Technical University of Crete
Electronic & Computer Engineering Department

TCP Performance over UMTS Network

Diploma Thesis

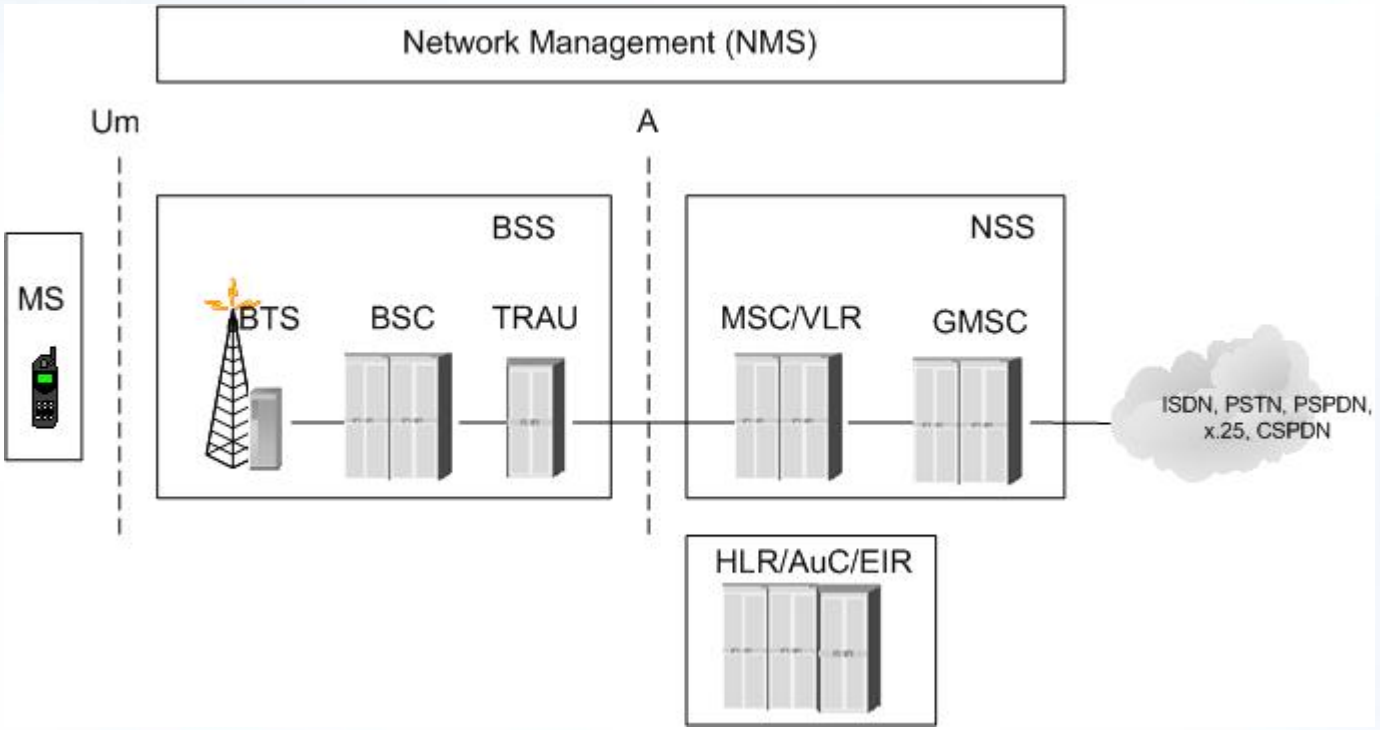
by

Smaragdakis Georgios

OUTLINE

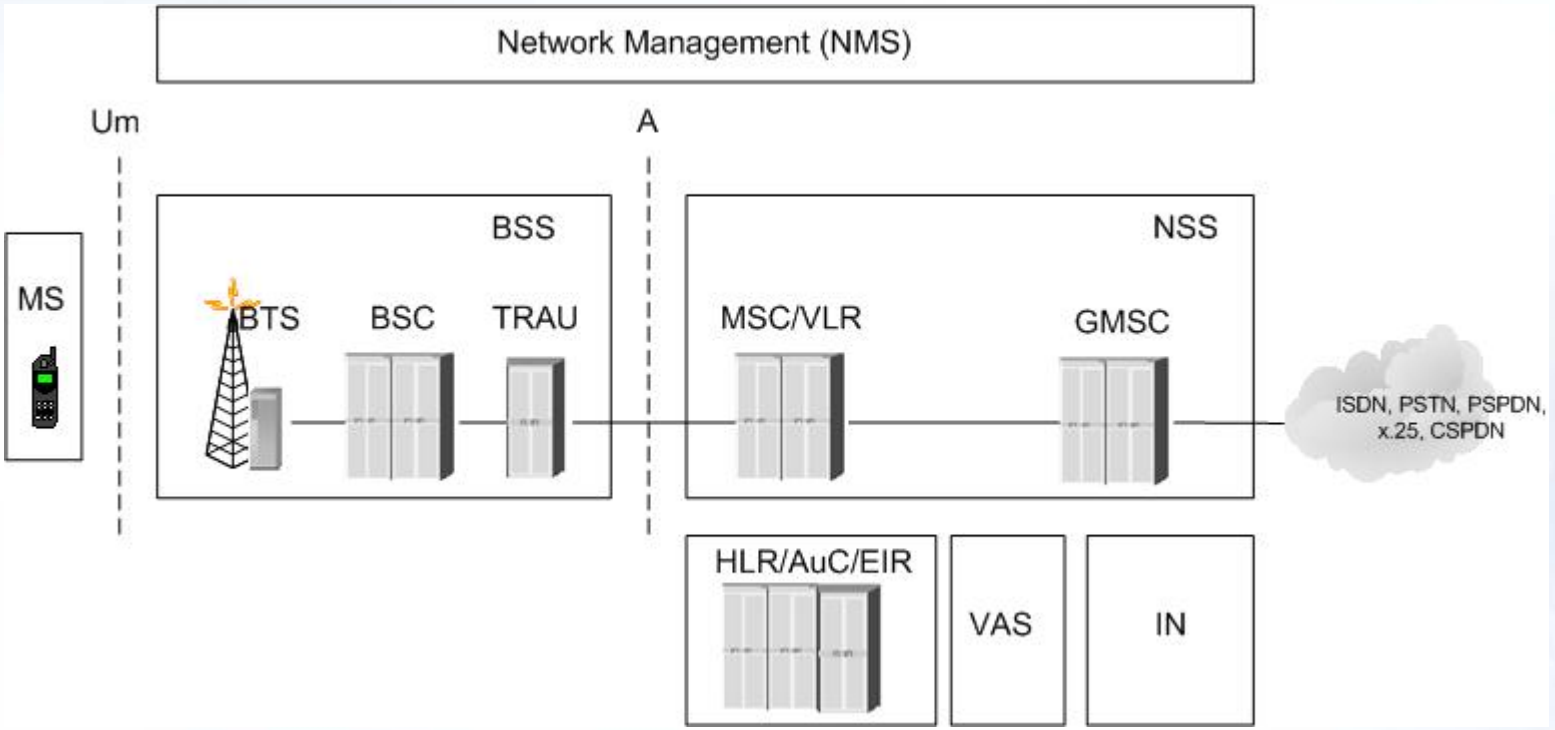
- From Mobile Networks to UMTS Revolution
- UMTS Radio Access Network
- UMTS Core Network
- TCP/IP
- Simulation Parameters
- NS2
- Our Simulator
- Results & Discussion
- Conclusions & Future Work

From Mobile Networks to UMTS Revolution (1/6)



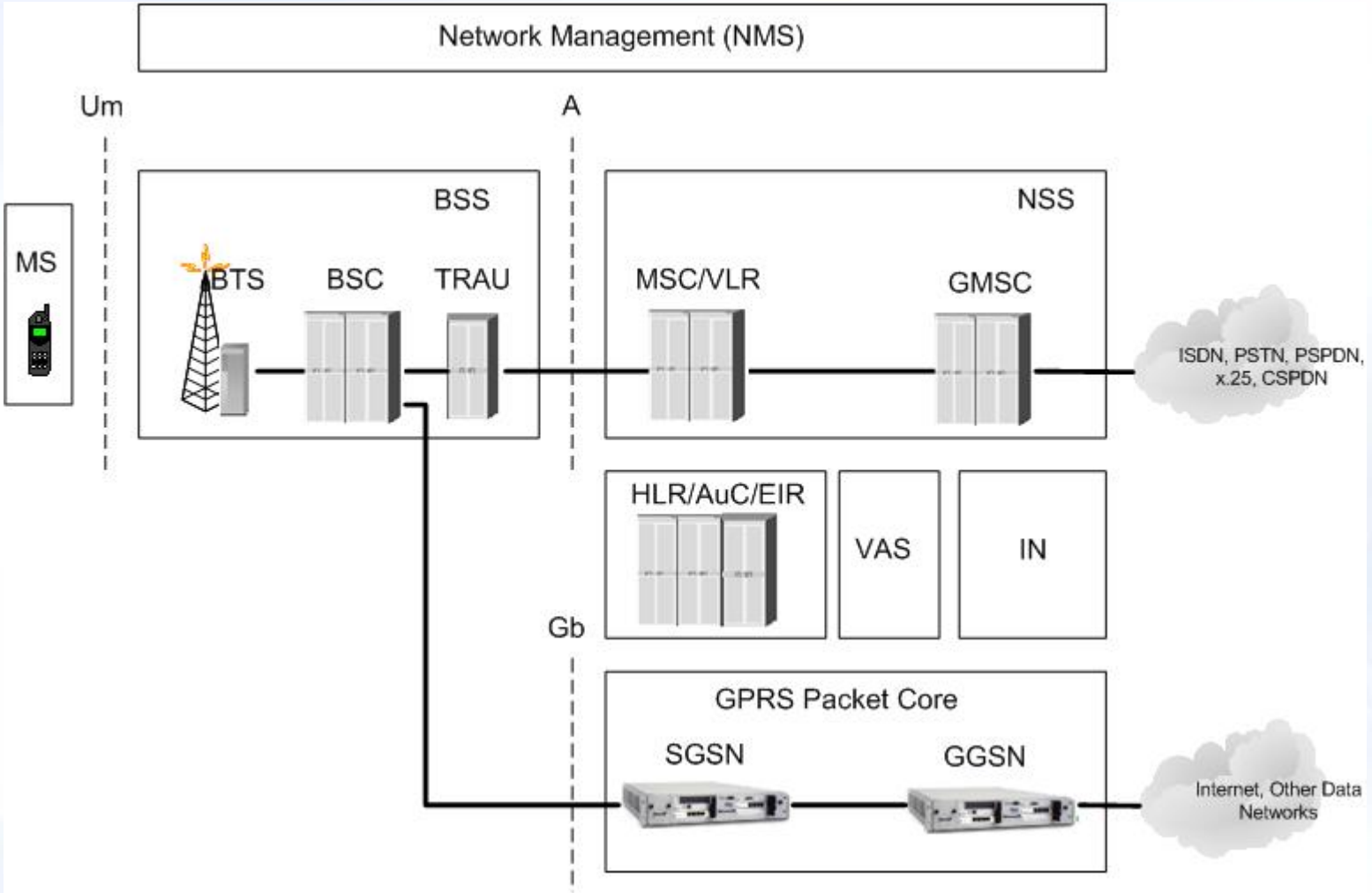
GSM Network

From Mobile Networks to UMTS Revolution (2/6)



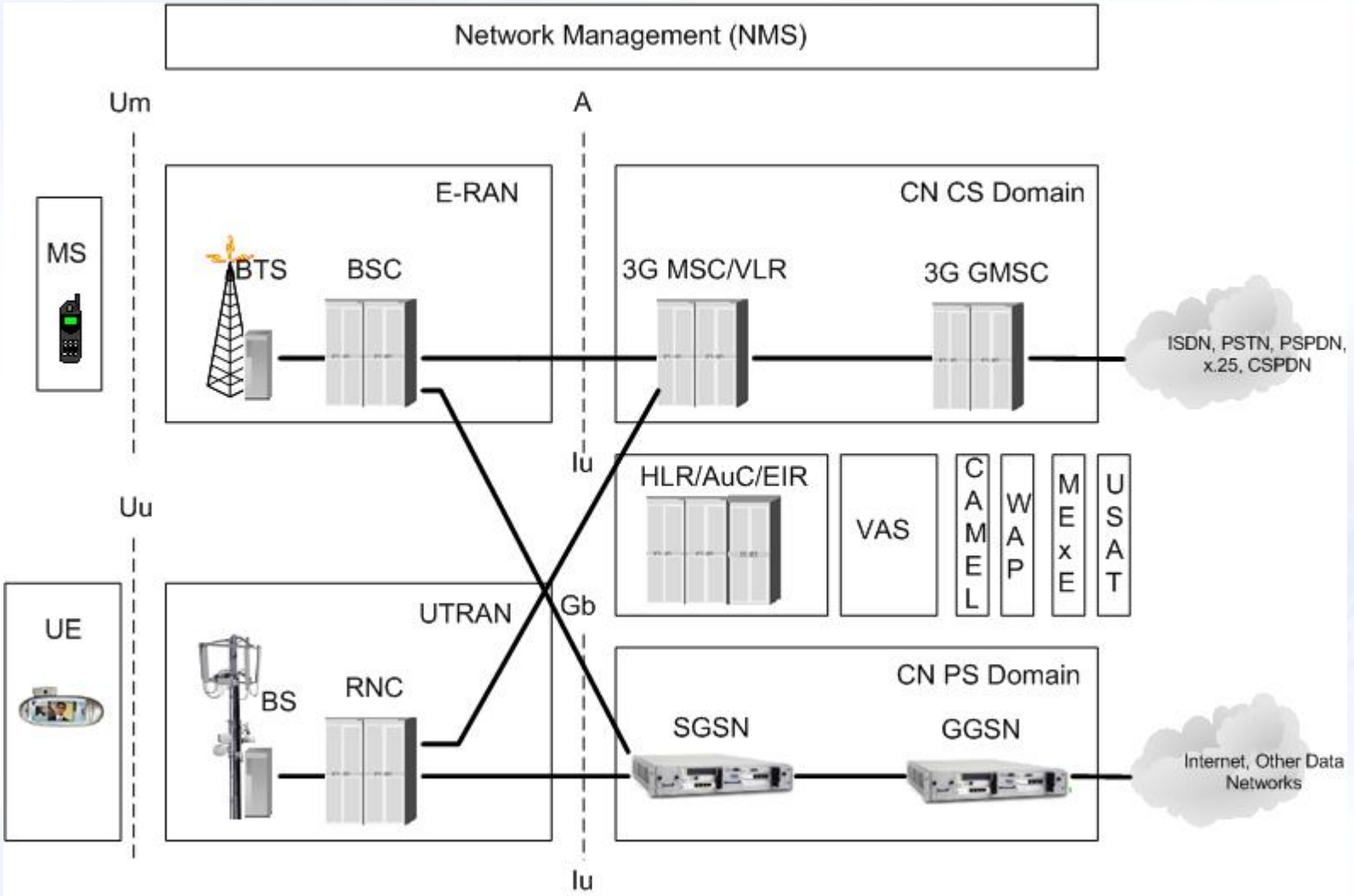
VALUE ADDED Service Platform

From Mobile Networks to UMTS Revolution (3/6)



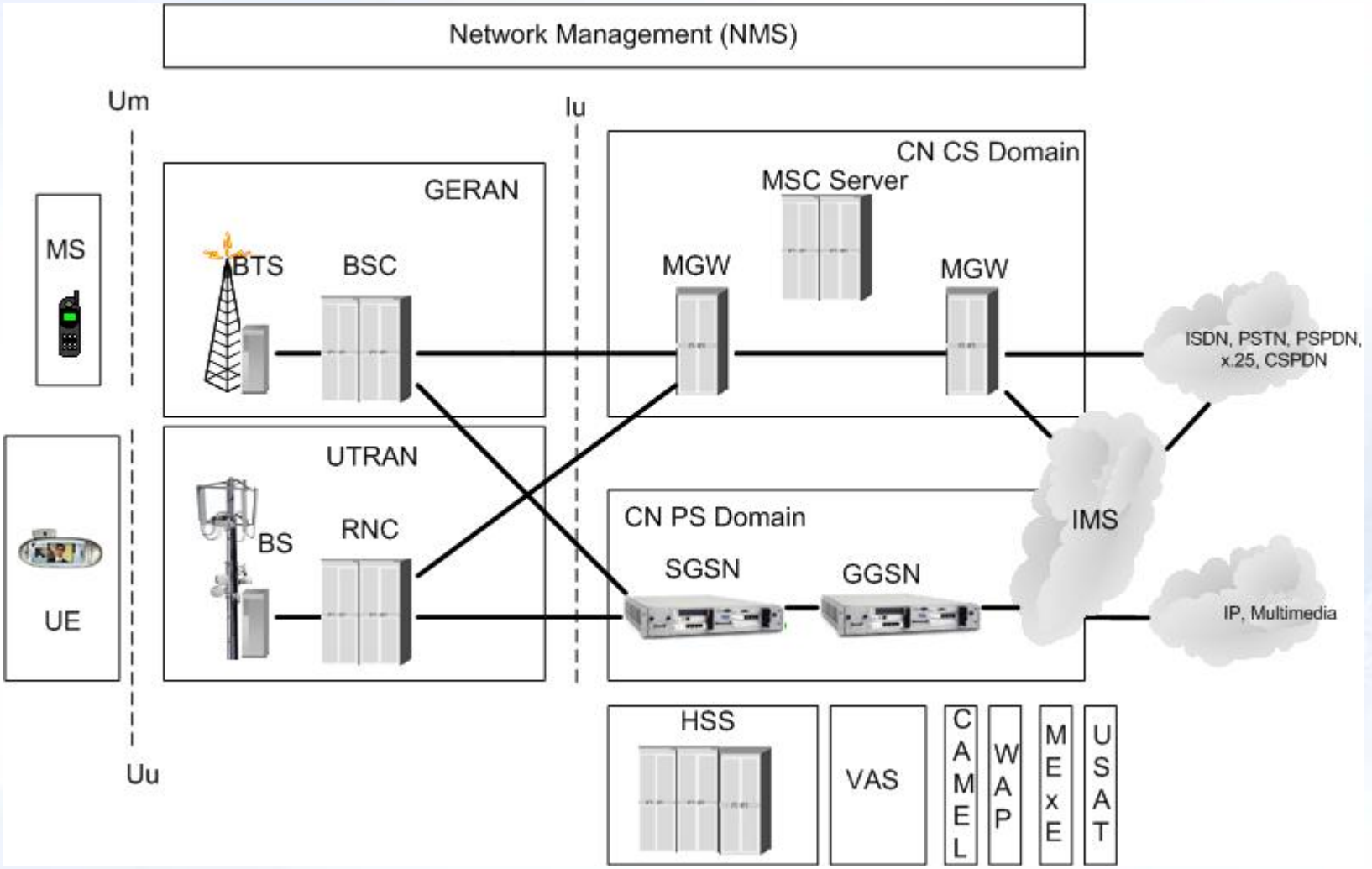
GPRS Service

From Mobile Networks to UMTS Revolution (4/6)



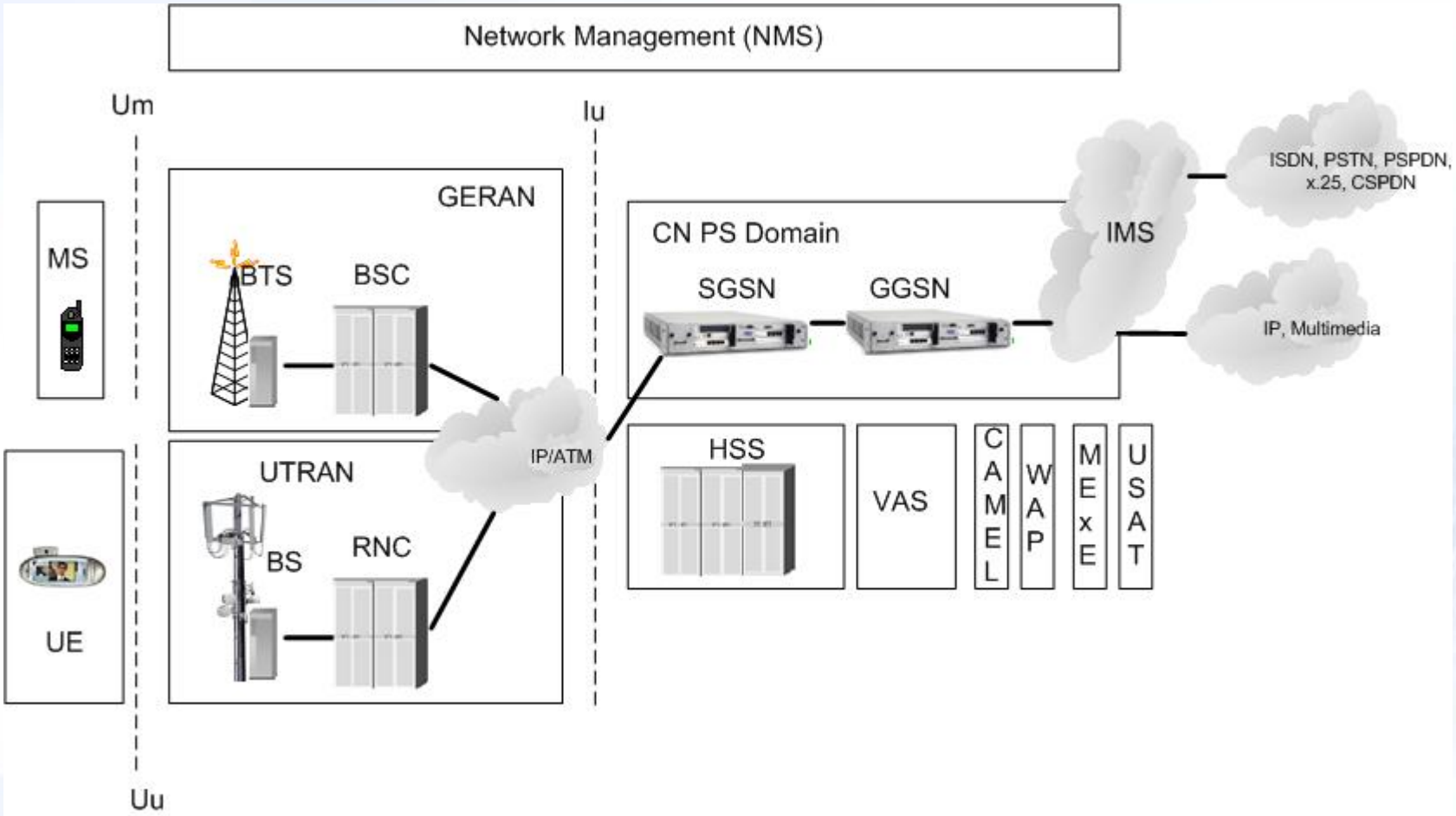
3GPP R99 Network

From Mobile Networks to UMTS Revolution (5/6)



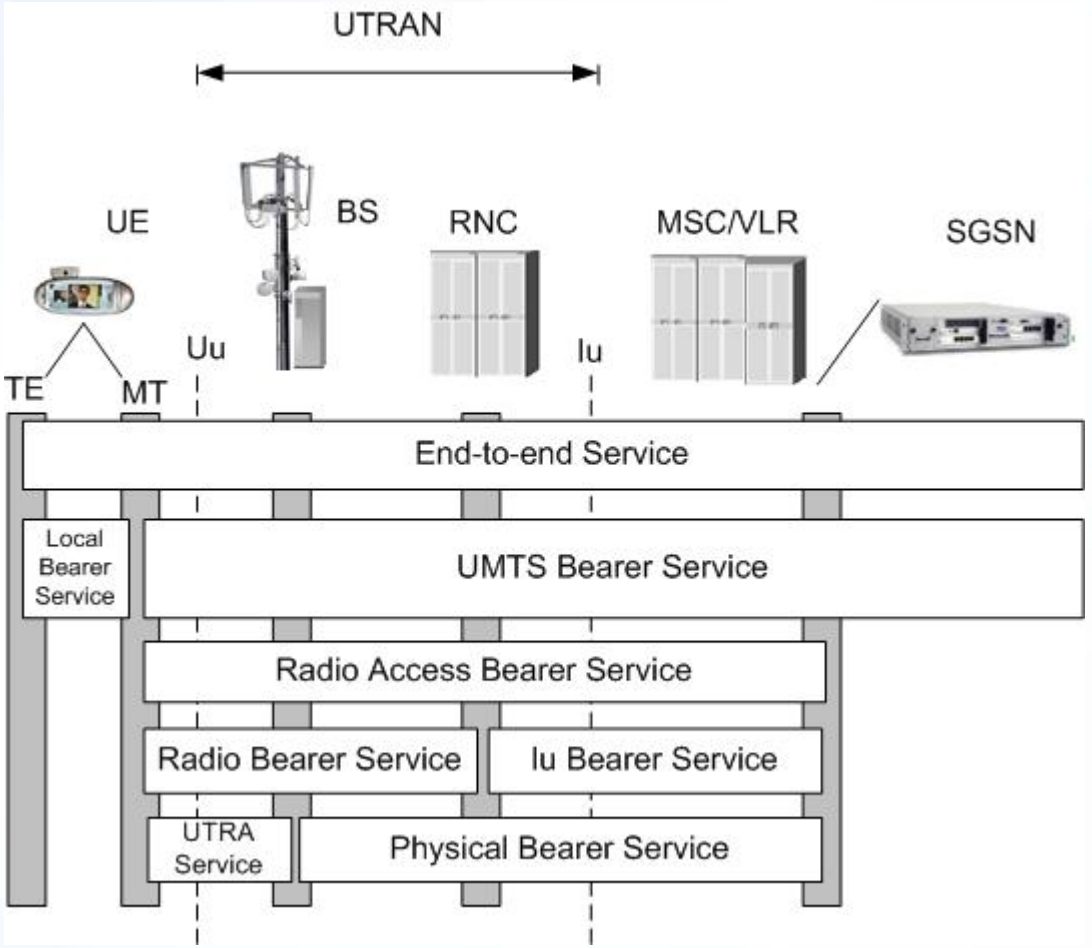
3GPP R4 Network

From Mobile Networks to UMTS Revolution (6/6)



3GPP R5 Network

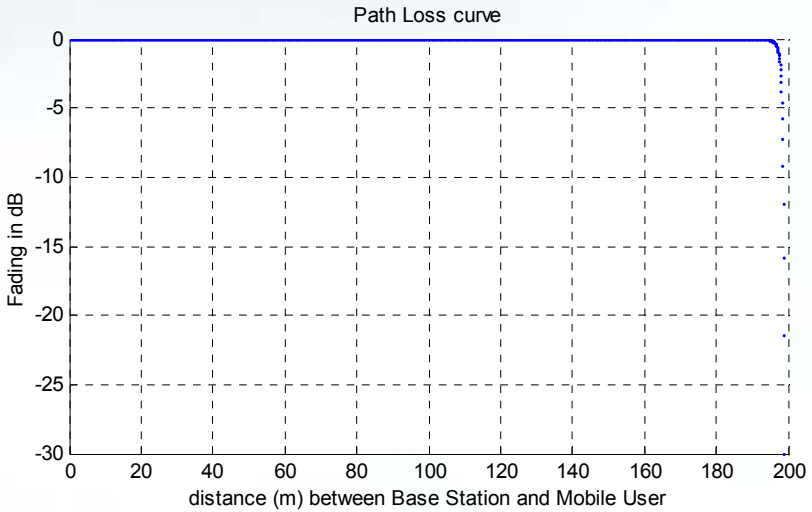
UMTS Radio Access Network (1/7)



UMTS Radio Access Network (2/7)

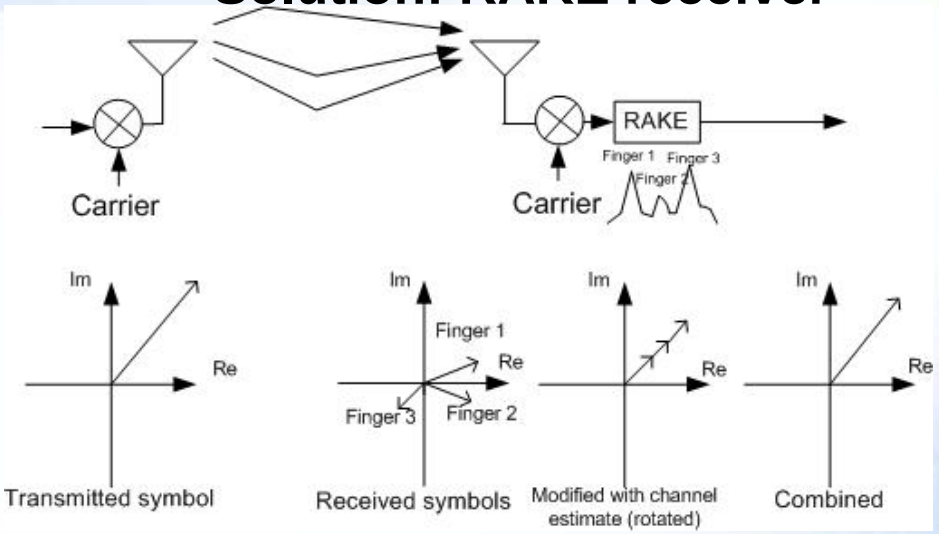
LOSS due to:

- Path Loss (distance)

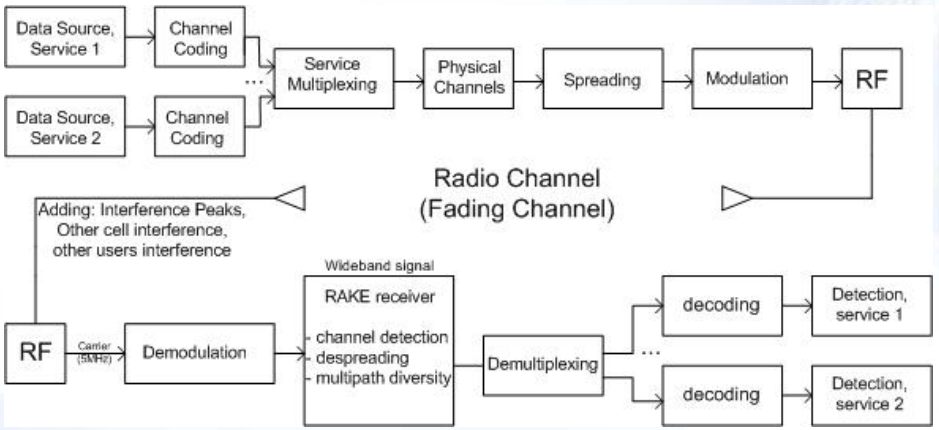


- Shadowing
- Doppler (movement)
- Multipath Transmission

Solution: RAKE receiver

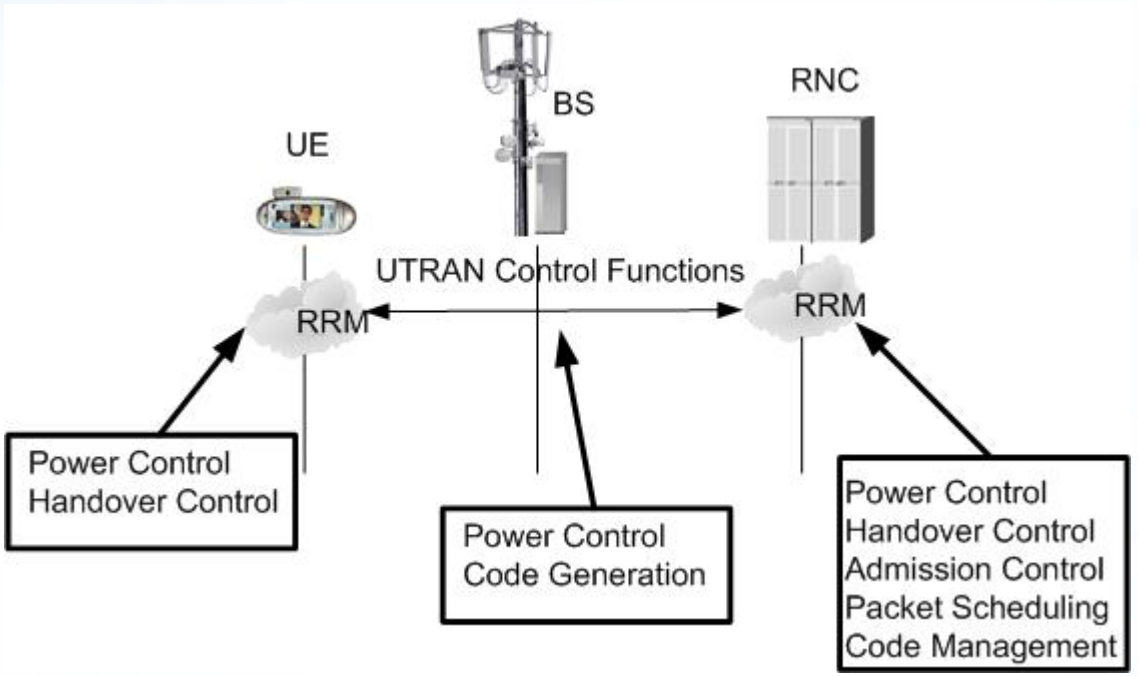


WCDMA Transmission:



UMTS Radio Access Network (3/7)

Resource Management



UMTS Radio Access Network (4/7)

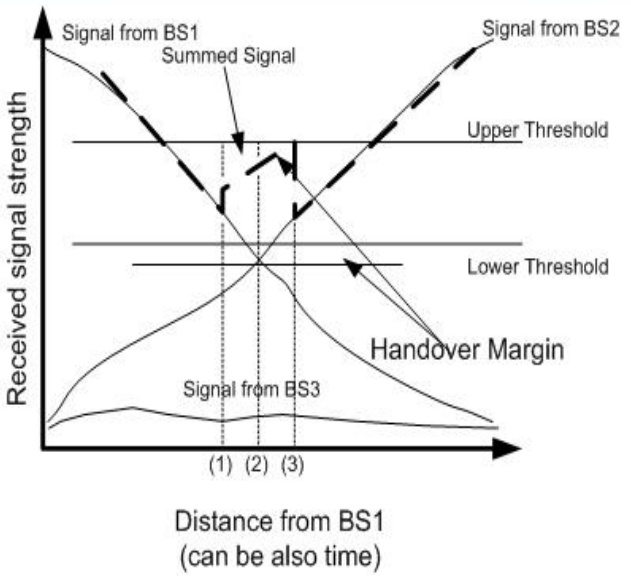
Handover:

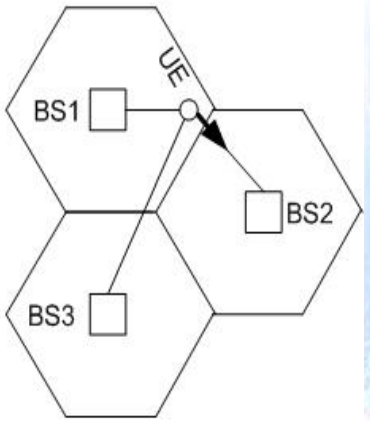
Softer where the User Equipment is connected to two sectors of the same Base Station simultaneously (no delays).

Soft where the User Equipment is connected to two sectors of different Base Stations simultaneously (no delays).

Hard where the User Equipment is connected to only one sector at the time.

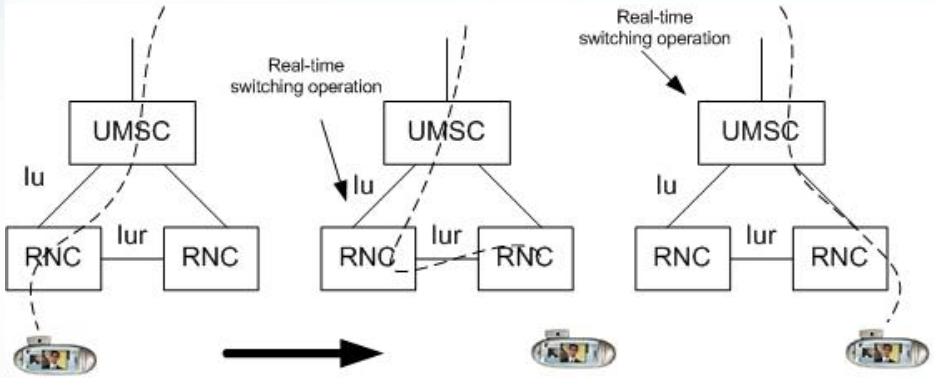
Algorithm:



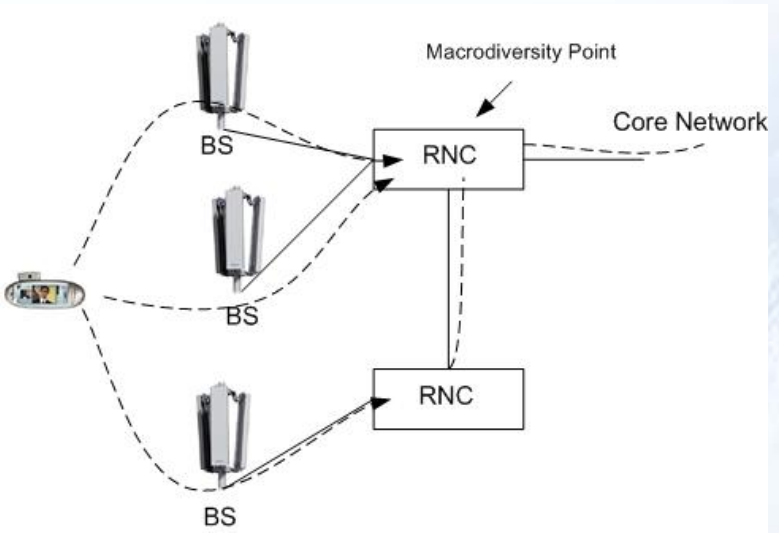


UMTS Radio Access Network (5/7)

RNC Changes:



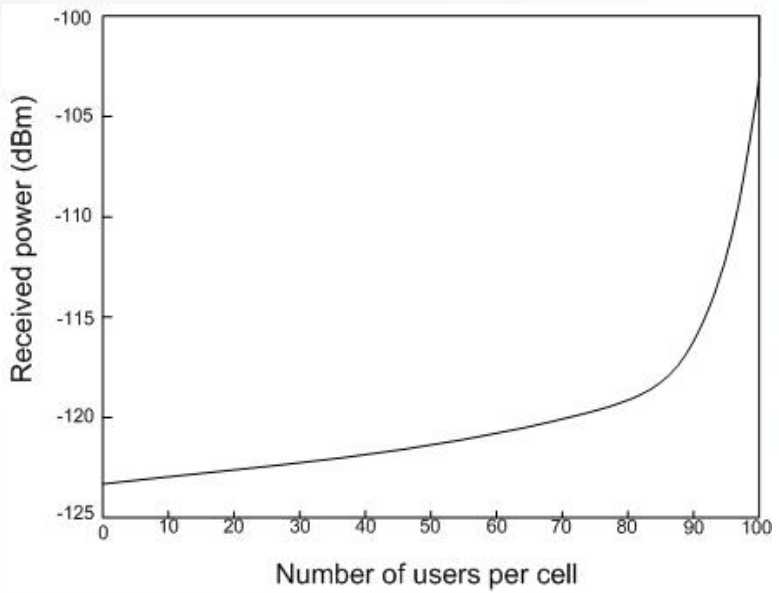
Macrodiversity:



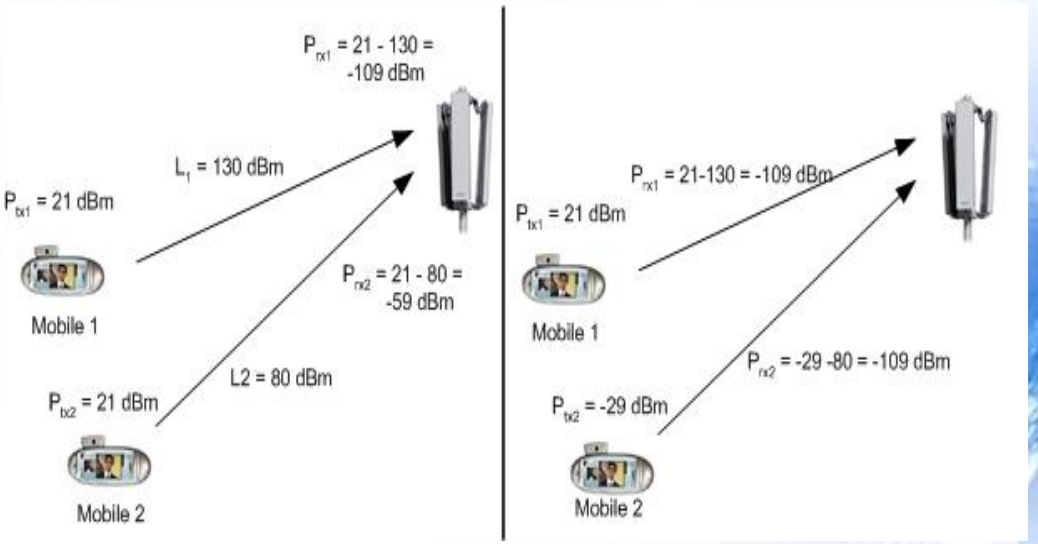
UMTS Radio Access Network (6/7)

Power Control :

WHY?



EXAMPLE



UMTS Radio Access Network (7/7)

Assumptions:

•All the subscribers under the TRX coverage are equally distributed so that they have equal distances to the TRX antenna.

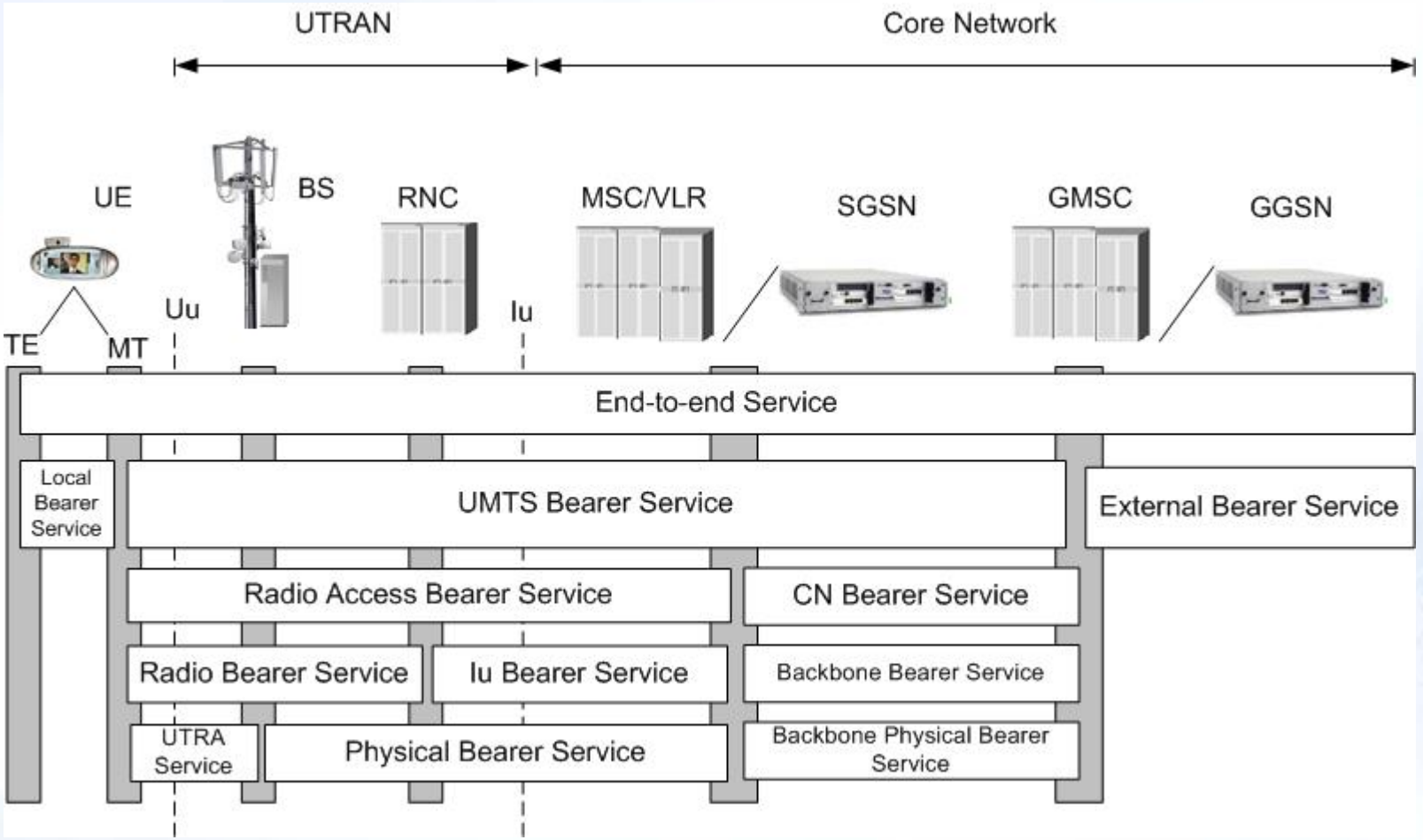
•The power level they use is the same and thus the interference they cause is on the same level.

•Subscribers under the TRX use the same base-band bit rate that is the same symbol rates.

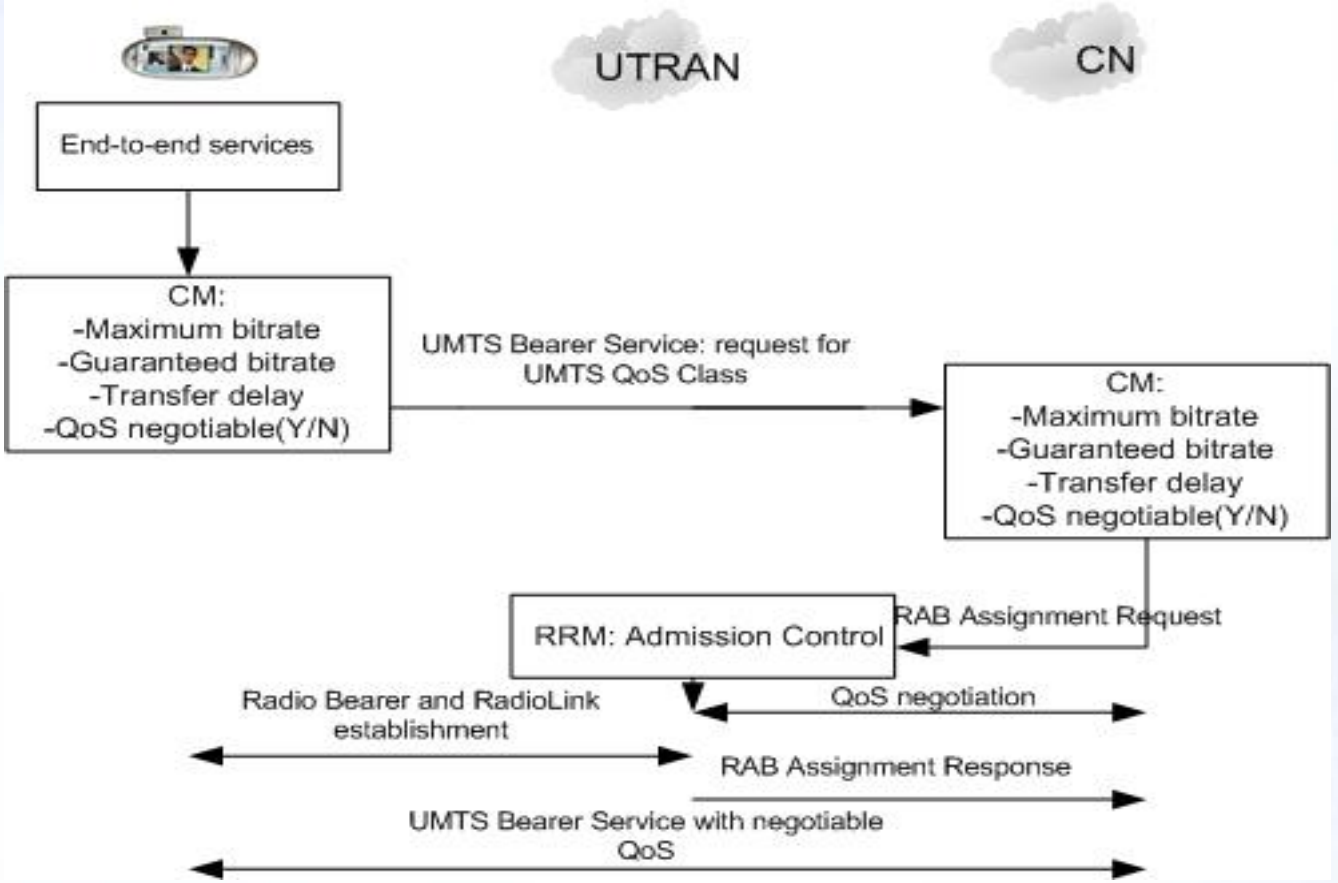
Speading Factor : $G_P = \frac{Chiprate}{Datarate}$

Number of users/cell: $X \approx \frac{G_P}{E_b / No}$

UMTS Core Network (1/2)



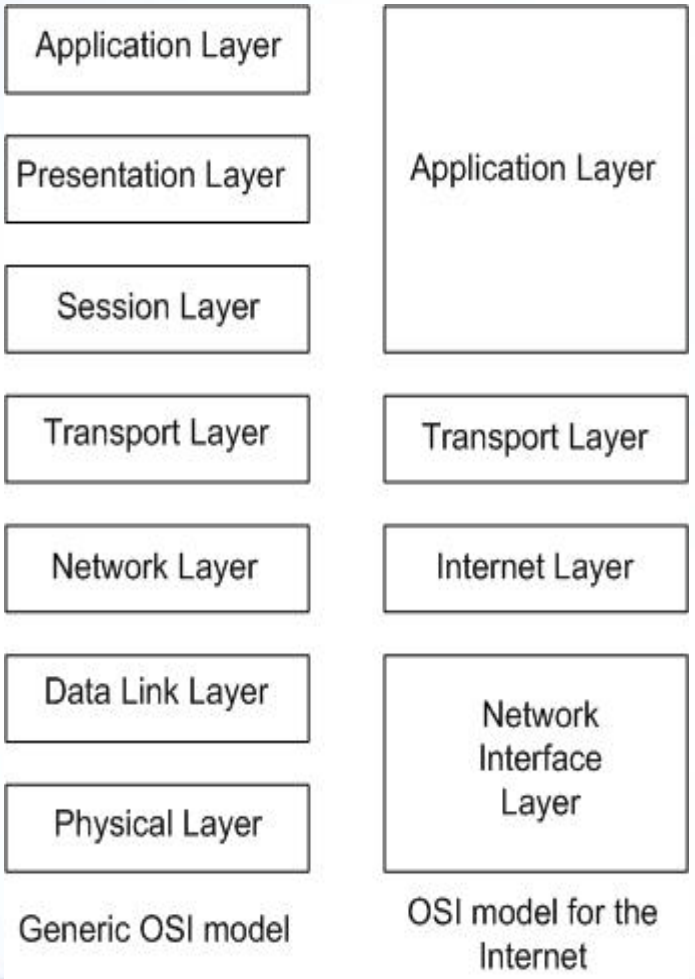
UMTS Core Network (2/2)



end-to-end Services

TCP/IP (1/12)

General OSI and OSI for the Internet



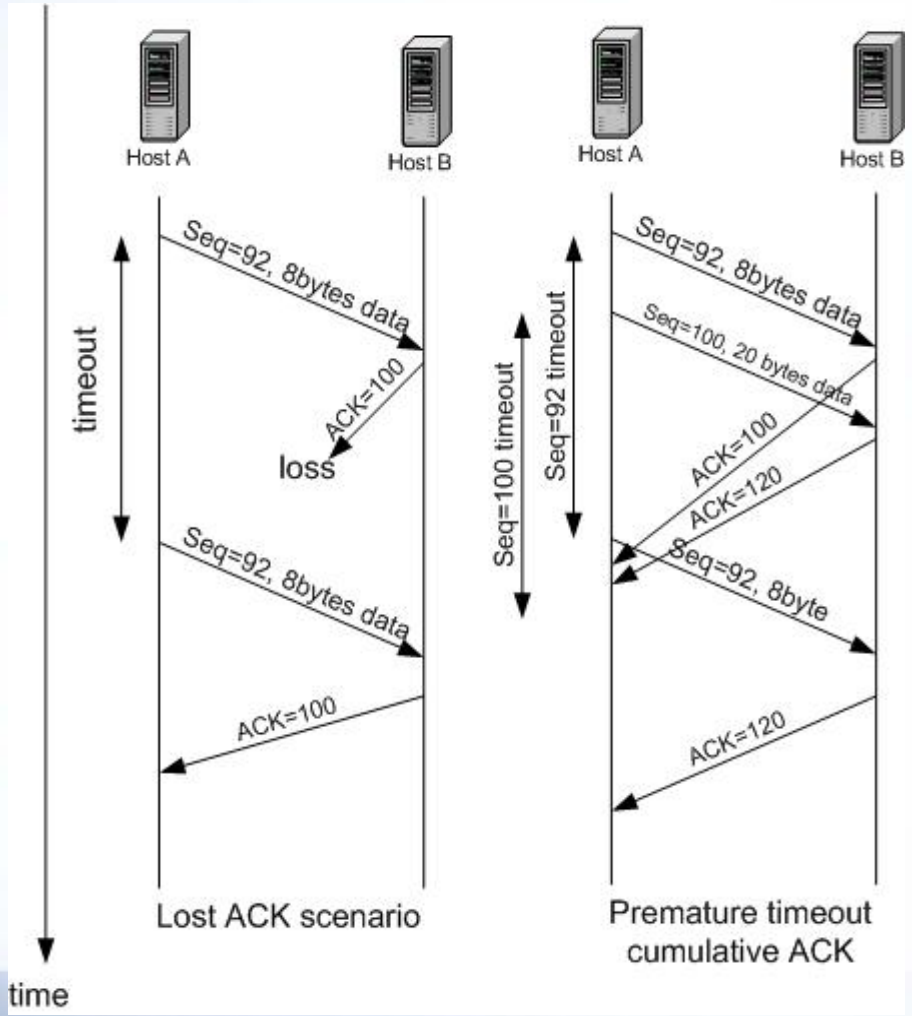
TCP/IP (2/12)

TRANSMISSION CONTROL PROTOCOL

- TCP is point-to-point protocol. This means that there is always one sender and one receiver. It is reliable and in-order in byte stream. Multicasting is not possible with TCP as it is.
- TCP is pipelined. This means that in TCP there is congestion window size is set.
- In TCP there are send and receive buffers.
- TCP uses full duplex data. There exists bi-directional data flow in the same connection. Moreover there is a Maximum Segment Size (MSS).
- TCP is connection-oriented. There exist handshaking that is exchange of control messages, initiating sender and receiver state before data exchange.
- At last but not least TCP is flow controlled, which means that sender will not overwhelm receiver.

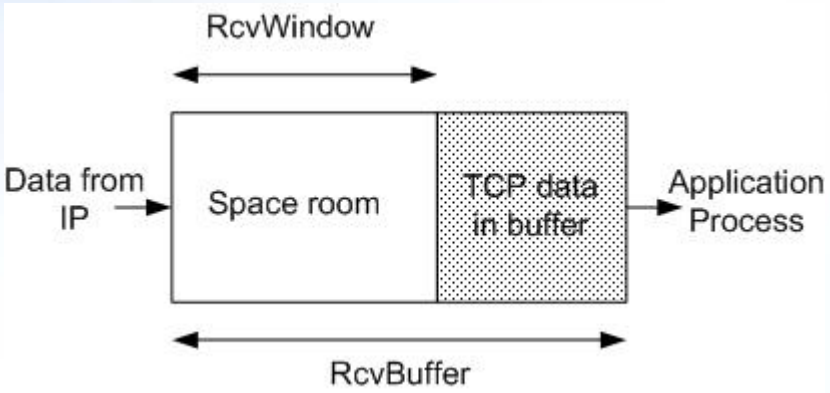
TCP/IP (3/12)

TCP Reliable data transfer and Retransmission



TCP/IP (4/12)

TCP Flow Control :



TCP Round Trip Time and Timeout :

$$EstimatedRTT = (1-x) \cdot EstimatedRTT + x \cdot SampleRTT$$

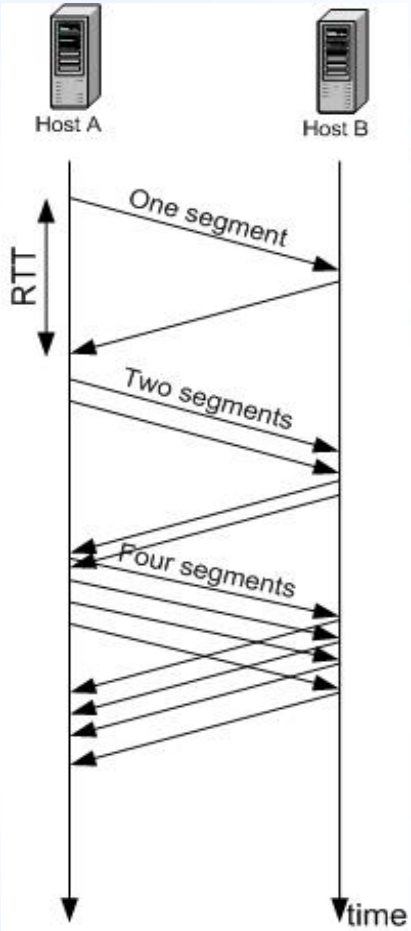
This average is an exponential weighted moving average. There is an influence of given sample decreases exponentially fast. A typical value for x is 0.125.

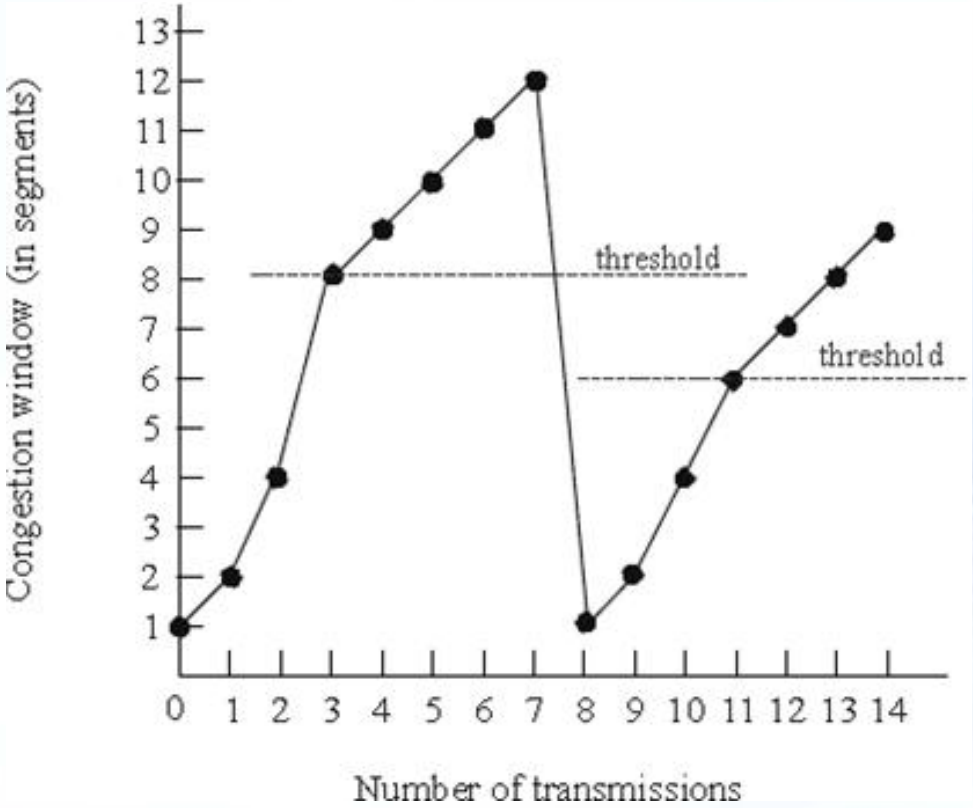
$$Timeout = EstimatedRTT + 4 \cdot Deviation$$

$$Deviation = (1-x) \cdot Deviation + x \cdot |\, SampleRTT - EstimatedRTT \,|$$

TCP/IP (5/12)

TCP Congestion Control:





TCP/IP (6/12)

TCP Tahoe:

- TCP Tahoe is the oldest version of TCP
- TCP Tahoe congestion algorithm includes Slow Start and Congestion Avoidance.
- In order to overcome a loss event three timeouts have to be passed. This is its main drawback as when a segment is lost, the sender side of the application may have to wait a long period of time for the timeout. It implements an RTT-based estimation.

TCP/IP (7/12)

TCP Reno:

- TCP Reno except from Slow Start and Congestion Avoidance includes also Fast Retransmit. In Fast retransmit mechanism, three duplicate acknowledgements carrying the same sequence number triggers a retransmission without waiting for the associated timeout event to occur. The window adjustment strategy for this early timeout is the same as for the regular timeout and Slow Start is applied.
- The problem, however, is that the Slow Start is not always efficient, especially if the error was purely transient or random in nature and not persistent. In such a case the shrinkage of the congestion window is, in fact, unnecessary and renders the protocol unable to fully utilize the available bandwidth of the communication channel during the subsequent phase of window re-expansion.

TCP/IP (8/12)

TCP NewReno:

- TCP NewReno introduces Fast Recovery in conjunction with Fast Retransmit. The idea behind Fast Retransmit is that an ACK is an indication of available channel bandwidth since a segment has been successfully delivered. This, in turn, implies that the congestion window should actually be incremented upon one ACK delivery. Then instead of entering Slow Start, the sender increases its current congestion window by the threshold number.
- TCP NewReno's Fast Recovery can be effective when there is only one segment drop from a window of data, given the fact that NewReno retransmits at most one dropped segment per RTT. The problem with the mechanism is that is not optimized for multiple packet drops form a single window, and this could negatively impact performance.

TCP/IP (9/12)

TCP Vegas:

- TCP Vegas approaches the problem of congestion from another perspective. The basic idea is to detect congestion in the routers between source and destination before packet loss occurs and lower the rate linearly when this imminent packet loss is detected. The longer the round-trip times of the packets, the greater the congestion in the routers. Every two round trips delays the following quantity is computed:

$$\rho = (\text{WindowSizeCurrent} - \text{WindowSizeOld}) \cdot (\text{RTTCurrent} - \text{RTTOld})$$

If $\rho > 0$ the window size is decreased by 1/8 . Else the window size is increased by one minimum segment size.

- One problem that it does not seem to overcome is the path asymmetry. The sender makes decisions based on the RTT measurements, which, however, might not accurately indicate the congestion level of the forward path. Furthermore, packet drops caused by retransmission deficiencies or fading channels may trigger a Slow Start.

TCP/IP (10/12)

TCP SACK:

- TCP Selective Acknowledgements (SACK) is a TCP enhancement which allows receivers to specify precisely which segments have been received even in the presence of packet loss. TCP SACK is an Internet Engineering Task Force (IETF) proposed standard which is implemented for most major operating systems.
- SACK enables receiver to give more information to sender about received packets allowing sender to recover from multiple-packet losses faster and more efficiently. On the contrary TCP Reno and New-Reno can retransmit only one lost packet per round-trip time because they use cumulative acknowledgements.

TCP/IP (11/12)

Latency:

If $W\frac{S}{R} > RTT + \frac{S}{R}$ Then Latency = 2 RTT + O/R

Else

$$\text{Latency} = 2RTT + O/R + (K-1)\left[S/R + RTT - W\frac{S}{R}\right]^{+}$$

Where $K = \left\lceil \log_2\left(\frac{O}{S} + 1\right) \right\rceil$

Error adds Latency !

TCP/IP (12/12)

Discussion

Even today's wired internet has so many problems that some people persist to call it the World Wide Wait. These problems will enlarge in the future Mobile Internet. Although TCP was introduced years ago on the wired internet it will continue to be THE transfer protocol. Its hard defensive behavior towards any kind of loss will be its weakest point over wireless and mobile networks. TCP can not infer if an error is due to traffic congestion or losses over a wireless link. Our approach to this problem is to find the proper version of TCP to each profile of application and user over the UMTS network. Notice that the UMTS network is implemented using TCP and end-to-end QoS.

Simulation Parameters (1/6)

NETWORK PARAMETERS

PARAMETER	VALUE
Number of cells	7 (hexagonal)
Cell Side	200m
Path loss propagation exponent β	3.5
Path loss propagation parameter A	–30dB
Shadowing parameter σ	4dB
Number of oscillators (Jakes)	8
Number of multipath rays	5
Noise	-132 dbW
Doppler frequency	0,6,20,80 Hz
Chip rate	3.84 Mcps
PC frequency	1500 Hz
PC power range	80dB
PC step	0.5 dB
PC SIR objective	2.5 to 3 dB
PC loop delay	1,2,3,4,10
Maximum TX Power	-16 dBW
Packet length(MTU)	1000 bytes

Simulation Parameters (2/6)

Users with 120 Kbps bit rate

Spreading Factor = 32

Approximately 16 users per cell
(with one TRX per cell)

Approximately 110 users total

Approximately 330 users total
if there are 3 TRX per cell

Performance is declined rapidly if
total users > 160

RTT ≈ 120 msecs
(but can be increased)

Users with 240 Kbps bit rate

Spreading Factor = 64

Approximately 8 users per cell
(with one TRX per cell)

Approximately 55 users total

Approximately 160 users total
if there are 3 TRX per cell

Performance is declined rapidly if
Total users > 100

RTT ≈ 80 msecs
(but can be increased)

Simulation Parameters (3/6)

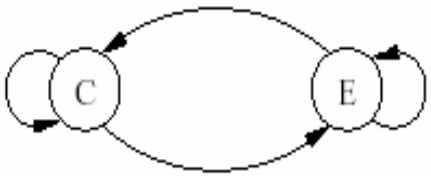
Error Statistics over UMTS (WCDMA) air interface

Let X(i) = E if the block I is in error (which occurs with probability Pe(i)) and C if it is correct.
For an ergodic process X(i) we have:

P[burst length k] = P[X(t)=E,t=2,…,k|X(0)=C,X(1)=E]

$$= \frac{P[X(0)=C, X(t)=E, t=1,..,k]}{P[X(0)=C, X(1)=E]}$$

The above model can be described by a Two-State Markov error model as shown in the following Figure :



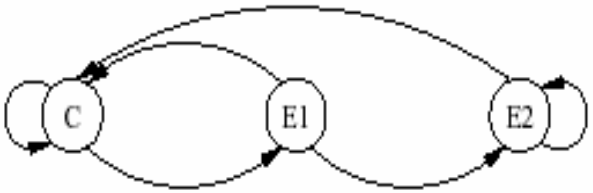
The model is fully characterized by its transition matrix: $P = \begin{bmatrix} p_{CC} & p_{CE} \\ p_{EC} & p_{EE} \end{bmatrix}$ and

$P(E) = \frac{p_{CE}}{p_{CE}+P_{EC}}$, P[burst length>1] = P[E|E] = p_{EE} , P(E|E) = $p_{EE} = \frac{p[burstlength > k]}{p[burstlength > k-1]}$

Simulation Parameters (4/6)

Error Statistics over UMTS (WCDMA) air interface

Simple as it is the two-state Markov Model does not capture all types of behavior. An obvious way to get around this problem is use a multistate Markov Model. We therefore further restrict ourselves to a three-state model such as the one shown in the next figure



Two error states are now present and the second of them, E2, can only be entered from the first, E1. Also exiting E2 necessarily, leads to the Correct State (not to E1). The transmission matrix for such a model is as follows:

$$P = \begin{bmatrix} p_{CC} & p_{C1} & 0 \\ p_{1C} & 0 & p_{12} \\ p_{2C} & 0 & p_{22} \end{bmatrix}$$

And

$P[E]=P[E1]+P[E2]=a_0$

$P[\text{burst length}>1]=p_{12}= a_1$

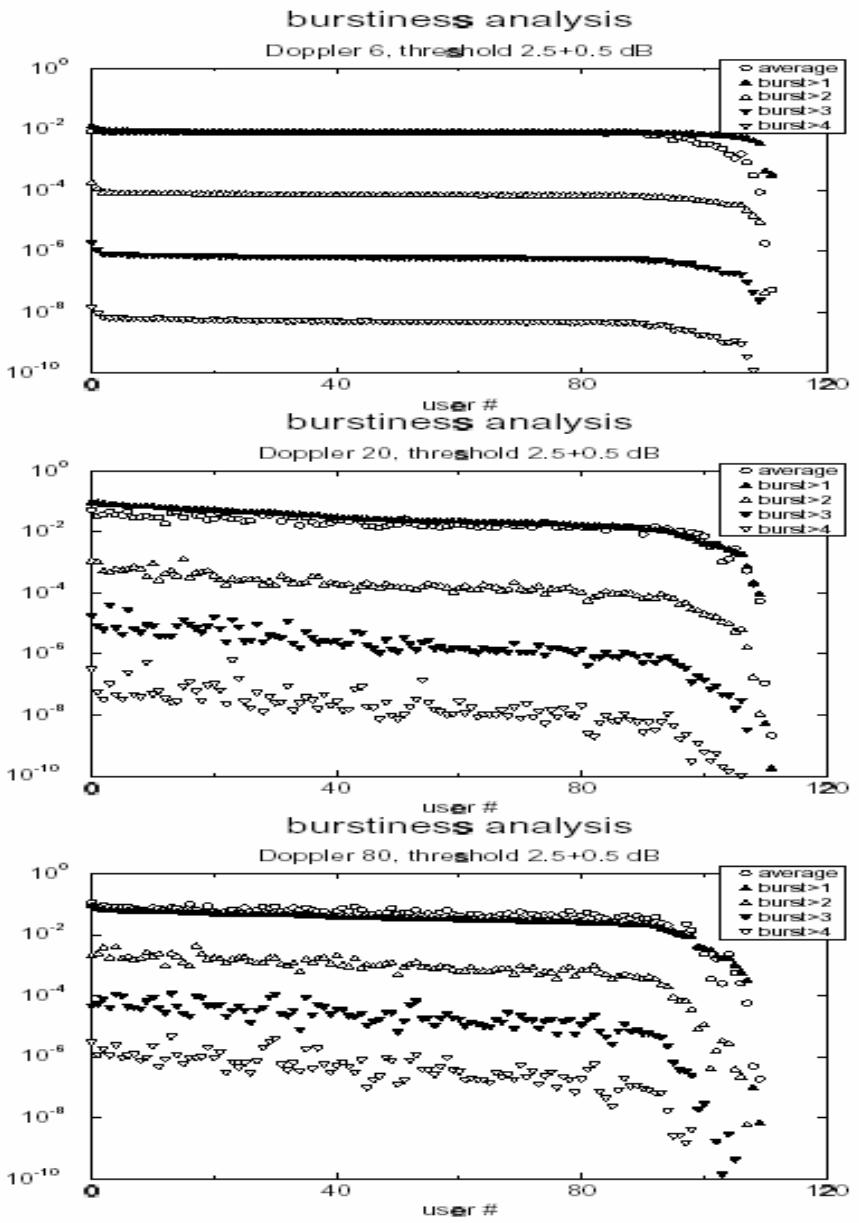
$$\frac{p[burstlength > 2]}{p[burstlength > 1]} = p_{22} = a_2$$

Finally $P(C)=P(E1)/p_{C1}$ where

$$p_{C1} = \frac{P(E)\cdot(1-p_{22})}{(1-P(E))\cdot(1-p_{22}+p_{12})}$$

$$P(E1) = \frac{P(E)\cdot(1-p_{22})}{(1-p_{22}+p_{12})}$$

Simulation Parameters (5/6)



•The results in the first plot, for slow fading, show that most users will see independent errors, since power control equalizes SIR so well that all randomness due to fading and interference is absorbed. Also, the fading randomness equalizes performance across most users, since all will essentially see the same average. This behavior resembles the behavior observed in the **two-state Markov model**.

•As expected, as fading rate increases everything gets more mixed, but a similar qualitative trend can still be observed. Note that a positive burst correlation can be observed i.e. P[E|E] ≈ P[burst>1]>P[E] in most cases for values of the Doppler up to 20Hz. This indicates that the system tends to stay in the bad state i.e. errors tend to be clustered, which is intuitive pleasing since the channel has memory. This resembles to the behavior observed in the **three-state Markov model**.

•On the other hand, for fast fading the opposite is observed, i.e. the system tends to escape from error states. This can be explained by noting that the power control tends to increase the transmitted power when an error is observed while at the same time the dynamics of the channel tend to make the channel exit quickly from bad conditions, so that right after an error the probability that a transmission is successful is higher than average. We refer to this case in which errors tend to occur isolated as the case of “anticorrelated” errors. This resemble the behavior observed in the **two-state Markov model**

Simulation Parameters (6/6)

TCP Improvements Proposed for 3G

- Large Window Size (Sender and Receiver)
- Increased Initial Window (Sender). The initial CWND must be equal to min (4 MSS , max (2 MSS , 4380 bytes))
- Limited Transmit (Sender)
- IP larger than Default:

  576bytes (min IP compatible) < MTU < 1500bytes

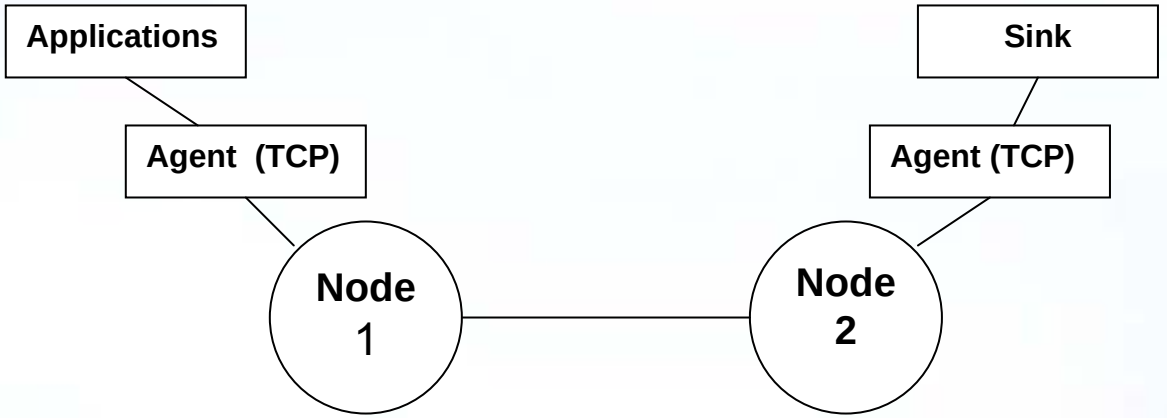
  Typical 1000 bytes (12/1000 bytes = only 1,2% overhead)
- MTU Discovery (Sender and Intermediate Routers)
- Explicit Congestion Notification activated (Sender, Receiver, Routers) for congestion avoidance.

NS 2 (1/2)

- In order not to “reinvent the wheel”, as base for our simulations we use the widely used Network Simulator (NS) version 2 (2.1b9).
- NS2 is THE Network Simulator, widely used from the network community, providing procedures for all versions of TCP and TCP applications.
- It is an object-oriented, discrete event driven network simulator developed at Lawrence Laboratories - UC Berkely (maintained by Information System Institute-ISI) written in C++ and OTcl and running in the UNIX operating system.
- All simulation results can be viewed by the Network Animator (NAM) which is a simulation display tool.

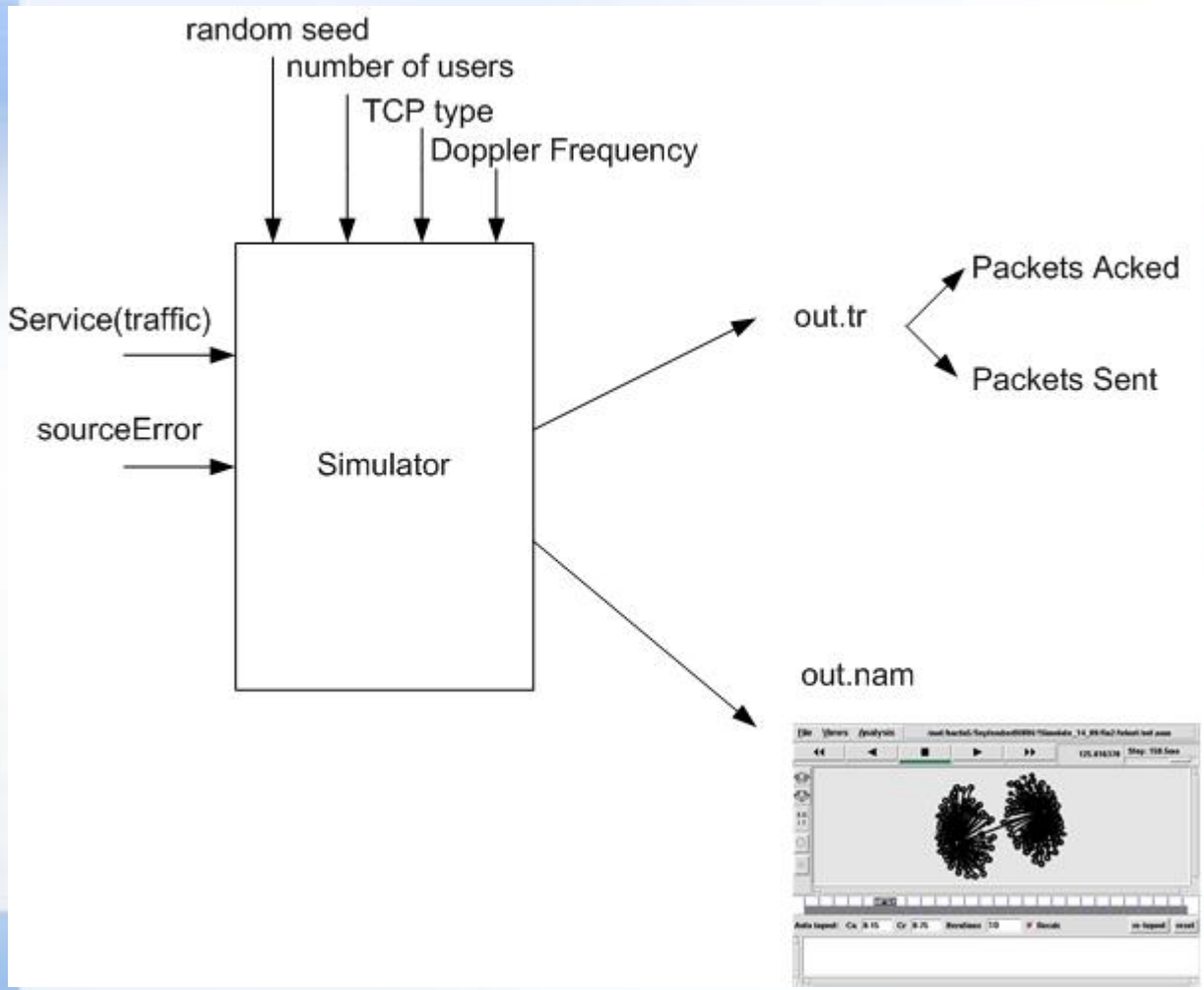
NS 2 (2/2)

How it works



By observing the intermediate nodes or edge nodes we can examine the behavior of the network and applications. Statistics can also be taken

Our Simulator (1/5)

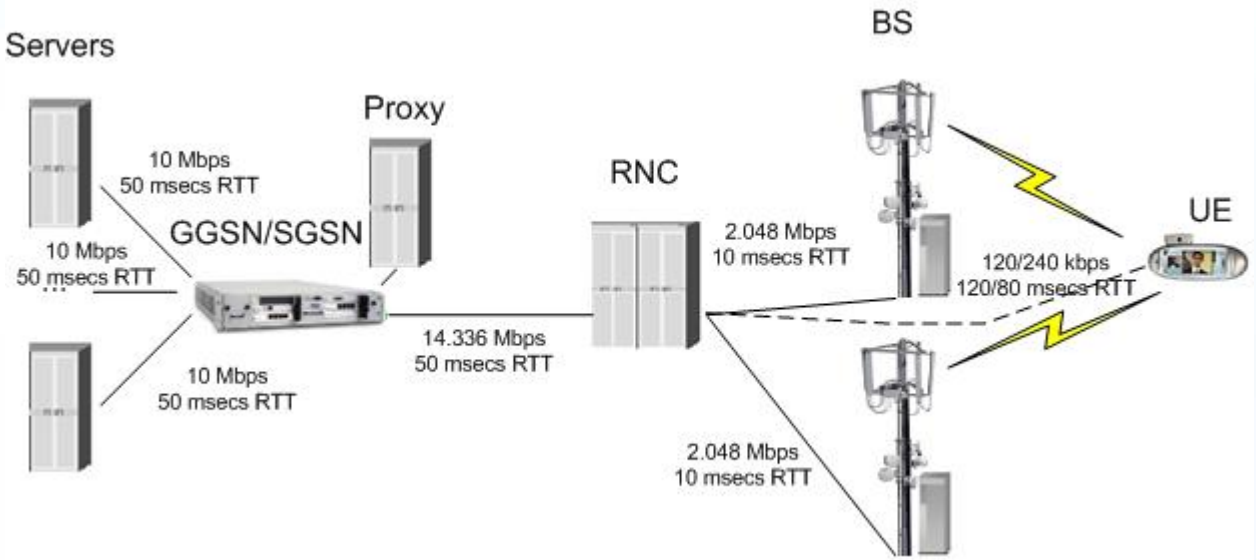


By monitoring the intermediate and edge nodes, NS provides specific output format. By using pattern recognition tools: awk and cat we can determine the number of packets sent and correctly acked end-to-end.

With NAM tool we can have a visible representation of topology and behavior of our simulations

Our Simulator (2/5)

General Topology of our Simulations



Our Simulator (3/5)

More Specifications

We use three TCP Applications: FTP, Telnet and HTTP

- FTP: Poisson Request adapted per user
- Telnet: Poisson Request adapted per user
- HTTP: HTTP/1.1 with Proxy functionalities
  - Inter-page interval exponential (mean duration) 1 second
  - Number of Page (mean duration) 4
  - Inter object interval exponential (mean duration) 0.01 second
  - Number of packets per object Pareto distributed shape 1.2 and scale 10

Our Simulator (4/5)

We examine TCP Performance, Energy Consumption and Application Efficiency for users with 120 and 240 kbps

- for each version of TCP
- for Doppler 6 , 20 and 80 Hz
- Duration of simulation : 600 seconds

- TCP Performance : $\frac{TotalPacketsCorrectlyAcked(TCP\_Version)}{TotalPacketsCorrectlyAcked(TCP\_Tahoe)}$

Our Simulator (5/5)

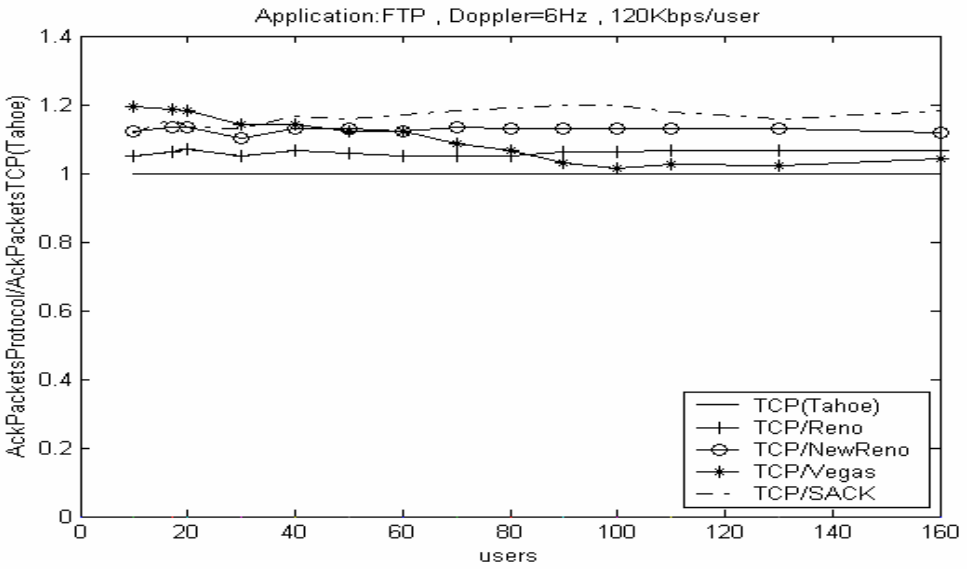
- TCP Energy Consumption:
  (or Resource Consumption including Power(WCDMA), CPU,RAM)

$$\frac{TotalPacketsCorrectlyAcked(TCP\_Version)}{TotalPacketsSent(TCP\_Version)}$$

- Efficiency of TCP Applications:
  (literature has shown that this factor is load independent)

$$\frac{TotalPacketsSent(TCP\_Version)}{TotalPacketsSent(TCP\_Version)no-errors}$$

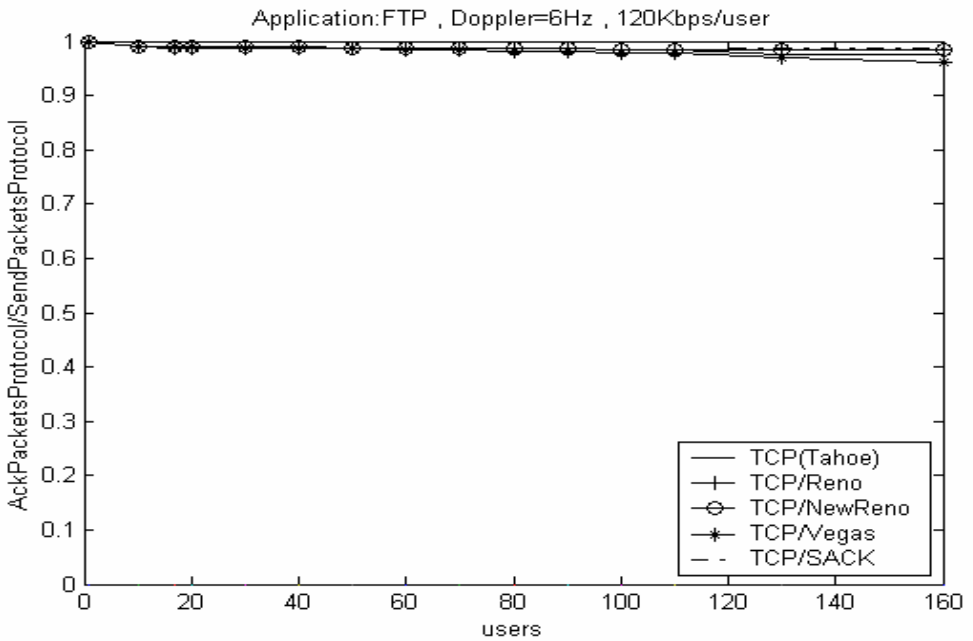
Results and Discussion (1/14)



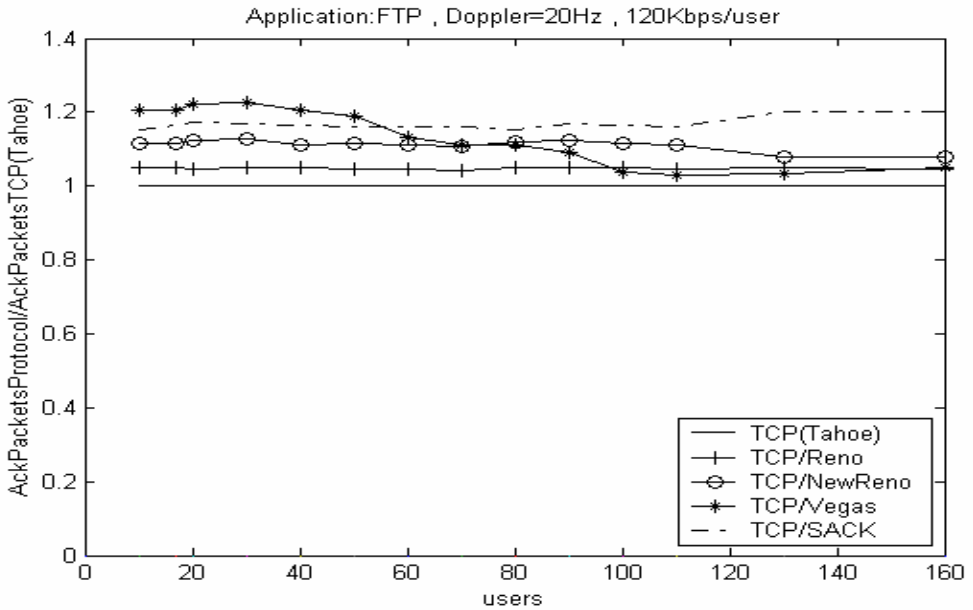
TCP Vegas outperforms only when load is light

If number of users > 30
TCP SACK outperforms
And TCP New Reno follows

TCP SACK outperforms slightly
In all cases and especially as number of users increases



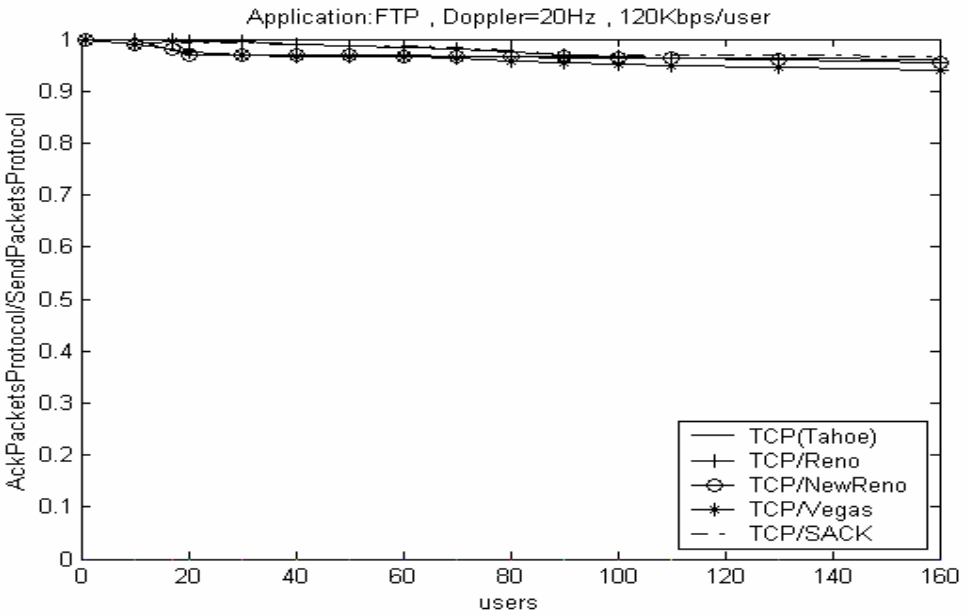
Results and Discussion (2/14)



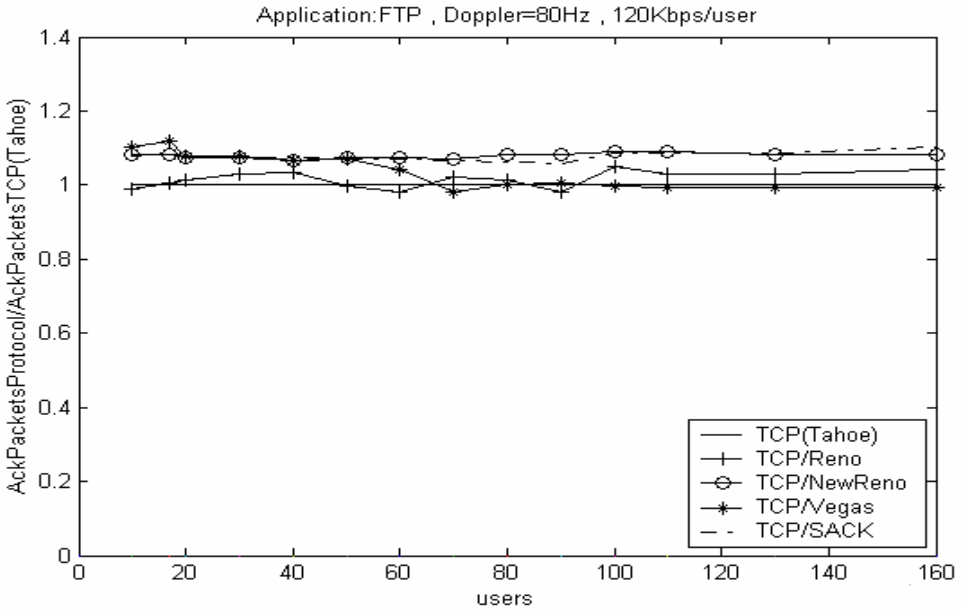
TCP Vegas outperforms only when load is light

If number of users > 50
TCP SACK outperforms
And TCP New Reno follows

TCP SACK outperforms slightly
In all cases and especially as number of users increases



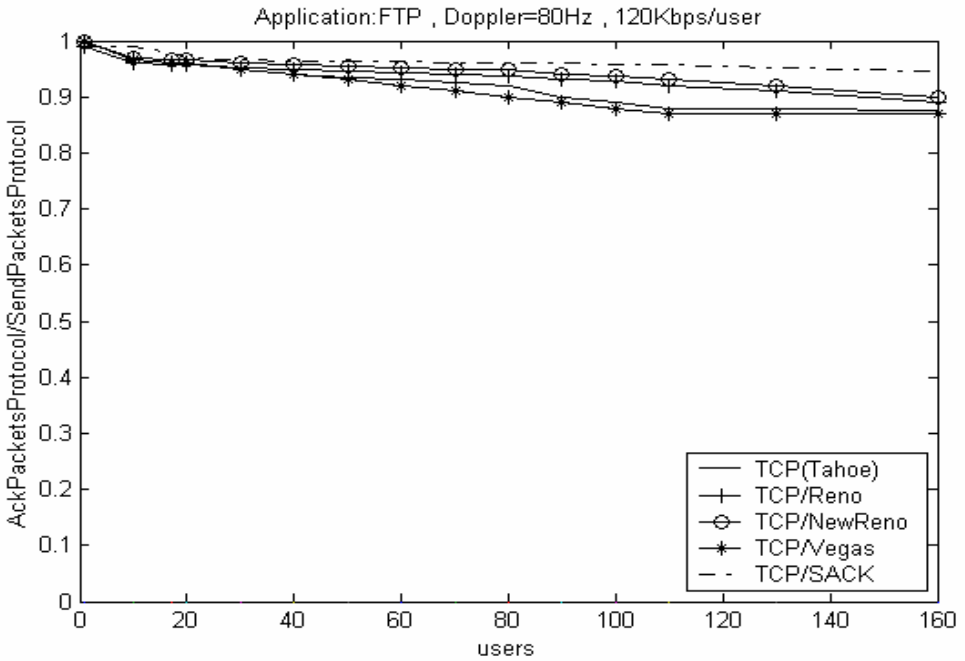
Results and Discussion (3/14)



TCP Vegas outperforms only when load is very light

If number of users > 20
TCP SACK and TCP New Reno share the same performance

TCP SACK outperforms slightly
In all cases and especially as number of users increases when the difference is obvious

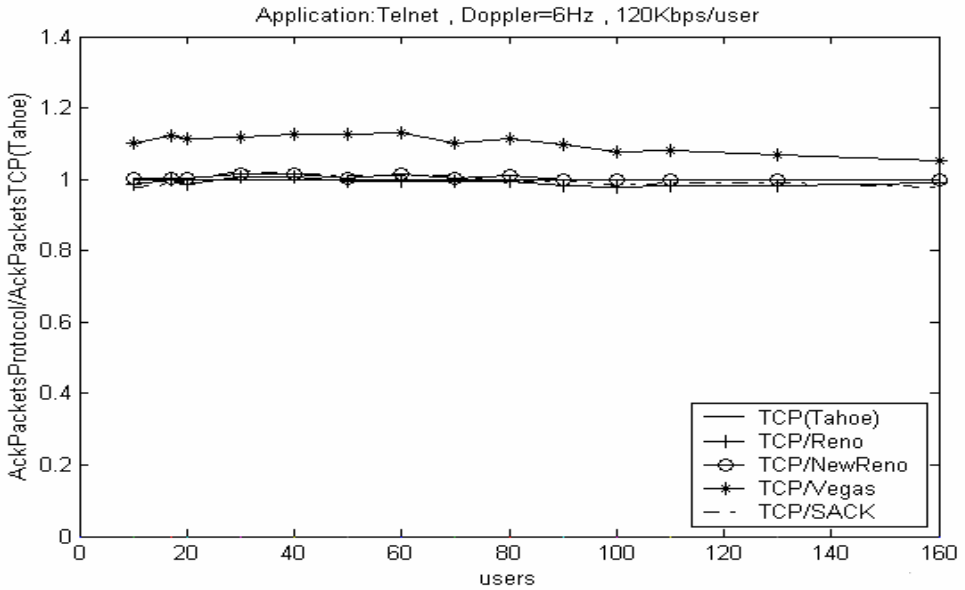


Results and Discussion (4/14)

FTP Application

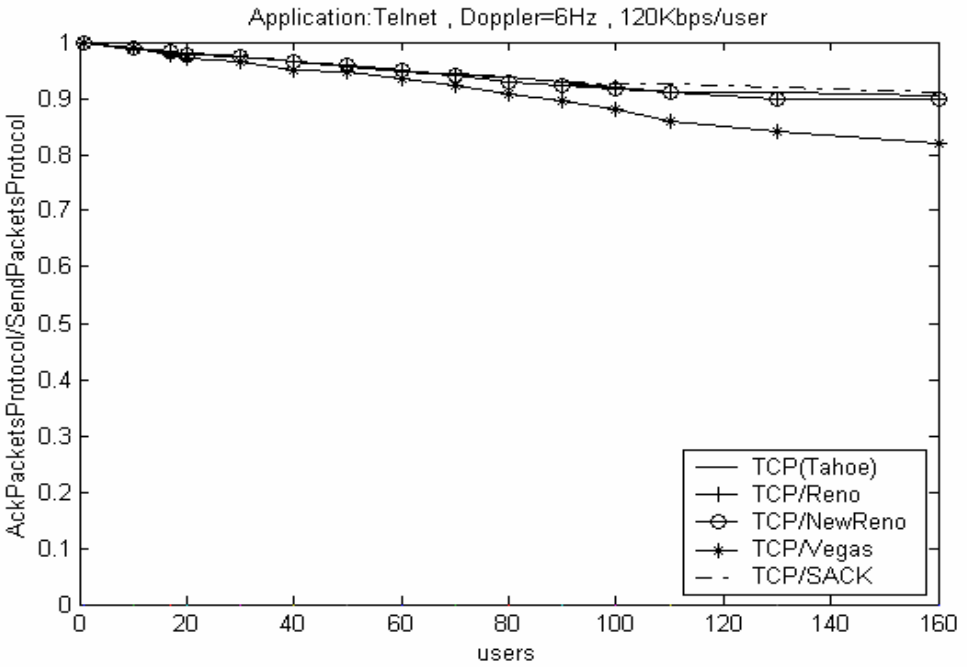
- TCP Vegas outperforms when load is light
- When the load is not very light TCP SACK outperforms TCP New Reno and all the other TCP versions follows
- If Doppler is equal to 80Hz TCP New Reno share the same performance as TCP SACK
- In all cases TCP SACK consumes less energy than any other TCP version

Results and Discussion (5/14)

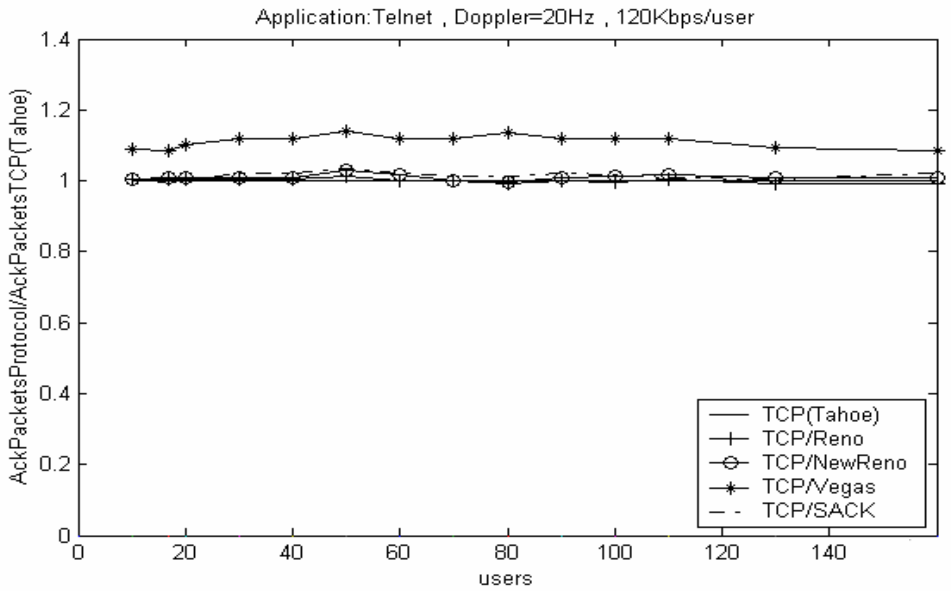


TCP Vegas outperforms all TCP Versions

TCP SACK outperforms slightly
In all cases and especially as number of users increases

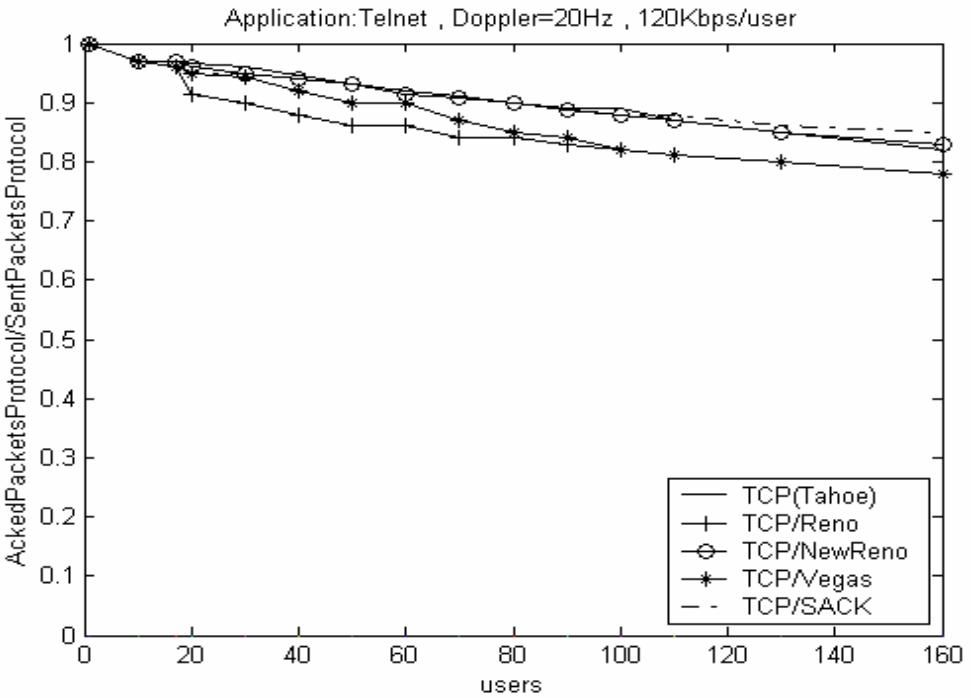


Results and Discussion (6/14)

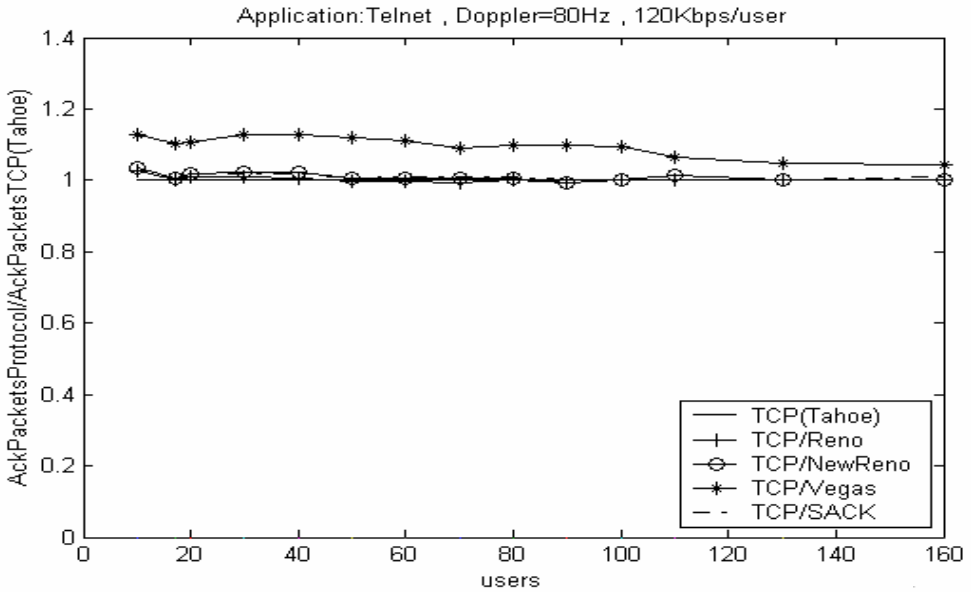


TCP Vegas outperforms all TCP Versions

TCP SACK outperforms slightly
In all cases and especially as number of users increases

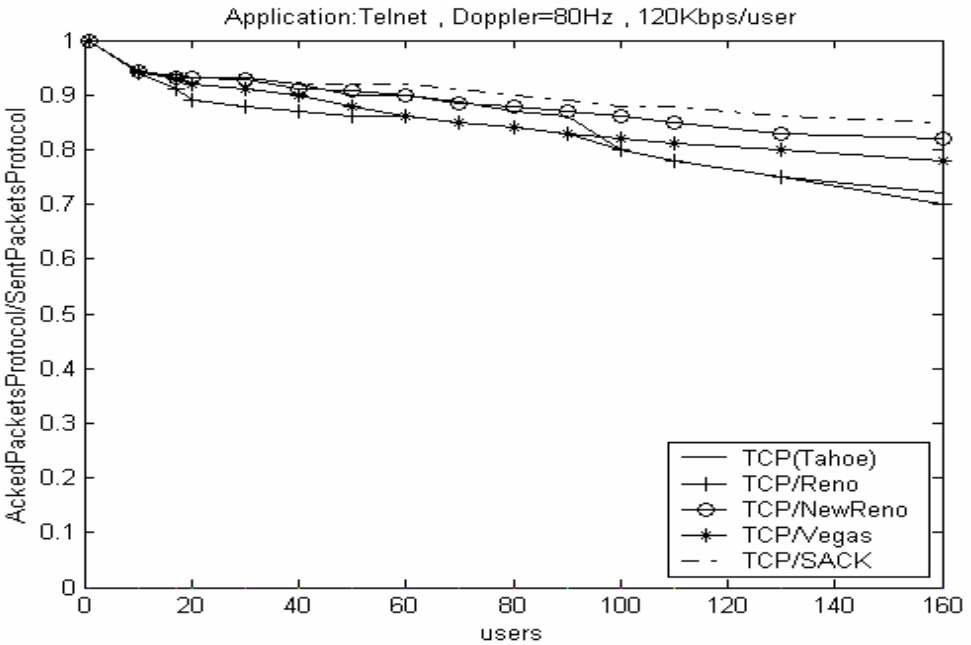


Results and Discussion (7/14)



TCP Vegas outperforms all TCP Versions

TCP SACK outperforms slightly
In all cases and especially as number of users increases

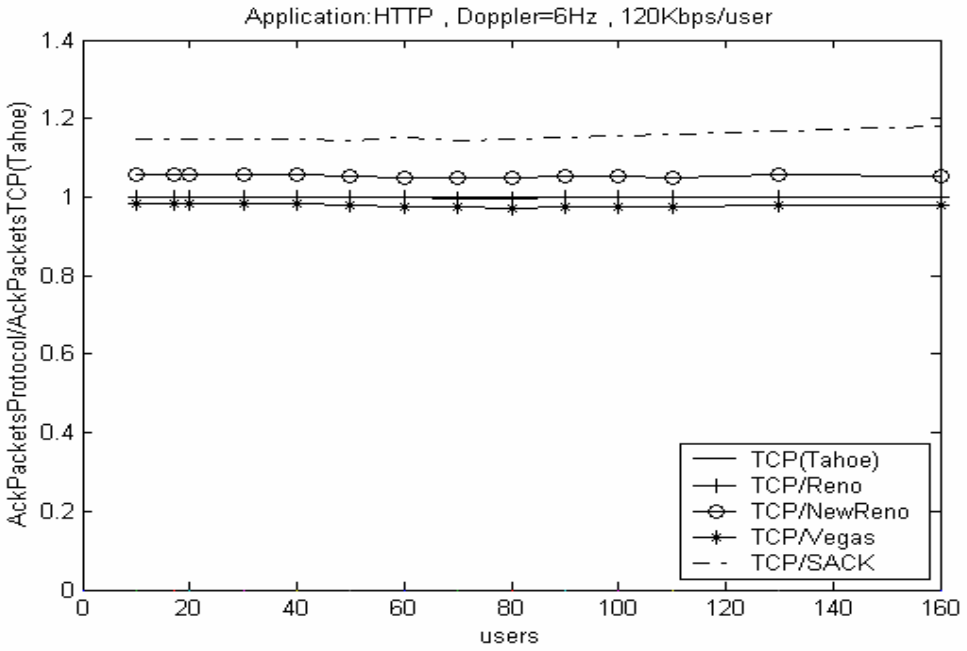


Results and Discussion (8/14)

Telnet Application

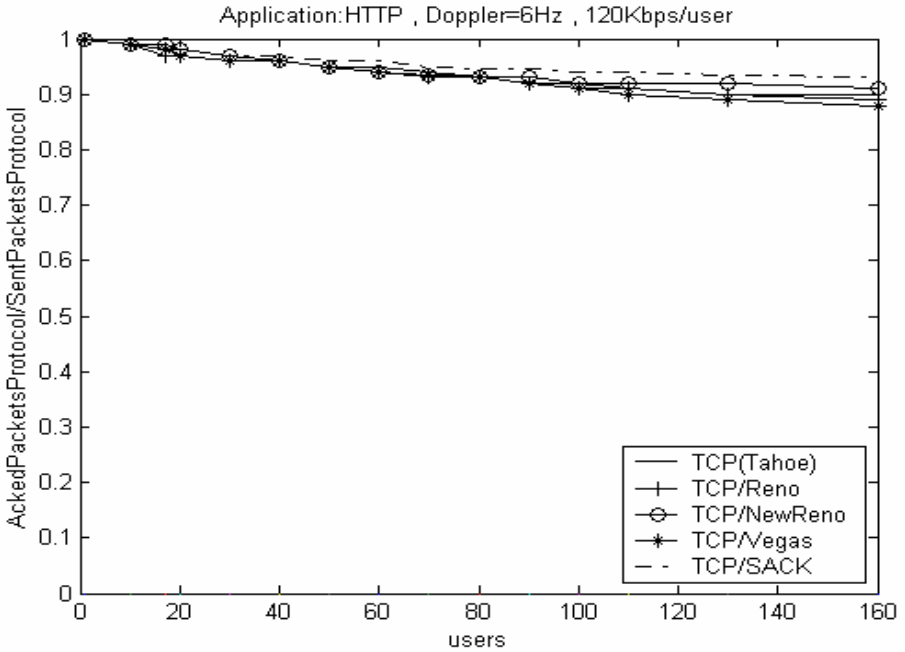
- TCP Vegas outperforms in all cases
- In all cases TCP SACK consumes less energy than any other TCP version

Results and Discussion (9/14)

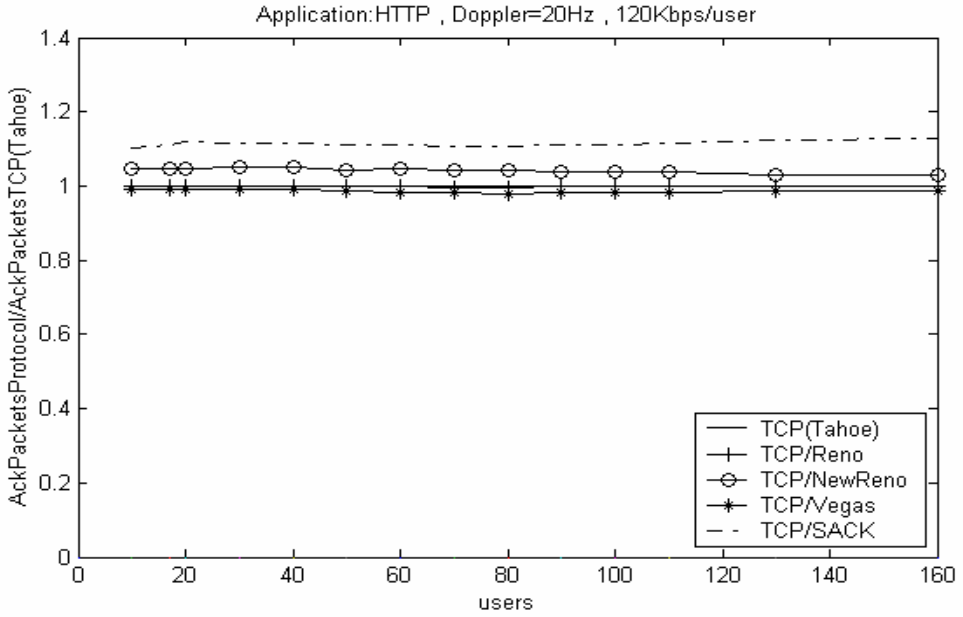


TCP HTTP outperforms and
TCP New Reno follows

TCP SACK outperforms in all cases
and especially as number
of users increases

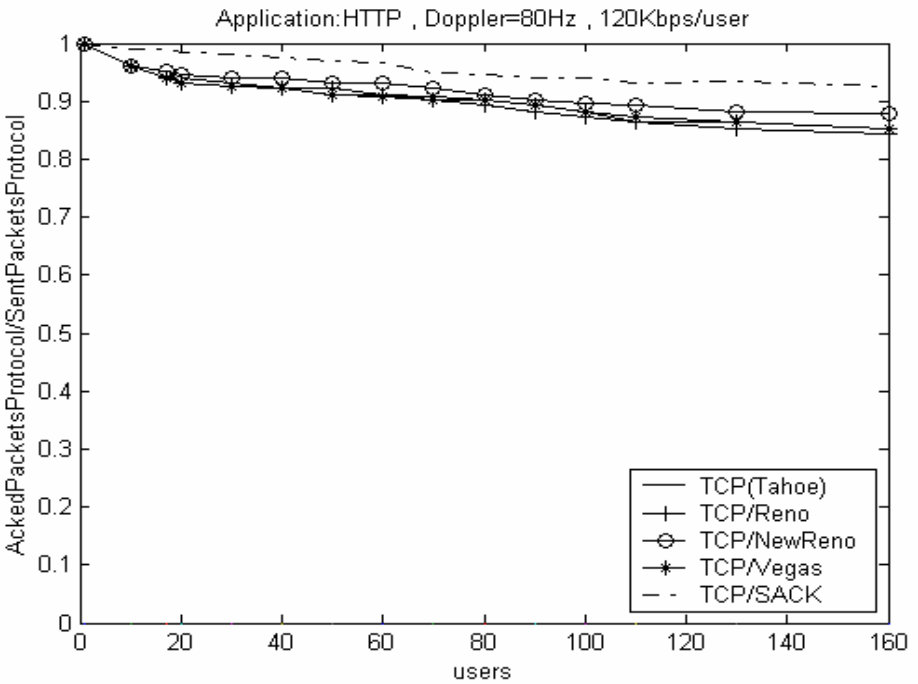


Results and Discussion (10/14)

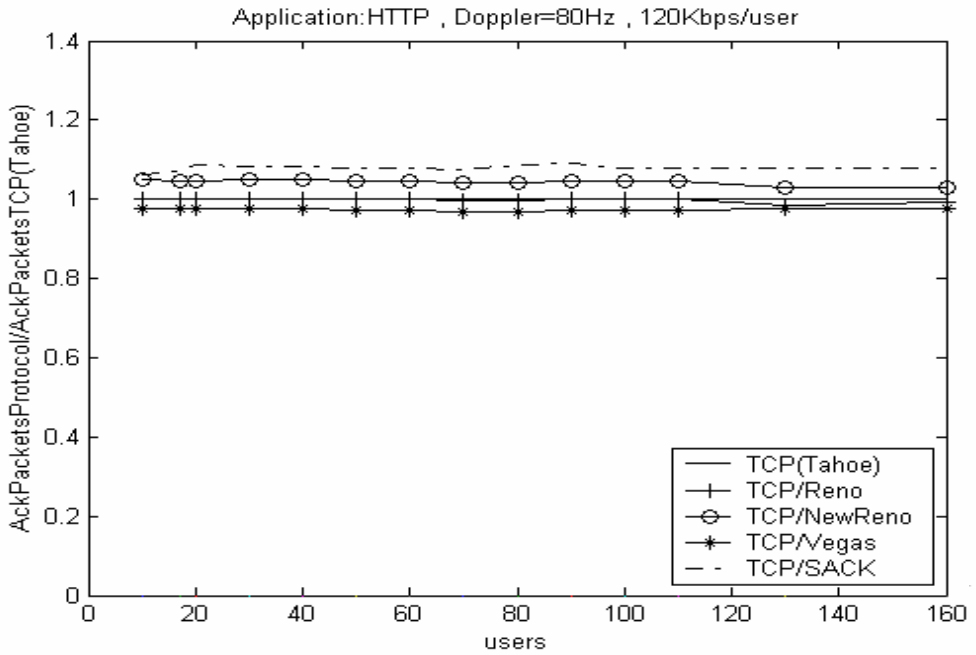


TCP HTTP outperforms and
TCP New Reno follows

TCP SACK outperforms in all cases
and especially as number
of users increases

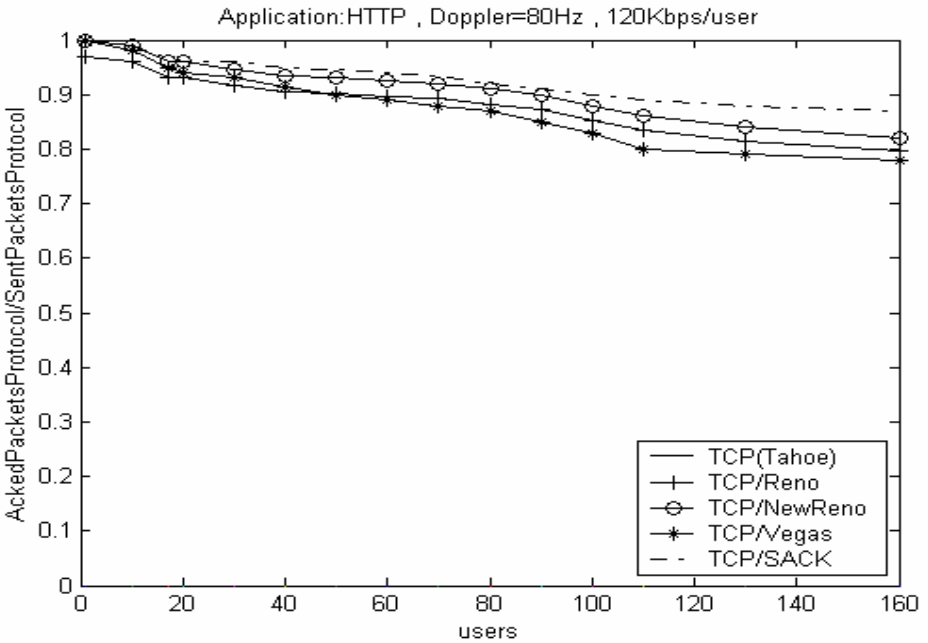


Results and Discussion (11/14)



TCP HTTP outperforms and
TCP New Reno follows

TCP SACK outperforms in all cases
and especially as number
of users increases



Results and Discussion (12/14)

HTTP Application

- TCP SACK outperforms in all cases
- In all cases TCP SACK consumes less energy than any other TCP version

Results and Discussion (13/14)

Statement for users with 240 Kbps bit rate

In the case of 240Kbps users we noticed the same behavior with 120Kbps link capacity in all applications and similar results about energy consumption have been observed. The only difference worth mentioning is that in the case of 240Kbps in HTTP application, TCP SACK still outperforms the others but to a smaller extend than before

Results and Discussion (14/14)

Application Efficiency

				Doppler (in Hz)	
TCP Type	BitRate/User (Kbps)	Application	6	20	80
TCP (Tahoe)	120	FTP	0.80	0.40	0.26
TCP Reno	120	FTP	0,85	0.41	0.26
TCP NewReno	120	FTP	0,87	0.44	0.27
TCP Vegas	120	FTP	0,83	0.42	0.23
TCP SACK	120	FTP	0,88	0.45	0.28
TCP (Tahoe)	120	Telnet	0,96	0.94	0.92
TCP Reno	120	Telnet	0,97	0.95	0.93
TCP NewReno	120	Telnet	0.97	0.95	0.93
TCP Vegas	120	Telnet	0.97	0.96	0.93
TCP SACK	120	Telnet	0.97	0.96	0.93
TCP (Tahoe)	120	HTTP	0.93	0.70	0.50
TCP Reno	120	HTTP	0.92	0.71	0.52
TCP NewReno	120	HTTP	0,93	0.72	0.53
TCP Vegas	120	HTTP	0.90	0.70	0.50
TCP SACK	120	HTTP	0.95	0.75	0.55
TCP (Tahoe)	240	FTP	0.76	0.37	0.24
TCP Reno	240	FTP	0,8	0.38	0.24
TCP NewReno	240	FTP	0,8	0.37	0.25
TCP Vegas	240	FTP	0,77	0.37	0.21
TCP SACK	240	FTP	0,82	0.40	0.28
TCP (Tahoe)	240	Telnet	0,95	0.94	0.92
TCP Reno	240	Telnet	0,97	0.95	0.92
TCP NewReno	240	Telnet	0.96	0.95	0.93
TCP Vegas	240	Telnet	0.97	0.95	0.93
TCP SACK	240	Telnet	0.97	0.96	0.92
TCP (Tahoe)	240	HTTP	0.88	0.69	0.48
TCP Reno	240	HTTP	0.90	0.71	0.50
TCP NewReno	240	HTTP	0,91	0.72	0.50
TCP Vegas	240	HTTP	0.85	0.68	0.44

Conclusions & Future Work (1/3)

- There is not a TCP Version that outperforms all the other in all cases, however, TCP SACK outperforms all the other TCP Version in Energy Consumption.

- More than 75% of UMTS load will be HTTP load. In HTTP Application TCP SACK outperforms all the other TCP Versions

- Most of Operating Systems use only one type of TCP

UMTS must support TCP SACK (or TCP New Reno if TCP SACK can not be used). If more TCP versions can be used TCP Vegas will be suitable for special applications.

Conclusions & Future Work (2/3)

An IETF Draft published in August is in clear agreement with the results of our Thesis:

IETF Draft (draft-ietf-pilc-2.5g3g)

“TCP over 2.5G/3G SHOULD support SACK. In the absence of SACK feature, the TCP should use New Reno”

Conclusions & Future Work (3/3)

Future Work

Future Real Time Application Protocols will use TCP not only for control data but also for information delivery. This protocols can be used over UMTS. If the traffic of this TCP Application can be modeled then we can select the best TCP version.

THE END

QUESTIONS if any… ☺