Project	IEEE 802.16 Broadband Wireless Access Working Group <**http://ieee802.org/16**>
Title	Text Proposal for Traffic Models
Date Submitted	2006-07-20
Source(s)	Gamini Senarath, Wen Tong, Peiying Zhu, Hang Zhang, David Steer, Derek Yu, Mark Naden, Dean Kitchener Nortel 3500 Carling Avenue Ottawa, On, K2H 8E9 Canada gamini@nortel.com wentong@nortel.com Voice: 1-163-763-1316
Re:	Response to a call for contributions for the Relay TG
Abstract	This document captures traffic model for IEEE802.16j
Purpose	Text proposal for IEEE C802.16j-06/040
Notice	
Release	
Patent Policy and Procedures	

Text Proposal for Traffic Models

1 Introduction

This is a text proposal input to the contribution IEEE C802.16j-06/040:"Multi-hop System Evaluation Methodology".

+++++++++++++++++++*Start Text Proposal* ++++++++++++++++++++++++++

2 Traffic models

[Editor's note: Full buffer is baseline model, need to choose on real-time traffic model, adopt [4] and use references]

This section describes the traffic models in detail. Section 2.1 addresses forward link and Section 2.2 the reverse link.

A major objective of multi-hop simulations is to provide the operator a view of how many users can be supported for a given service under a specified multihop configuration at a given coverage level. The traffic generated by a service should be accurately modeled in order to find out the performance. Traffic modeling can be simplified, as explained below, by not modeling the user arrival process and assuming full queue traffic. These are explained below.

Modeling of user arrival process: All the users are not active and they might not register for the same service. In order to avoid different user registration and demand models, the objective of the proposed simulation is made limited to evaluate the performance with the users who are maintaining a session with transmission activity. These can be used to determine the number of such registered users that can be supported. This document does not address the arrival process of such registered users, i.e. it does not address the statistics of subscribers that register and become active.

Full Queue model: In the full queue user traffic model, all the users in the system always have data to send or receive. In other words, there is always a constant amount of data that needs to be transferred, in contrast to bursts of data that follow an arrival process. This model allows the assessment of the spectral efficiency of the system independent of actual user traffic distribution type.

At the relay station, however, the traffic availability depends on the forwarded traffic from either base station, user or by another relay even in the full queue model.

The traffic models provided in the next sections describe only the non-full queue case.

2.1 Traffic Modeling for Down Link Services

The services to be modeled for the forward link are listed in Table 1. FTP, Web browsing and VOIP was included in [1] and [2]. We propose live video services such as video conferencing to be included for simulation. Traffic models for live video will be proposed as a future contribution.

Table 1: Services to be considered for the forward link

#	Application	Traffic Category	Mandatory /Optional
1	FTP	Best-effort	M
2	Web Browsing	Interactive	M
3	VoIP	Real-time	M
4	Video Streaming	Streaming	M
5	Live Video	Interactive Real-time	O

.

2.1.1 FTP [1]

It is proposed that traffic model in [1] be used for FTP. A description is extracted from [1].

In FTP applications, a session consists of a sequence of file transfers, separated by *reading times*. The two main parameters of an FTP session are:

S : the size of a file to be transferred

D_{pc} : reading time, i.e., the time interval between end of download of the previous file and the user request for the next file.

The underlying transport protocol for FTP is TCP. The packet trace of an FTP session is shown in Figure 1.

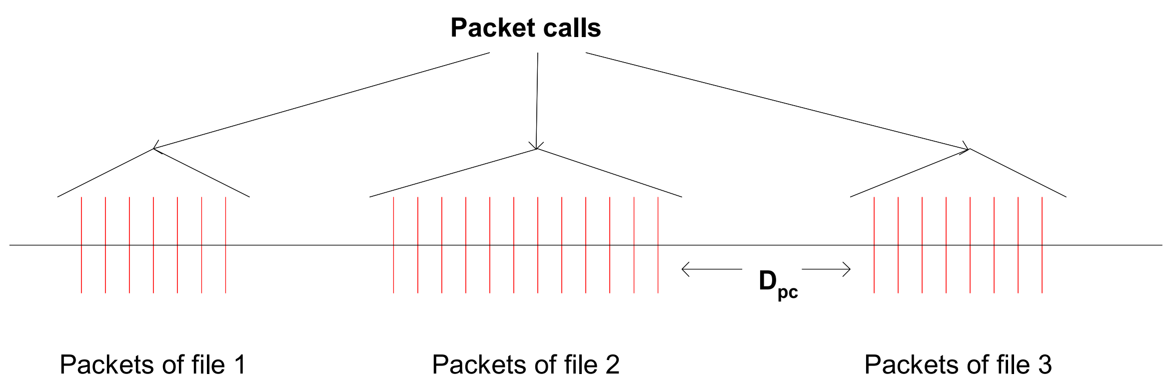


Figure 1 Packet Trace in a Typical FTP Session [1]

The parameters for the FTP application session are described in .

Table 2 FTP Traffic Model Parameters

Component	Distribution	Parameters	PDF

File size (S)	Truncated Lognormal	Mean = 2Mbytes Std. Dev. = 0.722 Mbytes Maximum = 5 Mbytes	$f_x = \frac{1}{\sqrt{2\pi}\sigma x} \exp\left[\frac{-(\ln x - \mu)^2}{2\sigma^2}\right], x \geq 0$ $\sigma = 0.35, \mu = 14.45$
Reading time (D_{pc})	Exponential	Mean = 180 sec.	$f_x = \lambda e^{-\lambda x}, x \geq 0$ $\lambda = 0.006$

2.1.2 Web Browsing [1]

Web browsing is the dominant application for broadband data systems, and has been studied extensively. It is proposed that the traffic model in [1] be used. The descriptions provided in [1] are included below.

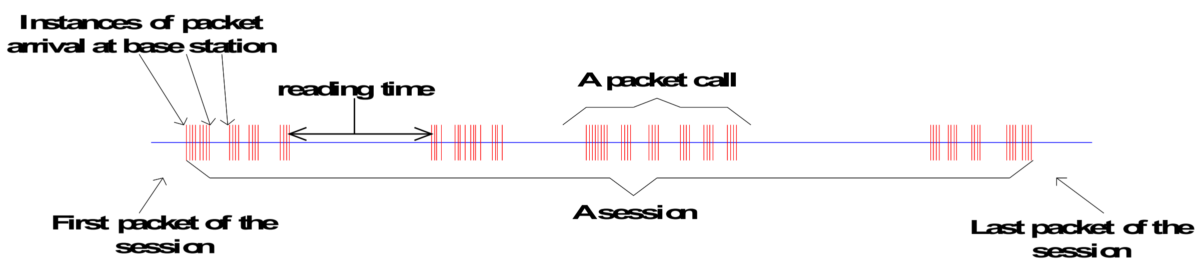


Figure 2 Packet Trace of a Typical Web Browsing Scheme [1]

Figure 2 shows the packet trace of a typical web browsing session. The session is divided into ON/OFF periods representing web-page downloads and the intermediate reading times. In Figure 2, the web-page downloads are referred to as packet calls. These ON and OFF periods are a result of human interaction where the packet call represents a user's request for information and the reading time identifies the time required to digest the web-page.

As is well known, web-browsing traffic is self-similar. In other words, the traffic exhibits similar statistics on different timescales. Therefore, a packet call, like a packet session, is divided into ON/OFF periods as in Figure-3. Unlike a packet session, the ON/OFF periods within a packet call are attributed to machine interaction rather than human interaction. In general, a web-page is constructed from many individually referenced objects. A web-browser will begin serving a user's request by fetching the initial HTML page using an HTTP GET request. After receiving the page, the web-browser will parse the HTML page for additional references to embedded image files such as the graphics on the tops and sides of the page as well as the stylized buttons. The retrieval of the initial page and each of the constituent *objects* is represented by ON period within the packet call while the parsing time and protocol overhead are represented by the OFF periods within a packet call. For simplicity, the term "page" will be used in this paper to refer to each packet call ON period. As a rule-of-thumb, a page represents an individual HTTP request explicitly initiated by the user. The initial HTML page is referred to as the "main object" and the each of the constituent objects referenced from the main object are referred to as an "embedded object".

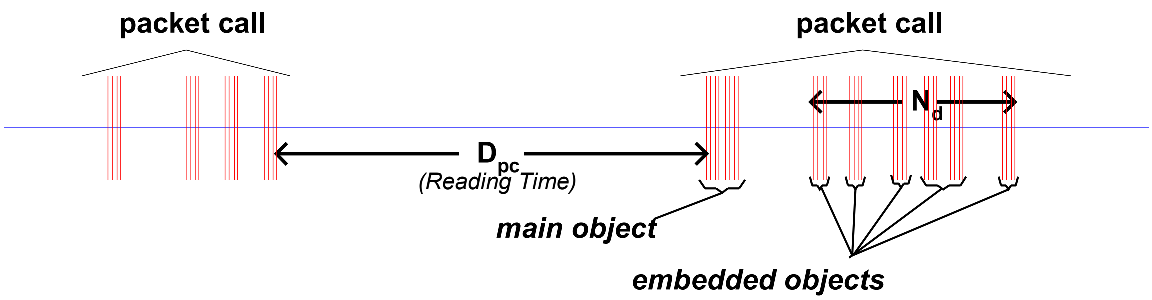


Figure-3 Contents in a Packet Call

The parameters for the web browsing traffic are as follows:

S_M: Size of the main object in a page

S_E: Size of an embedded object in a page

N_d: Number of embedded objects in a page

D_{pc}: Reading time

T_p: Parsing time for the main page

Table 3 HTTP Traffic Model Parameters [1]

Component	Distribution	Parameters	PDF
Main object size (S_M)	Truncated Lognormal	Mean = 10710 bytes Std. dev. = 25032 bytes Minimum = 100 bytes Maximum = 2 Mbytes	$f_x = \frac{1}{\sqrt{2\pi}\sigma x}\exp\left[\frac{-(\ln x-\mu)^2}{2\sigma^2}\right], x \ge 0$ $\sigma = 1.37, \mu = 8.35$
Embedded object size (S_E)	Truncated Lognormal	Mean = 7758 bytes Std. dev. = 126168 bytes Minimum = 50 bytes Maximum = 2 Mbytes	$f_x = \frac{1}{\sqrt{2\pi}\sigma x}\exp\left[\frac{-(\ln x-\mu)^2}{2\sigma^2}\right], x \ge 0$ $\sigma = 2.36, \mu = 6.17$
Number of embedded objects per page (N_d)	Truncated Pareto	Mean = 5.64 Max. = 53	$f_x = \frac{\alpha k^\alpha}{x^{\alpha+1}}, k \le x < m$ $f_x = \left(\frac{k}{m}\right)^\alpha, x = m$ $\alpha = 1.1, k = 2, m = 55$ Note: Subtract k from the generated random value to obtain N_d
Reading time (D_{pc})	Exponential	Mean = 30 sec	$f_x = \lambda e^{-\lambda x}, x \ge 0$ $\lambda = 0.033$
Parsing time (T_p)	Exponential	Mean = 0.13 sec	$f_x = \lambda e^{-\lambda x}, x \ge 0$ $\lambda = 7.69$

Note: When generating a random sample from a truncated distribution, discard the random sample when it is outside the valid interval and regenerate another random sample.

2.1.3 Voice over IP (VoIP) [2]

A VoIP call shall be assumed to be between one user and one wired user. In order to get an evaluation of the air interface the wireline and core network impairments are neglected.

VoIP Traffic Source

The G.729A decoder shall be simulated with an assumed 4 byte IP header. Each packet produced by the G.729A vocoder shall be appended with a 4 byte header that accounts for UDP/IP overhead, after header compression.

2.1.4 Video Streaming [1]

It is proposed, that the following model for streaming video traffic on the forward link which was extracted from [1], be used. Figure 4 describes Video streaming

the steady state of video streaming traffic from the network as seen by the base station. Latency of starting up the call is not considered in this steady state model.

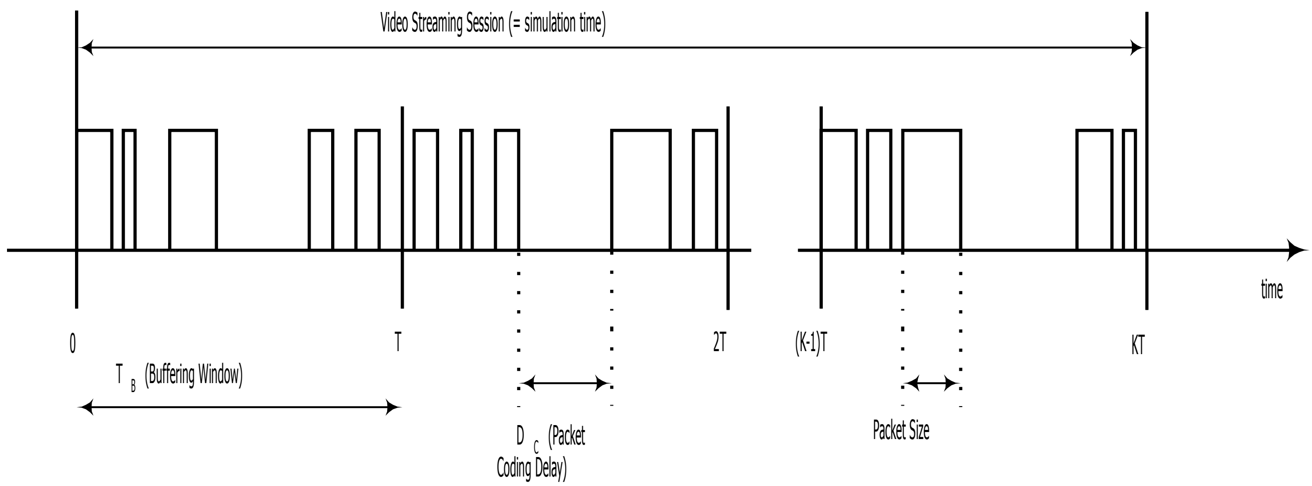


Figure 4 Near Real-Time Video Traffic Model [1]

A video streaming session is defined as the entire video and associated audio streaming call time, which is equal to the simulation time for this model.

Each frame of video data arrives at a regular interval T determined by the number of frames per second (fps). Each frame is decomposed into a fixed number of slices, each transmitted as a single packet. The size of these packets/slices is distributed as a truncated Pareto. Encoding delay, Dc, at the video encoder introduces delay intervals between the packets of a frame. These intervals are modeled by a truncated Pareto distribution. The parameter T_B is the length (in seconds) of the dejitter buffer window in the mobile station used to guarantee a continuous display of video streaming data. This parameter is not relevant for generating the traffic distribution but is useful for identifying periods when the real-time constraint of this service is not met. At the beginning of the simulation, it is assumed that the mobile station de-jitter buffer is full with (T_B x source video data rate) bits of data. Over the simulation time, data is "leaked" out of this buffer at the source video data rate and "filled" as forward link traffic reaches the mobile station. As a performance criterion, the simulation shall record the length of time, if any, during which the de-jitter buffer runs dry.

The de-jitter buffer window for the video streaming service is a maximum of 5 seconds.

Using a source rate of 64 kbps, the video traffic model parameters are defined Table 4.

Table 4 Near Real-Time Video Traffic Model Parameters [1]

Information	Inter-arrival	Number of	Packet (slice)	Inter-arrival time

types	time between the beginning of each frame	packets (slices) in a frame	size	between packets (slices) in a frame
Distribution	Deterministic (Based on 10fps)	Deterministic	Truncated Pareto (Mean= 50bytes, Max= 125bytes)	Truncated Pareto (Mean= 6ms, Max= 12.5ms)
Distribution Parameters	100ms	8	K = 20bytes = 1.2	K = 2.5ms = 1.2

2.1.5 Live Video Services

As mentioned before a traffic model will be provided for live video as a future contribution.

2.1.6 Modeling Reverse Link Traffic for the Forward Link only Simulations

HTTP requests and TCP ACKs come under this category. It is not known, what percentage of traffic would be acks and HTTP requests in a broadband systems. It is clear that the size of the access page increases with time, while ACK messages and HTTP requests remains the same. Therefore, we can expect that in the future systems, the impact of AC K and HTTP requests will be negligible compared to size of the data contents.

2.2 Traffic Modeling for Up Link Services

This section discusses the traffic modeling related to reverse link data traffic. Reverse link traffic to support forward link activities may be modeled in a forward link only simulations as per Section 2.1.6.

The modeling of the forward link traffic to support the reverse link activity when carrying out reverse link only simulations, is described in Section 2.1.4.

2.2.1 FTP Upload / Email [1]

It is proposed that the FTP Upload and Email models used in [1] be used. They are provided below for ease of reference.

Since FTP uses TCP as its transport protocol, the TCP traffic model described in Appendix A [1], is used to represent the distribution of TCP packets for the FTP upload traffic on the UL.

The file upload and email attachment upload are modeled as in Table 2.2.1-5.

Table 2.2.1-5: FTP Characteristics [1]

Arrival of new users	Poisson with parameter
Upload file size	Truncated lognormal; lognormal pdf: $f_x = \frac{1}{\sqrt{2\pi}\sigma x}\exp\left[-\frac{(\ln x-\mu)^2}{2\sigma^2}\right], x \geq 0$ $\sigma = 2.0899, \quad \mu = 0.9385$ Min = 0.5 kbytes Max = 500 kbytes

	If the value generated according to the lognormal pdf is larger than Max or smaller than Min, then discard it and regenerate a new value. The resulting truncated lognormal distribution has a mean = 19.5 kbytes and standard deviation = 46.7 kbytes

The FTP traffic is simulated as follows:

At the beginning of the simulation there are 5 FTP users[1] waiting to transmit.

Before transmitting, call setup is performed for each user

Afterwards, FTP upload users arrive according to the Poisson arrival process, as defined in Table 2.2.1-5.

For each new FTP upload user coming into the system, call setup is performed

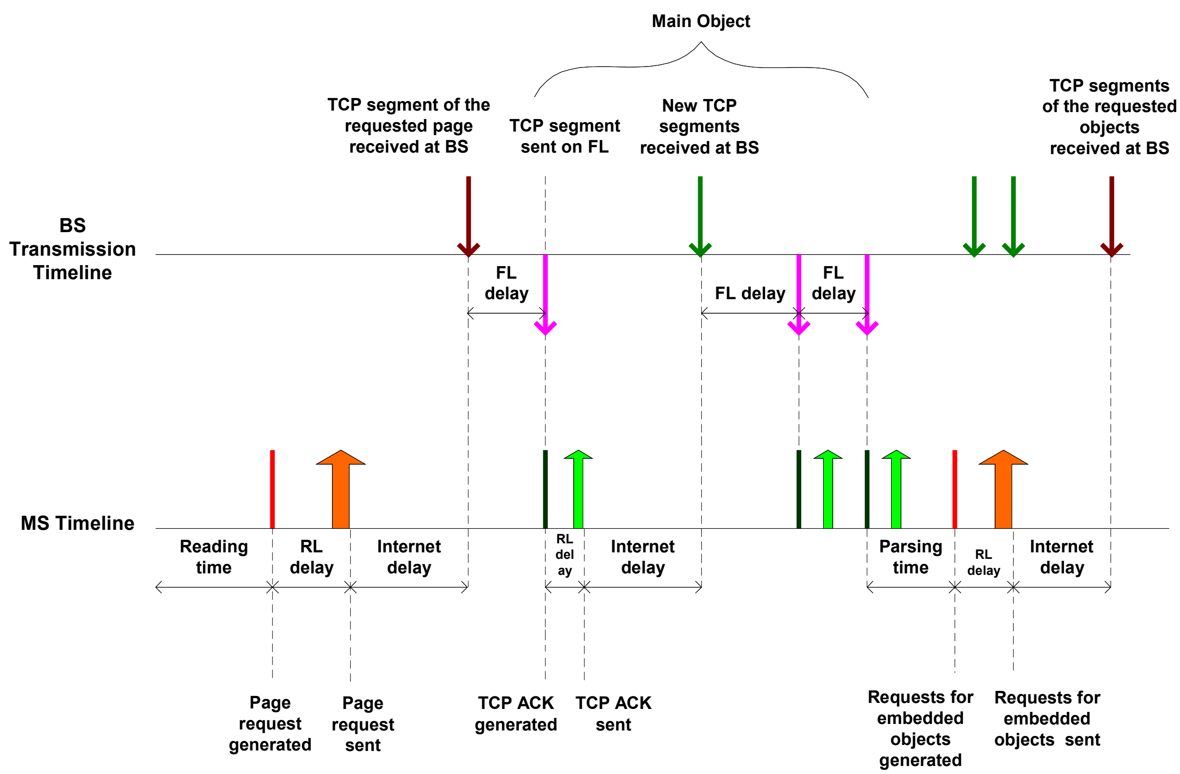
Each FTP upload user stays in the system until it finishes the transmission of its file

After an FTP upload user finishes the transmission of its file, it immediately leaves the system.

Since the arriving FTP users are dropped uniformly over 19 cells, it is possible the number of users can exceed the sector capacity. In that case, the new arrival should be blocked. The sector capacity is 43 in total. The blocking rate should be recorded.

2.2.2 HTTP Model [1]

The following figure is an example of events occurring during a HTTP session. The diagrams and the descriptions have been extracted from [1].



[1] In order to skip the transient period, the number of 5 initial FTP users is taken to represent the number of users at steady state.

Figure-5: Example of events occurring during web browsing.

HTTP Traffic Model Parameters

Reading time (D_{pc}): modeled as in Table 2.2.2.1-6

Internet delay (D_I): modeled as an exponentially distributed random variable with a mean of 50ms

Parsing time (T_p): modeled as in Table 2.2.2.1-6

UL delay: specific for the implemented system. Includes UL packet transmission delay and scheduling delay (if scheduled)

FL delay (D_{FL}): defined as the time a TCP segment is first in the queue for transmission until it finishes transmission on forward link. The delay includes transmission delay and forward link scheduling delay. If there are multiple packets, each packet has its own additional contribution to the overall $D_{FL.}$

Number of TCP segments in the main object (N_M). $N_M = S_M /(MTU-40)$. The main object size, S_M, is generated according to Table 2.2.2.1-6

Number of TCP segments in embedded object (N_E). $N_E = S_E /(MTU-40)$. The embedded object size, S_E, is generated according to Table 2.2.2.1-6

Number of embedded objects (N_d). Modeled according to Table 2.2.2.1-6

HTTP1.1 mode

The opening and the closing of the TCP connections is not modeled[2]

HTTP request size = 350 bytes

Requests for embedded objects are pipelined – all requests are buffered together

MTU size = 1500 bytes

ACK size = 12 bytes[3]

Every received TCP segment is acknowledged.

Table 2.2.2.1-6: HTTP Traffic Model Parameters

Component	**Distribution**	**Parameters**	**PDF**
Main object size (S_M)	Truncated Lognormal	Mean = 9055 bytes Std. dev. = 13265 bytes Minimum = 100 bytes Maximum = 100 Kbytes	If x > max or x < min, then discard and re-generate a new value for x. $f_x = \frac{1}{\sqrt{2\pi}\sigma x} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2} \right], x \geq 0$ $\sigma = 1.37, \quad \mu = 8.35$

[2] This does not have much influence since in HTTP1.1 persistent TCP connections are used to download the objects (located at the same server) and the objects are transferred serially over a single TCP connection.

[3] Compressed TCP/IP header (5 bytes from 40 bytes) and HDLC framing and PPP overhead (7 bytes).

Embedded object size (S_E)	Truncated Lognormal	Mean = 5958 bytes Std. dev. = 11376 bytes Minimum = 50 bytes Maximum = 100 Kbytes	$f_x = \frac{1}{\sqrt{2\pi}\sigma x}\exp\left[\frac{-(\ln x-\mu)^2}{2\sigma^2}\right], x \geq 0$ $\sigma = 1.69, \quad \mu = 7.53$ If x > max or x < min, then discard and re-generate a new value for x.
Number of embedded objects per page (N_d)	Truncated Pareto	Mean = 4.229 Max. = 53	$f_x = \frac{\alpha_k^\alpha}{x^{\alpha+1}}, k \leq x < m$ $\alpha = 1.1, k = 2, m = 55$ Note: Subtract k from the generated random value to obtain N_d If x > max, then discard and re-generate a new value for x
Reading time (D_{pc})	Exponential	Mean = 30 sec	$f_x = \lambda e^{-\lambda x}, x \geq 0$ $\lambda = 0.033$
Initial reading time (D_{ipc})	Uniform	Range [0, 10] s	$f_x = \frac{1}{b-a}, \quad a \leq x \leq b$ $a = 0, b = 10$
Parsing time (T_p)	Exponential	Mean = 0.13 sec	$f_x = \lambda e^{-\lambda x}, x \geq 0$ $\lambda = 7.69$

Packet Arrival Model for HTTP

At the beginning of the simulation, call setup is performed for all HTTP users. After that, the simulation flow is described as follows:

Generate an initial reading time D_{ipc}.[4] Wait D_{ipc} seconds.

Initiate the TCP window size W=1

Generate a request for the main page

Wait for the requests to go through the UL and reach the bases station (UL delay):

In case these are requests for embedded objects, wait until all requests reach the base station.

Generate an Internet delay D_I. Wait D_I seconds.

Generate random delays, which define the time instances when each of the TCP segment transmission is completed the FL. The number of these instances is:

[4] The initial reading time is defined differently from subsequent reading times in order to ensure that all HTTP users finish the reading time within a limited period.

For the main page:

At the very beginning of the packet call: 1.

Afterwards: min(2n, #of outstanding TCP segments on FL), where n is the number of ACKs received in the last physical layer packet (from the step 2.2.2)

For embedded objects:

At the very beginning of the transmission of embedded objects: min(W, $\sum_{i=1}^{N_d} N_E^i$).

Afterwards: min(2n, #of outstanding TCP segments on FL), where n is the number of ACKs received in the last physical layer packet (from the step 9 a.i.

Every time instance of the completed TCP segment transmission on FL generates an ACK on UL

Continue UL simulation – when ACK is generated, reduce the number of outstanding TCP packets by 1

Examine if the transmission of the very last TCP segment of the HTTP object is completed:

If no:

Proceed with simulation until next ACK or a group of n ACKs within a single physical layer packet is transmitted

Increase W:=W+n

Go to step 2.2.2

If yes, for main page:

Generate T_p (parsing time)

Generate requests for embedded objects

Continue UL simulation - transmit outstanding ACK(s) for the main page and accordingly increment W:=W+n for each group of n ACKs transmitted, until requests for embedded objects are generated

Go to step 2.2.2

If yes, for embedded objects:

Generate D_{pc} (reading time)

Continue UL simulation - transmit outstanding ACK(s) for the embedded objects

Go to step 2.2.2 when reading time expires or until all ACKs are transmitted, whichever is longer.

2.2.2.1 Modeling of Forward Link Traffic when carrying out a Up Link only simulation

HTTP requests and TCP ACKs come under this category. It is not known, what percentage of traffic would be acks and HTTP requests in a broadband systems. It is clear that the size of the access page increases with time, while ACK messages and HTTP requests remains the same. Therefore, we can expect that in the future systems, the impact of AC K and HTTP requests will be negligible compared to size of the data contents.

++++++++++++++++++++*End of Text Proposal* ++++++++++++++++++++++

3 References

[1] 3GPP2/TSG-C.R1002, "1xEV-DV Evaluation Methodology (V14)", June 2003.
[2] 3GPP TR 25.996 V6.1.0, "Spatial channel model for Multiple Input Multiple Output (MIMO) simulations," September 2003.
[3] IEEE C802.16j-6/023, "Multi-hop System Evaluation Methodology" May, 2006
[4] IEEE C802.16j-6/024, "Multi-hop System Evaluation Methodology: Traffic Models", May, 2006
[5] IEEE C802.16j-6/025, "Multi-hop System Evaluation Methodology–Performance Metrics" May, 2006
[6] IEEE C802.16j-6/020r1,"Channel Models & Performance Metrics for IEEE 802.16j Relay Task Group" May, 2006
[7] IEEE C802.16j-6/009, "Correlated Lognormal Shadowing Model", May, 2006
[8] IEEE C802.16j-6/010, "Below Rooftop Path Loss Model", May, 2006
[9] IEEE C802.16j-6/011, "Multi-hop Path Loss Model (Base-to-Relay and Base-to-Mobile)", May, 2006
[10] IEEE C802.16j-6/012, "Multi-hop Network Simulation with Street Layout "May, 2006
[11] IEEE C802.16j-6/014, "Metrics for Multi-hop Systems" May, 2006
[12] IEEE C802.16j-6/015, "RF compatibility of RS with other MSS" May, 2006
[13] IEEE C802.16j-6/003, "Seaport Path Loss Model for Fixed Wireless Applications", May, 2006
[14] IEEE 802.16.3c-01/29r4, "Channel Models for Fixed Wireless Applications" July 21, 2001 http://www.wirelessman.org/tg3/contrib/802163c-01_29r4.pdf