**Politecnico di Milano**

*Facoltà di Ingegneria dell'Informazione*

# WI – 7 – TCP over wireless

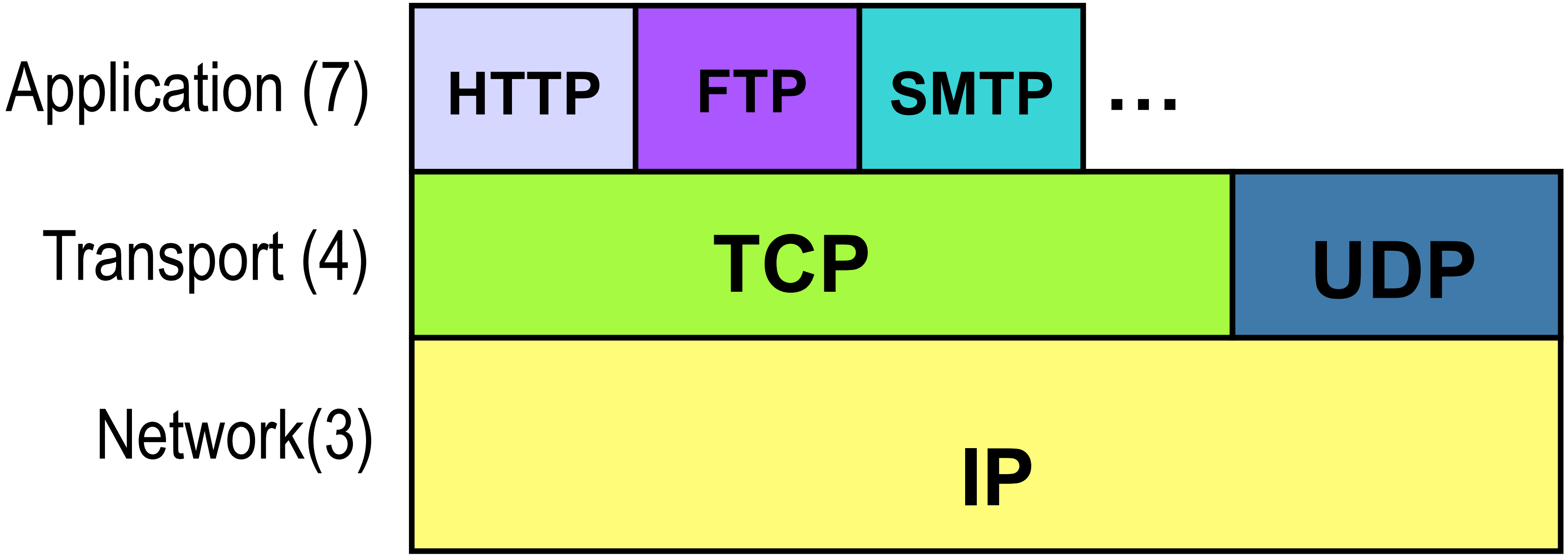
## Wireless Internet

Prof. Antonio Capone

# Outline

- TCP in heterogeneous wireless/wired networks
- Impact of wireless characteristics on TPC performance
- Taxonomy of TCP over wireless solutions
  - Link-Layer approaches
  - Indirect Protocols
  - End-to-end approaches
- Mathematical models of TCP performance with channel losses

# Internet protocol suite



Application (7)
HTTP
FTP
SMTP
…
Transport (4)
TCP
UDP
Network(3)
IP

- The Transmission Control Protocol (TCP) is the most widely used transport protocol, even if the RTP/UDP traffic is increasing
- Still today between 90-95% of Internet traffic is TCP even if UDP usage is increasing due to VOIP and IPTV
- (see e.g. CAIDA: Analyzing UDP usage in Internet traffic – www.caida.org)

# TCP over wired networks

- TCP has been designed and optimized in the last 20 years for reliable wired networks, and it has proved to be able to:
  - Use efficiently available bandwidth
  - Share fairly resources among other TCP connections
- However, the TPC flow and congestion control mechanisms as well as the retransmission scheme do not fit well with the characteristics of the wireless links that have been widely used in the last few years
- TCP main assumption for congestion control and fair resource sharing is that segment losses are due to network congestion
- If transmitting TCP detects a segment loss, it reduces the transmission rate halving the transmission window

# Radio channel

- Wireless networks shows some specific characteristics that have impact on TPC performance:
  - Random segment losses
  - High and variable delays
  - Relatively low capacity (the wireless link is often the connection bottleneck)
  - Asymmetric channels
  - Frequent disconnections

# Impact on TCP

## Random losses

- Radio networks have usually high loss probability due to for example:
  - Co-channel interference
  - User mobility
  - multipath fading
  - Discontinuous connection due to limitations in coverage
- Loss rate can be in the order of 1-10%
- This high loss rate can greatly reduce the TCP performance because of the congestion control mechanism that halves the rate at each loss
- We'll see in next slide some models able to quantify the performance degradation due to channel losses occurring with probability $p$.

# Impact on TCP

## Variable delay and low link capacity

- Some wireless channels can have a low capacity and, as a result, a high average Round Trip Time (RTT)
- The performance of TCP is approximately inversely proportional to the RTT measured (see models in the following slides)

$$RTO = RTT_{ave} + 4RTT_{dev}$$

- Moreover, it has been shown with practical experiments that in radio networks TCP delays can vary in a remarkable way.
- This characteristic can greatly affect the TCP, since the Re-transmission Time-Out (RTO) is set according to the estimated RTT

# Impact on TCP

## Asymmetric channels

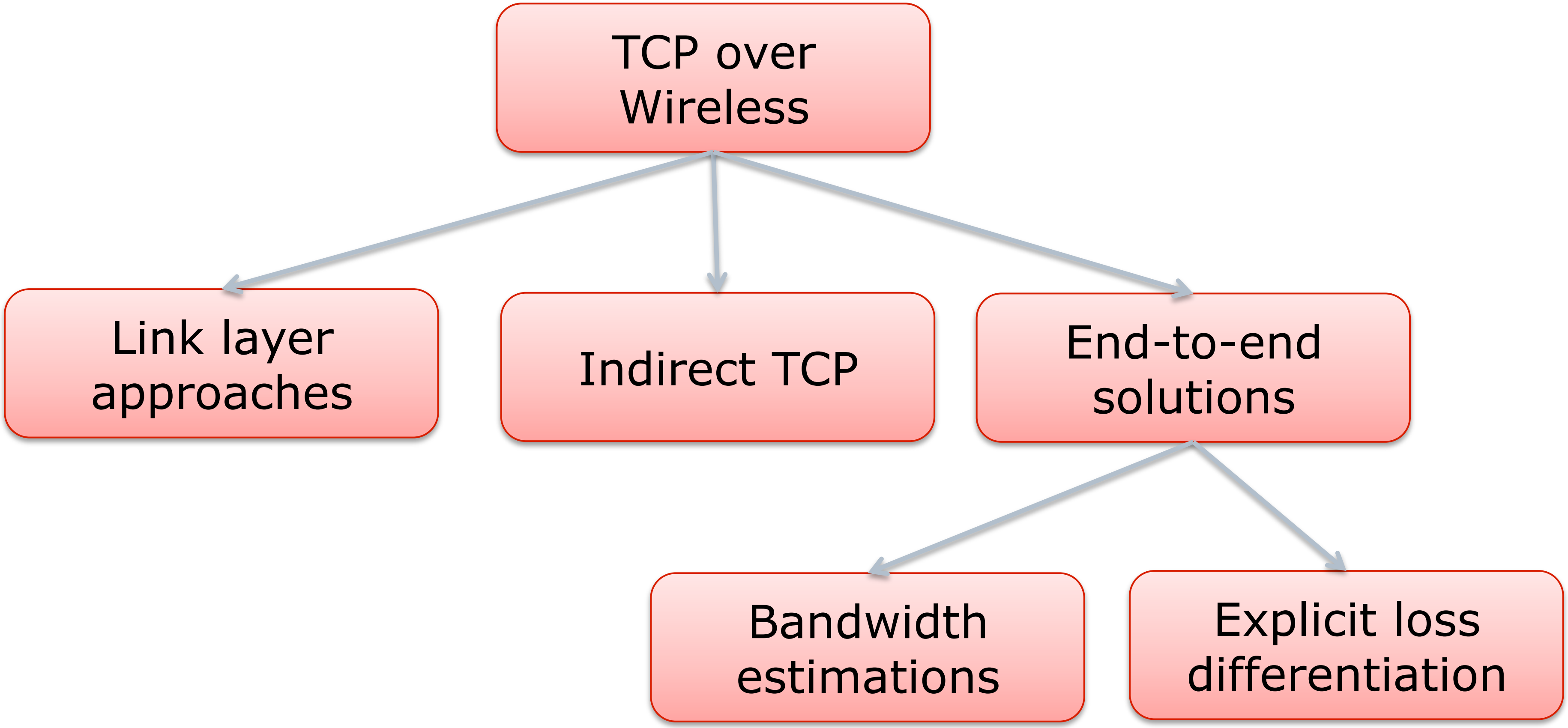
- TCP connection are “ACK Clocked”, which means that:
  - In case ACKs are, for some reason, delayed the transmitter cannot continue sending other packets and the transmission rate depends on the timing of ACK arrivals
- This can have an impact on TCP when, for example, there is a capture of the radio channel by some stations that prevent others from receiving the ACKs
- The effects on TCP are:
  - A performance degradation of the transmitting TCP
  - TPC transmissions occurring in bursts

# Impact on TCP

## Disconnections

- Due to user mobility, wireless links can experience disconnections due to:
  - Handovers between base-stations
  - Loss of coverage (coverage holes due to shadowing, indoor pure signal, etc.)
  - Deep fading due to multipath
- In these cases TCP connection can experience several consecutive retransmission time-out expirations
- When the blackout period ends, TCP source restarts transmitting in slow start with a very small window (usually 1 segment)
- As a result disconnections can greatly reduce TCP performance

# TCP over wireless: taxonomy



TCP over Wireless

Link layer approaches

Indirect TCP

End-to-end solutions

Bandwidth estimations

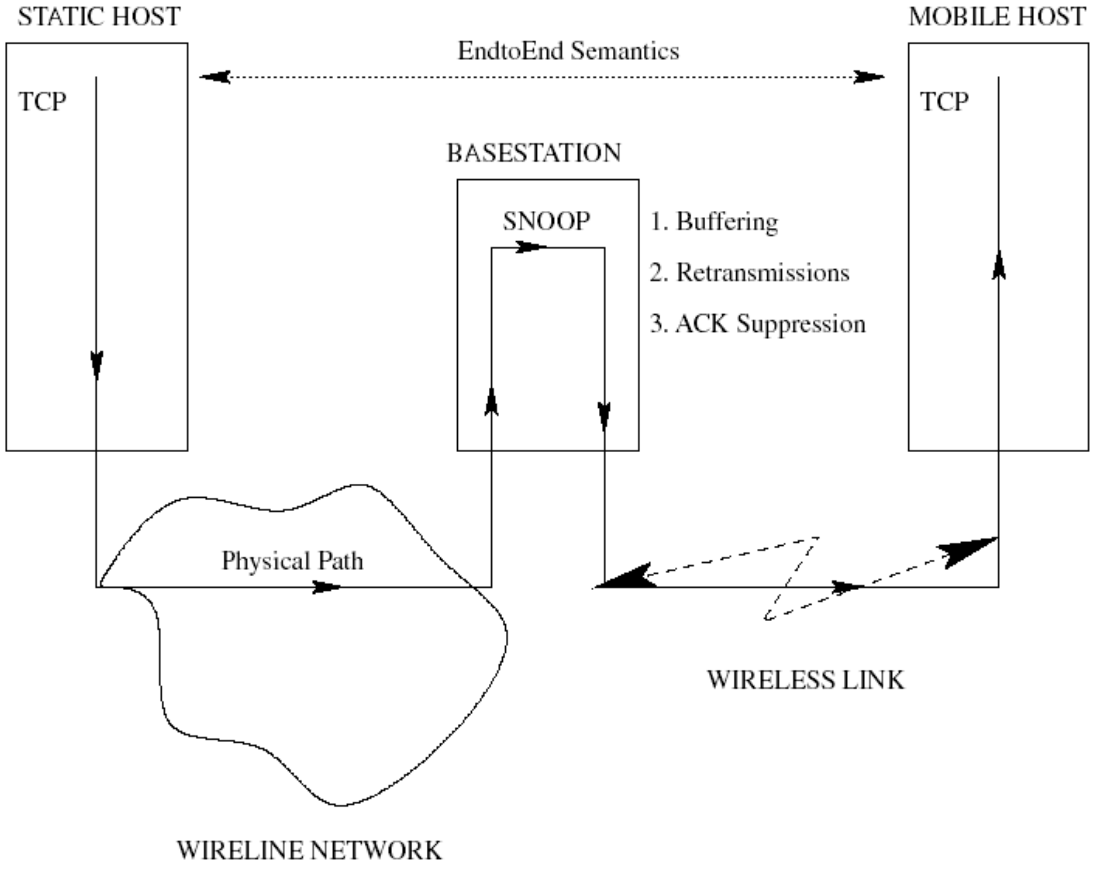
Explicit loss differentiation

# Link Layer approaches

- Solutions based on link layer approaches try to hide the characteristics of wireless network to TCP using functions implementation at layer 2
- This functions are transparent to the TCP protocol
- Usually these functions exploit the knowledge of TCP behavior to improve the performance

- Typical approach: buffering packets at wireless network nodes (e.g. access points) and retransmitting packets lost due to channel impairments
- Expected result: The TCP experiences only losses due to congestion like in wired networks
- But this may not be always achieved: variable delay due to retransmissions …

# Example: Snoop protocol



# Link Layer approaches

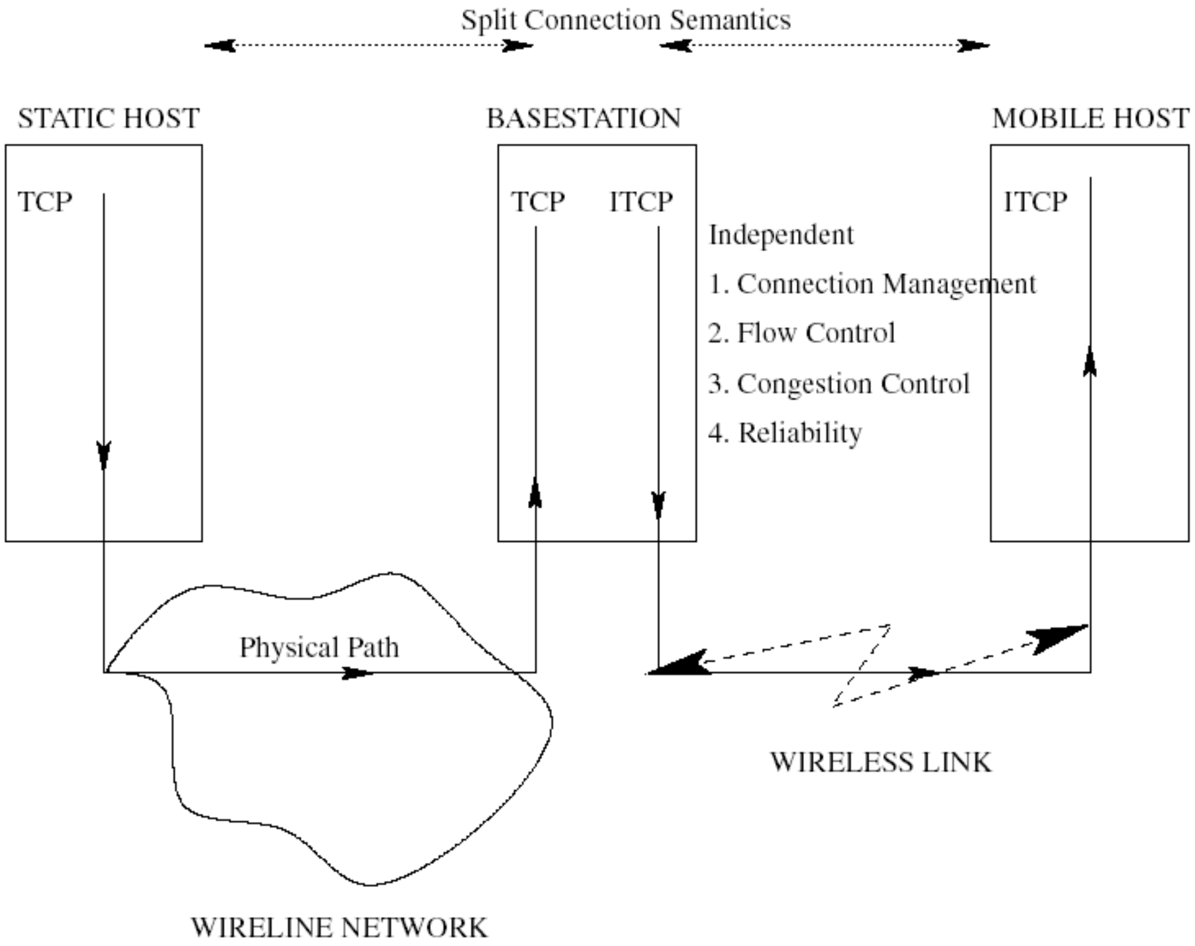
☐ Link layer solutions have the following characteristics:

1. Hide wireless link characteristics to TCP layer
2. Are transparent to TCP and do not require changes to the protocol stack of higher layer
3. Can in some cases be “TCP-aware” and adjust the link layer functions based on the status of TCP
4. Require additional “intelligence” in wireless nodes, buffers, retransmission, etc.
5. The keep the end-to-end semantic of TCP segments unchanged

# Indirect TCP

- Indirect TCP solutions are similar to those at link layer, but they split TCP connection in two parts at the radio access node (base station)
- A TCP connection is then divided into two sub-connections:
  - One between mobile host and base station
  - Another between base station and fixed host (e.g. server)
- In such a way the first sub-connection can use an ad hoc protocol to cope with the characteristics of wireless link

# Example: Indirect TCP



# Indirect TCP

- Indirect TCP solutions have the following characteristics:

1. End-to-end connection is split in two connections at the base station
2. End-to-end TCP semantic is NOT maintained
3. Special transport protocols are used for the wireless portion of the connection
4. These protocols require limited modifications to mobile host at the cost of more complexity in the base station

# End-to-end approaches

- End-to-end solutions maintain TCP protocol semantic
- However, they require modifications of TCP layer that can be localized at
  - Transmitting side (server)
  - Receiving side (host)
  - Both ends
- Modifications at server side only allows a easier diffusion of the solutions, while a change in the host side is much difficult to implement

# End-to-end approaches

☐ End-to-end solutions have the following characteristics:

1. They maintain the end-to-end semantic of TCP
2. They allows the use of sophisticated algorithms for congestion control and error recovery
3. They can be easily adopted and implemented in the Internet (server side modifications only)

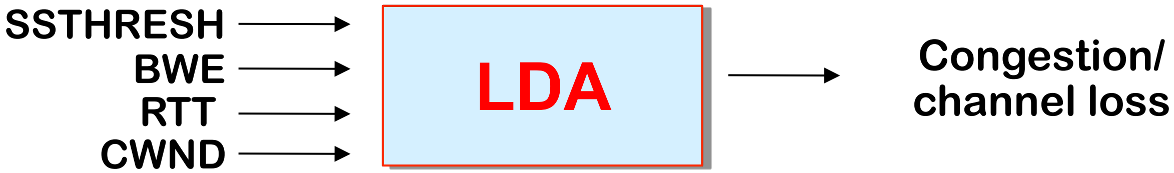
# Examples of End-to-End solutions

There are algorithms that provides estimations of the network status based on end-to-end transmissions:

- Algorithms that try to improve available bandwidth estimation of TCP and make it more robust to random losses due to the channel

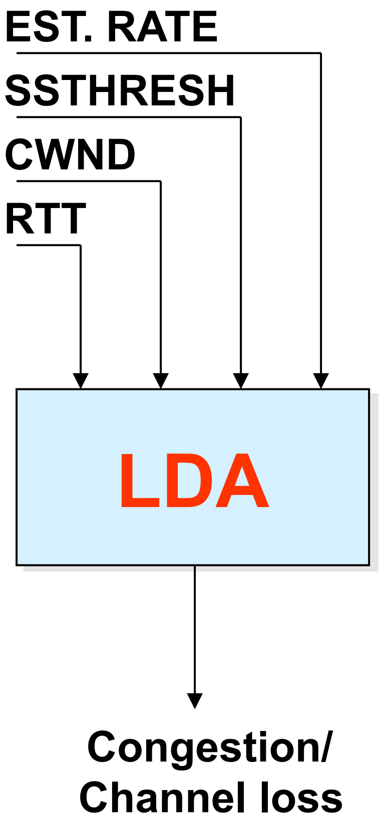
**TCP TIBET and TCP Westwood**

- Algorithms based on explicit loss differentiation (channel or congestion) - LDA - Loss Differentiation Algorithms – based on the type of loss different reactions are considered



**TCP NewReno-LDA**

# Loss Differentiation Algorithms (LDA)



- ☐ Estimation of the cause of segment losses based on the TCP state variables
- ☐ The estimation allows to take different reactions based on the type of loss
- ☐ Example of algorithms: TCP NewReno-LDA, TCP Veno, J-TCP, …

# Mathematical models

- Models of TCP performance over wireless

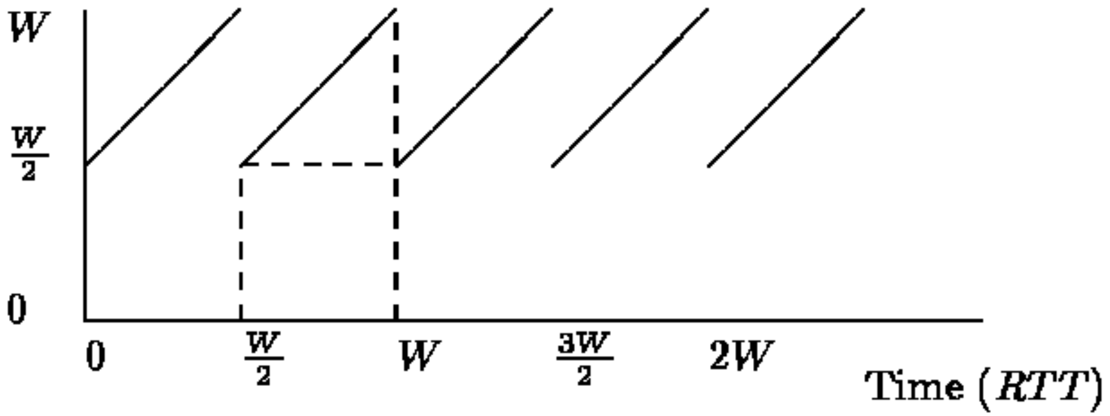
1. M. Mathis, J. Semke, J. Mahdavi, T.Ott, “The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm”, Computer Communications Review, vol. 27(3), July 1997
2. J. Padye, V. Firoiu, D. Towsley, J. Kurose, “Modeling TCP Reno performance: a simple model and its empirical validation”, IEEE/ACM Transactions on Networking, Vol. 8(2), April 2000, pagine: 133 - 145
3. N. Cardwell, “Modeling the Performance of Short TCP Connections”

# Long-Lived connections

- Assumptions:
  - Long file transfer, for instance with FTP ("long-lived" connections).
  - Connection is always in Congestion Avoidance
  - All losses are detected with 3 duplicated ACKs (no TimeOuts)
  - Round Trip Time (*RTT*) is constant
  - The wireless link has random losses with probability *p and* therefore on average with have $\frac{1}{p}$ correct transmission and then a loss

# Long-Lived connections

- With these assumptions the congestion window (cwnd) is periodic:



- W is the maximum value of the cwnd when a loss occurs
  - At equilibrium point the minimum window is therefore W/2
  - If at the receiver acknowledges each segment we need exactly W/2 Round Trip Times to come back to the value of W for the window

# Long-Lived connections

- The duration of each cycle is $\frac{W}{2}RTT$
- The total number of segments transmitted in a cycle is given by the area below the curve in the figure and therefore

$$\left(\frac{W}{2}\right)^2 + \frac{1}{2}\left(\frac{W}{2}\right)^2 = \frac{3}{8}W^2$$

- In each cycle we have 1/p segments transmitted and therefore

$$\frac{3}{8}W^2 = \frac{1}{p}$$

# Long-Lived connections

- W can be then calculated as a function of *p*

$$W = \sqrt{\frac{8}{3p}}$$

- The *goodput* of the connection is therefore:

$$Goodput = \frac{\text{data transmitted in a cycle}}{\text{cycle duration}} = \frac{MSS \cdot \frac{3}{8} W^2}{RTT \cdot \frac{W}{2}} = \frac{\frac{MSS}{p}}{RTT \cdot \sqrt{\frac{2}{3p}}}$$

  - MSS= Maximum Segment Size (bit)
  - RTT= Round Trip Time (secondi)

# Long-Lived connections

- Defining the constant $C = \sqrt{\frac{3}{2}}$ we get:

$$Goodput = \frac{MSS}{RTT} \frac{C}{\sqrt{p}}$$

- In case of Delayed ACKs policy (RFC 2581), one ACK every two segments is sent and the constant has a different value $C = \sqrt{\frac{3}{4}}$

# Long-Lived connections

- In J. Padye, V. Firoiu, D. Towsley, J. Kurose, "Modeling TCP Reno performance: a simple model and its empirical validation", IEEE/ACM Transactions on Networking, Vol. 8(2), April 2000, pagine: 133 – 145 you can find the complete model that considers also time-outs

$$Goodput \approx \frac{MSS}{RTT\sqrt{\frac{2bp}{3}} + T_0 \min(1, 3\sqrt{\frac{3bp}{8}})\,p(1+32p^2)}$$

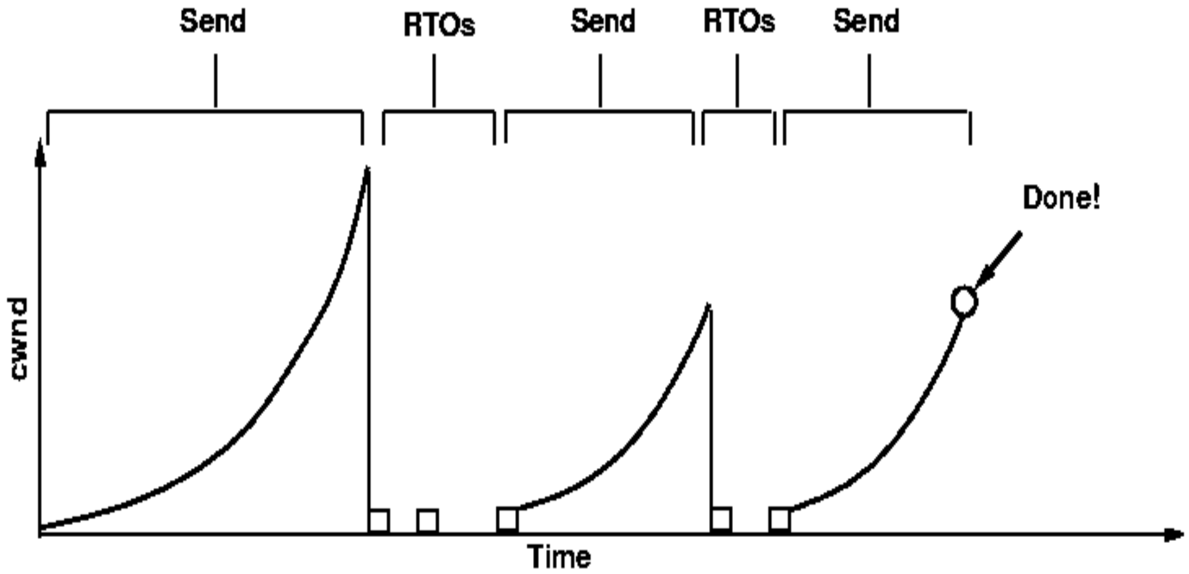
- ***p*** loss rate
- ***b*** equal to 2 in case of delayed ACK and equal to 1 otherwise
- $\boldsymbol{T_0}$ initial time out

# Short-Lived connections

- The model in: N. Cardwell, “Modeling the Performance of Short TCP Connections”, focuses on short connection with limited amount of data (“short-lived”), typical for example of HTTP (Web browsing)
- The performance figure here is the total transfer time

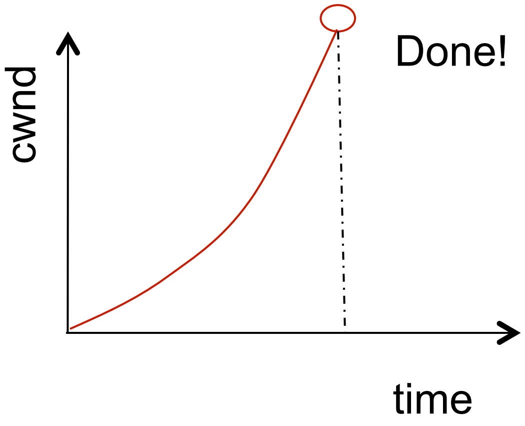
# Short-Lived connections

- If connections are short they spend most of the time in slow start state
- In case of channel losses we have multiple slow start phases as shown in the figure:



# Short-Lived connections

**ERROR FREE MODEL**



$$cwnd_{i+1} = cwnd_i + ack_i = \left(1+\frac{1}{b}\right)cwnd_i = \left(1+\frac{1}{b}\right)^i w_0$$

$$r = 1+\frac{1}{b}$$

$$data_i = \sum_{k=1}^{i}\left(1+\frac{1}{b}\right)^{i-1} w_0 = w_0\,\frac{r^i-1}{r-1}$$

$$i = \log_r\left(\frac{data(r-1)}{w_0}+1\right) = \log_r\left(\frac{d(r-1)}{MSS\cdot w_0}+1\right)$$

$$t_{TOT} = RTT\cdot\log_r\left(\frac{d(r-1)}{MSS\cdot w_0}+1\right) + RTT + t_{ACK}$$

# Short-Lived connections

## MODEL WITH RANDOM LOSSES

$l$ : number of lost segments

$p$ : loss probability

$$p = \frac{l}{l + data}$$

$$l = \frac{data \cdot p}{1 - p}$$

Probability that a loss leads to a RTO (proof in the paper) :

$$\mathrm{Q(p)} = \min\left( 1; \frac{3}{\sqrt{\frac{8}{3\mathrm{bp}}}} \right)$$

# Short-Lived connections

## MODEL WITH RANDOM LOSSES

$n:$ number of RTOs

$$n = l \cdot Q(p)$$

number of consecutive RTOs : $\dfrac{1}{1-p}$

$u:$ number of grups of RTOs

$$\mathrm{u} = \frac{l \cdot Q(p)}{\dfrac{1}{1-p}} = l \cdot Q(p) \cdot (1-p)$$

$$t_u = T_0 \frac{1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6}{1-p}$$

$$t_{RTO} = ut_u$$

# Short-Lived connections

## MODEL WITH RANDOM LOSSES

$\nu$ : number of slow start phases

$$\nu = u + 1$$

$e$ : data trasfered on average per phase

$$e = \frac{data + l}{\nu}$$

$$t_{xfer} = \nu \log_r \left( \frac{e(r-1)}{w_0} + 1 \right) RTT$$

$$t_{TOT} = t_{RTO} + t_{xfer} + RTT + t_{ACK}$$

# References

- M. Allmann, V. Paxson, W. Stevens, “*TCP Congestion Control*”, RFC 2581
- S. Floyd, T. Henderson, “*The NewReno Modification to TCP’s Fast Recovery Algorithm*”, RFC 2582
- V. Paxson , M. Allmann, “*Computing TCP Retransmission Timer*”, RFC 2988
- Van Jacobson, “Congestion Avoidance and Control”, SIGCOMM 1988
- S.Bregni, D.Caratti, F.Martignon, "*Improving TCP Performance over Wireless Networks using Loss Prediction*‟, IEEE GLOBECOM '03.
- A. Capone, L. Fratta, F. Martignon, *Bandwidth Estimation Schemes for TCP over Wireless Networks,* IEEE Trans. on Mobile Computing, vol. 3, no. 2, April 2004, pp. 129-143.