TCP Over Wireless

Issues

Packet loss due to bit errors and handoff, which is different than the traditional causes of in TCP environment – congestion.

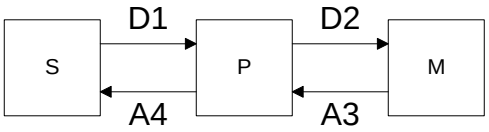
Possible Solutions

(1) Link layer retransmission with FEC (Forward Error Correction)
The wireless link implements a retransmission protocol coupled with FEC at the link layer.
Advantages
It improves the reliability of communication independent of the higher-level protocol.
Disadvantages
As error rates become significant, the performance will be degraded.

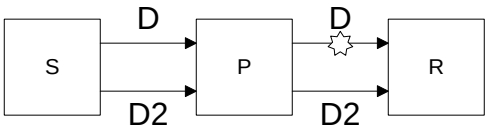
(2) Split transaction
It involves splitting a TCP connection between a fixed and mobile host into two separate connections at the base station -- one TCP connection between the fixed host and the base station, and the other between the base station and the mobile host.
Advantages
It achieves a separation of flow and congestion control of the wireless link from that of the fixed network and hence results in good bandwidth at the sender.
Disadvantages
Loss of Semantics; Application relinking Software overhead; Handoff latency.

Solution in the paper

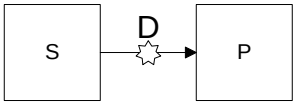
(Hari Balakrishnan et al. - Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks)



1. The receiver is mobile, or data lost remotely



2. The sender is mobile, or data lost locally



Load Balancer / Network Address Translation (NAT)

Issues

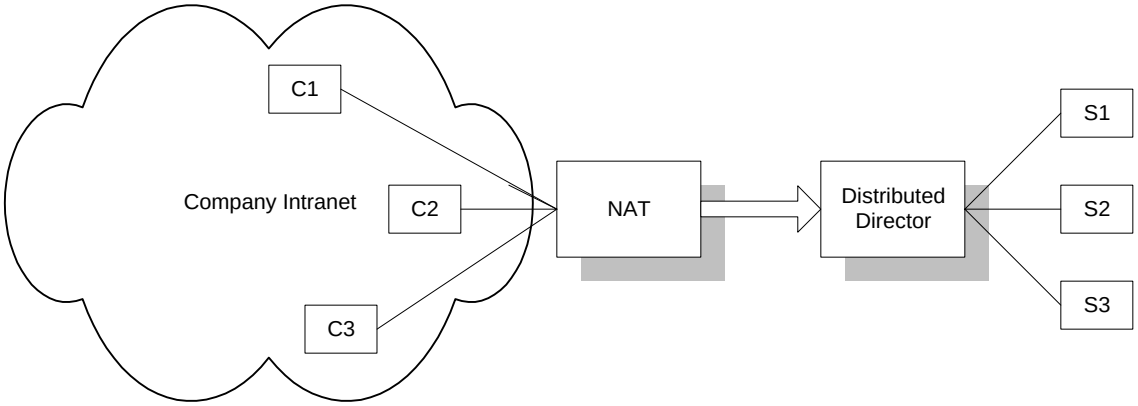
IP must be globally uniquely addressable. Address space is not big enough.
Firewall – required by security or proxy server.

Translation
NAT assigns a special port (ID) for each client Ci on the proxy so as to talk to the server, so that the port can be used to identify the hosts when the data sent back from the server.

Distributed Director
Distribute the traffic equally to the separate hosts.

What if NAT fails?
May add redundancy for the NAT

Diagram



Security in Data Communication

Issues

- Faked identification
- Authentication
- Secrecy
- Integrity

Solutions

- Shared Secrecy (Symmetric Encryption)
  DES, RC4 etc
  Data -> f(secret, data) = ciphertext → f(secret, ciphertext) = Data

- Public Key (Asymmetric Encryption)
  The public key is used to decrypt the data encrypted by private key; and private key is used to decrypt the data encrypted by public key. The owner of the private key is the only person who can encrypt and decrypt messages from the either end of communication.
  $E_{Ke}(E_{Kd}(Data)) = E_{Kd}(E_{Ke}(Data))$

  The problem is that public key method requires CA (Certificate Authority)

- Diffie-Hellman, some sort of the public key without CA

  Protocol (Page 513, Bruce Schneier's *Applied Cryptograhpy 2nd Edition*)

  Alice and Bob agree on a large prime, *n* and *g*, such that *g* is primitive mod *n*. These two numbers don't have to be secret. …Then the protocol goes as follows:
  (1) Alice chooses a random large integer *x* and sends Bob
  $X = g^x$ mode n
  (2) Bob chooses a random large integer *y* and sends Alice
  $Y = g^y$ mode n
  (3) Alice computes
  $k = Y^x$ mode n
  (4) Bob computes
  $k' = X^y$ mode n

  Both *k* and *k'* are equal to g^{xy} mode *n*. No one listening the channel can compute that value; they only know *n*, *g*, *X* and *Y*. Unless they can compute the discrete logarithm and recover *x* and *y*, they do not solve the problem. *k* is the secret key that both Alice and Bob computed independently.