc**. O conector **Sc** portanto, terá o seguinte sinal:

$$33,3592 - 1,6 = 31,7592$$

Sejam **Ed** e **Sd** os conectores de postes de entrada e saída do poste *D*. Então, o sinal em **Ed** será:

$$\begin{aligned} 31,7592 - \text{atenuação do cabo de } C \text{ a } D \\ = 31,7592 - 17 * 0,0512 = 30,8888 \end{aligned}$$

e aplicaremos o algoritmo

Da_Melhor_Caminho_Desde(Ed, TRUE, 30,8888)

Como D tem residências, precisamos procurar o supertap adequado para este poste e um sinal de 30,8888 dBm. Ele é aquele de 8 saídas e atenuação 11 para residência. Lembremos que o tap de 11 não tem saída de cabo CC5.

Indicaremos o caminho de postes usando só cabo e supertaps no grafo G . Se G_A é o subgrafo de G , dos postes ligados a partir do poste A , então percorreremos este subgrafo em pré-ordem, começando no poste A , usando o algoritmo

Ligar_Postes_com_os_Conectores_de_Postes_a_partir_do_Poste(A)

e assim, iremos ter o grafo ligado.

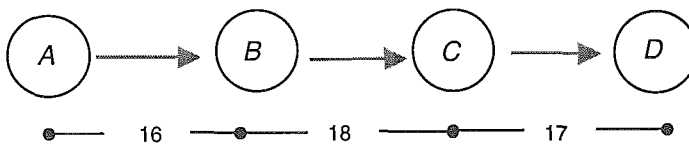


Figura 4.22. Postes do grafo já ligados.

É importante notar que apesar deste exemplo ser pequeno, num contexto maior, a solução para este subgrafo alcançou o projeto ótimo.

EXEMPLO 2. Seja o seguinte grafo G com os postes A, B, C, D, E, F :

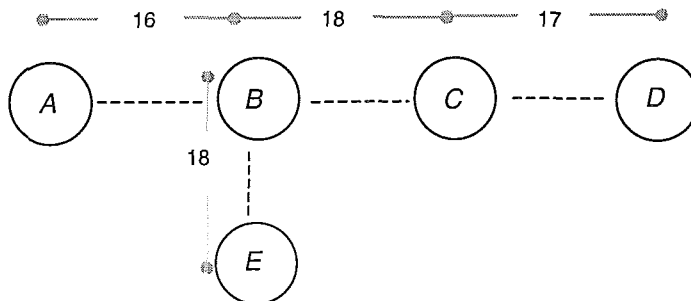


Figura 4.23. Grafo de postes a ligar.

onde as distâncias entre os postes são:

$$d(A, B) = 16\text{m}; \quad d(B, C) = 18\text{m};$$

$$d(C, D) = 17\text{m}; \quad d(B, E) = 18\text{m};$$

As linhas ----- indicam a relação L.

As residências para cada poste são dadas pela seguinte tabela:

Poste	Residências
A	8
B	6
C	5
D	8
E	3

Tabela 4.2. Exemplo de um grafo de postes a ligar.

Ao aplicar o procedimento **Ligar_Postes_a_partir_de(A)**, instalaremos o amplificador no poste A. Logo, ele vai ficar com dois conectores de postes **Ea** e **Sa** para entrada e saída de sinal respectivamente.

Após vamos procurar o melhor caminho de postes usando somente cabo e supertaps com poste inicial em A usando a função

Da_Melhor_Caminho_Desde(Ea, TRUE, 37)

Como A tem 8 residências, precisamos procurar o supertap adequado para este poste e um sinal de 37dBm. O supertap adequado é o de 8 saídas e atenuação 20 para residência.

O seguinte conector de **Ea** é **Sa**. O conector **Sa** portanto, vai ter um sinal igual a $37 - 0.9 = 36,1$.

A partir de B vamos ter que pesquisar os melhores caminhos que partam deste poste. O sinal em E_b será:

$$36,1 - \text{atenuação do cabo de } A \text{ a } B = 36,1 - 16 * 0,0512 = 35,2808$$

e aplicaremos o algoritmo

Da_Melhor_Caminho_Desde(E_b , TRUE, 35,2808)

Como B tem residências, precisamos procurar o supertap adequado para este poste e um sinal de 35,2808 dBm. Ele é aquele de 8 saídas e atenuação 17 para residência. O ligador de cabo da saída do supertap tem um sinal $35,2808 - 1 = 34,2808$.

Sejam $Sb1$ e $Sb2$ dois conectores de postes de B , tal que o primeiro aponta ao conector de postes Ec do poste C , e o segundo, aponta ao conector de postes Ee do poste E .

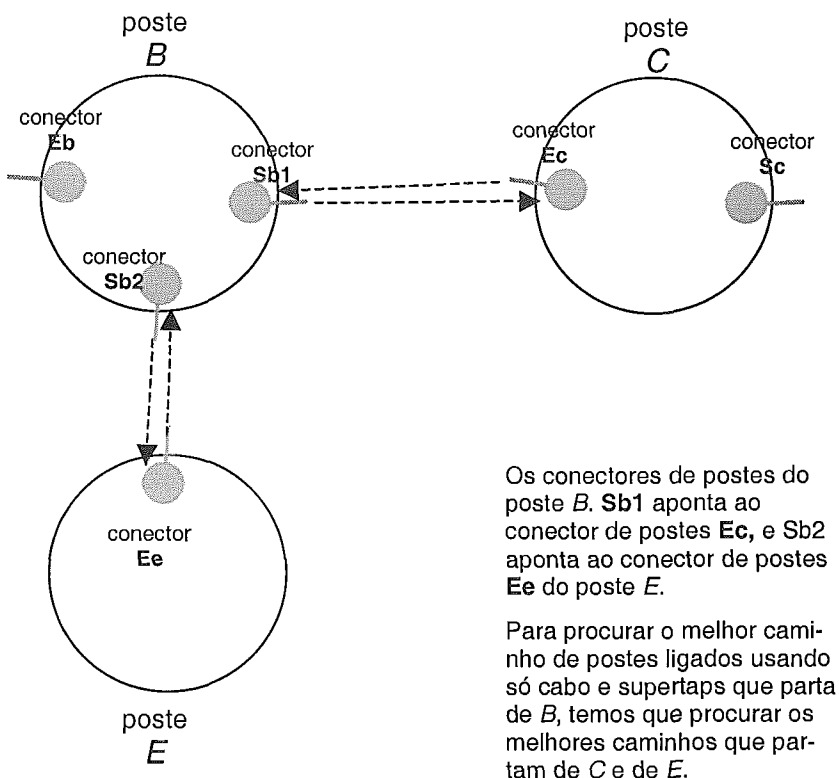


Figura 4.24. Procura do melhor caminho a partir do poste B .

Se o sinal que sai pelo conector **Sb1** é 34,2808, então o caminho encontrado será o mesmo do exemplo 1.

Se o sinal que sai pelo conector **Sb2** for 34,2808, então é fácil de ver que vamos encontrar o caminho com o único poste *E*.

Indubitavelmente, o melhor caminho usando cabo e supertaps vai ser *ABCD*. Deveremos então indicá-lo no grafo **G**. Porém ainda não vamos instalar os supertaps.

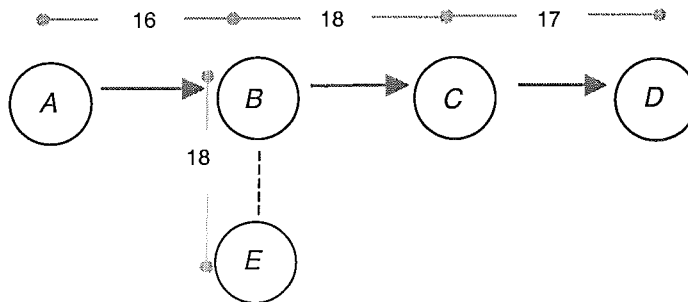


Figura 4.25. O caminho de postes ligados usando só cabo e supertaps *ABCD*.

Desta forma, o grafo G_A estará formado pelos postes *ABCD*. Deveremos então percorrê-lo em pré-ordem aplicando ao poste *A* o algoritmo

Ligar_Postes_com_os_Conectores_de_Postes_a_partir_do_Poste(A);

Assim, instalaremos agora o supertap definitivo em *A*. Como *A* não tem vizinhos desligados, então devemos continuar com o poste *B*. O sinal que chega a *B* não foi alterado, então instalaremos de modo definitivo o supertap calculado anteriormente. Deveremos assim escolher o ligador de cabo da saída do supertap para instalar um distribuidor binário.

Seja qual for o distribuidor binário, ele vai receber um sinal igual a 34,2808, e portanto deveremos pesquisar o que acontecerá com qualquer distribuidor binário.

Para o divisor. Lembremos que ele tem duas saídas, com igual atenuação. Assim, o sinal para cada saída é $34,2808 - 3,9 = 30,3808$ (onde 3,9 é a atenuação do divisor). Se instalarmos este distribuidor binário, uma de suas saídas apontaria a **Sb1** e a outra a **Sb2**. Vamos ter que pesquisar o que aconteceria nos postes *C* e *D*, executando os procedimentos:

Da_Melhor_Caminho_Desde(Sb1, FALSE, 30,3808)

Da_Melhor_Caminho_Desde(Sb2, FALSE, 30,3808)

O conector **Ec** receberia um sinal

$$30,3808 - \text{atenuação de } B \text{ a } C = 30,3808 - 18 * 0,0512 = 29,4592$$

deveremos pesquisar o supertap, executando o procedimento

Tentar_Definir_STAP7(5, 29,4592 , STAP, Y)

o único tap de 8 saídas é aquele de atenuação 11 para residência e não poderíamos avançar para o poste *D*.

Também o conector **Ee** receberia um sinal 29,4592, e deveremos pesquisar o supertap, executando o procedimento

Tentar_Definir_STAP7(3, 29,4592 , STAP, Y)

o tap escolhido é aquele de 4 saídas com atenuação de 11 para residência.

Assim, ligaríamos o poste *E*, mas perderíamos o poste *D*.

Os resultados importantes pela instalação do divisor seriam:

Residências que se atenderia nos caminhos apontados por **Sb1** e **Sb2**: 8

Residências que podem ser atendidas no caminho apontado por **Sb1**: 5

Para o acoplador direcional de 8. Lembremos que ele tem duas saídas, com diferente atenuação. Assim, o sinal para a saída mais *forte* (de menor atenuação) é $34,2808 - 1,5 = 32,7808$. E para a saída mais *fraca* (de maior atenuação) é $34,2808 - 8 = 26,2808$.

Se instalarmos este distribuidor binário, a sua saída mais forte apontaria a **Sb1** e a outra a **Sb2**. Vamos ter que pesquisar o que aconteceria nos postes *C* e *D*.

Ao aplicar o procedimento

Da_Melhor_Caminho_Desde(Sb1, FALSE, 32,7808)

o conector **Ec** receberia um sinal

$$32,7808 - \text{atenuação de } B \text{ a } C = 32,7808 - 18 * 0,0512 = 31,8592$$

Deveremos pesquisar o supertap para *C*, executando o procedimento

Tentar_Definir_STAP7(5, 31,8592 , STAP, Y)

o único tap de 8 saídas é aquele de atenuação 14 para residência. O conector **Sc** portanto, vai receber um sinal de $31,8592 - 1,6 = 30,2592$.

Portanto, o conector **Ed** vai receber um sinal

$$30,2592 - \text{atenuação de } C \text{ a } D = 30,2592 - 17 * 0,0512 = 29,3888$$

Deveremos pesquisar o supertap para *D*, executando a instrução

Tentar_Definir_STAP7(8, 29,3888 , STAP, Y)

O único tap de 8 saídas é aquele de atenuação 11 para residência. Lembremos que este não tem saída para cabo CC5.

Desde que o conector **Sb2** vai receber um sinal de 26,2808, ao aplicar o procedimento

Da_Melhor_Caminho_Desde(Sb2, FALSE, 26,2808)

o conector **Ee** vai receber um sinal de

$$26,2808 - \text{atenuação de } B \text{ a } E = 26,2808 - 18 * 0,0512 = 25,3592$$

Deveremos pesquisar o supertap para *E*, executando o procedimento

Tentar_Definir_STAP7(3, 25,3592 , STAP, Y)

O único tap de 4 saídas é aquele de atenuação 8 para residência. Lembremos que este não tem saída para cabo CC5.

Os resultados importantes pela instalação do divisor seriam:

Residências que se atenderia nos caminhos apontados por **Sb1** e **Sb2**: 16

Residências que podem ser atendidas no caminho apontado por **Sb1**: 13

Para o acoplador direcional de 12. Lembremos que ele tem duas saídas, com diferente atenuação. Assim, o sinal para a saída mais *forte* (de menor atenuação) é $34,2808 - 1,2 = 33,0808$. E para a saída mais fraca (de maior atenuação) é $34,2808 - 12 = 22,2808$.

Se instalarmos este distribuidor binário, a sua saída mais forte apontaria a **Sb1** e a outra a **Sb2**. Vamos ter que pesquisar o que aconteceria nos postes *C* e *D*.

Ao aplicar o procedimento

Da_Melhor_Caminho_Desde(Sb1, FALSE, 33,0808)

o conector **Ec** receberia um sinal

$$33,0808 - \text{atenuação de } B \text{ a } C = 33,0808 - 18 * 0,0512 = 32,1592$$

Deveremos pesquisar o supertap para *C*, executando o procedimento

Tentar_Definir_STAP7(5, 32,1592 , STAP, Y)

o único tap de 8 saídas é aquele de atenuação 14 para residência. O conector **Sc** portanto, vai receber um sinal de $32,1592 - 1,6 = 30,5592$

Portanto, o conector **Ed** vai receber um sinal

$$30,5592 - \text{atenuação de } C \text{ a } D = 30,5592 - 17 * 0,0512 = 29,6888$$

Deveremos pesquisar o supertap para *D*, executando o procedimento

Tentar_Definir_STAP7(8, 29,6888 , STAP, Y)

O único tap de 8 saídas é aquele de atenuação 11 para residência. Lembremos que este não tem saída para cabo CC5.

Desde que o conector **Sb2** vai receber um sinal de 22,2808, ao aplicar o procedimento

Da_Melhor_Caminho_Desde(Sb2, FALSE, 22,2808)

o conector **Ee** vai receber um sinal de

$$22,2808 - \text{atenuação de } B \text{ a } E = 22,2808 - 18 * 0,0512 = 21,3592$$

Deveremos pesquisar o supertap para E , executando o procedimento

Tentar_Definir_STAP7(3, 21,3592 , STAP, Y)

A variável STAP indicaria um supertap não válido, pois não existe um tap de 4 saídas para este caso. Assim, não poderá ficar habilitado o conector de postes **Sb2**.

Os resultados importantes pela instalação do divisor seriam:

Residências que se atenderia nos caminhos apontados por **Sb1** e **Sb2**: 13

Residências que podem ser atendidas no caminho apontado por **Sb1**: 13

Conclusão: o melhor distribuidor binário é o acoplador direcional de 8 e após sua instalação no poste B , teremos:

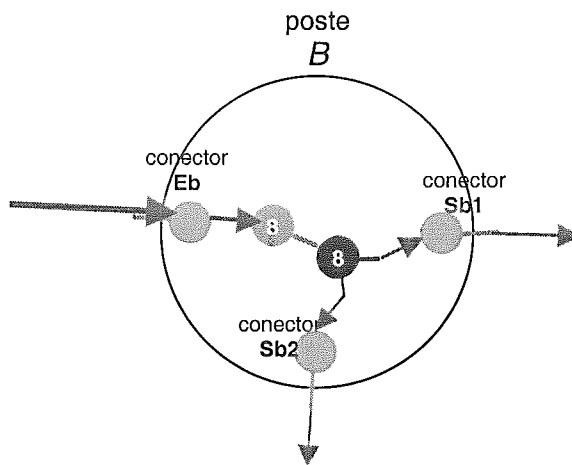


Figura 4.26. O poste B após a instalação do distribuidor binário de 8.

e o grafo G ficará

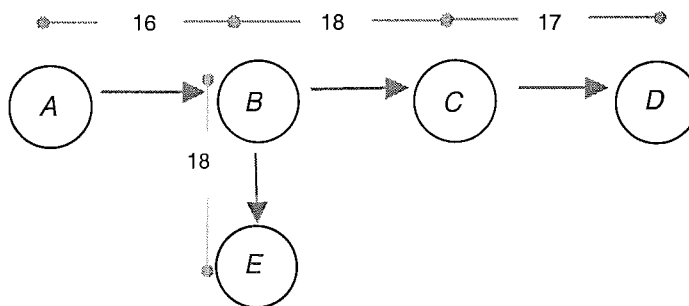


Figura 4.27. Postes já ligados.

Neste exemplo, também conseguimos atender o número ótimo de residências.

EXEMPLO 3. No grafo da figura 4.23, vamos instalar o amplificador no poste *B*. Assim, ele vai ficar:

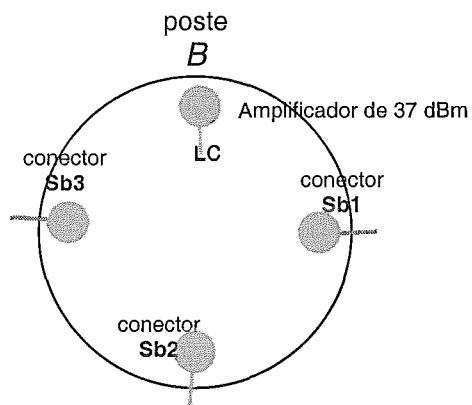


Figura 4.28. O poste *B* após a instalação do amplificador.

Como *B* tem 6 residências, então vamos procurar o supertap para este poste com o procedimento

Tentar_Definir_STAP7(6, 37, STAP, Y)

e obteremos o tap de 8 com $A = 20$. Portanto, a saída do tap irá ter um sinal de

$$37 - 0,9 = 36,1$$

Seja agora **LC** o ligador de cabo do tap.

Agora, vamos procurar o melhor caminho PLUSCS que parte de *B* com o sinal que sai do ligador **LC** do tap.

O caminho PLUSCS apontado por *Sb1* tem os postes *C* e *D*. O número de residências neste caminho é 13.

O caminho PLUSCS apontado por Sb2 tem o poste *E*. O número de residências neste caminho é 3.

O caminho PLUSCS apontado por Sb3 tem o poste *A*. O número de residências neste caminho é 8.

Portanto, o caminho apontado por Sb1 é o melhor e será indicado no grafo **G**. Assim, **LC** apontará a Sb1.

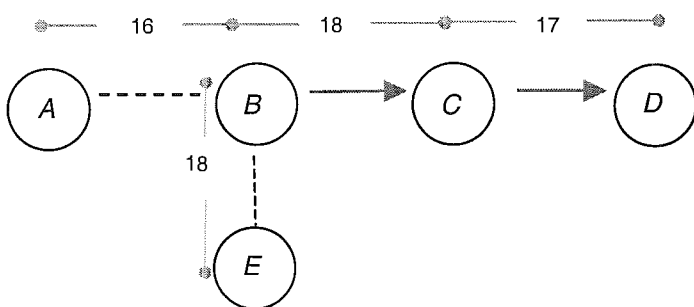


Figura 4.29. O melhor caminho PLUSCS, *BCD* que parte de *B*.

Vamos pesquisar o que aconteceria se instalássemos o *divisor* no ligador de cabo **LC**.

Como as saídas têm a mesma atenuação, então cada uma terá um sinal de:

$$36,1 - 3,9 = 32,2$$

Com este sinal vamos pesquisar o que aconteceria com o caminho *CD*:

O sinal que chegaria ao poste *C* seria: $32,2 - 18 * 0,0512 = 31,2784$

e seu tap respectivo seria um de 8 e $A = 14$. Portanto, o sinal que sairia deste tap para o cabo CC5 seria: $31,2784 - 1,6 = 29,6784$

O sinal que chegaria ao poste *D* seria: $29,6784 - 17 * 0,0512 = 28,808$

e seu tap respectivo seria um de 8 e $A = 11$. Como este tap é terminal então a linha não prossegue.

Portanto, o caminho *CD* pode ser atendido.

Usaremos a mesma intensidade 32,2 para saber qual é o melhor conector de postes entre *Sb3* e *Sb2*. Facilmente, o conector *Sb3* é o melhor, pois o caminho apontado por ele tem 8 residências.

Portanto, se instalarmos o divisor, uma de suas saídas apontaria a *Sb1* e a outra a *Sb3*, atendendo a um número de 27 residências.

Vamos pesquisar o que aconteceria se instalássemos o *acoplador direcional* de 8 no ligador de cabo **LC**.

A saída mais forte do acoplador direcional de 8 irá ter um sinal igual a

$$36,1 - 1,5 = 34,6$$

Com este sinal, vamos poder alimentar o caminho apontado por *Sb1*.

O sinal na saída mais fraca será: $36,1 - 8 = 28,1$. Para este valor, vamos encontrar o melhor conector de postes entre *Sb2* e *Sb3*.

Se *Sb2* receber o sinal 28,1 então o poste *E* receberá um sinal 27,1784. Como *E* tem 3 residências, então o melhor tap para este sinal é um de 4 e $A = 8$. Portanto, o caminho apontado por *Sb2* tem 3 residências.

Se *Sb3* receber o sinal 28,1 então o poste *A* receberá um sinal 27,2808. Como *A* tem 8 residências, então não existe um tap de 8 para este poste.

Portanto, o melhor conector para o sinal fraco do acoplador direcional de 8 é *Sb2*, e se for instalado este dispositivo, então o número total de residências atendidas será 22.

Vamos pesquisar o que aconteceria se instalássemos o acoplador direcional de 12 no ligador de cabo **LC**.

A saída mais forte do acoplador direcional de 8 irá ter um sinal igual a

$$36,1 - 1,2 = 34,9$$

Com este sinal, vamos poder alimentar o caminho apontado por *Sb1*.

O sinal na saída mais fraca será: $36,1 - 12 = 24,1$. Para este valor, vamos encontrar o melhor conector de postes entre Sb2 e Sb3.

Se Sb2 receber o sinal 24,1, então o poste *E* receberá um sinal 23,1784. Como *E* tem 3 residências, precisaremos instalar um tap de 4, mas não existe tap de 4 para este valor de sinal.

Evidentemente, não poderemos ligar o poste *A* partindo de Sb3.

O melhor conector de postes para o sinal que sai da saída mais fraca do acoplador direcional de 12 é Sb2. Mas o caminho apontado por ele é vazio.

Portanto, o número de residências que se atenderia com o acoplador direcional de 12 seria 19.

Logo, o melhor distribuidor binário de sinal será o divisor. Portanto, este será instalado na saída **LC**. O poste *B* ficará:

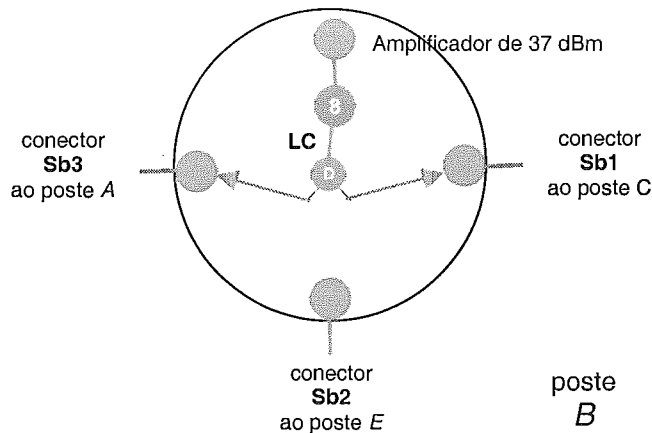


Figura 4.30. O poste *B* após a instalação do divisor.

Como ainda falta o conector Sb2, o algoritmo pesquisará entre os ligadores de cabo que apontam a Sb3 e Sb1 aquele que atende a menor quantidade de residências. Neste caso, o melhor é o conector Sb3. Seja assim, **LC** o ligador de cabo que aponta a Sb3. Portanto o sinal que vai sair de **LC** é 32,2.

Se instalássemos o divisor em **LC**, então o sinal que sairia em cada uma de suas saídas seria:

$$32,2 - 3,9 = 28,3$$

Com este sinal, o poste *A* vai receber $28,3 - 0,8192 = 27,4808$. Mas isto não é suficiente para atender ao poste *A*. Portanto, o divisor não é elegível para instalar em **LC**.

Se instalássemos o acoplador direcional de 8, então a saída mais forte deste distribuidor iria ter um sinal igual a $32,2 - 1,5 = 30,7$. O poste apontado por *Sb3* receberia um sinal igual a $30,7 - 0,8192 = 29,8808$. Para este valor, existe um tap de 8 e $A = 11$. Portanto o acoplador direcional de 8 é elegível para instalar em **LC**.

A saída mais atenuada deste distribuidor terá o valor de $32,2 - 8 = 24,2$. Assim, o poste *E* poderá receber $24,2 - 0,9216 = 23,2784$. Para este valor não existe tap de 4 para atender as residências de *E*.

Se instalássemos o acoplador direcional de 12, então a saída mais forte deste distribuidor iria ter um sinal igual a $32,2 - 1,2 = 31,0$. Portanto poderemos atender ao poste *A* mas não poderemos atender ao poste *E*.

Assim, não será instalado nenhum outro distribuidor e as ligações no poste *B* ficarão como na figura 4.30.

EXEMPLO 4. Seja o seguinte grafo *G* com os postes *A, B, C, D, E, F, G*.

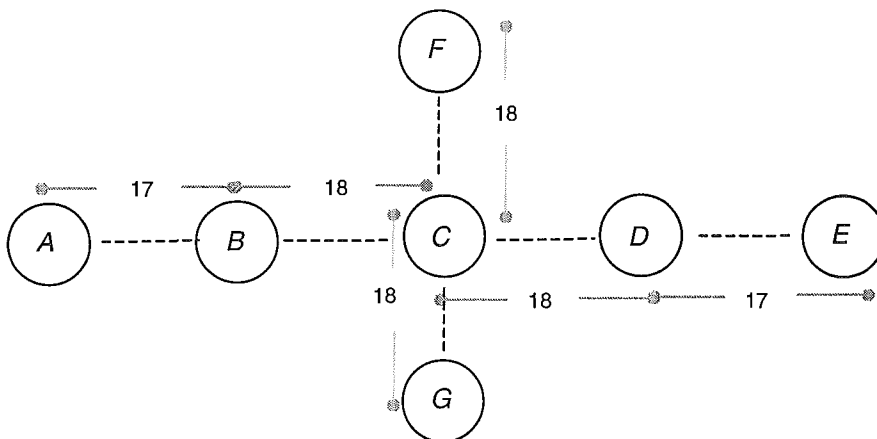


Figura 4.31. Grafo de postes a ligar.

onde as distâncias entre os postes são:

$$d(A, B) = 17\text{m}; \quad d(B, C) = 18\text{m};$$

$$d(C, D) = 18\text{m}; \quad d(D, E) = 17\text{m};$$

$$d(C, F) = 18\text{m}; \quad d(C, G) = 18\text{m};$$

As linhas ----- indicam a relação L . Vamos supor que a saída do amplificador tem um sinal igual a 47 dBm.

As residências para cada poste são dadas pela seguinte tabela:

Poste	Residências
<i>A</i>	8
<i>B</i>	6
<i>C</i>	5
<i>D</i>	8
<i>E</i>	3
<i>F</i>	2
<i>G</i>	4

Tabela 4.4. Exemplo de um grafo de postes a ligar.

Vamos instalar o amplificador no poste C . Como C tem 5 residências, o tap utilizado será de 8 saídas e atenuação para residência $A = 26$. Portanto, a saída **LC** do tap terá um sinal 46,2.

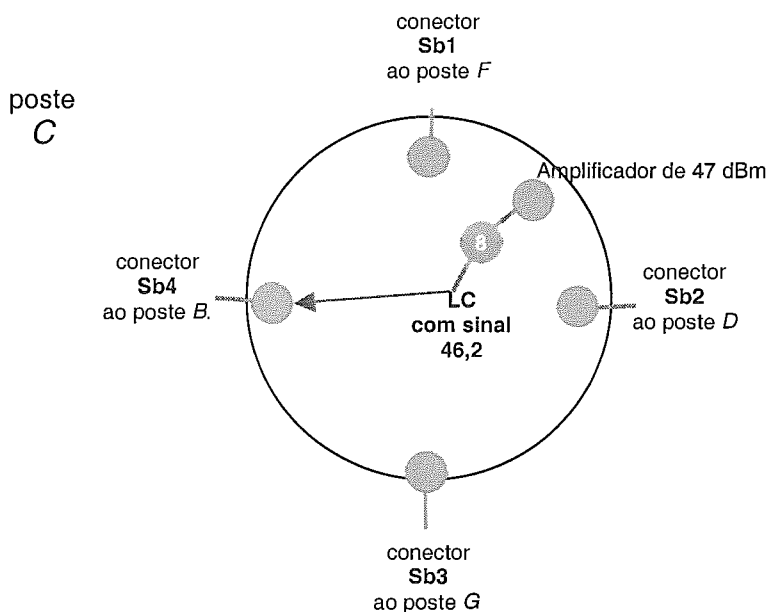


Figura 4.32. O poste C após a instalação do amplificador e seu tap. A saída LC do tap apontará inicialmente ao conector Sb4.

Com o sinal que sai de **LC**, vamos procurar o melhor conector de postes dentre Sb1, Sb2, Sb3 e Sb4.

Se **LC** for ligado a Sb1, mais duas residências serão atendidas.

Se **LC** for ligado a Sb2, mais 11 residências serão atendidas.

Se **LC** for ligado a Sb3, mais 4 residências serão atendidas.

Se **LC** for ligado a Sb4, mais 14 residências serão atendidas.

Portanto, **LC** apontará inicialmente a Sb4. Agora, vamos tentar instalar um dos distribuidores binários (ver fig. 4.31).

Vamos supor que a lista dos distribuidores binários está ordenada assim:

Acoplador direcional de 12, Acoplador direcional de 8, divisor.

Análise para o acoplador direcional de 12. Se o acoplador direcional de 12 for instalado em **LC** então sua saída mais forte ficará ligada a Sb4. Com a saída mais fraca vamos tentar ligar um dos outros ligadores de cabo. Assim, para a saída mais forte o sinal será $46,2 - 1,2 = 45$ e para a saída mais fraca será $46,2 - 12 = 34,2$.

Com este valor, poderá chegar ao poste *D* um sinal $34,2 - 0,9216 = 33,2784$. O tap para este valor será um de 8 e atenuação para residência *A* = 14. Portanto, a saída deste tap terá uma intensidade de $33,2784 - 1,6 = 31,6784$. O poste *E* poderá receber um sinal $31,6784 - 0,8704 = 30,808$. Portanto, o tap adequado para o poste *A* é aquele de 4 saídas e atenuação *A* = 11 para residência.

Com o acoplador direcional de 12 poderão se atender 30 residências.

Análise para o acoplador direcional de 8. Se o acoplador direcional de 8 for instalado em **LC** então sua saída mais forte ficará ligada a *Sb4* e com a outra vamos tentar ligar um dos outros ligadores de cabo. Assim, para a saída mais forte o sinal será $46,2 - 1,5 = 44,7$ e para a saída mais fraca será $46,2 - 8 = 38,2$.

Com este valor, poderá chegar ao poste *D* um sinal $38,2 - 0,9216 = 37,2784$. O tap para este valor será um de 8 e atenuação para residência *A* = 20. Portanto, a saída deste tap terá uma intensidade de $37,2784 - 0,9 = 36,3784$. O poste *E* poderá receber um sinal $36,3784 - 0,8704 = 35,508$. Portanto, o tap adequado para o poste *A* é aquele de 4 saídas e atenuação *A* = 17 para residência.

Com o acoplador direcional de 8 poderão se atender 30 residências.

Análise para o divisor. Se o divisor for instalado em **LC** então uma de suas saídas ficaria ligada a *Sb4* e com a outra vamos tentar ligar um dos outros ligadores de cabo. Assim, cada uma das saídas do divisor terá um sinal $46,2 - 3,9 = 42,3$.

Com este valor, poderá chegar ao poste *D* um sinal $42,3 - 0,9216 = 41,3784$. O tap para este valor será um de 8 e atenuação para residência *A* = 23. Portanto, a saída deste tap terá um intensidade de $40,5784$. O poste *E* poderá receber um sinal $40,5784 - 0,8704 = 39,708$. Portanto, o tap adequado para o poste *A* é aquele de 4 saídas e atenuação *A* = 20 para residência.

Então o distribuidor binário escolhido segundo o algoritmo

Melhor_DSTRB_e_Cmho_para_conec_em

será o acoplador direcional de 12.

Portanto, o conector de postes escolhido é *Sb2*. O poste *C* ficará

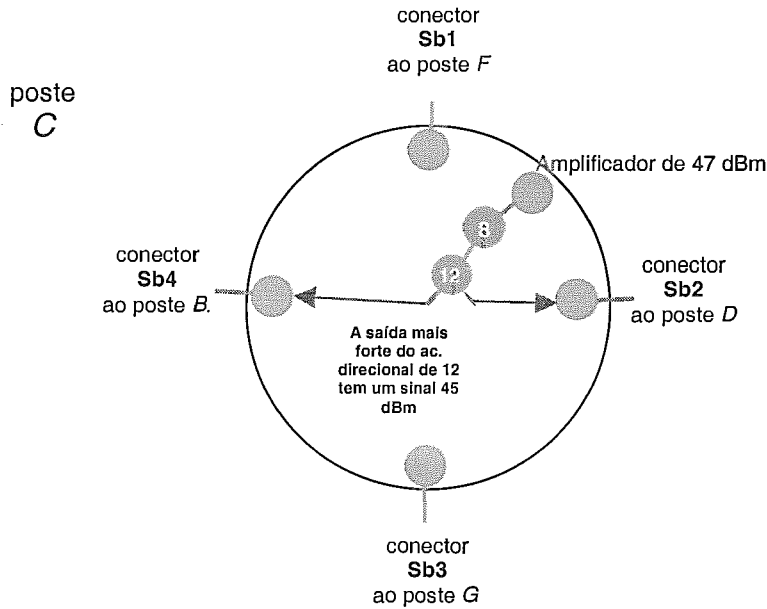


Figura 4.33. O poste C após a instalação do acoplador direcional de 12.

O algoritmo

Escolha_dum_Ligador_de_Cabo_para_instalar_um_novo_DS_PP1

vai escolher o ligador de cabo correspondente a saída mais forte do acoplador direcional de 12. Seja **LC** este ligador. Por tanto ele vai ter um sinal 45 dBm.

Para instalar um distribuidor binário em **LC** repetiremos a análise.

Análise para o acoplador direcional de 12. Se o acoplador direcional de 12 for instalado em **LC** então sua saída mais forte ficará ligada a Sb4 e com a outra vamos tentar ligar um dos outros ligadores de cabo. Assim, para a saída mais forte o sinal será $45 - 1,2 = 43,8$ e para a mais fraca será $45 - 12 = 33$.

Com este valor, poderá chegar ao poste G um sinal $33 - 0,9216 = 32,0784$. O tap para este valor será um de 4 e atenuação para residência A = 14.

Com o acoplador direcional de 12 poderão se atender 34 residências.

É fácil ver que o distribuidor binário a instalar será o acoplador direcional de 12. O poste C ficará:

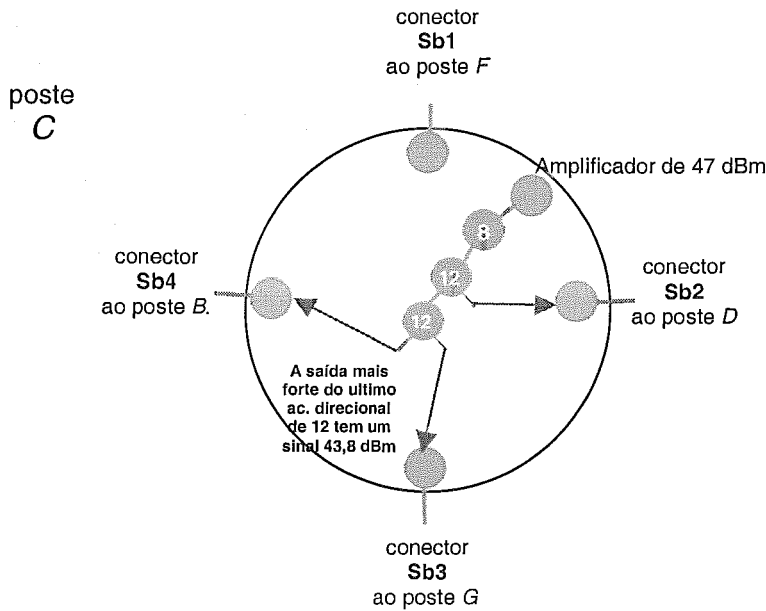


Figura 4.34. O poste C após a instalação do novo acoplador direcional de 12.

O algoritmo

Escolha dum Ligador de Cabo para instalar um novo DS_PP1

vai escolher o ligador de cabo correspondente a saída mais forte do último acoplador direcional de 12 instalado. Seja **LC** este ligador. Portanto ele vai ter um sinal 43,8 dBm.

Para instalar um distribuidor binário em **LC** repetiremos a análise.

Análise para o acoplador direcional de 12. Se o acoplador direcional de 12 for instalado em **LC** então sua saída mais forte ficará ligada a Sb4 e com a outra vamos tentar ligar um dos outros conectores de postes. Assim, para a saída mais forte o sinal será $43,8 - 1,2 = 42,6$ e para a saída mais fraca será $43,8 - 12 = 31,8$.

Com este valor, poderá chegar ao poste *F* um sinal $31,8 - 0,9216 = 30,8784$. O tap para este valor será um de 2 e atenuação para residência *A* = 11.

Com o acoplador direcional de 12 poderão se atender 36 residências.

É fácil ver que o melhor distribuidor binário a ser instalado será o acoplador direcional de 12. O poste *C* ficará:

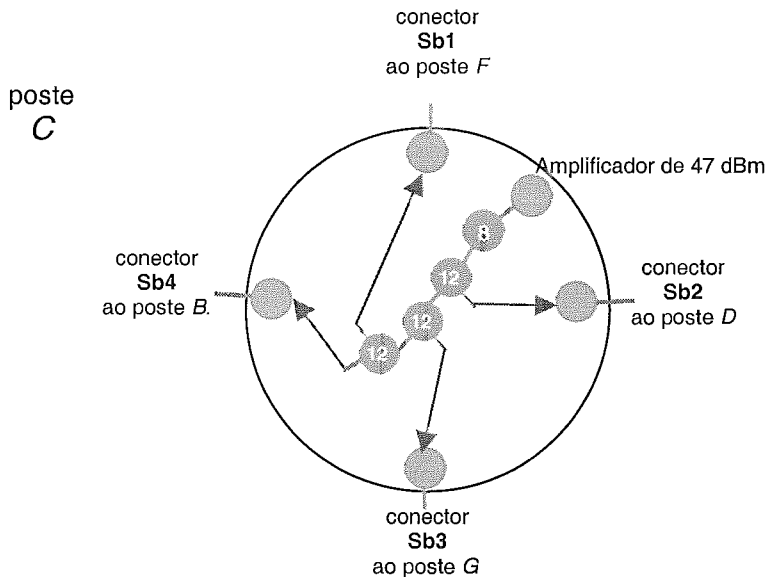


Figura 4.35. O poste *C* após a instalação do último acoplador direcional de 12.

Melhoras ao ALGORITMO de JCCZ

1. Poderíamos aumentar a velocidade do algoritmo, usando sistematicamente as funções

`Escolha_dum_Ligador_de_Cabo_para_instalar_um_novo_DS_PP1(poste P)`

`Escolha_dum_Ligador_de_Cabo_para_instalar_um_novo_DS_PP2(poste P)`

Assim, utilizaríamos somente a primeira se um poste tem muitos vizinhos. Mas se o poste tem poucos vizinhos, poderíamos utilizar só a segunda naquele poste.

2. Embora não tenha sido dito antes, existe uma distância $\delta > 0$, tal que se dois postes vizinhos *P* e *Q* cumprem que a distância entre eles é menor ou igual que δ , os dois estão num mesmo caminho de postes ligados usando somente cabo e super-taps, e *Q* recebe sinal de *P*, então um certo número de residências de *Q* pode passar para *P*.

Assim, se P tem um supertap de S saídas, e o número de residências r_p (maior que zero) de P é menor que S , então poderíamos passar um certo número maior que zero de residências r ($r < r_q$, onde r_q é o número de residências do poste Q) do poste Q ao poste P :

$$r_p + r \leq S$$

a fim de tentar ligar todas as saídas S .

Se o poste P não tem residências, mas o poste Q tem, então poderíamos passar algumas residências de Q a P e instalar um supertap adequado em P .

A ação de passar residências de um poste para outro é conhecido como *balanceamento*.

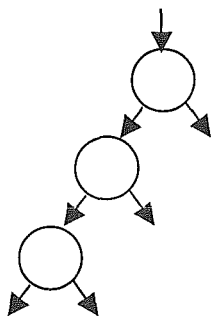
Um poste P que recebeu residências de outro poste Q pelo balanceamento, não poderá ser usado para outro balanceamento com outro poste R ($R \neq P$, $R \neq Q$).

O resultado de aplicar o dito anteriormente, é que poderemos ter mais possibilidades de atender a todas as residências do caminho de postes ligados usando só cabo e supertaps ao que P e Q pertencem.

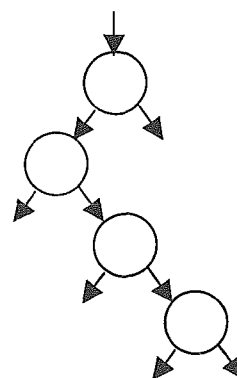
Outras Formas de Modelagem

Usando Restrições e Constraint Logic Programming (CLP)

Seja $G = (P, L)$ um grafo de postes para ligar. Lembremos também que para todo poste P de P , $|V(P)| \leq 6$. Assim, poderíamos ter árvores binárias de distribuição de sinal como as mostradas na seguinte figura.



Árvore de distribuição de sinal de 4 saídas.



Árvore de distribuição de sinal de 5 saídas.

Figura 5.1. Exemplos da árvores de distribuição de sinal

A árvore de distribuição de sinal de maior altura seria

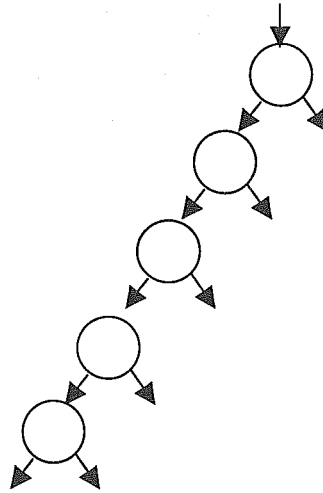


Figura 5.2

Podemos observar que existem muitas árvores de distribuição de sinal.

Cada distribuidor binário pode-se expressar com as restrições:

$$Y_1 = X - A_1$$

$$Y_2 = X - A_2$$

onde X é a intensidade de sinal que entra no distribuidor, A_1 e A_2 são as atenuações nas saídas do distribuidor e, Y_1 e Y_2 são as intensidades de sinal nas saídas do distribuidor.

Se inserirmos dois novos valores para o par (A_1, A_2) : $(0, 100)$, $(100, 0)$, então construímos um novo distribuidor binário, onde por uma saída, passa o sinal perfeitamente, enquanto pela segunda, o sinal não passa simplesmente.

Portanto, a partir da estrutura

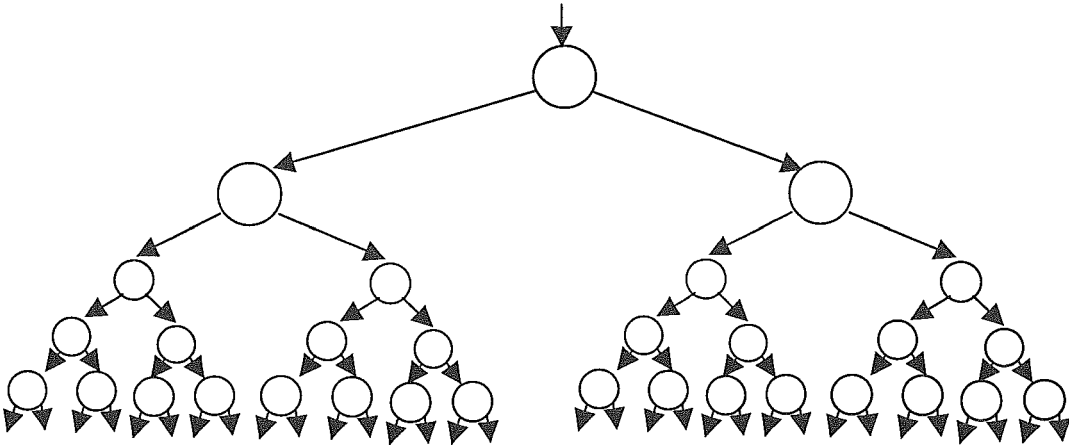


Figura 5.3

e assumindo que cada



é como uma variável que pode representar qualquer distribuidor binário (inclusive o novo), então poderemos representar qualquer árvore de distribuição de sinal com no máximo 6 saídas.

Assim, poderemos construir um programa com restrições (que não será feito nesta tese), que avalie o número ótimo de residências atendidas. De qualquer forma, problemas como o de postes ilhados precisam ser tratados.

Devemos observar, que esta modelagem depende do conhecimento do número máximo de vizinhos e que o processo de otimização vai depender do conhecimento do número de postes e arestas. Isto não acontece com o algoritmo *JCCZ*. De fato, o algoritmo trabalha com qualquer número de vizinhos, qualquer número de postes e qualquer número de arestas. E mais ainda, constrói a árvore de distribuição de sinal sem ter que pesquisar por todas as árvores de sinal existentes. Suas limitações estão na quantidade de memória do computador e a velocidade do mesmo.

Usando o Minimal Spanning Tree

Um algoritmo bastante utilizado na literatura para problemas similares é o que encontra a árvore geradora de menor peso (minimum weight spanning tree) [13]. Uma restrição para a utilização deste algoritmo para resolver o nosso problema é que a árvore mínima pode ter um fator de ramificação acentuado, o que pode acarretar uma utilização excessiva de distribuidores de sinal por poste. Além disso, nem todos os postes do grafo precisam ou podem estar ligados na árvore final. Para a utilização de minimum weight spanning tree é necessário saber com antecedência os pesos das arestas ou dos nós. No nosso caso, a intensidade de sinal e número de residências de um caminho que poderiam ser utilizados como peso, somente podem ser calculados durante a varredura do grafo, ou seja, de forma dinâmica.

O algoritmo JCCZ procura evitar o uso excessivo dos distribuidores binários tentando minimizar a quantidade a ser instalada nos diferentes postes. A cada nova aresta adicionada à árvore final, tenta maximizar o número de residências, minimizando o comprimento do caminho e a utilização de distribuidores.

Conclusões

Neste trabalho apresentamos uma modelagem formal, estruturas de dados e algoritmos para a solução totalmente automática de projeto de redes de TV a cabo (CATV). O problema consiste em passar sinal de TV por cabo utilizando como infraestrutura os postes de rua para instalação de dispositivos de distribuição e transmissão de sinal (cabo, taps, divisores e acopladores de sinal). Apesar de o problema ter sido modelado para transmissão de sinal por cabo, nossos algoritmos estão parametrizados para utilizar qualquer equipamento de transmissão (por exemplo, fibra ótica). Para a obtenção de informações detalhadas e precisas sobre o problema, entrevistamos um especialista na área de projeto de CATV. Segundo a literatura e entrevistas com especialistas, há pouca informação sobre este tipo de problema. Portanto, baseamos nossos algoritmos nas informações obtidas do especialista, mas não sem antes fazer uma modelagem formal matemática do problema para poder aplicar os algoritmos.

Além do algoritmo geral para projeto automático, apresentamos algoritmos auxiliares que escolhem o melhor dispositivo para atender residências (supertaps), procuram o melhor caminho PLUSCS a partir do poste onde foi instalado o amplificador (backbone), algoritmos para detecção de postes ilhados, algoritmos para recalcular a intensidade de sinal, número de residências e comprimento do cabo.

Para isto foi necessário definir os conceitos topológicos do problema usando como referência o livro de Schwartz [15].

Nossos algoritmos utilizam heurísticas provenientes das características do problema. Para encontrar o melhor caminho PLUSCS de postes ligados usando somente cabo e supertaps utilizamos um algoritmo branch-and-bound que verifica se o caminho parcial encontrado é melhor do que o anterior. Após encontrar este caminho PLUSCS (*backbone inicial*), utilizamos outras heurísticas – PP1 ou PP2 – para encontrar o melhor distribuidor de sinal a ser escolhido para um poste, além de heurísticas para escolher o melhor vizinho que esteja no conjunto de postes críticos. Damos prioridade a vizinhos que são terminais. Os postes fronteiraços podem não ser ligados se a intensidade de sinal não for suficiente para alcançá-los. Neste caso, marcamos estes postes para posterior tratamento em outra microcélula.

Aplicamos nossos algoritmos a exemplos reais retirados de um mapa da região do Rio de Janeiro, e conseguimos resultados melhores ou equivalentes aos obtidos com o projeto manual (este projeto manual foi realizado com software de auxílio do tipo do LODE DATA). As microcélulas de nossos exemplos têm um máximo de 7 postes e 36 residências para simplificar os cálculos manuais que validam os algoritmos. Estes demonstram o potencial para resolver o problema de forma totalmente automática mesmo para um número maior de postes e residências.

Há várias extensões e melhoramentos possíveis a este trabalho, dentre eles destacamos o problema geral de projeto de CATV que consiste em localizar os amplificadores em regiões maiores que as microcélulas. O problema que resolvemos considera que somente um amplificador é colocado em um único poste da microcélula. No problema geral, assumimos que a região é particionada em células e microcélulas. Aplicamos nossos algoritmos e encontramos uma coleção de postes fronteiraços. Com este conjunto de postes fazemos um novo particionamento, e aplicamos novamente os algoritmos. Desta forma, conseguimos resolver o problema mais geral. Quanto aos algoritmos apresentados, há muitos melhoramentos a serem realizados e assim que tivermos uma primeira implementação pretendemos testar heurísticas diferentes.

Referências Bibliográficas

- [1] XAVIER ADILSON E. *Recobrimento por Círculos de uma Região do Plano. X CLAIO – Mexico, 4-8 Setembro 2000 e 17th International Symposium on Mathematical Programming, Atlanta, EUA, 7-11 Aug.*
- [2] BARBOSA VALMIR C. *An Introduction to Distributed Algorithms.* 1 ed. The MIT press. London 1996.
- [3] BARTLETT EUGENE R. *Cable Television Technology & Operations. HDTV and NTSC systems.* 1 ed. Mc Graw Hill, 1990.
- [4] BERTSEKAS DIMITRI & GALLAGER ROBERT. *Data Networks,* 2 ed. Prentice Hall International Editions, 1992.
- [5] SANSÒ BRUNILDE. *Issues in ATM network Planning: An operations research perspective.* In: *Telecommunications network planning,* Kluwer, Norwell, MA. pp. 79-95. 1999.

- [6] COLBOURN CHARLES J.. *Reliability Issues in telecommunications network planning*. In: *Telecommunications network planning*, Kluwer, Norwell, MA. pp. 135-143. 1999.
- [7] JERROLD COMMUNICATIONS. *Dual-Output Amplifier, model MB-550D-H, Installation Manual*.
- [8] JERROLD COMMUNICATIONS. *Line Extenders, models JLX-7-600 P/LC and JLX-7-550P/LC, Instalation Manual*.
- [9] MARRIOTT KIM & STUCKEY PETER J.. *Programming with Constraints. An Introduction*. The MIT Press. 1998.
- [10] RESENDE MAURICIO. *Entrevista Pessoal*. web.
<http://www.research.att.com/mgcr>.
- [11] DUTRA RENATO C.. *Entrevistas Pessoais*.
- [12] DUTRA RENATO C.. *Redes de Telecomunicações*. Notas de aula. 2000.
- [13] ROBERTAZZI T. G., *Planning Telecommunication Networks*, 1 ed., IEEE Press, New Jersey, 1999.
- [14] SANSÒ B. & SORIANO P., *Telecommunications Network Planning*, 1 ed., Massachusetts, Kluwer Academic Publishers, 1999.
- [15] SCHWARTZ LAURENT. *Analyse*. 1ed. Hermann, 1970.
- [16] SZWARCFITER JAYME & MARKENZON LILIAN. *Estruturas de Dados e seus Algoritmos*. 2 ed. Livros Técnicos e Científicos S.A. 1994.
- [17] TENENBAUM AARON M.. *Estruturas de dados Usando C*. 1 ed. Makron Books, São Paulo, 1995.
- [18] Mapinfo. <http://www.mapinfo.com/>
- [19] LODE DATA. <http://www.lodedata.com/>