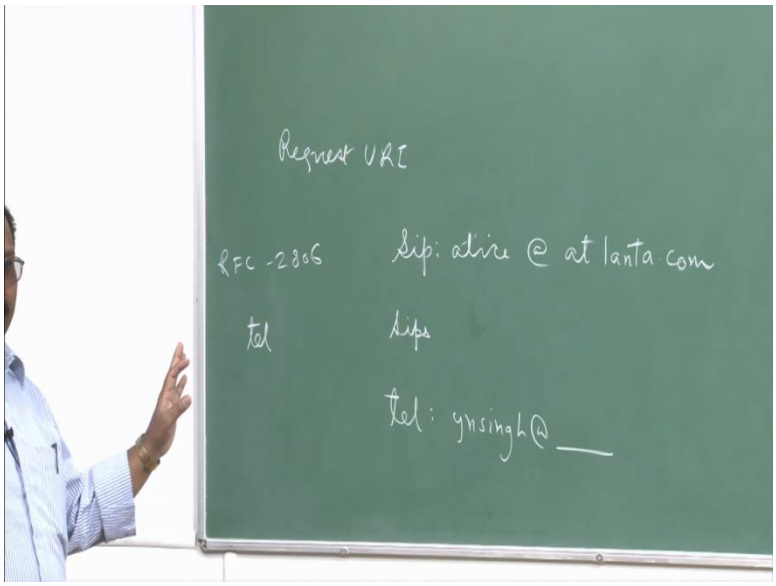
Digital Switching
Prof. Y. N. Singh
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

Lecture – 37

So, we will actually go ahead from where we left yesterday evening; I had actually given you at that time 6 possible methods which have been define in RSE 3 to 6 1, which actually can be used, out of this 2 actually can be send without creation of any session without basically technically creating any dialogue.

So, one of them is option other one is invite. So, these I have not told I am telling it for first time, invite we already aware of… So, invite I can cancel or technically used for setting of the session by is for terminating, register is for registering and options are for querying. The other entity what are the capabilities with that server actually.

(Refer Slide Time: 01:04)



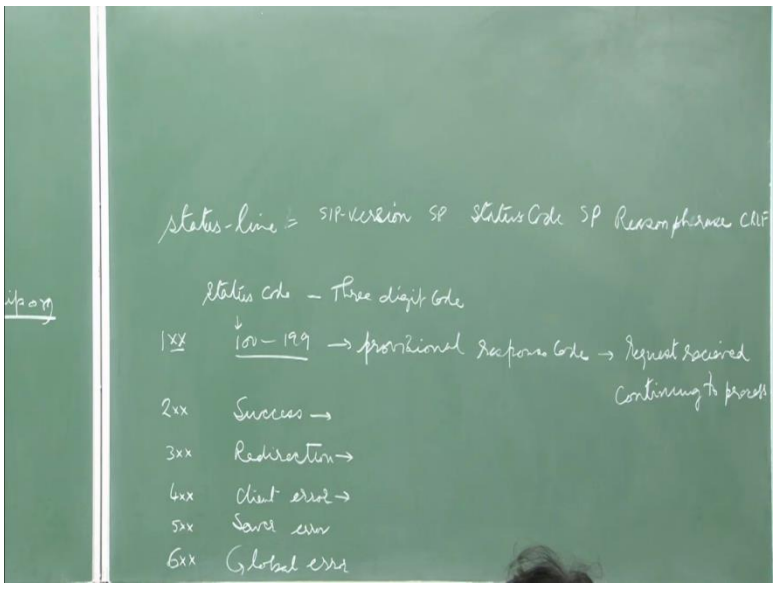
So, the request URI which is formed; this universal resource identifier so far, I have only told about Sip for example: at lanta.com kind of thing. So, the I have only talked about so for this thing, but as far as the RFC is concerned. When you make an implementation you need not only use this, you can actually you Sip, you can sip, even other kind of URI is standards can be followed and you can even implement those URI's. URI's also can be used in the system that is permitted RFC as for that does not say anything is specifically,

only the problem is if you implement some other kind of URI scheme. Then it is the problem of the Sip entity, it has to figure out what has to be done with that.

Whether it has to be mapped on to another Sip URI or it has to be map to a Non-Sip URI another 1, but that will be head ache of the entity. As a sip will not bother, but we require. So, this is what we called the scheme. So, sip a sort to schemes you can even use other a schemes URI structure has to be there has to be a domain name from domain name you will find out where the request actually has to go. So, that is the only important thing. So, domain name is important, but scheme actually can change so in fact, for telephone there has been already a scheme which was built.

So, which can be use alternately this is as per RFC 2806 is scheme is known as tell. So, you have might actually seen this thing at lot of places, is a telephone thing are may be whatever is a domain name kind of concept which can be there. So, tell URI is also there. So, this also can be used, this is one example, they many others each people have built over the time, but the translation from this Non-Sip URI to Sip URI or Non-sip to Non-Sip is a responsibility of the entity, which is implementing the protocol, but you can use even not Sip URI's that is 1 important thing, usually because when I am been telling people will actually conceive an idea, a notion that always Sip URI has to be used, no it not delay you can use something else also.

(Refer Slide Time: 04:11)



So now, coming to this status line because last time it was the request thing, which was actually talked about. So, either there was a request line oblique a status line. So, request line is structure I have defined last time. So, status line so now mostly it is I am just discussing is about the rules, which have to be followed. So, status line typically will be written as this should be equal to. So, this should contain a Sip version a single is space character, a status code another single is space character and then of course, reason phrase and in the end you have to have CRFL carriage written line field character which has to be present. So, that is BNF definition for the status line. So, request line I have already told. So, this is basically human readable entity this will not be interpreted by the machine.

This will be interpreted by the machine, this you are going to write. So, that some entity might be actually using a older version of RFC Sip, Sip v1. So, in that case you will behave accordingly or you will reject if it is not as per your this thing. So, you have to specify. When sip v3 for example, he will come. So, v2 and v3 machines will co-exists. So, depending on what kind of response is coming are what kind of request is coming you have to behave in that way. So, usually when you are going to implement a machine or implement software, you will ensure that you will implementation can handle both kind of requests.

So, you do not have to implemental only 1, but you have to implement both kinds of RFC's. So, v2 as well as v1 usually that is commercial entities will do that. Open sipped I think that is provide both the options, if you are going to use open Sip library. So, in fact, the complete implementation you can find out as open Sip dot r. So, this where you can get the library and then this has the complete implementation of this. So, tomorrow if you want to actually buildup in a application you can use the code from here or you can contribute back. Now, a status code here is 3 digit codes, this I have already mentioned earlier. And only the first digit will specify there 3 digits, this will specify the classification. So, 101 or 1002 199 series will correspond to what we call a provisional response of code.

You can also write it as 1xx, xx represents going from 00 to 99, is specification does not talk about what will be for 00 will be use what will be for 01, 02 something, but this is going to be provision, but depending on usage they essentially become now. So, this there in other I have see not in this 1. So, this typically for example is, provisional means

that request is received and we are continuing to process the request. I think this I have mentioned earlier sometime back also, similarly you will have 2xx category, this corresponds to success. So, action has been or method has been successfully received or action has been successfully received it is understood and it is accept, that is what it means actually. So, 200 is for example, 200 they can be other variant also, but most of them will signify it is a success.

Then you will have 3xx code which in come this is usually for redirection. So, this says further action need to be taken in order complete the request that is what the meaning is. 4xx is client error; client is the 1 who has send the request, response is send by server. So, something which has come from the client is meal form there is an error actually in that. So, I am not able to parse and understand what was the request that is the meaning of this 4xx. So, technical this request will contain either words syntax are cannot be fulfilled at this server that is what it means see for example, you are sending a message to registrar and you send a invite to registrar a registrar would not understand it can only understand a registrar request and the response will be success when the registration happens, but that is the only thing which can be done nothing else. But so in that case it tells I cannot fulfill this request. So, whatever you sending means anyway the thing.

So, that is what the 4 xx is reserved for then you will have

I lets wove a example Sir, 4 as 6 as 4 8 words are same Sir from the server side, that is from the side bob side. The bob should be the server then Sir.

What was that 4 8 4?

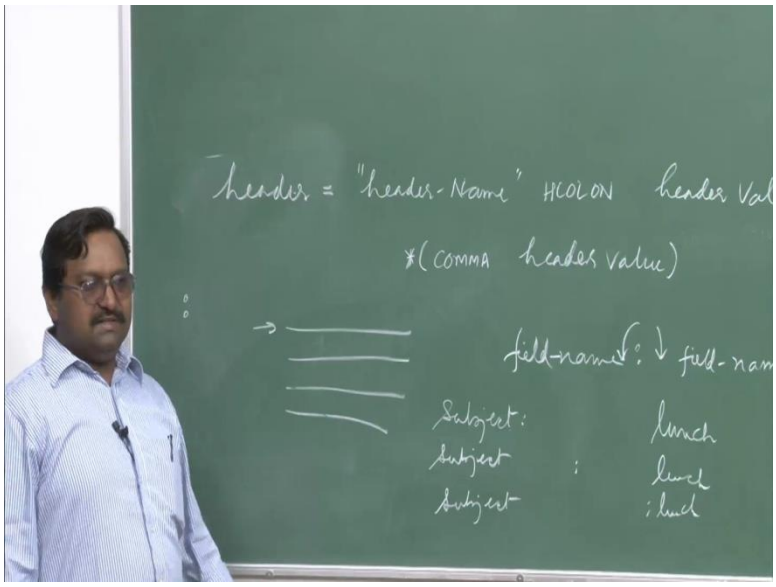
4 8 6 are busy and 4 8 8 was not acceptable.

So, it is actually assuming the client is an error, because I am busy he should be aware of that; maybe that is not the server error. Server can still fulfill the request, but as of now I am not possible because it is coming at the wrong time mostly that is the meaning. So, the 4 series was given in that. So, there is 5 in 6 also, let me just come to that I have not noted it down. I have not noted down 5xx in 6xx, but. So, there is a 5xx also I think that is a server error. If I remember correctly and there is 6xx , I think local global feel area I think I have written it is something like it must be there on the some other sheet same

thing as repeated actually. So, that will be the status code now coming to the header fields.

So, for we have not discussed header fields this only a sample actually we were shown. So, moving to from request line to header fields. So, header fields will contain typically a structure which is going to be a header name and header value field name and filed value technically.

(Refer Slide Time: 12:06)



So, a structure usually will be header, this will be usually consists of some header Name. So, that is a grammar actually header value. So, a star actually shows that there can be a variable number of parameters a variable number of values which can exist field value there will be comma. So, this can repeat h colon h colon is this 2 dots a colon actually h colon is written in the grammar. So, I have just copied that actually its colon. So, header will usually consist of multiple lines each line usually will contain one field.

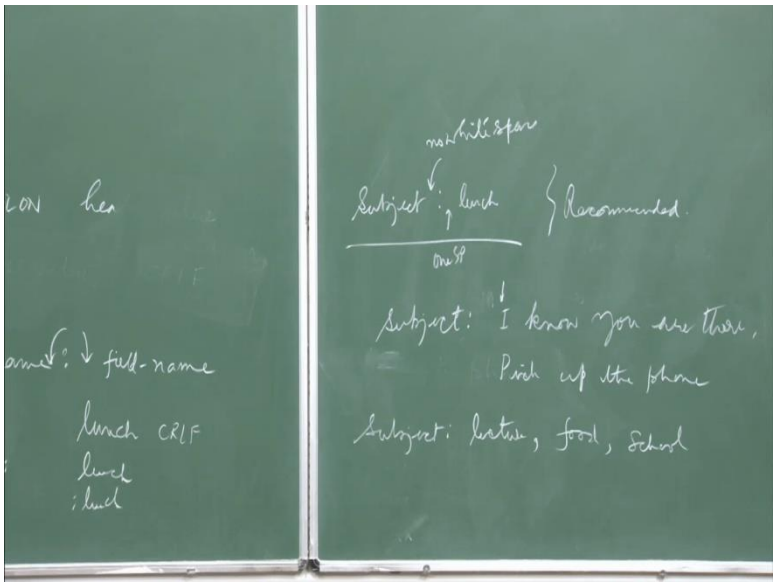
we can actually have multi line fields also are possible and every header filed will you start from character number 1 in a new line. If in the beginning there is in a space then it is not in new field that is nothing but the previous header field in the real line has been broken. So, I can is split one single long line into multiple lines, but I have to insure in the beginning the first character should be either at least 1 witty space or 1 tape. So, that is the condition I will come this particular example. So, usually header field format is going to be field name.

So, I had 1 to actually do away they the space I am going to put a dash in between actually to be more clear. And this will be usually a field name and you can have orbiter amount of witty space here, before and after the colon that is permitted. So, this is not illegal. So, in case if you write for example, some field which is there is usually witty spaces are not there you will note that way where I am putting, I am actually putting a hyphen for joining the words and make it single word I am putting a space, because the space is taken as a separator a delimiter actually for the various fields. So, again this a structure, but that is fine. So, subject I am not writing capital. In fact, it is case insensitive capital in small does not matter for example, in Email you might have seen that you send an Email to me with full capitals.

You send it to me like this, will both the mails reach to me or not, they will always reach to me is case is insensitive actually, case does not matter. The same is to even for Sip URI; basically true for any header field expect under special cases which are is specified or there is a string under the quotations. If there is the string under the quotations than it is case sensitive thing entity, case does matter in that case they are not same. So, I think this again as been taken from… So, if a parameters under invited codes that whatever is there under the codes has to be case sensitive. So, in fact, that is also true for a SDTPURL's.

So, subject I can actually put no space, larger number of space and I can write say lunch. I am taking same example given in the RFC or I can now put. So, this are all valid whatever I am writing, I can write all 3 all valid actually headers. As for as the syntax is can subject is I think is not the header field there, I have not verified that, but the what is recommended practice is. So, when you write a software or code you have to ensure that you follow a recommended practice, but you should be willing to accept, assume that the other guy who going to send can send you a mall form thing he may now be following the recommendation, but you are suppose to follow. So, you should not be able arbitrary.

(Refer Slide Time: 17:58)



So, other guy who is sending the Message may not follow that is fine, you will be you should be able to accept all this, but when you send it is recommended you are always going to put no white space here no space is character and there will be exactly 1 space character and then whatever is a value. So, that is the recommended format and header fields can be split over multiple lines they can be extended that is permitted because every new header field will be starting from first character in a new line. So, that is essentially q is being used for the encoding purpose. So, that you can actually spilt a same header field over multiple lines.

So, again taking an example, when this is being constructed from example, you write…So, I am using a standard there is no space, white space 1 white space I can put up a string, but if I write something like this. So, I think this is very similar sentence which there is in the text. Actually this will whole thing will be 1 value this not under invited codes, but it is 1 value because values are separated by comma, I can have multiple values for 1 field name, but then they will be separate by comma there is this only 1 comma. So, this 1 field, this another field in this case.

So, another value actually field value field name is subject. So, before this comma this is the field, this is this is the field a special meaning can be taken off if I put inverted codes then it is a 1 single value I can have an option to split into 2 lines. I can put like this, now if I put this invalid actually because I am starting from the first character in the next line.

This has to be then a header actually field I cannot do that. So, this is basically an identification mechanism. So, that on the receiver side your software can parse it easy can understand what is the Message? So, there as to be at least 1 space or 1 tab or more than that any number.

So, usually the practice is wherever your first character starts you will that is the recommended tricks. It is actually becomes human readable that is the reason actually why you do it is this way. So, this is in the earlier version both are same. So, I can have multi line header fields. So, we have done this is spacing is we have done multi line the sequencing of headers is that important.

Sir, this here in the field.

In the end CRLF will be there in end.

In the format.

Because in the next header as to start only from the new lines. So, CRLF has to be there this only for 1 header. I am actually you are right there has to be CRLF in the that is true there has to be there has to be a formally if you put it.

Accept in inside that what does not that will signify.

Where?

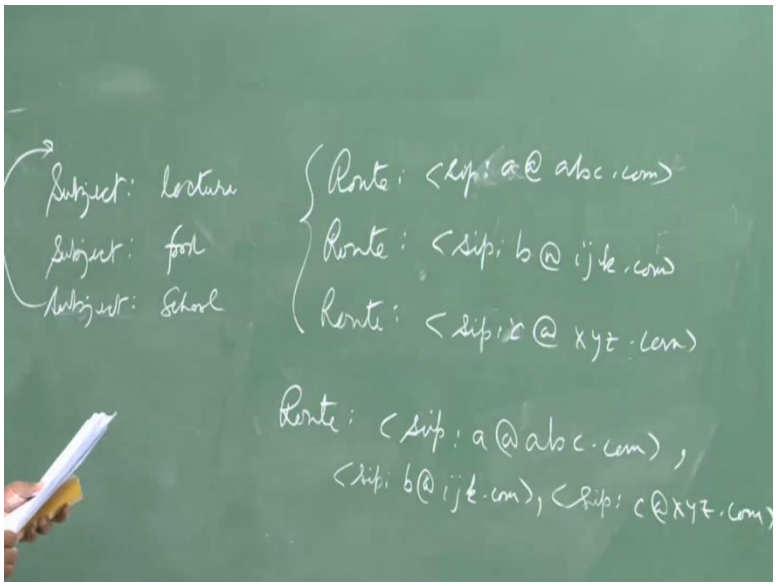
After that aspects comma.

This 1?

And the header value is.

The header value is this, this is the field you can have multiple of them. I can actually put for example, subject: you can say lecture. So, that is the 1 value you can put a comma then you put you can say food that is another value. You can put say if School that is another value. So, 3 values for this particular thing. In fact, this is equal to interestingly this is equal to I will come to that I was coming to the ordering.

(Refer Slide Time: 22:51)



But let me give me this a especial ordering this is nothing but equal to actually writing this subject lecture. So, that is actually equal to 3 header fields, header names or header field names can actually repeat that is permitted. And the order does matter, order does matter because, if I actually for example move this thing on the top then the structure this and this are not same. So, whatever is the sequence of the values here, in same order they should appear. That is important see for example this, the problem is for example.

I am deciding a root field, root say the message as to go from me to you first and then from there to another guy from there to another guy. So, I am defining even the sequence by this message should go. If I change the sequence I can either the define a root header for example, I can say root colon and I define a sip URI for sip: a. abc.com; I can define another root b@ Ijk.com similarly I define third 1. So, this actually defines the sequence I cannot disturb the sequence.

I can write the third entry as. So, this how the message will be forwarded it. So, it as to go 1 by 1, it as to first of all go to this from there is will be send to this guy, from there it will be send to this. I can setup a multicast conference actually through this mechanism.

It is usually grammar you are write.

Yeah, I am just giving what is the structure.

What are the rules?

And just like syntax.

Yeah syntax is specification.

So, I can have see @ xyz the equivalent of this as per this. So, as you propagating you can keep on relating or removing. So, remember why a header field has to follow a sequence, you cannot simply swap, but I can also put them in comma separated form that is permitted. A comma separates a field value not the semicolon I will be also I can also use semicolon; a field value can also have parameters. So, a field value semicolon parameter name is equal to parameter value and any number of parameters can be there.

So, that is also there another extension in this. So, root for example, here will be this is equivalent now, I have to put a comma remember, they multiple values. I say I am now putting it multi lines. So, I should not distract from the first character. So, I can start somewhere here, I do not want to the multi line I can write in this line its self. And of course, you have to use in all URI's you have to use escaping this I think I have already mentioned earlier.

So, all is special characters have to be escape in that case, if I know which are characters have to be. So, the relative what here does matter, but relative water of fields actually does not matter unless the names are same, when the names are same within them order does matter they can be placed anywhere. So, header field 2 you can put in the end for example, from you can always put first, only first thing as to be request line and after that whatever is the header thing can be in any sequence, but if the names are same within those things the order as to be maintain because that is important.

They only even can be combining into a 1 single entry; this combination cannot be done only for now there is a recommendation again is not a rule. So, when you are writing software's. So, other guy can do anything.

Sir, whose sharing this book this constant will take.

This is from the source, when you are sending invite that time you can decide.

The clients will.

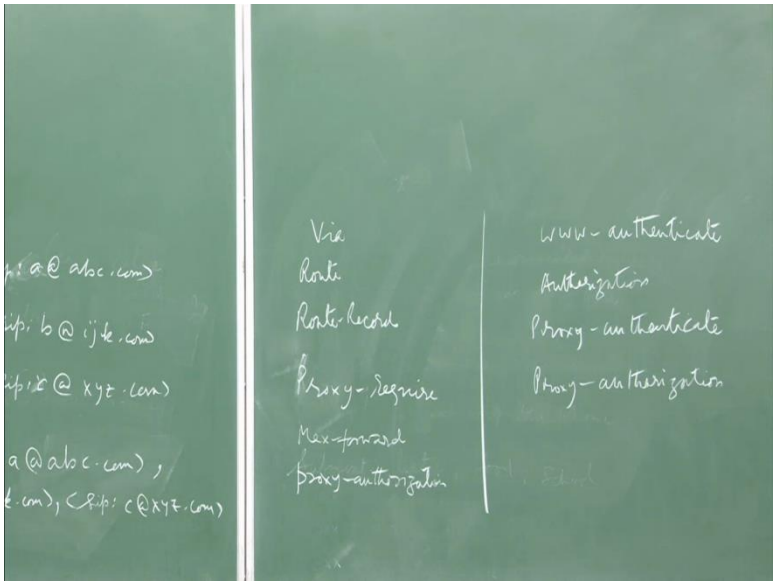
It is like source routing.

See, when this is root this are users I am not worried about proxies, I want to signal you. So, I will tell my proxies to find a root you and then pass the signal on to you after proper authentication, they I say after you should now send a message to somebody else. So, I will pass on this old header to you. So, it will go to you first and then from that you go to you, then from air it will go to you, then to the final destination which is 2. So, all 3 guys intermediate will be from this are not proxies this are clients.

Then this is input for example, conference.

Yes, a kind of over late multi casting when it is to be created, see I do not want to communicate to everybody, I want to just same message as to broad cast to 5 guys. So, it is has to go in this particular sequence.

(Refer Slide Time: 29:10)



So, recommendation is that, this particular fields root, record, proxy-require see does not matter whether it is small letter capital letter I have told already it is case insensitive. I can write all capitals, I can write all smalls is permitted in this case. So, 1 capital is a small capital a small that is also going to be same, not necessary, say I have as the mistake earlier, because that is say it was written and that mixed more legivalent beautiful that is only thing. So, that is a recommendatory practice, but need not be. So, when you receive something you should be, we should prepare yourself for the worst. When you are sending follow the recommendation, then max forward and proxy authorization. These fields are recommended to be always on the top of the message,

why they have to kept on the top of the message why do not want to them to get the bottom what is a benefit?

Computation will be this.

No, computation list will be the same only thing it will take less time to parse, become the message may when the message receive at a proxy, this are going to receive all this header fields first. So, parsing can finish much faster for the proxy. So, when I am sending a message to you all, the header fields which I am expecting you to actually parse require should be always kept on the top of the header, that way you will be able to parse it much faster. You go every header field which comes it starts separate computation process after parsing. So, this parsing there computation starts much earlier, when it comes at the later say it will be starting much later. So, I can save time in setting above the calls or managing the calls, again this is a recommendation not mandatory thing.

Always time to request like first of all.

Now, first thing will you always request in a status line that you cannot change, after there headers fields. So, I am only talking about header fields now, not the requestness status line. And then, there is in the end then there is a CRLF and then a message body. So, message body also can be of various types, it can be a binary object only the content length will be specified. This I have already told multiple field values can be combined into 1 single field well you in this way by using comma separated field values actually list. This combination combining cannot be done for certain kind of errors; you can do it for everything else except, there are 4 cases we have this cannot be done. Where you cannot do it for this particular header field name, you cannot do it for authorization, you cannot do it for proxy authenticate. And you cannot do if for proxy authorization.

So, why you think it is not permitted in this case?

It is order.

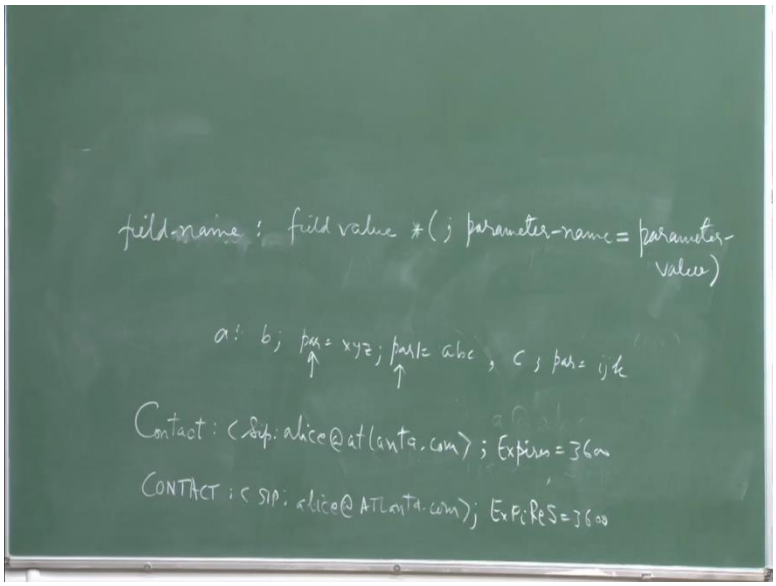
You simply, I think drop the header fields after looking at it you do not have to reframe them in this case. In earlier cases, you have to reframe for example, this when you reach to this particular root this guy you have already reached to this will be dropped out of

this line only 2 will be left out. You are reframing for this, reframing is not permit. I think is probably because the hash is also must be going to check for the validity of whatever data you are sending in this. And thus my most likely guess I have not figure out the information, because everything has to do either we are authentication and authorization. So, whenever you are sending some keyword or something, it is a good idea to actually always also send hash, will that, hash will be broken the movement you put that thing and hash should be part of that history.

Sir, were saying about the comma and semicolon Sir you are.

Yeah, that is now I am coming. So, now, field names also can have parameters, field values. So, I think one of the example was tag which we did or branch. In case of your invite and response thing now, that is addition, now remember the field values are still separated by commas this I found straightly strange because usually whatever is there, as 1 single entities separated by a semicolon. So, within that I can separate the subfields by commas, that is usually is the practice normally English language, but here it is then the other way around, Practice normally English language, but here it is then the other way around. I do not know the reason why this is done this way, but that is it.

(Refer Slide Time: 34:52)



So, a field value by definition, I will write field name. So, header field will look something like this; this is the header field, it wills continuing a field name: field value any number a semicolon parameter name is equal to parameter value. And any number of parameters can be attached, that is why the star is there and this forms 1 value actually, is there is another value then it has to be comma and then the next chain value start.

So, please repeat sir.

See, this whole thing is technically 1 parameter, 1 value, 1 field value. it is only 1 field value, but the field value has been pended with parameters. When the second field value has to be done there has to be comma and the next field value will come which can also be of the same form. So, branch and tag thing which we had earlier was nothing but, parameters. They always took the form semicolon whatever was the parameter name is equal to parameter value.

And 1 field may have more than 1 parameter values?

They can have more than 1 parameter value, it is permitted. You can have more than that is why a star is put, but the same parameter name cannot be used more than once. A parameter name cannot exist for more than once, it cannot be repeated, repetition is not permitted.

In ensure in a…

Within this field value, I am putting parameter 1 is equal to parameter value, then I cannot put semi colon parameter 1 is equal to another parameter value, now, because those both are parameter 1. So what I am saying is, if I write a colon b, I say pair is equal to xyz; I can put semicolon, I cannot now this is invalid, because this and this are same it is invalid.

Get separated by a sir comma then we get.

I can put pair 1 then it is valid, yes if I put if this thing comma, I am now going to have another value, which can be c and for which now I can use at the same parameter. This is a valid formation. So, basically you can send this are basically like parameters, the way tag is for example, random is string which is generated for every 2 n from fields there is a branch which is there in the via header field actually which has put. So, those are all parameters actually. So, case sensitivity I think one of the good example is contact I can write now this is not I am not going to put another field, I am putting a parameters I have to put a semicolon; this both are same or not same they are going to be same because all things are case insensitive.

Last examples are we show that contact header fields and that we were resolving to IP address dissolving to it IP address not.

I can put domain name also you see, 33. at toyanta.com

Yes, pc 33 that there will be because this will be s IP

Right, the currently it is given like this; I can put pc 33 at lanta.com also fine, I am giving an example.

PI ignore.

Contact will expire after this first time.

Is in where other time all speaking the reduce the same.

You are telling you have to connect to me, but I will you probably moving out. So, with after this time I will not develop there, unless I achieve the send you another invite or another update, you have to keep on updating.

Did you add this may?

The source, see when you are going to send a message somebody you will add your contact.

Will contact this expand t.

Expire also you will add, you are here say for example, in next 60 minutes after that certainly you will not be here. So, you kind of give a least t for this IP address you keep on correcting only for this much period. After this if I do not send you another least kind of thing or another inform update, yeah simply assume, I am that I am not there on its address.

Would you the distinct?

Is like, if you are going to your internet cafe.

You want to login, you would like to always set key, whether I work, do not work after 15 minutes. This system should logout, I should be logged out of it actually enormate should be erased, but if you are going to work for another 15 minutes, before the 15 minutes expire you will send another update, you will again do the authentication. So, 1 authentication is valid only for this much period. So, you have to again do a another authentication for increase your validity period, I think it for that purpose logical choice is this away should I have been designed.

Now, this header fields which are there, there again are catic can be categorized. Some header fields will be existing only request, some will be in response is some will be in both. So, this are categorizes request header fields and response header fields correspondingly. Now, you are writing in Engine software basically or a server you end up in getting a message, I request actually has come to you. And it gets a header fields which corresponds to a response.

What you will do with that, should you ignore the whole message, should you ignore only that header field should you start making noise. The rule is very simple in internet if you do not understand anything simply ignore it. So, ignore that header field whatever you can understand do all the parsing all actions have to be taken as per that that, is a

rule that is a thumb rule followed everywhere in the all almost all internet protocol calls. If you understand it is fine if you do not forget, now, there another interesting thing.

So, for I have given you all kind of very long names of header fields, sip also does have something think called short forms of this, short form of all header field names does exist and you can use that. So, when you write a software your software should actually understand both forms. A proxy mid way can actually change your header field name from longer to shorter form or shorter to longer form.

If they are multiple times a field is existing some of them can be in short and some of them can be long that is also permitted. So, both are valid entries they, but the both will mean the same thing. And why are these short form concepts was actually invented? No, there is a maximum transmission unit, maximum size of UDP packet. You cannot transmit larger than a size, if your message itself becomes too large then what you will do. If you say, you cannot you things whatever you are trying to transmit cannot be packet into 1 single UDP. You start compressing start is in shorter form that is a immediate thing, if you are using a t sip transport this is steamily transport.

So, in that case size can be anything, but in UDP it cannot be anything. So, it is basically for that compression purpose to converse on the space. So, one is says even reliable other 1 is also even reliable, what is short form according kind of them. Now, coming to there is another important thing that UDP it is fine, when you will get the message your first line will always be request line or a status line, but if you are not using UDP, you are using TCP. I send an earlier message, after that I was silent; I send some wide a space is of CRLFS. And then I send the next message, then your first line will not be a request line after the silence on the string thing.

So, again in your software irrespective whether it is UDP transport being TCP, if you receive a message all CRLFS carry ejected line feeds, which are happening before request line has to be simply discarded. This problem will not happen, if you are using UDP transport. So, but this has to be explicitly programmed otherwise this will create confusion, because CRLF is end of header.

And same thing, which again.

What happens when you are writing a streamed transport between 2 entities? So, you keep on sending bites and bites will arrive, is not 1 packet or message is being encapsulated into have UDP packet when send. So, you have sent a message after that silent, you might send lot of CRLFS or whatever it is and then suddenly you start you got the first request line. So, previous message end finished, and then you write a request line in between there can be wide spices. This have to be simply discarded actually by design if it is UDP packet will come, you will not see any wide a space things will start with request line, but not with the TCP transport. So, with this streamed transport this particular issue will come. So, that also has to be reserved.

But after those are wide space is in TCP, it changes condition or introduced still get the request line, Sir, you start up from address field.

You always get the first thing is always request line then header fields.

Question.

And then CRLF and then the body.

And the…

Body length will be always, there is always content lengths will always the part of a header. You cannot have a header without content length field; it will at least be 0.

After Sir, 1 message is after that, 1 message is pass then in teach because it is connection are entered we get lot of witty spaces URLF Sir, than again we get the message. So, again that message will have request line Sir.

First request line will be their always.

So.

Every message will have a request or response.

So, the TCP only I.

Request or a states.

The TCP only I should taken the witty spaces.

Yes see, but when your building up your transaction user, remember you are a building up a transaction client or transaction server, client transaction or server transaction which are going to use a transport. So, transport is at the bottom as a transaction user are a basically client or server transaction. You do not bother about the transport; transport is somebody else is addict. So, you have to remove all that thing in your client transaction or server transaction actually, which are they actually, now receive the message parse it and the invoke the code in transaction user to correspondingly to do whatever is a required, but it remains there to maintain the state of the transaction.

You mentioned that, do you have the coding of this messages I mean whenever speaking are voice land spaces are this continue.

No, that is in media this is purely signaling linked setting up of the call, was the call has been set up between 2 end points in this section for example, when they talk to each other, then I sending over RPT real time protocol and real time stream a protocol. I can use and I can send mp3 for example, in both directions for are due. So, mp3 encoding because I do not know want to send the packets unnecessarily, I can simply keep a if you are a speaking then only I will n code and send the packets.

So, otherwise you want only heart beats will give will keep on sending that the connection is a live media connection, then here the play back will happen. Where the media packets are is their otherwise, just you make sure you do not get annoyed some noise can be played. So, as if you feel is a connection is still alive. So, wacky tacky is you do not see other you have a something coming or nothing coming that actually usually is annoying.

Telephone, we do not see we control sly actually, see something coming from the other side. Even if there is silence you will get some hum. So, this probably can be locally created to converse all the bandwidth. A skype, I do not know how does to do, but a skype does have a silence detection and it cuts of the voice where it is not required, but even if there is silence and if you doing differential encoding in the voice, you still conserve on the bandwidth because the differential changes very small.

So, your voltage levels when they are varying in time. So, this differential changes what are sends. So, if the noise is like this, you differential changes very small. You have to send a very small amount number of bits; every sample interval actually. So, anywhere

consuming mush less man within in this. So, there is a live compression, differential, encoding, silence, detection everything is their actually at this. This 1 more last thing this in STTP for example, you can send a body there is a message CRLF and there is the message body. Where you put the payload part, this payload STTP allow the in chunked means; you can actually define multiple body's with each one of them having their own content length type, connect length basically specification.

So, larger message body can be broken into smaller parts and can be transported through STTP. Now, this is not permitted in sip, all that is going to use almost same kind of mine pipes, same content length, content type, content encoding either STTP, but chunked body transfer is not permitted with sip. So, I think that was kind of an introduction to sip we have not this basically giving all the syntax, how things will be done and some a issues you have try to understand. So, with that actually I am now, closing the all lecture series of e 6 to 9. So, next class I will be just taking up all the questions regarding whatever has been thought in the whole semester. So, discussion about whatever questions you have.

So, my request is you please send me all the questions which you want to me discuss before end. So, that I can at least go through the text, even if you do not do it I will try to give to my best whatever inserts I can gave in the class. So, next week is on Thursday actually last class. So, it will be all discussion about this thing, whatever we done so for in the whole semester. And of course, you feed back suggestions, because I think we should be opened and lot of your actually practice engineers I am not 1, people who work in the field, know the actually much more they can also provide their inputs.