ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων

Μελέτη επίδοσης TCP σε ασύρματα κανάλια επικοινωνίας

Δημήτριος Ιωσηφίδης

A.M. ME0439

Μεταπτυχιακή Διπλωματική Εργασία

Σεπτέμβριος 2009

Contents

ΠΕΡΙΛΗΨΗ

Είναι κοινά αποδεκτό πως μια από τις βασικές ανάγκες των ανθρώπων είναι αυτή για επικοινωνία. Στη σύγχρονη εποχή οι ασύρματες τεχνολογίες επικοινωνίας ενσωματώνονται ολοένα και περισσότερο στη καθημερινή ζωή του ανθρώπου. Συσκευές όπως κινητά τηλέφωνα, GPS, palmtops, laptops, και δορυφορικές συσκευές αλλάζουν το τρόπο που οι άνθρωποι επικοινωνούν και αλληλεπιδρούν. Επομένως η δυνατότητα για κινητικότητα κατά τη διάρκεια της επικοινωνίας είναι πλέον μία πολύ σημαντική και αναγκαία παράμετρος, οπότε η υλοποίηση τεχνολογιών που θα μπορούν να λειτουργούν με επάρκεια σε ασύρματα περιβάλλοντα και να παρέχουν ποιότητα υπηρεσιών είναι αναγκαία.

Σε αυτή την εργασία μελετάται η επίδοση κλήσεων σε ασύρματα δίκτυα όπου το κανάλι επικοινωνίας δημιουργεί διαλείψεις στο σήμα λήψης. Εξετάζονται δύο μορφές εξασθένισης σήματος, αυτή που ακολουθεί κατανομή lognormal για σκίαση μεγάλης κλίμακας και η Rayleigh για πολυδιαδομικό fading μικρής κλίμακας. Για κάθε μία κατανομή, εξάγονται κλειστού τύπου εκφράσεις που υπολογίζουν τη πιθανότητα ολοκλήρωσης της κλήσης και τη πιθανότητα αποτυχίας εξαιτίας πτώσης του σήματος λήψης κάτω από ένα κατώφλι.

Επιπρόσθετα, γίνεται έρευνα για βελτιώσεις σε παραμέτρους του SIP για καλύτερη επίδοση πάνω σε ασύρματο κανάλι. Ο χρόνος για την εγκαθίδρυση κλήσεων με διαφορετικές μεθόδους μεταφοράς (TCP και UDP) υπολογίζεται για διαφορετικές περιπτώσεις.

Πιο αναλυτικά η εργασία περιλαμβάνει τα εξής:

- Στο κεφάλαιο 1, γίνεται αναλυτική περιγραφή του Session Initiation Protocol (SIP). Έμφαση δίνεται στα στοιχεία από τα οποία αποτελείται το SIP, τις εφαρμογές του και παρατίθενται παραδείγματα επικοινωνίας.

- Στο κεφάλαιο 2, παρουσιάζεται το πρόβλημα της χρησιμοποίησης του TCP σε ασύρματα περιβάλλοντα και περιγράφονται τεχνικές που επιχειρούν να βελτιώσουν την επίδοση της επικοινωνίας.
- Στο κεφάλαιο 3, παρέχουμε την θεωρία μοντελοποίησης σε ασύρματα κανάλια με διαλείψεις και για τις περιπτώσεις των Lognormal και Rayleigh fading υπολογίζουμε την πιθανότητα απότομης διακοπής της σύνδεσης.
- Στο κεφάλαιο 4, αναλύουμε το Radio Link Protocol (RLP) και υπολογίζουμε τους χρόνους εκκίνησης για SIP πάνω σε UDP / TCP.

CHAPTER 1

1. Introduction

Session Initiation Protocol (SIP) is an application level control protocol developed in the Internet Engineering Task Force (IETF), published as IETF RFC. SIP is used for setting up, modifying and terminating multimedia sessions or Internet telephony calls between two or more parties. SIP may be used to invite participants to unicast and multicast sessions, and existing sessions may be modified to include new media and participants. In the Internet, SIP is used together with other protocols, e.g., using Real Time Transport Protocol (RTP) for media transport, and Session Description Protocol (SDP) for session description payloads. Borrowing from ubiquitous Internet protocols, such as HTTP and SMTP, SIP is text-encoded and highly extensible. SIP may be extended to accommodate features and services such as call control services, mobility, interoperability with existing telephony systems, and more.

SIP provides four basic functions:

- *Name translation and user location*, which involve the mapping between names to identify a callee and the eventual location of the callee, be it a telephone number, pager, voice mail, e-mail address, or Web page. This translation and location function can be more complex than a simple database lookup, but it may depend on caller and callee preferences; media and codec support at various points of presence; and service provider, or third-party, logic.
- *Feature negotiation*, which allows a group of participants to agree on the media to exchange, and their respective parameters. In a multiparty IP telephony conference, the set and type of media need not be uniform. Different participants can exchange different media types with each other; some may only receive audio, while others may receive audio and video. Such a mix may be due to limited computing facilities at an endpoint, a desire to use a particular format for a medium, or the lack of a single medium format common to all participants.

- *Call participant management*, which allows any call participant to invite new users into an existing call and terminate associations with other participants. During the call, participants should be able to transfer other participants and place them on hold.
- *Call feature changes*, which make it possible to adjust the composition of media sessions during the course of a call, either because the participants require additional or reduced functionality, or because of constraint simposed or removed by adding or removing call participants.

2. SIP Entities

A SIP network is composed of four types of logical SIP entities. Each entity has specific functions and participates in SIP communication as a client (initiates requests), as a server (responds to requests), or as both. One "physical device" can have the functionality of more than one logical SIP entity. For example, a network server working as a Proxy server can also function as a Registrar at the same time.

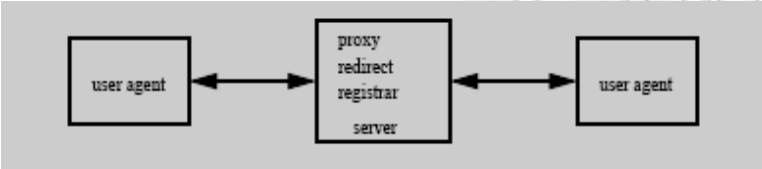

Figure 1.1 *SIP Entities*

Following are the four types of logical SIP entities:

2.1 User Agent

In SIP, a User Agent (UA) is the endpoint entity. User Agents initiate and terminate sessions by exchanging requests and responses. RFC 2543 defines the User Agent as an application, which contains both a User Agent client and User Agent server, as follows:

- *User Agent Client (UAC):* a client application that initiates SIP requests.
- *User Agent Server (UAS):* a server application that contacts the user when a SIP request is received and that returns a response on behalf of the user.

Some of the devices that can have a User Agent function in a SIP network are: Workstations, IP-phones, telephony gateways, call agents, automated answering services.

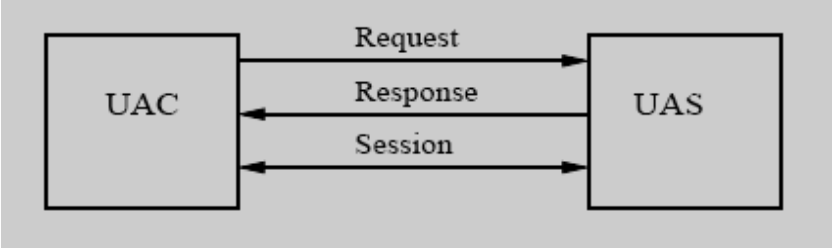

Figure 1.2 *Communication between two UAs*

2.2 Proxy Server

A *Proxy Server* is an intermediary entity that acts as both a server and a client for the purpose of making requests on behalf of other clients. Requests are serviced either internally or by passing them on, possibly after translation, to other servers. A Proxy interprets, and, if necessary, rewrites a request message before forwarding it. A proxy determines the next hop of the request, i.e., the next server in the call path, and forwards the message after possibly modifying some of the message headers. As such, the proxy has no knowledge of the type or make of the next hop server – it may be another proxy, a redirect server, a registrar or a user agent server. Hence, the request can traverse several proxies before reaching its final destination. Responses always return along the same call path as the request followed, but in reverse order. However, intermediary proxies in the call path don't have to maintain call state – all of the necessary information about the transaction state is encoded in the SIP message. All intermediary proxies add an entry to the Via header in order to create a routing table. All proxies listed in this table are visited by the response in reverse order.

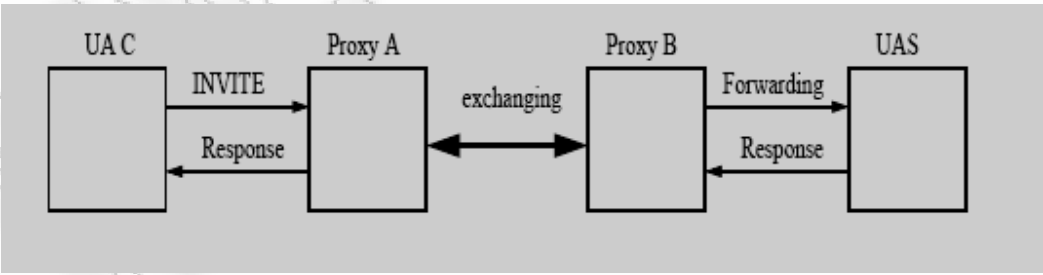

Figure 1.3 *Communication with proxies*

2.3 REDIRECT SERVER

A *Redirect Server* is a server that accepts a SIP request, maps the SIP address of the called party into zero (if there is no known address) or more new addresses and returns them to the client. Unlike Proxy servers, Redirect Servers do not pass the request on to other servers.

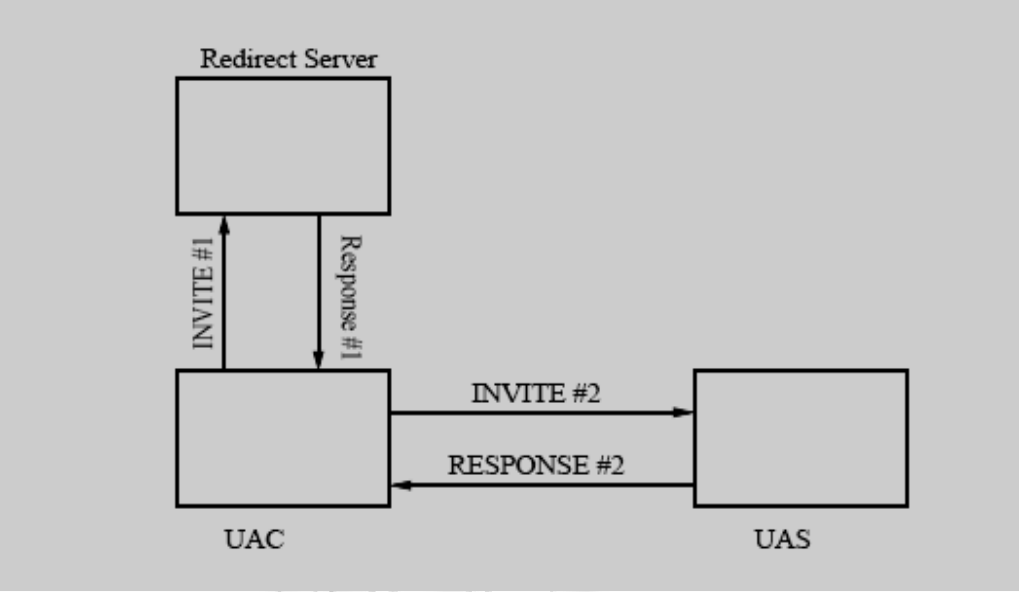

Figure 1.4 *Communication with redirect server*

2.4 REGISTRAR

A *Registrar* is a server that accepts register requests for the purpose of updating location database with the contact information of the user specified in the request. By registering, users impose the state of their availability and capability onto the network. Usually a registrar is served with a back-end database for storing the user information found in REGISTER messages. Both redirect and proxy servers utilize the information made available by the registrar in their call routing process. By registering, the user

introduces a binding between a global identity, e.g., aki.niemi@nokia.com, and a physical location, e.g., apniemi@[3012:110d:ab30::1].

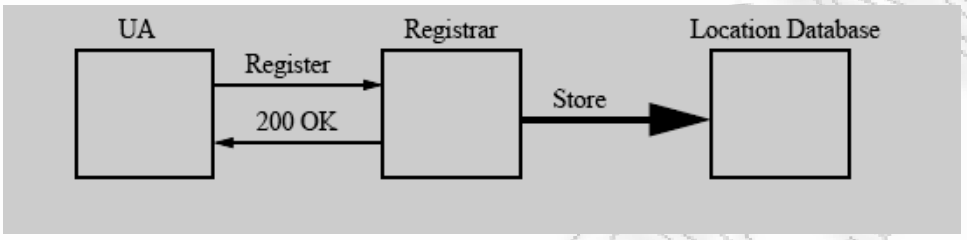

Figure 1.5 *Communication with registar server*

3. Entity Interaction

3.1 Example *of* communication *between UAs*

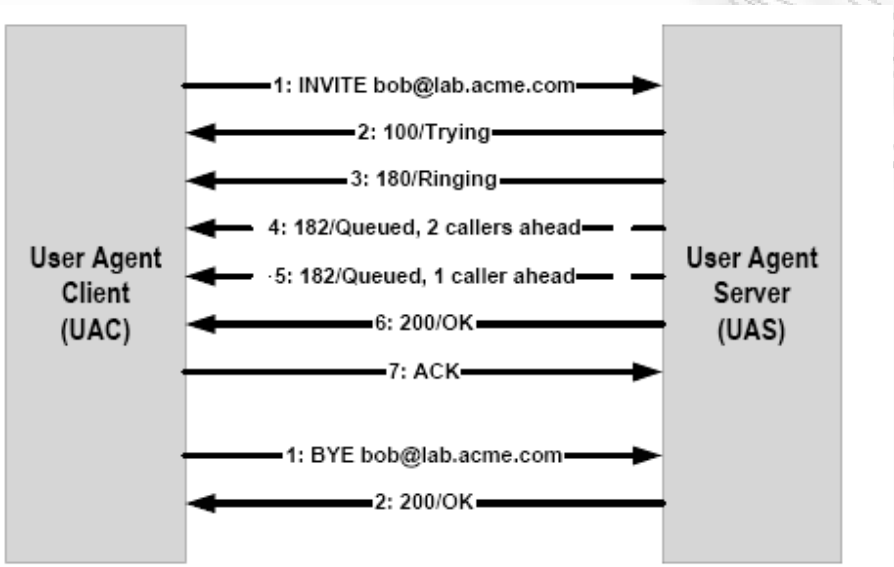

Figure 1.6 *Session Establishment and Call Termination*

CALL FLOW

1. The calling User Agent Client sends an INVITE message to Bob's SIP address: sip:bob@acme.com. This message also contains an SDP packet describing the media capabilities of the calling terminal.
2. The UAS receives the request and immediately responds with a 100-Trying response message.
3. The UAS starts "ringing" to inform Bob of the new call. Simultaneously a 180 (Ringing) message is sent to the UAC.
4. The UAS sends a 182 (Queued) call status message to report that the call is behind two other calls in the queue.
5. The UAS sends a 182 (Queued) call status message to report that the call is behind one other call in the queue.
6. Bob picks up the call and the UAS sends a 200 (OK) message to the calling UA. This message also contains an SDP packet describing the media capabilities of Bob's terminal.

7. The calling UAC sends an ACK request to confirm the 200 (OK) response was received.

SESSION TERMINATION

The session termination call flow proceeds as follows:

1. The caller decides to end the call and "hangs-up". This results in a BYE request being sent to Bob's UAS at SIP address sip:bob@lab.acme.com
2. Bob's UAS responds with 200 (OK) message and notifies Bob that the conversation has ended.
3.

3. 2 *Example of communication between UAs - Redirect Server*

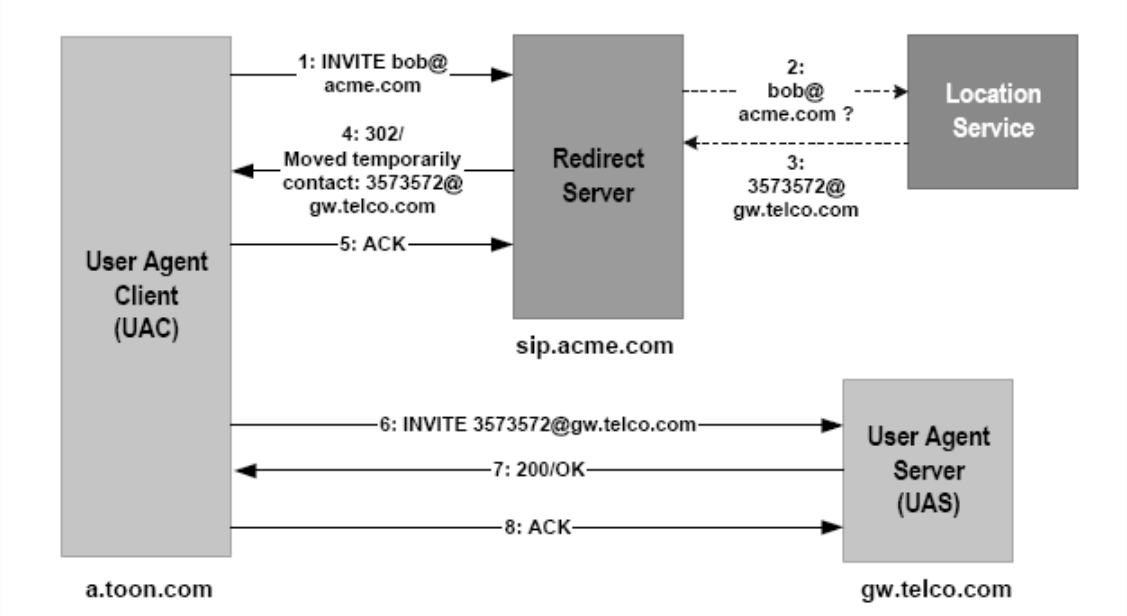

Figure 1.7 *Simple Call Redirection Using a Redirect Server.*

CALL FLOW

1. First a SIP INVITE message is sent to bob@acme.com, but finds the Redirect server sip.acme.com along the signalling path.

2. The Redirect server looks up Bob's current location in a Location Service using a non-SIP protocol (for example, LDAP).
3. The Location Service returns Bob's current location: SIP address 3573572@gwtelco.com.
4. The Redirect Server returns this information to the calling UAC using a 302 (Moved Temporarily) response. In the response message it enters a contact header and sets the value to Bob's current location, 3573572@gwtelco.com.
5. The calling UAC acknowledges the response by sending an ACK Message.
6. The calling UAC then continues the transaction directly with gw.telco.com by sending a new INVITE.
7. Gw.telco.com is able to notify Bob's terminal of the call and Bob "picks up" the call. A 200 (OK) response is sent back to the calling UAC.
8. The calling UAC acknowledges with an ACK message.

3.3 *Example of communication between UAs - Proxy Server*

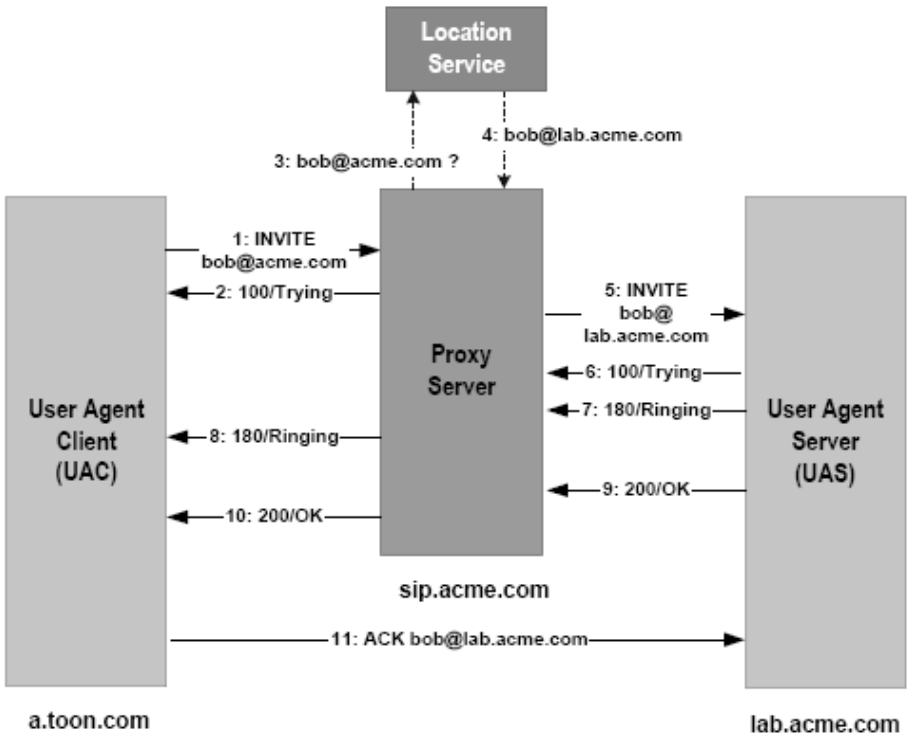

Figure 1.8 Call Proxy-ing Scenario

CALL FLOW

1. An INVITE message is sent to bob@ acme.com, but finds the Proxy server sip.acme.com along the signaling path.
2. The Proxy server immediately responds with a 100 (Trying) provisional response.
3. The Proxy server looks-up Bob's current location in a Location Service using a non-SIP protocol (For example, LDAP).
4. The Location Service returns Bob's current location: SIP address bob@lab.acme.com.
5. The Proxy server decides to proxy the call and creates a new INVITE message based on the original INVITE message, but with the request URI in the start line changed to bob@lab.acme.com. The Proxy server sends this request to the UAS at lab.acme.com.
6. The UAS responds first with a 100 (Trying).
7. The UAS responds with a 180 (Ringing) response.
8. The Proxy server forwards the 180 (Ringing) response back to the calling UA.
9. When the call is accepted by the user (for example, by picking up the handset) the UAS at lab.acme.com sends a 200 (OK) response. In this example, Bob's UAS inserts a Contact header into the response with the value bob@lab.acme.com. Further SIP communication will be sent directly to it and not via the Proxy Server. This action is optional.
10. The Proxy forwards the 200 (OK) response back to the calling UAC.
11. The calling UA sends an ACK directly to Bob's UA at the lab (according to the Contact header it found in the 200 (OK) responses).

4. SIP Messages

In SIP, messages are utilised as means of communication between SIP network elements.

The SIP messages are classified into two types: REQUEST and RESPONSE. Both types use a basic format to construct the message. It consists of a start-line, a message header that contains one or more header fields, an empty line indicating the end of the header and an optional message-body. The structure of message can be shown as:

generic-message = start-line

[message-header]
CRLF
[message-body]

The following is an example of a real SIP message:

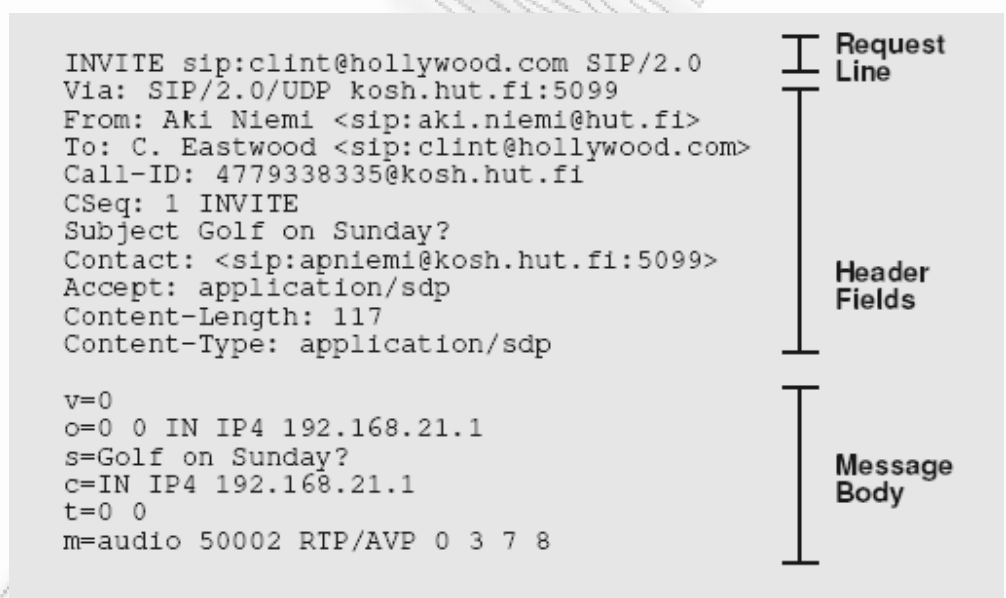

Figure 1.9 *SIP Message example*

4.1 SIP Request

An SIP *request* is the SIP message sent from a client to a server for the purpose of invoking a particular operation. We present the structure of the request message below, it consists of three parts:

Request = Request-line
[message-header]
CRLF
[message-body]

The Request-line begins with a method token, such as INVITE, and following the SIP version number. Then, the different types of headers contain the information of sender and the properties of the session requested. For example, an INVITE request-line can be: *INVITE SIP:office@tudelft.nl SIP/2.0*

The most important element in request-line are method. The *method* states the primary function of a request that decides what type of the message is and what operations should be executed. SIP uses six methods:

1. **INVITE**: This method indicates that a user or server is being invited to participate in a session. The message contains a description of the session, using Session Description Protocol (SDP), and the type of media that is to be used for the session, caller2 and callee3 addresses, user location, caller preferences, and desired features for the response. An INVITE request may be sent during a session on holding to modify the session characteristics.
2. **ACK**: This method is used with INVITE method. The purpose of ACK is to confirm that the client has received a final response to an INVITE request. The ACK can contain the message body which confirm the final session description to be used by the callee. But it is not necessary. If there

is no message body with ACK message, which means the callee accepts the session Description in INVITE request.

3. **OPTIONS**: This method Queries a server about the capabilities of the called party. A called user agent may also send an OPTIONS message reflecting how it would respond to an INVITE if it is in urgency. A server may respond to this request if believes it can contact the user.
4. **BYE**: The BYE can be generated by all communicating parties to indicate other one to end the call. A BYE is forwarded after a party has released the call. The other party that receives BYE has to release transmitting streams.
5. **CANCEL**: The CANCEL is used to end a pending INVITE request but does not affect completed requests. For example, a proxy server has received a response to one of its parallel searches; the other response should be cancelled. In practice, only the INVITE is cancelled.
6. **REGISTER**: This method is used by a client to register its address to an SIP server. The client may be in any locations in the same domain under the server. Usually, a user agent may register with a local server on start-up by sending an IP multi-cast to "all SIP servers" (sip.mcast.net, 224.0.1.75).

Method	Description	Status
INVITE	Inviting participants to sessions	RFC [43]
MESSAGE	Instant messaging	I-D [62]
OPTIONS	Query of server capabilities	RFC [43]
REGISTER	Registering address bindings	RFC [43]
ACK	Acknowledgement of responses	RFC [43]
PRACK	Acknowledgement of provisional responses	I-D [60]
BYE	Termination of sessions	RFC [43]
INFO	Information exchange	RFC [26]
REFER	Call transfer	I-D [69]
SUBSCRIBE	Subscription to an event	I-D [58]
NOTIFY	Notification in an event session	I-D [58]

Table 1.1 *SIP Main Methods*

Header fields

An *SIP header* is the description of the references in a SIP message, for example, the host address, the destination address, call sequence number etc. Each of the headers in the SIP message contains a number of fields. The contents of these fields are decided by UAC to inform the UAS what is the proposition for the required session. These header fields form the basic architecture of SIP message. There are many header fields defined in RFC3261, but not all the fields have to present in a header. Only several header fields are obligatory to be included in all the headers.

These are: Via, To, From, CSeq and Call-ID.

We give the description of these header fields and other several useful header fields.

- **To:** This field identifies the recipient of the request message. It can be the name-address (URL) or number-address (numeric IP address).
- **From:** This field indicates the initiator of the SIP request message. It is copied from the request to the response by server. It is name-address or number-address.
- **Via**: This field indicates the path that the request has traversed so far, and used to ensure that the response message takes the same inverse path as the request message. The client makes the request with a Via field containing its host name or network address and the port number at which it wishes to receive responses. Each subsequent proxy server that forwards the request adds its own additional Via field before any existing Via fields.
- **CSeq:** The name of the field comes from the Command Sequence. It contains the request method (for example, INVITE), and a sequence number.
- **Call-ID:** It identifies a particular SIP invitation or all registrations for a specific client uniquely. A multimedia conference results in several calls with different Call-IDs. The REGISTER and OPTIONS methods use this parameter to match requests and responses.
- Contact: This parameter provides a URI (Universal Resource Identifier) thatthe user can be used for further communications. For example, when the INVITE

request is forwarded, the request message is sent to both *To* address and *Contact* address. Both parties can response the request and set up the communication.

- Content-Length: This field indicates the length of the message body, in decimal number of octets.
- Proxy-Authenticate: This field is used to support a proxy authentication operation. Its value is verified by authentication scheme, and the parameters that are applicable to the proxy for the operation.

Message Body

A *message body* is the data that describes the properties of the session. Before we set up a session, the participants must agree on the media they will use to communicate with each other. This media is described in the message body using SDP (Session Description Protocol), which is one of the most important supporting protocols to IP Call processing. SDP defines a session as a set of media streams. Due to the properties of the media streams, a SDP description should contain the following information about a session:

– The name of the session and its purpose.

– The time during which the session will be active.

– The information needed to build a session, such as media type, transport protocol, media format etc.

Here is an example of a message body:

```
v=0
o=mhandley 29739 7272939 IN IP4 126.5.4.3
s=SIP Call
t=3149328700 0
c=IN IP4 135.180.130.88
m=audio 49210 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC/8000
```

Figure 1.10 *Message Body*

A message body contains several optional fields. The normally used optional fields are explained in the following:

v=Protocol version

o=owner/creator and session identifier

s=session name

c=connection information

u=URI of description

e=e-mail address

t=time the session is active

r=repeat times

m=media and transport address

4.2 SIP Response

The *SIP response* is the SIP message to indicate the state to a request. The difference between a request and a response in the message structure is only the start line. So, the response message can be shown as:

Response = Response-line
(general-header — response-header — entity-header)*
CRLF
[message-body]

The response line consists of SIP-Version, Status-Code and Reason-Phrase, where the SIP-Version is the version of the protocol being used by the message, for example, SIP/2.0. The Status-Code is a 3 digit value. The Reason-Phrase is a short text string that explains Status-Code. The example of a SIP response message is given below:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 135.180.130.133
From: sip:home@container.com
To: sip:office@tudelft.nl
Call-ID: 0077@10.0.0.1
CSeq: 1 INVITE
Content-Length: 0
```

Figure 1.11 *Response SIP Message*

SIP has six classes of response status code, using the first digit to indicate the different classes:

Code	Class	Description	Final
1xx	Informational	Request received, continuing	No
2xx	Success	Request was received, understood and accepted	Yes
3xx	Redirection	Need further action to complete the request	Yes
4xx	Client error	Request cannot be fulfilled at this time	Yes
5xx	Server error	Server failed upon an apparently valid request	Yes
6xx	Global failure	Request cannot be fulfilled anywhere	Yes

Table 1.2 *SIP Response Table*

- **1xx:** Information responses. It indicates that the request has been received and is being processed. Note that at this time, it does not know if the request is forwarded successful. Typically, the proxy server sends a response message with the status code 100 (Trying) when it starts processing an INVITE and user agents send response message with the status code 180 (Ringing) that means the callee has received the request. This is similar to the ringing of a traditional telephone.
- **2xx:** Success response. The request has been successfully processed and accepted. For example, when the callee pick up the phone, the response with code 200 (OK) will be sent. A UAC may receive several 200 messages to a single INVITE request because of forking proxy. The forking proxy can fork the request so it will reach several UAS and each of them will accept the invitation and reply 200 response. In this case, each response is distinguished by the tag parameter in *To* header field.
- **3xx:** Redirection responses. It means the server has to do more actions to complete the request process. A redirection response presents information about the user's new location or an alternative service that the caller might use to reach the callee. Usually, it it sent by proxy server. For example, when a proxy server receives a request but can't process it for any reason, it will send a redirection response to the caller and put another location into the response. The caller can use this new location information to find the callee.
- **4xx:** Client error responses. The request had an error or could not be processed by the server.

- **5xx:** Server error responses. The request is valid but the server failed in processing.
- **6xx:** Global failure responses. The request could not be processed by any server. The message body of a response message is same as the message body that we have introduced in request message. Normally, the response message doesn't have the message body.

5. SIP vs. H323

H.323 is an ITU-T recommendation and acts as an umbrella for several other protocols, which together provide services similar to SIP. H323 was initially designed for use with reliable transport protocols and to provide multimedia conferencing over switched circuit networks.

In the era of network convergence, a key challenge for the network operators and service providers is how to ensure interoperability between different communication protocols. SIP has been widely accepted by service providers because it can deliver enhanced services over next-generation networks. SIP supports interoperability with H.323 and ISUP (ISDN User Part) -- key protocols from both the IP and SS7 environments and hence gives service providers an advantage to offer new SIP services that can go well beyond VoIP. SIP interoperability has been demonstrated in SIP bakeoffs. The purpose of the bake-off is to test for interoperability of SIP implementations, determine the source of incompatibilities, and if the specification is at fault, prepare a "fix" for the SIP draft revision. So far 6 bakeoffs have taken place and leading SIP-products vendors have participated. The number of companies joining these bakeoffs has increased tremendously since the first SIP bakeoff of April 1999.

In comparison to SIP, H.323 offers a diverse range of services for multimedia conferencing which at the same time means rather complex. It is based on reliable transport for the call signalling and uses a binary message representation, while SIP uses simple, text-based messages that can be sent using unreliable transport protocols.

There are many similarities in terms of the services provided by the SIP and H.323, and both use RTP for media transport and run over IP. With H.323v3, UDP and TCP can be used for call setup messages, as SIP does. SIP does not offer any of the conference control provided by H.225.0, instead it relies on other protocols to provide such facilities. SIP has a loop detection algorithm, while H.323 solves that problem by defining a maximum number of gatekeepers a message is allowed to traverse.

SIP uses a more user oriented scheme for endpoint location, the e-mail like SIP URL, while H323 uses H323ID and E.164, which are more telephony oriented.

The capability of negotiation for deciding what media types to use for the multimedia session is more flexible with H323, which is provided by ITU, that with SIP.

H.323	SIP
Robust but consumes more call set up time.	Simple, scalable, and extensible.
Requires about twelve packets for call-setup.	Requires about four packets for call-setup.
Provides floor control within a session.	Cannot provide.
Has more elaborate capability exchange (H.245).	Minimal capability exchange, enough for IP telephony.
Provides a multipoint controller for conferences.	Not required for SIP multicast conferences.
Requires both TCP and UDP during the call-setup.	Basically runs on UDP. Reliability achieved through retransmissions. Supports TCP also, if UDP is not supported.
Implementation is complex and time-taking.	Easy to implement.

Table 1.3 *SIP vs H.323 Protocol*

6. SIP Applications

SIP has been described as “a simple protocol with profound implications”. It addresses many of the major issues of the development of Internet telephony — a technology that is predicted to change the way businesses and people talk to each other. The main applications implemented with SIP are:

- Unified Communications — A SIP session can contain any combination of media (voice, data, video, etc.). These sessions can be modified at any time by adding new parties or by changing the nature of the session. SIP allows browsers to become augmented with multimedia capability. Using SIP, simple, but very powerful, services like click-to-dial become possible. User profiles can be managed through a web interface and voice plug-ins are incorporated into browser technology. SIP uses MIME, the de facto standard for describing content on the Internet to convey information about the protocol used to describe the session and has an URL-style addressing system. It uses the Domain Name System (DNS) to deliver requests to the server that can appropriately handle them.
- Unified Messaging, e-mail, voicemail, faxes, and phone messages are accessible from the same box. Alternatively, people use many different devices to communicate. Unified messaging helps people that use different communication devices, media, and technologies to communicate at any time and under their own control.
- Directory Services — Directory services are to a network what white pages are to the telephone system. They store information about things in the real world, such as people, computers, printers, and so on, as objects with descriptive attributes. People can use the service to look up objects by name; or, like the yellow pages, they can be used to look up services. Network managers use directories to manage user accounts and network resources. From a manager's viewpoint, a directory service is like an inventory of all the devices on the network. Any device can be

located by using a graphic interface or by searching for its name or some properties (e.g., "colour printer"). Once located, a manager can control the device (e.g., disable it or block certain users from accessing it). The directory is a central database where all objects and users are managed.

- IP-PBX functionality — Software based IP_PBX that is compliant with the SIP standard can be utilized in a single office setting or multiple office locations, offering flexibility and options for future expansions.
- Voice-enhanced e-commerce — a website contains click-to dial links that establish a session between the end user and the website organization. This kind of service could be a part of a value-added web-hosting service offered by a service provider or it could be developed by an enterprise's IT department.
- Web Call Centers — a web page may be popped when a particular number is called (with SIP, it is just as easy to direct an user to a web page as it is to a telephone). SIP supports IVR (*Interactive Voice Response*) features, navigating users through options and providing auto-responses to common requests. In addition, SIP's forking facility is perfect for fulfilling the ACD (*Automated Call Distribution*) function.
- Instant Messaging (IM) and Presence — because a SIP session can consist of any form of communication, it is possible to promote an IM session to a telephone call or even a whiteboard or video session at the click of a button. It is also easy to invite other people to join your session, creating spontaneous conference calls. Using third party call control, a conference service could even check the presence status of people due to join a conference and when all the parties are available it could establish the session by connecting them all to a conference bridge. Presence goes hand-in-hand with the evolution of voice services. A network that has dynamically updated information about a user's preferences and availability can perform more intelligent call routing than today's PSTN or existing find-me/follow-me services.
- Mobile phones and PDAs — Because SIP client software is lightweight, it can be embedded in mobile phones and PDAs so that these services can cross all platforms. Using SIP as the signalling protocol means that sessions can be

established between different devices that then negotiate the appropriate media capability. These devices become means of accessing those services associated with a user instead of being closed, proprietary systems.

- Wireless LAN VoIP Telephone Handsets — dedicated portable telephone handsets, supporting Voice over IP on an 802.11 wireless LAN connection. They may use SIP and other proprietary protocols (i.e., Skinny) and may also support wireless telephony protocols (i.e., GSM)
- Desktop Call Management – SIP enables a convergence at the desktop. Voice services can be assimilated into other applications to change the way we use our computers. The information management capabilities of the Internet can be used to transform communication systems and improve productivity. Using SIP features such as user profiling, presence management and instant messaging, third party call control and integration with media, many services can be created by service providers or enterprise IT departments. All the advanced telephony services inherited from the Intelligent Network are supported by SIP. This includes services such as call forwarding, call hold, call waiting, etc.
- SIP can be integrated into product such as:
    - ✓ IP phones.
    - ✓ Media Gateways.
    - ✓ Web-enabled telephony portals.
    - ✓ Internet call- centers
    - ✓ Soft switches.
    - ✓ Application servers.

CHAPTER 2

1. TCP essentials for wired networks

The most popular transport layer protocol on the Internet is TCP, which offers a reliable byte stream service. TCP is a reliable stream delivery service that guarantees delivery of a data stream sent from one host to another without duplication or losing data. Since packet transfer is not reliable, a technique known as positive acknowledgment with retransmission is used to guarantee reliability of packet transfers. This fundamental technique requires the receiver to respond with an acknowledgment message as it receives the data. The sender keeps a record of each packet it sends, and waits for acknowledgment before sending the next packet. The sender also keeps a timer from when the packet was sent, and retransmits a packet if the timer expires. The timer is needed in case a packet gets lost or corrupted.

TCP provides transparent segmentation and reassembly of user data and handles flow and congestion control. TCP packets are cumulatively acknowledged as they arrive in sequence, with out of sequence packets causing duplicate acknowledgments to be generated. The sender detects a loss when multiple duplicate acknowledgments (usually 3) arrive, implying that the next packet was lost. IP may reorder datagrams, thus TCP cannot immediately assume that all gaps in the packet sequence signify losses. When the session becomes idle or acknowledgments are lost, TCP detects losses using timeouts. Retransmission timers are continuously updated based on a weighted average of previous round trip time (RTT) measurements. Accuracy is critical, since delayed timeouts slow down recovery, while early ones may lead to redundant retransmissions.

A prime concern for TCP is congestion. Congestion occurs when routers are overloaded with traffic that causes their queues to build up and eventually overflow, leading to high delays and packet losses. Since most Internet traffic is carried by extremely reliable wired links, TCP assumes that all losses indicate congestion.

Therefore, when losses are detected, besides retransmitting the lost packet, TCP also reduces its transmission rate, allowing router queues to drain. Subsequently, it gradually increases its transmission rate so as to gently probe the network's capacity, which reflects the network's efficiency in terms of throughput and link utilization.

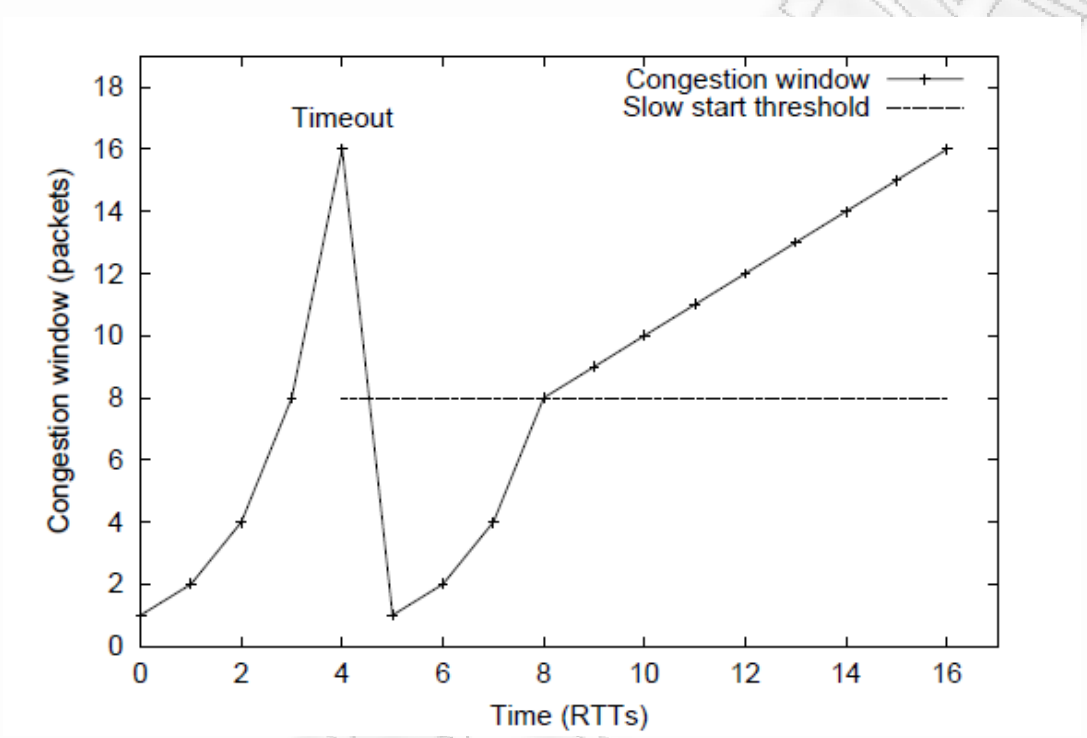

Figure 2.1 *Example on how timeout – network congestions affect performance.*

TCP maintains a *congestion window*, which is an estimate of the number of packets that can be in transit without causing congestion. New packets are only sent if allowed by both this window and the receiver's advertised window. The congestion window starts at one packet, with new acknowledgments causing it to be incremented by one, thus doubling after each RTT. This is the slow start phase (exponential increase).
In figure 2.1 an example is shown of how TCP handles network congestion. A slow start threshold is then set to half the value of the congestion window, the congestion window is reset to one packet, and the lost packet is retransmitted. Slow start is repeated until the threshold is reached after 3 RTTs, allowing routers to drain their queues. Subsequently, the congestion window is incremented by one packet per RTT. This is the congestion

avoidance phase (linear increase). When losses are detected by duplicate acknowledgments, indicating that subsequent packets have been received, TCP retransmits the lost packet, halves the congestion window, and restarts with the congestion avoidance phase

2. TCP in wireless environments

2.1 TCP limitations in wireless links

TCP was originally designed primary for wired networks. In such environments, packet losses are mainly considered to be caused by network congestion. However, this assumption does not hold in wireless networks, in which high bit error rates, unstable channel characteristics, user mobility may lead to packet losses. In this case, studies have shown that the unmodified standard TCP performs poorly in a wireless environment due to its inability to distinguish packet losses caused by network congestion from those attributed to transmission errors.

Unfortunately, when packets are lost in networks for reasons other than congestion, these measures result in an unnecessary reduction in end-to-end throughput and hence, sub-optimal performance. Communication over wireless links is often characterized by sporadic high bit-error rates, and intermittent connectivity due to handoffs. TCP performance in such networks suffers from significant throughput degradation and very high interactive delays.

In [20], research has identified several TCP limitations that cause its performance to be unacceptable for wireless links, without some modification. Wireless links exhibit much higher BERs than wired links due to factors such as urban obstacles, user mobility, multipath interferences and more. Limitation in radio coverage and user varying location require handoffs, which result in **temporal disconnections and reconnections**. In [20], it has been shown that a short disconnection event can actually stall the TCP transmission for a much longer period. Figure 2.2 demonstrates this:

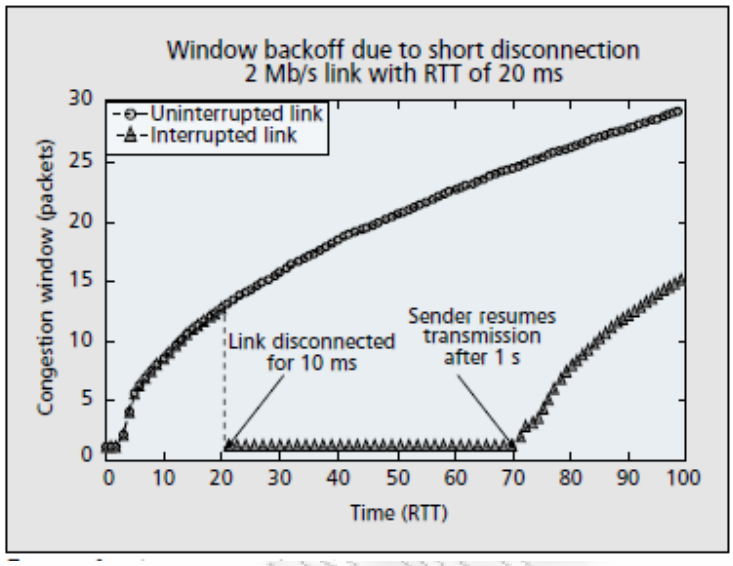

Figure 2.2 *Disconnection effect in TCP*

Another limitation underlined in [20], is that standard TCP cannot handle the high BER and frequent disconnections effectively. The fast retransmit and fast recovery algorithms introduced by TCP Reno [27] can recover from sporadic random packet losses fairly quickly **if such losses only occur once within an RTT**. However, noises and other factors in the wireless environment usually cause random bit errors to occur in short bursts, thus leading to a higher probability of multiple random packet losses within one RTT. This is shown in Figure 2.3

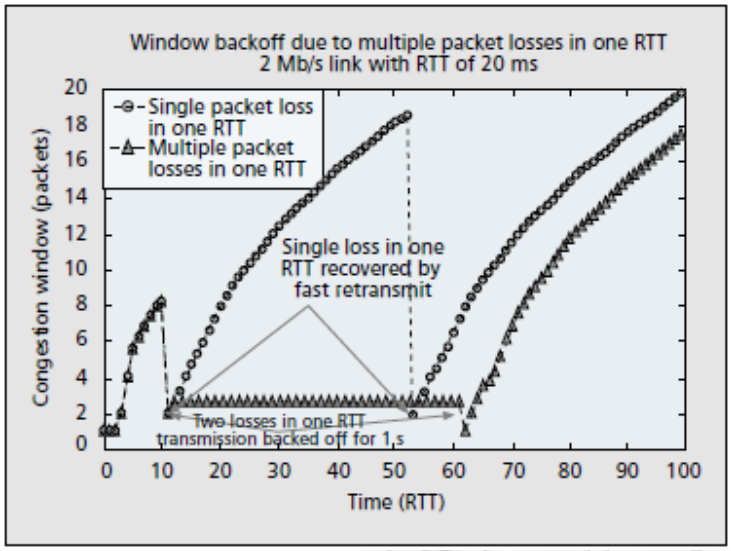

Figure 2.3 *Multiple packet losses in one RTT*

2.2 Improvements to Wireless TCP performance

In [21], [22], [25], [26] different techniques are mentioned:

- **Link-layer protocols:**

Instead of modifying TCP, we may hide wireless losses from it. The main advantage of employing a link-layer protocol for loss recovery is that it fits naturally into the layered structure of network protocols. The link-layer protocol operates independently of higher-layer protocols and does not maintain any per-connection state.

The two main classes of techniques [28] employed by these protocols are: error correction, using techniques such as forward error correction (FEC), and retransmission of lost packets in response to automatic repeat request (ARQ) messages.

- **Split connection protocols**

Split connection protocols [29] split each TCP connection between a sender and receiver into two separate connections at the base station — one TCP connection between the sender and the base station, and the other between the base station and the receiver.

Over the wireless hop, a specialized protocol tuned to the wireless environment may be used.

- **The Snoop Protocol**

The snoop protocol [30] introduces a module, called the snoop agent, at the base station. The agent monitors every packet that passes through the TCP connection in both directions and maintains a cache of TCP segments sent across the link that have not yet been acknowledged by the receiver. A packet loss is detected by the arrival of a small number of duplicate acknowledgments from the receiver or by a local timeout. The snoop agent retransmits the lost packet if it has it cached and suppresses the duplicate acknowledgments

- **Transport Layer Solutions**

The degraded performance of TCP over wireless links is mostly due to mistaking wireless losses for congestion. During handoffs packets may be delayed or even lost. Recovery from these losses should be initiated right after handoff completion, without waiting for a timeout. TCP can achieve this by receiving appropriate signals from lower layers [22], [23]. Alternatively, TCP can exploit mobility hints from lower layers to heuristically distinguish losses due to handoffs. For these losses, TCP can avoid halving the slow start threshold during recovery, thus skipping the congestion avoidance phase.

Another approach is for wireless link endpoints to choke TCP senders during handoffs, by transparently closing the receiver's advertised window [31]. The sender then freezes all pending timers and starts periodically probing the receiver's window. Shrinking the advertised window however violates TCP guidelines. After handoffs, congestion avoidance helps probe the capacity of the new link.

The Eifel scheme modifies TCP so as to avoid the spurious timeouts and fast retransmits due to handoffs or delayed link layer retransmissions [22]. Since these problems are due to TCP's inability to distinguish between acknowledgments for original packet transmissions and retransmissions, Eifel adds TCP timestamps to outgoing

packets. Timestamps are echoed in acknowledgments, thus allowing spurious timeouts to be easily avoided, without changing TCP semantics. The end-to-end TCP recovery however is not accelerated. While TCP enhancement schemes would be attractive if only the endpoints needed modifications, in practice additional changes are needed. Some approaches require signaling from lower layers to detect handoffs. Others require software to be installed and state to be maintained at pivot points. In addition, split TCP schemes need alternative, TCP compatible, protocols to be deployed over wireless links for more efficient error recovery.

CHAPTER 3

1. INTRODUCTION

The explosive demand for wireless services in the recent years has led to the development and deployment of new generation of wireless systems that offer a variety of high speed wireless applications including voice and data services, multimedia services, navigation services, text-video messaging, and internet browsing [35]. It becomes increasingly clear that the dominant consideration in the design of systems employing such technologies will be their ability to perform with adequate margin over a channel perturbed by a host of impairments, such as multipath fading.

In a wireless network, the instantaneous received signal suffers from short-term fading which is usually modelled by the Rayleigh, Rician or Nakagami probability density functions (pdfs) and from slow variation of its local mean power (known as long-term fading) that is usually modelled by the lognormal distribution. A parameter that is commonly used to quantify the effect of channel fading on the performance of a wireless network is the minimum time (duration) that the received signal level stays below a preset threshold, causing the call to be dropped by the base station. The minimum duration for an outage event is derived using a level crossing analysis and it usually spans widely different time-scales for the short-term and the long-term fading channels [36].

In this thesis, it is tried to model the probability of link breakdown against some variables, such as fade margin, velocity of the mobile user, or time duration of fading in a short term (Rayleigh) fading environment and long term (lognormal) fading environment.

2. FADING CHANNEL MODELING

The distinction between slow and fast fading is important for the mathematical modeling of fading channels and for the performance evaluation of communication systems operating over these channels. This notion is related to the *coherence time Tc* of the channel, which measures the period of time over which the fading process is correlated (or equivalently, the period of time after which the correlation function of two samples of the channel response taken at the same frequency but different time instants drops below a certain predetermined threshold). The coherence time is also related to the channel *Doppler spread* f_d by

$$T_c \simeq \frac{1}{f_d}$$

The fading is said to be slow if the symbol time duration Ts is smaller than the channel's coherence time Tc; otherwise it is considered to be fast. In slow fading a particular fade level will affect many successive symbols, which leads to burst errors, whereas in fast fading the fading decorrelates from symbol to symbol.

2.1 Rayleigh fading

Rayleigh fading is a statistical model for the effect of a propagation environment on a radio signal, such as that used by wireless devices.

Rayleigh fading models assume that the magnitude of a signal that has passed through such a transmission medium (also called a communications channel) will vary randomly, or fade, according to a Rayleigh distribution — the radial component of the sum of two uncorrelated Gaussian random variables.

Rayleigh fading is viewed as a reasonable model for tropospheric and ionospheric signal propagation as well as the effect of heavily built-up urban environments on radio signals [6]. Rayleigh fading is most applicable when there is no dominant propagation along a line of sight between the transmitter and receiver. If there is a dominant line of sight, Rician fading may be more applicable.

The requirement that there be many scatterers present means that Rayleigh fading can be a useful model in heavily built-up city centres where there is no line of sight between the transmitter and receiver and many buildings and other objects attenuate, reflect, refract and diffract the signal.

In troposphere and ionosphere signal propagation the many particles in the atmospheric layers act as scatterers and this kind of environment may also approximate Rayleigh fading. If the environment is such that, in addition to the scattering, there is a strongly dominant signal seen at the receiver, usually caused by a line of sight, then the mean of the random process will no longer be zero, varying instead around the power-level of the dominant path. Such a situation may be better modelled as Rician fading.

The instantaneous SNR per symbol of the channel γ is distributed according to an exponential distribution given by

$$p_\gamma(\gamma) = \frac{1}{\overline{\gamma}} \exp\left(-\frac{\gamma}{\overline{\gamma}}\right)$$

The MGF corresponding to this fading model is given by

$$M_\gamma(s) = (1 - s\overline{\gamma})^{-1}$$

In addition, the moments associated with this fading model can be expressed by

$$E(\gamma^k) = \Gamma(1 + k)\ \overline{\gamma}^k$$

where $\Gamma(.)$ is the gamma function.

The Rayleigh fading model therefore has an AF equal to 1, and typically agrees very well with experimental data for mobile systems where no LOS path exists between the transmitter and receiver antennas

2.2 Log Normal fading

In terrestrial and satellite land-mobile systems, the link quality is also affected by slow variation of the mean signal level due to the shadowing from terrain, buildings, and trees. Communication system performance will depend only on shadowing if the radio receiver is able to average out the fast multipath fading or if an efficient "micro" diversity system is used to eliminate the effects of multipath. Empirical measurements reveal a general consensus that shadowing can be modeled by a log-normal distribution for various outdoor and indoor environments [38], [39], in which case the path SNR per symbol γ has a PDF given by the standard lognormal expression

$$p_\gamma(\gamma) = \frac{\xi}{\sqrt{2\pi}\sigma\gamma} \exp\left[-\frac{(10\log_{10}\gamma - \mu)^2}{2\sigma^2}\right]$$

where ξ = 10/ ln(10) = 4.3429, and μ (dB) and σ (dB) are the mean and the standard deviation of 10 log10 γ, respectively.

The gamma distribution can be used as a substitute to the log-normal distribution to describe the shadowing phenomenon on terrestrial and satellite channels. The advantage of using the gamma distribution as an alternative to the log-normal distribution is that it can lead to simpler composite multipath/shadowing models.

2.3 Rice (Nakagami-n)

The Nakagami-n distribution is also known as the Rice distribution. It is often used to model propagation paths consisting of one strong direct LOS component and many random weaker components. Here the channel fading amplitude follows the distribution

$$p_\alpha(\alpha) = \frac{2(1+n^2)e^{-n^2}\alpha}{\Omega} \exp\left(-\frac{(1+n^2)\,\alpha^2}{\Omega}\right) I_0\left(2n\alpha\sqrt{\frac{1+n^2}{\Omega}}\right), \qquad \alpha \geq 0$$

where *n* is the Nakagami-*n* fading parameter, which ranges from 0 to ∞.

This parameter is related to the Rician K factor by $K = n^2$ which corresponds to the ratio of the power of the LOS component to the average power of the scattered component.

3. Link Breakdown Probability in a Fading Environment

It is well known that a wireless link is inherently time-varying and susceptible to performance degradation due to signal fading, interference, and noise in the physical link. When the link is severely degraded, a call in progress will be terminated and disconnected from the base station. In [41], an analytical model to study the interaction between the Rayleigh fading in the physical channel and the wireless network performance was introduced through the probability of link breakdown, which transfers the effects of physical layer characteristics (e.g., carrier frequency, Doppler frequency, and fade margin) to higher layer performance metrics of the wireless network. Moreover, the probability of link breakdown in a fading channel is usually studied in terms of the minimum link outage duration [42].

Outage probability is an important measure of the quality of a wireless link. It represents the probability that the instantaneous received signal-to-noise and interference ratio (SNIR) falls below a preset threshold. However, in practice, it is not the instantaneous drop of the SNIR below the threshold that is really important but the duration that it stays below the threshold [43]. Consequently, a link breakdown or outage event may be defined as the event that the received SNIR stays below the system threshold for a time period longer than a minimum duration τ_m. The fade duration τ_f is the time that the received signal stays below the required threshold. Therefore, the allowed minimum fade duration τ_m is the minimum value of τ_f that the system can tolerate without losing its connection to the network. Furthermore, the link breakdown duration τ_{link} is defined as the fade duration τ_f, given that τ_f is greater than or equal to τ_m. The pdf of link breakdown duration is given by [44]

$$f_{\tau_{link}}\left(\tau_{link}\right)=\begin{cases}\dfrac{f_{\tau_f}\left(\tau_{link}\right)}{\Pr\left(\tau_f \geq \tau_m\right)}, & \tau_{link} \geq \tau_m \\ 0, & \text{otherwise}\end{cases} \tag{1}$$

where $f_{\tau_f}(\tau)=\dfrac{-d\Pr\left(\tau_f \geq \tau\right)}{d\tau}$ and $1-\Pr\left(\tau_f \geq \tau\right)$ are, respectively, the pdf and the cdf of τ_f.

Then the average duration of link breakdown can be evaluated as

$$E(\tau_{link}) = \int_{\tau_m}^{\infty} \tau_{link} f_{\tau_{link}}(\tau_{link})\, d\tau_{link} \,. \tag{2}$$

The frequency of link breakdown is given by [13]

$$\begin{aligned} f_{link} &= \{\textit{Level crossing rate}\} \times \{\textit{Probability that crossing leads to link breakdown}\} \\ &= N_R \Pr(\tau_f \geq \tau_m) \end{aligned} \tag{3}$$

where N_R is the level crossing rate. Finally, the probability of link breakdown is given by

$$p_{link} = f_{link}\, E(\tau_{link}) \,. \tag{4}$$

3.1 Probability of link breakdown due to Rayleigh fading

In this section, we assume a pure multipath fading environment where fast fluctuations due to multipath fading dominate in the received signals and the effect of lognormal shadowing can be ignored. Using level crossing analysis, the link breakdown may be studied as a problem of a single Rayleigh signal envelope which fades below the required signal target [41], [44].

It is well known that the average fade duration $\overline{\tau}_f \triangleq E(\tau_f)$ for a Rayleigh-faded signal is given by [41]

$$\overline{\tau}_f = \frac{\exp(\rho^2) - 1}{\sqrt{2\pi} f_d \rho}, \tag{5}$$

while the level crossing rate is given by

$$N_R = \sqrt{2\pi} f_d\, \rho \exp(-\rho^2) \tag{6}$$

where

f_d is the maximum Doppler frequency and $\rho = R_{req} / R_{rms}$ is the ratio of the required amplitude level to the local root mean square (rms) of the instantaneous received signal amplitude.

Also, the cdf of fade duration τ_f is given by

$$\Pr\left(\tau_f \geq \tau\right) = 2\left(\frac{\overline{\tau}_f}{\tau}\right)\exp\left(-\frac{2}{\pi}\left(\frac{\overline{\tau}_f}{\tau}\right)^2\right) I_1\left(\frac{2}{\pi}\left(\frac{\overline{\tau}_f}{\tau}\right)^2\right) \tag{7}$$

where $I_1(\cdot)$ is the modified Bessel function of the first kind [30]. Therefore, the average duration of link breakdown is given by

$$E\left(\tau_{link}\right) = \frac{\overline{\tau}_f}{\Pr\left(\tau_f \geq \tau_m\right)} \cdot \int_{\tau_m/\overline{\tau}_f}^{\infty} (-u)\, d\left[\frac{2}{u}\exp\left(\frac{-2}{\pi u^2}\right) I_1\left(\frac{2}{\pi u^2}\right)\right] \tag{8}$$

where $u = \tau_{link} / \overline{\tau}_f$.

The integral in (8) can be evaluated in closed-form as

$$E\left(\tau_{link}\right) = \frac{\tau_m}{2} + \frac{\tau_m}{2}\left\{\frac{1}{\exp\left(\frac{-2}{\pi}\left(\frac{\overline{\tau}_f}{\tau_m}\right)^2\right) I_1\left(\frac{2}{\pi}\left(\frac{\overline{\tau}_f}{\tau_m}\right)^2\right)} - \frac{I_0\left(\frac{2}{\pi}\left(\frac{\overline{\tau}_f}{\tau_m}\right)^2\right)}{I_1\left(\frac{2}{\pi}\left(\frac{\overline{\tau}_f}{\tau_m}\right)^2\right)}\right\}. \tag{9}$$

By using (6) and (7), the frequency of link breakdown in (3) is given by

$$\mathrm{f}_{link} = 2\sqrt{2\pi} f_d \cdot \rho \cdot \exp\left(-\rho^2\right) \cdot \left(\frac{\overline{\tau}_f}{\tau_m}\right) \cdot \exp\left(-\frac{2}{\pi}\left(\frac{\overline{\tau}_f}{\tau_m}\right)^2\right) \cdot I_1\left(\frac{2}{\pi}\left(\frac{\overline{\tau}_f}{\tau_m}\right)^2\right). \tag{10}$$

The probability of link breakdown in a Rayleigh fading channel can then be computed using (9) and (10) in (4).

3.2 Probability of link breakdown due to lognormal shadowing

The link breakdown condition in lognormal shadowing is usually formulated on a dB scale as the level crossings of a Gaussian process with variance σ^2 about the threshold. The level crossing rate of the received signal in a lognormal fading channel is given by

$$N_R = \frac{v_m}{2\pi d_c} \exp\left(-\frac{\gamma^2}{2\sigma^2}\right) \tag{11}$$

where d_c is the correlation distance [2] and γ is the fade margin in dB. The pdf of fade duration below the level τ_f is given by

$$f_{\tau_f}(\tau) = (2\lambda)\tau \exp\left(-\lambda\tau^2\right) \tag{12}$$

where $\lambda = \frac{1}{2}\left(\frac{\gamma v_m}{2\sigma d_c}\right)^2$.

Using (12) and (1) in (2), the average duration of link breakdown can be shown to be given by

$$E(\tau_{link}) = \tau_m + \sqrt{(\pi/\lambda)} \exp\left(\lambda\tau_m^2\right) Q\left(\sqrt{2\lambda}\,\tau_m\right) \tag{13}$$

where $Q(x) = 1/\sqrt{2\pi}\int_x^\infty e^{-t^2/2}dt$ is the tail of the Gaussian integral.

Similarly, the frequency of link breakdown for a lognormal shadowing channel is given by

$$f_{link} = \frac{v_m}{2\pi d_c} \exp\left(-\lambda\left(\tau_m^2 + \frac{4d_c^2}{v_m^2}\right)\right). \tag{14}$$

4. Results – Plots

4.1 Rayleigh fading

Probability of link breakdown due to Rayleigh fading can be calculated with the theory presented at section 3. Using Matlab we can calculate this probability against several variables that is related with.

- against the minimum duration of link breakdown:

```
clear;

f_c = 1900*10^6;

v_m = 30*0.447;
ro_db=-8;

ro = 10^(ro_db/20);

fd = v_m*f_c/(3*10^8);

t_f = (exp(ro^2)-1)/(sqrt(2*pi)*fd*ro);

N_R = sqrt(2*pi)*fd*ro*exp(-ro^2);

i=1;
for t_m=0.001:0.001:0.03

a = t_f/t_m;
b = exp((-2/pi)*a^2)*besseli(1,(2/pi)*a^2);
f_link = 2*N_R*a*b;

t_link = (t_m/2)+(t_m/2)*((1/b)-
(besseli(0,(2/pi)*a^2)/besseli(1,(2/pi)*a^2)));

%t2_link = t_m - (N_R*t_f/f_link)*(exp((-2/pi)*a^2)*((besseli(0,-
(2/pi)*a^2))-besseli(1,-(2/pi)*a^2))-1)

p_link_30_8(i) = t_link*f_link;
tm(i)=t_m;
i=i+1;
end

semilogy(tm, p_link_30_8,'k-')
```

```
axis([0 0.03 10^(-6) 10^(0)])
xlabel('\fontsize{10}\it\tau_m \rm(s)')
ylabel('\fontsize{10}Probability of link breakdown')
```

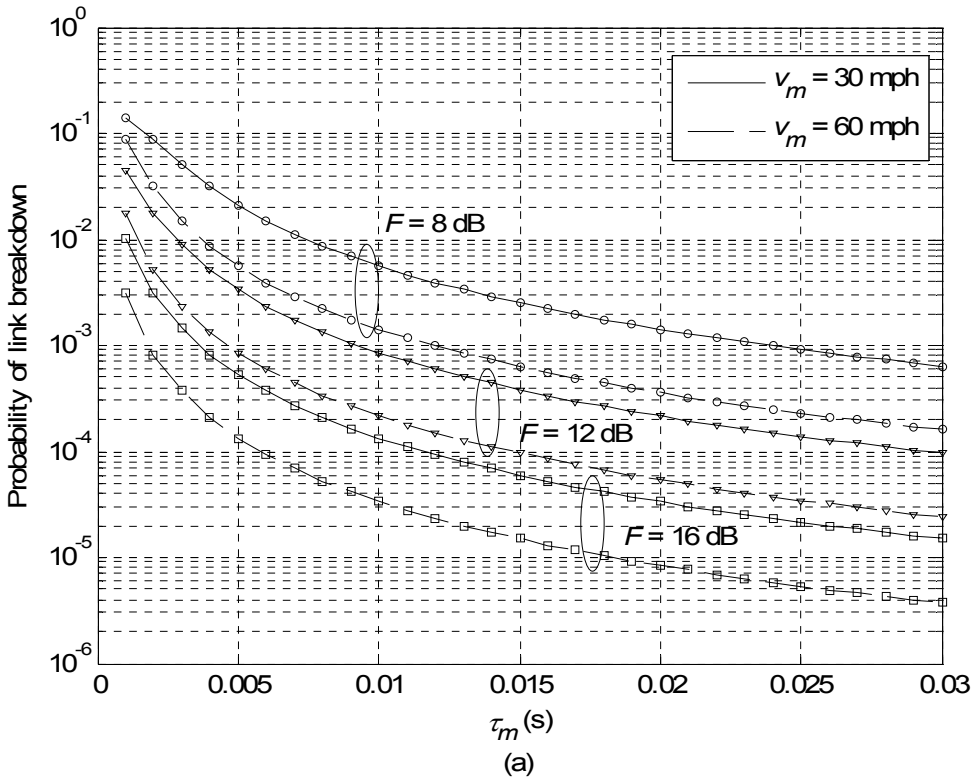

Figure 3.1 *Probability of link breakdown vs. tm*

From the above we can derive that as fading margin increases, probability of link breakdown improves. In addition, as the minimum duration of link breakdown increases the probability of link breakdown decreases.

From the plot, it can be seen as well that as the speed of the mobile user increases, the value of p_{link} decreases.

- against the fading margin

```
clear;
v_m = 30*0.447;

f_c = 850*10^6;
t_m = 0.001;

i=1;
for ro_db = 0:5:25
```

```
ro_db = -ro_db;
ro = 10^(ro_db/20);

fd = v_m*f_c/(3*10^8);

t_f = (exp(ro^2)-1)/(sqrt(2*pi)*fd*ro);

N_R = sqrt(2*pi)*fd*ro*exp(-ro^2);

a = t_f/t_m;
b = exp((-2/pi)*a^2)*besseli(1,(2/pi)*a^2);
f_link = 2*N_R*a*b;

t_link = (t_m/2)+(t_m/2)*((1/b)-
(besseli(0,(2/pi)*a^2)/besseli(1,(2/pi)*a^2)));

t2_link = t_m - (N_R*t_f/f_link)*(exp((-2/pi)*a^2)*((besseli(0,-
(2/pi)*a^2))-besseli(1,-(2/pi)*a^2))-1)

p_link_8_1(i) = t_link*f_link;
r(i)=-ro_db;
i=i+1;
end

semilogy(r, p_link_8_1,'k--')
axis([0 25 10^(-7) 10^(0)])
xlabel('\fontsize{10}\itF \rm(dB)')
ylabel('\fontsize{10}Probability of link breakdown')
hold on
```

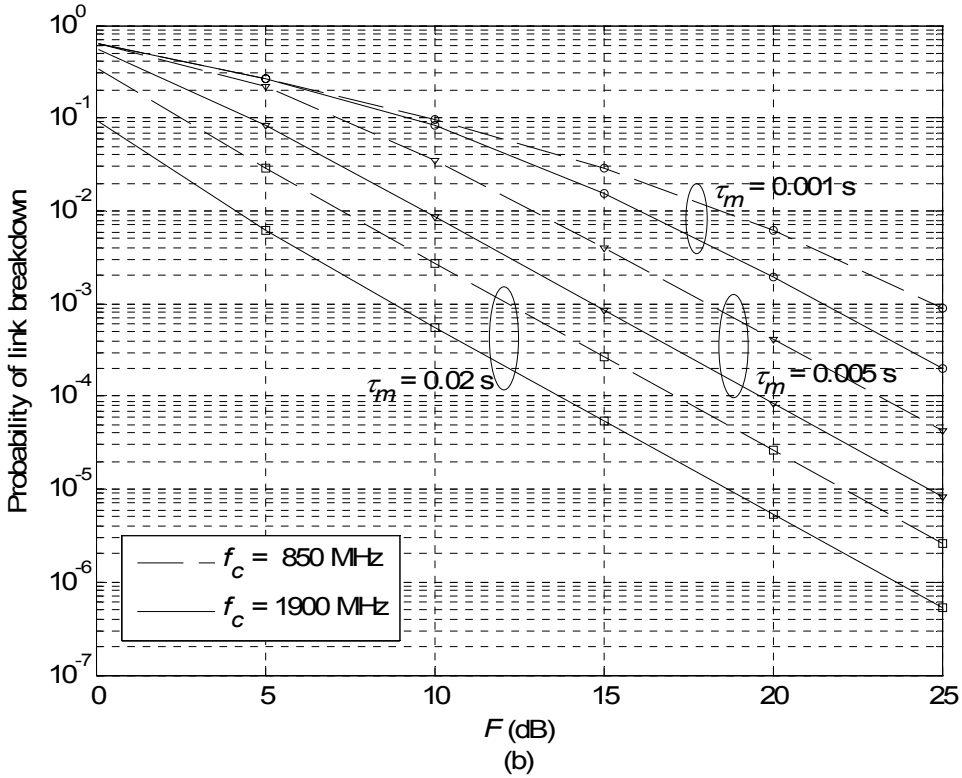

Figure 3.2 *Probability of link breakdown vs fading margin*

From the above figure, it can be derived that when carrier frequency f_c increases, the value of p_{link} decreases.

4.2 Lognormal fading

Similarly, we can calculate the probability of link breakdown due to lognormal fading.

- against the minimum duration of link breakdown:

```
clear;
v_m = 30*0.447;

f_c = 850*10^6;
t_m = 0.001;

i=1;
```

```
for ro_db = 0:5:25

ro_db = -ro_db;
ro = 10^(ro_db/20);

fd = v_m*f_c/(3*10^8);

t_f = (exp(ro^2)-1)/(sqrt(2*pi)*fd*ro);

N_R = sqrt(2*pi)*fd*ro*exp(-ro^2);

a = t_f/t_m;
b = exp((-2/pi)*a^2)*besseli(1,(2/pi)*a^2);
f_link = 2*N_R*a*b;

t_link = (t_m/2)+(t_m/2)*((1/b)-
(besseli(0,(2/pi)*a^2)/besseli(1,(2/pi)*a^2)));

t2_link = t_m - (N_R*t_f/f_link)*(exp((-2/pi)*a^2)*((besseli(0,-
(2/pi)*a^2))-besseli(1,-(2/pi)*a^2))-1)

p_link_8_1(i) = t_link*f_link;
r(i)=-ro_db;
i=i+1;
end

semilogy(r, p_link_8_1,'k--')
axis([0 25 10^(-7) 10^(0)])
xlabel('\fontsize{10}\itF \rm(dB)')
ylabel('\fontsize{10}Probability of link breakdown')
```

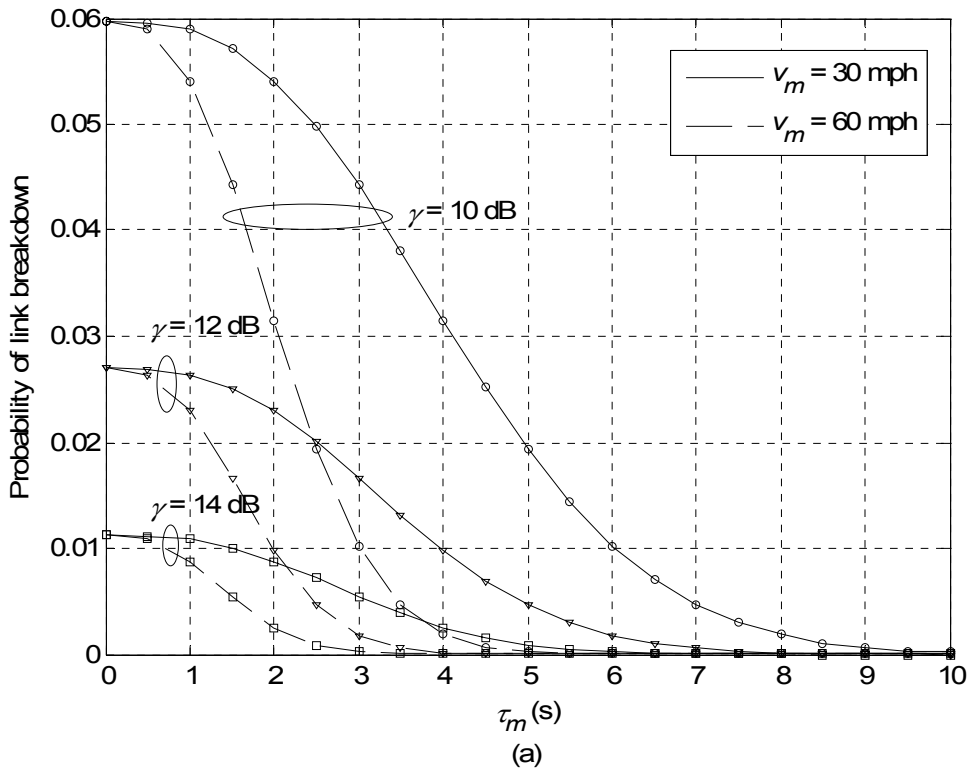

Figure 3.3 *Probability of link breakdown vs time*

From the above plot, we can see that probability of link breakdown increases as fade margin decreases. For faster mobile users, this probability gets smaller. This conclusion was also derived for the Rayleigh fading.

- against the fade margin

```
clear;

sigma = 6;
dc = 30;

v_m = 30*0.447;
t_m = 1;

i=1;
for ro_db=2:2:20
```

```
ro = ro_db/sigma;

lamda = ((ro*v_m/(2*dc))^2)/2;

N_R = (v_m/(2*pi*dc))*exp(-ro^2/2);

a = (v_m/(2*pi*dc));
b = (4*dc^2)/(v_m^2);

f_link = a*exp(-lamda*(t_m^2+b));

t_link = t_m+sqrt(pi/lamda)*exp(lamda*t_m^2)*qfunc(sqrt(2*lamda)*t_m);

p_link_1_30(i) = t_link*f_link;
r(i)=ro_db;
i=i+1;
end
figure(1)
semilogy(r, p_link_1_30, 'k-')
axis([2 20 10^(-7) 10^0])
xlabel('\fontsize{10}\it\gamma \rm(dB)')
ylabel('\fontsize{10}Probability of link breakdown')
```

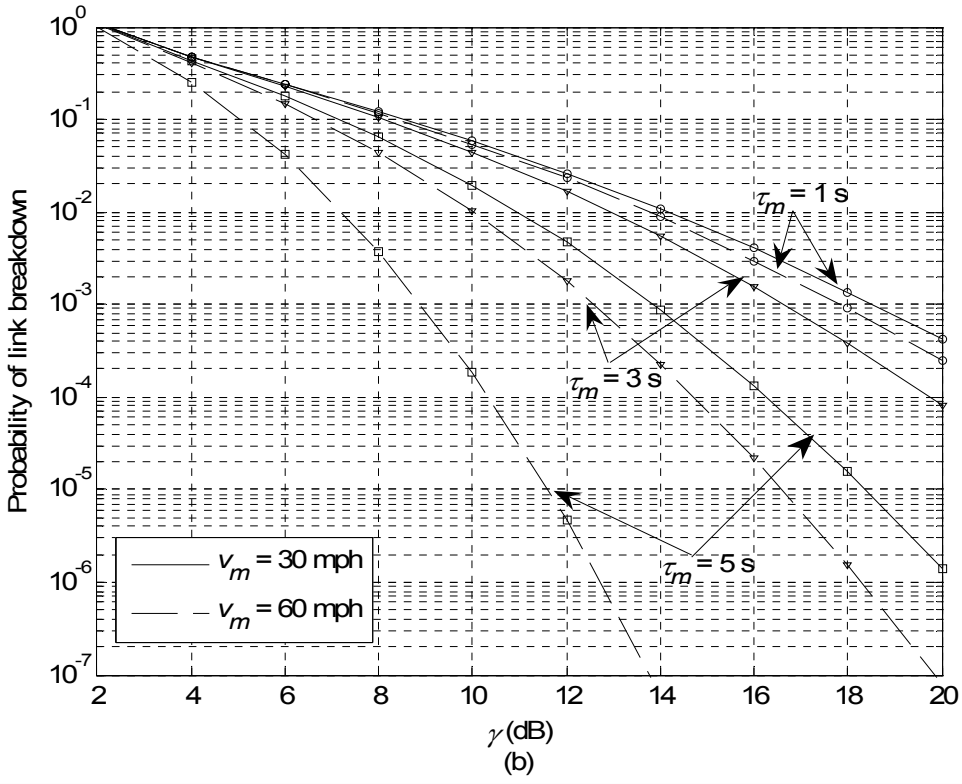

Figure 3.4 *Probability of link breakdown vs Fading margin.*

Again, the plot shows the impact that mobile speed, fading margin and minimum breakdown time values have at the probability of link breakdown. Comparing figure 3.3 and 3.4, it can be seen that velocity has bigger impact at the p_{link} value than t_m.

CHAPTER 4

1. Introduction

In cellular communications, voice and data traffics are usually treated separately. With the deployment of VoIP, it is possible to send voice over data channels (IP based packet switched networks). However, unlike the wire line links, the wireless link is error prone. There are errors due to fading and shadowing, resulting in a high Frame error rate (FER), as much as 10 %. In order to cope with the high FER, the Data link layer in wireless networks includes a Radio Link Protocol (RLP) [50] sub-layer along with the MAC and Logical link sub-layers. RLP was proposed to provide extra reliability to wireless networks.

An important aspect of telephony services is signalling. For VoIP, two signalling schemes are popular. The first one is H.323, specified by ITU-T for the implementation of multimedia services. It is not a single standard but is an umbrella of standards for video conferencing. The second standard is Session Initiation Protocol (SIP) [51] developed by Internet Engineering Task Force (IETF). It is an application layer signalling protocol that can establish, modify and terminate multimedia sessions. SIP is developed exclusively for the Internet unlike H.323 and is similar to HTTP protocol and SMTP in its structures.

2. Radio Link Protocol (RLP)

Radio Link Protocol (RLP) [52] is an automatic repeat request (ARQ) fragmentation protocol used over a wireless (typically cellular) air interface. Most wireless air interfaces are tuned to provide 1% packet loss, and most coders are mutually tuned to sacrifice very little voice quality at 1% packet loss. However, 1% packet loss is intolerable to all

variants of TCP, and so something must be done to improve reliability for voice networks carrying TCP/IP data.

An RLP detects packet losses and performs retransmissions to bring packet loss down to .01%, or even .0001%, which is suitable for TCP/IP applications. RLP also implements stream fragmentation and reassembly, and sometimes, in-order delivery. Newer forms of RLP also provide framing and compression, while older forms of RLP rely upon a higher-layer PPP protocols to provide these functions.

An RLP transport cannot ask the air interface to provide a certain payload size. Instead, the air interface scheduler determines the packet size, based upon constantly changing channel conditions, and up-calls RLP with the chosen packet payload size, right before transmission. Most other fragmentation protocols, such as those of 802.11b and IP use payload sizes determined by the upper layers and call upon the MAC to create a payload of a certain size. These protocols are not as flexible as RLP, and can sometimes fail to transmit during a deep fade in a wireless environment.

An RLP protocol can be ACK-based or NAK-based. Most RLPs are NAK-based, meaning that forward-link sender assumes that each transmission got through, and the receiver only NAKs when an out-of-order segment is received. This greatly reduces reverse-link transmissions, which are spectrally inefficient and have a longer latency on most cellular networks. When the transmit pipeline goes idle, a NAK-based RLP must eventually retransmit the last segment a second time, in case the last fragment was lost, to reach a .01% packet loss rate. This duplicate transmission is typically controlled by a "flush timer" set to expire 200-500 milliseconds after the channel goes idle.

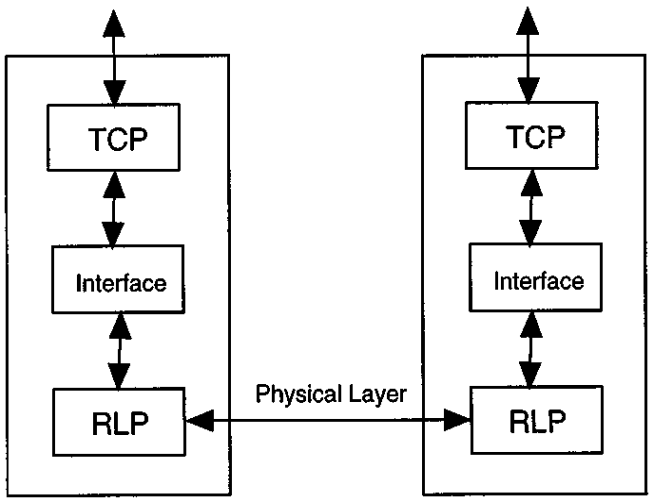

Figure 4.1 *TCP/IP protocol stack*

When RLP at the receiving end finds a frame in error (or as missing), it sends back a NAK requesting retransmission of the lost frame. A NAK retransmission timer is set for the lost frame. A guard interval is added to the retransmission timeout in order to account for the buffering delays and the segmentation of retransmitted frames [54]. When the retransmission timer expires for the first time, RLP resets the timer and sends back NAK twice. Each NAK received at the transmission end triggers a retransmission of the requested frame. When the timer expires for the second time, RLP resets the timer and sends back three NAK's. This process continues until the number of timer expirations reaches a certain limit. RLP will abort the attempt after unsuccessful retransmissions and pass control to the TCP layer.

During the data transfer phase (excluding the link setup and tear-down phases), RLP maintains a 8-b sending sequence number count *V(s)*, and two sequence numbers *V(R)* and *V(N)* for receiving. All operations on these RLP frame sequence numbers are carried out in unsigned modulo 256 arithmetic. *V(S)* is incremented whenever a new RLP data frame of non-zero bytes is sent out. That is, *V(S)* is the sequence number of the next frame to be sent. *V(R)* contains the sequence number of the next new frame expected to be received. *V(N)* is the next frame needed for sequential delivery. In other words, *V(N)* is the oldest sequence number of the missing frames. By denoting the sequence number of a

newly received frame by *i*, the RLP transmission procedure can briefly be described by the following rules: [54]

- If $i < V(N)$ or if the frame is already stored in the resequencing buffer, discard the frame.
- If $i = V(N)$, update $V(N)$ to the next oldest missing frame sequence number. Pass received frames up to $V(N) - 1$ to the upper layer.
- If $V(N) < i < V(R)$, store frame *i* in resequencing buffer if it is missing.
- If $i = V(N) = V(R)$, pass all received frames up to $V(R)$ to the upper layer.
- If $i = V(N) \neq V(R)$ or $i > V(R)$, increment $V(R)$ and store frame *i* into resequencing buffer.
- For all cases, send NAK's of missing frames if their retransmission timers are not yet set or expired.

3. Calculation of Session Start-up Time

For the session initiation protocol, session start-up time is defined as the time between the initiation of the INVITE request and the instant of reception of the ACK at the destination user agent.

So, in this research we try to investigate how session start-up time is affected by using TCP/UDP (both supported by SIP) and moreover, how the addition of an RLP layer affects the session start-up time.

3.1 UDP without RPL

In order SIP to ensure a reliable delivery of the signalling messages, it employs a retransmission mechanism that consists of retransmitting the INVITE message after time Tr and then doubles after each retransmission.

The retransmission timer is:

$$T_{rc} = 2^{i-1} T_r$$

In order to calculate the session start-up time we need to consider:

- p is the probability that a frame is lost
- k number of frames that a UDP packet consists of
- q probability of retransmission, meaning that the transaction failed
- l number of air link frames in the UDP datagram
- τ inter-frame time
- d propagation delay

So, retransmission timer is:

$$Tr = d + (l-1)\tau$$

Probability of retransmission

$$q = 1 - (1-p)^l$$

The normalized delay for the ith UDP datagram:

$$T_N(i)_{UDP} = \frac{1}{1-q^N}\{(1-q)(d+(l-1)\tau) + (1-q)q(d+T_r+(l-1)\tau) + \ldots + (1-q)q^{N-1}((2^{N-1}-1)T_r + d + (l-1)\tau)\}$$

$\Rightarrow$

$$T_N(i)_{UDP} = d + (l-1)\tau - T_r + \frac{(1-q)(1-(2q)^N)}{(1-q^N)(1-2q)} T_r$$

The session start-up time can be now calculated by the sum:

$$T_{udp} = \sum_{i=1}^{N} T_{iUDP}$$

Numerical result:

If we consider inter-frame time τ to be 20ms, average number of frames to be 37 per message; propagation delay 100ms we can then generate the figure (appendix - UDP w/o RLP section) with MATLAB:

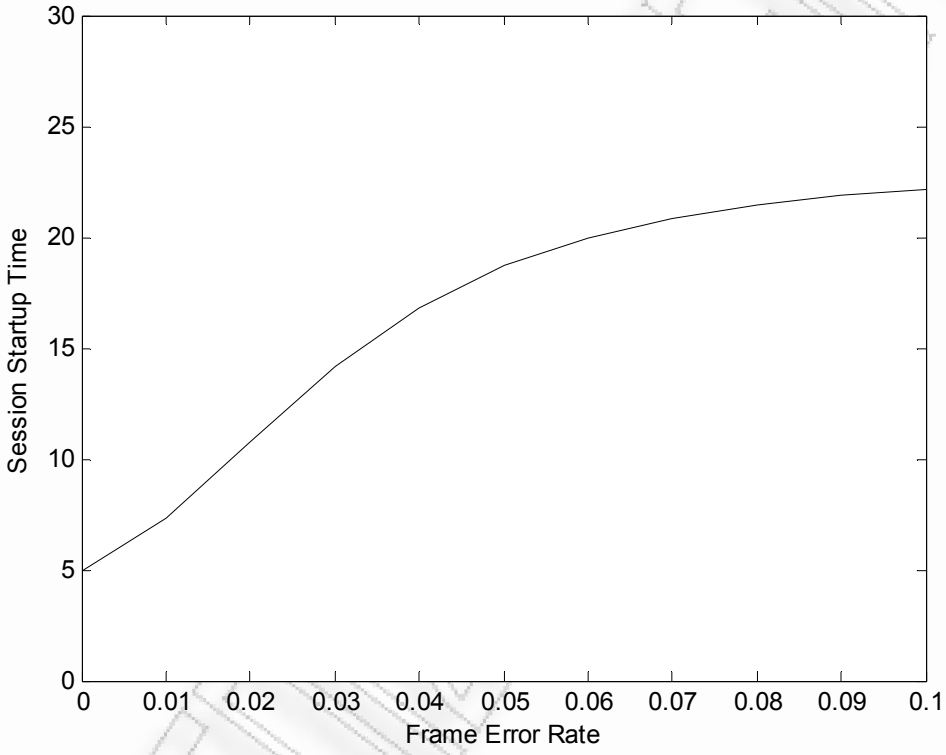

Figure 4.2*: Session start-up time in relevance with Frame error rate (UDP)*

3.2 TCP without RPL

For this one, analysis is the same with the exception of N (number of retransmissions). N is larger for TCP can be higher than for UDP owing to the larger number of messages sent and retransmitted in the event of loss.

Numerical result:

If we consider inter-frame time τ to be 20ms, average number of frames to be 37 per message; propagation delay 100ms we can then generate the figure (appendix - TCP w/o RLP section) with MATLAB:

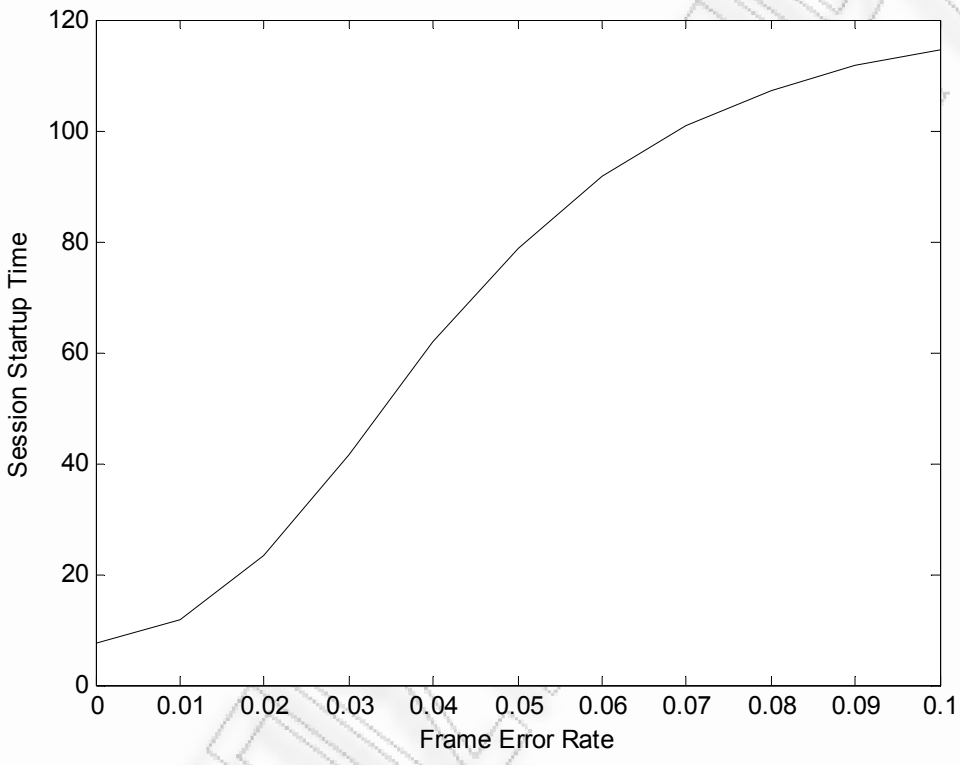

Figure 3.3: *Session start-up time in relevance with Frame error rate (TCP)*

3.3 TCP with RLP

The expressions for calculating the session start-up time over TCP/UDP with RLP can be verified to be:

$$T_N = d(1-p) + \sum_{i=1}^{n}\sum_{j=1}^{i} 2id + 2(j-1)C_{i,j}\tau$$

where, d is the propagation delay in sending the frames, p is the FER, and $C_{i,j}$ refers to the first frame that was received correctly at the destination, which is the i^{th} retransmitted frame at the j^{th} retransmission trail, and $C_{i,j}$ was computed to be

$$C_{i,j} = p(1-p)^2((2-p)p)^{\frac{(i^2-1)}{2}+j-1}$$

4. Remarks

From the above analysis, we can see a direct relationship between FER and average session setup delay. It can be seen that session start-up time for TCP gets much higher than in case of TCP with RLP. Furthermore, the delay in case of UDP is lower when compared with the TCP analysis.

Reliability of the protocol can be achieved through RLP mechanism - retransmission timers; however in case of RLP failure TCP (higher level) is again responsible to provide reliability.

APPENDIX

Chapter 4

M-file: Simulation for session start-up time – UDP / TCP

```
clear;
%Variables
d=100*10^-3; %propagation delay
l=37; %frame number
t=20*10^-3; %inter-frame time

N=6; %no_of retransmissions

% Retransmission timer
i=1;
Tr=d+ ( l-1 )*t +d;
for p=0:0.01:0.1
    q = 1-(1-p)^l;   %propability of
packet lost
    T_udp_whole=0;
    for retrans = 1:1:N
        T_udp(retrans) = d + ( l-1)
*t - Tr + (( (1-q)*(1-
(2*q)^retrans) * Tr) /  ((1-
q^retrans) * (1-2*q)) );
        T_udp_whole = T_udp_whole +
T_udp(retrans);
    end
    Tudp(i)=T_udp_whole;
    p1(i)=p;
    i=i+1;
end

plot(p1,Tudp, '-k')
xlabel('\fontsize{10}Frame Error
Rate')
ylabel('\fontsize{10}Session Startup
Time')
axis([0,0.1, 0,30])
```

p is the probability that a frame is lost

k number of frames that a UDP packet consists of

q probability of retransmission

l number of air link frames in the UDP datagram

τ inter-frame time

d propagation delay

$$q = 1-(1-p)^l$$

$$T_N(i)_{UDP} = d + (l-1)\tau - T_r + \frac{(1-q)(1-(2q)^N)}{(1-q^N)(1-2q)} T_r$$

$$T_{udp} = \sum_{i=1}^{N} T_{iUDP}$$

M-file: Simulation for session start-up time – UDP / TCP with RLP

```
clear;
%Variables
d=100*10^-3; %propagation delay
l=37; %frame number
t=20*10^-3; %inter-frame time

%N=6; %no_of retransmissions
imax=6

counter=1;
for p=0:0.01:0.1
    for i=0:imax
        for j=1:i
            a=((i^2-i)/2)+j-1;
            b=((2-p)*p);
            c=p*((1-p)^2);
            cij=c*(b^a);
            sum=2*i*d+2*(j-1)*cij
        end
    end
    t(counter)=d*(1-p)+sum;
    p1(counter)=p;
    counter=counter+1;
end

plot(p1,t, '-k')
xlabel('\fontsize{10}Frame Error
Rate')
ylabel('\fontsize{10}Session Startup
Time')
```

$$T_N = d(1-p) + \sum_{i=1}^{n}\sum_{j=1}^{i} 2id + 2(j-1)C_{i,j}\tau$$

$$C_{i,j} = p(1-p)^2((2-p)p)^{\frac{(i^2-1)}{2}+j-1}$$

REFERENCES

[1] S. Donovan. "RFC 2976: The SIP INFO Method", October 2000

[2] M. Handley, H. Schultzrinne, E. Scooler, and J. Rosenberg. RFC 2543: "SIP: Session Initiation Protocol", March 1999.

[3] Niemi. "Authentication of SIP calls. In Helsinki University of Technology *Seminar on Network Security"*, December 2000.

[4] Roach. "SIP-Specific Event Notification". Internet Draft, Internet Engineering Task Force, July 2001. Work in progress

[5] H. Schultzrinne. "Why SIP?", April 2001. "SIP Services and Applications (Marcus Evans)",

[6] Wikipedia, "http://en.wikipedia.org/wiki/Session_Initiation_Protocol", access on 02/08-2009

[7] R. Sparks. "SIP Call Control – Transfer", [Referenced 19.02.2002], Internet Engineering Task Force, July 2001. Work in progress.

[8] Henning G. Schulzrinne and Jonathan D. Rosenberg, "The Session Initiation Protocol: Providing Advanced Telephony Services Across theInternet"

[9] M. Handley, H. Schulzrinne, E. Schooler, andJ. Rosenberg, "SIP: session initiation protocol,"Internet Draft, IETF, Sept. 1998.

[10] H. Schulzrinne and J. Rosenberg "A comparison of SIP and H.323 for Internet telephony", Cambridge, U.K., July 1998.

[11] J. Rosenberg and H. Schulzrinne. "Reliability of Provisional Responses in SIP" Internet Draft

[12] Camarillo Gonzalo, Sip demystified, New York: MCGraw, 2002.

[13] Jan Janak, Sip introduction, 2003

[14] A. Johnston, S. Donovan, R. Sparks, C. Cunningham, and K. Summers, Session initiation protocol (sip) basic call flow examples, RFC3665, http://www.ietf.org/rfc/rfc3665.txt, Desceber 2003.

[15] Alan B. Johnston, Sip: understanding the session initiation protocol, Artech House, 2001.

[16] Jiangbo Yin, "Session Initiation Protocol Benchmark Suite", Computer Engineering Delft University of Technology,

[17] Aki Niemi, "Authentication, Authorization and Accounting in Session Initiation Protocol Networks", March 7, 2002

[18] J. Lennox, H. Schulzrinne, and J. Rosenberg. "RFC 3050: Common Gateway Interface for SIP", January 2001.

[19] Tuomas Pikivi, "Implementation of event subsystem based on session initiation protocol", May 2004

[20] Ye Tian, Kai Xu, Nirwan Ansari, "TCP in wireless environments: problems and solutions", IEEE Radio communications, march 2005

[21] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, and R.H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," Proceedings of the ACM SIGCOMM '96, Aug. 1996, pp. 256–269.

[22] R. Ludwig and R.H. Katz, "The Eifel algorithm: making TCP robust against spurious retransmissions," Computer Communications Review, vol. 30, no. 1, Jan. 2000, pp. 30–36.

[23] W. Stevens, "TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms," Internet Request For Comments 2001, Jan. 1997.

[24] F. Lefevre and G. Vivier, "Understanding TCP's Behavior over Wireless Links," Proc. Commun. and Vehic. Tech., 2000 SCVT-200, 2000, pp. 123–30.

[25] Xylomenos, George and Polyzos, George C. (2001) "*TCP Performance Issues Over Wireless Links",* IEEE Communications Magazine.

[26] Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan and Randy H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", Computer Science Division, Department of EECS, University of California at Berkeley

[27] S. Floyd and T. Henderson, "The New Reno Modification to TCP's Fast Recovery Algorithm," RFC 2582, Apr. 1999

[28] E. Ayanoglu, S. Paul, T. F. LaPorta, K. K. Sabnani, and R. D. Gitlin. AIRMAIL: A Link-Layer Protocol for Wireless Networks. ACM ACM/Baltzer Wireless Networks Journal, 1:47–60, February 1995

[29] R. Yavatkar and N. Bhagwat. “Improving End-to-End Performance of TCP over Mobile Internetworks”, In Mobile 94 Workshop on Mobile Computing Systems and Applications, December 1994.

[30] S. Vangala, M. A. Labrador, "Performance of TCP over Wireless Networks with the Snoop Protocol," lcn, pp.0600, 27th Annual IEEE International Conference on Local Computer Networks (LCN'02), 2002

[31] K. Brown and S. Singh, “M-TCP: TCP for mobile celullar networks,” Computer Communications Review, vol. 27, no. 5, Oct. 1997, pp. 19–43.

[32] Suwat Pattaramalai, Valentine A. Aalo, George P. Efthymoglou, “Evaluation of Call Performance in Cellular Networks with Generalized Cell Dwell Time and Call Holding Time Distributions in the Presence of Channel Fading”, Sept. 2006

[33] Chockalingam, A.; Gang Bao, “Performance of TCP/RLP protocol stack on correlated fading DS-CDMA wireless links”, vehicular Technology, IEEE Transactions on Volume 49, Issue 1, Jan 2000 Page(s):28 - 33

[34] Marvin K. Simon, Mohamed-Slim Alouini “Digital Communication over Fading Channels”, 2005

[35] M. Schwartz, “Mobile Wireless Communications.” Cambridge Univ. Press, 2005.

[36] J. Lai and N. B. Mandayam, “Minimum duration outages in Rayleigh fading channels,” IEEE Trans. Commun., vol. 49, no.10, pp. 1755-1761, Oct. 2001.

[37] Bernard Sklar (July 1997). "Rayleigh Fading Channels in Mobile Digital Communication Systems Part I: Characterization". IEEE Communications Magazine 35 (7): 90–100.

[38] H.Suzuki, A statistical model for urban multipath propagation,” IEEE Trans. Commun, July 1977

[39] P. Yegani and C. McGlilem, “A statistical model for the factory radio channel,” IEETrans. Commun., , October 1991

[40] J. G. Proakis, Digital Communications, 3rd ed. New York, NY: McGraw-Hill, 1995.

[41] Y. Zhang, and M. Fujise, “Performance analysis of wireless networks over Rayleigh fading channel,” IEEE Trans. Veh. Technol., vol. 55, no. 5, pp. 1621-1632, Sep. 2006

[42] J. Lai and N.B. Mandayam, “Fade margins for minimum duration outages in lognormal shadow fading and Rayleigh fading,” in Proc. 31st Asilomar Conf. Signals, Systems and Computers, Nov. 1997, vol. 1, pp. 609-613.

[43] Y. Zhang, M. Ma, and M. Fujise, “Call completion in wireless networks over lossy links,” IEEE Trans. Veh. Technol., vol. 56, no. 2, pp. 929-941, Mar. 2007.

[44] N. B. Mandayam, P. C Chen, and J. M. Holtzman, “Minimum duration outage for CDMA cellular systems: A level crossing analysis,” Wireless Pers. Commun., vol. 7, no. 2-3, pp. 135-146, Aug. 1998.

[45] G. Bao, "Performance evaluation of TCP/RLP protocol stack overCDMA wireless link," Wireless Networks, pp. 229–237, 1996

[46] A. Chockalingam, M. Zorzi, and R. R. Rao, "Performance of TCP Renoon wireless fading links with memory," in Proc. IEEE ICC'98, vol. 2,GA, June 1998, pp. 595–600.

[47] TIA/EIA/IS-99, Data services option standard for wideband spread spectrum digital cellular system, 1995

[48] Chockalingam, A.; Gang Bao, "Performance of TCP/RLP protocol stack on correlated fading DS-CDMA wireless links", vehicular Technology, IEEE Transactions on Volume 49, Issue 1, Jan 2000 Page(s):28 - 33

[49] Vijay Sundar Rajaram, "Session Initiation Protocol for wireless Channels", Texas A&M University, Dec 2006

[50] J. Harris and M. Airy,"Analytical model for radio link protocol for IS-95 CDMA systems," in IEEE Vehicular Technology Conference Proceedings, 2000, vol. 3, pp. 229–237

[51] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson et al. "SIP:The Session Initiation Protocol," IETF RFC 3261, June 2002

[52] "Radio Link Protocol (RLP)", http://en.wikipedia.org/wiki/Radio_Link_Protocol, accessed July 05, 2009.

[53] H. Schulzrinne, E. Wedlund, "Application-layer mobility using SIP," ACM SIG-MOBILE Computing and Communications Review, vol.4, no. 3, pp. 47–57, July2000.

[54] TIA/EIA/IS-99, Data services option standard for wideband spread spectrum digital cellular system, 1995