Lista de Exercícios – Redes

1) Hipercubos Desenhe um hipercubo binário (n=3,k=2) e atribua os endereços aos nós. Dois nós são vizinhos se seus endereços "distam" de uma potência de 2: $(a, 1, c) - (a, 0, c) = 2^i$. Posto de outra forma, $(a, 1, c) \rightleftharpoons (a, 0, c)$, onde $\rightleftharpoons$ indica uma conexão entre dois nós. Repita para um hipercubo de dimensão 4 (n=4,k=2).

2) Roteamento em malhas e cubos O roteamento numa malha 3D (cubo) é efetuado uma dimensão por vez cada. Por exemplo, o caminho mínimo do nó (a,b,c) até o nó (p,q,r) é percorrido primeiro na dimensão x até (p,b,c), então na dimensão y até (p,q,c), e então até o destino. Este método é chamado de *dimension-order routing*. Use este método para escolher uma rota no hipercubo 4D do exercício anterior para todas as distâncias possíveis a partir do nó (0,0,0). Pistas: os endereços dos nós diferem em apenas um bit; a operação *xor* pode ser útil.

3) Comutação em árvore gorda A rede do CM-5 usa roteamento da minhoca, com roteadores/comutadores com capacidade para 4 bits por porta. Compare a eficiência de roteamento por re-despacho com roteamento da minhoca para uma máquina com 128 nós, através da árvore quaternária, usando mensagens de 20 bytes (4+16). Cada comutador tem latência de $0,25\mu$s e a taxa de transferência é de 20MB/s.

RE-DESPACHO $L_{\text{rd}} = \#\text{comut} \times (\mathcal{T}_{\mathsf{comut}} + \mathcal{T}_{\mathsf{transfer}})$ $\mathcal{T}_{\mathsf{transfer}} = |\text{msgm}|/\text{banda}$

CUT-THROUGH $L_{\text{ct}} = (\#\text{comut} \times \mathcal{T}_{\mathsf{comut}}) + \mathcal{T}_{\mathsf{transfer}}$

WORMHOLE $L_{\text{wh}} \cong L_{\text{ct}}$ [HP QA-2E Ex pg 592]

3.1) Refaça os cálculos do exercício 2 para tamanhos 64, 256 e 1024 nós. [HP QA-2E 7.6]

4) Quais as diferenças, em termos de desempenho, das políticas de roteamento (i) armazenar e encaminhar (*store and forward*), (ii) corte transversal (*cut-through*), e (iii) *wormhole*? Quais as diferenças em termos de custo da implementação?

5) Escreva a equação da latência total para transmitir uma mensagem e explique sucintamente cada um de seus termos.

6) Roteamento em hipercubos Considere um n-cubo com $N = 2^n$ nós. Cada nó b é identificado em binário como $b = b_{n-1}b_{n-2}\cdots b_1b_0$. Assim, um nó fonte é $f = f_{n-1}\cdots f_1f_0$ e um nó destino é $d = d_{n-1}\cdots d_1d_0$. Deseja-se determinar uma rota de f para d com um número mínimo de passos. As n dimensões são denotadas como $i = 1, 2 \ldots n$ de modo que a i-ésima dimensão corresponda ao $(i-1)$-ésimo bit no endereço de um nó. Seja $v = v_{n-1}\cdots v_1v_0$ um nó qualquer na rota; a rota entre f e d é determinada univocamente como segue:

1. Para cada uma das n dimensões, compute o "bit de direção" $r_i = f_{i-1} \oplus d_{i-1}$. Inicie com $v = f$ e $i = 1$.
2. Compute a rota do nó corrente v para o próximo nó $v \oplus 2^{i-1}$ se $r_i = 1$; pule este passo se $r_i = 0$.
3. Avance para a próxima dimensão $(i \leftarrow i + 1)$; se $i \leq n$ repita o passo 2, senão terminou.

(i) Desenhe um hipercubo de 16 nós $(n = 4)$ e demonstre o algoritmo para $f = 0110$ e $d = 1101$. Compute a lista de nós ao longo da rota.

(ii) Compute a rota entre os nós $f = 101101$ e $d = 011010$ numa rede com 64 nós $(n = 6)$. Compute a lista de nós ao longo da rota. Não é necessário desenhar o hipercubo.

Cálculo Alternativo da Rota *(Li & McKinley, Computer, fev93)* com o *roteamento E-cube*: cada nó é representado por um número binário de n bits. Cada nó tem n canais de saída e o i-ésimo canal corresponde à i-ésima dimensão. Um pacote carrega o endereço d do destino. Quando um nó v recebe um pacote, computa $c = d \oplus v$; se $c = 0$ entrega o pacote ao processador local; caso contrário, retransmite o pacote no canal de saída k, sendo k a posição do bit em 1 mais significativo (mais à direita) em c.