

MULTIHOMING, MULTICAST E MOBILIDADE

Capítulo 9

Patterns in Network Architecture

- *Ela (a terra) se move.*
- *Galileo Galilei*

3

Introdução

Onde estamos

- No Capítulo 5 olhamos para a nossa compreensão sobre a nomeação e o endereçamento
 - ▣ Nossos *insights* a partir dos resultados de Saltzer e
 - ▣ da nossa compreensão das camadas superiores, no Capítulo 4.
- No Capítulo 6, percebemos que os endereços não eram exatamente o que pensávamos que fossem: não são apenas nomes das máquinas de protocolos, mas são identificadores *internos* a um recurso IPC distribuído.

Onde estamos

- No Capítulo 7, montamos um modelo de arquitetura baseado no que aprendemos nos capítulos anteriores, montando os elementos para uma arquitetura completa de nomeação e endereçamento.
- Finalmente, no Capítulo 8, consideramos o que significa ser dependente de localização em uma rede e mostramos como os conceitos de topologia se aplicam ao endereçamento que usados em um arquitetura recursiva podem criar um esquema de roteamento escalável e efetivo.

Para onde vamos

6

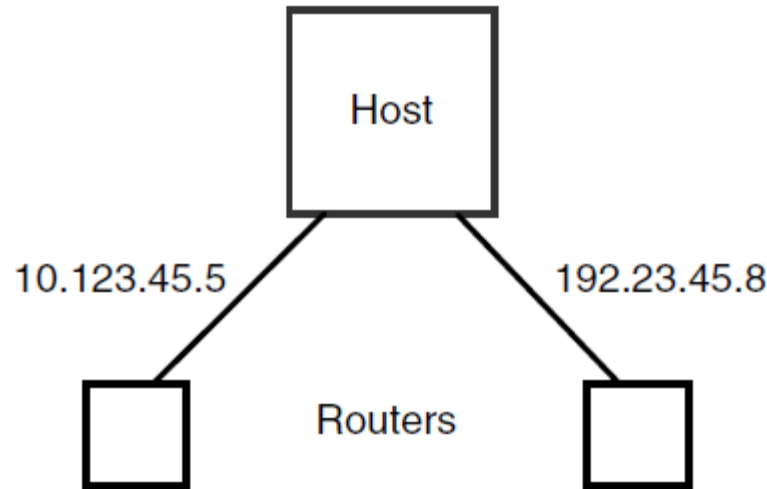
- Agora devemos considerar alguns outros tópicos relacionados com o endereçamento:
 - *Multihoming*
 - *Multicast*
 - *Mobilidade*

7

Multihoming

Multihoming

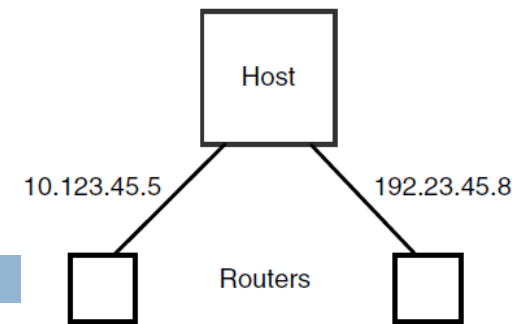
8



- *Multihoming* acontece quando um host possui mais de uma conexão à rede.
- Por usar um endereço diferente para cada interface, os roteadores não têm como identificar que as duas interfaces dão no mesmo lugar.

Multihoming

9



- O host pode escolher qual das interfaces utilizar ao abrir uma conexão, por exemplo, para balancear a carga.
- No entanto, se um dos links falhar, falharão todas as conexões que passam pelo mesmo.
 - ▣ Qualquer tráfego em trânsito para aquele host será perdido.
- A solução é clara:
 - ▣ Precisamos de um espaço de endereçamento lógico sobre o espaço de endereçamento físico.
 - ▣ E isto já estava contemplado na proposta de Saltzer.

Outras propostas

- SCTP (*Stream Control Transmission Protocol*)
 - Definido na RFC 3286 atualizada pelas RFCs 6096 e 6335.
 - Resolvem o problema mudando a definição para algo que podem resolver...
 - O que o SCTP provê é a habilidade de mudar o endereço IP (ou seja, o PoA) sem interromper a conexão de transporte.
 - A máquina de protocolo do transmissor (transporte) deve estar ciente da mudança do roteamento.

Outras propostas

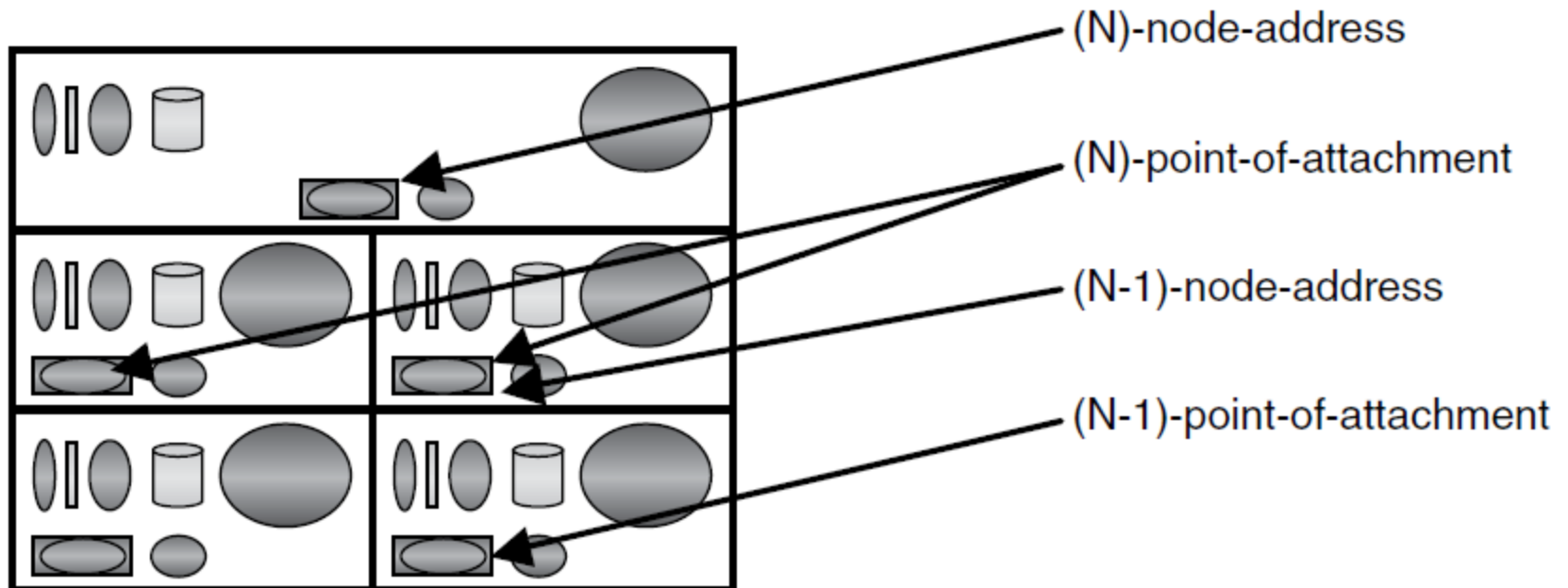
11

- BGP (*Border Gateway Protocol*)
 - ▣ Definido na RFC 1771
 - ▣ Protocolo de roteamento interdomínio
 - Troca informações de roteamento sobre ASes.
 - ▣ A solução seria tratar o AS como um endereço do nó.
 - Um host ou sítio que queira ter múltiplas conexões adquire um número de AS do seu ISP e anuncia rotas para o mesmo através do BGP.
 - ▣ Desvantagens:
 - O escopo do BGP é toda a Internet
 - Explosão do número de ASes

Modelo proposto

12

- O modelo construído adota a abordagem de Saltzer generalizando-a no contexto do modelo recursivo.
- Um endereço-(N) é um endereço de nó na camada-(N) e um endereço PoA para a camada-(N+1).



Suporte a *Multihoming*

- *Multihoming* é suportado como consequência da estrutura.
- O roteamento é realizado em termos dos endereços dos nós com a seleção do caminho feito como um passo separado.
 - ▣ Esta separação ocorre em termos do cálculo da rota para gerar a tabela de repasse, e não no repasse propriamente dito.
- Ao contrário da Internet atual, suporta *multihoming* tanto para hosts como para roteadores.
 - ▣ Isto reduz significativamente o número de rotas que devem ser calculadas.
- Portanto, a estrutura recursiva garante que os mecanismos vão escalar.

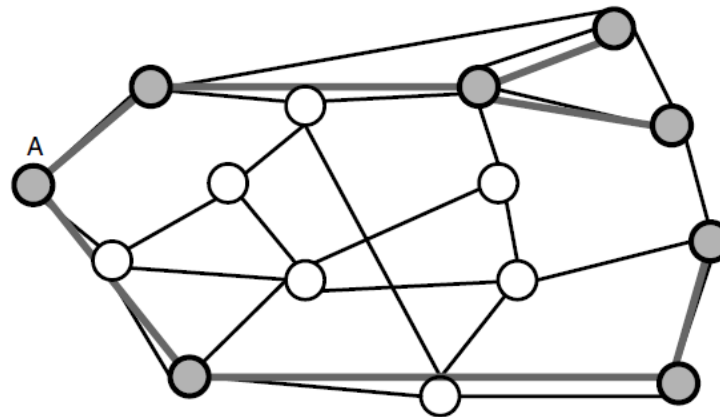
14

Arquitetura *Multicast*

Multicast

15

- Habilidade de enviar uma única PDU para um conjunto selecionado de destinos.
 - ▣ Incluiremos aqui o caso do *broadcast* que é facilitado em meios compartilhados.
- Árvore de expansão enraizada em A para o conjunto de membros do grupo *multicast*.



- É basicamente um modo de economizar largura de banda.
 - ▣ As vantagens para o usuário são mínimas.

Características do *Multicast*

16

- Características:
 - Centralizada x Descentralizada
 - População estática x dinâmica
 - População conhecida x desconhecida
 - Isotrópica x anisotrópica (todos os membros se comportam da mesma forma?)
 - Quórum
 - Confiável x não confiável
 - Simplex x duplex

Características do *Multicast*

- Dentro destas características existem múltiplas políticas que podem ser usadas, tornando a faixa de protocolos potenciais bem grande.
 - ▣ Não dá para esperar que uma única solução acomode eficientemente toda esta faixa.
 - ▣ Mas, criar soluções específicas para aplicações específicas levaria a uma proliferação e complexidade dentro da rede, o que não é bem vista pelos provedores!

○ Modelo *Multicast*

18

- *Multicast* é visto como comunicação com um grupo (um conjunto de usuários).
 - ▣ Alguns modelos assumem que todos os participantes são membros do grupo (multiparceiro)
 - ▣ Outros assumem que o grupo é uma entidade com a qual outros, fora do grupo, podem se comunicar.

APIs

19

- Primitivas unicast:
 - ▣ *Connect (allocate), send, receive e disconnect*
- Primitivas multicast:
 - ▣ *Connect* – cria o grupo e torna o iniciador um membro do grupo
 - ▣ *Join* – usado para entrar em um grupo existente
 - ▣ *Leave* – usado para deixar um grupo existente
 - ▣ *Send*
 - ▣ *Receive*
 - ▣ *Disconnect* – encerra a participação no grupo

Modelo de serviço

20

- A aplicação tem que saber se a comunicação é *unicast* ou *multicast*!
 - ▣ Um grupo *multicast* com dois membros é diferente de uma comunicação *unicast*!
 - ▣ Seria melhor um modelo no qual um dos modos “colapsa” naturalmente no outro.
 - ▣ Ou, ainda mais forte: **por que a aplicação precisa estar ciente do uso do *multicast*?**

Modelo de Serviço

21

- A função principal da primitiva “*connect*” é criar o grupo.
 - ▣ Criar um estado compartilhado suficiente dentro da rede para permitir que uma instância de comunicação seja criada.
 - Isto é o que chamamos de **fase de registro**.
 - ▣ Se o modelo de serviço for reformulado de modo que criar o grupo seja uma função de registro
 - Então “*join*” e “*leave*” tornam-se sinônimos de “*connect*” e “*disconnect*”.

Modelo Único para Três Modos de Comunicação

22

- A sequência de interação (máquina de estados parcial associada com a interação do serviço) é a mesma tanto para *multicast* como para *unicast* e sem conexões.
- ▣ A única diferença seria que o parâmetro que nomeia o destino nomearia um conjunto de nomes ao invés de um único nome (ou seria este um conjunto com um único elemento?).

“Endereçamento” *Multicast*

- Um “endereço” *multicast* é o nome de um conjunto de endereços.
 - ▣ Referenciar o conjunto é equivalente a referenciar todos os membros do conjunto.
- Os primeiros protocolos *Multicast* da Internet viam um endereço *multicast* essencialmente como um endereço ambíguo ou não único.
 - ▣ Era uma tentativa de imitar a semântica dos endereços *broadcast* ou *multicast* de uma LAN.
 - ▣ Isto tornava a inundação a única estratégia que poderia ser usada.
 - Mas o seu custo eliminava a vantagem do *multicast* em economizar largura de banda numa rede.

Nomes *Multicast*

- Endereços multicast não são endereços.
 - ▣ Da forma como os definimos, os endereços são dependentes da localização.
 - ▣ “Endereços” *multicast* são nomes de um conjunto
 - Os elementos do conjunto são endereços,
 - Mas o nome do conjunto não pode ser um endereço.
- A aplicação teria um nome de aplicação *multicast* definido como um conjunto de nomes de aplicações.
 - ▣ Isto já foi definido como nome-aplicação-distribuída.
 - ▣ Este nome seria passado num pedido de abertura para o DIF, que alocaria um nome *multicast* a partir do espaço de endereçamento dos DIFs.
 - Este nome *multicast* nomearia o conjunto de endereços ao qual as aplicações estariam associadas.

Distribuição *Multicast*

25

- O principal foco da pesquisa sobre *multicast* tem sido resolver o problema da distribuição *multicast* na camada de rede para redes sem meio físico de múltiplo acesso.
- Abordagens:
 - ▣ Inundação!
 - ▣ Algoritmos distribuídos para gerar árvores de expansão.
- Interesse em comunicação “multiparceiros”.

Problemas

26

- Escalar as técnicas para dar suporte a grandes grupos
- Manutenção de uma árvore de expansão ótima ou quase ótima com mudanças no grupo de membros
- Redução da complexidade do multiparceiros não exigindo o cálculo de uma árvore de expansão distinta para cada membro do grupo

Protocolos *Multicast* Padronizados

27

- DVMRP (RFC 1075)
- PIM (RFC 2362)
- CBT (RFC 2189)
- O foco destes algoritmos é o de encontrar árvore de expansão ótima dados múltiplos transmissores no grupo.
 - ▣ A raiz da árvore de expansão não está em um dos transmissores,
 - ▣ Mas está em um “centro de gravidade” para o grupo.
 - O problema se torna então encontrar este centro de gravidade.

Operações de Nomeação Sentencial

28

- Duas formas:
 - ▣ “universal” (i.e., *multicast*)
 - ▣ “existencial”: nome “*anycast*”.
 - Nome de um conjunto de modo que quando o nome é referenciado, seja retornado um elemento do conjunto de acordo com alguma regra associada ao conjunto.
- O uso de nome sentencial é resolvido mais cedo ou mais tarde para um endereço:
 - ▣ A regra associada com o conjunto é aplicada quando a tabela de repasse é criada para produzir uma lista de endereços que satisfaçam a regra.
 - ▣ Quanto uma PDU com um endereço de destino sentencial é analisada em cada *relay*, ela então é enviada para os elementos desta lista (naturalmente a lista pode ter apenas um elemento).

Operações de Nomeação Sentencial

29

- *Anycast*:
 - Será selecionado um endereço do conjunto e repassado para aquele endereço.
- *Multicast*:
 - Serão selecionados os endereços a jusante na árvore de expansão e repassada uma cópia para cada um de seus ramos.
 - Com o uso de endereços topológicos, os endereços no conjunto *multicast* podem ser ordenados numa árvore de expansão virtual baseada na topologia do espaço de endereços.
 - Isto simplificaria a tarefa de cada nó, na qual, dado o próprio endereço conhece onde está na árvore de expansão daquele conjunto *multicast* e para que ramos deve repassar cópias da PDU.
 - Deste modo, protocolos *multicast* especializados são desnecessários.
 - Mas ainda precisamos dos algoritmos distribuídos de árvore de expansão.

Operações de Nomeação Sentencial

30

- A definição dos conjuntos *multicast* terão que ser distribuídos a cada membro do DIF junto com as políticas associadas com o mesmo e o algoritmo de árvore de expansão aplicado.
 - ▣ Esta informação é então usada para gerar entradas na tabela de repasse.
- *Unicast* é um subconjunto do *multicast*, mas o *multicast* se reduz a *unicast*.

Distribuição *Multicast* em uma Arquitetura Recursiva

31

- Um grupo *multicast* na camada N , $(N) - G$, é definido como um conjunto de endereços $\{(N) - A_i\}$, de modo que uma referência a $(N) - G$ por um membro do grupo produza todos os elementos de $\{(N) - A_i\}$.
- $(N)-G$ é identificado por um endereço-grupo- (N) , $(N)-GA$.
 - Nome-grupo- $(N) \forall (N) - A_i \ni (N) - A_i \in G = \{(N) - A_i\}$
- A ordem de um $(N) - G$ (ou seja, o número de elementos no conjunto) é $|\{(N) - A_i\}|$ ou $|(N) - G|$.

Árvore de Expansão Mínima

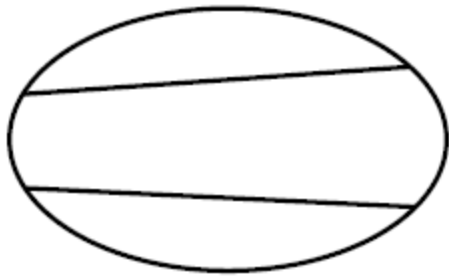
32

- Uma árvore de expansão mínima é um grafo acíclico $T(n, a)$ que consiste de n nós e a arcos imposto em um grafo mais geral da rede.
 - ▣ Esta árvore representa o número mínimo de arcos necessários para conectar (ou cobrir) todos os membros de $(N) - G$.
 - ▣ As folhas ou nós terminais da árvore são os elementos do conjunto $\{(N) - A_i\}$.
 - ▣ Os nós não terminais são *relays* intermediários na camada-(N).

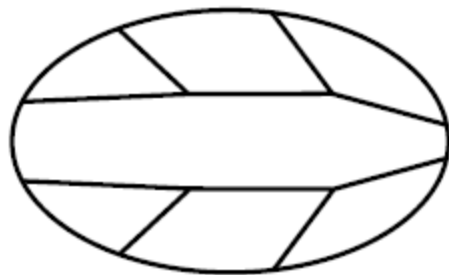
Grupos Multicast

33

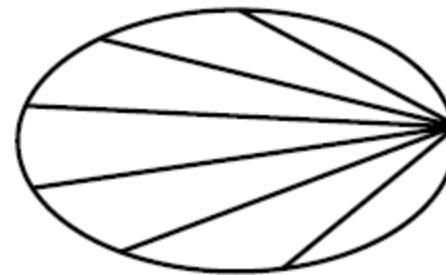
Sub-redes de trânsito, que são *unicast*:



Sub-redes de borda, que usam árvores de expansão:



Este segundo caso é mais simples se os fluxos *unicast* do roteador de borda forem usados sem nenhuma ramificação interna (tornando a distribuição *multicast* em um subconjunto do roteamento *unicast*):



Multiplexando Grupos *Multicast*

34

- *Multicast* sempre foi considerado um problema tão difícil que multiplexar grupos *multicast* para se obter uma maior eficiência nunca foi considerado.
- ▣ Mas, dada uma abordagem que nos permite decompor o problema, a multiplexação se torna algo mais direto.
- ▣ Há dois casos a serem considerados:
 - Grupos de trânsito e
 - Grupos finais

Multiplexando Grupos *Multicast*

35

- O número de subredes é muito menor do que o número de *hosts* que participam de um grupo, o mesmo é verdade para o número de *relays*-(N).
 - ▣ Portanto, é provável que existam alguns grupos de trânsito que possuam fluxos entre os mesmos *relays*-(N).
 - Estes são os candidatos a serem multiplexados.
- É interessante multiplexar árvores “similares”.

Multicast Confiável

36

- Se os algoritmos de distribuição *multicast* forem a camada de rede, então como devem se parecer os mecanismos para que um protocolo de transporte *multicast* fornecesse confiabilidade fim a fim?
 - ▣ Podemos adaptar protocolos como o TCP para prover confiabilidade entre múltiplos parceiros?

Multicast Confiável

37

- Questões a serem consideradas:
 - ▣ O que ocorre se um dos membros fica para trás e não pode aceitar dados o suficientemente rápido?
 - Os dados para os mais lentos serão descartados?
 - O membro mais lento será descartado?
 - Os dados para o mais lento serão bufferizados?
 - Por quanto tempo?
 - ▣ Serão realizadas retransmissões para todos os membros do grupo ou apenas para aqueles que enviaram nack ou para os quais estourou o temporizador?
 - Neste segundo caso, isto não elimina a vantagem do *multicast*? Ao torná-lo em $(m - 1)$ conexões *unicast*?

Multicast Confiável

38

- Questões a serem consideradas (cont.):
 - ▣ O que ocorre se um membro não receber nada depois de diversas tentativas?
 - O membro deve ser removido?
 - Todo o grupo deve ser terminado?
 - ▣ Múltiplos transmissores podem transmitir ao mesmo tempo?
 - Neste caso, que ordem deve ser mantida se houver diversos transmissores no grupo?
 - Relativa? Ordenação parcial? Ordenação total? Nenhuma ordem?
 - ▣ Se o grupo possui inscrições dinâmicas e um novo membro entra no grupo, a partir de que ponto ele começa a receber os dados?
 - E se ele sair e depois voltar, a resposta seria diferente?

Multicast Confiável

39

- Pode-se verificar que muitas, se não todas estas seriam desejáveis para alguma aplicação.
 - ▣ A estratégia de separar mecanismo de política desenvolvido aqui pode ser usado para endereçar pelo menos parcialmente estas questões.
- Há ainda outros problemas:
 - ▣ Uma característica importante dos protocolos de controle de erro e de fluxo é que eles possuem mecanismos de realimentação.
 - ▣ Isto leva ao problema de **implosão de acks**.

Multicast confiável

- Na estrutura desenvolvida aqui:
 - ▣ O protocolo de controle de erro e de fluxo, assim como
 - ▣ Os processos de repasse e multiplexação
 - ▣ Estão todos no mesmo DIF e podem compartilhar informação.
- Dado que o EFCP é estruturalmente dois protocolos, as PDUs de controle podem ser roteadas separadamente das PDUs de transferência.
 - ▣ Uma árvore de expansão pode ser usada para dar suporte à transferência de dados
 - ▣ Enquanto uma árvore de expansão separada em paralelo dá suporte ao fluxo de controle com máquinas de protocolo em cada nó agregando-as à medida que retornam na árvore.

Multicast Confiável

- Assumindo que há aplicações que necessitam de confiabilidade, parece existir a necessidade de rever e possivelmente refinar a teoria dos EFCPs em relação aos graus de integridade (fim-a-fim) e o papel do processamento nos intermediários.

42

Mobilidade

Formas comuns de Mobilidade

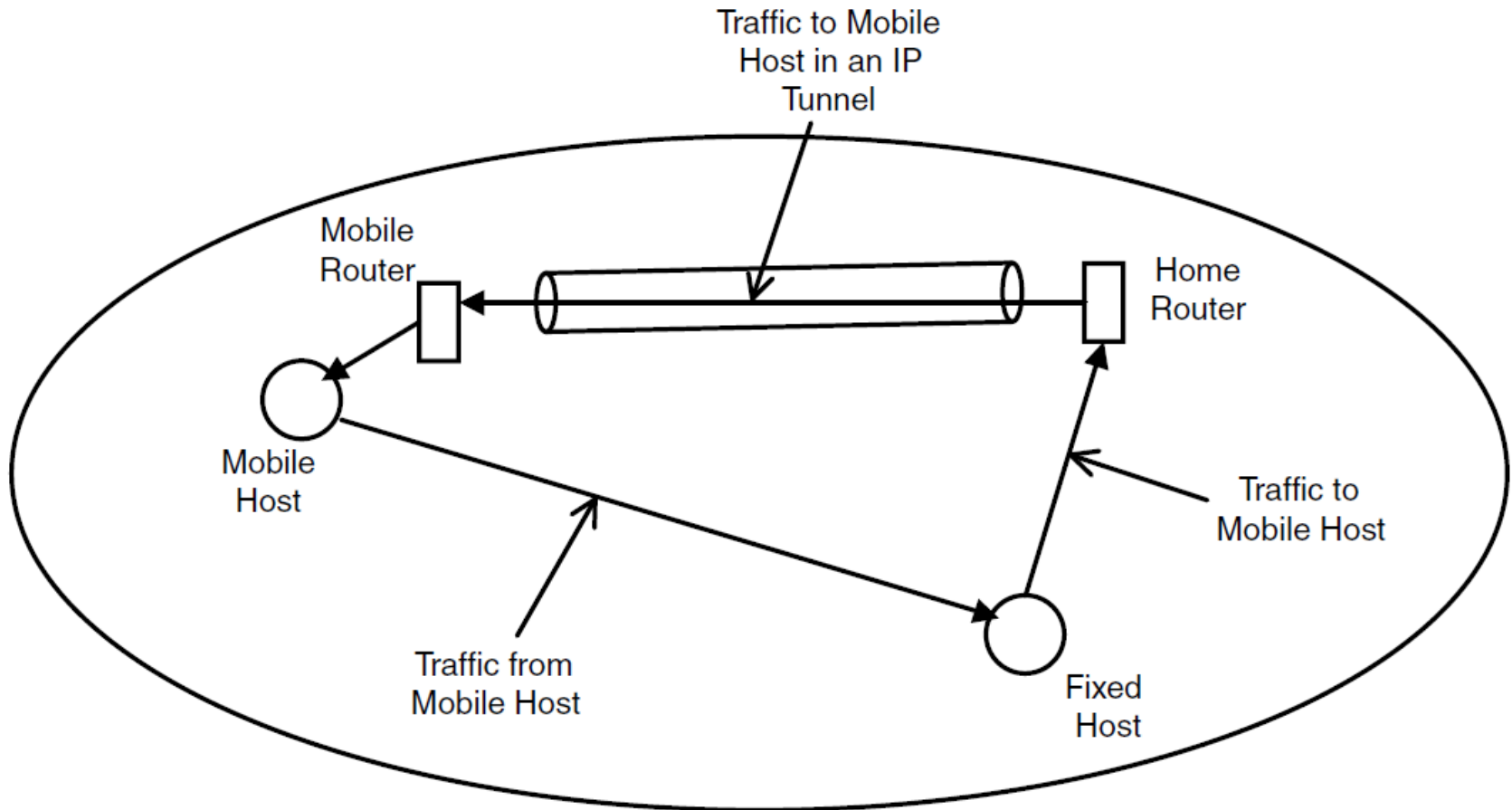
43

- Hosts ou sub-redes de borda móveis em relação a uma rede fixa,
 - ▣ por exemplo, redes tradicionais de celulares.
- Hosts ou sub-redes móveis em relação umas às outras,
 - ▣ por exemplo, redes ad hoc.
- Aplicações que se movem de host a host independentemente dos hosts serem fixos ou móveis.

- As aplicações móveis pela sua natureza estão confinadas à periferia de uma rede ou para um número relativamente pequeno de redes independentes ou sub-redes.

Mobilidade no IP

44



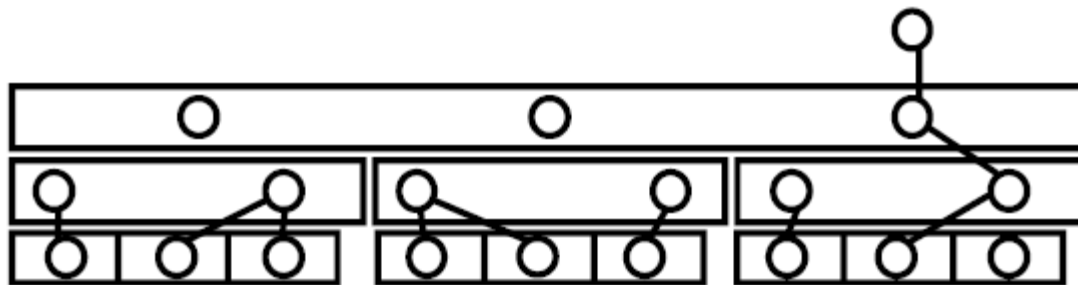
Mobilidade no NIPCA

45

- Um sistema móvel é como um macaco saltando de galho em galho através da selva
 - ▣ O sistema móvel balança de torre para torre.
- A diferença principal entre os sistemas de rede e os sistemas celulares é que no sistema celular é o galho que decide quando largar!
 - ▣ Obviamente iríamos preferir que o macaco decida quando largar o galho.
- Um sistema móvel adquire novos PoAs físicos à medida que se move.
 - ▣ Estritamente falando, cada um destes novos PoAs físicos está se unindo a um novo DIF ou camada.

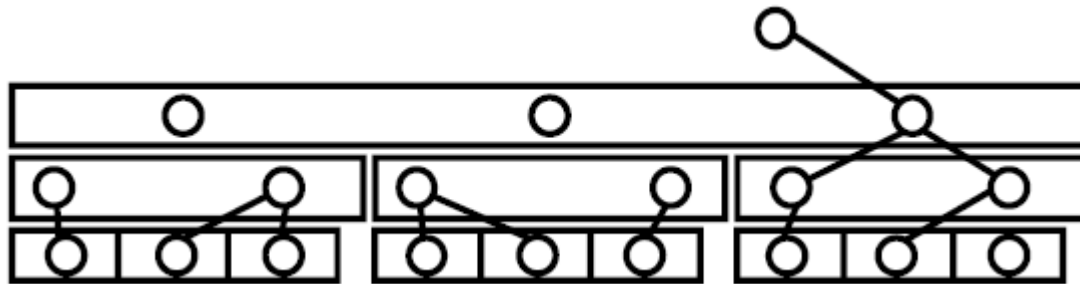
Exemplo de Mobilidade

46



Exemplo de Mobilidade

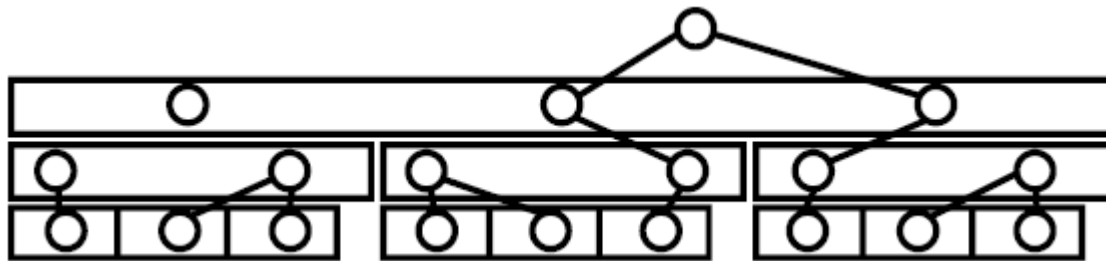
47



À medida que se move, seu ponto de conexão muda e pode ficar com múltiplas conexões (*multihomed*)

Exemplo de Mobilidade

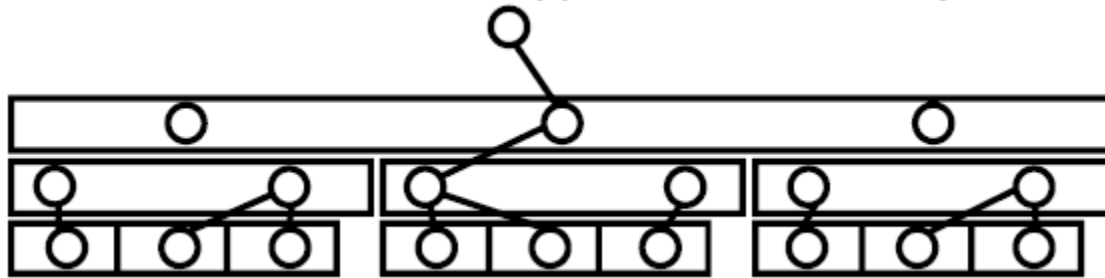
48



Antigas conexões são liberadas e
novas conexões são estabelecidas

Exemplo de Mobilidade

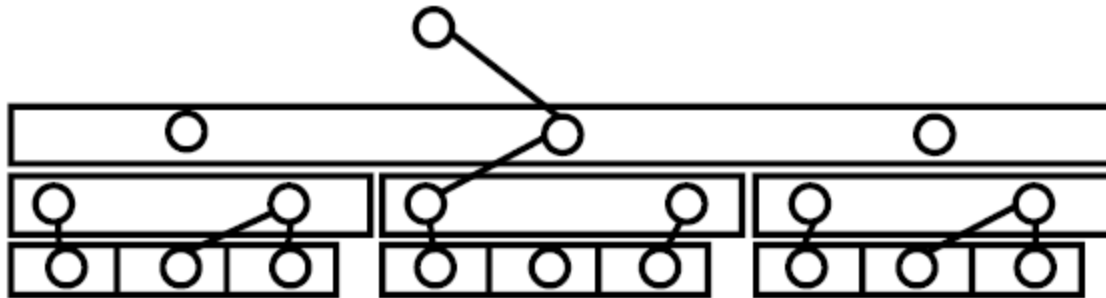
49



O movimento provoca atualizações nas informações de associação. Mudanças mais rápidas ocorrem em DIFs com menor escopo e menor número de elementos podendo ser realizadas mais rapidamente.

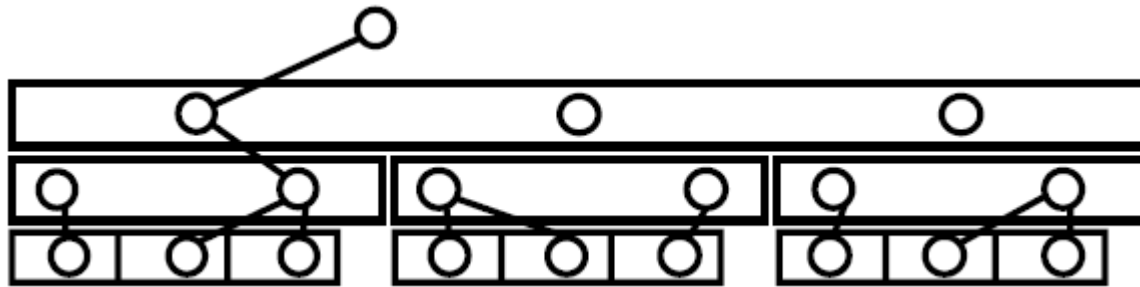
Exemplo de Mobilidade

50



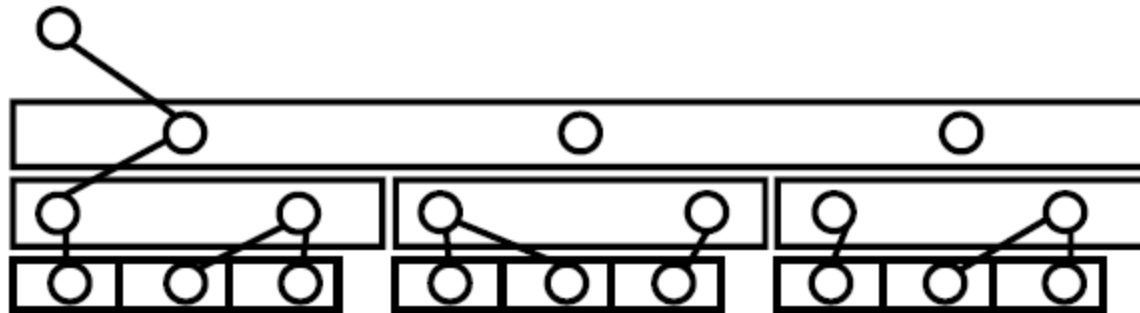
Exemplo de Mobilidade

51



Exemplo de Mobilidade

52



Mobilidade

53

- Embora gostemos de falar na mudança de endereços associados a sistemas móveis para refletir a sua nova localização na camada,
 - ▣ É mais correto dizer que o sistema móvel adquire novos endereços e deixa de utilizar os antigos.
- Quantas camadas deve ter um sistema móvel?
 - ▣ Tantas quantas forem necessárias.
 - ▣ A resposta é fundamentalmente um problema de engenharia. Depende de:
 - Tamanho da rede
 - Taxa de aquisição de novos endereços.
 - ▣ Dada a natureza recursiva da arquitetura, teremos número diferentes de camadas em diferentes partes da rede
 - mais em áreas mais densamente usadas, e menos em áreas menos densamente usadas

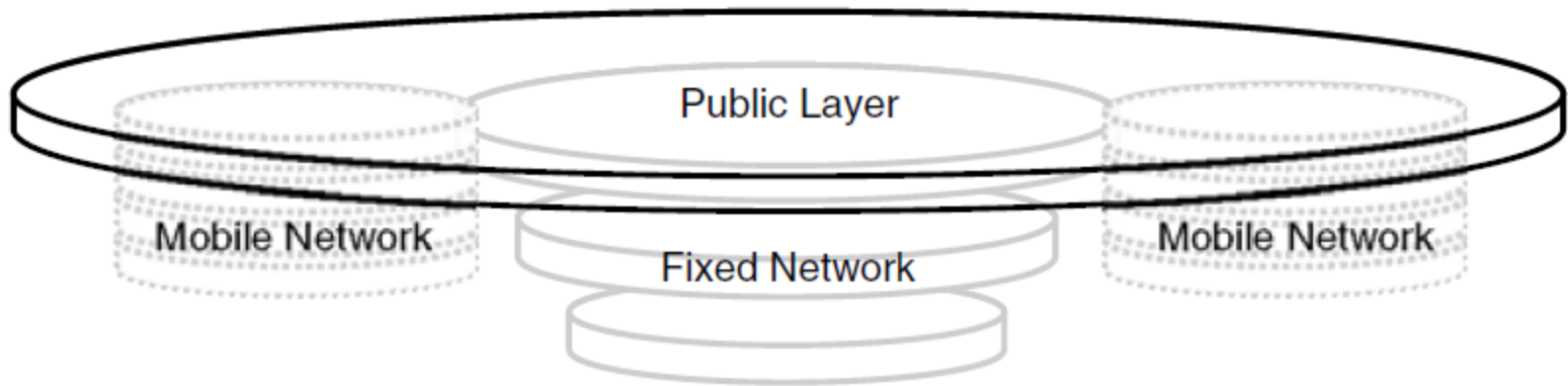
Mobilidade

54

- Haverá algum problema com a propagação destas mudanças de endereço através da rede?
 - ▣ Há uma tendência em ver estas atualizações como sendo de DNS. Mas, esta é uma atualização de roteamento.
 - ▣ As mudanças só precisam ser propagadas para outros processos IPC dentro do (N)-DIF.
 - Quanto mais alta for a camada, menor será a taxa de mudança.
 - As mudanças serão maiores naquelas camadas com menor escopo e onde os tempos para propagar as atualizações são os menores.

Relacionamento com as Camadas de Colinas e Vales

55



- Camadas empilhadas para uma rede móvel e as camadas de colinas e vales de uma rede fixa que dá suporte a uma rede pública comum.

Redes Móveis Ad Hoc

56

- A operação de uma camada em uma rede ad hoc funciona da mesma forma
- Os limites da camada-(1) são determinados pela camada física.
- Os limites das camadas mais altas possuem menos restrições e são determinadas por considerações de localidade, gerenciamento, padrões de tráfego, etc.
- Geralmente as sub-redes são formadas por elementos na mesma área.
- Há questões específicas relacionadas às políticas e que não impactam a arquitetura.

Redes Móveis Ad Hoc

57

- Podem ser alocados endereços topológicos a uma rede ad hoc?
 - ▣ Isto irá depender da rede ad hoc específica.
 - ▣ Ela é grande o suficiente para ser vantajoso o uso de endereços topológicos?

Processos Móveis de Aplicação

58

- Processos de aplicação que se movem de host para host.
- Estamos interessados no que o modelo nos diz sobre esta forma de mobilidade.
- Obviamente o único caso que nos interessa é de aplicações que se movem enquanto estão sendo executadas.
 - ▣ Todo o estado deve ser “recuperado” pelo processo de aplicação
 - ▣ Para o IPC não há diferenças entre um fluxo recuperado e um novo fluxo.

Processos Móveis de Aplicação

59

- O nosso interesse não está em mover processos em execução
- E sim inicializar uma nova instância do processo em um outro sistema e
 - ▣ Então sincronizar o seu estado com a cópia existente.
 - ▣ Depois a cópia existente pode desaparecer.
- **Aplicações móveis são apenas uma forma de multicast!**

Exemplo

60

- Você está fazendo uma comunicação de voz através do seu computador de mesa.
- No entanto, você deve sair para uma outra reunião mas quer continuar a chamada.
- Para mover a sua chamada de voz você faz com que o seu celular entre na conversa.
- O telefone cria uma instância da comunicação de voz no seu celular.
- A comunicação agora é um grupo multicast com três participantes.
- Quando você estiver satisfeito com o seu celular na chamada, o computador de mesa pode deixar a comunicação
 - ▣ Ou pode continuar gravando-a, por exemplo.

Dependência/Independência do Sistema

61

- Endereços-(N-1) (PoAs) são dependentes de localização e também dependentes de interface ou rota.
- Endereços-(N) (endereços dos nós) são dependentes de localização mas independentes de interface ou de rota.
- APM-ids são independentes de localização mas são dependentes do sistema
- Nomes AP são independentes da localização e do sistema.