TITLE:	**A Comparison of H.323v4 and SIP**
DATE:	**5th January 2000**
SOURCE:	**Nortel Networks**
ABSTRACT:	This contribution compares and contrasts SIP to H.323v4 to help aid operators and vendors in the selection of a single least common denominator control protocol for the ps "domain" or perhaps more appropriately "plane" of UMTS Release 2000. The format anticipates the concerns, in the form of questions, which may arise from 3GPP members.
RECOMMENDATION:	For facilitation of protocol selection.

Note: Every effort has been made to ensure the completeness and accuracy of this document. Any inaccuracies or discrepancies are unintentional and should be brought to the attention of Nortel Networks for clarification/correction.

1 Background

Two standards have emerged for signaling and control of VoIP telephony: ITU-T H.323 and the IETF Session Initiated Protocol (SIP). These protocols, although resulting in the same end-user service (telephony), differ in the approach to providing signaling functions. H.323 is based more on a monolithic bloc derived from H.320 for traditional of the traditional circuit-switched ISDN multimedia, and SIP favors a more lightweight approach based on HTTP. In order to provide telephony services, the UMTS Release 2000 network requires a call control protocol to initiate and manage connection establishment,

1.1 Key Motivators in Network Protocol Reconsideration:

- A single call signaling protocol is desired with distributed functionality across the UMTS Release 2000 network elements.
- A single call control signaling protocol enhances the ability to provide a virtual Home Environment for telephony and multimedia services.

1.2 Protocol Selection Options:

Option A: Deploy a UMTS Release 2000 architecture using only the ITU-T H.323-based Call Control Protocol.

Option B: Deploy a UMTS Release 2000 architecture using only the Session Initiation Protocol (SIP) based Call Control Protocol.

1.3 Scope of the Requested Effort:

The scope of the analysis is for the vendor community (and Carriers) to come up with a broad comparative analysis of call control protocol options. The analysis should aim to answer the several questions that are listed in the following section of this document. The criteria for the analysis are:

- **Time to market**
- **Estimated quantification of the work effort required**
- **Identification and Qualification of the Impact on Network Elements**

The analysis should conform broadly to the assumptions outlined in this ad-hoc document.

1.4 Key Assumptions for the Analysis:

- Standardization completed by 3GPP Release 2000 (EOY 2000).
- Supporting requirements and assumptions set forth in UMTS release 2000 (It is recognized that this work is still in progress).

2 H.323/SIP Comparison: Important Factors to Consider When Choosing a Protocol

For responding, consider H.323 protocol version 4. Backwards compatibility to older versions need not be considered.

Does either call control protocol (H.323 or SIP) provide a significant advantage over the other in terms to capability, time-to-market, complexity, operations, administration, management, intersystem operation, compatibility with other technologies, or evolution? Please quantify, and consider the following:

2.1 Complexity

2.1.1 Message set comparison

We consider air-link optimized call flows for setting up a mobile to PSTN call using SIP and H.323v4. The number of messages exchanged and typical message sizes are compared. A functional comparison is made of the various messages in the SIP and H.323 protocol suites.

We have assumed that a mass deployed default audio codec is known to both sides and thus a graceful media negotiation mechanism is not necessary.

The transport layer for both SIP and H.323 is assumed to be UDP (again for fair comparison). However, since the last message for call set-up (200 OK for SIP and CONNECT for H.323) is related to a billable event, the receipt of these messages must be acknowledged.

In the SIP case, the 200 OK message is always acknowledged. For H.323, we have chosen to utilize the ANNEX E mixed mode feature of H.323 to provide an ACK message for the H.225 CONNECT message.

The flows assume that pre-answer announcements can occur.

2.1.1.1 Optimized Call Setup Flows

Figures s1 and s2 shows the call flows for SIP and H.323, respectively.

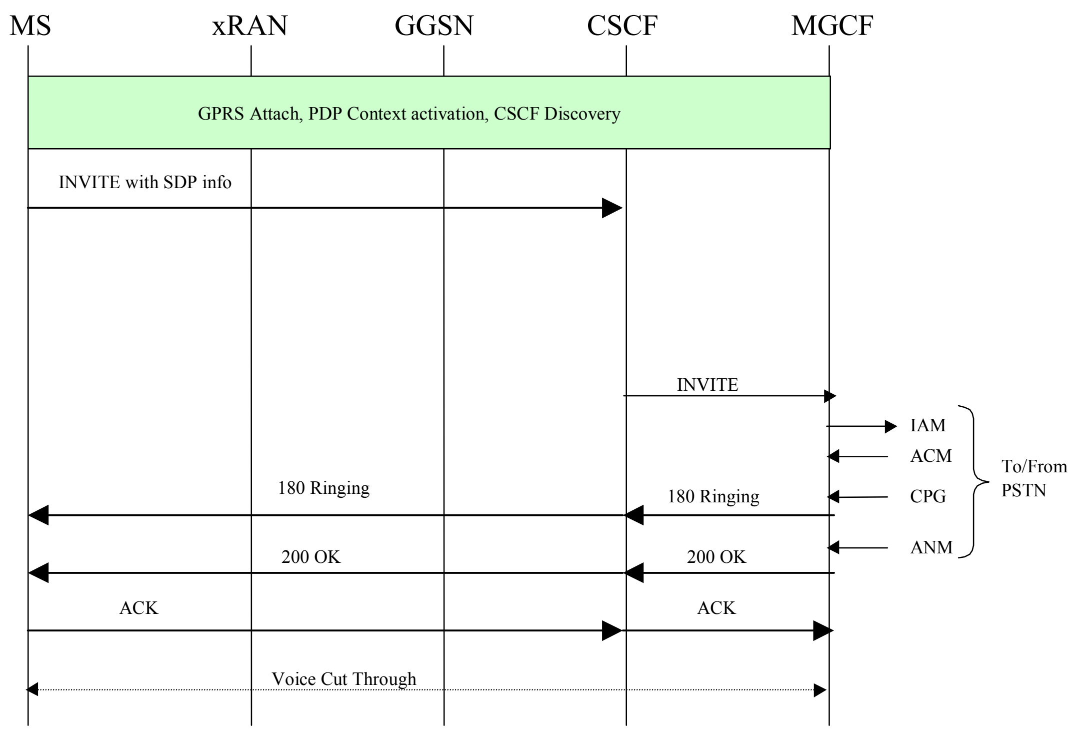

Figure s1. Partial call flow for a mobile-PSTN call using SIP

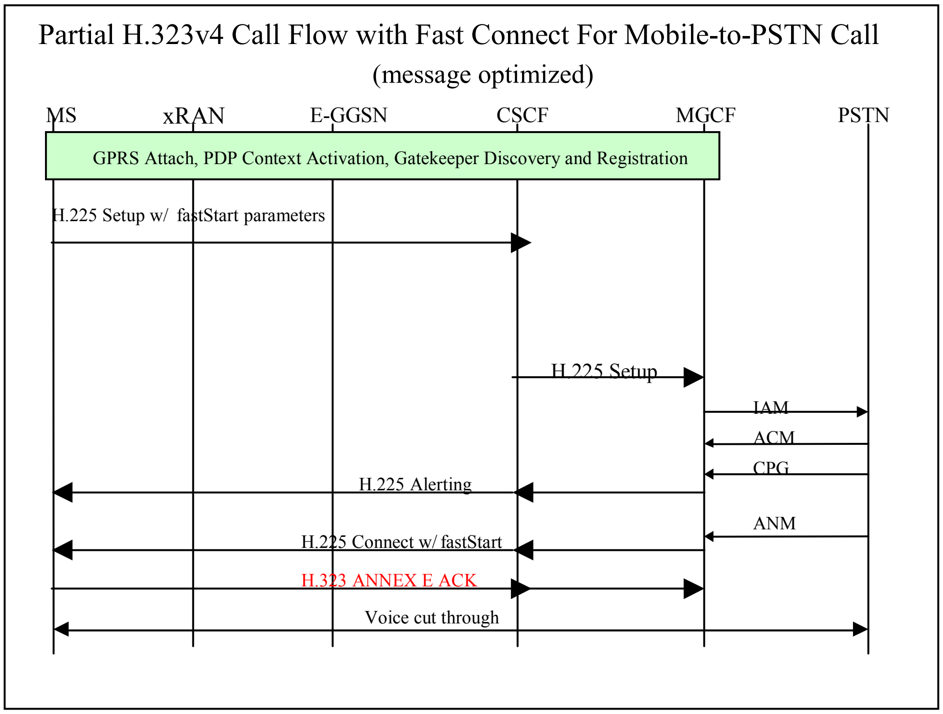

Figure s2. Partial call flow for a mobile-PSTN call using H.323

Figure s2 employs the most optimal method for call setup that H.323 could provide that matches SIP reliability. It employs the use of the H.323 Annex E mixed mode transport protocol; however. It should be noted that the use of Annex E as a transport layer is not well established and issues such as as gatekeeper to gatekeeper acking are still in question.

2.1.1.2 Message Size Comparison

Table st1 compares the number of message exchanged and typical sizes of the call control messages exchanged. For H.323, the messages are already ASN.1 PER byte-aligned encoded and hence, compressed. For the SIP messages, we have applied both a simple tokenization technique and a text compression algorithm similar to LZ77. We recommend that only the tokenizing method is used for last hop wireless specific signaling.

	SIP	**H.323v4**
# Of Messages Exchanged	4	4
Typical Message Size (bytes)	INVITE : 231 OK : 271 Ringing: 165 ACK : 171	Setup : 390 Alerting : 97 Connect : 280 ANNEX E ACK: 32
Total bytes	838	799

Table st1 : Message set comparison based on call flows

The following table compares the results of the various compression schemes applied to the original SIP messages of our example trace. All sizes are in bytes.

SIP messages	original size	size after simple tokenization	size after tokenization & LZ77 compression
INVITE 180 Ringing 200 OK ACK	346 217 440 237	297 180 396 218	231 165 271 171
Total	1240	1091	838

Table st2 : Comparison of the compression schemes on SIP messages

2.1.1.3 Functional Message Set Comparision

Functions	SIP	H.323
Registration	REGISTER, ACK	RAS: RRQ, RCF, RRJ, GRQ, GCF, GRJ
Admission	INVITE	RAS: ARQ, ACF, ARJ
Call Initiation	INVITE	H.225 Setup
Capability Exchange	SDP, OPTIONS, NOT SUPPORTED	H.245 Open Logical Channel (Can be embedded in H.225 for Fast Start). H.245 Terminal Capability Set and Ack can be used for slow-start.
Resource Allocation	No specific signaling	Some consider ARQ, GRQ a crude form of resource reservation at admission. For per-session allocation there is No specific signaling.
Status	Any response/acknowledgement messages (1xx-5xx) ex. O.K., Ringing, Progress, ACK	Alterting, Progress, Call Proceeding, Connect
Teardown	BYE, Ack	Release Complete
Reliability	Has timers and session methods/messages to achieve reliability. SIP is reliable over UDP. SIP can also bundle requests and responses.	Relies on TCP, SCTP or Annex E. H.323 over UDP alone is not reliable. H.323 has timers as well.

Conclusions:
In terms of message set comparison for complexity SIP and H.323 are similar. SIP implements an ACK to the OK (answer) message; whereas, H.323 was designed to rely on either TCP or Annex E for this function. There may still be a few issues with the proxy of the Annex E ack between gatekeepers. Annex E is not widely deployed and there are many variants. We do not believe it's interoperability has been significantly tested in open interoperability events.

In terms of pure message size SIP and H.323 using TCP are roughly the same. SIP has 4 basic messages in the session setup. H.323 fast-connect has only 3 but must rely on the transport layer for reliability, increasing the transport level messaging and bytes transmitted. If Annex E mixed mode with Gatekeeper proxying of transport ACKs is used in conjunction with H.323 for reliability then H.323 fast connect has less total bytes transmitted than SIP even after applying compression.

H.323 over pure UDP is not currently reliable. H.225 could be altered in H323v4 timeframe to have a session level ACK message to the Connect in order to provide reliability over UDP.

2.1.2 Encoding, parsing and generation

H.323 signaling protocols such as H.225 and H.245 use ASN.1 byte-aligned Packed Encoding Rules (PER). This involves the use of encoding and decoding code generated by an ASN.1 compiler from a text based ASN.1 syntax definition file.

It should be pointed out that for very simple messages that employ ASN.1 PER, direct generation of the encoding can be used. This greatly increases development time and usually limits parameter optionality. In other words, it defeats a primary reason for using ASN.1 in the first place.

This does eliminate the need for a separate encoding process but can only be used under very restricted circumstances.

This analysis assumes that encoding is performed as a separate process from generation.

SIP is a pure text based protocol patterned after HTML. SIP generation involves no separate encoding overhead. Tokenized compression does not employ any extra processing. Total compression mechanisms would involve extra CPU processing. On the receiving end SIP employs a text parser similar to the parsers used in web browsers to determine the values of fields identified by tags.

2.1.2.1 Encoding and Generation:

Encoding and generation are separate processes in H.323; whereas in SIP, there is no separate encoding process; therefore, it is safe to say that there may be close to 2 times the number of operations performed per field in H.323 as there are in SIP. This is due to each field having to be looked at once for generation and once for encoding (i.e. twice the number of reads and twice the number of writes into the correct stream format).

We should point out again; however, that direct combined generation and encoding can occur on the H.323 signaling. In this case SIP and H.323 are roughly equivalent. But again, this has a dramatic affect on development time and this method may not be able to be used at all depending on the optionality of the fields contained within the message.

SIP message generation employs string writes of headers and string writes of values corresponding to these headers. It is fairly straightforward and easy to program making it less complex. Many processors sold today are optimized for string operations using a single clock cycle.

If tokenizing compresssion is employed on the SIP messages which involves simple substitution of the header fields and other known reserved words with tokens, no additional overhead is incurred. It is a part of generation. In addition, this form of compression does not destroy the extensibility mechanisms of SIP. If the header or field does not have a matching token, the header or field is not tokenized. They are left as is.

We also need to point out that PER aligned encoding employs the use of a preamble bit map to describe characteristics of the records and primitive types contained thereafter. The encoding of the preamble for each field that employs optionality and extensibility is deterministic upon the values of the fields at run time. Having to deal with these bit operations during the encoding process incurs a significant amount of additional per-field processing overhead on H.323 that does not exist with SIP.

This processing translates to up to 5 extra write operations per field.

Conclusion: SIP compression/generation overhead is less than H.323 ASN.1 PER encoding/generation. This will have an effect on the mobile stations' power budget as well as the overall complexity of the CSCF.

2.1.2.2 Decoding and Parsing:

H.323 signaling protocols employ ASN.1 PER compiler generated decoding procedures to decode incoming messages.

SIP employs a text parser, which identifies fields via tags and values thereafter separated by delimiters. Comparisons with each TAG parsed must be string compared with a list of all of the available headers. Hashing functions are means of optimizing this process. SIP has been designed with TAG values that facilitate this hashing optimization. The number of operations to perform this hash is expected to be on average 3. SDP is parsed in a similar manner.

Both methods are order N operations. SIP parsing involves byte and string comparisons; whereas, H.323 involves per-field bit map operations as well as string comparisons. Many of the same points above apply to decoding as well for H.323.

H.323 decoder processing translates to 5 read and comparison operations per field.

If simple tokenizing is employed to reduce SIP tags to 1 byte tokens, the affect on processing overhead is actually reduced as tokenizing takes only one hash operation.

Conclusion: SIP parsing overhead is about the same as H.323 ASN.1 PER decoding. It’s basically as wash. SIP parsing overhead used in conjunction with tokenized compression is less than H.323 ANS.1 PER decoding. This will have an effect on the mobile stations' power budget as well as the overall cost of the CSCF.

2.1.2.3 Overall Conclusion and Ramifications:

SIP is less complex than H.323 for parsing, decoding and compression, encoding and generation.

CPU overhead is implementation specific and could vary accordingly.

Care should be taken in the selection of the processor of mobiles, MGCFs and CSCFs if SIP is chosen as the least common denominator. Processors should be optimized for string operations.

Another point that should be made here is ease of programming and vendor selection. As H.323 is complex with regard to encoding/decoding, the number of competing vendors and operators for the source and services which employ the protocol may be reduced.

2.1.3 Debugging

SIP debugging is quite simple as the signaling is based on text and tags similar to HTML. This means that SIP debugging is self-adapting as standards and new services are defined and deployed. No work or extra time is needed to update the debugging tools.

H.323 debugging requires specialized tools that must adapt to the standards as they change.

Conclusion:
The advantages detailed above favor SIP and the affect is reduced TTM intervals as well as reduced complexity for development.

2.1.4 Reuse of existing code and procedures

H.323's signaling methods and flows were patterned upon the ISDN multimedia standards. However, as the standard progressed, most of the fields that H.225 and H.245 required (ex. RTP port) were added and have little similarity to the ISDN world. The H.323 standard did base their signaling off of Q.931 but for a quite different purpose. H.225.0 is used to set up calls completely independently of media connections, contrary to ISDN. Furthermore, H.225.0 only includes a subset of the Q.931 protocol, and very often, the meaning for each common element is completely different. The original idea was to facilitate interworking, but in retrospect, it seems to have created more confusion in the end.

Most vendors have implemented H.323 using generic H.323 stacks (there are a very limited number of interoperable and maintained stacks. These stacks have little in common with existing ISDN stacks.

SIP comes out of the web world. One could possibly re-use or pattern a SIP parser off an MGCP or HTML parser. As SIP has been designed in a modular fashion, it was designed to be a defacto push technology for the web. So one could say that it was designed to re-use and interoperate with all of the web technologies and even some H.323 technologies such as: HTTP, ACAP, LDAP, FTP, RTP, RTSP, T.120, etc.. providing the session layer and user location functions seamlessly for appropriate application session establishment without any need for changes to these other protocols and applications.

Conclusion:
Some of the issues related to PSTN interworking have been thought-out for a longer period of time in H.323 than with SIP (ex addressing received earlier attention in H.323, tones and announcements, etc.). However, the underlying principles are "portable" (and were indeed ported over from the ISDN). The porting of these principles to SIP is under way right now largely due to the interest of operators and vendors that recognize the long-term benefits.

If you see more future generated revenues based upon the fast integration of web technologies and see the TTM value and flexibility that SIP provides operating in control of both circuit-based endpoints as well as IP endpoints, then SIP wins.

We do need to mention the HTTP URL parameter that will be part of H.323v4. This parameter significantly evens the playing field for web-based applications.

2.1.5 Service and feature implementation and protocol interactions

Please refer to the previous and following sections.

2.1.6 Options and methods for implementing services that are available in the protocol

Both protocols can be made to support the IN model, the CGI model, 3rd Party redirection, feature code signaling, etc..

H.323 has in the past been geared towards a particular set of services: those defined in H.450 (itself derived from QSIG), T.120 data applications and basic voice/video setup. We now see that H.323 can be used to push an http URL in version 4.

SIP was designed to be the defacto push and presence application. It was designed to use SDP as it's media description language and which allows for many of the new features already such as pushing a URL. As these protocols were built to interwork with web technologies from the start, it basically has an advantage over H.323 in this respect. Many of the web-related functions are already built in; whereas, H.323 will likely have to add new fields such as an HTTP URL parameter. SIP and SDP can be used to facilitate T.120 applications as well.

Conclusion:

H.323 seems to have to add specific fields or new values for parameters in the signaling in order to allow newer web related services. This means that operators will be waiting on their vendors intervention to provide these new related web services more often than with SIP. A prime example of this is the addition of an http URL parameter in H.323v4.

There is no need to extend SIP or even SDP to achieve the same functionality. Due to it's modular design, it does and always has done this function transparently.

2.1.7 Interworking with the PSTN

This is a problem that has to be ironed out period. The signaling protocol has little to do with it. Most of the issues being debated now deal with inband streaming and QOS interactions. The signaling protocol employed whether it be foo or bar is irrelevant.

One could try to make the case made in section 2.1.4 Reuse of existing code and procedures but that's the end of it. SIP used to have a bit of leg up by providing for ISUP encapsulation. H.323 has now included a similar mechanism in H.323v4.

2.1.8 Implementation in UMTS Release 2000 network elements and user devices

SIP stack size is less than H.323. This has the effect of reducing the cost (memory) of the devices.

We believe SIP has a significant TTM advantage due to it's text-based debugging and modular design.

2.2 Extensibility

2.2.1 Compatibility among versions (built into the protocol?)

Both suites of protocols have version identifiers that can be used to control extensibility mechanisms based on version.

SIP: SIP does not have explicit requirements for compatibility among versions. Unknown/unsupported headers are ignored by default. This reduces code size and protocol complexity. Also, this provides flexibility in terms of developing/evolving features and makes encoding/decoding clean and concise.

An adverse effect of this may be that features supported by the older versions may not be supported by newer version. Some may see this as an advantage, however; baggage gets dropped over time and code size is kept in check. Keep in mind that this perceived adverse affect is optional. You can keep supporting the past as long as you wish or drop it at any time you deem fit.

The REQUIRE header allows for an end system implementation to require a specific header. The PROXY REQUIRE and a proposed SUPPORTED header in addition to the REQUIRE header allows for intermediary proxies, registrars and redirect servers to require a specific header.

H.323: H.323 requires full backward compatibility. This ensures continuous support of existing features. Note that, although standards explicitly specify backward compatibility, vendors may chose to support only last 2 or 3 versions. This may reduce size of messages and protocol/ implementation complexity. Due to enormity of H.323's baggage already, we recommend that support of old versions prior to H.323v4 not be supported.

2.2.2 Feature evolution

SIP: Using SIP, feature may be evolved by extending or defining new SIP header information. Current SIP RFC defines default headers and some extensions. New extensions can be added as a part of separate RFCs. Refer to "Ability of carriers to define own features and services" section of "Extensibility" of this document for details.

H.323: H.323 defines NonStandardParameter structure to extend vendor specific (proprietary) features. For extension/modifications to NonStandardsParameter, refer to "Ability of carriers to define own features and services" section of "Extensibility" of this document. If changes are made to existing capabilities or control message parameters other than NonStandardParameter, new version of corresponding specifications may need to be issued. Additionally, new features could be implemented using the new H.450.1 generic functional protocol.

2.2.3 Ability for carriers to define own features and services

SIP: In general, SIP defines methods, default request/response headers and status codes. Service definition is included in the header itself. The extensions to existing headers may require changes in RFCs where original header is defined. But the new headers can be easily added by having definition in separate RFC or other standards process.

It provides a hierarchical namespace of status and also for features/services. In this way, new features/services can be added easily without changing message contents. For example, new (perhaps proprietary) features/services can be introduced in two ways 1). By registering new features/services with Internet Assigned Numbers Authority (IANA). 2). By deriving hierarchically from the feature owner's Internet domain name, giving hints to where further information might be found.

Also, SIP client can inquire about SIP server abilities first or proceed under the assumption that the server supports the extension and then back off if the assumption was wrong.

Currently, SIP does not employ a traditional graceful mechanism for two-way service negotiation. It does have two ungraceful methods, however; "not supported" and the more pessimistic "options" method. The thinking here was that negotiation will likely not have to be done on most session transactions. To tax the state machine and increase messaging because of this seemed sub-optimal. The MMUSIC WG in the IETF has an action to resolve some general issues with the simultaneous asymmetric multiple media negotiation. We expect them to make minor enhancements to SDP to resolve the issue as opposed to SIP, though.

Organization headers can also be used in conjunction with the contact and record-route headers to implement operator specific enhancements. If the proxy require is not used and given that operators have control over the source for their application proxies as well as the source in the UA's, new services can be deployed over an existing proxy based wireless capable IP environment without any intervention of vendors.

The built-in extensibility mechanisms, ASCII nature of SIP and it's modularity will significantly impact the ability of operators to define and deploy their own services without intervention from vendors in short market windows.

H.323: In H.323, non-standard capabilities and control messages may be issued using the NonStandardParameter structure defined in ASN.1. This NonStandardParameter structure consists of vendor codes (NonStandardParameter identifiers) and data associated with the particular code. Note that the type of data is Octet String and may be unlimited. OBJECT IDENTIFIER can be used to identify vendors, service providers or organization and are also hierarchical (but not maintained by IANA).

Since vendor codes are defined as a part of ASN.1 definition, additional vendors need to request change in specifications. Definition of services and some other parameters (other than NonStandardParameter) in ASN.1 may be extensible and hence modifiable. If additional parameters are required, they can be added to existing version of H.323 as optional parameters in the end of message structure.

Due to the complexity of defining new parameters in H.323. Deployment of new services will likely result in the operator waiting for their vendors to add the new parameter.

2.2.4 Modularity of the protocol to allow for the easy evolution to new services and features

SIP: SIP encompasses mainly user location, registration and basic session signaling. For advanced services/features, other functions like capability exchange, service discovery, QoS, directory access, conference control are essential. All these functions are resided in separate protocols and can be used with SIP without making any changes to the SIP protocol. In addition, SIP's extensibility allows for new headers to be added without changes to pass-through proxies or even client user agents.

H.323: H.323 is an umbrella feature containing vertically integrated sub-protocol suite of H.225, H.245, H.450, RAS, Q.931 etc. Hence, from feature/service perspective, there is no clean separation of these sub-protocols. This results in to more interactions between its sub-protocols. Also, most of the services are in-built and intertwined between more than one sub-protocol.

2.2.5 Ability to work with existing and new multimedia codecs

SIP: SIP uses Session Description Protocol (SDP) to convey the codecs supported by an endpoint in a session. Codecs are identified by string names that can be registered by any person or group with IANA and then used. This means that SIP can work with any codec and other implementation can determine the name of codec and contact information for it, from IANA.

H.323: The GenericCapability type in H.323 allows new codecs to be specified in such a way that a new version of H.245 syntax does not need to be issued.

2.2.6 Third party call control mechanisms

This mechanism can be defined as the ability for a party to set up a call between other parties without necessarily participating in the call e.g. a secretary dials for a manager, operator service etc.

These mechanisms allow a third party to instruct another entity to create and destroy calls to other entities upon requested by server. Here, a call control protocol like SIP and H.323 can be used between the server and third party. As the third party executes the instructions, status messages are passed back to the server. This allows the server to take further actions based on some local program execution.

SIP: SIP can support this type of mechanisms by using "also" header in requests and responses. In addition the contact and record-route headers allow for a 3rd parties or home elements to always be in the loop.

H.323: There is no standard comprehensive way to do third party call control in H.323. The FACILITY redirection feature can be used to “deflect” a call at setup time to another location.

2.3 Scalability

2.3.1 Support for large numbers of domains (wide area addressing, user location, etc.)

H.323: The initial intent of the protocol was for the support of LANs, so it was not inherently designed for wide area addressing. The concept of a zone was added to accommodate wide area addressing. Procedures are defined for ”user location” across zones for email names. Annex G defines communication between administrative domains, describing methods to allow for address resolution, access authorization and usage reporting between administrative domains. In multi-domain searches, there is no easy way to perform loop detection. Performing the loop detection can be done (using the PathValue field), but introduces other issues related to scalability (e.g. how to define the field value and how the value should change when the network configuration changes).

SIP: SIP inherently supports wide area addressing. When multiple servers are involved in setting up a call, SIP uses a loop detection algorithm similar to the one used in BGP, which can be done in a stateless manner, thus avoiding scalability issues. The SIP Registrar and redirect servers were designed to support user location.

In both SIP and H.323, the burden of scalability is in the SIP server or Gatekeeper, the underlying transport layer, and in the way they can communicate with their peers. They can both accommodate different topologies (hierarchical, flat, etc.). They can both make use of DNS, directories, internal translation databases or other location and translation mechanisms which can facilitate global deployment.

2.3.2 Ability to handle large numbers of calls

This issue is primarily implementation/deployment specific. Theoretically, a stateless implementation allows a gateway/server to support a larger number of calls. Both protocols can or can be made to support n to n load balancing. The complexity and statefullness of the distribution nodes and endpoints is dependent upon the transport protocols used as well as the translation mechanisms employed.

H.323: Although not initially supported, H.323 call control can be implemented in a stateless manner. A gateway can use messages defined in H.225 to assist the gatekeeper in performing load balancing across gateways.

SIP: Call control can be implemented in a call stateless manner. SIP supports n to n scaling between U/As and servers. SIP takes less CPU cycles to generate signaling messages; therefore a server could theoretically handle more transactions. SIP has specified a method of load balancing based upon the DNS SRV record translation mechanisms. This method may not be appropriate for a carrier grade environment. For instance it does not handle local translations. There is a current proposal to alter DNS for this task, however (local DNS).

2.3.3 Maintaining of call states (stateless or stateful) effect on scalability

This issue is also implementation/deployment specific. However, typically, scalability is reduced when a server must maintain call states as a given call must use the same server while the call is active or the call states must be transferred if a server goes down. Stateless processing allows a given transaction to be processed by any server providing the required functionality, since theoretically no transient information need be maintained by the server.

H.323: Supports both stateful and stateless processing. Scalability is reduced in the stateful mode as the same server (or shared memory) must be used while the call is active. Most existing carrier grade implementations of H.323 gatekeepers are designed to be call stateful and employ hot-sparring technologies in order to meet carrier grade requirements. This increases the complexity of these devices. There is nothing in the H.323 specifications that prevent gatekeepers from being implemented using call-stateless N+1 redundancy and load sharing technologies, however.

SIP: Transactions in servers and gateways can be stateless or stateful. The stateful mode decreases scalability as the same server (or shared memory) must be used while the transaction is active. Most current implementations of SIP proxies are designed to be call stateless and rely on N+1 redundancy and load-sharing technologies to meet carrier grade requirements. This reduces the complexity of these devices. There is nothing in the SIP specification that prevents the use of hot-sparing technologies from being used, however.

2.3.4 Elements that must maintain states

This issue is primarily implementation dependent.

H.323: In a call-stateful implementation, the terminal, the Gatekeeper and any Gateways must maintain states. In a call-stateless implementation, the terminal must maintain states. Most current H.323 gatekeeper implementations are designed to be call stateful.

SIP: In a call-stateless implementation, only the terminal (i.e. endpoints) must maintain states. In a call-stateful implementation, the terminal, the proxy server and perhaps the redirect and registrar servers (if implemented separately) would maintain call states. Most SIP proxy implementations are designed to be call-stateless.

2.3.5 Signaling message processing

This issue is implementation dependent; however, 2 factors greatly influencing the scalability are whether call states are maintained and the use of UDP versus TCP to transport the signaling messages.

No connection states are required in UDP, so scalability is improved. Stateless calls using are more scaleable than stateful calls.

SIP can be based on UDP and stateless call processing without any loss of reliability. H.323 requires either TCP or the transports defined in H.323 Annex E to have the same level of reliability.

The encoding/decoding/parsing mechanisms also have an impact on scalability due to consumption of CPU processing.

H.323: Messages are ASN.1 encoded, using aligned PER.

SIP: Messages are text based. Same message set is used between services and call control entities.

2.3.6 Conference sizes, conference control (centralized vs. distributed)

The implementation of a central control point for conferences reduces the size and number of supportable multiparty conferences. Distributed conference processing scales more easily to larger numbers and sizes of multiparty conferences.

H.323: H.323 conferencing was initially based on a centralized conferencing control mechanism requiring an MC. To support larger conferences, H.323 allows an application layer multicast conference concept. H.245 provides feedback during conferencing allowing receivers to control encodings, transmission rates and error recovery. This mechanism does not easily scale for multipoint conferences. However, H.332 extends H.323 for ”loosely-coupled” conferences.

SIP: SIP conferencing is originally based on distributed conference control, thus larger conferences can be easily supported. SIP uses RTCP for conference feedback. Since RTCP is also distributed this feedback mechanism easily scales to support larger conferences. SIP can also work in conjunction with centralized conferences. While SIP can be used as the signaling protocol to implement an MC function (SIP robot or the UA itself), SIP does not provide any specific method of floor control, etc..

2.4 Resource Utilization and Management

2.4.1 Resource required during the call

The resources required during the call is addressed in two parts - (i) resource required during call set-up in terms of air-link bandwidth, and (ii) resource required in terms of CPU power and memory in the client and the server.

Air-link bandwidth required during call set-up:

The air-link bandwidth required during call set-up is related to the number and size of messages exchanged. It is expected that there will not be any spectacular difference in the number of messages and the total number of bytes exchanged over the air link for a typical call set-up using SIP or H.323. Comparison for a typical mobile-PSTN call set-up scenario is given in Tables st1 and st2.

Resource required in terms of CPU power and memory in client and server:

H323V4 : H.323v4 messages can be storage efficient if packed encoding rules (PER) are used. H.323v4 is a fairly complex set of protocols and includes H.225 for call signaling, H.245 for call control, H.332 for large conferences, H.450.x for supplemental services, H.235 for security and encryption and H.246 for inter-operability with circuit-switched services. Many services require complex interaction between these sub-protocols. The protocol state interactions and management in the client is directly related to these interactions. The H.323v4 server can be either stateful or stateless as TCP or UDP can be used for the transport (this is different from call state). Usage of UDP or mixed Annex E transport significantly reduces the memory requirements. Call state information needs to be kept in the Gatekeeper when a "Gatekeeper Routed" call model is used. If a "Direct Routing" call model is used, then the Gatekeeper may be used only for Registration, admission control and address translation.

SIP : Textual formats used in SIP are less space efficient than ASN.1 PER. On the other hand, the SIP parser is fairly simple and can be written in less than 500 lines of code, which is much less than a general ASN.1 decoder.

SIP/SDP are also less complicated than H.323 suite. This implies that maintenance of protocol state information at the client and server are expected to be less burdensome than H.323v4. In SIP, the transaction through proxy-servers may be either stateful or stateless. In case of stateful servers, the requirements for maintaining state information are expected to be roughly equivalent for both SIP and H.323v4.

2.4.2 Resource minimization

In the wireline case, there are three proposals dealing with this issue all using SIP: (i) DOCSIS proposal (PacketCable Dynamic QoS Specification), which distinguishes between the *authorized, admitted* and *committed* resources (each upper bounded by the previous one), (ii) Usage of SIP to provide QoS guaranteed path (draft-gibson-sip-qos-resv-00.txt), which attempts to minimize the call set-up delay by tightly coupling the transport resource allocation with the session set-up, and (iii) Interdomain IP Communications with QoS … (draft-sinreich-interdomain-sip-qos-osp-00.txt), which proposes end-to-end usage of RSVP (for signaling only across multiple inter-operator domains – not int-serv) to set-up resource reservation. The current proposal for H.323 is similar to (ii) referenced above, however, it is expected that similar solutions could be worked out for H.323 as well and hence, the wireline resource reservation / minimization issue is by-and-large protocol independent.

However, one of the primary concerns for the cellular wireless access is minimization of the amount of information exchanged over the air-link. This calls for minimizing the number of messages (as well as the message sizes) exchanged for call set-up/resource reservation over the wireless link. We have proposed ways for minimizing the over-the-air messaging for both SIP and H.323 for a typical mobile-to-PSTN call as shown in Figures s1 and s2. Also, the ASN.1 PER encoding followed by H.323 will generally allow smaller message sizes than the basic textual SIP messages. Since the existence of suitable compression / encoding techniques for SIP messages is unknown, we have used simple text compression techniques to compress the SIP messages. The comparison is shown in Table st2.

2.4.3 If compression is applied to each signaling protocol (H.323 and SIP), what gains could be achieved percentage wise.

As H.323 signaling protocols employ ASN.1 PER aligned encoding rules, subsequent compression on these messages would be minimal.

We have chosen to apply both simple tag tokenizing techniques which we recommend to the SIP messages in order to achieve an optimal compression with no extra CPU overhead. In addition we have also applied a common text compression technique. We do not recommend the use of this method. The results are characterized below:

Method	SIP	H.323
Tokenization	13 – 19 %	Not Applicable
LZ77 after Tokenization	18 – 22 %	1-3 %
TOTAL	31 – 41 %	1-3 %

We expect that work to progress on tokenized compression methods for SIP and SDP will result in no need for applying a separate LZ77 compression mechanism.

2.5 Services

2.5.1 Services supported

H.323: Call Hold, Call Transfer (Blind, Alternative and Operator Assisted), Call Forwarding, Call Waiting, Conferencing (Multicast, Multi-unicast, Bridged, Consultative), Call Park, Call Pickup, Call Completion on Busy Subscriber, Calling Line ID, Message Waiting Indication**.** The services supported are standardized in the H.450 series of specifications, however, the actual support of these services is implementation specific and they are not currently widely deployed. There are some doubts in the industry if the H.450-series services will ever become widely deployed.

SIP: Call Hold, Call Transfer (Blind, Alternative and Operator Assisted), Call Forwarding, Call Waiting, Conferencing (Multicast, Multi-unicast, Bridged), Call Park, Directed Call Pickup, Calling Line ID, Call Return, Follow-me, Find me, Camp On, Call Queuing, Automatic Call Distribution, Do Not Disturb, Third Party Call Control. Note that the actual support of the services is implementation specific. The current SIP RFCs do not rigorously define the services. These are usually left to white-papers and perhaps informational RFCs.

2.5.2 Delay times to acquire services, both basic (such as dial tone or post-dial delay) and supplementary services

H.323: Example: With Fast Call Setup, there is a delay of 3-4 roundtrips (using TCP). Using UDP, call setup delay can be 1.5-2.5 round trips, depending upon whether or not a gatekeeper is involved. (Note, however, that with the fast call setup, capabilities are not exchanged, but can be exchanged later using H.245). Simultaneously sets up TCP connection as well to provide support in the case that the UDP setup fails.
SIP: Call setup delay is equivalent to H.323 Fast call setup. However, the establishment of the TCP connection (as a backup for the UDP setup failure) is sequential.

2.5.3 Billing and accounting

It is expected that distributed billing models can be applied to both protocols equally well.

H.323: Billing and accounting are not explicitly defined by the protocol, however, mechanisms existed and can be defined depending upon the requirements of the service provider. Gatekeepers can maintain logs and generate CDRs. A Gatekeeper can also instruct gateways to send copies of specific messages, for billing and accounting purposes. Version 4, which is still in draft form, may be adding procedures to provide the billing information (call duration, call termination cause, etc.) from the gateways to its gatekeeper to aid in the generation of CDRs. Annex G specifies that administrative domains may request other domains to provide them information about the usage of resources in specific calls. UsageIndication messages may be provided at any stage of the call. The ETSI TIPHON group has defined OSP for this purpose.

SIP: The functionality for billing for SIP depends on whether the service provider plans to charge for SIP services, for gateway services to the PSTN, or for carrying media data. For SIP services, the Authorization header can be used to indicate a customer identity that associates a SIP request with a billable entity. SIP server operations can be charged based on server logs or, for real-time billing, via AAA. For gateway services, the gateway can generate call detail records (CDRs). When QoS mechanisms are involved in a call, it would seem likely that these mechanisms would be responsible for the charging mechanism. Actual accounting records may be generated by AAA protocols or log files. Also, note that the DCS group drafts propose a billing extension to SIP for messaging between proxies. There is also a current proposal in the SIP working group to use OSP for Accounting purposes.

2.5.4 Comparison of services with existing wireless or wireline services (do the same set of services exist, not necessarily implemented the same).

H.323: Service set equivalent to existing wireless and a subset of existing wireline (specifically centrex). Additional location based and internet specific services are being proposed for version 4. Version 4 also plans to transport more of the wireline PBX features.

SIP: Service set equivalent to existing wireless and a subset of existing wireline (specifically centrex). Additional location based and internet specific services are supported by design intent.

2.5.5 Capabilities exchange services provided in the protocol

H.323: Uses H.245 protocol for capabilities exchange. The complete set of what a terminal can receive and decode is made known to the other terminal via its capability set. Precise information about each terminal's capabilities can be expressed in the CapabilityDescriptor structure. In the case of fastStart, an endpoint presents OpenLogicalChannel structures all of the capabilities it can support in both directions. The receiving end will choose from that list and return the OpenLogicalChannel structures it chose to use in a response message.

SIP: Uses SDP for media capabilities exchange. SIP itself also relies on the require and proxy-require headers with the Not Supported and Options methods for session signaling capability exchange. Typically, each endpoint tells the other what capabilities it can receive. The confirmation of which one is chosen is implicit in where the media streams will be sent. Callers can use an OPTION request to find out the capabilities of the callee. Another option for capabilities exchange is to use an INVITE message, which is replied by a 480 Unsupported message, which results in a second INVITE. One drawback is that SDP does not currently support asymmetric and simultaneous capabilities of audio and video encoding. The MMUSIC working group of the IETF has a work item to solve the asymmetic negotiation issue. They will likely enhance SDP "m=" tag's value formats. The value of providing this ability is somewhat in question, however.

2.5.6 Personal mobility services (delivery of services wherever the subscriber is located, network independent) and location based services

H.323: H.323 can redirect a caller to other addresses. Gatekeepers offer an inherent way for terminals to register/unregister at different locations. User preferences can be specified in user-user signaling. There are plans in version 4 to define these services (a substantial amount of work on mobility for H.323 has been started in ITU-T SG16). In the ETSI TIPHON WG 7, proposals have been made for the support of IP mobility, but nothing has yet been standardized.

SIP: SIP inherently supports the concept of personal mobility and location based services, when a call is being setup, using the SIP redirect server, server forking, registrar server concept and allowing users to proxy requests. Additional proposals have been made, but not accepted, for the support of IP mobility (characterized by frequent roaming and changing of location during a call) using the SIP redirect server. SIP was designed to use any protocol to access a location server – including SIP to a registrar server. We support the use of directories enabled with authentication and authorization mechanisms for home domain location servers.

2.5.7 Interworking and interoperability with legacy networks (wireline and wireless)

H.323: There are specific standards (H.246) specifying interoperability with legacy circuit switched networks. However, the H.246 recommendation and it's various annexes are not sufficiently complete in scope to be very valuable and should be viewed as a "guide" rather than a specification.

SIP: The standards approved to date do not specify interworking or interoperability. However, internet drafts have been written to define the interoperability.

2.5.8 Interworking and interoperability with other IP call control protocols (e.g., cable)

H.323: No explicit interworking and interoperability is defined with other IP call control protocols.

SIP: The DCS group drafts in the IETF SIP WG specify the interactions required in an IP cable environment.

ITU SG-16 has taken on the task of interoperability between SIP with H.323.

2.5.9 Security services provided, authentication of users and network elements, data privacy and encipherment

H.323: Authentication and security for H.323 is optional; however, if it is provided it must be in accordance with Recommendation H.235. RTP, which supports encryption, can be used to carry media. Between administrative domains, when authentication, data integrity and encryption is desired for messages, the IETF IPsec procedures are applicable (specifically, RFC 1825, 1826 and 1827). The ETSI TIPHON specifications define countermeasures to ensure a secure TIPHON compliant system. Security requirements are based on customer, service and network provider objectives for confidentiality, integrity, accountability, availability and non-repudiation. Lawful interception (a requirement in some countries) is also to be supported.

SIP: SIP can encrypt and authenticate signaling messages. RTP which carries the media supports encryption. IPsec procedures are also applicable between inter-domain network elements. The DSC group has made a proposal for support of lawful interception with SIP; however, there was great resistance in the IETF plenary to take on this type of work in the IETF. Perhaps this specification should be a work item for the ITU.

2.6 Wireless standards consideration

2.6.1 Where and how would any wireless specific changes to the protocol be implemented?

Ideally, there would be no wireless specific changes to these protocols. For location and personal mobility based services, a common wireline/wireless unified directory should provide user and terminal location as necessary for these protocols.

Tokenized compression of SIP and SDP could be standardized by the IETF, 3GPP or the WAP forum and issued to the IETF as an information RFC. It may be that the existing message sizes are acceptable and as such no wireless specific optimizations would be needed.