

TCP Móvel

FEUP – 2003

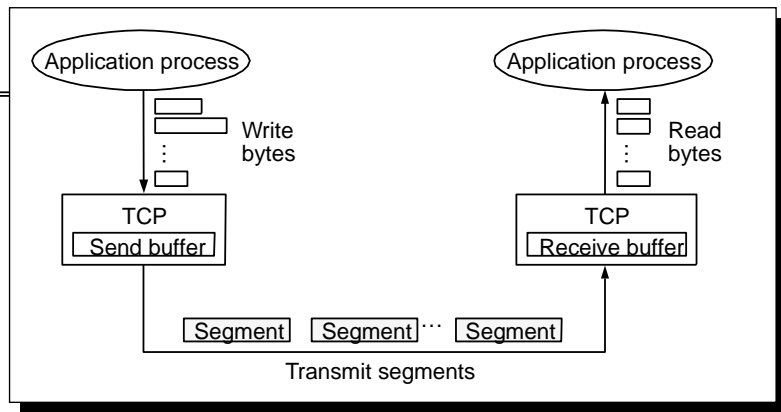
MPR

Introdução

- ◆ TCP
- ◆ TCP – Controlo de fluxo
- ◆ TCP – Controlo de Congestionamento
- ◆ Problemas do TCP em comunicações móveis
- ◆ Potenciais soluções

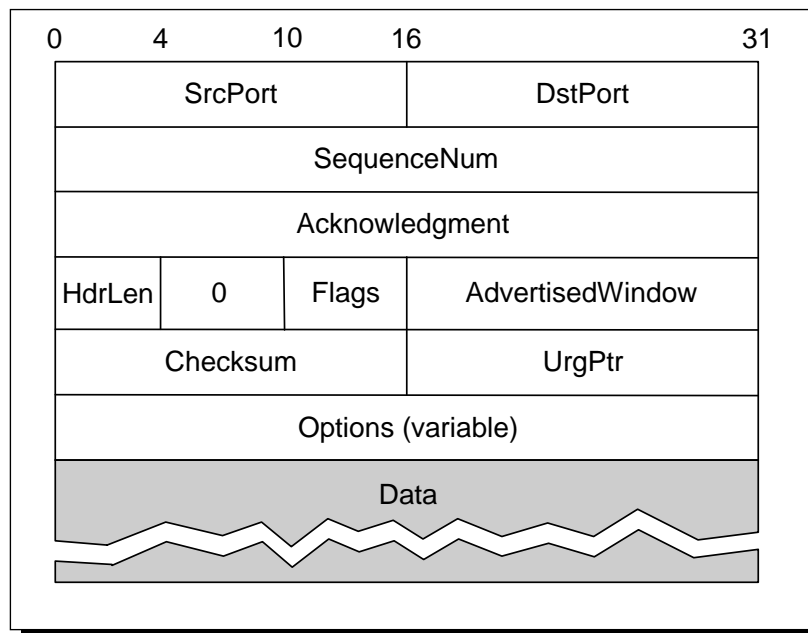
TCP

- ◆ Orientado às ligações
- ◆ Fluxo de bytes
- ◆ Full-duplex



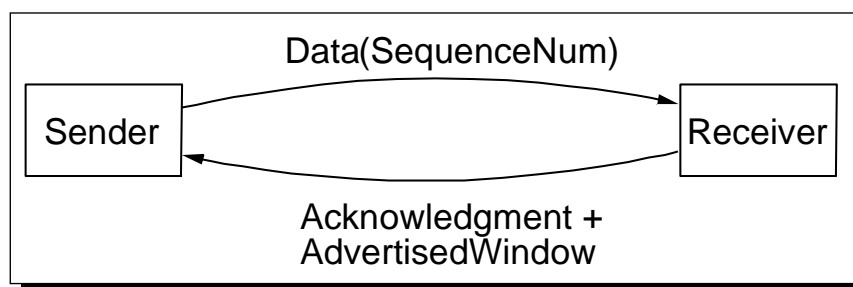
- ◆ Controlo de fluxo
 - » Evita *congestionamento do receptor*
 - » Controlado por receptor
- ◆ Controlo de congestionamento
 - » Evita *congestionamento da rede*
 - » Controlado por emissor

TCP - Formato do Segmento (./.)

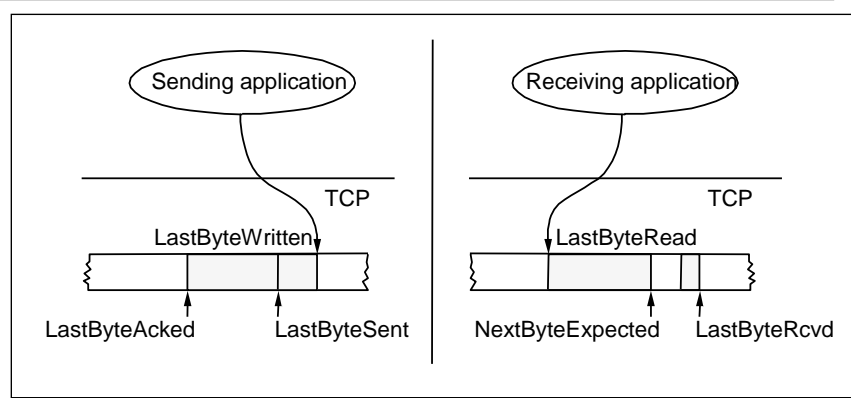


TCP - Formato do Segmento (../..)

- ◆ Cada ligação identificada pelo vector
(SrcPort, SrcIPAddr, DstPort, DstIPAddr)
- ◆ Janela deslizante + controlo de fluxo
SequenceNum, Acknowledgment, AdvertisedWindow

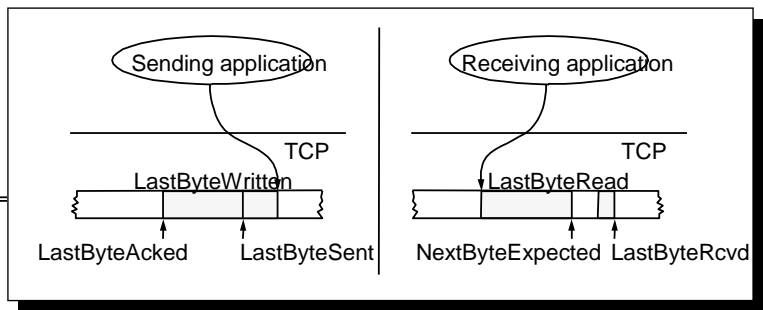


TCP, Controlo de Fluxo – Mecanismo de Janela Deslizante



- » No emissor
 - LastByteAcked \leq LastByteSent
 - LastByteSent \leq LastByteWritten
 - Buferiza bytes entre LastByteAcked e LastByteWritten
- » No receptor
 - LastByteRead $<$ NextByteExpected
 - NextByteExpected \leq LastByteRcvd + 1
 - Buferiza bytes entre NextByteRead e LastByteRcvd

TCP – Controlo de Fluxo



◆ Comprimento do buffer

- no emissor → MaxSendBuffer
- no receptor → MaxRcvBuffer

◆ No receptor

$LastByteRcvd - LastByteRead \leq MaxRcvBuffer$

$AdvertisedWindow = MaxRcvBuffer - (LastByteRcvd - LastByteRead)$

◆ No Emissor

$LastByteSent - LastByteAacked \leq AdvertisedWindow$

$EffectiveWindow = AdvertisedWindow - (LastByteSent - LastByteAacked)$

$LastByteWritten - LastByteAacked \leq MaxSendBuffer$

- Processo emissor bloqueia (função write não retorna) enquanto
 $(LastByteWritten - LastByteAacked) + y > MaxSenderBuffer$
 $y \rightarrow$ número de bytes que processo tenta enviar

WTCP 8

TCP - Controlo de Congestionamento (./.)

◆ Princípio

- » Rede best-effort
- » Cada fonte determina capacidade de transporte oferecida
- » **ACKs** regulam transmissão (relógio da fonte)

◆ Dificuldade

- » Determinar capacidade de transmissão oferecida
- » Ajuste a mudanças de capacidade disponível

TCP – Controlo de Congestionamento (../..)

- ◆ Congestionamento → filas de espera dos routers enchem
 - » Atrasos dos pacotes aumentam; perdem-se pacotes
 - » Generalidade do tráfego IP transportado em ligações com fios → fiáveis
 - » TCP: perda de pacote = congestão. Nesta situação, o TCP
 - Retransmite pacote perdido
 - Reduz o débito de transmissão, para que as filas dos routers se esvaziem
 - Aumenta gradualmente o débito de transmissão

- ◆ O emissor detecta uma perda quando
 - a) Recebe 3 confirmações repetidas → pacote seguinte perdido
 - b) Ocorre timeout

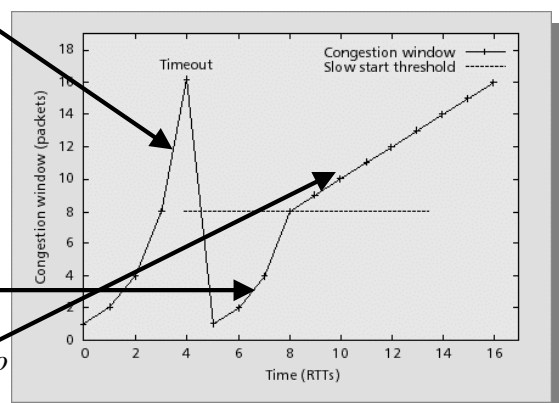
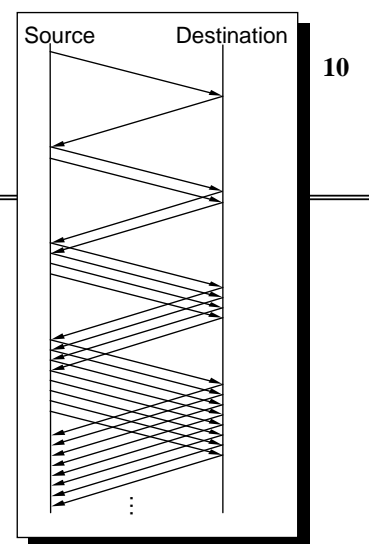
(Para cada segmento transmitido é activado um timer com valor $2 * RTT$)

TCP – Arranque Lento

- ◆ TCP tem `congestionWindow`
 - » número de pacotes em trânsito, sem causar congestão
 - » Novos pacotes enviados se permitidos por
 - `congestionWindow`, e
 - `Advertisedwindow`, do receptor

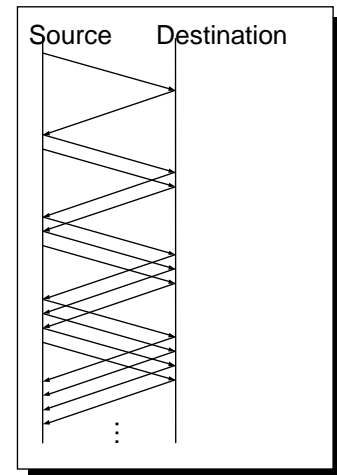
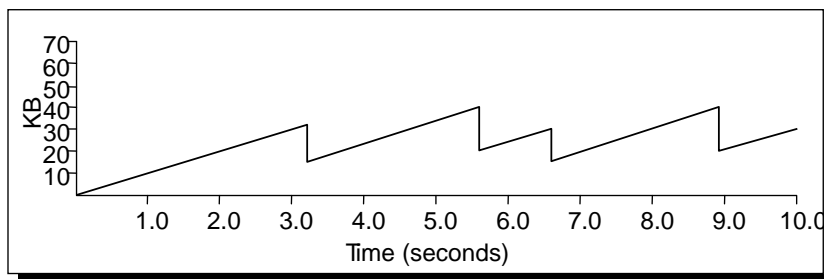
- ◆ *Arranque Lento* ☺
 - » Emissor começa com `congestionWindow=1sgm`
 - » Duplica `congestionWindow` em cada `RTT`

- ◆ Quando detecta perda de pacote, por timeout
 - » `threshold = ½ congestionWindow`
 - » `congestionWindow=1sgm`
(router esvazia filas)
 - » Pacote perdido retransmitido
 - » *Arranque Lento* enquanto
`congWindow < threshold`
 - » Depois → fase de *Prevenção De Congestionamento*



Prevenção De Congestionamento (Congestion Avoidance)

- ◆ *Prevenção De Congestionamento* (subida aditiva)
 - » Incrementa `congestionWindow` de 1 sgm, por cada RTT
- ◆ Detecção de perda de pacote, por recepção de 3 ACKs duplicados
 - » Deduz que pacote se perdeu,
 - não por congestionamento severo, porque segm seguintes chegaram destino
 - » Retransmite pacote perdido
 - » `congestionWindow = congestionWindow / 2`
 - » Fase de *Prevenção De Congestionamento*



TCP – Controlo de Congestionamento

- ◆ Na realidade, um pouco mais complexo
- ◆ RFC 2001, “TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms”

Redes Sem Fios - Características

- ◆ **Ligações em WLAN e Redes Celulares**
 - Atrasos de propagação equiparáveis às ligações com fios
 - Taxas de erro elevadas ← interferência externa
 - Mobilidade faz variar as taxas de erro da ligação
- ◆ **WLANs**
 - Atraso de transmissão baixo (débitos elevados)
 - Frame Error Ratio – 1-2 %
- ◆ **Redes Celulares (GSM)**
 - Débitos modestos → 9.6 Kbit/s
 - Atrasos maiores que em WLANs (débito menor, distâncias maiores)
 - FER - 1-2 %
 - Para dados, Radio Link Protocol (RLP)
 - ◆ Segmentação e reassemblagem de datagramas IP
 - ◆ Recuperação de erros → ARQ, Selective Repeat
 - BER: reduzido de 10^{-3} para 10^{-8} , mas
 - Diminui débito útil. Aumenta atraso de propagação

TCP em Redes sem fios - Problemas

- ◆ **TCP assume que perdas se devem a congestionamento**
 - problemático para as ligações sem fios
- ◆ **WLANs**
 - FER = 1 - 2 %
 - Segmentos dados TCP competem com ACKs → colisões → FER aumenta
 - Mobilidade aumenta FER
 - Experimentalmente, FER = 1.5 % → redução débito em 30 % ☹
- ◆ **Redes Celulares**
 - » links de voz, transparente (sem RLC)
 - Tramas pequenas, FER = 1 - 2%
 - Pacote IP segmentado em n tramas físicas (ex. 50)
 - Probabilidade de datagrama IP bem recebido → 60% !
 - » Com RLC
 - RLC recupera trama errada antes de timeout TCP expirar, mas
 - RTT muito alto e variável: 600 ms – 10 s (medido com ping em GSM)
 - link muito comprido, janelas de transmissão grandes

TCP, Problemas – Trajectos Extremo-a-Extremo

WTCP 15

- ◆ Trajectos com n ligações sem fios → problema agrava-se
 - » Débito baixa ainda mais
- ◆ Quando pacote se perde na rede, depois de atravessar várias ligações sem fios, tem que ser retransmitido através delas outra vez → Desperdício de recursos
- ◆ Nas situações de handover TCP fará retransmissões, com conseqüente redução de débito. Na realidade seria melhor suspender o processo

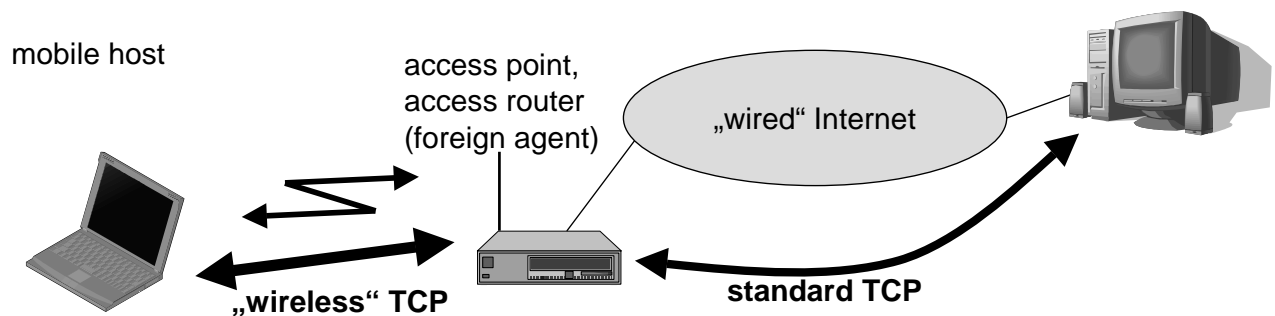
WTCP 16

Terminologia

- ◆ Nas soluções que se seguem é usada, por simplicidade, a terminologia do IP Móvel, v4
- ◆ Foreign Agent, poderá designar um Router de Acesso

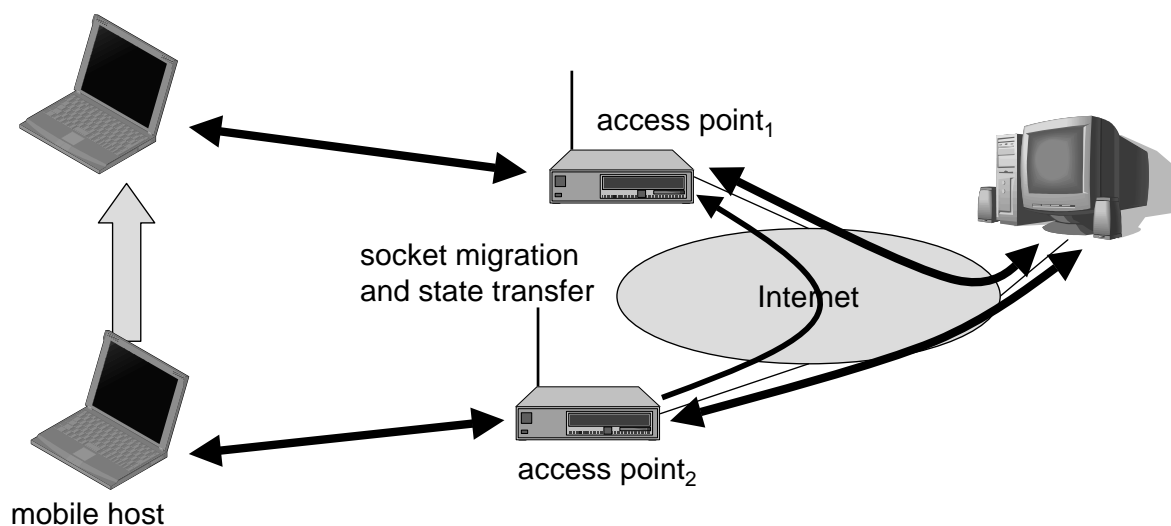
I-TCP, Indirect TCP (./..)

- » I-TCP transforma ligação TCP em 2 ligações TCP
 - ➔ wireless TCP + standard TCP
- » TCP modificado nos MNs; CNs desconhecem modificação
- » TCP no FA
 - Termina as 2 ligações, actuando como proxy
 - ◆ CN envia segmento; FA envia ACK para CN e segmento para MN
 - ◆ MN envia segmento; FA envia ACK para MN e segmento para CN
 - Bufferização de segmentos



I-TCP, Handover - Migração de Sockets e de Bufers

- ◆ Handover requer
 - » Redireccionamento de pacotes entretanto buferizados (e confirmados)
 - » Migração de toda a informação dos sockets
 - portas devem manter-se; ligação não pode ser re-estabelecida



I-TCP, Avaliação

◆ Vantagens

- » TCP mantém-se para CNs
- » Erros de transmissão na ligação sem fios não se propagam para rede fixa
- » TCP modificado pode ser usado num só troço (ex. FA – MN)
- » Retransmissão rápida de pacotes

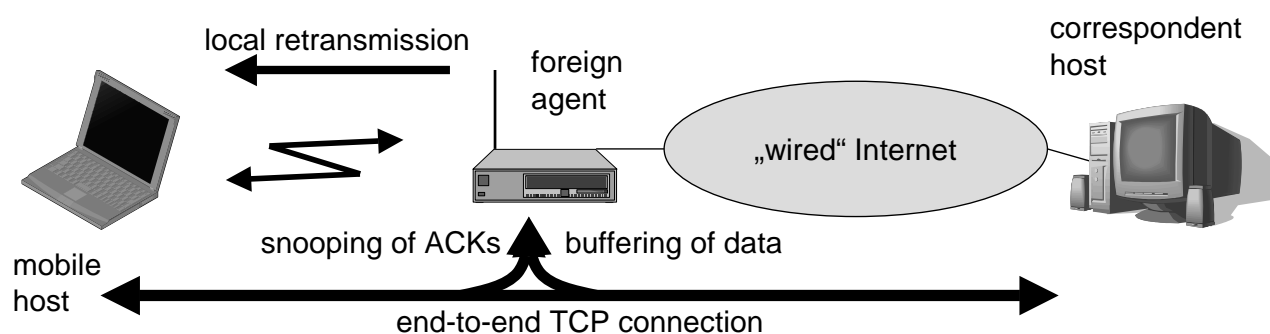
◆ Desvantagens

- » Perda do significado extremo-a-extremo característico do TCP
 - Um ACK deixa de significar que o receptor realmente recebeu o pacote (FA pode entretanto avariar)
- » Possíveis atrasos introduzidos pela buferização
- » Complexidade no handover

Snooping TCP (./.)

◆ Adaptação do TCP no FA

- » Buferização no FA dos pacotes enviados para MN
- » Pacotes perdidos na ligação sem fios são retransmitidos de imediato
 - Pelo FA ou pelo MN
- » FA
 - monitora fluxos de pacotes
 - identifica ACKs nos 2 sentidos
 - filtra ACKs



Snooping TCP (../..)

- ◆ Transferência de segmentos para o MN
 - » FA buferiza segmentos até receber ACK do MN
 - » FA detecta perda de segmentos por duplicação de ACKs ou timeout
 - » Em caso de perda, FA retransmite para MN, evitando *ArranqueLento* no CN
Timeouts no FA são mais pequenos que no CN → uma só ligação
- ◆ Transferência de segmentos do MN
 - » FA detecta perda de segmentos na ligação s/ fios monitorando números sequência
 - » FA responde ao MN enviando um NACK (TCP modificado)
 - » MN retransmite dados com pequeno atraso
- ◆ Integração com o nível MAC
 - » MAC pode ter mecanismos semelhantes
 - » MAC pode detectar pacotes duplicados e eliminá-los
- ◆ Problemas → Snooping TCP
 - » não isola as ligações sem fios tão bem como o I-TCP
 - » pode deixar de funcionar se se usarem técnicas de cifragem

M-TCP, Mobile TCP

- ◆ Problemas de terminação frequente da ligação (handovers, zonas escuras)
- ◆ M-TCP divide ligação em 2, como o I-TCP
 - » TCP normal do CN até ao FA (Supervisory Host - SH, no M-TCP)
 - » TCP optimizado entre o MN e o SH
- ◆ Supervisory Host
 - » Não buferiza, não retransmite
 - » monitora todos os pacotes; quando detecta terminação da ligação
 - Coloca a janela do emissor, `advertisedWindow`, a zero
 - Emissor passa a modo persistente, tentando enviar um pacote
 - » O antigo ou o novo SH (situação de handover) reabrem a janela
- ◆ Vantagens
 - » Mantém semântica, resolve terminações de ligação, não buferiza
- ◆ Desvantagens
 - » Perdas nas ligações sem fios são propagadas à rede fixa
 - » TCP adaptado, na ligação sem fios

Retransmissão Rápida / Recuperação Rápida

- ◆ Mudança de FAs → perda de segmentos
 - » Apesar de não haver congestionamento, TCP pode reagir com *ArranqueLento*
- ◆ Alternativa → Forçar MN a fazer Retransmissão Rápida
 - » Assim que se regista no novo FA, MN
 - Envia, propositadamente, ACKs duplicados
 - » CN detecta perda por ACKs duplicados, e evita *ArranqueLento*
 - » Adicionalmente, no MN,
 - TCP é forçado a manter a mesma janela
 - Retransmite os segmentos ainda não confirmados
- ◆ Vantagem
 - » Mudanças simples → grande melhoria de desempenho
- ◆ Desvantagem
 - » Necessário interligar IP e TCP (detecção de handovers)
 - » Com handover demorado, CN dá mesmo timeout

Suspensão dos Timeouts

- ◆ MN podem ser desligados durante longos períodos
(túnel, células sobrecarregadas)
 - » TCP termina ligação
- ◆ Congelamento do TCP
 - » MAC pode conseguir prever esta interrupção
 - » MAC informa TCP
 - » TCP suspende envio → timeouts suspensos → congestão não deduzível
 - » MAC avisa TCP quando ligação voltar
- ◆ Vantagem
 - » Mecanismo independente dos dados
- ◆ Desvantagem
 - » TCP tem que ser alterado → MN e CN
 - » Depende do nível MAC

Quadro Comparativo

Approach	Mechanism	Advantages	Disadvantages
Indirect TCP	splits TCP connection into two connections	isolation of wireless link, simple	loss of TCP semantics, higher latency at handover
Snooping TCP	“snoops” data and acknowledgements, local retransmission	transparent for end-to-end connection, MAC integration possible	problematic with encryption, bad isolation of wireless link
M-TCP	splits TCP connection, chokes sender via window size	Maintains end-to-end semantics, handles long term and frequent disconnections	Bad isolation of wireless link, processing overhead due to bandwidth management
Fast retransmit/fast recovery	avoids slow-start after roaming	simple and efficient	mixed layers, not transparent
Transmission/time-out freezing	freezes TCP state at disconnect, resumes after reconnection	independent of content or encryption, works for longer interrupts	changes in TCP required, MAC dependant

TCP Improvements I

$$BW \leq \frac{0.93 * MSS}{RTT * \sqrt{p}}$$

- max. TCP BandWidth
- Max. Segment Size
- Round Trip Time
- loss probability

- ◆ Initial research work
 - » Indirect TCP, Snoop TCP, M-TCP, T/TCP, SACK, Transmission/time-out freezing, ...
- ◆ TCP over 2.5/3G wireless networks
 - » Fine tuning today's TCP
 - » Learn to live with
 - Data rates: 64 kbit/s up, 115-384 kbit/s down; asymmetry: 3-6, but also up to 1000 (broadcast systems), periodic allocation/release of channels
 - High latency, high jitter, packet loss
 - » Suggestions
 - Large (initial) sending windows, large maximum transfer unit, selective acknowledgement, explicit congestion notification, time stamp, no header compression
 - » Already in use
 - i-mode running over FOMA
 - WAP 2.0 (“TCP with wireless profile”)

TCP Improvements II

- ◆ Performance enhancing proxies (PEP, RFC 3135)
 - » Transport layer
 - Local retransmissions and acknowledgements
 - » Additionally on the application layer
 - Content filtering, compression, picture downscaling
 - E.g., Internet/WAP gateways
 - Web service gateways?
 - » Big problem: breaks end-to-end semantics
 - Disables use of IP security
 - Choose between PEP and security!
- ◆ More open issues
 - » RFC 3150 (slow links)
 - Recommends header compression, no timestamp
 - » RFC 3155 (links with errors)
 - States that explicit congestion notification cannot be used
 - » In contrast to 2.5G/3G recommendations!

