



**Universidade Federal do Pará**  
**Departamento de Engenharia Elétrica e de Computação**

**INTRODUÇÃO**

Nos últimos anos, o mundo tem vivido um constante crescimento no uso de transferências eletrônicas de dados, a tal ponto de superar a de qualquer outro tipo de comunicação. Devido à variedade de aplicações a que estes dados são submetidos o volume e a importância que estas transferências de dados apresentam, também cresce proporcionalmente. Uma dessas aplicações desenvolvidas nos últimos anos são as aplicações móveis, ou seja, a capacidade de se trocar dados estando em qualquer lugar possível sem a necessidade de se recorrer a um terminal fixo, são as chamadas redes sem fio.

Para se atender a este novo tipo de aplicação seria necessária a criação de um novo modelo de protocolos de rede, porém como isto implicaria em mudanças muito trabalhosas e caras, se pensou num tipo modificado dos protocolos já existentes. Tomando-se como ponto de partida a estrutura de camadas já existentes (modelo OSI), tem-se a camada física responsável por criar 1s e 0s no meio físico (neste estudo trata-se do meio terrestre), em seguida a camada de enlace que prove transporte confiável em um link físico e a camada de rede que compreende a camada de Internet com seu Protocolo de Internet (IP) que não foi projetado para suportar a mobilidade, entretanto, devido ao seu grande uso e popularidade, foi iniciado, por volta de 1995, o desenvolvimento do IP Móvel.

Entretanto dar suporte à mobilidade apenas na camada de rede com o IP móvel não é suficiente para proporcionar suporte à mobilidade para todas as aplicações. Em vista disto foi desenvolvido o TCP móvel, estando este na camada de transporte e trabalhando junto com o IP Móvel para proporcionar à mobilidade tanto almejada pelos usuários de redes sem fio.

Além das redes sem fio, novos serviços têm surgido para um novo segmento de redes móveis, a telefonia celular. As redes móveis atuais não provêm a flexibilidade necessária quando se deseja implementar um novo serviço deste segmento. Assim, para atender a competitividade no mundo atual e a busca constante para manter clientes satisfeitos, além da conquista de novos, fez-se necessário a criação de uma nova tecnologia.

Assim, surgiu o WAP, com o propósito de disponibilizar variados tipos de informações, dispensando a conexão à rede telefônica com fio. Tal tecnologia já está sendo utilizada na Europa, Estados Unidos, Japão e no Brasil. Agora, usuários podem utilizar seus aparelhos



## **Universidade Federal do Pará**

### **Departamento de Engenharia Elétrica e de Computação**

celulares com recursos próprios da conhecida Internet, pois estes são capazes de exibir páginas *web*. É a chamada “Internet de bolso” que vem revolucionando o mercado e ocasionando a união das empresas da área comercial, tecnológica e de telecomunicações para a criação de uma rede que alcance tanto os usuários já acostumados com a Internet como aqueles que ainda nem possuem computador ou nunca acessaram a rede.

Em vista disso, tais tecnologias foram escolhidas como objeto de estudo deste trabalho. Tendo ainda a imensa dificuldade de se obter materiais em português sobre os assuntos tratados, propôs-se a elaboração deste trabalho de conclusão de curso com o objetivo de contribuir como material didático para utilização nos Cursos Avançados de Ênfase em Telecomunicações do Curso de Graduação em Engenharia Elétrica da Universidade Federal do Pará, bem como em Cursos de Atualização de Engenheiros Eletricistas, ou profissionais de áreas afins.

No capítulo 1 será abordada a camada de rede móvel, apresentando o protocolo de IP Móvel projetado para habilitar mobilidade na internet sem mudar sistemas fixos existentes, o DHCP como um mecanismo completamente automático para o nó móvel adquirir toda a informação necessária para ser integrado em uma rede, e finalmente as redes *ad hoc* oferecendo um modo completamente novo de comunicações móveis que vem atender os usuários de redes que não podem depender de uma infra-estrutura já existente.

No capítulo 2 será abordada a camada de transporte móvel com o protocolo TCP Móvel orientado a conexão em um ambiente móvel. Várias abordagens do TCP tradicional serão apresentadas e também as limitações do TCP e a evolução para o TCP Móvel.

No capítulo 3 será abordado o suporte à mobilidade para dispositivos portáteis, apresentando a nova tecnologia WAP bem como sua correspondência com o modelo de internet atual, WWW, e também fazendo uma descrição sucinta de todas as camadas e seus respectivos protocolos, sendo apresentado o ambiente de aplicação sem fio e suas linguagens WML e WMLScript, abordando-se algumas aplicações de telefonia sem fio.



# Universidade Federal do Pará

## Departamento de Engenharia Elétrica e de Computação

### CAPÍTULO 1

#### CAMADA DE REDE MÓVEL

##### 1.1 – INTRODUÇÃO

O IP móvel é um protocolo da camada de rede capaz de suportar a mobilidade da rede Internet atual. Quando os usuários com seus *laptops* e *notebooks* viajam por todo o mundo e assim realizam conexões com dezenas de diferentes redes surge a necessidade de camadas com protocolo IP (*Internet Protocol*) que suportem esta mobilidade.

Este capítulo foi baseado nas referências bibliográficas [1] e [2]

##### 1.2 – OBJETIVOS, SUPOSIÇÕES E EXIGÊNCIAS

A computação móvel é o paradigma do futuro. A Internet é a rede global de comunicações utilizada por centenas ou milhares de usuários. Visto isso, vem a idéia de simplesmente usar computadores móveis na Internet, entretanto, depararíamos com o problema de transmissão de pacotes, pois, assim que o computador saísse de sua rede local não receberia mais os pacotes da mesma forma. A razão disso é simples, quando se considera mecanismo de roteamento na internet. Um *host* envia um pacote IP com o cabeçalho contendo o endereço destino e outros campos preenchidos. O endereço destino não somente indica o receptor do pacote, mas também a rede física desse receptor. Por exemplo, o endereço destino 129.13.42.99 mostra que o receptor pode estar conectado na sub-rede física com o prefixo de rede 129.13.42. Roteadores na Internet lêem o endereço destino do pacote entrante e enviam este de acordo com a consulta em tabelas internas. Para evitar uma explosão de tabelas de roteamento, somente os prefixos são armazenados e posteriormente otimizações são aplicadas. De outro modo um roteador deveria ter armazenado os endereços de todos os computadores conectados a Internet, o que é absolutamente impraticável. Enquanto o receptor estiver sendo conectado dentro de uma sub-rede física ele envia pacotes, assim que ele se move para fora desta sub-rede, não irá alcançar mais nenhum pacote. Assim, um *host* precisa do chamado endereço topologicamente correto.



### 1.2.1 - SOLUÇÕES RÁPIDAS

Não é muito provável uma solução rápida para o problema de designação de um endereço IP topologicamente correto para um novo computador. Assim, o computador movendo-se para uma nova localização iria significar um novo endereço. Agora o problema é que ninguém conhece esse novo endereço. É quase impossível encontrar um *host* (móvel) dentro da Internet o qual tem apenas mudado seu endereço. Especialmente, o DNS (*Domain Name System*) precisa de um tempo antes de enviar dados de suas tabelas internas necessárias para o mapeamento de um nome lógico para um endereço IP. Esse acesso não funciona se o nó móvel mover-se com muita frequência. Entretanto, a internet e o DNS não foram projetados para freqüentes mudanças. Imagine milhões de nós movendo-se ao mesmo tempo. O DNS nunca poderia apresentar uma visão consistente de nomes e endereços, ele usaria *caching* para melhorar a estabilidade. Isto é demais dispendioso para rápidas mudanças.

Entretanto, há um sério problema com protocolos de camada superiores como o TCP que precisa do endereço IP. Mudar o endereço IP enquanto ainda tem uma conexão TCP aberta significa quebrar a conexão. Uma conexão IP pode ser identificada pelo conjunto endereço IP de origem, porta de origem, endereço IP de destino e porta de destino, também conhecida como um *socket*. Por isso, uma conexão TCP não pode sobreviver a mudanças de endereços. Quebrar as conexões TCP não é uma opção, usando programas como *telnet* seria impossível. Adicionalmente, o nó móvel teria que notificar todas as comunicações parceiras sobre o novo endereço.

Outro método é a criação de roteadores específicos para os nós móveis. Os roteadores sempre escolhem o melhor prefixo para decisões de roteamento. Se um roteador tem no instante uma entrada para um prefixo 129.13.42 e um endereço 129.13.42.99, ele irá escolher a porta associada com o final para a transmissão, se um pacote com endereço de destino 129.13.42.99 entrar. Enquanto é teoricamente possível, mudar todas as tabelas de roteamento do mundo para criar rotas específicas para um nó móvel, isso seria impossível com o número de nós móveis existentes na internet. Os roteadores são construídos para transmissões extremamente velozes, mas não para suportar rápidas mudanças nas tabelas de roteamento. Enquanto o primeiro é feito com um suporte de *hardware* especial, o último é tipicamente



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

uma parte do *software*, o qual não pode manipular uma carga de freqüentes atualizações. Entretanto, roteadores são o “cérebro” da internet, são responsáveis por todas as ações da rede. Nenhum provedor de serviços ou administrador de sistemas, permitiria trocas nas tabelas de roteamento, provavelmente sacrificando a estabilidade, somente para a mobilidade de usuários individuais.

#### 1.2.2 – EXIGÊNCIAS

Desde que as soluções rápidas não tiveram muito êxito, uma arquitetura mais geral foi projetada. Muitos testes e sistemas proprietários finalmente levaram o IP móvel a um padrão para possibilitar a mobilidade na Internet. Várias exigências acompanham o desenvolvimento desse padrão:

- **Compatibilidade:** é enorme a quantidade de computadores conectados na internet através dos protocolos TCP/IP. Um novo padrão não pode requerer mudanças para aplicações ou protocolos de rede já em uso. As pessoas querem usar seu favorito *browser* para WWW e não mudanças nas aplicações apenas para mobilidade. O mesmo vale para sistemas operacionais. Não iriam usar outro sistema operacional somente para a mobilidade, assim, o IP móvel tem que estar integrado aos sistemas operacionais existentes ou ao menos trabalhar junto com eles. Os roteadores na internet deveriam não necessariamente requerer outro software. Enquanto é possível acentuar a capacidade de alguns roteadores para suportar a mobilidade, é quase impossível mudar todos os roteadores. Além disso, o IP móvel tem que permanecer compatível com todas as camadas mais baixas usadas para o padrão IP não móvel. Isso significa que o IP móvel não deve requerer meios ou protocolos MAC/LLC (*Medium Access Control/Logical Link Control*) especiais. Assim o IP móvel tem que usar algumas interfaces e mecanismos para acessar as camadas inferiores como o IP faz. Finalmente, sistemas fins enriquecidos com a implementação do IP móvel seriam capazes de realizar a comunicação com sistemas fixos sem o IP móvel. O IP móvel tem que garantir que os usuários possam ainda acessar todos os outros servidores e sistemas na Internet. Mas isto também sugere o uso do mesmo formato de endereços e mecanismos de roteamento.

- **Transparência:** A mobilidade deveria permanecer invisível para muitas aplicações e protocolos de camadas superiores. Além de trabalhar com larguras de bandas pequenas e algumas interrupções no serviço, camadas superiores continuariam trabalhando até mesmo se



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

o computador móvel mudasse seu ponto de ligação na rede. Para TCP, por exemplo, isso significa que o computador guardaria seu endereço IP, como explicado acima. Se a interrupção da conectividade não for muito longa, a conexão TCP sobrevive à mudança do ponto de ligação. Claramente, muitas aplicações de hoje não foram designadas para serem usadas em ambientes móveis. Entretanto, o único efeito da mobilidade seria um atraso maior e uma menor largura de banda. Contudo, há algumas aplicações para que isto seja melhor estando-se ciente da mobilidade. São exemplos, roteamentos baseados em custos ou compressão de vídeo. Sabendo-se que é atualmente possível usar diferentes redes, o *software* poderia escolher a mais barata. Ou se uma aplicação de vídeo sabe que no instante a menor largura de banda está disponível, ela usará um diferente esquema de compressão. Entretanto, os mecanismos adicionais são necessários para informar essas aplicações sobre mobilidade.

- **Eficiência:** Introduzir um novo mecanismo na internet não pode comprometer a eficiência da rede. Usar IP para mobilidade não deve gerar várias mensagens novas no grande fluxo de toda a rede. Entretanto, um cuidado especial tem que ser dado considerando pequenas larguras de banda em *links* sem fio. Muitos sistemas móveis terão um *link* sem fio para a ligação de pontos. No entanto, somente alguns pacotes adicionais seriam necessários entre um sistema móvel e um nó na rede. Vendo o número de computadores conectados na internet e a taxa de crescimento das comunicações móveis, está claro que milhares de aparelhos irão participar da Internet como componentes móveis. Exatamente planeja-se que carros, caminhões, telefones móveis, todo assento dentro de todo avião ao redor do mundo e etc – muitos deles terão alguma implementação IP e irão mover-se entre diferentes redes, assim requerendo IP móvel. Dessa forma é indispensável para um IP móvel ser escalonável sobre um grande número de participantes em toda a internet.

- **Segurança:** A mobilidade apresenta muitos problemas de segurança. Um requisito mínimo é a autenticação de todas as mensagens relacionadas ao gerenciamento do IP móvel. Isto deve garantir para a camada IP que se ela enviar um pacote a um *host* móvel este *host* móvel deve ser realmente o receptor do pacote. A camada IP pode apenas garantir que o endereço IP do receptor esteja correto. Não há forma de prevenir os endereços IP fraudados ou outros ataques. De acordo com a filosofia da Internet isto é deixado para camadas superiores.

Então, a finalidade de um IP móvel pode ser resumida como suportar a mobilidade para sistemas fins ao passo que mantém a estabilidade, a eficiência, e a compatibilidade em todos aspectos com os protocolos Internet e as aplicações existentes.



### 1.3 – ENTIDADES E TERMINOLOGIAS

A Figura 1.1 mostra um exemplo de uma rede de IP móvel, foram definidos várias entidades e termos necessários para o entendimento do IP móvel.

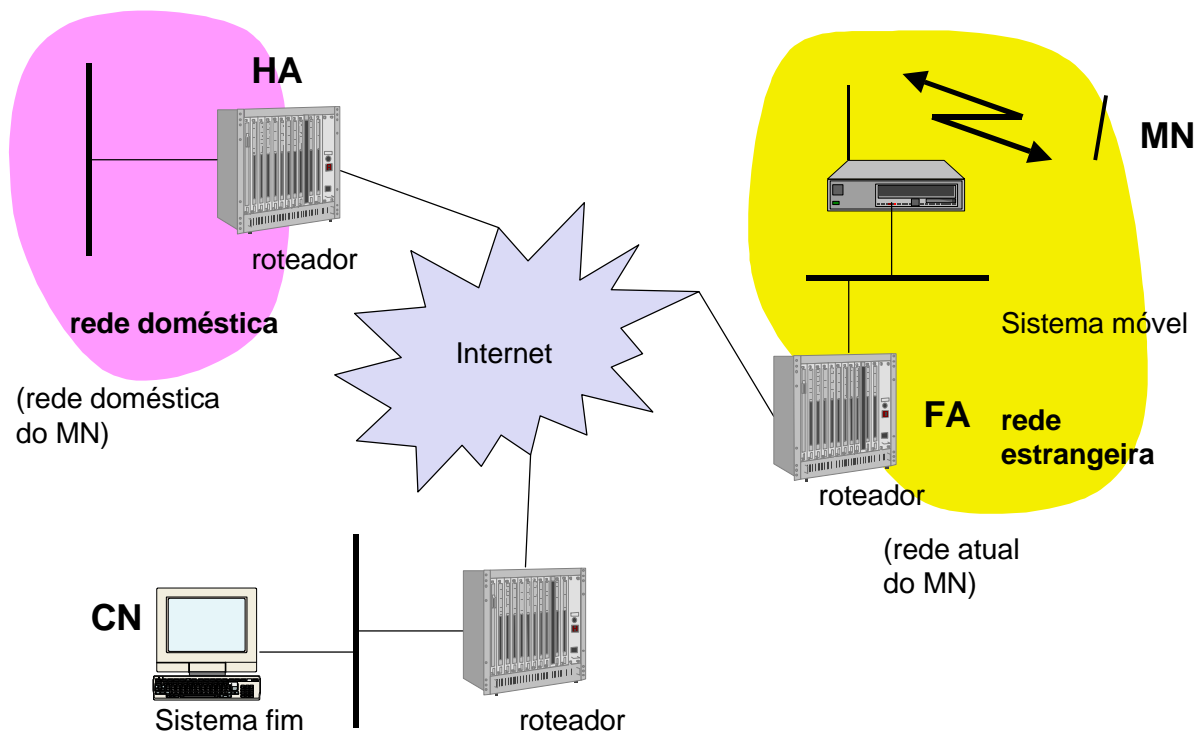


Figura 1.1 – Exemplo de uma rede IP móvel

- **Nó móvel (MN):** Um nó móvel é um sistema fim ou um roteador que pode mudar seu ponto de ligação para a Internet usando o IP móvel. O MN guarda seu endereço IP e pode continuamente comunicar-se com outros sistemas na internet enquanto a conectividade do *link* da camada existir.
- **Nó correspondente (CN):** No mínimo um parceiro é necessário para a comunicação. O CN representa esse parceiro para o MN. O CN pode ser um nó fixo ou móvel.
- **Rede Doméstica (Rede Local):** A rede doméstica é a sub-rede a qual o MN pertence com seu respectivo endereço IP. Dentro da rede doméstica não é necessário o suporte para IP móvel.





## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

- Rede estrangeira: A rede estrangeira é a sub-rede a qual o MN visita e da qual a rede doméstica não faz parte.
- Agente estrangeiro (FA): O FA pode proporcionar vários serviços para o MN durante sua visita a uma rede estrangeira. O FA pode ter o COA (definido abaixo), assim atuando como a porta de saída de túneis e enviando pacotes para o MN. Além disso, o FA pode ser o roteador padrão para o MN. Os FAs também podem proporcionar serviços seguros pertencentes às redes estrangeiras em oposição ao MN apenas visitante. Para o funcionamento do IP móvel, os FAs não são totalmente necessários. Tipicamente um FA está implementado no roteador da sub-rede a qual o MN está ligado.
- *Care-of address* (COA): O COA define a localização atual do MN do ponto de vista do IP. Todos os pacotes IPs enviados ao MN são entregues ao COA e não diretamente ao endereço IP do MN. A entrega de pacotes para o MN é feita usando um túnel, como explicado anteriormente. Portanto, para ser mais preciso, o COA indica a porta de saída do túnel, isto é, o endereço onde os pacotes saem do túnel. Há duas diferentes possibilidades para a localização do COA:
  - Agente COA estrangeiro: O COA poderia estar localizado no FA, isto é, o COA é um endereço IP do FA. Assim o FA é a porta de saída do túnel e envia pacotes para o MN. Vários MNs usando o FA podem dividir o COA como um COA comum.
  - COA co-aloado: O COA é chamado de co-aloado se o MN adquirir temporariamente um endereço IP adicional na qual atua como COA. Esse endereço está agora topologicamente correto, e a porta de saída do túnel está no MN. São endereços co-aloados.
- Agente Doméstico (HA): O HA provê vários serviços para o MN e está localizado na rede doméstica. O túnel para os pacotes trafegarem para o MN inicia no HA. Além disso o HA mantém um registro de localização, isto é, ele é informado da localização do MN pelo COA atual.

Existem três alternativas para implementação do HA:

- O HA pode ser implementado em um roteador que é responsável pela rede doméstica. Esta é obviamente a melhor posição, porque sem otimizações no IP móvel, todos os pacotes para o MN têm que passar de qualquer maneira pelo roteador.
- Se não for possível mudar o *software* do roteador, o HA seria também implementado em um nó arbitrário na sub-rede. Uma desvantagem desta solução é um cruzamento dobrado





## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

do roteador pelo pacote se o MN estiver em uma rede estrangeira. Um pacote para o MN vem através do roteador; o HA envia-o através do túnel que novamente cruza o roteador.

- Finalmente, uma rede doméstica não é necessária. O HA estaria novamente no roteador mas dessa vez atuando apenas como gerente para os MNs pertencentes a uma rede doméstica virtual. Com esta solução todos os MNs estão sempre em uma rede estrangeira.

A rede na Figura 1.2 mostra a seguinte situação: Um CN está conectado através de um roteador na Internet, assim como estão a rede doméstica e a rede estrangeira. O HA é implementado no roteador conectando a rede doméstica com a Internet, e um FA é implementado no roteador para a rede estrangeira. O MN está atualmente na rede estrangeira. O túnel para os pacotes enviados à MN inicia no HA e termina no FA, pois nesse exemplo o FA tem o seu COA.

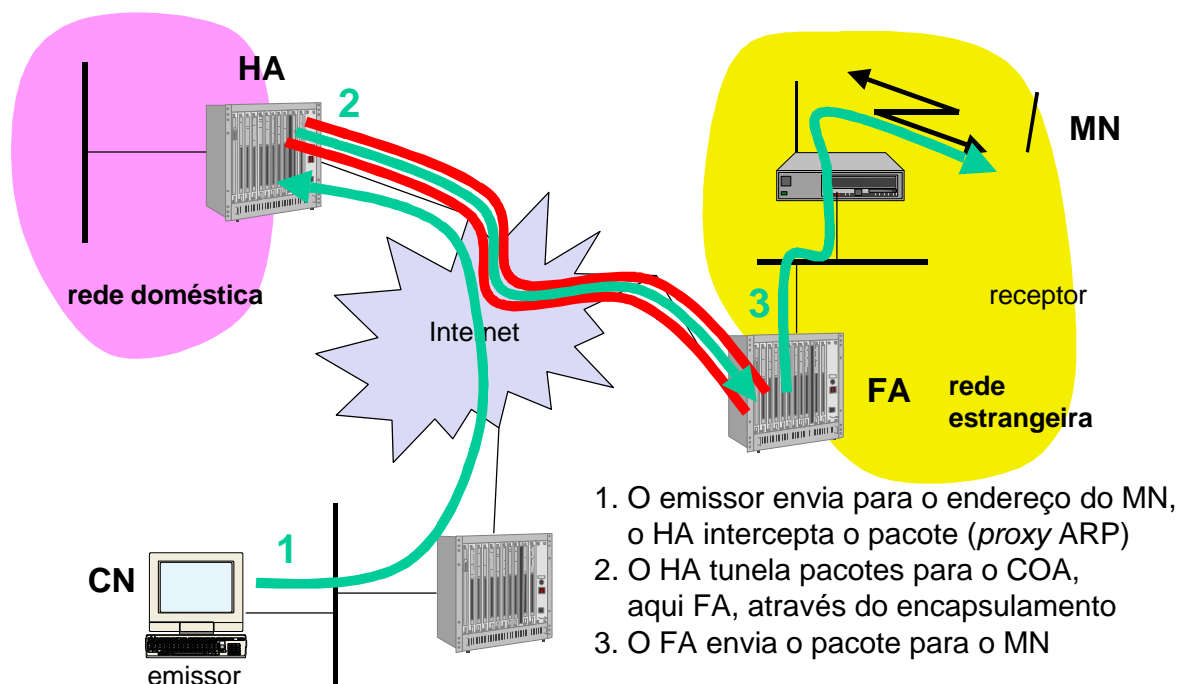


Figura 1.2 – Transferência de dados para um sistema móvel

#### 1.4 - ENTREGA DE PACOTE IP

As figuras 1.2 e 1.3 mostram o tráfego de pacotes para o MN e do MN, respectivamente. Um nó correspondente CN quer enviar um pacote IP para o MN. Uma das



**Universidade Federal do Pará**  
**Departamento de Engenharia Elétrica e de Computação**

exigências do IP móvel é suportar transparência da mobilidade do MN. Desta forma, o CN não precisa saber nada sobre a localização atual do MN e envia o pacote como sempre para o endereço IP do MN. Isto significa que o CN envia um pacote IP com o MN como endereço de destino e o CN como endereço de origem. A Internet, não tendo informações da localização atual do MN, roteia o pacote para o roteador responsável pela rede doméstica do MN. Isto é feito usando mecanismos de roteamento padrão da Internet.

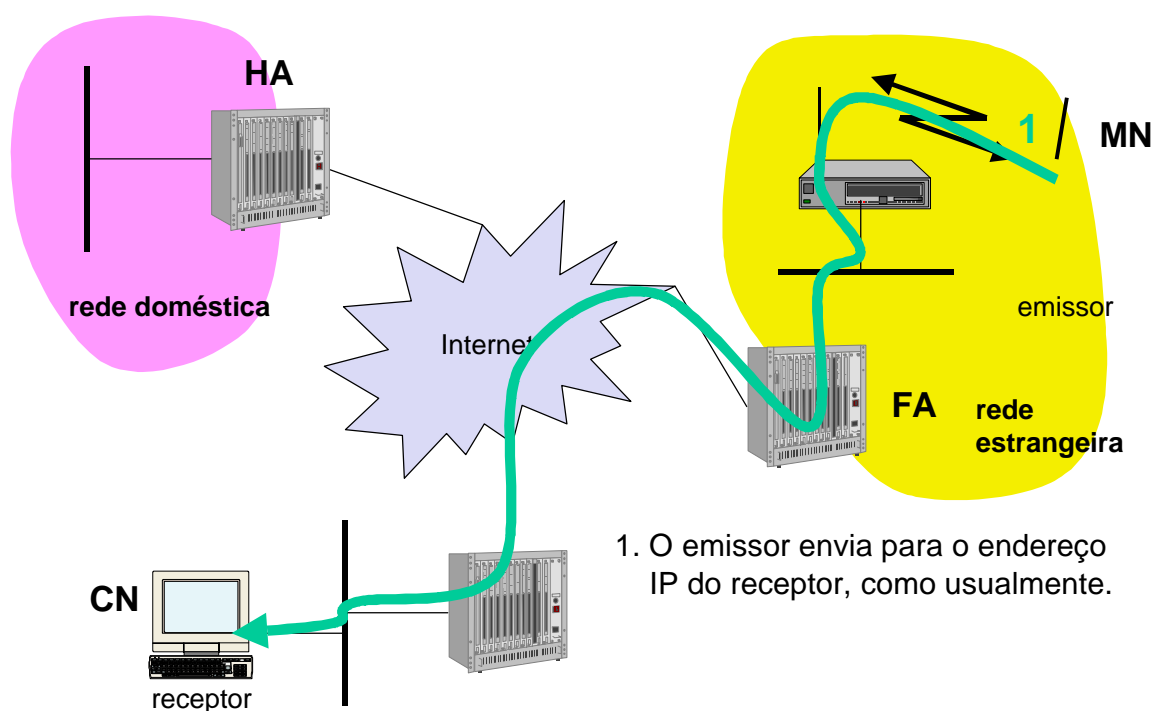


Figura 1.3 – Transferência de dados do sistema móvel

O HA intercepta o pacote, sabendo que o MN não está atualmente em sua rede doméstica. Assim, o pacote não é enviado como sempre para a sub-rede, mas encapsulado e tunelado para o COA. Isto é feito pela colocação de um cabeçalho na frente do antigo cabeçalho IP mostrando o COA como novo destino e o HA como origem do pacote encapsulado. O agente estrangeiro agora desencapsula o pacote, isto é, remove o cabeçalho adicional, e envia o pacote original com o CN como origem e o MN como destino para o MN.



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

Novamente, para o MN a mobilidade não é visível. O Nó Móvel recebe o pacote com os mesmos endereço de emissor e receptor como teria feito na rede doméstica.

O MN envia o pacote como sempre com seu próprio endereço IP como origem e o endereço CN como destino. O roteador com o FA atua como um roteador padrão e envia o pacote da mesma maneira como faria para qualquer outro nó na rede estrangeira. Se o CN também fosse um nó móvel residindo em uma rede estrangeira, os mesmos mecanismos como descrito acima seriam aplicados, só que agora na outra direção.

São necessários alguns mecanismos adicionais para o IP móvel trabalhar, algumas otimizações para o protocolo e alguns problemas de segurança e eficiência serão abordados a seguir.

#### 1.4.1 - DESCOBERTA E ANÚNCIO DO AGENTE

Um problema inicial de um MN depois de mover-se é como encontrar um agente estrangeiro. A movimentação do MN é detectada pelos agentes estrangeiros e agentes domésticos que anunciam sua presença periodicamente, usando mensagens de anúncio de agente especiais. Estas mensagens de anúncio podem ser vistas como um *broadcast* em uma sub-rede. Para estes anúncios, as mensagens do ICMP (*Internet Control Messange Protocol*) são usadas com algumas extensões à mobilidade. Roteadores na rede fixa também implementam este padrão anunciando seus serviços de roteamento periodicamente às ligações fixas.

Um anúncio de agente efetua as seguintes funções:

- Permite a detecção de agentes móveis;
- Lista um ou mais *care-of-addresses* disponíveis;
- Informa o nó móvel sobre capacidades especiais disponibilizadas por agentes estrangeiros, por exemplo, técnicas alternativas de encapsulamento;
- Permite aos nós móveis determinar o endereço de rede e estado das suas ligações à Internet;
- Permite aos nós móveis saber se o agente é um agente doméstico ou um agente estrangeiro e se portanto está na sua rede doméstica ou numa exterior, respectivamente.



**Universidade Federal do Pará**  
**Departamento de Engenharia Elétrica e de Computação**

O pacote de anúncio de agente de acordo com o RFC 1256 (*Request for Comments* – relatórios técnicos sobre IP/TCP) com a extensão para mobilidade é mostrado na figura 1.4. A parte superior representa o pacote ICMP enquanto que a parte inferior é a extensão necessária para mobilidade. Claramente, os nós móveis devem ser alcançados com o apropriado endereço de camada de ligação. Além disso, o campo TTL do pacote IP é configurado para 1 para todos os anúncios evitando o renvio de anúncios. O endereço de destino IP de acordo com anúncios do roteador padrão pode ser configurado para o endereço *multicast* 224.0.0.1 ou para o endereço *broadcast* 255.255.255.255.

Os campos na parte do ICMP são definidos da seguinte maneira: o campo *type* é configurado com 9, o campo *code* pode ser 0, se o agente também roteia tráfego de nós não móveis, ou 16, se não roteia nada além de tráfego móvel. O número de endereços anunciados com este pacote está no campo *#addresses* enquanto os reais endereços estão como mostrado na figura 1.4. O campo *Lifetime* denota o tempo em que este anúncio é válido. Os níveis de preferências para cada endereço, ajuda um nó escolher o roteador mais propenso para adquirir um nó novo.

0	7	8	15	16	23	24	31				
type		code		checksum							
#addresses		addr. size		lifetime							
router address 1											
preference level 1											
router address 2											
preference level 2											
...											
type		length		sequence number							
registration lifetime				R	B	H	F	M	G	V	reserved
COA 1											
COA 2											
...											

Figura 1.4 – Pacote de anúncio de agente

As diferenças comparadas com anúncios do padrão ICMP são vistas após os endereços de roteadores. Esta extensão para mobilidade tem os seguintes campos definidos: *type* é



**Universidade Federal do Pará**  
**Departamento de Engenharia Elétrica e de Computação**

configurado com 16, *length* depende do número de COAs providos com a mensagem. Um agente mostra o número total de anúncios enviados desde a inicialização no campo *sequence number*. Através do registrador *lifetime* o agente pode especificar o tempo de vida máximo, em segundos, que um nó pode requisitar durante o registro. Os bits seguintes especificam as características de um agente em detalhes. O bit R (registro) mostra se um registro com esse agente é requerido ao invés de usar um COA co-alocado no MN. Se o agente está muito ocupado para aceitar novos registros ele pode configurar o bit B. Os dois bits seguintes denotam se o agente oferece serviços como um agente doméstico (H) ou o agente estrangeiro (F) na ligação onde o anúncio foi enviado. Os bits M e G especificam o método de encapsulamento usado para o túnel. Enquanto, o encapsulamento IP-em-IP é o padrão obrigatório, M pode especificar encapsulamento mínimo e G encapsulamento de roteamento genérico. O bit V finalmente pode especificar o uso de compressão de cabeçalho. Os campos seguintes contêm os COAs anunciados. Um agente estrangeiro configura o bit F para anunciar um COA.

Um nó móvel em uma sub-rede pode receber anúncios de agentes, do seu agente doméstico ou de um agente estrangeiro. Isto é um modo para o MN descobrir sua localização. Se nenhum anúncio de agente está presente e um MN não recebeu um COA através de outros meios, por exemplo, o DHCP (*Dynamic Host Configuration Protocol*), o nó móvel deve enviar solicitações ao agente. Certos cuidados devem ser tomadas para que estas mensagens de solicitações não sobrecarreguem a rede, mas basicamente, um MN pode procurar um FA enviando mensagens de solicitação eternamente. Descobrir um agente novo pode ser feito a qualquer momento. Considera-se o caso em que um MN está procurando uma melhor conexão enquanto ainda envia pacotes pelo antigo caminho. Este é o caso em que se move por várias células de diferentes redes sem fio.

Após esses passos de anúncios e descoberta de agente o MN pode receber um CÔA de um FA ou um COA co-alocado. Além disso, o MN sabe sua localização (rede doméstica ou rede estrangeira) e a capacidade do agente. O próximo passo para o MN é o registro com o HA se o MN estiver em uma rede estrangeira.

Usar um padrão como RFC 1256 para algo diferente que o propósito original de anúncios de roteador causa alguns problemas. Um problema bastante óbvio é o intervalo mínimo de três segundos entre dois anúncios. Isto toma um sentido em redes fixas porque a



topologia muda lentamente (leva algum tempo para substituir roteadores inoperantes, etc). Porém, em redes sem fio altamente dinâmicas com MNs se movendo e provavelmente com aplicações que requerem contínuo fluxo de pacotes, três segundos é muito tempo. Um MN teria sempre que esperar por no mínimo três segundos antes de notar que um agente não é mais alcançável. Mas talvez este anúncio já esteja perdido. Assim, para ter certeza da necessidade de um novo agente, um MN tem que esperar muito tempo. A emissão de solicitações não é a solução real para sobrecarga desnecessária na sub-rede.

#### 1.4.2 - REGISTRO

Depois de ter recebido um COA, o MN tem que registrar com o HA. O propósito principal do registro é informar o HA da localização para o envio correto de pacotes. O registro pode ser feito de dois modos diferentes, que dependem da localização do COA.

- Se o COA está no FA, o registro é feito como mostrado na figura 1.5. O MN envia seu pedido de registro contendo o COA, para o FA o qual envia o pedido para o HA. O HA agora configura uma ligação com a mobilidade contendo o endereço IP doméstico do nó móvel e o atual COA. Adicionalmente, a ligação com a mobilidade contém o tempo de vida do registro o qual é negociado durante o processo de registro. O registro expira automaticamente após o tempo de vida e é deletado; dessa forma, o MN deve se re-inscrever antes do registro expirar. Este mecanismo é necessário para evitar ligações com mobilidade que não são mais usadas. Após configurar uma ligação com mobilidade, o HA envia de volta uma mensagem de resposta para o FA que envia para o MN.

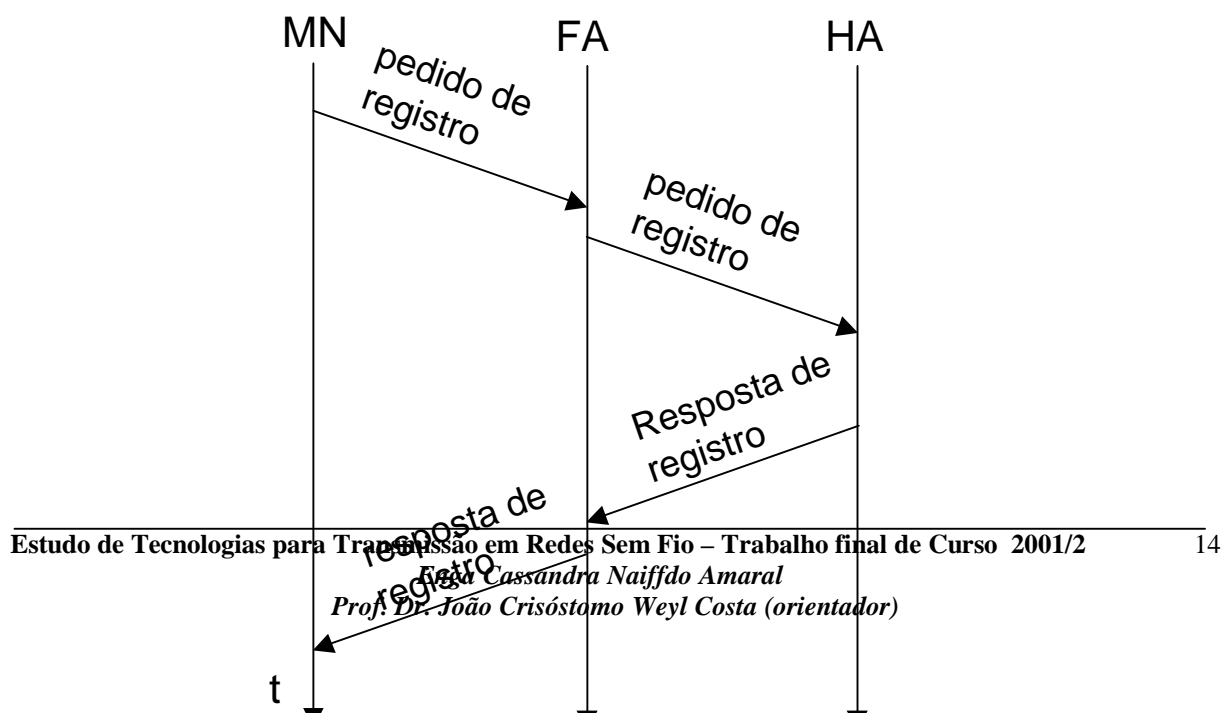


Figura 1.5 – Registro de um nó móvel via FA

- Se o COA está co-allocado, o registro é mais simples, como mostrado na figura 1.6. O MN envia um pedido diretamente ao HA e vice-versa. A propósito, isso também é o procedimento de registro para MNs retornando para sua rede doméstica. Aqui eles também se registram diretamente com o HA.

Para pedido de registro, pacotes UDP (*User Datagram Protocol*) são usados. O endereço IP fonte é configurado para o endereço de interface do MN, o endereço IP destino é do FA ou HA (dependendo da localização do COA). A porta UDP destino é configurada para 434. O UDP é usado por razão de baixo *overhead* e melhor performance comparado ao TCP em ambientes sem fio.

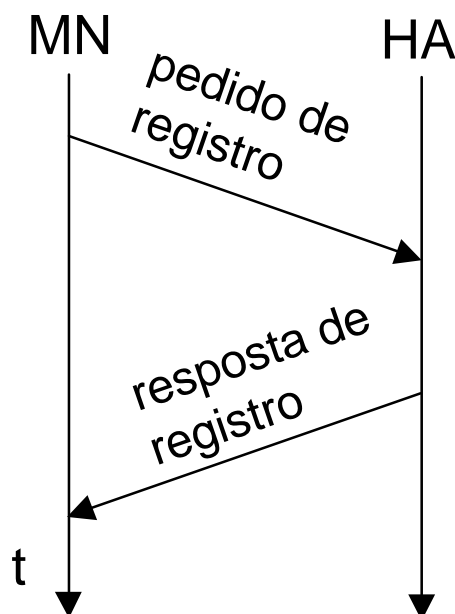




Figura 1.6 – Registro direto de um MN com o FA

### 1.4.3 - TUNELAMENTO E ENCAPSULAMENTO

A figura 1.2 mostrou os mecanismos usados para enviar pacotes entre o HA e o COA. Um túnel estabelece um tubo virtual para pacotes de dados entre a entrada e a saída do túnel. Pacotes são enviados através do túnel sem alterá-lo. O envio de pacotes através de um túnel é realizado usando o encapsulamento.

O encapsulamento é o mecanismo de manipular um pacote que consiste de cabeçalho e dados, e colocá-lo na parte de dados de um novo pacote. A operação reversa, retira um pacote da parte de dados de outro pacote, isto é chamado de desencapsulamento. O encapsulamento e o desencapsulamento são operações tipicamente executadas quando um pacote é transferido de uma camada de protocolo mais alta para uma camada mais baixa ou vice-versa, respectivamente.

Este mecanismo é mostrado na figura 1.7 e descreve exatamente o que o HA faz na entrada do túnel. O HA leva o pacote original com o MN como destino, coloca-o na parte de dados de um pacote novo e configura-o no cabeçalho IP de tal modo que o pacote é roteado ao COA. O novo cabeçalho é também chamado de cabeçalho externo. Adicionalmente, há um cabeçalho interno que pode ser idêntico ao cabeçalho original como no caso do encapsulamento IP-em-IP, ou o cabeçalho interno pode ser computado durante o encapsulamento.

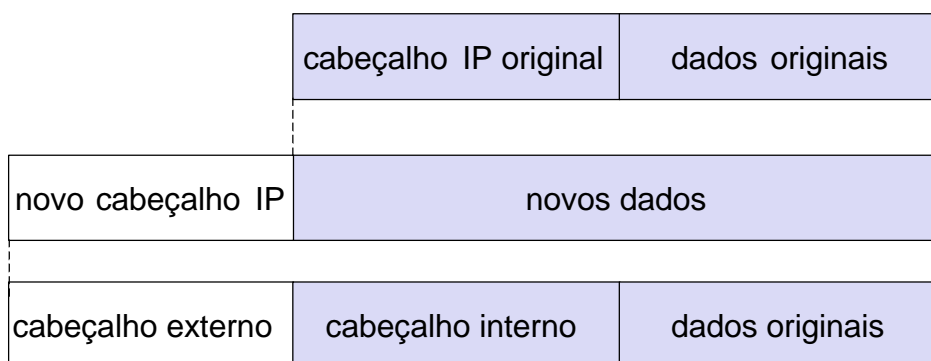


Figura 1.7 – Encapsulamento IP

### 1.4.4 - ENCAPSULAMENTO IP-EM-IP



Há diferentes formas de executar um encapsulamento necessário para o túnel entre o HA e o COA. O encapsulamento IP-em-IP é obrigatório para o IP móvel. Os campos deste novo pacote seguem a especificação padrão do protocolo IP como definido em RFC 791.

Ao usar IP-em-IP o agente doméstico, no início do túnel, insere um novo cabeçalho IP à frente do cabeçalho IP de qualquer datagrama endereçado ao endereço de origem do MN. O novo cabeçalho do túnel usa o COA do MN como endereço IP destino, ou destino do túnel. O endereço IP do início do túnel é o agente doméstico e o cabeçalho do túnel usa o número do nível mais alto do protocolo (número 4), indicando que o próximo cabeçalho é novamente um cabeçalho IP. No IP-em-IP o cabeçalho original IP é preservado como a primeira parte dos dados a ser lida pelo túnel. Portanto, para recuperar o pacote original o agente estrangeiro apenas tem de eliminar o cabeçalho do túnel e entregar o resto ao nó móvel.

#### 1.4.5 - ENCAPSULAMENTO MÍNIMO

A forma de encapsular datagramas consiste em pôr o datagrama original (= cabeçalho IP + dados) dentro de outro envelope IP, obtendo assim o pacote a enviar (= cabeçalho IP externo + COA + datagrama original). Os campos do cabeçalho IP externo adicionam muito *overhead* ao datagrama final – vários campos são duplicados do cabeçalho IP interior. Este desperdício de espaço não é desejável.

A solução para este desperdício seria o mecanismo chamado encapsulamento mínimo, sendo este uma nova opção para encapsular o datagrama. A abordagem do método de encapsulamento é a seguinte:

Em vez de inserir um novo cabeçalho, o cabeçalho original é modificado para refletir o COA e é introduzido entre o cabeçalho IP modificado e os dados não modificados um cabeçalho mínimo de *forwarding* para guardar o endereço fonte original e o endereço destino original. Quando o agente estrangeiro tentar desencapsular o pacote, ele simplesmente restaurará os campos no cabeçalho de *forwarding* do cabeçalho IP e retirará o cabeçalho de *forwarding*.

Existe uma restrição ao uso do método de encapsulamento mínimo. Se o datagrama original já estiver fragmentado o encapsulamento mínimo não pode ser usado uma vez que não sobra espaço para guardar informação fragmentada.



#### 1.4.6 - ENCAPSULAMENTO DE ROTEAMENTO GENÉRICO

Enquanto o encapsulamento IP-em-IP e o encapsulamento mínimo trabalham apenas para IP, o esquema de encapsulamento de roteamento genérico (*Generic Routing Encapsulation* - GRE) suporta outros protocolos de camada de rede além do IP. O GRE permite o encapsulamento de pacotes de um determinado protocolo na área de dados de outro protocolo. A Figura 1.8 mostra este procedimento. O pacote de um protocolo com o seu cabeçalho e seus dados originais são encapsulados a um cabeçalho de GRE novo. Juntos eles formam uma nova área de dados de um novo pacote, posteriormente um cabeçalho externo é inserido

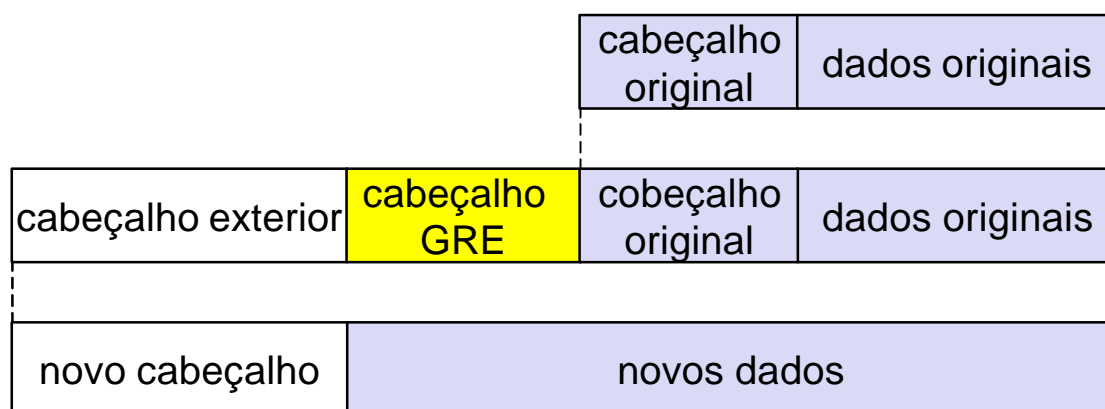


Figura 1.8 – Encapsulamento de Roteamento Genérico

Os números de seqüência devem ser usados por um desencapsulador para restabelecer a ordem do pacote. Isto pode ser importante, se um protocolo que garante transmissão ordenada é encapsulado e transferido usando um protocolo o qual não garante entrega ordenada como o IP. Agora o desencapsulador na saída do túnel deve restabelecer a seqüência para manter a característica do protocolo.

#### 1.5 - OTIMIZAÇÕES



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

Com o IP móvel básico todos os pacotes para o MN vão através do HA. Isto pode causar *overhead* desnecessário para a rede entre o CN e o HA, mas também entre o HA e o COA, dependendo da localização atual do MN. Além disso, a latência pode aumentar drasticamente. Isto está particularmente insatisfatório se MNs e HAs são separados, por exemplo por ligações transatlânticas.

Uma forma de otimizar a rota é informar ao CN da atual localização do MN. O CN pode aprender a localização através de *caching* em um *cache* de ligação que é uma parte da tabela de roteamento local para o CN. A entidade apropriada para informar o CN da localização é o HA. O protocolo IP móvel otimizado precisa de quatro mensagens adicionais:

- Pedido de ligação: Qualquer nó que queira saber a localização atual de um MN pode enviar um pedido de ligação para o HA. O HA pode checar se o MN permitiu disseminação da sua localização atual e, caso ao HA seja permitido revelar a localização, ele envia de volta uma atualização da ligação.
- Atualização de ligação: Esta mensagem informa sobre a localização atual de um MN. A mensagem contém o endereço IP fixo do MN e do COA. A atualização da ligação pode pedir uma confirmação.
- Confirmação da ligação: Se requisitado, um nó retorna esta confirmação após receber uma mensagem de atualização de ligação.
- Avisos da ligação: Finalmente, se um nó desencapsula um pacote para um MN, mas ele não é o atual FA para este MN, este nó envia um aviso de ligação para o HA do MN. O aviso contém o endereço IP do MN e o endereço do nó que tentou enviar o pacote para este MN. O HA deve enviar uma atualização de ligação ao nó que obviamente tem um COA errado para o MN.

A Figura 1.9 explica estas quatro mensagens adicionais juntas com o caso de um MN mudando seu FA. Em primeiro lugar, o CN pode pedir a atual localização do MN ao HA. Se permitido pelo MN, o HA retorna o COA do MN através de uma mensagem de atualização. O CN confirma esta mensagem de atualização e guarda a ligação de mobilidade. Agora o CN pode enviar seus dados diretamente ao atual agente estrangeiro FA<sub>old</sub>. O FA<sub>old</sub> envia os pacotes ao MN. Este cenário mostra um COA localizado no FA. O encapsulamento de dados para o tunelamento do COA é agora feito pelo CN, não mais pelo HA.

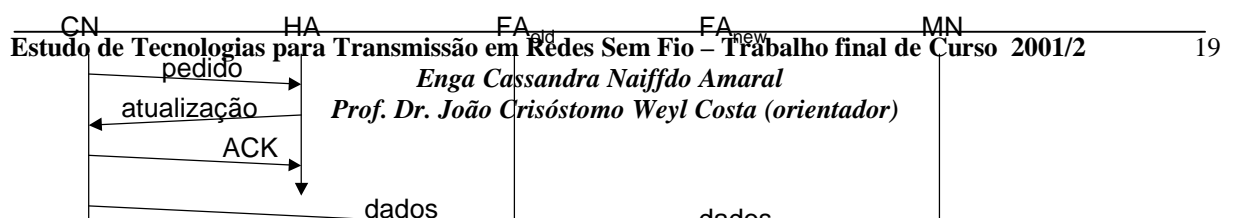




Figura 1.9 – Mudança de um agente estrangeiro com o IP Móvel otimizado

O MN deve mudar sua localização e se registrar com um novo agente estrangeiro, o  $FA_{new}$ . Este registro também é enviado para o HA atualizar seu banco de dados de localização. Além disso, o  $FA_{new}$  informa ao  $FA_{old}$  sobre o novo registro do MN. A mensagem de registro do MN contém o endereço do  $FA_{old}$  para este propósito. A passagem desta informação é realizada por uma mensagem de atualização, o qual é confirmada pelo  $FA_{old}$ . A resposta de registro não é mostrada neste cenário. Sem a informação gerada pelo novo FA, o FA antigo não conseguiria saber nada sobre a nova localização do MN. Neste caso, seguramente o CN não sabe nada sobre a nova localização. Assim, ele ainda envia seus pacotes para o MN através de um túnel para o FA antigo, o  $FA_{old}$ . Este FA agora notifica pacotes com destino ao MN, mas também sabe que não é mais o atual FA do MN. O  $FA_{old}$  deve agora enviar estes pacotes para o novo COA do MN o qual neste exemplo é o  $FA_{new}$ . Esse envio de pacotes é outra otimização do IP Móvel básico que provê suaves *handovers*. Sem esta otimização, todos os pacotes em trânsito seriam perdidos enquanto o MN move-se de um FA para outro. Especialmente com o TCP como protocolo da camada superior isto resultaria em sérias degradações de performance.

Para dizer ao CN que ele tem um *caching* de ligação desatualizado, o  $FA_{old}$  envia uma mensagem de aviso ao HA. O HA agora envia uma atualização ao CN informando-o sobre a



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

nova localização. O CN confirma esta atualização. Agora o CN pode enviar seus pacotes diretamente para o FA<sub>new</sub>, assim evitando o roteamento triangular (conduta ineficiente de um IP móvel não aperfeiçoado).

Infelizmente, esta otimização do IP móvel para evitar o roteamento triangular causa vários problemas de segurança.

#### 1.6 - TUNELAMENTO REVERSO

O MN pode diretamente enviar seus pacotes ao CN como em qualquer outra situação IP padrão. O endereço de destino dos pacotes é o do CN. Mas há vários e sérios problemas associados com esta solução.

- *Firewalls*: Quase todas as companhias e muitas outras instituições dão segurança as suas redes internas conectadas à Internet com ajuda de um *firewall*.

Além de muitas outras funções, os *firewalls* podem ser configurados para filtrar endereços não confiáveis do ponto de vista do administrador. Frequentemente, os *firewalls* permitem apenas que pacotes com endereços topologicamente corretos passem. Porém, o MN ainda envia pacotes com o seu endereço IP fixo como fonte o qual não é o topologicamente correto em uma rede estrangeira. Além disso, os *firewalls* geralmente filtram os pacotes que vêm de fora contendo o endereço fonte dos computadores de uma rede interna. Isto é feito para evitar que outros computadores possam usar endereços internos e possam reivindicar serem computadores internos. Porém, isto também insinua que um MN não pode enviar um pacote a um computador que reside em sua rede doméstica. Isto não significa apenas que os endereços de destino enviam pacotes IP, mas também o endereço de fonte devido a preocupações de segurança.

- *Multicast*: Túneis reversos são necessários para o MN participar em um grupo de *multicast*. Enquanto os nós em uma rede doméstica devem participar em um grupo *multicast*, um MN em uma rede estrangeira não pode transmitir pacotes de *multicast* na qual eles emanam de sua rede doméstica sem um túnel reverso.

- **TTL**: Considera-se um MN ainda em sua rede doméstica, enviando pacotes com um certo TTL. O TTL deve ser pequeno o suficiente de forma que nenhum pacote seja transmitido fora de uma certa região. Se o MN agora se mover para uma rede estrangeira, este TTL deve ser também pequeno para que os pacotes encontrem os mesmos nós como antes. O



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

IP móvel não é mais transparente se um usuário tiver que ajustar o TTL enquanto se move. Então, um túnel é necessário representando apenas um *salto* não importando quantos *saltos* são realmente necessários do agente estrangeiro à rede doméstica.

Todas estas considerações conduziram a RFC 2344 definir tunelamento reverso como uma extensão para o IP móvel. Esta RFC é compatível ao IP móvel e define o tunelamento reverso topologicamente correto como necessário para controlar o problema descrito acima. Obviamente, o tunelamento reverso cria o problema de roteamento triangular, agora na direção reversa. Todos os pacotes de um MN para um CN passam pelo HA. O RFC 2344 não oferece uma solução para este roteamento triangular reverso, porque não está claro se o CN pode desencapsular pacotes. Lembrando que o IP móvel deve trabalhar junto com todos os nós IPs tradicionais. Desta forma, não se pode assumir que um CN seja capaz de ser uma porta de saída do túnel.

O tunelamento reverso adicionalmente aumenta várias características de segurança que não haviam sido resolvidas até o momento. Por exemplo, túneis iniciando em uma rede privada de uma companhia e alcançando a Internet poderiam ser capturados e usados para enviar pacotes através do *firewall*. Assim não está claro se companhias devem permitir configurar túneis através de *firewall* sem futuras verificações de pacotes. É mais provável que uma companhia configure uma rede virtual especial para nós móveis visitantes fora do *firewall* com completa conectividade para a Internet. Isto permite que convidados usem seus equipamentos móveis e ao mesmo tempo que os padrões de segurança atuais sejam mantidos.

#### 1.7 - IPv6

Vários mecanismos que tiveram de ser separadamente especificados para suporte de mobilidade se vêm livres em IPv6. Um assunto é segurança que é agora uma característica exigida para todos os nós IPv6. Nenhum mecanismo especial é necessário para dar segurança em registros IP móvel. Todo nó IPv6 apresenta autoconfiguração de endereço, assim os mecanismos para adquirir um COA já são construídos no IPv6. A descoberta de vizinhança como mecanismo obrigatório para todo nó também é incluída na especificação, desta forma, os agentes estrangeiros especiais não são mais necessários para esses serviços. Combinar as características de auto-configuração e descoberta de vizinhança significa que todo nó móvel é capaz de criar ou obter um endereço topologicamente correto para o ponto atual de ligação.





## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

Além disso, vários nós de IPv6 podem enviar atualizações de ligações para outros nós, assim o MN pode enviar seu atual COA para o CN e para o HA. Estes mecanismos são uma parte integrante do IPv6. Além do que, um *soft handover* é possível com o IPv6. O MN envia seu novo COA para o antigo roteador servindo o MN no antigo COA, e o antigo roteador encapsula todos os pacotes para o MN e os envia para o novo COA.

O IP móvel em redes IPv6 requer poucos mecanismos adicionais de um CN, MN e HA. O FA não é mais necessário. Um CN apenas tem de ser capaz de fazer atualizações de ligações, isto é, criar ou atualizar uma entrada no *caching* de roteamento. O próprio MN tem que ser capaz de desencapsular pacotes, detectar quando necessário, um novo COA e determinar quando enviar atualizações de ligação para o HA e para o CN. O HA deve encapsular pacotes.

#### 1.8 - PROTOCOLO DE CONFIGURAÇÃO DE *HOST* DINÂMICO

O protocolo de configuração de *host* dinâmico é principalmente usado para a simplificação da instalação e manutenção de computadores em rede. Se um novo computador é conectado a uma rede, o DHCP pode provê-lo com todas as informações necessárias para a sua completa integração do sistema à rede, por exemplo, endereços de um servidor DNS e o roteador padrão o mascaram de sub-rede, o nome de domínio, e um endereço IP. Especialmente a última capacidade provê o endereço IP, tornando o DHCP muito atraente para IP móvel como uma fonte de *care-of adress*. Enquanto os mecanismos básicos do DHCP são bastante simples, muitas opções são disponíveis como descrito no RFC 2132.

O DHCP é baseado em um modelo de cliente/servidor como mostrado na Figura 1.10. Os clientes DHCP enviam um pedido a um servidor (DHCPDISCOVER no exemplo) para o qual o servidor responde. Um cliente envia pedidos usando o *broadcast* MAC. Um revezamento de DHCP pode ser necessário para enviar pedidos ao servidor DHCP.

O DHCP é um bom candidato para suportar as aquisições do *care-of adress* para nós móveis. O mesmo guardará todos os outros parâmetros necessários, como um endereço do roteador padrão, servidores DNS, e etc. Assim, um servidor DHCP estaria localizado em uma sub-rede do ponto de acesso do nó móvel, ou pelo menos um *relay* DHCP proveria o envio das mensagens.

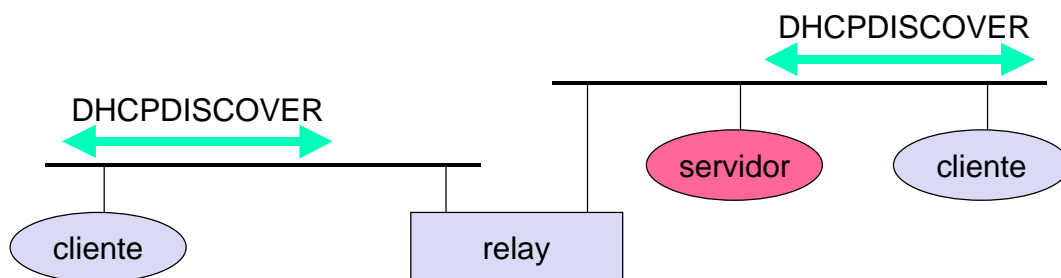


Figura 1.10 – Configuração do DHCP

Infelizmente, há vários problemas relacionados ao uso do DHCP. Um assunto principal é segurança. Não tem havido nenhuma autenticação de mensagens DHCP especificadas. Isto significa que o nó móvel não pode confiar em um servidor DHCP, e o servidor DHCP não pode confiar no nó móvel. Além disso, não há nenhum protocolo para configuração de servidor-servidor, isto é, um servidor DHCP não pode se comunicar com outro servidor DHCP e trocar as configurações atualmente usadas. Assim, configurações no servidor devem ser implementadas manualmente. Por exemplo, espaços de endereços que servidores de DHCP podem usar para clientes têm que ser deslocados. Um administrador tem que ter cuidado para que todo servidor DHCP tenha seu próprio espaço de endereço para clientes. Isto tipicamente resulta em fragmentação espacial de endereço.

### 1.9 - REDES *AD HOC*

Suporte à mobilidade confia na existência de pelo menos alguma infra-estrutura. O IP móvel requer, por exemplo, um agente doméstico, túneis, e roteadores padrão. O DHCP requer servidores e capacidades de *broadcast* no meio, alcançando todos os participantes do *relay* aos servidores. Redes de telefone celulares requerem estações bases, estrutura de redes, etc. Porém, há várias situações onde os usuários de uma rede não podem depender de uma infra-estrutura, a infra-estrutura é muito cara, ou não há nenhuma infra-estrutura. As redes *ad hoc* nestas situações são a única escolha.

Exemplos para o uso de redes *ad hoc* são:

- Infra-estrutura imediata: Encontros são planejados, comunicações interpessoais espontâneas e etc, não podem depender de nenhuma infra-estrutura. Infra-estruturas precisam



de planejamento e administração. Levaria muito tempo para configurar este tipo de infraestrutura, então, a conectividade *ad hoc* deve se usada.

- Áreas de desastre: Infra-estruturas tipicamente quebram em áreas de desastre. Furacões cortam linhas de energia e telefônicas, inundações destroem estações básicas, fogos queimam servidores. Assim, grupos de emergências só podem confiar em uma infra-estrutura que eles podem montar. Nenhum planejamento pode ser feito, e a configuração deve ser feita de forma extremamente rápida e confiável. O mesmo se aplica a muitas atividades militares.

- Áreas distantes: Até mesmo se pudessem ser planejadas infra-estruturas, às vezes é muito caro para montar uma infra-estrutura em áreas de pouco povoadas. Dependendo do padrão de comunicação, redes *ad hoc* ou infra-estrutura de satélite podem ser uma solução.

- Efetividade: Serviços de infra-estruturas existentes poderiam ser muito caros para certas aplicações. Se por exemplo, uma rede de celulares orientada a conexão existe, mas uma aplicação envia apenas uma informação pequena de estado a cada minuto, uma rede *ad hoc* orientada a conexão mais barata poderia ser uma solução melhor. Além disso, procedimentos de registro poderiam levar muito tempo, e o *overhead* seria muito grande com redes existentes.

### 1.9.1 - ROTEAMENTO

Enquanto em redes sem fio com infra-estrutura e suporte a estação base alcança sempre todos os nós móveis, isto não é sempre o caso em uma rede *ad hoc*. Um nó destino pode estar fora da faixa de um nó fonte transmitindo pacotes. Assim, o roteamento é necessário para encontrar um caminho entre a fonte e o destino e enviar os pacotes adequadamente. Em redes sem fios usando uma infra-estrutura, células devem ser definidas. Dentro de uma cela, uma estação base pode alcançar todos os nós móveis sem rotear via *broadcast*. No caso da rede *ad hoc*, cada nó deve ser capaz de enviar dados para outros nós, isto, no entanto cria muitos problemas adicionais.

A Figura 1.11 mostra um exemplo simples de uma rede *ad hoc*. Em um certo momento  $t_1$  a topologia da rede pode aparecer no lado esquerdo da figura. Cinco nós, de N1 à N5 são conectados dependendo das características de transmissão atuais entre eles. Neste momento da rede, N4 pode receber N1 através de um bom *link*, mas N1 só recebe N4 por um *link* ruim. Assim, *links* não nulos necessariamente não têm as mesmas características em ambas as



direções. Razões para isto são, por exemplo, características de antenas diferentes ou potência de transmissão. N1 não pode receber nada de N2 e N2 recebe um sinal de N1.

Esta situação pode mudar rapidamente como mostrado no tempo  $t_2$ . N1 não pode receber N4, N4 só recebe N1 por um *link* ruim. Mas agora N1 têm um *link* assimétrico bidirecional para N2 que não existia antes.

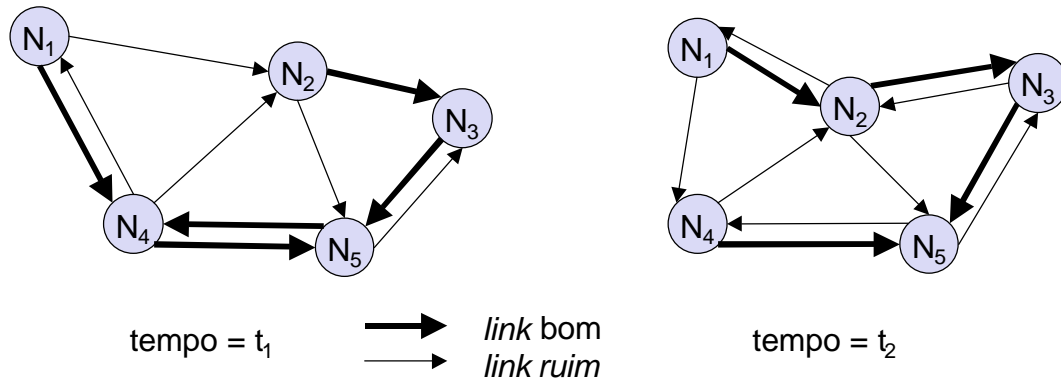


Figura 1.11 – Exemplo de redes *ad hoc*

Este exemplo simples já mostra algumas diferenças fundamentais entre redes com fio e redes *ad hoc* sem fio relacionadas ao roteamento.

- *Links* redundantes: Redes fixas também têm *links* redundantes para sobreviver à falhas nos *links*. Porém, há apenas alguma redundância em redes fixas que, adicionalmente, é controlado por um administrador de rede. Em redes *ad hoc* não se controla redundância, assim poderia haver muitos *links* redundantes até o extremo de uma topologia completamente confusa. Algoritmos de roteamento para redes fixas podem controlar alguma redundância, mas uma redundância alta pode causar um grande *overhead* computacional para atualizações nas tabelas de roteamento.

- Links assimétricos: Se um nó A recebe um sinal do nó B este não informa nada sobre a qualidade da conexão na direção reversa. O nó B não poderia receber nada, tendo um *link* ruim, ou até mesmo tendo um *link* melhor que a direção reversa. Assim, a informação de roteamento coletada de uma direção é de quase nenhum uso para a outra direção. Porém, muitos algoritmos de roteamento para redes fixas confiam em um usuário assimétrico.



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

- Interferência: Em redes fixas *links* existem somente onde o cabeamento existe, e conexões são planejadas por administradores de rede. Isto não é o caso de uma rede *ad hoc* sem fios. Os *links* vêm e vão dependendo das características de transmissão, uma transmissão poderia interferir em outra, e os nós poderiam escutar transmissões de outros nós. Interferências assim, criam novos problemas por *links* não planejados entre nós: se dois nós se cruzarem durante duas transmissões, eles poderiam interferir e destruir um ao outro. Por outro lado a interferência poderia ajudar no roteamento. Um nó pode apresentar a topologia com ajuda de pacotes escutados.

- Topologia dinâmica: O maior problema para o roteamento surge da topologia altamente dinâmica. Os nós móveis poderiam se mover como mostrado na figura 1.11 ou características do meio poderiam mudar. Isto resulta em mudanças freqüentes na topologia. Em redes *ad hoc*, tabelas de roteamento devem refletir de alguma maneira na topologia e algoritmos de roteamento devem ser adaptados. Algoritmos de roteamento usados em redes fixas reagiriam muito devagar ou gerariam muitas atualizações para refletir todas as mudanças na topologia. Atualizações na tabela de roteamento em redes fixas, por exemplo, levam 30 segundos. Essas frequentes atualização poderiam ser muito baixas para serem úteis em redes *ad hoc*. Além disso, alguns algoritmos confiam em uma tabela completa de toda a rede. Enquanto estes trabalham em redes fixa onde mudanças são raras, eles falham completamente em redes *ad hoc*. A topologia muda durante a distribuição do atual estado instantâneo da rede, tornando essa atualização inútil.

Uma outra situação seria assumir que o nó N1 quer enviar dados para N3 e precisa de uma confirmação. Se N1 tiver uma visão completa da rede no tempo  $t_1$ , o qual não é sempre o caso em redes *ad hoc*, ele escolheria o caminho N1, N2, N3, para isto, requerem apenas dois saltos (usando-se saltos como métrica). Confirmações não podem seguir o mesmo caminho, N3 escolhe N3, N5, N4, N1. Isto leva três saltos e já mostra que o roteamento também influência na função das camadas mais altas. Por exemplo, o TCP faz medidas de tempo de ida-e-volta que assumem o mesmo caminho em ambas as direções. Isto está obviamente errado no exemplo mostrado e conduz assim a interpretações incorretas de medidas e ineficiências.

Em um momento posterior, o tempo  $t_2$ , mudou a topologia. Agora N3 não pode percorrer o mesmo caminho para enviar confirmação de volta para N1, enquanto N1 ainda pode usar o antigo caminho para N3. Embora mais complicado que redes fixas, este exemplo



ainda assume que os nós podem ter uma avaliação completa da situação atual. O conhecimento ótimo para todos os nós seria uma descrição da conectividade atual entre todos os nós, o fluxo de tráfego esperado, capacidades de todos os *links*, o atraso de cada *link*, e a capacidade de processamento de cada nó. Enquanto até mesmo em redes fixas os tráfegos não são exatamente conhecidos, para *links* de rede *ad hoc* as capacidades são adicionalmente desconhecidas. A capacidade de cada *link* pode mudar de 0 ao máximo da tecnologia de transmissão usada. Em redes *ad hoc* na realidade nenhum nó sabe todos estes fatores, e estabelecer atualizações instantâneas na rede é quase impossível.

Redes *Ad hoc* usando nós móveis enfrentam problemas adicionais devido a limitações de *hardware*. Usar protocolos de roteamento padrão com atualizações periódicas causa desperdícios de energia sem enviar qualquer dado de usuário e incapacita modos de espera. Além disso, atualizações periódicas desperdiçam largura de banda em *links* sem fio e recursos de largura de banda já escassos. Um problema adicional que não existe em redes fixas é a interferência entre duas ou mais transmissões que não usam os mesmos nós para se comunicarem. Por exemplo, se uma segunda transmissão do nó N4 ao nó N5 (Figura 1.11) acontecer ao mesmo tempo com uma transmissão do nó N1 ao nó N3, eles poderiam interferir.

Interferência poderia acontecer no nó N2 que pode receber sinais de N1, e N4, ou em N5; recebendo N4 e N2. Se protegido corretamente, não haverá nenhuma interferência entre os dois fios.

Considerando todas as dificuldades adicionais em comparação com as redes fixas, faz-se as seguintes observações com relação ao roteamento em redes *ad hoc* com nós móveis:

- Algoritmos de roteamento tradicionais das redes fixas não trabalharão eficientemente ou falham completamente. Estes algoritmos não foram projetados com uma topologia dinâmica, *links* assimétricos, ou interferência em mente.
- Roteamento em redes *ad hoc* sem fios não podem depender apenas de 3 camadas. As informações das camadas inferiores relativas a conectividade ou interferência pode auxiliar os algoritmos de roteamento a encontrarem um melhor percurso.
- Abordagens centralizadas realmente não funcionaram, porque levam muito tempo para coletar o estado atual e disseminá-lo novamente. Durante este tempo a topologia já mudou.



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

- Muitos nós necessitam da capacidade de roteamento. Ao menos um roteador tem que estar dentro do alcance de cada nó. Algoritmos têm que tomar o cuidado com limitações de potência destes nós.
- A noção de uma conexão com certas características não pode trabalhar apropriadamente. Redes *ad hoc* trabalharão sem conexão, porque não é possível manter uma conexão em um ambiente que varia rapidamente e envia dados que seguem esta conexão. Os Nós têm que tomar decisões locais para entregar e enviar os pacotes para o destino final
- Uma última alternativa para enviar um pacote através de uma topologia desconhecida é o *flood-ing*. Esta abordagem sempre trabalha se a carga é baixa, mas é muito ineficiente. Um controlador de salto é necessário em cada pacote para evitar o *loop*, e o diâmetro da rede *ad hoc*, isto é, o número máximo de saltos deve ser conhecido.
- *Clusters* hierárquicos de nós podem ajudar. Se for possível identificar certos grupos de nós que trabalham juntos, os *clusters* podem ser estabelecidos. Enquanto os nós individuais podem se mover mais rápido, o *cluster* inteiro pode ser bastante estacionário. Assim o roteamento entre os *clusters* devem ser mais simples e menos dinâmico.

No capítulo que segue, dar-se-á continuidade a mobilidade apresentando a camada de transporte móvel, e seu respectivo protocolo de transporte, o TCP móvel. Uma rápida visão do TCP tradicional será apresentada juntamente com otimizações do mesmo, chegando em fim em suas dificuldades e limitações para com a mobilidade e então o TCP como um protocolo orientado a conexão em um ambiente móvel será apresentado com algumas aplicações e também seus respectivos problemas. Este capítulo seguinte foi baseado na referência bibliográfica [1].





**CAPÍTULO 2**  
**CAMADA DE TRANSPORTE MÓVEL**

**2.1 – INTRODUÇÃO**

O suporte à mobilidade apenas nas camadas inferiores da camada de rede não é suficiente para proporcionar suporte de mobilidade para todas as aplicações . Muitas aplicações dependem da camada de transporte, como o TCP (*Transmission Control Protocol*) ou o UDP, no caso da Internet. Duas funções da camada de transporte na Internet são *checksum* sobre os dados do usuário e a multiplexação/demultiplexação de dados de/para aplicações. Enquanto a camada de rede apenas endereça o *host* , as portas no UDP ou TCP permitem o endereçamento de aplicações dedicadas. Enquanto o UDP não trabalha orientado a conexão e não dá certas garantias sobre a entrega dos dados, o TCP é muito mais complexo e, portanto, precisa de mecanismos especiais para ser útil em ambientes móveis. Para o UDP trabalhar, basta que haja suporte de mobilidade apenas no IP (como ocorre no IP móvel).

A principal diferença entre o UDP e o TCP é que o TCP oferece conexões entre duas aplicações. Dentro de uma conexão TCP pode-se dar certas garantias, como a entrega na ordem ou transmissão segura de dados usando técnicas de retransmissão. Além disso, se o TCP encontrar pacotes perdidos, ele assume que há um congestionamento na rede e então diminui sua taxa de transmissão. Isto também é uma das razões principais para escolher protocolos como o TCP e não escolher outros mais simples como o UDP. O UDP requer que as aplicações controlem a entrega segura e em ordem dos dados e etc. Além disso, o UDP não se comporta bem em “redes amigáveis”, por exemplo, não diminuindo sua taxa de transmissão em caso de congestionamento .

A seguir será dada uma visão dos mecanismos TCP que fazem um papel importante no uso do TCP à mobilidade. O principal problema dos muitos mecanismos é que estes têm sido projetados sob suposições completamente diferentes daquelas em redes móveis. Baseado nestes problemas os quais podem conduzir a um completo colapso no tráfego do TCP foi desenvolvido um conjunto de soluções, algumas destas serão discutidas a seguir com suas vantagens e desvantagens.



## 2.2 - TCP TRADICIONAL

Serão destacados vários mecanismos do TCP, e sua influência à eficiência em um ambiente móvel.

### 2.2.1 - CONTROLE DE CONGESTIONAMENTO

Protocolos de camada de transporte como o TCP foram projetados para redes fixas fim-a-fim. Na transmissão de dados eles usam adaptadores de rede, fibra óptica, fios de cobre, *hardware* especial para roteadores e etc. Este *hardware* trabalha tipicamente sem introduzir erros de transmissão. Além disso, se o *software* for eficiente o bastante, não derrubará pacotes ou inverterá bits. Então, se um pacote em seu caminho de um emissor para um receptor for perdido em uma rede fixa, não quer dizer que houve erros de *hardware* ou *software*. Mas, o mais provável seria considerar que houve uma sobrecarga temporária a algum ponto do caminho de transmissão, por exemplo, um estado de congestionamento de algum nó.

O congestionamento pode aparecer de vez em quando até mesmo em redes cuidadosamente projetadas. Os *buffers* de pacotes do roteador estão preenchidos e o roteador não pode enviar os pacotes rapidamente porque a soma da taxa de entrada de pacotes destinados para um *link* de saída é maior que a capacidade deste *link* de saída. A única coisa que o roteador pode fazer nesta situação é descartar pacotes. Um pacote descartado é perdido na transmissão, e o receptor nota um *gap* (espaço) no fluxo de pacotes. Agora, o receptor não “diz” diretamente para o emissor qual pacote está faltando, e continua confirmando todos os pacotes na sequência até o perdido.

O emissor nota a falta da confirmação de um pacote perdido e assume a perda do pacote devido a congestionamento. Retransmitir o pacote perdido e continuar com a mesma taxa de transmissão, não seria prudente, pois isto poderia somente aumentar o congestionamento. Embora não esteja garantido que todos os pacotes de uma conexão TCP trafeguem no mesmo caminho através da rede, esta suposição é levada em consideração para a maioria dos pacotes. Para aliviar o congestionamento, o TCP diminui sua taxa de transmissão dramaticamente. Todas as outras conexões de TCP que experimentam o mesmo congestionamento fazem



exatamente o mesmo, assim, o congestionamento é logo resolvido. Esta co-operação de conexões TCP na Internet é uma das principais razões para a sobrevivência da Internet até os dias atuais. Até mesmo sob uma grande quantidade de tráfego, o TCP garante pelo menos o compartilhamento da largura de banda.

### 2.2.2 - INÍCIO LENTO

A reação do TCP para uma confirmação perdida é bastante drástica, mas é necessária para livrar-se de um congestionamento o mais rápido possível. O comportamento do TCP após a detecção do congestionamento é chamado de início lento.

O emissor sempre calcula uma janela de congestionamento para um receptor. O tamanho inicial da janela de congestionamento é um segmento (pacote TCP). Agora o emissor envia um pacote e espera a confirmação. Se esta confirmação chegar, o emissor incrementa a janela de congestionamento, agora enviando dois pacotes (janela de congestionamento = 2). Depois da confirmação correspondente aos dois pacotes que chegaram, o emissor soma novamente 2 à janela de congestionamento, um para cada uma das confirmações. Agora a janela de congestionamento iguala-se a 4. Este esquema dobra a janela de congestionamento toda vez que as confirmações efetivam-se o qual leva um tempo de retardo de transmissão (*Round Trip Time* - **RTT**), que consiste no tempo total que um único pacote ou datagrama leva para alcançar o outro e retornar. Isto é chamado de crescimento exponencial da janela de congestionamento no mecanismo de início lento.

É muito perigoso dobrar a janela de congestionamento a cada vez que uma confirmação é recebida, pois os passos poderiam ficar muito grandes. Então, o crescimento exponencial pára no limiar de congestionamento. Assim que a janela de congestionamento alcançar o limiar de congestionamento, o aumento adicional da taxa de transmissão é somente linear incrementando a janela de congestionamento cada vez que uma confirmação é recebida.

O aumento linear continua até acontecer um *timeout* no emissor que ocorre devido à perda de uma confirmação, ou até que o emissor detecte a falta de um dado transmitido por causa das contínuas confirmações de um mesmo pacote. Nesse caso o emissor configura o limiar de congestionamento para a metade da janela de congestionamento atual. A própria janela de congestionamento é configurada para um segmento e o emissor envia um único segmento. O crescimento exponencial como já descrito inicia mais uma vez até um novo limiar de congestionamento, então a janela cresce de modo linear.



### 2.2.3 – RETRANSMISSÃO RÁPIDA/RECUPERAÇÃO RÁPIDA

Estas são duas razões, dirigidas para a redução do limiar de congestionamento, como mencionado antes. Uma é o emissor recebendo contínuas confirmações de um mesmo pacote. Isto informa ao emissor sobre duas coisas. Uma é que o receptor adquiriu todos os pacotes até o pacote confirmado na sequência. Em TCP, um receptor envia as confirmações apenas se receber algum pacote do emissor. Assim, receber confirmações de um receptor mostra adicionalmente que o receptor recebe continuamente algo do emissor. Dessa forma o *gap* no fluxo de pacotes não é devido um congestionamento severo, mas a uma simples perda de pacotes devido a um erro de transmissão. O emissor pode agora retransmitir o pacote perdido antes que o tempo expire. Este comportamento é chamado de retransmissão rápida.

Além disso, a recepção de confirmações mostra que não há nenhum congestionamento que justifique um início lento, assim o emissor pode continuar com a atual janela de congestionamento. O emissor executa uma rápida recuperação da perda dos pacotes. Este mecanismo pode melhorar a eficiência do TCP.

A outra razão para ativação do início lento mencionado anteriormente foi um *timeout* devido à falta de confirmação. Esta é a única situação TCP usando retransmissão rápida/recuperação rápida que interpreta como congestionamento na rede e ativa o mecanismo de início lento.

### 2.2.4 - IMPLICAÇÕES NA MOBILIDADE

Enquanto o início lento é um dos mais usuais mecanismos em redes fixas, este, diminui a eficiência do TCP se usado junto com receptores ou emissores móveis. A razão para isto é o uso do início lento sob as suposições erradas. A partir da falta de uma confirmação, o mecanismo de início lento conclui uma situação de congestionamento. Isto também pode acontecer em redes com sistemas-fins móveis e sem fio, mas esta não é a principal razão para perda de pacotes.

Taxas de erros em *links* sem fio são muito maiores quando comparadas à fibra fixa ou ligações de cobre. Assim, pacotes perdidos são muito mais comuns e nem sempre podem ser compensados através de retransmissão da camada dois. Tentar retransmitir na camada dois



poderia, por exemplo, ativar uma retransmissão TCP se ela levar muito tempo. Agora a camada dois enfrenta o problema de transmitir o mesmo pacote duas vezes sobre um provável *link* ruim. Detectar estas duplicatas na camada dois não é uma opção, porque cada vez mais conexões usam codificação fim-a-fim, tornando impossível traduzir o pacote.

Além disso, a mobilidade por si só pode causar perda de pacotes. Há muitas situações onde o *soft handover* de um ponto de acesso para outro não é possível para um sistema-fim móvel. Por exemplo, usando o IP móvel, ainda poderia haver alguns pacotes em trânsito para o agente estrangeiro velho enquanto o nó móvel move-se para um agente estrangeiro novo. Pode ser agora o caso que o agente estrangeiro velho não pode enviar esses pacotes ao agente estrangeiro novo ou nem mesmo os pacotes armazenados se o nó móvel levar muito tempo desconectado. Esta perda de pacote não tem nada a ver com acesso sem fio mas é causada por problema de reencaminhamento de tráfego.

O mecanismo TCP detecta confirmações perdidas através do *timeout* e conclui a perda de pacote devido a congestionamento, ele não poder distinguir entre as diferentes causas. O TCP reage com início lento o qual não ajuda no caso de erros de transmissão sobre *links* sem fio e realmente não ajuda durante o processo de *handover*. Este comportamento resulta em uma degradação da performance severa de um inalterado TCP se usado junto com *links* sem fio ou nós móveis.

Porém, não se pode mudar completamente o TCP apenas para suportar usuários móveis ou *links* sem fio. Os mesmos argumentos que foram usados para manter IP inalterado também se aplicam ao TCP. A base instalada de computadores que usam TCP é muito grande para ser mudada e, mais importante, mecanismos como início lento mantêm a Internet operável. Então, toda alteração no TCP tem que se manter compatível ao padrão TCP e não deve por em risco o comportamento cauteloso do TCP no caso de congestionamento.

### 2.3 - TCP INDIRETO

Duas situações conflitantes conduziram ao desenvolvimento de TCP indireto (I-TCP). Uma é aquela que executa TCP pobremente em *links* sem fio, a outra é a que usa TCP dentro da rede fixa e que não pode ser mudado. Então, I-TCP segmenta uma conexão TCP em uma parte fixa e uma parte sem fios. A figura 2.1 mostra um exemplo com um *host* móvel conectado através de um *link* sem fio a um ponto de acesso para a Internet com fio onde o



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

*host* correspondente reside. O nó correspondente poderia também usar acessos sem fios. O que segue poderia também ser aplicado ao *link* de acesso do *host* correspondente.

Entre o computador fixo e o ponto de acesso o padrão TCP é usado. Assim, nenhum computador na Internet reconhece qualquer mudança para TCP. Agora em vez do *host* móvel, o ponto de acesso termina a conexão padrão de TCP, atuando como *proxy* (como um representante). Isto significa que o ponto de acesso é agora visto como um *host* móvel para o *host* fixo e como um *host* fixo para o *host* móvel. Entre o ponto de acesso e o *host* móvel, um TCP especial, adaptado para *links* sem fio, é usado. Porém, variar o TCP para o *link* sem fio não é uma exigência. Até mesmo um TCP inalterado pode beneficiar-se com o menor RTT, assim iniciando a retransmissão rapidamente. Um lugar bom para segmentar a conexão entre o *host* móvel e o *host* correspondente está no agente estrangeiro do IP móvel. O agente estrangeiro controla a mobilidade do *host* móvel de qualquer maneira e também pode entregar a conexão ao próximo agente estrangeiro quando o *host* móvel se mover. Porém, pode-se também imaginar separar as conexões de TCP em um servidor especial, por exemplo, no ponto de entrada para uma rede de telefonia móvel.

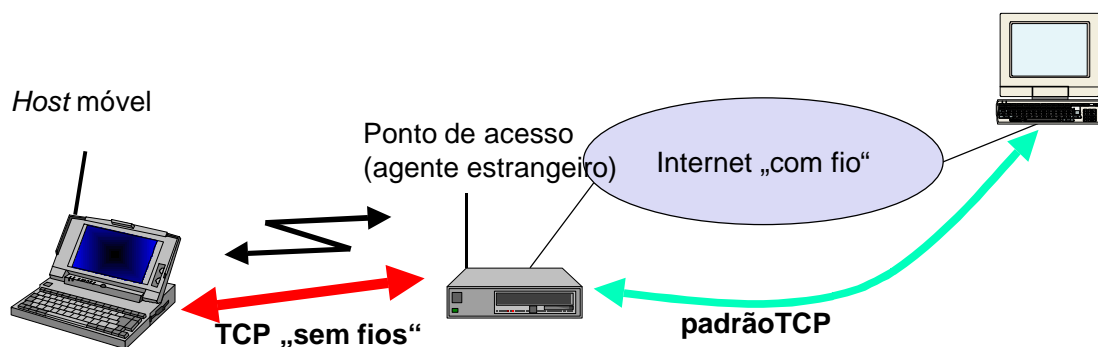


Figura 2.1 – Segmentos de TCP indireto segmenta uma conexão TCP em duas partes

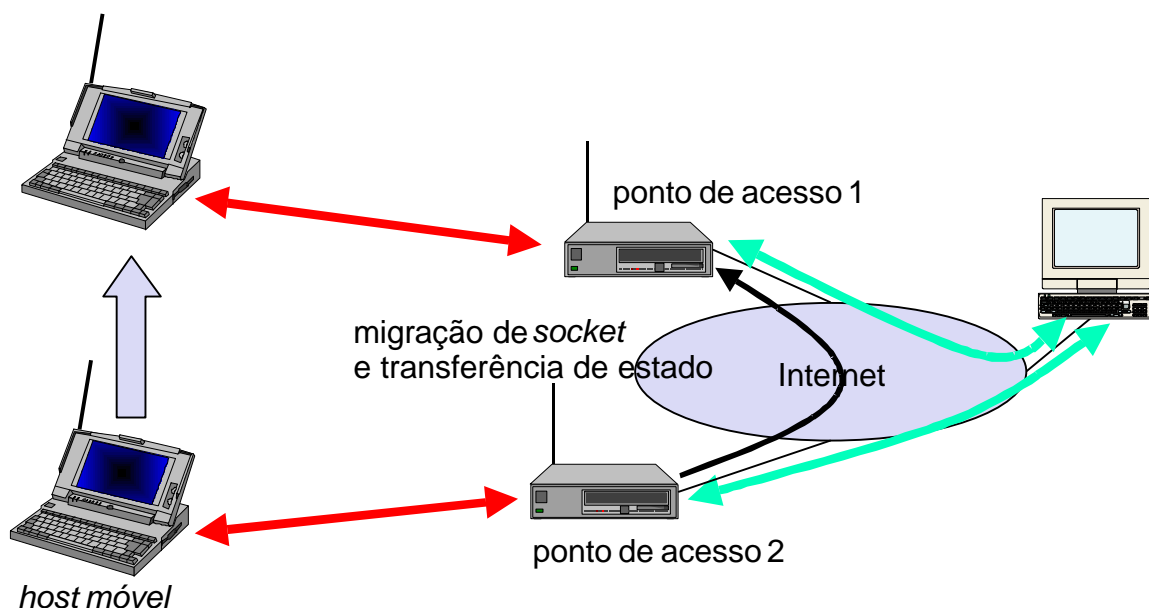
O *host* correspondente na rede fixa não nota o *link* sem fio ou a segmentação da conexão. O agente estrangeiro atua como um *proxy* e retransmite todos os dados em ambas as direções. Se o *host* correspondente enviar um pacote, o agente estrangeiro confirma este pacote. Então o agente estrangeiro tenta enviar o pacote para o *host* móvel. Se o *host* móvel receber o pacote, ele confirma o pacote. Porém, esta confirmação é apenas usada pelo agente estrangeiro. Se um pacote é perdido em um *link* sem fio devido a um erro de transmissão, o



*host* correspondente não notará isto. Porém, o agente estrangeiro tenta retransmitir este pacote localmente para manter seguro o transporte de dados.

Analogamente, se o *host* móvel enviar um pacote, o agente estrangeiro reconhece este pacote e tenta remete-lo ao *host* correspondente. Se o pacote for perdido no *link* sem fio, o *host* móvel pode notar isto muito mais rápido devido ao menor RTT e pode diretamente retransmitir o pacote. Perdas de pacotes na rede fixa são agora controladas pelo agente estrangeiro.

O I-TCP requer várias ações assim que um *handover* ocorra. Como demonstra a figura 2.2, não só os pacotes têm que ser redirecionados usando, por exemplo, IP móvel. No exemplo mostrado, os pontos de acesso atuam como *proxys* e *buffers* de pacotes para retransmissão. Após o *handover*, o velho *proxy* tem que enviar os dados armazenados para o novo *proxy* porque este já confirmou os dados. Após o registro com o novo agente estrangeiro, este novo agente estrangeiro pode informar o velho sobre seu local para facilitar o envio de pacotes. Além do conteúdo protegido do *buffer*, os *sockets* (ponto por meio do qual os programas de aplicações podem enviar e receber dados) de *proxy*, também, devem migrar para o novo agente estrangeiro localizado no ponto de acesso. O *socket* reflete o estado atual da conexão TCP, isto é, números de sequência, endereços, portas etc. Nenhuma conexão nova pode ser estabelecida para o *host* móvel, e o *host* correspondente não deve “ver” nenhuma mudança no estado da conexão.







## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

Figura 2.2 - Migração de estado e *socket* após o *handover* de um *host* móvel

Várias vantagens vêm com o I-TCP:

- O I-TCP não requer nenhuma mudança no protocolo TCP usado pelos *hosts* na rede fixa ou outros *hosts* em uma rede sem fio que não usa esta otimização. Assim, todas as otimizações atuais para o TCP ainda trabalham entre o agente estrangeiro e o *host* correspondente.
- Devido à rígida partição em duas conexões, erros de transmissão no *link* sem fio e pacotes perdidos, não podem se propagar em redes fixas. Sem partição, retransmissão de pacotes perdidos aconteceriam entre o *host* móvel e o *host* correspondente através de toda rede. Agora apenas os pacotes em sequência sem *gaps* deixam o agente estrangeiro.
- É sempre perigoso introduzir novos mecanismos em uma rede enorme como a Internet sem saber exatamente seus comportamentos. Porém, novos mecanismos são necessários para melhorar a performance do TCP (por exemplo, desabilitar o início lento sob certas circunstâncias, mas com I-TCP apenas entre o *host* móvel e o agente estrangeiro). Podem ser testadas diferentes soluções ou podem ser usadas ao mesmo tempo sem comprometer a estabilidade da Internet. Além disso, otimizações destes novos mecanismos são bastante simples porque eles só cobrem um único salto.
- O curto atraso entre o *host* móvel e o agente estrangeiro pode ser determinado e é independente de outros fluxos de tráfego. Então, um TCP otimizado pode usar *timeouts* precisos para garantir a retransmissão tão rápida quanto possível. Até mesmo o TCP padrão se beneficia do curto RTT, assim, recuperando rapidamente a perda do pacote.
- Partição em duas conexões também permitem o uso de diferentes protocolos na camada de transporte entre o agente estrangeiro e o *host* móvel ou o uso de cabeçalhos comprimidos e etc. Os agentes estrangeiros podem agora atuar como *gateways* para se comunicarem entre os diferentes protocolos.

Mas a idéia de segmentação em I-TCP também apresenta algumas desvantagens:

- A perda das semânticas fim-a-fim do TCP poderia causar problemas se o agente estrangeiro que particiona as conexões TCPs colidisse. Se um emissor recebe uma confirmação, ele assume que o receptor recebeu o pacote. Receber agora uma confirmação significa somente (para o *host* móvel e o *host* correspondente) que o agente estrangeiro recebeu o pacote. O nó correspondente não sabe nada sobre a partição, assim um nó de acesso



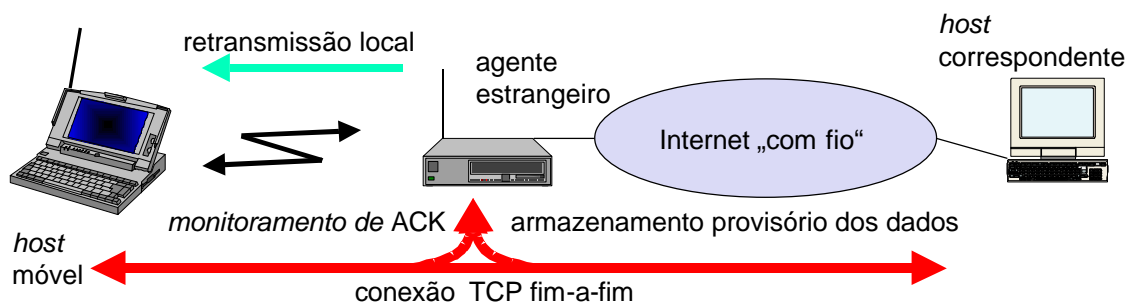
pode também colidir aplicações que correm no nó correspondente e que assume entrega fim-a-fim segura.

- Na prática, um aumento na latência de *handover* pode ser muito mais problemático. Todos os pacotes enviados pelo *host* correspondente são armazenados pelo agente estrangeiro e enviados ao *host* móvel (se a conexão TCP é particionada no agente estrangeiro). O agente estrangeiro remove um pacote do *buffer* assim que uma confirmação apropriada chegue. Se o *host* móvel, agora executar um *handover* para outro agente estrangeiro, demora um tempo antes do agente estrangeiro velho poder enviar os dados armazenados ao agente estrangeiro novo. Durante este tempo podem chegar mais pacotes. Todos estes pacotes devem ser enviados primeiro para o agente estrangeiro novo antes que se possa iniciar o envio dos pacotes novos redirecionados.

- O agente estrangeiro deve ser uma entidade de confiança porque as conexões TCP terminam neste ponto. Se os usuários aplicam codificação fim-a-fim, de acordo com o RFC 825 o agente estrangeiro tem que ser integrado a todos os mecanismos de segurança.

## 2.4 - TCP MONITOR

Um das desvantagens do I-TCP é a segmentação de uma única conexão TCP em duas conexões TCP, perdendo a semântica original TCP fim-a-fim. A seguinte otimização TCP trabalha completamente transparente e deixa intacta a conexão TCP fim-a-fim. A principal função da otimização é armazenar provisoriamente dados perto do *host* móvel para executar uma rápida retransmissão local no caso de perdas de pacotes. Novamente, um lugar bom para a otimização do TCP poderia ser no agente estrangeiro no contexto de IP Móvel (Figura 2.3).





## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

Figura 2.3 – TCP monitor como uma extensão transparente do TCP

Nesta aproximação, o agente estrangeiro armazena provisoriamente todos os pacotes com destino ao *host* móvel e adicionalmente monitora o fluxo de pacote em ambas as direções reconhecendo confirmações. A razão para armazenar provisoriamente pacotes para o nó móvel é permitir que o agente estrangeiro execute uma retransmissão local no caso da perda de pacote em *links* sem fio. O agente estrangeiro armazena todos os pacotes até receber uma confirmação do *host* móvel. Se o agente estrangeiro não receber uma confirmação do *host* móvel dentro de um certo intervalo de tempo, ou o pacote ou a confirmação foi perdido. Alternativamente, o agente estrangeiro poderia receber um ACK (confirmação) duplicado que também mostra a perda de um pacote. Agora o agente estrangeiro retransmite o pacote diretamente do *buffer*, executando assim uma retransmissão muito mais rápida quando comparada ao *host* correspondente. O *timeout* para as confirmações pode ser fixado muito mais curto, para que ele reflita somente o tempo de um salto mais o tempo de processamento.

Para permanecer transparente, o agente estrangeiro não deve confirmar dados ao *host* correspondente. Se assim fizesse o *host* correspondente poderia acreditar que o *host* móvel havia recebido os dados. Isto violaria a semântica fim-a-fim no caso de um fracasso do agente estrangeiro. Porém, o agente estrangeiro pode filtrar as confirmações duplicadas para evitar retransmissões desnecessárias de dados do *host* correspondente. Se agora o agente estrangeiro falhar, o *timeout* do *host* correspondente trabalha e ativa uma retransmissão. Além disso, o agente estrangeiro deve descartar pacotes duplicados já retransmitidos localmente e confirmados pelo *host* móvel. Isto evita tráfegos desnecessário no *link* sem fio.

Transferir dados do *host* móvel para o *host* correspondente ocorre de seguinte maneira. O agente estrangeiro monitora o fluxo de pacotes para descobrir *gaps* nas seqüências de números do TCP. Assim que o agente estrangeiro descobre um pacote perdido, ele devolve uma confirmação negativa (NACK) para o *host* móvel. O *host* móvel pode agora retransmitir imediatamente o pacote perdido. A reordenação de pacotes é feita automaticamente no *host* correspondente pelo TCP.

Estender as funções de um agente estrangeiro com um TCP monitor apresenta várias vantagens:

- Uma grande vantagem desta abordagem é a preservação da semântica fim-a-fim do TCP. Não importa em que momento o agente estrangeiro falha (se este é o local de



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

armazenamentos provisórios e dos mecanismos de monitoração), nem o *host* correspondente nem o *host* móvel tem uma visão inconsistente da conexão TCP como é possível com o I-TCP. A aplicação retrocede automaticamente ao modelo TCP padrão se a otimização para de trabalhar.

- Nenhum *host* correspondente precisa ser mudado, a maioria dos ajustes está no agente estrangeiro. Suportar somente o fluxo de pacote do *host* correspondente para o *host* móvel também não requer mudanças no *host* móvel.

- Um bom efeito desta abordagem é que não é necessário um *handover* de estado assim que o *host* móvel se mova para outro agente estrangeiro. Assuma que devem ainda haver dados no *buffer* não transferidos para o próximo agente estrangeiro. Tudo o que acontece é um *timeout* ao *host* correspondente e a retransmissão dos pacotes, possivelmente já para um novo endereço.

- Além disso, não importa se o novo agente estrangeiro usa a variação ou não. Se não, a abordagem automaticamente volta para a solução padrão. Isto é um dos problemas do I-TCP, pois o agente estrangeiro velho pode já ter sinalizado a recepção correta de dados via confirmações para o *host* correspondente e agora tem que transferir estes pacotes para o *host* móvel via agente estrangeiro novo.

Porém, a simplicidade desta abordagem também resulta em algumas desvantagens:

- O TCP monitor não isola o comportamento do *link* sem fio tão bem quanto o I-TCP. Por exemplo, o TCP monitor leva algum tempo até que o agente estrangeiro possa retransmitir com sucesso o pacote do seu *buffer* devido a problemas no *link* sem fio. Embora o *timeout* no agente estrangeiro possa ser muito mais curto que em um *host* correspondente, após o tempo de *timeout* o *host* correspondente ativa uma retransmissão. Assim, os problemas em *links* sem fios são agora também visíveis para o *host* correspondente e não completamente isolado. A qualidade do isolamento que o TCP monitor oferece depende fortemente da qualidade do *link* sem fio, valores de *timeout*, e características de tráfegos adicionais.

- O uso de confirmações negativas entre o agente estrangeiro e o *host* móvel assume mecanismos adicionais no *host* móvel. Assim, esta abordagem não é tão transparente para os *hosts* móveis arbitrários.

- Finalmente, todo esforço para monitorar e armazenar provisoriamente dados podem ser inúteis se certos esquemas de codificação forem aplicados fim-a-fim entre o *host* correspondente e o *host* móvel. Usando encapsulamento seguro do *payload* do IP o cabeçalho



do protocolo TCP também será codificado, assim a monitoração dos números de sequência não funcionarão mais. As codificações fim-a-fim são usadas por muitas aplicações, e ainda não está claro como este esquema poderia ser usado no futuro.

## 2.5 - TCP MÓVEL

Uma baixa taxa no envio de pacotes devido a um *handover* ou altas taxas de erros de bits não são os únicos fenômenos em *links* sem fios e mobilidade. Outro é a ocorrência de prolongadas e/ou freqüentes desconexões. Para usuários móveis isso acontece com bastante freqüência na qual eles não podem conectar nada.

Um emissor TCP tenta retransmitir dados sendo controlado por um temporizador de retransmissão que dobra com cada tentativa de retransmissão malsucedida, até um máximo de um minuto. Isto significa que o emissor tenta retransmitir um pacote não confirmado durante um minuto e cessará após doze retransmissões. Se a conexão melhorar, nenhum dado é transmitido com sucesso nesse período de um minuto. O *timeout* da retransmissão é ainda válido e o emissor tem que esperar. Além disso, o emissor entra em início lento porque ele assume congestionamento.

No caso de I-TCP se a mobilidade é desconectada o *proxy* tem que armazenar provisoriamente cada vez mais dados, e assim, quanto maior o período de desconexão mais *buffer* é necessário. Adicionalmente, se um *handover* segue a desconexão, que é típico, igualmente mais estados tem que ser transferidos para um novo *proxy*. A abordagem de monitoramento também sofre para ser desconectada. A mobilidade não é capaz de enviar ACKs, e assim, a monitoração não pode ajudar nesta situação.

A abordagem M-TCP (TCP móvel) tem aproximadamente as mesmas metas que o I-TCP e o TCP monitor: evitar que a janela do emissor diminua se erros de bits ou desconexões ocorrerem não sendo estes/estas entendidos como congestionamento. O M-TCP pretende melhorar o processamento global, diminuir o atraso, manter a semântica fim-a-fim do TCP, e proporcionar *handover* mais eficiente. Adicionalmente, o M-TCP é especialmente adaptado aos problemas que surgem de desconexões prolongadas ou freqüentes.

O M-TCP particiona a conexão TCP em duas partes como o I-TCP faz. Um TCP inalterado é usado na conexão padrão entre *host* e *host-supervisory* (SH), enquanto um TCP otimizado é usado na conexão de SH-MH. O *host* supervisor é responsável pelo intercâmbio



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

de dados entre ambas as partes, similar ao *proxy* no I-TCP (figura 2.1). A abordagem M-TCP assume baixa taxa de erro de bit no *link* sem fio. Então, não executa armazenamento/retransmissão de dados no SH. Se um pacote estiver perdido no *link* sem fio, tem que ser retransmitido pelo emissor de origem. Isto mantém a semântica fim-a-fim do TCP.

O SH monitora todos os pacotes enviados ao MH e todas as ACKs retornadas do MH. Se o SH não receber um ACK durante algum tempo, ele assume que o MH está desconectado. Ele então faz com que o emissor configure o tamanho da janela para 0. Configurar o tamanho da janela para 0 força o emissor no modo persistente, isto é, o estado do emissor não mudará não importa quanto tempo o receptor esteja desconectado. Isto significa que o emissor não tentará retransmitir dados. Assim que o SH (o SH antigo ou um SH novo) detecte novamente a conectividade, reabre a janela do emissor para o valor antigo. Assim, o emissor pode continuar enviando em taxa máxima. Este mecanismo não requer mudanças no TCP do emissor.

No lado sem fio um TCP adaptado é usado para que possa recuperar muito mais rapidamente a perda de pacote. Este TCP modificado não usa início lento, assim, o M-TCP precisa de um gerenciador de largura de banda para implementar um compartilhamento justo sobre *links* sem fio.

As vantagens do M-TCP são as seguintes:

- M-TCP mantém semântica TCP fim-a-fim. O SH não envia nenhum ACK próprio, mas envia os ACKs do MH.
- Se o MH estiver desconectado, o M-TCP evita retransmissões desnecessárias, inícios lentos ou rompimento de conexões simplesmente forçando a janela do emissor para 0.
- Uma vez que o M-TCP não armazena provisoriamente dados no SH como o I-TCP faz, não é necessário reenviar estes dados para um novo SH. Pacotes perdidos serão automaticamente retransmitidos ao SH novo.

A falta de *buffers* e o TCP modificado na parte sem fios também têm algumas desvantagens:

- Como o SH não atua como o *proxy* atua no I-TCP, pacotes perdidos no *link* sem fio devido a erros de bit são propagados ao emissor. O M-TCP assume baixas taxas de erros de bits, a qual nem sempre é uma suposição válida.





- Um TCP modificado no *link* sem fio não apenas requer modificações ao *software* do protocolo do MH mas também novos elementos de rede como o gerenciador de largura de banda.

## 2.6 – RETRANSMISSÃO RÁPIDA/RECUPERAÇÃO RÁPIDA

Mover-se para um agente estrangeiro novo pode causar perda de pacote ou *timeout* nos *hosts* móveis ou *hosts* correspondentes. TCP conclui que houve um congestionamento e entra no início lento, embora não haja nenhum congestionamento. Já foi visto os mecanismos de retransmissão rápida/recuperação rápida que um *host* pode usar após receber duplicadas confirmações, concluindo assim que a perda de pacote não foi ocasionada por um congestionamento.

A idéia é forçar artificialmente a retransmissão rápida no *host* móvel e no lado do *host* correspondente. Assim que o *host* móvel se registra em um novo agente estrangeiro usando IP móvel, começa o envio de duplicadas confirmações para o *host* correspondente. A proposta é enviar três confirmações duplicatas. Isto força o *host* correspondente a entrar no modo retransmissão rápida e não ativar o início lento, o *host* correspondente continua enviando com a mesma taxa que enviava antes do *host* móvel se mover para outro agente estrangeiro.

Como o *host* móvel também pode entrar em início lento após se mover para um novo agente estrangeiro, esta abordagem coloca adicionalmente o *host* móvel em retransmissão rápida. Assim, o *host* móvel retransmite todos os pacotes não confirmados usando o tamanho da janela de congestionamento atual sem entrar em início lento.

A vantagem desta abordagem é sua simplicidade. Somente pequenas mudanças no software do *host* móvel já resultam em um aumento de desempenho. Nenhum agente estrangeiro ou *host* correspondente tem que ser mudado.

A principal desvantagem deste esquema é o isolamento insuficiente dos pacotes perdidos. Forçar rápidas retransmissões aumenta a eficiência, mas os pacotes retransmitidos ainda têm que cruzar a rede inteira entre o *host* correspondente e o *host* móvel. Além disso, se o *handover* de um agente estrangeiro para outro levar um tempo mais longo, o *host* correspondente já terá iniciado a retransmissão. A abordagem focaliza uma perda devido o *handover*, assim pacotes perdidos devido a problemas no *link* sem fio não são considerados.





Finalmente, esta abordagem requer mais cooperação entre o IP móvel e a camada TCP, fazendo com que seja difícil mudar um sem influenciar o outro.

## 2.7 - TRANSMISSÃO/CONGELAMENTO DO *TIMEOUT*

Enquanto as abordagens anteriormente apresentadas podem controlar interrupções curtas de conexão, devido a *handover* ou erros de transmissão em *links* sem fio, algumas foram projetadas para interrupções mais longas na transmissão. Exemplos são o uso de *hosts* móveis em um carro dirigido em um túnel, assim perdendo sua conexão, ou um satélite (porém, muitos túneis e metrô provêm conectividade por um telefone móvel), ou um usuário que se move dentro de uma cela sem nenhuma capacidade de deixá-la. Neste caso, o sistema de telefonia móvel interromperá a conexão. A reação do TCP, justamente com as otimizações anteriores, seria uma desconexão depois de um *timeout*.

Frequentemente, a camada MAC já havia notado problemas de conexão, antes da conexão ser de fato interrompida no ponto de vista TCP. Adicionalmente, a camada MAC conhece a real razão para a interrupção e não assume congestionamento, como assumiria TCP. Então, a camada MAC pode informar a camada TCP de uma provável perda de conexão ou que a interrupção atual não é causada por congestionamento. O TCP pode agora parar de enviar e “congelar” o estado atual de sua janela de congestionamento e temporizadores adicionais. Se a camada MAC notar uma provável interrupção cedo o suficiente, o *host* móvel e o *host* correspondente podem ser informados. Com uma rápida interrupção nos *links* sem fio, são necessários mecanismos adicionais no ponto de acesso para informar o *host* correspondente da razão da interrupção. Caso contrário, o *host* correspondente entra em início lento assumindo congestionamento e finalmente interrompe a conexão.

Assim que a camada MAC detectar novamente a conectividade, ela sinaliza para o TCP que ele pode retomar a operação exatamente no mesmo ponto onde tinha sido forçado a parar. Para o TCP simplesmente nenhum tempo avançou, e assim nenhum temporizador expirou.

A vantagem deste modo é que oferece um caminho para retomar conexões de TCP justamente depois de interrupções mais longas na conectividade. Além disso, é independente de qualquer outro mecanismo TCP, como confirmações ou sequência de números, assim pode ser usado junto com dados codificados. Porém há algumas desvantagens severas neste esquema. Não só o *software* no *host* móvel tem que ser mudado. Para ser mais efetivo, o *host*



correspondente não pode permanecer inalterado. Todos os mecanismos confiam na capacidade da camada MAC para detectar futuras interrupções. Finalmente, “congelar” o estado de TCP não ajuda no caso de alguns esquemas de codificação que usam números aleatórios variantes no tempo.

Esses esquemas necessitam de resincronismo após a interrupção.

## 2.8 - RETRANSMISSÃO SELETIVA

Uma extensão muito útil do TCP é o uso de retransmissão seletiva. Confirmações TCP são acumulativas, isto é, elas reconhecem em ordem de chegada de pacotes até um certo pacote. Se um único pacote estiver perdido, o emissor tem que retransmitir tudo a partir do pacote perdido. Isto obviamente desperdiça largura de banda, não só no caso móvel, mas em qualquer rede.

O TCP pode indiretamente pedir uma retransmissão seletiva de pacotes. O receptor pode reconhecer pacotes sozinhos, não apenas trens de pacotes ou pacotes em sequência. O emissor agora pode determinar precisamente qual pacote é necessário e pode retransmiti-lo.

A vantagem desta abordagem é óbvia: um emissor retransmite somente os pacotes perdidos. Isto diminui requisitos de largura de banda e ajuda especialmente no caso de *links* sem fios lentos. O ganho em eficiência não é restrito a *links* sem fios e ambientes móveis. Usar retransmissão seletiva também é benéfico em todas as outras redes. Porém, pode haver a desvantagem de *software* mais complexo no lado do receptor, porque agora mais *buffers* são necessários para resequenciar dados e esperar que *gaps* sejam preenchidos. Mas enquanto o tamanho das memórias e as performances das CPUs permanentemente crescem, a largura de banda do meio permanece quase a mesma. Portanto, a maior complexidade não é nenhuma real desvantagem como foi nos dias anteriores ao TCP.

## 2.9 – TRANSAÇÃO ORIENTADA A TCP

Assume-se uma aplicação rodando em um *host* móvel que envia pequenos requisitos ao servidor de tempo em tempo, o qual responde com uma curta mensagem. Se a aplicação requer transporte seguro dos pacotes, ela pode usar o TCP (muitas aplicações desse tipo usam UDP e garantem segurança em uma camada superior orientada à aplicação).



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

Usar TCP agora requer vários pacotes sobre um enlace sem fio. Em primeiro lugar, o TCP utiliza três formas de confirmação para estabelecer a conexão. É preciso pelo menos um pacote para transmissão do pedido de requisição, e finalmente, o TCP exige mais três pacotes para fechar a conexão das três formas de confirmação. Considerando-se conexões com um tráfego razoável ou de longa duração, o *overhead* é mínimo. Mas em um exemplo de apenas um pacote de dados, o TCP utiliza os sete pacotes conjuntamente.

Tal *overhead* levou ao desenvolvimento do TCP orientado à conexão, o qual pode combinar pacotes para estabelecimento e liberação de conexão com pacotes de dados de usuário. Isto pode reduzir o número de pacotes para apenas dois ao invés de sete.

A vantagem evidente para certas aplicações é a redução do *overhead* utilizado no TCP padrão para configurar e liberar a conexão. Porém, o T-TCP não é mais o TCP original, assim requer mudanças no *host* móvel e em todos os *hosts* correspondentes, o que certamente é a maior desvantagem.

O próximo capítulo introduz o estudo de uma nova arquitetura para o suporte à mobilidade para aparelhos portáteis, então temos a substituição dos *hosts* aqui abordados para os aparelho celulares. Assim, será apresentada a tecnologia WAP com uma breve comparação com a tradicional Internet. O WAP introduz suas próprias camadas e protocolos capazes de dar suporte à mobilidade da telefonia celular. O que temos é uma nova arquitetura que vem solucionando os problemas com a mobilidade. O capítulo seguinte foi baseado nas referências [1], [3] e [4].



CAPÍTULO 3  
SUPORTE À MOBILIDADE

3.1 – INTRODUÇÃO

A transferência de dados entre emissores e receptores requer aplicações capazes de realizar a comunicação em uma rede útil. Em um sistema de comunicação móvel são necessárias aplicações de redes fixas, como banco de dados, arquivo, segurança, contabilidade e mecanismos de faturamento, além de um *hardware* que suporte a mobilidade do usuário.

Este capítulo dará uma visão rápida de acesso para a rede de alcance mundial (*World Wide Web* - **WWW**), tendo como foco principal os sistemas distribuídos como suporte de dispositivos móveis, apresentando seus problemas como a estreita largura de banda em enlaces sem fios e operações desconectadas.

Uma vez que usuários tornam-se cada vez mais dependentes dos serviços oferecidos via Internet, e o fato que para acessá-los é necessário que eles estejam conectados através de uma rede cabeada, não está sendo satisfatório, pois, milhões de usuários passam muito tempo em trânsito e o fato de necessitarem de um cabo para a conexão torna-se um empecilho.

Nos últimos anos, as tentativas de acabar com essas barreiras e transformar a Internet em uma plataforma de serviços sem fio não foram bem sucedidas, pois o leque de padrões era muito extenso.

Em 26 de junho de 1997, a *Ericsson*, a *Motorola*, a *Nokia* e a *Unwired Planet* (hoje *Phone.com*) iniciaram a criação de um padrão para fazer dos serviços avançados dentro do mercado sem fio uma realidade. Este novo padrão foi chamado protocolo para aplicação sem fio (*Wireless Application Protocol* – **WAP**). Em dezembro do mesmo ano, o Fórum WAP foi oficialmente criado e a primeira versão das especificações do protocolo foi disponibilizada em abril de 1998.

Este capítulo enfocará o WAP bem como suas várias camadas de comunicação para mecanismos de segurança, protocolos de transação orientados, e suporte de aplicação. O WAP combina a rede de telefonia e a Internet integrando aplicações de telefonia na *web* usando sua própria linguagem de marcação sem fio e a linguagem de *scripting*.

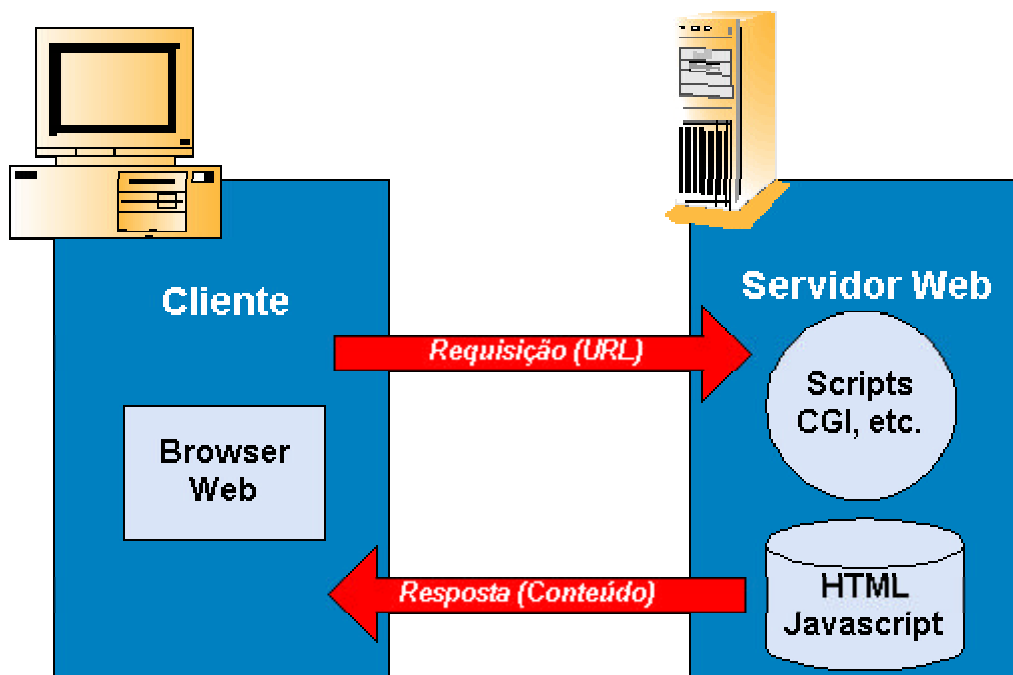


### 3.2 – O MODELO WWW

Durante os últimos anos, a Internet tem oferecido uma grande variedade de serviços que atraem os usuários, principalmente, pelo fato destes serviços serem convenientemente acessíveis pelo navegador *web*.

Os provedores de serviço também se beneficiam com o paradigma WWW, uma vez que seus serviços podem ser fornecidos para qualquer parte do mundo independente de plataforma e de sua localização. Estes serviços são criados e armazenados em um servidor, o que significa que sua alteração é muito fácil.

A arquitetura Internet provê um modelo de programação flexível e poderoso, onde aplicações e conteúdos são apresentados em formatos de dados padrões e são mostrados por aplicações conhecidas como *web browsers* ou navegadores *web*. O *web browser* é uma aplicação para rede, o que significa que ele envia requisições para objetos de dados nomeados para um servidor de rede e este responde com o dado codificado, usando os formatos padrões. Todo o processo é ilustrado na Figura 3.1





## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

Figura 3.1 – Modelo de Programação Internet

Os padrões WWW especificam a maioria dos mecanismos necessários para a construção de um ambiente de aplicação de propósito geral, incluindo:

- Modelo de nomeação padrão: todos os servidores e conteúdo na *web* são nomeados de acordo com o padrão da Internet, conhecido como localizador universal de recursos (*Universal Resource Locator* - **URL**).
- Tipificação de conteúdo: todo conteúdo na Internet é produzido de maneira que os navegadores *web* o processem corretamente, baseado em seu tipo.
- Formatos de conteúdo padrão: todos os navegadores *web* suportam um pacote de formatos de conteúdo padrões, que incluem o HTML e o *JavaScript*, entre outros.
- Protocolos de comunicação padrões: os protocolos padrões de rede, tais como o HTTP e o TCP/IP, permitem a qualquer navegador comunicar-se com qualquer servidor *web*. Estes protocolos definem três classes de servidores:
  - Servidor de origem: servidor onde um dado recurso reside ou será criado.
  - *Proxy*: um programa intermediário que age tanto como cliente quanto como servidor. Esse tipo de servidor reside entre os clientes e os servidores que não têm meios de comunicação direta.
  - *Gateway*: um servidor que age como intermediário para algum outro servidor. Diferente do *proxy*, um *gateway* recebe as requisições como se ele fosse o servidor de origem para o recurso solicitado.

Tal infra-estrutura permite aos usuários utilizarem facilmente aplicações e conteúdo de terceiros e permite também aos projetistas criarem aplicações e conteúdo para um grande número de clientes.

### 3.3 – PROTOCOLO WAP

O modelo de programação WAP é similar ao modelo de programação WWW. Isto significa que ele provê vários benefícios para a comunidade desenvolvedora de aplicações, incluindo um modelo de programação familiar e a capacidade de reutilização das ferramentas atuais, como os servidores *web*. Entretanto, otimizações e extensões foram feitas de maneira



que a característica do mundo *web* fosse ao encontro do ambiente sem fio. Sempre que possível, os padrões existentes foram plenamente adotados ou foram usados como ponto de partida para a tecnologia WAP.

O conteúdo e as aplicações WAP são especificados em um conjunto de formatos de conteúdo bastante conhecido baseado no modelo de formato de conteúdo WWW. O conteúdo é transportado usando um conjunto de protocolos padrões de comunicação baseados nos protocolos de comunicação WWW.

A figura 3.2 mostra o modelo de programação WAP. É clara a semelhança com o modelo da Internet. Sem o *gateway/proxy* os dois modelos seriam praticamente idênticos.

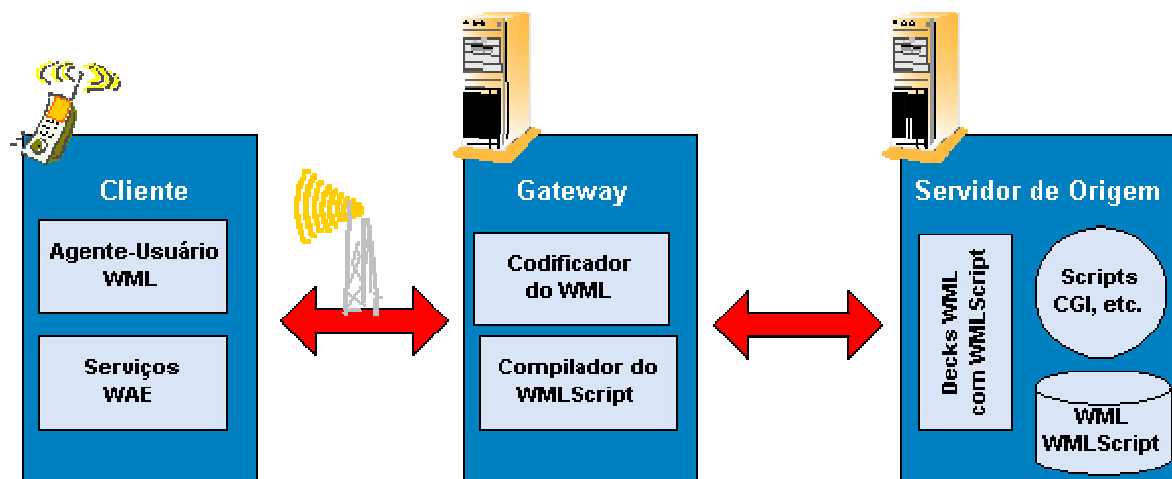


Figura 3.2 – O modelo de programação WAP

### 3.3.1 – ARQUITETURA WAP

A Figura 3.3 mostra a arquitetura WAP, seus protocolos e componentes, e compara esta arquitetura com a da Internet típica quando usa a WWW.

As bases para transmissão de dados são os diferentes serviços de portadores. O WAP não especifica os serviços de portadores, mas usa serviços de dados existentes e integrará futuros serviços. Exemplos são serviços de mensagem, como o serviço de mensagens curtas (*Short Message Service* - SMS) do GSM, os circuitos chaveados de dados, como circuitos chaveados de dados de alta velocidade (*High-Speed Circuit Switched Data* – **HSCDS**) no GSM, ou pacotes chaveados de dados, como o serviço de rádio de pacote geral (*General*

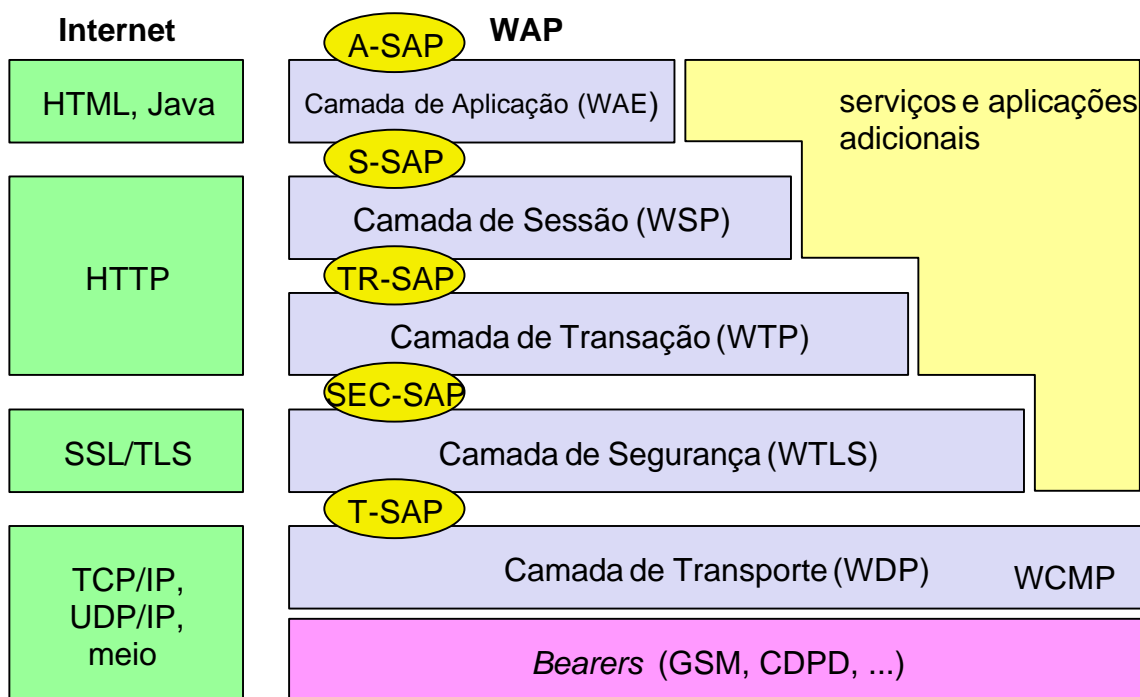




## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

*Packet Radio Service - GPRS*) no GSM. Muitos outros serviços de portadores são suportados, como CDPD, IS-136, PHS e etc. Nenhuma interface especial foi especificada entre os serviços de portadores e a próxima camada mais alta, a camada de transporte com seu protocolo de datagrama sem fio (*Wireless Datagram Protocol - WDP*) e o protocolo adicional de controle de mensagens sem fio (*Wireless Control Message Protocol - WCMP*), por causa da adaptação destes protocolos serem através de portadores específicos. A camada de transporte oferece um portador independente, um serviço orientado a datagrama consistente para as camadas mais altas da arquitetura WAP. A comunicação é feita transparentemente sobre um dos disponíveis serviços de portadores. O ponto de acesso do serviço da camada de transporte (*Transport Layer Service Access Point - T-SAP*) é uma interface comum a ser usada pelas camadas mais altas independente da rede subjacente.



WAE abrange WML (*Wireless Markup Language*), WML Script, WTAI etc.

Figura 3.3 – Componentes e interfaces da arquitetura WAP

A próxima camada de ordem mais alta, a camada de segurança com seu protocolo de segurança de camada de transporte sem fio (*Wireless Transport Layer Security - WTLS*), oferece seus serviços de segurança na SAP Segura (*Security SAP - SEC-SAP*). A WTLS é



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

baseada na segurança da camada de transporte (TLS, antiga SSL, camada de *sockets* seguro) já conhecida da WWW. Mas a WTLS foi otimizada para uso em redes sem fio com canais de faixa estreita.

A WTLS pode oferecer integridade de dados, privacidade, autenticação e proteção de serviços de “negação”. A camada de transação WAP com seu protocolo de transação sem fio (*Wireless Transaction Protocol* - **WTP**) oferece um serviço de transação simples à transação SAP (*Transaction SAP* - **TR-SAP**). Este serviço eficientemente provê requisitos seguros ou incertos e transações assíncronas. Em conjunto com esta camada está a próxima camada mais alta, se usada para serviço orientado à conexão. A camada de sessão com o protocolo de sessão sem fio (*Wireless Session Protocol* - **WSP**) oferece dois serviços à sessão-SAP (*Session Sap* - **S-SAP**), uma conexão orientada e uma não orientada se usada diretamente no topo do WDP. Um serviço especial para navegar na *web* (WSP/B) foi definido para oferecer a funcionalidade de HTTP/1.1, estado de sessão duradouro, sessão de suspensão e retorno, sessão de migração e outras características necessárias para acessos móveis sem fio na *web*.

Finalmente, no topo disto tudo, a camada de aplicação com o ambiente de aplicação sem fio (*Wireless Application Environment* - **WAE**) oferece um *framework* para a integração de diferentes aplicações de telefonia móvel e WWW.

Dessa forma ela oferece muitos protocolos em serviço com pontos de acesso de serviços especiais. As principais ferramentas usadas são linguagens de *script*, linguagem de marcação especial, interface a aplicações de telefonia e muitos formatos de conteúdos adaptados aos requisitos especiais dos dispositivos sem fio.

A figura 3.3 não apenas mostra a arquitetura WAP global mas também sua relação com a arquitetura tradicional Internet para aplicações WWW. A camada de transporte WAP junto com os portadores pode ser comparada aos serviços oferecidos por TCP ou UDP sobre IP e diferentes mídias na Internet. Se um portador na arquitetura WAP já oferece serviços IP (por exemplo, GPRS, CDPD) então o UDP é usado como WDP. Como já mencionado, as camadas TLS/SSL da Internet também foram adotadas pela arquitetura WAP com algumas mudanças necessárias para otimização. As funcionalidades das camadas de sessão e transação podem ser rigorosamente comparadas com as regras do HTTP na arquitetura *web*. Porém, o HTTP não oferece todos os mecanismos adicionais necessários para eficiente acesso móvel sem fio (por exemplo, migração de sessão, suspensão/retorno). Finalmente, a camada de aplicação oferece características semelhantes como HTML e Java. Novamente, especiais formatos e



características aperfeiçoadas para cenários sem fio têm sido definidos e acesso de telefonia tem sido adicionado.

WAP não requer que todas as aplicações usem sempre toda a arquitetura de protocolo. As aplicações podem usar só uma parte da arquitetura como mostrado na figura 3.3. Isto significa, por exemplo, que se uma aplicação não necessita segurança mas o transporte seguro de dados, pode usar um serviço da camada de transação, e não precisa da camada de segurança. Aplicações simples podem usar WDP diretamente.

Cenários diferentes são possíveis para a integração de componentes WAP a redes fixas e sem fio existentes. São mostrados na figura 3.4, no lado esquerdo, diferentes redes fixas, como a Internet tradicional e a rede pública de telefonia chaveada (*Public Switched Telephone Network* - **PSTN**). Não se pode mudar protocolos e serviços destas redes já existentes e, então, vários elementos novos serão implementados entre estas redes e os dispositivos móveis sem fios baseados em WAP em uma rede sem fio no lado direito.

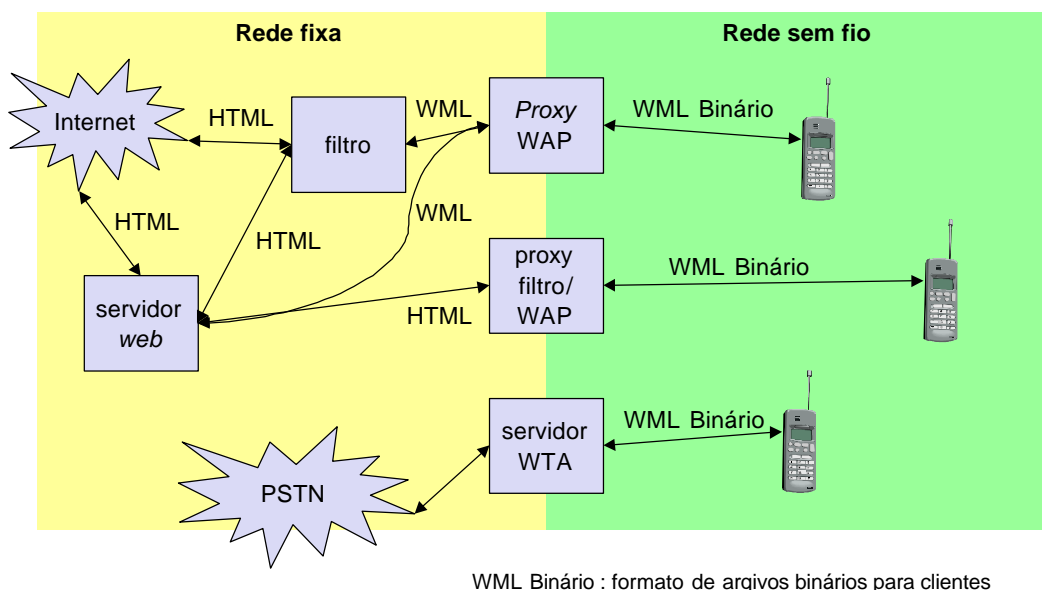


Figura 3.4 – Exemplos para integração de componentes WAP

A atual WWW na Internet oferece páginas *web* com ajuda de servidores *web* e HTML. Para poder navegar nestas páginas ou páginas adicionais com dispositivos portáteis, uma linguagem de marcação sem fio (*Wireless Markup Language* - **WML**) foi definida em WAP. Filtros especiais dentro da rede fixa podem agora traduzir HTML em WML, servidores *web* já



podem prover páginas em WML, ou os *gateways* entre as redes fixas e sem fio podem traduzir HTML em WML. Estes *gateways* não apenas filtram páginas mas atuam como *proxys* para o acesso à *web*. Para transmissões mais eficientes, WML é adicionalmente convertido em WML binário.

De um modo semelhante, um *gateway* especial será implementado para acessar serviços de telefonia tradicional por WML binário. Esses serviços de aplicação de telefonia sem fio (*Wireless Telephony Application* - **WTA**) traduzem, por exemplo, a sinalização de redes telefônicas (chamada entrante e etc.) em eventos de WML mostrados no dispositivo portátil. É importante notar a visão integrada para o cliente sem fio de todos os diferentes serviços, telefonia e *web*, pela WAE.

A Figura 3.5 mostra três representações de possíveis implementações da tecnologia, o que permite ter uma idéia prática do uso do protocolo. A pilha 1 representa um exemplo típico de uma aplicação WAP, o agente-usuário WAE, sendo executado sobre o topo da pilha WAP. Já a pilha 2, é voltada para aplicações e serviços que requerem transações com ou sem segurança. E, por fim, a pilha 3 é voltada para aplicações e serviços que somente requerem datagrama de transporte sem ou com segurança.

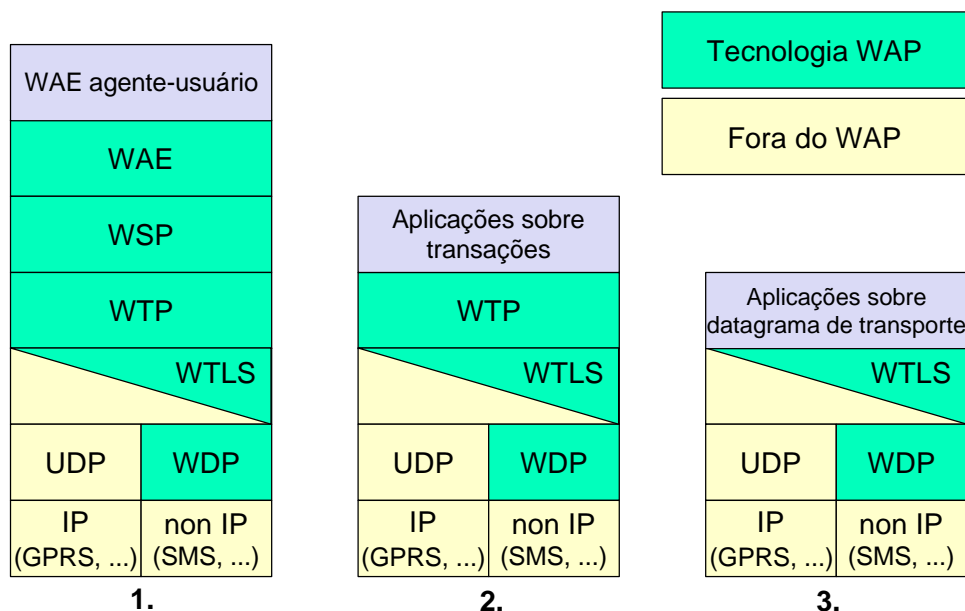


Figura 3.5 – Exemplos de pilhas WAP

### 3.4 – PROTOCOLO WDP



O WDP compõe a camada de transporte do WAP e opera acima dos serviços de transporte de dados suportados pelos vários tipos de redes. Por ser um datagrama de serviço de caráter geral, oferece um serviço consistente para os protocolos das camadas superiores (Segurança, Transação e Sessão) e se comunica de maneira transparente com um dos serviços de portadores disponíveis.

A adaptação necessária na camada de transporte para oferecer este serviço consistente pode diferir muito dependendo dos serviços dos portadores. Se o portador já oferece serviços IP, UDP é usado como WDP. Assim, WDP oferece mais ou menos os mesmos serviços que o UDP oferece.

WDP oferece os números da porta de destino e origem usados para multiplexação e demultiplexação de dados respectivamente. O serviço primitivo para enviar um datagrama é *T-DUnitdata.req* com o endereço de destino (DA), porta de destino (DP), endereço de origem (SA), porta de origem (SP), e dados de usuário (UD) como parâmetros obrigatórios (Figura 3.6). Endereço de destino e origem são endereços únicos para o receptor e o emissor dos dados de usuário. Estes poderiam ser basicamente um número de telefone, endereço IP, ou qualquer outra identificação única. O *T-DUnitdata.ind* é o serviço primitivo que indica a recepção dos dados. Aqui o endereço e a porta de destino são apenas parâmetros opcionais.

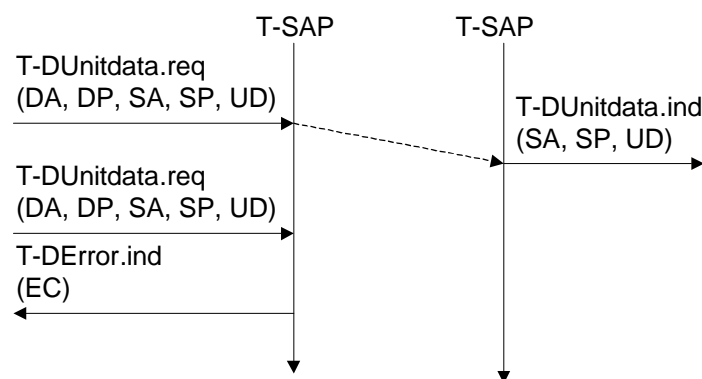


Figura 3.6 – Serviços primitivos de WDP

Se a camada superior requisita um serviço e o WDP não poder realizar, este erro é indicado com o serviço primitivo de *T-DError.ind* como mostrado na figura 3.6. Um código de erro (EC) é retornado indicando a razão do erro para a camada superior. Porém, este



**Universidade Federal do Pará**  
**Departamento de Engenharia Elétrica e de Computação**

primitivo não deve ser usado pelo WDP para indicar problemas com o serviço de portador, apenas para problemas locais, como um tamanho de dados de usuário muito grande.

Se nenhum erro acontece quando datagramas de WDP são enviados de uma entidade WDP para outra (por exemplo o destino está inalcançável, nenhuma aplicação está escutando a porta de destino especificada e etc.), o protocolo **WCMP** provê mecanismos de controle de erro para WDP e então deveria ser implementado. O WCMP contém mensagens de controle que se assemelham às mensagens do protocolo de mensagens de controle da Internet (*Internet Control Message Protocol* – **ICMP**) para IPv4 ou para IPv6 e também pode ser usado para diagnóstico e informações fins. O WCMP pode ser usado pelos nós WDP e por *gateways* de dados sem fios para reportar erros encontrados no processamento dos datagramas. Porém, mensagens de erros WCMP não devem ser enviadas como resposta para outras mensagens de erros WCMP. Em redes baseadas em IP o ICMP será usado como WCMP (por exemplo, CDPD, GPRS).

A Entidade de Gerenciamento WDP é usada como uma interface entre a camada WDP e o ambiente do dispositivo. Ela provê informação para o WDP de mudanças no ambiente dos dispositivos, o que pode impactar na correta operação do WDP.

A Entidade de Gerenciamento WDP deve monitorar o estado entre os serviços e as capacidades do ambiente móvel e deve notificar à camada WDP se um ou mais dos serviços assumidos não estão disponíveis. Por exemplo, se um celular está fora da área de cobertura de um serviço de portador (CDMA, TDMA, etc.), então a Entidade de Gerenciamento do Portador deveria reportar à Entidade de Gerenciamento WDP que a transmissão ou recepção por aquele portador não é mais possível; após isto a Entidade de Gerenciamento WDP deve indicar à camada WDP para fechar todas as conexões ativas sobre aquele portador. Outros eventos como bateria fraca também são tratados de maneira similar pela Entidade de Gerenciamento WDP.

Além de ser usada para monitorar o estado do ambiente, esta ferramenta pode ser usada como interface para ajustes de parâmetros de várias configurações usadas pelo WDP, como, por exemplo, endereço do dispositivo. Adicionalmente, também pode ser usada para implementar funções disponíveis ao usuário, como a capacidade de interromper todas as conexões.



Em resumo, a Entidade de Gerenciamento WDP trata de toda a circulação de dados relacionada a inicialização, configuração e reconfiguração dinâmica de recursos, desde que eles pertençam à camada WDP

### 3.5 – SEGURA WTLS

O protocolo da camada de segurança na arquitetura WAP provê para as camadas superiores um serviço de interface de transporte seguro, preservando o serviço de interface de transporte abaixo dele. A camada WTLS é modular, isto é, o seu uso ou não depende do nível de segurança requerido em uma dada aplicação. Ou seja, é uma camada opcional da pilha WAP. Além da segurança, o WTLS fornece uma interface para gerenciamento de conexões seguras.

A WTLS pode prover níveis diferentes de segurança (privacidade, integridade de dados, e autenticação entre duas aplicações) e foi aperfeiçoado para pequena largura de banda e alto atraso em portadores de rede. Além disso, WTLS leva em conta o baixo poder de processamento e capacidade de memória muito limitada de dispositivos móveis para algoritmos criptográficos. O WTLS provê funcionalidade similar ao TLS 1.0, além de adicionar novas funcionalidades, como suporte a datagrama, *handshake* otimizado, e atualização de chave dinâmica, e ser otimizado para redes de banda estreita com grande latência.

Antes de dados poderem ser trocados por WTLS, uma sessão segura tem que ser estabelecida. Este estabelecimento de sessão consiste em vários passos; a figura 3.7 mostra a sucessão de serviços primitivos necessário para um denominado *handshake* completo (várias otimizações são possíveis). O cliente e o servidor da sessão segura ambos podem interromper o estabelecimento da sessão a qualquer momento, por exemplo, se os parâmetros propostos não são aceitos.

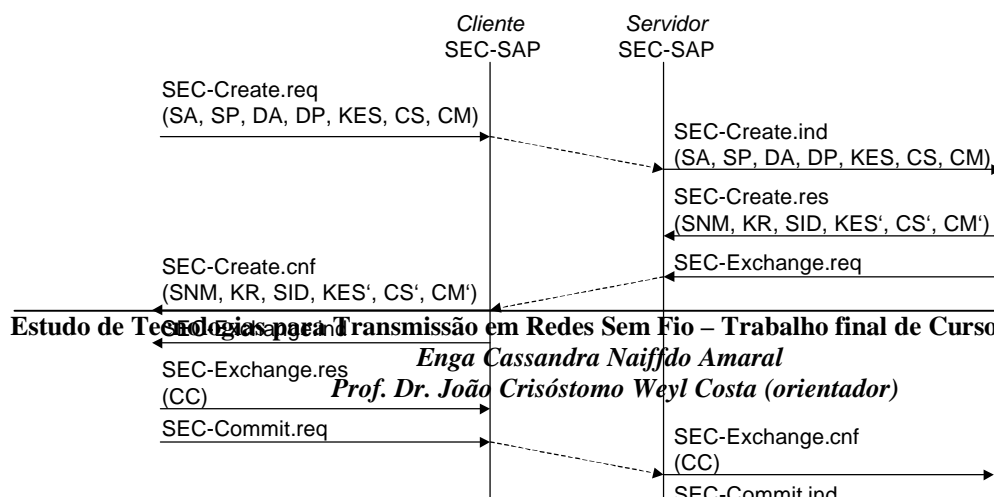






Figura 3.7 – Estabelecimento de uma sessão segura WTLS

O primeiro passo é iniciar a sessão com a primitiva *SEC-Create*. Para tanto, faz-se necessário o uso de cinco parâmetros: endereço de origem (SA), porta de origem (SP) do cliente, endereço de destino (DA), porta de destino (DP) do servidor. Além disso, o cliente propõe uma troca de chave serial (KES), o número de série (CS), e um método de compressão (CM). As respostas com parâmetros do servidor para o modo da sequência de números (SNM), a chave de ciclo de *refresh* (KR) (com que frequência chaves são atualizadas dentro desta sessão segura), o identificador de sessão (SID), e a selecionada troca de chave serial (KES), número de série (CS), método de compressão (CM). O servidor também emite uma primitiva *SEC-Exchange* (segunda troca). Isto indica que os servidores desejam executar autenticação de chave pública com o cliente, o servidor pede um certificado para o cliente.

O primeiro passo da criação de sessão segura, a negociação das serias e os parâmetros de segurança são indicados no lado do cliente, seguidos de um pedido para um certificado. O cliente responde com seu certificado e uma primitiva *SEC-Commit.req*. Esta primitiva indica que o *handshake* foi completado para o lado do cliente e que o cliente agora quer passar para o estado de conexão recentemente negociado. O certificado é entregue para o lado do servidor e o *SEC-Commit* é indicado. A camada WTLS do servidor envia de volta uma confirmação para o cliente. Isto conclui o *handshake* para uma confirmação de sessão segura.

Após configurar uma conexão segura entre os dois lados, podem ser trocados dados de usuário. Isto é feito usando uma simples primitiva *SEC-Unitdata* como mostrado na figura 3.8. A primitiva *SEC-Unitdata* tem exatamente as mesmas funções como a *T-DUnitdata* na camada WDP, isto é, transfere um datagrama entre um emissor e um receptor. Ainda esta transferência de dados é incerta, mas agora segura. Isto mostra que WTLS pode ser facilmente utilizada na pilha de protocolo no topo de WDP. As camadas superiores simplesmente usam *SEC-Unitdata* em vez de *T-DUnitdata*. Assim, os parâmetros são os mesmos: endereço de



origem (SA), porta de origem (SP), endereço de destino (DA), porta de destino (DP), e dados de usuário (UD).

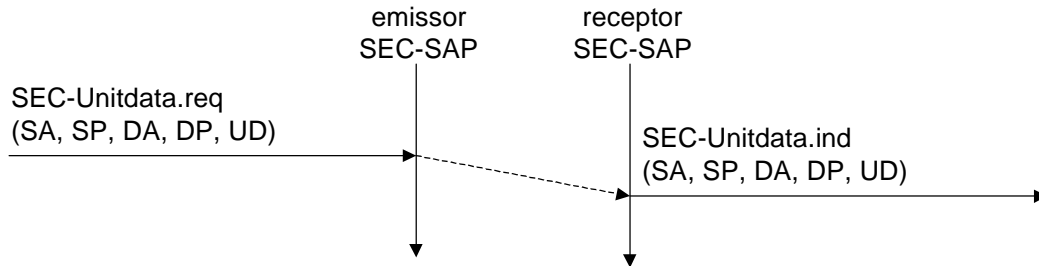


Figura 3.8 – Transferência de datagrama WTLS

Embora a WTLS permita diferentes mecanismos de codificações com diferentes tamanhos de chaves, é bastante claro que devido o poder de cálculo dos dispositivos portáteis e os regulamentos de exportação em alguns países, a codificação proporcionada não pode ser muito potente. Porém, aplicações ou usuários são livres para botar codificações mais potentes no topo da pilha inteira de protocolos se necessário - os algoritmos apropriados estão disponível mundialmente.

Trabalhos futuros na camada de WTLS incluem um suporte consistente para aplicação em nível de segurança (por exemplo, assinatura digital) e diferentes classes de implementações com diferentes capacidades a selecionar.

### 3.6 – PROTOCOLO WTP

O WTP roda no topo do serviço de datagrama e, opcionalmente, de um serviço de segurança e foi definido como um protocolo orientado à transação leve que é adequado para implementações em clientes com poucos recursos (estações móveis), além de operar eficientemente sobre redes de datagrama sem fio. Em vista disso, o WTP oferece benefícios como a melhoria da segurança sobre os serviços de datagrama, o que libera a camada superior de retransmissões e reconhecimentos necessários quando os serviços de datagrama são



usados. Outro benefício é a melhora da eficiência dos serviços orientados à conexão, uma vez que o WTP não tem nenhuma fase de criação e destruição de conexão explícita. Por último, o WTP é orientado a mensagens e é projetado para serviços orientados a transações, como navegação.

O WTP oferece muitas características a camadas superiores. As bases são três classes de serviços de transação. A classe 0 que provê transferência de mensagem não confiável sem nenhuma mensagem de resultado. As classes 1 e 2 que provêm transferência de mensagem segura, a classe 1 sem mensagem de resultado, a classe 2 com somente uma mensagem de resultado confiável (o caso típico de pedido/resposta). O WTP atinge confiabilidade através do uso de identificadores únicos de transação, retransmissão, confirmações e remoção de duplicação. Nenhuma classe WTP requer nenhuma fase de conexão de configuração ou desconfiguração. Isto evita o *overhead* desnecessário em *links* de comunicação. O WTP permite transações assíncronas, aborto de transações, concatenação de mensagens, e pode informar sucesso ou fracasso de mensagens seguras (por exemplo, um servidor não pode controlar os pedidos).

Para ser consistente com a especificação, no que segue o termo iniciador é usado para uma entidade WTP iniciando uma transação, e o termo respondedor para a entidade WTP respondendo a uma transação (no caso servidor). Os três serviços primitivos oferecidos por WTP são *TR-invoke* para iniciar uma nova transação, *TR-resul* para enviar de volta o resultado de uma transação previamente iniciada, e *TR-abort* para abortar uma transação existente. Os PDUs trocados entre duas entidades WTP para transações normais são o *invoke* PDU, *ack* PDU, e *result* PDU.

Uma característica especial de WTP é a habilidade de prover uma confirmação do usuário ou, alternativamente, uma confirmação automática pela entidade WTP. Se a confirmação de usuário é requerida, um usuário de WTP tem que confirmar toda mensagem recebida por uma entidade de WTP. Uma confirmação de usuário provê uma versão mais potente de um serviço confirmado para garantir que a resposta veio do usuário do WTP e não da própria entidade WTP.

### 3.6.1 CLASSES WTP



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

As transações de classe 0 provêm um serviço de datagrama não confiável e podem ser usadas por aplicações que requerem um serviço de “*push* não confiável”. Esta classe não objetiva melhorar o serviço de transação com a capacidade de aplicação usando o WTP para, ocasionalmente, enviar um datagrama dentro do mesmo contexto de uma sessão existente usando WTP. Aplicações que requerem um serviço de datagrama como recurso principal para entrega de dados devem usar o WDP e não o WTP.

Já as transações de classe 1 provêm um serviço de datagrama confiável e podem ser usadas por aplicações que requerem um serviço de “*push* confiável”. Em transações desse tipo, uma mensagem de execução é enviada do Iniciador para o Respondedor, que confirma a mensagem de execução e mantém a informação de estado por algum tempo, mesmo depois que a confirmação foi enviada, para tratar de possível retransmissão de confirmação, caso esta se perca e/ou o Iniciador retransmita a mensagem de execução. No Iniciador, a transação termina quando a mensagem de execução é recebida; isto significa que a transação pode ser abortada a qualquer momento.

As transações de classe 2 oferecem o serviço básico de pedido/resposta. Uma sessão WSP pode consistir de uma série de transações deste tipo. Uma mensagem de execução é enviada do Iniciador para o Respondedor, que responde com uma mensagem de resultado, confirmando explicitamente a mensagem de execução. Se o Respondedor demorar na tarefa de execução mais tempo do que o seu intervalo de temporização, ele pode responder com uma mensagem de “aguarde” antes de enviar a de resultado. Este procedimento evita a retransmissão desnecessária da mensagem de execução. O Respondedor envia a mensagem de resultado de volta ao Iniciador, que é confirmada por este. O Iniciador mantém a informação de estado por algum tempo mesmo depois que a confirmação foi enviada para tratar de possível retransmissão de confirmação, caso esta se perca e/ou o Respondedor retransmita a mensagem de resultado. No Respondedor, a transação termina quando a confirmação é recebida; isto significa que a transação pode ser abortada a qualquer momento.

### 3.7 – PROTOCOLO WSP



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

O WSP provê meios para a troca organizada de conteúdo entre aplicações cooperativas cliente/servidor. Especificamente, ele fornece às aplicações meios para estabelecer uma sessão confiável do cliente ao servidor e liberar esta sessão de maneira ordenada e, concordarem com um nível comum da funcionalidade do protocolo usando capacidade de negociação, bem como trocarem conteúdo entre cliente e servidor usando codificação compacta, além de fornecer meios para suspender e retomar a sessão.

Para ambos os tipos, a segurança pode ser inserida usando a camada de segurança WTLS se preciso for. Basicamente, WSP provê um estado compartilhado entre um cliente e um servidor para aperfeiçoar a transferência de conteúdo. Um protocolo WSP tenta substituir dentro do domínio sem fio, o HTTP, o qual já causa muitos problemas em redes fixas. Muitos provedores de conteúdo *web* usam “*cookies*” para guardar algum estado em uma máquina de cliente o que não é uma solução elegante. Os estados são necessários em navegadores *web*, por exemplo, retomar a navegação exatamente no mesmo contexto no qual o navegador foi suspenso. Isto é uma importante característica para clientes e servidores. Os clientes podem continuar trabalhando onde eles deixaram o navegador ou quando a rede foi suspensa, ou os usuários podem adquirir seus ambientes personalizados toda vez que eles iniciarem o navegador. Os provedores de conteúdo podem personalizar suas páginas às necessidades dos clientes e não têm que retransmitir as mesmas páginas inúmeras vezes. O WSP oferece as características gerais necessárias para o intercâmbio de conteúdo entre clientes e servidores:

- Gerenciamento de sessão: o WSP introduz sessões que podem ser estabelecidas de um cliente para um servidor e podem ter uma longa duração. Também podem ser lançadas de uma maneira ordenada. Importantes para aplicações móveis são as capacidades de suspender e retomar uma sessão. Por exemplo, um dispositivo móvel que foi desligado - seria útil para um usuário poder continuar a operação exatamente no ponto onde o dispositivo foi desligado. O tempo de vida da sessão é independente do tempo de vida da conexão de transporte ou operações contínuas de uma rede de portadores.

- Capacidade de negociação: clientes e servidores podem concordar com o nível comum de funcionalidade de protocolo durante estabelecimento da sessão. Exemplo de parâmetros para negociação são: máximo tamanho do cliente SDU, máximo pedidos, opções de protocolo, tamanho de servidor SDU.



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

- Conteúdo codificado: o WSP também define um eficiente codificador binário para o conteúdo que ele transfere. O WSP oferece conteúdo digitado e objetos compostos.

Enquanto WSP é um protocolo de sessão de propósito geral, o WAP tem um navegador/protocolo de sessão sem fio específico (WSP/B), que inclui quase todos os serviços e protocolos para aplicações do tipo navegação. Além das características gerais de WSP, WSP/B oferece as características adaptadas para navegadores *web*:

- Funcionalidade HTTP/1.1: WSP/B suporta as funções HTTP/1.1 oferecidas, como extensivos métodos de pedido/resposta, objetos compostos e negociação do tipo do conteúdo. Basicamente, WSP/B é uma forma binária de HTTP/1.1. Assim, o cabeçalho do HTTP/1.1 é usado para definir o tipo do conteúdo, as características da codificação, linguagens e etc, mas codificadores binários são definidos para cabeçalhos bem conhecidos para reduzir o *overhead* de protocolos.

- Troca de cabeçalhos de sessão: clientes e servidores podem trocar cabeçalhos de pedido/resposta que permaneçam constante durante a sessão. Estes cabeçalhos devem incluir tipos de conteúdos, conjuntos de caracteres, linguagem, capacidades do dispositivo, e outros parâmetros estáticos. O WSP/B não interpretará informações de cabeçalho mas passará todo o cabeçalho diretamente para serviços do usuários.

- Transferência *push* e *pull* de dados: *Pulling* dados de um servidor é o mecanismo tradicional da *web*. Isto também é suportado pelo WSP/B usando o mecanismo de pedido/resposta de HTTP/1.1. Adicionalmente, WSP/B suporta três mecanismos de *push* para transferência de dados: um *push* de dados confirmados dentro de um contexto de sessão existente, um *push* de dados não confirmados dentro de um contexto de sessão existente, e um *push* de dados não confirmados fora do contexto de sessão existente.

- Requisições assíncronas: Opcionalmente, o WSP/B suporta um cliente que pode enviar múltiplos pedidos para um servidor simultaneamente. Isto melhora a eficiência dos pedidos e as respostas podem agora ser fundidas em mensagens menores. A latência também é melhorada, pois cada resultado pode ser enviado ao cliente assim que esteja disponível.

O WSP/B pode rodar sobre um serviço de transação WTP ou em um serviço de datagrama WDP.



### 3.8 – AMBIENTE WAE

O WAE é um ambiente de aplicação de propósito geral baseado, fundamentalmente, nas tecnologias e filosofias *web* e tem como objetivo primário estabelecer um ambiente interoperável que permita a operadores e provedores de serviços construírem aplicações e serviços que possam atingir uma grande variedade de plataformas sem fio de maneira útil e eficiente. Porém, o WAE não assume nenhum modelo específico da interface homem-máquina, mas permite uma variedade de dispositivos, cada um com suas próprias capacidades e provavelmente fabricantes específicos. O WAE já integrou tecnologias de HTML, *JavaScript* e os dispositivos portáteis de marcação da linguagem HDML que forma a base da linguagem de marcação de redes sem fio (WML) e a linguagem de *scripting* WMLScript e as adaptou para uso em um ambiente sem fio com dispositivos portáteis de baixa potência. A troca de formato para cartões de negócio e listas telefônicas vCard e para calendários vCalendar foram incluídos. As URLs conhecidas da *web* podem ser usadas. Além disso, um grande fluxo de tecnologias de telecomunicações móveis vem sendo adotadas e integradas na aplicação de telefonia sem fio (WTA).

Além de depender de uma tecnologia madura instalada, o WAE tem um foco em dispositivos com capacidades muito limitadas, ambientes de faixa estreita, e características de segurança especial e controle de acesso. Uma meta global do WAE é minimizar o tráfego no meio e consumo de recurso no dispositivo portátil. Esta meta é refletida no modelo lógico WAE subjacente (Figura 3.9). O WAE adota um modelo que de perto segue o modelo WWW, mas assume *gateways* adicionais que podem aumentar a eficiência de transmissão. Um cliente emite um pedido codificado para uma operação em um servidor remoto. Codificar é necessário para minimizar dados enviados sobre o meio e economizar recursos nos dispositivos portáteis.

Decodificadores em um *gateway* agora traduzem estes pedidos codificados em um pedido padrão como entendido pelos servidores de origem. Este poderia ser um pedido para adquirir uma página *web* ou um pedido para montar uma chamada. O *gateway* transfere este pedido para um apropriado servidor de origem como se viesse de um cliente padrão. Servidores de origem poderiam ser servidores *web* padrão rodando HTTP e gerando





**Universidade Federal do Pará**  
**Departamento de Engenharia Elétrica e de Computação**

conteúdos usando *scripts*, provendo páginas usando um banco de dados, ou aplicando qualquer outra tecnologia. O WAE não especifica qualquer gerador ou servidor de conteúdo padrão, mas assume que a maioria seguirá a tecnologia padrão usada na WWW de hoje.

Os servidores de origem irão responder ao pedido. O *gateway* agora codifica a resposta e seu conteúdo (se houver) e transfere a resposta codificada com o conteúdo para o cliente. O modelo lógico do WAE não apenas inclui esse esquema de pedido/resposta padrão, mas também serviços de *push*. Então um servidor de origem envia conteúdo para o *gateway*. O *gateway* codifica o conteúdo enviado e transmite ao cliente.

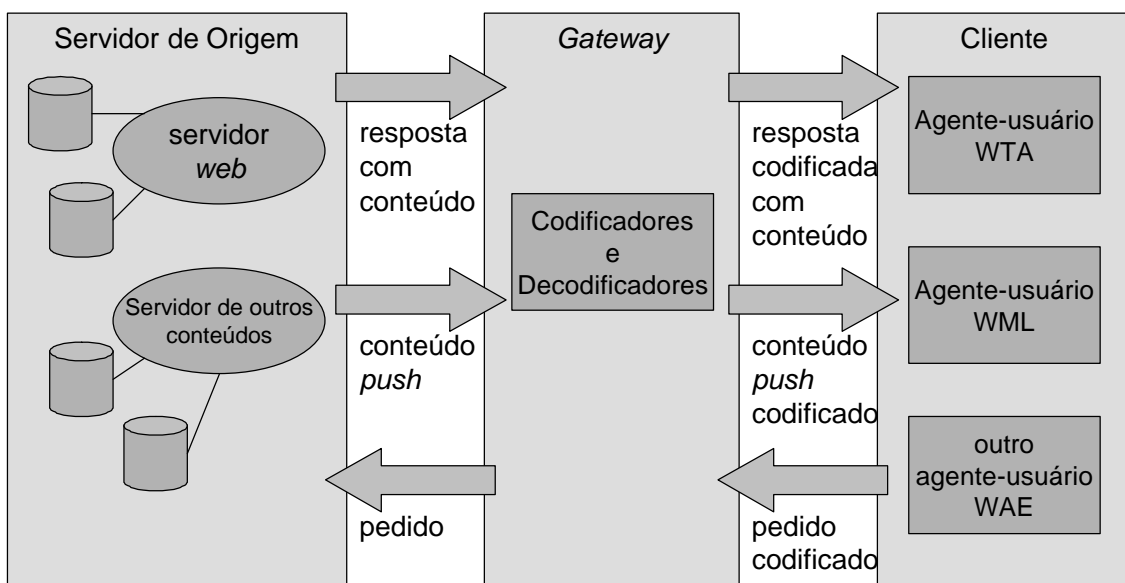


Figura 3.9 – Modelo lógico WAE

Dentro de um cliente podem residir vários agentes-usuário. Agentes-usuário incluem tais itens como navegadores, agendas telefônicas, editores de mensagem e etc. O WAE não especifica nenhum número de agentes-usuário nem suas funcionalidade, mas assume um agente-usuário WML básico que suporta WML, WMLscript, ou ambos. Isto é um navegador WML. Porém, agente-usuário de domínio específico com arquitetura variante pode ser implementado. Este agente-usuário controla acesso para e interação com características de telefones móveis (como controle de chamada).

### 3.9 – LINGUAGEM WML



A WML é uma linguagem de marcação baseada no XML (*Extensible Markup Language*) e objetiva ser usada na especificação de conteúdo e interface para dispositivos de banda estreita, incluindo telefones celulares e *paggers*. A WML e seu ambiente de suporte foram projetados tendo em vista as restrições dos dispositivos de banda estreita, o que inclui pequenos *displays*, facilidades de entrada do usuário limitadas, conexões de rede de banda estreita e recursos de memória e de processamento limitados.

A linguagem é baseada em um subconjunto do HDML versão 2.0, entretanto a WML mudou alguns elementos adotados do HDML e introduziu novos elementos, alguns dos quais foram modelados de acordo com elementos similares do HTML.

A WML introduziu a metáfora de *card* e *deck*, o que contém conceitos de que à interação com o usuário é descrita em um conjunto de *cards*, os quais podem ser agrupados em um documento, que normalmente é referido por *deck*. O usuário navega através de um conjunto de *cards* WML, onde pode ver o conteúdo de cada um, pode entrar com informações requisitadas, pode fazer escolhas e pode se mover de um *card* para outro. As instruções contidas nos *cards* podem invocar serviços no servidor, se for necessário interação de maneira particular, e os *decks* são requisitados ao servidor conforme sejam necessários. Os *decks* WML podem ser armazenados em arquivos estáticos ou podem ser gerados dinamicamente por geradores de conteúdo como, por exemplo, *scripts*.

A WML não especifica como a implementação de um navegador WML tem que interagir com um usuário. Ao invés disso, a WML descreve a intenção de interação de uma maneira abstrata. O agente-usuário de um dispositivo portátil tem que decidir como melhor apresentar todos os elementos de um *card*. Esta apresentação depende muito da capacidade do dispositivo.

A WML inclui várias características básicas:

- Suporte a textos e imagens: A WML provê aos autores de conteúdo várias maneiras de especificar os textos e as imagens que serão mostradas aos usuários, incluindo sugestões de *layout* e apresentação. Como todas as outras linguagens de marcação, o WML requer que o autor especifique a apresentação em termos gerais e dá ao agente-usuário uma grande liberdade para determinar como ele deve apresentar a informação ao usuário final.
- Suporte à entrada de dados do usuário: A WML possui um pequeno conjunto de controles de entrada e suporta validação de dados no lado do cliente, pois permite ao autor



chamar *scripts* em momentos apropriados para checar a entrada do usuário. A linguagem inclui também controles de invocação de tarefas que quando ativados iniciam uma tarefa de navegação ou uma tarefa de gerenciamento de história, como passar de um *link* para outro *card* ou *script* ou mesmo retirar o *card* atual do *deck* de história. O agente-usuário é livre para escolher como irá apresentar estes controles. Ele pode, por exemplo, ligá-los às teclas do dispositivo ou a comandos de voz.

- **Navegação:** A WML permite vários mecanismos de navegação usando URL, incluindo o uso de *hyperlinks* semelhante ao HTML e elementos de navegação entre *cards*, assim como elementos de navegação na história.

- **Gerenciamento de contexto:** A WML permite salvar o estado entre diferentes *decks* sem interação de servidor, isto é, estados variados podem durar mais tempo que um único *deck*, e assim o estado pode ser compartilhado por diferentes *decks*. *Cards* podem ter parâmetros definidos pelo uso deste estado sem acessar o servidor sobre um canal sem fio de faixa estreita.

A WML pode ser codificada usando uma representação binária compacta para economizar largura de banda na ligação sem fio. Esta representação compacta está baseada no formato de conteúdo XML binário como especificado no Fórum WAP. A codificação binária WML é só uma versão especial deste formato, a representação compacta é em geral válida para conteúdo de XML. O formato compacto permite transmissão sem perda de funcionalidade ou de informação semântica. Por exemplo, o prefixo URL `http://` o qual é muito comum em URLs será codificado como 4B. Outros comandos próprios do WML são codificados e estes simples códigos de byte são muito mais eficientes que o texto claro usado em HTML.

### 3.10 – WMLScript

A WMLScript é uma linguagem de *script* simples, que realça as facilidades de navegação e apresentação do WML, além de possuir capacidades de: suporte a comportamento da interface do usuário (*User Interface* – **UI**) de modo mais avançado, adicionar inteligência ao cliente, prover um mecanismo conveniente de acesso aos dispositivos e seus periféricos e reduzir a necessidade de idas e voltas ao servidor. A linguagem é fracamente baseada no *ECMAScript*, que é um subconjunto da linguagem de



*script* da Internet - *JavaScript*, entretanto é refinada para dispositivos de banda estreita. Enquanto todo o conteúdo WML é estático, WMLScript oferece várias capacidades não suportadas por WML:

- Cheque de validade da entrada do usuário: antes da entrada do usuário ser enviada para o servidor, o WMLScript pode conferir a validade e economizar largura de banda e latência em caso de um erro. Caso contrário, o servidor tem que executar todos os cheques que sempre incluem pelo menos um tempo de ida-e-volta se problemas acontecerem.
- Acesso para facilidades de dispositivo: WMLScript oferece funções para acessar componentes de hardware e funções de software do dispositivo. Em um telefone um usuário poderia, por exemplo, fazer uma chamada telefônica, acessar a agenda telefônica, ou enviar uma mensagem através de um serviço de mensagem do telefone móvel.
- Interação de usuário local: Sem introduzir atrasos de ida-e-volta WMLScript pode diretamente e localmente interagir com um usuário, mostrar mensagens ou o *prompt* para entrada. Por exemplo, somente o resultado de várias interações poderia ser transmitido para um servidor.
- Extensões para o software do dispositivo: Com a ajuda do WMLScript um dispositivo pode ser configurado e novas funcionalidades podem ser adicionadas até mesmo depois de desenvolvidas. Os usuários podem receber novos *softwares* de fabricantes e, assim, atualizar seus dispositivos facilmente.

A WMLScript está baseada em *JavaScript*, mas adaptada para ambientes sem fios. O WAE inclui um conjunto de formatos do conteúdo, que facilitam a troca de dados, entretanto o método de troca irá depender dos dados e do agente-usuário alvo do WAE. Os dois formatos mais importantes na camada são os de *bytecodes* codificados do WML e do WMLScript, o que torna a transmissão destes mais eficiente e minimiza a necessidade de capacidade computacional do cliente. Além destes, o WAE adota outros formatos de dados como imagens, mensagem e formatos específicos do agente-usuário.

Os tipos de dados suportados pelo WMLScript são lógico, inteiro, ponto flutuante, cadeia de caracteres e inválido, e a própria linguagem tenta fazer a conversão automaticamente entre os diferentes tipos. O suporte ao tipo ponto flutuante varia de acordo com as capacidades do dispositivo alvo.



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

A linguagem suporta também várias categorias de operações, como: atribuições, aritméticas, lógicas e comparações, e vários tipos de funções, como locais, externas e de bibliotecas, além de definir uma série de bibliotecas incluindo uma para linguagem (*Lang*), uma para cadeia de caracteres (*String*), uma para navegador (*WMLBrowser*), uma para ponto flutuante (*Float*) e uma para diálogo (*Dialogs*).

#### 3.11 – APLICAÇÃO WTA

Navegar na *web* usando o navegador WML é apenas uma aplicação para dispositivos portáteis do usuário. Adicionalmente, um usuário ainda quer fazer ligações telefônicas e acessar todas as características da rede de telefonia móvel como com um telefone móvel tradicional. Isto é onde a aplicação de telefonia sem fio (*Wireless Telephony Application - WTA*), o agente-usuário WTA (como mostrado na figura 3.9), e a interface de aplicação de telefonia sem fio (*Wireless Telephony Application Interface - WTAI*) entram. A WTA é uma coleção com extensões específicas de telefonia para chamadas e características de mecanismos de controle, fundindo redes de dados e redes de voz.

O *framework* WTA integra serviços de telefonia avançada usando uma interface de usuário consistente (por exemplo, um navegador WML) e permite que os operadores de rede aumentem a acessibilidade para vários serviços especiais em suas redes. Além disso, um operador de rede pode alcançar mais dispositivos fins usando WTA porque este é integrado no WAE o qual controla ambientes e características específicas de dispositivos. Mas o WTA deveria também habilitar desenvolvedores bem como operadores de rede a criar conteúdo de rede-independente que acesse as características básicas de um portador de rede. Porém, a maioria das funcionalidades WTA está reservada para operadores de rede por segurança e razões de estabilidade.

A WTA estende o modelo de aplicação WAE básico de três modos:

- Conteúdo *push*: Um servidor original WTA pode enviar conteúdos, isto é, *decks* WML, WMLScript para o cliente. Este conteúdo pode, por exemplo, habilitar o cliente a controlar novos eventos de rede que eram desconhecidos antes. Um exemplo é dado na figura 3.10.

- Controle de eventos de rede: Um dispositivo pode ter uma tabela indicando como reagir a certos eventos de uma rede móvel. Eventos podem ser uma chamada entrante ou uma



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

mensagem de texto. O dispositivo pode observar como reagir, por exemplo, observar em uma lista de telefones privada para mapear o número do telefone entrante para um nome.

- Acesso para funções de telefonia: Aplicações rodando no cliente podem acessar funções de telefonia da WML ou WMLScript de uma forma muito simples. Muitas funções são disponíveis em bibliotecas para configurar chamadas, fazer entradas em agendas telefônicas e etc.

Estas bibliotecas têm sido definidas na especificação de WTAI segundo o Fórum WAP, e permite a criação de aplicações de telefonia usando o agente-usuário WTA. Funções de bibliotecas podem ser usadas na pilha WML ou WMLScript. O agente-usuário WTA exibe *cards* de um *deck* ou executa um script, e recebe eventos de uma rede móvel ou conteúdo enviado do servidor de origem WTA. Três classes de bibliotecas foram definidas:

- Serviços de redes comuns: Esta classe contém bibliotecas para serviços comuns em todas as redes móveis. A biblioteca de controle de chamadas contém, por exemplo, funções para configurar chamadas, aceitar chamadas, e liberar chamadas. Redes de textos contém funções para enviar, ler, e excluir mensagens de texto. Agendas telefônicas permite a manipulação das entradas de agendas telefônicas locais (por exemplo, ler, escrever e excluir). Finalmente, a biblioteca contém miscelâneas, por exemplo, uma função para indicar dados entrantes, e-mail, fax, ou mensagens de voz.

- Rede de serviços específicos: Bibliotecas nesta classe dependem da capacidade da rede móvel. Adicionalmente, esta classe deve conter bibliotecas de operadores específicos.

- Serviços Públicos: Esta classe contém bibliotecas com funções publicamente disponíveis, isto é, funções que os provedores podem usar, não apenas operadores de rede. Um exemplo é 'fazer ligação' para montar uma chamada.

A arquitetura WTA inclui um elaborado mecanismo de controle de eventos. Os clientes e os agentes-usuários WTA devem ser capazes de controlar eventos baseados em telefonia que são em tempo real. Isto significa que uma reação tem que ocorrer dentro de um tempo rígido. Há duas formas de eventos que podem localizar um cliente. Ou o sistema operacional do dispositivo móvel detecta o evento ou o servidor WTA envia um evento ao cliente. O primeiro caso denota um evento tradicional que chega no dispositivo através de uma rede telefônica móvel que é então convertida em um evento WTA. O segundo evento vem de um servidor WTA usando o serviço *push WSP*.



## Universidade Federal do Pará

### Departamento de Engenharia Elétrica e de Computação

Não importa de onde o evento veio, o agente-usuário tem que primeiro emparelhar este evento para qualquer ligação de evento WML existente no contexto atual. Se há qualquer ligação com aquele evento, a URL indicada deve ser carregada. Isto significa que o evento tem apenas ativado a carga de um novo *card*. Se não houver tal evento WML ligado, o evento tem que ser testado em uma tabela de eventos local. Esta tabela deve indicar uma nova URL a qual tem que ser carregada no caso deste evento. Então o agente-usuário WTA carrega esta nova URL. Isto mostra que os eventos são sempre mapeados em URLs.

Se eventos são enviados para um servidor WTA, este servidor pode enviar conteúdo adicional que mostra como controlar este evento. Se apenas este método é usado, isto é, todos os eventos vêm de um servidor WTA e a rede móvel também envia seus eventos primeiro ao servidor, então eventos WTA são usados no então chamado modo *server-centric*. Se o cliente controla eventos de uma rede móvel, enquanto o servidor provê apenas, por exemplo, atualizações em ligações de evento, eventos WTA são usados no então chamado *server-centric*.

Esses dois modos extremos podem ser combinados agora em uma arquitetura que permite um operador escolher o modo que se ajusta melhor para cada evento.

Uma visão da arquitetura lógica WTA é dada na figura 3.10

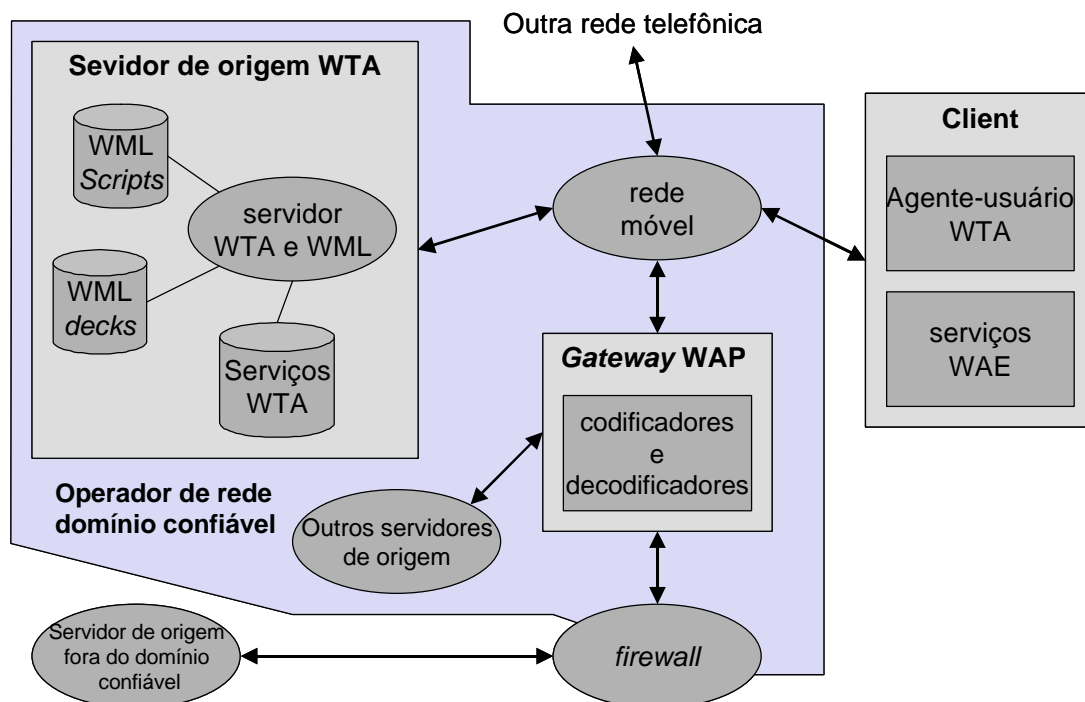






Figura 3.10 – Arquitetura lógica WTA

Os componentes mostrados não são todos obrigatórios nesta arquitetura, porém, *firewalls* ou outros servidores de origem podem ser úteis. O cliente está conectado através de uma rede móvel com um servidor de origem WTA, outras redes de telefone (por exemplo, a fixa PSTN), e um *gateway* WAP. Tem-se ainda um agente-usuário WML rodando no cliente ou outros agentes-usuários. O cliente pode ter conexões de voz e dados sobre uma rede móvel. Podem ser conectados outros servidores de origem dentro de domínios confiáveis através de um *gateway* WAP. Um *firewall* é útil para conectar servidores de origem fora do domínio confiável.

Uma diferença entre servidores de origem WTA e outros servidores além de seguranças é o controle mais restrito de QoS (*Quality of Service*). Um operador de rede sabe exatamente a latência, confiabilidade, e capacidade de sua rede móvel e, assim, pode ter mais controle sobre o comportamento dos serviços. Outros servidores, provavelmente localizados na Internet, podem não ser capazes de dar boas garantias de QoS como os operadores de rede. Similarmente, o agente usuário WTA tem uma gerência de contexto muito rígida e em tempo real comparado com o padrão de agente-usuário WML usados por navegadores da *web*.

A figura 3.11 mostra uma interação exemplar entre um cliente WTA, um servidor WTA, a rede móvel (com provavelmente muitos mais servidores) e um servidor de caixa de voz. Alguém poderia deixar uma mensagem em um servidor de caixa de voz como indicado. Isto ativa uma mensagem do servidor de caixa de voz para o servidor WTA que significa que uma nova mensagem acabou de chegar. O servidor de WTA pode então dinamicamente gerar um novo *deck* contendo, por exemplo, referências para todas as mensagens de voz atualmente armazenadas no servidor de caixa de voz. Ele então envia este *deck* ao cliente WTA.

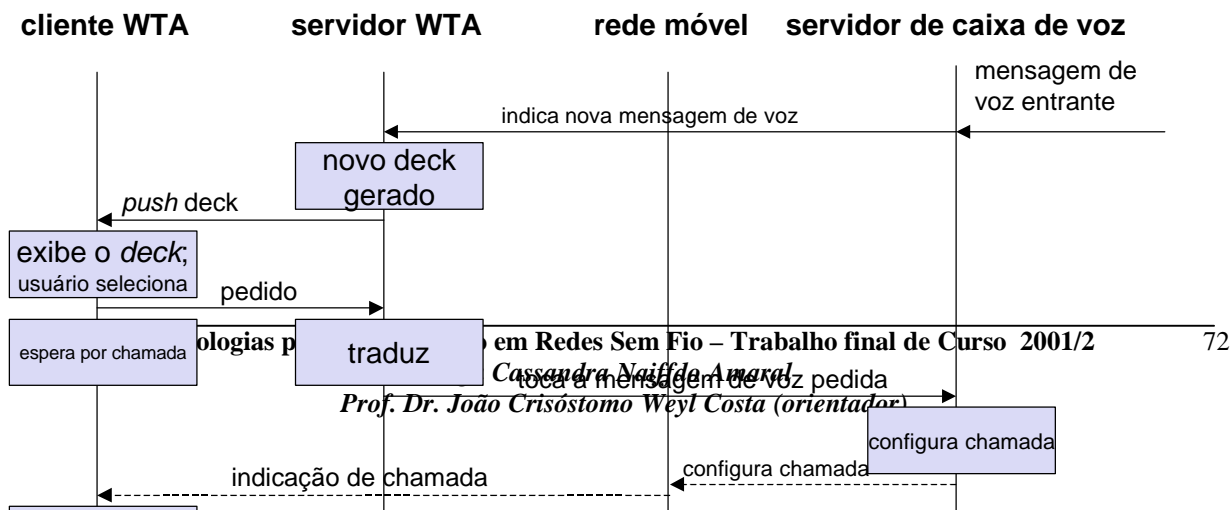




Figura 3.11 – Exemplo WTA: Mensagem de voz

O cliente WTA exibe o novo *deck* e o usuário pode escolher qual mensagem de voz quer escutar. Este pedido é mandado de volta ao servidor WTA e o cliente WTA exibe um *card* novo ao usuário, por exemplo, mostrando uma mensagem como 'por favor escute'. O servidor de WTA traduz o pedido para uma mensagem de voz específica em um pedido para o servidor de caixa de voz tocar a mensagem de voz selecionada. Este pedido está agora em um formato satisfatório para o servidor de caixa de voz.

O servidor de caixa de voz então configura a chamada para o cliente WTA usando procedimentos de configuração de chamada padrão da rede de telefonia móvel.

A rede indica a chamada ao cliente e a chamada é respondida automaticamente sem interação do usuário desde que o cliente esteja preparado para este evento. A chamada é aceita e finalmente uma conexão de voz é estabelecida entre o servidor de caixa de voz e o cliente WTA. As linhas tracejadas indicam a interação da rede móvel padrão não específico para WTA.

Outro exemplo seria a integração de WTA, chamada a biblioteca WTAI, WML e WMLScript, e como os autores podem usar as funções de WTA. Imagina-se um cliente assistindo um espetáculo na televisão. No término o cliente pode votar em seu campeão. O método tradicional é que cada candidato adquira um número de telefone associado e o cliente tem que discar o número – se o número for muito grande fica propenso a erros. Usando WAP com WTA, o operador da rede pode enviar um *deck* com vários *cards* para o dispositivo portátil do cliente e assim pode apresentar uma escolha simples para votar.



## CONCLUSÃO

Neste trabalho foram apresentadas três tecnologias para transmissão em redes sem fio, assim, tem-se o suporte à mobilidade na camada de rede sendo de suma importância devido à camada de rede unir a grande Internet com o protocolo IP comum, o suporte à mobilidade na camada de transporte, apresentando os mecanismos do TCP tornando-o um protocolo acessível em redes sem fio e enfim suporte à mobilidade para dispositivos portáteis, onde se abordou uma nova tecnologia conhecida como o WAP.

Atendendo as necessidades do mundo globalizado, tem-se a Internet evoluindo a passos largos e seus usuários cada vez mais ligados a esta grande rede mundial, esta dependência levou aos usuários exigirem cada vez mais de suas aplicações e de toda facilidade e eficiência que esta rede possa oferecer. Então usuários não queriam ter apenas um ponto fixo para acessá-la; o acesso ao mundo não era mais satisfatório visto que o usuário não poderia disponibilizar desse acesso o tempo inteiro, então o usuário exigiu o acesso à internet em seus *laptops* e *notebooks*, o que levou ao aprimoramento dos protocolos móveis como o IP Móvel e o TCP Móvel.

Como uma saída imediata às exigências dos usuários esses protocolos estão sendo satisfatórios, entretanto há várias otimizações a serem feitas como por exemplo, o IP móvel apresenta um ponto ainda sem resolução quando se trata de segurança e de eficiência do fluxo de pacotes, e do suporte a qualidade dos serviços, não muito diferente tem-se o TCP com vários problemas de eficiência, assumindo o congestionamento na rede se confirmações não chegarem e principalmente não evitando de forma satisfatória o problema da perda de pacotes causada pela mobilidade.

Não bastando essas dificuldades nas redes sem fio, os usuários cada vez mais exigentes pedem agora pelo acesso a rede Internet através de seus aparelhos portáteis, ou seja, não se trata mais de apenas redes sem fio agora a evolução alcança as redes móveis celulares. Então uma nova tecnologia foi desenvolvida, o WAP que veio atender as novas exigências da sociedade atual.

Agora, os usuários ao invés de ir a *sites* de pesquisa, ou grandes *sites* portais agora vão a *sites* portais pequenos e acessam os serviços que realmente eles estão interessados. Assim tem-se que a palavra chave para o sucesso WAP é utilidade, e se essa utilidade não for suficiente ou necessária, dificilmente ganhará espaço nesta sociedade.



**Universidade Federal do Pará**  
**Departamento de Engenharia Elétrica e de Computação**

A partir deste ponto de vista, tem-se o WAP com sérias dificuldades de adaptação, pois da forma como vem sendo aplicado compromete seu sucesso. As altas tarifas, lentidão na transmissão de dados e recursos limitados deixam os usuários receosos de utilizá-lo.

O que se tem é a maioria das pessoas usando o WAP uma única vez e então desistindo de utilizá-lo. É trabalho demais para pouca recompensa. Enquanto o usuário está tendo dificuldades com um teclado de apenas 9 teclas tentando digitar URLs, sua conta telefônica cresce a cada segundo.

Essas dificuldades tendem a desestimular os usuários que passarão a utilizar o WAP apenas em situações muito específicas e raras. Com a diminuição dos acessos, toda a cadeia produtiva da tecnologia poderá ter sua receita e retorno de investimentos ameaçados.

Assim sendo, ficam em aberto algumas soluções para estas dificuldades, e como um passo seguinte na continuação deste trabalho sugere-se um estudo das novas tecnologias, como as tarifações por pacotes de 3ª geração, a fim de proporcionar um acesso rápido, fácil, gráfico, com baixo custo e o mínimo de restrições.

Uma sugestão mais sofisticada seria o desenvolvimento prático de simuladores como conversores de WML em HTML ou então o estudo dos muitos já desenvolvidos pelas diferentes prestadoras de serviços WAP, como exemplo, tem-se o simulador *GELON*, *WAPSILON*, *M3GATE*, *Phone.com*, *Samart Phone Ericson R380* e outros.

Em relação ao IP móvel sugere-se a implementação das redes *ad hoc* em combinação com o IP Móvel, assim como o estudo de novos mecanismos de encapsulamento e algoritmos de roteamento.

Por fim, sugere-se o estudo da Qualidade de Serviço em Redes Móveis Sem Fio, visto que a garantia da qualidade de serviço em redes de computadores tem se tornado uma necessidade para os novos tipos de aplicações, sendo esta uma tarefa complexa principalmente na presença de um ambiente móvel sem fio.



**Universidade Federal do Pará**  
**Departamento de Engenharia Elétrica e de Computação**

**REFERÊNCIAS BIBLIOGRÁFICAS**

- [1] SHILLER, JOCHEN H., “MOBILE COMMUNICATIONS”, 2º EDIÇÃO, ADDISON WESLEY, 2000.
- [2] HISATUGU, W. H., “IP MÓVEL”, UNIVERSIDADE ESTADUAL DE LONDRINA, CIÊNCIA DA COMPUTAÇÃO, TCC, 2000.
- [3] SILVA JR., G. F., “PROTOCOLO WAP”, TECNOLOGIA EM PROCESSAMENTO DE DADOS, TCC, 2000.
- [4] OLIVEIRA, A., COSTA, A. M., RABELO, D., APOLÔNIO, I. G., SILVA, P. E., “A TECNOLOGIA WAP NO BRASIL”, UNIVERSIDADE PAULISTA, ADMINISTRAÇÃO DE EMPRESAS COM HABILITAÇÃO EM ANÁLISE DE SISTEMAS, TCC, 2001.
- [5] WIRELESS BR, ARTIGOS DE TELECOMUNICAÇÕES. PESQUISA NO SITE <http://sites.uol.com.br/wireless>.