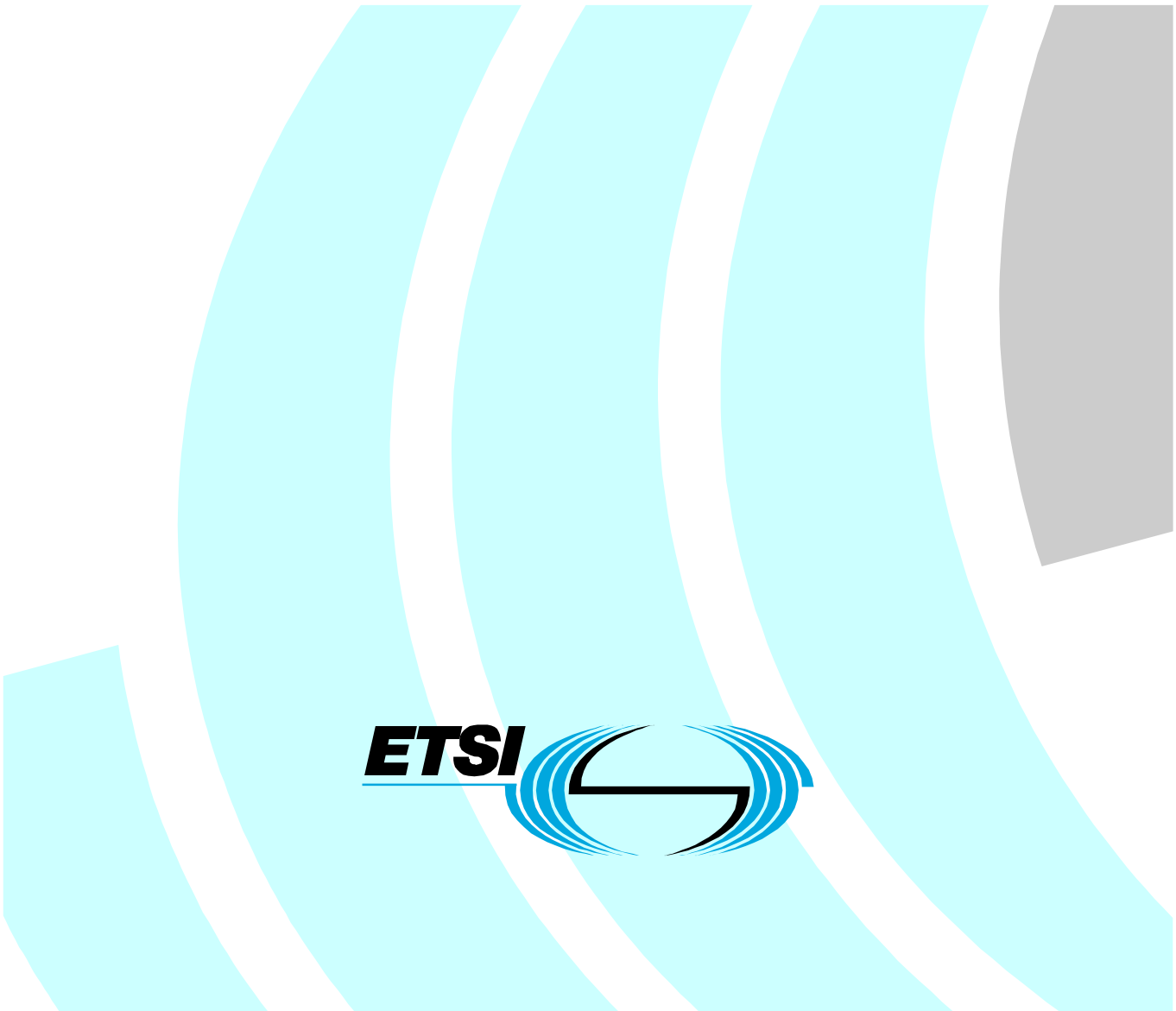
ETSI TR 102 019 V1.1.1 (2002-02)

Technical Report

Services and Protocols for Advanced Networks (SPAN); Design requirements for ITU J.arch, J.istp and J.tgcp



Reference
DTR/SPAN-000010

Keywords
cable, IP, protocol

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Services and Protocols for Advanced Networks (SPAN).

The present document describes the requirements used in the design and specification of the general architecture, the IPCablecom Signalling Transport Protocol (ISTP), and the Trunking Gateway Control Protocol of the ITU IPCablecom set of specifications. The requirements listed in the present document have been obtained by reverse engineering of the specific ITU Recommendations, i.e. J.istp or ITU-T RecommendationJ.165 [2] for the IPCablecom Signalling Transfer Protocol, J.arch or ITU-T Recommendation J.160 [1] for the architectural framework, and J.tgcp or ITU-T Recommendation J.171 [3] for the Trunking Gateway Control Protocol.

1 Scope

The present document specifies the design and specification requirements obtained from reverse engineering a set of ITU recommendations, consisting of J.arch (architectural framework), J.istp (IPCablecom Signalling Transfer Protocol), and J.tgcp (Trunking Gateway Control Protocol). These documents are part of a set of ITU documents describing the IPCablecom framework and its subparts. These documents are based on the PacketCable 1.0 specifications written by CableLabs in the United States, and modified to meet more global requirements and formatting conventions of the ITU.

The requirements obtained from reversed the specified documents can be explicitly listed or implied, commercial or technical. The present document does not contain all of the IPCablecom requirements, which cover a set of documents addressing additional areas of quality of service, security, lawful intercept, network interface, remote digital thermals, on network call singling, managed object provisioning, event messaging, and other major areas. Some operator requirements (example: billing, OSS presentation, etc.) are deliberately not covered in IPCablecom; these are considered normal to vendor/operation (RFQ/RFI) negotiations to meet subscriber needs in a market. These operational requirements are not covered here except where they are defined in the source documentation.

In the present document, all requirements have been presented with "MUST", "SHALL", "SHOULD", or "MAY" as the active verb. "It" in the present document when coupled to a requirement refers to IPCablecom in general. In particular, must and must no were used instead of shall or shall not to specify requirements, should or should not were used to specify recommendations, may or may not were used to specify permissions or possibilities, and can or can not were not used to specify possibly.

The present document contains the original requirements as created for the PacketCable 1.0 specifications (written by CableLabs in the United States). These base requirements can be realized by using other protocol stacks and are not restricted to those specified in the present document. The base requirements that are implied by the protocols should be included in any protocol implementation.

2 References

For the purposes of this Technical Report (TR) the following references apply:

[1] ITU-T Recommendation J.160 (J.arch): "Architectural framework for the delivery of time critical services over cable television networks using cable modems".

[2] ITU-T Recommendation J.165 (J.istp): "IPCablecom Signalling Transport Protocol".

[3] ITU-T Recommendation J.171 (J.tgcp): "IPCablecom Trunking Gateway Control Protocol".

[4] ITU-T Recommendation J.112: "Transmission systems for interactive cable television services".

[5] ITU-T Recommendation G.711: "Pulse code modulation (PCM) of voice frequencies".

[6] ITU-T Recommendation J.162: "Network call signalling protocol for the delivery of time critical services over cable television networks using cable modems".

[7] ITU-T Recommendation E.164: "The international public telecommunication numbering plan".

[8] IETF RFC 1899: "Request for Comments Summary RFC Numbers 1800-1899".

[9] IETF RFC 1890: "RTP Profile for Audio and Video Conferences with Minimal Control".

[10] IETF RFC 2131: "Dynamic Host Configuration Protocol".

[11] IETF RFC 2327: "SDP: Session Description Protocol".

[12] IETF RFC 821: "Simple Mail Transfer Protocol".

[13] ITU-T Recommendation J.170: "IPCablecom security specification".

[14] ITU-T Recommendation Q.724: "Telephone user part signalling procedures".

[15] ITU-T Recommendation T.30: "Procedures for document facsimile transmission in the general switched telephone network".

[16] ITU-T Recommendation V.21: "300 bits per second duplex modem standardized for use in the general switched telephone network".

[17] ITU-T Recommendation V.8: "Procedures for starting sessions of data transmission over the public switched telephone network".

[18] ITU-T Recommendation Q.35: "Technical characteristics of tones for the telephone service".

[19] ITU-T Recommendation V.18: "Operational and interworking requirements for DCEs operating in the text telephone mode".

[20] ITU-T Recommendation V.15: "Use of acoustic coupling for data transmission".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

Access Node (AN): As used in the present document, an Access Node is a layer two termination device that terminates the network end of the J.112 connection. It is technology specific. In ITU-T Recommendation J.112 [4] annex A it is called the INA while in annex B it is called the CMTS.

Cable Modem (CM): layer two termination device that terminates the customer end of the J.112 connection

gateway: devices bridging between the IPCablecom IP Voice Communication world and the PSTN

NOTE: Examples are the Media Gateway which provides the bearer circuit interfaces to the PSTN and transcodes the media stream, and the Signalling Gateway which sends and receives circuit switched network signalling to the edge of the IPCablecom network.

IPCablecom: ITU-T project that includes an architecture and a series of Recommendations that enable the delivery of real time services over the cable television networks using cable modems

Signalling Gateway (SG): signalling agent that receives/sends SCN native signalling at the edge of the IP network

NOTE: In particular the SS7 SG function translates variants ISUP and TCAP in an SS7-Internet Gateway to a common version of ISUP and TCAP.

3.2 Abbreviations

For the purposes of the present document, the following symbols apply:

AN	Access Node
ANC	Announcement Controller
ANP	Announcement Player
ANS	Announcement Server
CA	Call Agent
CIC	Circuit Identification Code
CM	Cable Modem
CMS	Call Management Server
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DPC	Destination Point Code
DTMF	Dual Tone Multi-Frequency
FQDN	Fully Qualified Domain Name
GC	Gate Controller
HFC	Hybrid Fibre Coax

HTTP	HyperText Transfer Protocol
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISTP	Internet Signalling Transport Protocol
ISUP	Integrated Services Digital Network User Part
LNP	Local Number Portability
MAC	Media Access Control
MF	Multi-Frequency
MG	Media Gateway
MGC	Media Gateway Controller
MGCP	Media Gateway Control Protocol
MIB	Management Information Base
MTA	Multimedia Terminal Adapter
MTP	Message Transfer Part
MWD	Maximum Waiting Time
NCS	Network based Call Signalling
NTP	Network Time Protocol
OSS	Operational Support System
PSTN	Public Switched Telephone Network
QoS	Quality-of-Service
RKS	Record Keeping Server
RTCP	Real-Time Control Protocol
RTO	Retransmission TimeOut
RTP	Real-time Transfer Protocol
SCCP	Signalling Connection Control Part
SCP	Service Control Point
SCTP	Stream Control Transmission Protocol
SDP	Session Description Protocol
SG	Signalling Gateway
SIP	Session Initiation Protocol
SNMP	Simple Network Management Protocol
SS7	Signalling System No.7
TCAP	Transaction Capabilities Application Part
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TGCP	Trunking Gateway Control Protocol
TOS	Type of Service
UDP	User Datagram Protocol
OA&M	Operation, Administration and Maintenance

4 Overview and purpose

The present document describes the design and specification requirements for a sub-set of the ITU-T IPCablecom documents. These requirements have been obtained through reverse engineering of the applicable documents, i.e. the General Architecture (J.arch or J.160 [1], the IPCablecom Signalling Transport Protocol (J.istp or J.165 [2]), and the Trunking Gateway Control Protocol (J.tgcp or J.171 [3]). These documents form a sub-set of the complete set describing the IPCablecom architecture. They have been derived from the PacketCable specification, version 1.0, as completed by CableLabs in 1999. The complete IPCablecom set of specifications consist of the following documents, describing:

- Architecture Framework (J.arch).
- Audio/video Codecs (J.161).
- Dynamic Quality-of-Service (J.dqos).
- Network-Based Call Signalling (NCS) (J.162).
- Event Messages (J.em).

- Internet Signalling Transport Protocol (J.istp).
- MIB Framework (J.mibfrw).
- MTA MIB (J.mtamib).
- NCS MTA MIB (J.ncsmib).
- MTA Device Provisioning (J.mtadpv).
- Security (J.sec).
- PSTN Gateway Call Signalling (J.tgcp).

The requirements obtained from the subset of documents are described in the clauses following this one. This clause contains a further explanation of the basics of the IPCablecom framework.

4.1 History and relationships to other standards

This historical background is given for information only. The IPCablecom architecture is stand alone, and can be considered, with its references, complete.

Historically, it may be of interest to note that PacketCable set of specifications is a complete design of a decomposed set of components, focused on Northern America and the requirements of North American operators. The first version of the PacketCable set of specifications were finalized in 1999 as PacketCable 1.0, though later version have been created. The ITU versions of the PacketCable specifications have been submitted in 2000. The main differences between the originals and the ITU versions are that the Northern American references have been moved to the appendices, and that ITU recommendations have been used the normative reference. Within the ITU, Study Group 9 works on recommendations for the use of cable and hybrid networks primarily designed for multimedia internet, voice, television and sound program delivery to the home, using integrated broadband networks to data and time-critical services. ETSI is now in the process of formulating its own versions of the specifications, using ETSI references.

Other project are concurrently defining alternate architectures, and the ITU, in the Y series of recommendations, has begun some work on harmonization of these. However, as with wireless (3GPP) and ATM networks, market imperatives require that work continue in these areas without waiting for a universal interoperable harmonization of all multi-media networks, protocols, and systems.

One exception is the IETF, which has built over the years a number of protocols that are extensively used in IPCablecom. Interfaces to the PSTN are based on ITU recommendations, as well as some voice over IP recommendations (such as ITU-T Recommendation G.711 [5]). Other interfaces (such as ANSI, ETSI, IEEE etc.) are considered to be, annexes, appendices, or supplements to the main body of the documentation.

It should also be noted that cable, with broadcast and interactive video, supports a number of features that cannot or are not envisioned by alternate architectures, and thus has different requirements.

4.2 ITU-T IPCablecom framework

The objective of the IPCablecom Architecture Framework document (J.arch) is to provide a high-level reference framework that identifies the functional components and defines the interfaces necessary to implement the capabilities detailed in the individual IPCablecom specifications as listed above. It is not a complete requirement specification for voice or other services, but rather a "framework" for enabling the development and testing for certification of interoperable components, as well as allowed options on implantations; many operator requirements as well as the options are deliberately left open to vendor, operators, subscriber, and market negotiation.

The ITU-T IPCablecom SG9 work was aimed at defining interface specifications that can be used to develop interoperable equipment capable of providing packet-based voice, video, and other high-speed multimedia services over hybrid fibre coax cable systems, utilizing the DOCSIS protocol. Within ITU-T Recommendation J.112, the DOCSIS-1.1 protocol carries the layer 2 services, while the Network Control Signalling protocol, known as ITU-T Recommendation J.162 [6] carries voice signalling and other information to provide quality, secure, call services.

Note there the are other proposed framework architectures for packetized voice on other media based system: cell based (3GPP, voATM, etc.), IP based on open IP networks, (VON, IETF, H.323, TIPHON, etc.), and IP based on closed networks (voDSL, and IPCablecom itself); the IPCablecom architecture was designed to meet the commercial requirements of cable operators, users, and vendors, which a phased deployment to meet time to market needs as well, and to allow future multi-media and interactive video services to be rapidly deployed.

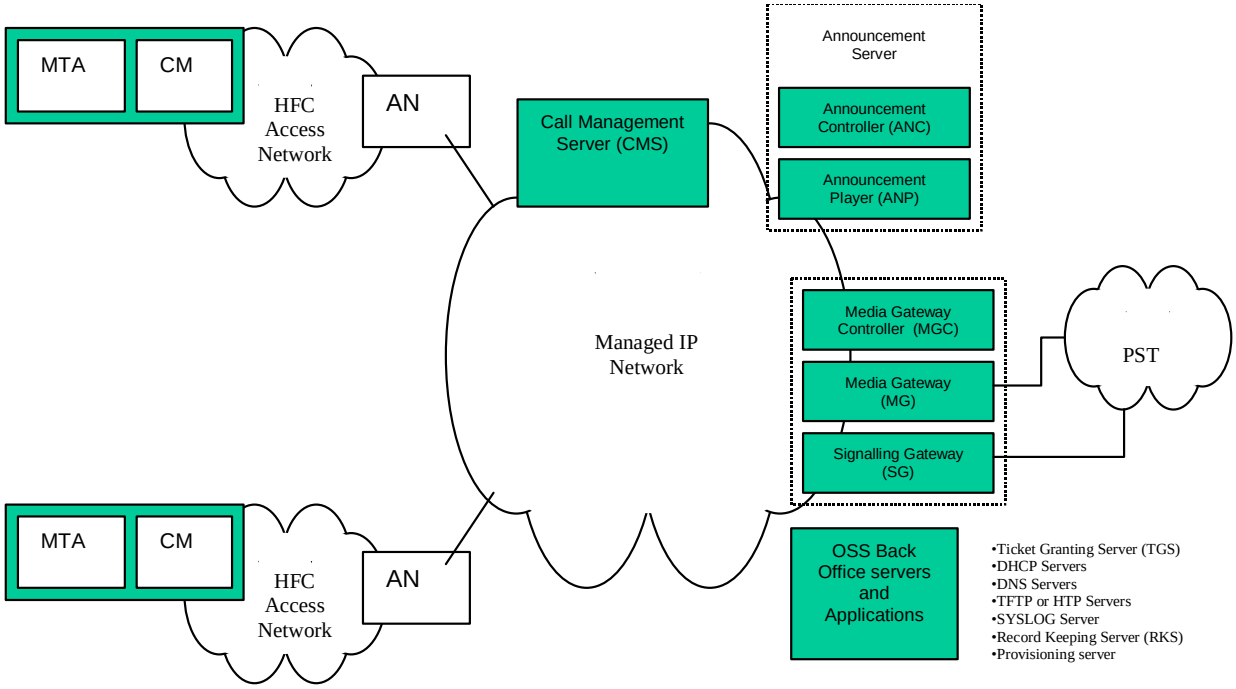


Figure 1: IPCablecom reference architecture

4.2.1 IPCablecom reference architecture

At a high level, the IPCablecom architecture references three networks:

- The J.112/J.162 HFC Access Network.
- The Managed IP Network.
- The PSTN.

A general overview of the IPCablecom architecture with most of its components is shown in figure 1. The access node (AN) provides connectivity between the J.112 HFC Access Network, and the Managed IP Network. Both the Signalling Gateway (SG) and the Media Gateway (MG) provide connectivity between the Managed IP Network and the PSTN. Other than an interface for Lawful Intercept (LI) to a Law Enforcement Agency, all the remaining components are considered internal.

The J.112 HFC Access Network provides high-speed, reliable, and secure transport between the customer premise and the cable head-end. This access network may provide all J.112 and J.162 capabilities including Quality-of-Service. The J.112 access network includes the following functional components: the Cable Modem (CM), Multimedia Terminal Adapter (MTA), and the Access Node (AN).

The Managed IP Network serves several functions. First, it provides interconnection between the basic IPCablecom functional components responsible for signalling, media, event collection (billing), operations, maintenance, administration, provisioning, and quality-of-service establishment. In addition, the managed IP network provides long-haul IP connectivity between other Managed IP and J.112 HFC networks. The Managed IP network includes the following functional components: Call Management Server (CMS), Announcement Server (ANS), several Operational Support System (OSS) back-office servers, Signalling Gateway (SG), Media Gateway (MG), and Media Gateway Controller (MGC).

The PSTN provides access to the public network for off-network calls. It can also be considered as a (very large) gateway to other networks, such as wireless, DSL, ATM, and VOIP. In addition the SS7 interface supports on-network calls as well for numbering queries and other information, and using PSTN based media servers is also acceptable as an option.

4.2.2 Interfaces

Between the networks, a number of interfaces can be distinguished. These interfaces are:

- NCS over J.112 between the managed IP network and the subscriber. Since this is an open IP interface, is not considered "trusted" and operations initiated from the HFC must be validated, authorized, authenticated, and made secure.
- SS7 signalling and trunks between Managed IP and PSTN. This interface can support MF in-band, but on to support "operator assisted" backup systems; this may not be a requirement in all markets, particularly in Europe. The PSTN is considered a "trusted" network out to, but not including the subscriber.
- Lawful intercept between the Managed IP Network and the "Law Enforcement Agency" (LEA). The LEA is (of course) considered trusted, but requires in some countries a court order to allow the interface.

The specific location of these interfaces has been shown in figure 2.

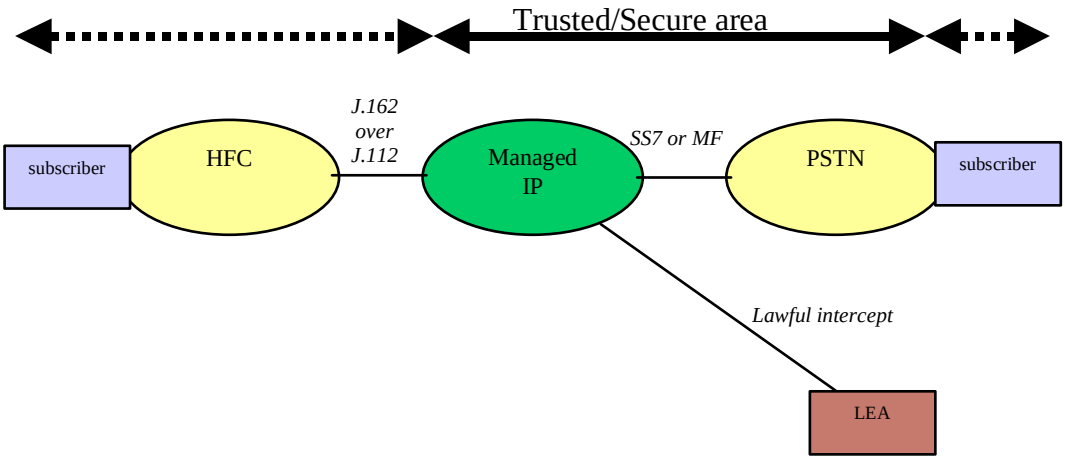


Figure 2: IPCablecom framework interfaces and external view

4.2.3 Bundling and unbundling of components

The architecture with the several components shown in figure 1 does not specifically require any bundling or unbundling of components. The components shown are logical components, and a vendor **may** or **may not** bundle any managed IP component with another. As a result, "TGCP" and "ISTP" can be considered to be "internal" interfaces, or can be requested to be interoperable and be open, based on the operators" RFI requests. In the future, the SIP based interface between CMS and CMS could also be considered an open interoperable interface based on RFI requirements. In fact, some phases of the architecture (LCS, DCS) bundle the components almost entirely into a local exchange or subscriber set top/PC.

4.2.4 IPCablecom zones and domains

Within the IPCablecom architecture, zones and domains have been defined to support various operation business and network deployment options:

- A zone is the set of MTA's controlled by a single CMS.
- Domains are one or more zones managed by an administrative entity, sharing a common Managed IP network.

The following requirements and remarks can be observed about zones and domains:

- Operators **may** or **may not** support zones and domains.
- Interfaces between zones have not been defined yet.
- Many small companies may wish to co-operate and share equipment costs, while still retaining subscriber ownership.

5 Requirements obtained from J.arch

5.1 Overview

5.1.1 Scope

This clause describes the requirements obtained from reverse engineering the ITU-T J.arch document (J.160 [1]). The source document used has been written by the Study Group 9 of the ITU-T. The title of this file is "Draft New Recommendation J.160 (J.arch): Architectural framework for the delivery of time critical services over cable television networks using cable modems", as of December 2000 [1]. The present document has been downloaded from the ITU-T web site and is not publicly accessible.

Note that J.arch will be used instead of J.160 [1] since it is easier for the reader.

5.1.2 What is J.arch?

J.arch is a globalized ITU version of the PacketCable architecture specification. The commercial and regulatory issues have been removed from it, though, as compared to the original document. Furthermore, ITU references are used instead of North American ones, which have been moved to annexes, appendices, or supplements as is proper in ITU recommendations. Other country specific references will be added in the future, also as annexes, appendices, or supplements.

J.arch is a document that describes the cable equivalent of a central office, but with some additions and some and missing features. The main differences is the interface to the set-top unit, which is a sort of web-like, text based multi-media HFC interface called J.162 or J.ncs, instead of an analogue. In addition to the normal CO functions, J.arch also defines:

- Lawful Intercept (though there is a trend now to define this as a CO mandated requirement.
- Security/privacy/fraud prevention (not usually an issue internal to a CO with closed signalling interfaces and networks).
- Quality-of-Service (a solved issue in TDM based networks).

However, it is missing some capabilities:

- Operations, Administration, and Maintenance (OA&M) is not strongly defined, except the requirement for SNMP and MIBs provision, which is defined in another document, not covered here. Billing is not defined, though support for billing is covered in the Event Messages document, also not covered here.
- ISDN and other legacy network interfaces, and many complex call features are not supported as they are not deemed necessary by operators.
- Business services are considered to be future work; the document is focused on home subscribers.

Also contained by the J.arch is the IPCablecom architecture, that has been outlined in the previous clause. For reasons of commonality with the J.istp and J.tgcp, this architecture has specifically been discussed in the overview clause.

5.1.3 Three architectures, possible three phases

Operators can choose to deploy one or more of the phased in architectures

- Line Control Signalling (LCS) supporting GR-303 or V5.2 remote digital terminal interface; this is mostly for operators who have local/central office switches or must rollout today. Most IPCablecom functions are bundled in the local switch.
- Network Control Signalling (NCS), which is also the name of the protocol to the set top box (STB). This moves features into a set of components which optionally may be bundled or physically distributed. It is overlapping in terminology, but different in significant ways from the IETF and soft-switch architectures.
- Distributed call signalling (DCS), which moves most features into the STB, using a SIP like architecture.

J.arch generally will evolve to support all three protocols, but in the first release of IPCablecom it focuses on the NCS signalling architecture.

5.2 Commercial requirements for IPCablecom

5.2.1 Implicit commercial requirements

The following commercial requirements are implicit in the IPCablecom architecture, and may have had more impact on the design of the framework than even explicit requirements:

- The architecture **must** support the time to market business considerations cable operators for deploying packet based services, including the bundled use of existing systems and their phased replacement in the future as newer technology becomes available.
- The architecture **should** evolve to meet operator business requirements and to accommodate the advances resulting from the maturing of IP based technology, including easy upgraded to new IP based standards, higher density equipment, and higher bandwidth and media capability to subscribers" homes.
- The architecture **must** support the deployment of time critical services over a cable network. This **may** include future video based services and expansion into business subscribers.
- The system **should** allow cable operators to compete successfully with other voice access technologies and other voice company services, such as narrowband wired-line and wireless phone or broadband DSL, maximizing the use of cable networks where such use can save money, and leverage the video capability and other cable offerings, while providing equivalent or superior voice services to present a end to end bundled offering to subscribers.
- The system **must** allow the start phased deployment in 2001 for NA systems; it **should** be able to start deployment in Europe in 2002.

5.2.2 Legal/Regulator observations

- Legal/Regulatory issues for voice over IP and cable voice services are not settled yet in law or regulation agencies.
- These issues **may** be different for voice over IP over cable than for phone companies.
- No particular regulatory issues are assumed by the use of PSTN or voice over IP.

5.3 Technical requirements

5.3.1 Architecture goals of J.arch

5.3.1.1 General architecture goals

- Voice quality **must** be comparable to or better than the PSTN as perceived by the end-user.
- The architecture **must** be scalable, and capable of supporting up to millions of subscribers.
- One-way delay for local IP access and IP egress (i.e. excluding the IP backbone network) **must** meet the delay requirements for all IPCablecom real-time services, including voice.
- Packet loss rate, jitter, and latency (delay) for the Managed IP Network **must** meet the requirements for all IPCablecom real-time services, including voice.
- It **must** support primary and/or secondary or multi-line residential voice communications capabilities.
- It **must** leverage existing standards where they meet IPCablecom goals. IPCablecom strives to specify open, approved industry standards that have been widely adopted in commercial communication networks. This includes standards approved by the ITU, IETF, IEEE, and other communications standards organizations.
- It **must** leverage and build upon the data transport and Quality of Service capabilities enabled by the J.112 infrastructure.
- It **must** allow multiple vendors to rapidly develop low-cost interoperable solutions to meet time-to-market requirements.
- The probability of blocking a call **must** be engineered to meet service provider's requirements.
- Cut-offs and call defects **must** be engineered to be less than 1 per 10 000 completed calls.
- It **must** support modems (up to V.90 speeds) and fax (up to 14,4 kbit/s).
- Frame slips due to unsynchronized sampling clocks or due to lost packets **must** occur less than 0,25 per minute

5.3.1.2 Call signalling requirements

- It **must** define a network-based signalling paradigm that allows operator management of set top terminals, with the intelligence for features mostly residing in the operators equipment and managed IP network: this is called Network Control Signalling (NCS). It is somewhat similar to the IETF and soft-switch approaches, but those architectures do not meet many of the goals of the cable operators, or support full-range multimedia (interactive broadcast video). This requirement also distinguishes it from the LCS and DCS architectures.
- It **must** provide end-to-end call signalling for the following call models:
  - calls that originate from the PSTN and terminate on the cable network;
  - calls that originate on the cable network and terminate on the cable network within a single IPCablecom zone;
  - calls that originate from the cable network and terminate on the PSTN;
  - calls that originate within one IPCablecom zone and terminate in another IPCablecom zone are for further study;
  - calls that originate on the PSTN, transit the IPCablecom network, and terminate on the PSTN are not specifically considered in this architecture.
- It **must** support the direct dial any domestic or international telephone number (ITU-T Recommendation E.164 [7] address).
- It **must** receive a call from any domestic or international telephone number supported by the PSTN.

- It **must** ensure that a new subscriber is able to retain current phone number via Local Number Portability (LNP).
- It **must** give the subscriber the ability to use the carrier of choice for long distance calls. This includes pre-subscription and per call selection.
- It **must** support Call Blocking/Call Blocking Toll restrictions (e.g. blocking calls to specific prefixes).

5.3.1.3 Call features

This clause lists the call features the architecture **must** support. Though the architecture **must** support these features, an operator **may** chose to support them or not, and a subscriber may be offered a choice of the features based upon the operators marking strategy.

While this is a first cut feature set, it is expected and planned that many new feature, including those that rely on broadcast video, will be developed and deployed in the future.

The terms given here are generic, and may be called by other names in other markets, or branded by operators.

- Call Waiting.
- Cancel Call Waiting.
- Call Forwarding (no-answer, busy, variable).
- Three-way Calling.
- Voice mail Message Waiting Indicator.
- Calling Number Delivery.
- Calling Name Delivery.
- Calling Identity Delivery On Call Waiting.
- Calling Identity Delivery Blocking.
- Anonymous Call Rejection.
- Automatic Call back.
- Automatic Recall.
- Distinctive Ringing/Call Waiting.
- Customer Originated Trace.

5.3.1.4 Quality of Service (QoS) requirements

With respect to Quality of Service (QoS), the following requirements can be listed for the architecture:

- It **must** provide set of policy mechanisms to provide and manage QoS for IPCablecom services over the access network.
- It **must** provide priority admission control mechanisms for both upstream and downstream directions.
- It **must** allow dynamic changes in QoS in the middle of IPCablecom calls.
- It **must** enable transparent access to all of the QoS mechanisms defined in J.112. IPCablecom clients need not be aware of specific J.112 QoS primitives and parameters.
- It **must** minimize and prevent abusive QoS usage including theft-of and denial-of service attacks. Ensure QoS policy is set and enforced by trusted IPCablecom network elements.
- It **must** provide a priority mechanism for emergency and other priority based signalling services.

5.3.1.5 Codec and media stream requirements

With respect to the codec and media stream requirements, the following requirements can be listed:

- It **must** minimize the effects that delay, packet-loss, and jitter have on voice-quality in the IP telephony environment.
- It **must** define a minimum set of audio codecs that must be supported on all IPCablecom endpoint devices (MTAs). Evaluation criteria for mandatory codecs have selected those most efficient with respect to voice quality, bandwidth utilization, and implementation complexity.
- It **must** accommodate evolving narrow-band and wide-band codec technologies.
- It **must** specify echo cancellation and voice activity detection mechanisms.
- It **must** support transparent, error-free DTMF transmission and detection.
- It **must** support terminal devices for the deaf and hearing impaired.
- It **must** provide mechanisms for codec switching when fax and modem services are required.

5.3.1.6 Device provisioning and OSS requirements

With respect to device provisioning and the OSS, the following requirements can be listed:

- It **should** support dynamic and static provisioning of customer premise equipment (MTA and CM).
- Normal provisioning changes **should not** require a reboot of the MTA.
- It **should** allow dynamic assignment and management of IP addresses for subscriber devices.
- It **should** ensure that real-time provisioning and configuration of MTA software does not adversely affect subscriber service.
- It **should** define SNMP MIBs for managing customer premise equipment (MTA).

5.3.1.7 Security requirements

The following security requirements can be listed:

In the context of IPCablecom, security is not a good word to use, as it has been used to mean Lawful Intercept (as in national security) as well as subscriber and operator security and privacy. However, till now, no one has come up with a better word, so the context must distinguish between the lawful Electronic Surveillance supports for "National Security" monitoring and intercept, and unlawful Fraud, Attack, and Intercept.

This area is a major difference from the PSTN. Since IP is an open protocol, and there are many energetic people who use it, there are several sorts of security issues:

- Privacy - the encryption and protection of subscribers against unlawful intercept.
- Fraud-prevention - the protection of operator revenue against attempt to gain access to unbilled services.
- Virus/traffic disruption - the protection of subscribers and operators against malicious vandalism of both the operation of the network and network data repositories.

As such, the following security requirements can be listed for IPCablecom:

- It **must** support residential voice capabilities with the same or higher level of perceived privacy as in the PSTN.
- It **must** provide protection against attacks on the MTA.
- It **must** protect the cable operator from various denial of service, network disruption and theft of service attacks.
- Design considerations **should** include confidentiality, authentication, integrity, non-repudiation and access control.

5.3.1.8 Managed IP network goal

The managed IP network **must** be able to assume bounds on the QoS performance parameters, e.g. packet loss, for packets traversing the network.

5.3.2 Requirements from components

A question in general can be: are there any component level requirements? The J.arch document contains the following phrase:

> "This clause describes the functional components present in an IPCablecom network. Component descriptions are not intended to define or imply product implementation requirements but instead to describe the functional role of each of these components in the reference architecture."

As such, the answer to the question in general is **yes**, simply because there are functional requirements identified in the components that are not explicitly identified in the architecture section. In the present document, these requirements will be called "functional requirements".

With respect to the components in an IPCablecom network, the following can be mentioned:

- Specific product implementations **may** combine or bundle functional components as an allowed part of the architecture.
- All components **may not** be present in an IPCablecom Network.

5.3.2.1 Trust

The motto here is the famous "trust, but verify". Trust means that information is sent over channels that are assumed to be secure. In the PSTN this is gained more by technology than intent, as channels, particularly digital channels, are difficult to tap or intercept for bearer channels, and SS7 is very difficult to intercept without expensive equipment.

With IP it is another story, and a lower layer security with encryption will likely be a native part of the Managed IP network. As such, with respect to Trust, the following requirements can be listed:

- Set top (MTA and CM) are "untrusted" and **must** be verified.
- Managed IP network components are trusted but **must** have security.

NOTE: PSTN is "implicitly" trusted, but there is no mention of it explicitly in the document.

5.3.2.2 Subscriber side requirements

- IPCablecom MTA **must** support the J.112 Network Call Signalling (NCS) protocol.
- An embedded MTA (E-MTA) is a single hardware device that incorporates a cable modem as well as an IPCablecom MTA component.
- IPCablecom specifications currently only **must** support embedded MTAs; "stand alone" MTAs without CMs are thus not necessarily supported. The term MTA in the document means "Embedded MTA".

Note that NCS means two things:

- The protocol to the set top from the Managed IP network.
- The described network based architecture.

This is actually a market issue, to cover the fact that cable modems are currently being deployed without phone interfaces. The first roll out of cable voice will require new set top boxes; in the future, it may be possible to have a non embedded MTA, but there are serious issues with operator management of such devices that are deferred for the future.

5.3.2.3 MTA Functional requirements

The MTA functional requirements listed here are for information only, as the J.arch document does not cover the MTA requirements. But it is very necessary to understand what the MTA expects when trying to understand the architecture.

Note that this is not a comprehensive MTA list; a large number of requirements and features and options are not listed here, including local tones and announcements, software replacement, etc. Taking this into account, the following functional requirements can be listed:

- It **must** support J.112 (j.ncs) NCS call signalling with the CMS.
- It **must** support QoS signalling with the CMS and the AN.
- It **must** ensure authentication, confidentiality and integrity of important messages between the MTA and other IPCablecom network elements.
- It **must** map media streams to the MAC services of the J.112 access network.
- It **must** encode/decode media streams.
- It **must** provide multiple audio indicators to phones, such as ringing tones, call-waiting tones, stutter dial tone, dial tone, etc.
- It **must** support standard PSTN analogue line signalling for audio tones, voice transport, caller-id signalling, DTMF, and message waiting indicators.
- It **must** support the G.711 audio Codec; note that other Codecs are future TBD.
- It **must** support one or more analogue lines and may support one or more ISDN BRI interface(s); note that ISDN may be a market specific (European) requirement, as it goes against the commercial interests of the operators and subscribers in other markets.

5.3.2.4 Access Node (AN) MTA functional requirements

These are the requirements on IPCablecom from the Access Node:

- It **must** support NCS/J.112/HFC based MTA subscriber access to the Managed IP network, including:
  - It **must** provide required QoS to the CM based upon policy configuration.
  - It **must** allocate upstream bandwidth in accordance to CM requests and network QoS policies.
  - It **must** classify each arriving packet from the network side interface and assigning it to a QoS level based on defined filter specifications.
  - It **must** police the TOS field (Type of Services, also known as Diff. Serv.) in received packets from the cable network to enforce TOS field settings per network operator policy.
  - It **must** alter the TOS field in the downstream IP headers based on the network operator's policy.
  - It **must** perform traffic shaping and policing as required by the flow specification.
  - It **must** forward downstream packets to the J.112 network using the assigned QoS.
  - It **must** forward upstream packets to the backbone network devices using the assigned QoS.
  - It **must** convert and classify QoS Gate parameters into J.112 QoS parameters.

5.3.2.5 Access Node (AN) MTA managed IP functional requirements

- It **must** support in the Managed IP network:
  - It **must** support the reservation of any backbone QoS and bandwidth necessary to complete the service reservation.
  - It **must** record usage of resources per call using IPCablecom Event Messages.

5.3.2.6 CMS Call agent functions

Call Agent functions are responsible for providing signalling services using the NCS protocol to the MTA. This is often called "line side" call processing, and includes the functions of:

- It **must** implement call features, and manage subscriber access to other subscribers possibly over the network.
- It **must** maintain call progress state.
- It **must** manage the use of Codecs within the subscriber MTA device.
- It **must** collect and pre-process dialled digits.
- It **must** collect and classify user actions.

5.3.2.7 CMS Announcement controller functions

These are the requirements on the CMS for announcement services. The announcement server may optionally be network based using SS7 TCAP messages (possibly based on INAP). However, if it is an internal component, using IP RTP streams, then it must provide the features as listed below. In the future, it may provide video announcements as well.

- It **must** support call management and enhanced features.
- It **must** support Directory Services and Address translation.
- It **must** provide Call routing.
- It **must** record usage of local number portability services.
- It **must** support QoS admission control.

NOTE: Zone-to-zone call signalling is for further study.

5.3.2.8 Implicit CMS goal

"By centralising call state and service processing in the CMS, the service provider is in a position to centrally manage the reliability of the service provided. In addition the service provider gains full access to all software and hardware in the event that a defect that impacts subscriber services occurs. Software can be centrally controlled, and updated in quick debugging and resolution cycles that do not require deployment of field personnel to the customer premise. Additionally, the service provider has direct control over the services introduced and the associated revenue streams associated with such services."

This is deliberately quoted and not listed as a requirement, since such a goal is an unsolved problem in complex concurrent and cooperative systems.

5.3.2.9 MGC requirements

This clause only contains a very small part of the MGC requirements. The MGC has the major responsibility of handling network signalling, which is covered in TGCP and ISTP in much more detail. Both the TGCP and ISTP are listed in following clauses of the present document.

The following requirements can be listed here:

- It **must** receive and mediate call-signalling information between IPCablecom network and the PSTN.
- The MGC controls the MG and **must** instruct it to create, modify, or delete PSTN connections.

5.3.2.10 MGC functions

This overview of MGC functions is only included for information to aid the understanding of the MGC functions. As with the MGC requirements, not all requirements are list here, and many more are in TGCP.

In reality, the real purpose of the MGC is to support a 'global' view of trunks to the PSTN, providing a single (redundant) point of contact for all types of trunks, and supporting recovery actions on trunks that go in and out of service, and to support the OA&M functions of trunks as well (operations, administration, and maintenance, including congestion management, monitoring of traffic, etc.).

- It **must** support "trunk side" call control functions.
- It **must** support IPCablecom signalling.
- It **must** control the MG.
- It **must** support external resource monitoring.
- It **must** support call routing.
- It **must** support security.
- It **must** support usage recording via event messages.

5.3.2.11 MG requirements

The MG requirements are:

- It **must** provide bearer channel and may provide in-band signalling connectivity to the PSTN.
- It **must** implement all the call state and intelligence and controls the operation of the Media Gateway through the TGCP protocol: this includes creation, modification and deletion of connections as well as in-band signalling information to and from the MG. TGCP is an extended variant of the IETF's MGCP call signalling protocol, and thus aligned with NCS.

5.3.2.12 MG functions

The following is a list of functions performed by the Media Gateway:

- It **must** terminate and control physical circuits in the form of bearer channels from the PSTN.
- It **must** discriminate between media and Channel Associated In-band signalling information from the PSTN circuit.
- It **must** detect events on endpoints and connections as requested by the MGC. This includes events needed to support in-band signalling, e.g., MF.
- It **must** generate signals on endpoints and connections, e.g., continuity tests, alerting, etc. as instructed by the MGC. This includes signals needed to support in-band signalling.
- It **must** create, modify, and delete connections to and from other endpoints as instructed by the MGC.
- It **must** controls and assigns internal media processing resources to specific connections upon receipt of a general request from the Media Gateway Controller.
- It **must** perform media trans-coding between the PSTN and the IPCablecom network. This includes all aspect of the trans-coding such as Codecs, echo cancellation, etc.

- It **must** ensure that any entity communicating with the MG adheres to the security requirements.
- It **must** determine usage of relevant resources and attributes associated with those resources, e.g., number of media bytes sent and received.
- It **must** track and report usage of resources to the MGC.

5.3.2.13 SS7 signalling gateway functions

These are the SS7 signalling gateway functions. More information on this is to be found in the ISTP.

- It **must** terminate physical SS7 signalling links.
- It **must** implement security features.
- It **must** terminate Message Transfer Part level 1, 2, 3.
- It **must** implement MTP network management functions.
- It **must** support ISUP address mapping.
- It **must** support TCAP address mapping.
- It **must** provide mechanism for trusted entities.
- It **must** implement transport protocols required to transport signalling information between the signalling gateway and the MGC or CMS.

5.3.2.14 PSTN signalling requirements

- It **must** support SS7 ISUP/TCAP/SCCP/MTP3, 2, 1:
  - The signalling gateway function only supports non-facility associated signalling in the form of SS7 (SS7).
  - ISUP and TCAP are both required, but not INAP or AIN (the MGC could use these but they are not mentioned in the Architecture).
  - It supports initialization, address mapping from the SS7 domain to the IP domain, message delivery for SS7 ISUP and TCAP, congestion management, fault management, maintenance operations; and redundant configuration support.
- It **must** support in band signalling in the form of MF by the MG function for operator assisted/911 type calls.

NOTE: A non-requirement- ISDN PRI to the network **should not** be supported.

5.3.2.15 OSS requirements

With respect to IP/IETF servers, the following requirements can be listed:

- The Ticket Granting Server **must** use a Kerberos server. A ticket contains information used to set up authentication, privacy, integrity and access control for the call signalling between the MTA and the CMS.
- It **must** use a Dynamic Host Configuration Protocol Server (DHCP) to dynamically allocate IP address.
- It **must** support a Domain Name System Server (DNS) to map ASCII domain names to IP addresses using DNS protocols.
- It **must** support either a Trivial File Transfer Protocol Server and a HyperText Transfer Protocol Server (TFTP or HTTP) to download configuration files. These servers are BackOffice network elements used during the MTA device provisioning process to download configuration files to the MTA.

With respect to OA&M, the following requirements can be listed:

- System errors and traps **must** be maintained on a system error log server.
- Event Messages from trusted IPCablecom network elements such as the CMS, AN, and MGC **must** be maintained in a record keeping server (RKS). The RKS **may** assemble the Event Messages into coherent sets or Call Detail Records (CDRs), which can then made available to other back office systems such as billing, fraud detection, and other systems.
- Informational tones and messages **must** be played in response to events that occur in the network.

NOTE: OSS requirements assume a modern UNIX and IP based technology, with "syslog", and a client server architecture. The assumption of client server models for non-real time functions in the architecture is so strong that it is not mentioned.

5.3.2.16 Media streams: RTP

- IPCablecom **must** use the IETF standard RTP (RFC 1899 [8] - Real-time Transport Protocol) to transport all media streams in the IPCablecom network.
- IPCablecom **must** use the RTP profile for audio and video streams as defined in IETF RFC 1890 [9].

5.3.2.17 Provisioning

- MTA provisioning:
  - It **must** provision the MTA's devices and endpoints: provisioning includes:
    - initialization;
    - authentication;
    - registration;
    - other configuration functions.
  - It **must** use SNMPv3 to interface the MTA to element management systems for MTA device provisioning.

NOTE: The SNMP v.3 managed object model is extensively defined and used in IPCablecom, but other documents define the managed objects in detail.

5.3.2.18 Event management

- It **must** generate and store data records containing information about network usage and activities as sets of event messages to provide a complete record of the service. This complete record of the service is often referred to as a Call Detail Record (CDR).
- It **must** offer Event Messages or CDRs may be sent to one or more back office applications such as a billing system, fraud detection system, or pre-paid services processor, or to external interfaces such as a law enforcement agency (LEA).
- It **must** use Remote Access Dial-In User Service (RADIUS) the transport protocol internal to the Managed IP network.
- The RKS **must** be the single point of contact for external services requiring Event information; these interfaces may be proprietary.

5.3.2.19 Quality of Service

5.3.2.19.1 Static QoS

- IPCablecom Quality of Service signalling to the MTA **may** be handled at the application layer (SDP parameters), network layer 3 (RSVP), or at the data-link layer 2 (J.112 QoS).

5.3.2.19.2 Dynamic QoS

- IPCablecom Dynamic QoS (D-QoS) **must** authorize resources at the time of the call to theft of service attack types: this includes:
  - **Maximum Allowed QoS Envelope** - The maximum allowed QoS envelope enumerates the maximum QoS resource (e.g., "2 grants of 160 bytes per 10 ms") the MTA is allowed to admit for a given media stream bearer flow. If the MTA requests a value greater than the parameters contained within the envelope the request will be denied.
  - **Identity of the media stream endpoints** - The GC/CMS authorizes the parties that are involved in a media stream bearer flow. Using this information the AN can police the data stream to ensure that the data stream is destined and originated from the parties that are authorized.
  - **Billing Information** - The GC/CMS creates opaque billing information that the AN does not have to decode. The information might be as simple as billing identity or the nature of the call. The AN forwards this billing information to the RKS as the call is activated or terminated.

5.3.2.20 Announcement options

- It **must** allow the MTA to play locally stored announcements to provide informative progress tones to the end user independently of the network state (e.g., congestion).
- It **must** support simple, fixed-content announcements (e.g., all-lines-busy) stored at the Media Gateway to provide announcements to PSTN users.
- It **must** use NCS with an announcement package to define the announcement interfaces.
- It **must** use RTP for the announcement media stream.
- The announcement server **may** be one bundled component, **may** be bundled into the Managed IP system, or **may** be two separate components: an announcement player (ANP) and an announcement controller (ANC).

5.3.2.21 Security: Privacy and fraud prevention

- It **must** address both subscriber and operator needs.
- It **must** specify security required for each external protocol interface.
- It **must** consider authentication, access control, integrity, confidentiality and non-repudiation.
- An IPCablecom protocol interface **may** employ zero, one or more of these services to address its particular security requirements.

5.3.2.22 Time of day requirements

- In order to maintain service quality all network equipment clocks **must** be maintained to within 200 ms of Universal Time Co-ordinated (UTC).
- It **may** use the Network Time Protocol (NTP); this is the "recommended" protocol for IPCablecom time synchronization.
- All systems that generate billing event messages **must** synchronize their clocks to a network clock source.
- Synchronization **must** be done to ensure that the reporting device's own clock remains within ±100 ms of the last synchronization value.

5.3.2.23 Clocks

- In order to minimize the overrun or under-run of play-out buffers due to the difference in clock speed between plesiochronous entities, all ANs **must** lock their downstream transmission rate to a clock derived from a source that reflects a Stratum-3 clock.
- MTAs **must** use the downstream transmission rate to derive the clock used to determine packetization period.
- MTAs **must** also use this clock to determine the rate of play-out from the receive buffer.

5.3.2.24 IP addressing

5.3.2.24.1 One CM and one MTA

- IPCablecom IP address **must** use IPv4:
  - All IPCablecom embedded MTAs **must** have two IP addresses: one for the CM and one for the MTA.
  - All IPCablecom embedded MTAs **must** have 2 MAC addresses: one for the CM and one for the MTA.
- It **must** support a private IP address for the CM host function in the case where NAT translation is not provided elsewhere in the IPCablecom network.
- IPCablecom operator **must** be able to route the voice service packets over a voice backbone and all other packets (data) over a data backbone.
- The IPCablecom operator **must** be able to separate or simplify network side administration and management functions using separate IP addresses to support such concepts as policy filters that block or permit traffic from the MTA component of the node, source address screening services, and network traffic statistics and diagnostics.

5.3.2.24.2 Dynamic IP addressing

- Since calls are based on a mapping of a subscriber's service to an endpoint identifier and an IP address, the operator **must** be able to configure the system to minimize changes in IP address during a call.

NOTE: Fully Qualified Domain Name (FQDN) issues relating to DHCP and DNS are not part of the scope of the j.arch, but, some recommendations on 'best practices' are outlined in IETF RFC 2131 [10].

5.3.2.25 Packet prioritization

- Prioritization for QoS of packets over the managed IP backbone is out of scope and considered implementation dependent.
- Prioritization for signalling packets is out of scope and considered implementation dependent:
  - TGCP address some of the issues;
  - ISTP uses SCTP to handle these issues.

5.3.2.26 Fax support

- IPCablecom **must** support real-time fax transmission.
- Fax transmission **must** use ITU-T Recommendation G.711 [5] for audio encoding/decoding.
- If a call is established using a compressed Codec, the embedded MTA **must** be instructed to look for fax tones. If fax tones are detected, the CMS **must** be notified and the MTA will be instructed to switch to using G.711. Note that this places a requirement on the embedded device to monitor the media stream and detect fax tones.

- It **must** support switching over to fax from a voice call, however, switching back to voice from fax is not required (i.e., monitoring the fax media stream for an ending signal and then switching back to a low bandwidth Codec).
- Local termination of fax and translating the fax stream to an IP fax relay data stream is not required in this version of the architecture.

5.3.2.27 Analogue modem support

- The MTA **must** detect modem tones and, when such tones are detected, the CMS **must** instruct the MTA to switch over to the G.711 Codec if it is not already in use. Note that this places a requirement on the embedded device to monitor the voice stream and to detect analogue modem tones.
- It **must** support switching over to G.711 to support analogue modem signalling from a voice call, however, switching back to voice from modem signalling will not be required to be supported (i.e., monitoring the modem media stream for an ending signal and then switching back to a low-bandwidth Codec).
- Local termination of modems and translating the modem stream to an IP modem relay data stream is not required in this version of the architecture.

6 Requirements obtained from J.istp

6.1 Overview

6.1.1 Scope

This clause describes the requirements obtained from reverse engineering the ITU-T J.istp document (J.165 [2]). The source document used has been written by the Study Group 9 of the ITU-T. The title of this file is "Revised Draft Recommendation J.165 (J.istp): IPCablecom signalling transport protocol", as of October 2001 [2]. The present document has been downloaded from the ITU-T web site.

"It" in this clause refers to "the SG and ISTP", and includes the ISTP stack that exits in the MGC, CMS or other potential IPCablecom components and elements.

6.1.2 What is J.istp?

J.istp is an ITU-T version of the Internet Signalling Transport Protocol (ISTP) released in 1999 by Cablelabs. For the moment, this ISTP has been adapted by the ITU in draft j.istp. Note that, unlike TGCP, there is no API defined in ISTP only a protocol, since most current SS7 vendors supply their own API for TCAP and ISUP, and there was not advantage in specifying an alternative standard. The following characteristics can be listed:

- ISTP Provides a complete 7 layer transport within the IPCablecom architecture, adhering to the framework and cable operator requirements for interfaces to the public SS7 based PSTN.
- ISTP uses as much of the IETF SIGTRAN work as was available in 1999 when the specification was released, and, is available as of November 2001.
- It is a secure, reliable, real-time way of transporting ISUP signalling messages and TCAP transaction messages over the Managed IP network to any element requiring SS7 interfaces.
- It is a set of protocol components, a "stack", running over SCTP (or TCP as an option) over IP in many elements of a Managed IP network.

Note that ISTP differs from the IETF SIGTRAN protocols in several areas:

- The framework is different.
- The recovery mechanisms meet stronger requirements.

- It is certifiable (can be tested for interoperability by third party bodies).
- It provides a complete layer 7 solution up to TCAP and ISUP; IETF provides M3UA and SUA as the highest levels, and provides lower level M2UA solutions no required by cable operators.

6.1.2.1 Note on SG and ISTP

A note on SG and ISTP: SG and ISTP are not the same thing:

- The signalling gateway (SG) is a logical component (a set of equipment) that forms the single point of interface for SS7 messages to the PSTN and contains all layers of the SS7 stack; the MG is the interface for bearer channels only.
- ISTP is a protocol implemented as a stack element running over IP and runs on many IPCablecom components.
- ISTP could be used in non-IPCablecom architecture.
- J.istp defines requirements on both ISTP the protocol and the SG as implemented in an IPCablecom network. Where the distinction matters it will be identified in the requirement; otherwise, "it" will mean "the SG and ISTP", and includes the ISTP stack portions of the MGC and CMS. There are other requirements on the MGC, CMS, and SG components not covered here.

6.1.2.2 Protocol distribution in IPCablecom elements

Figure 3 shows the protocol distribution in IPCablecom elements:

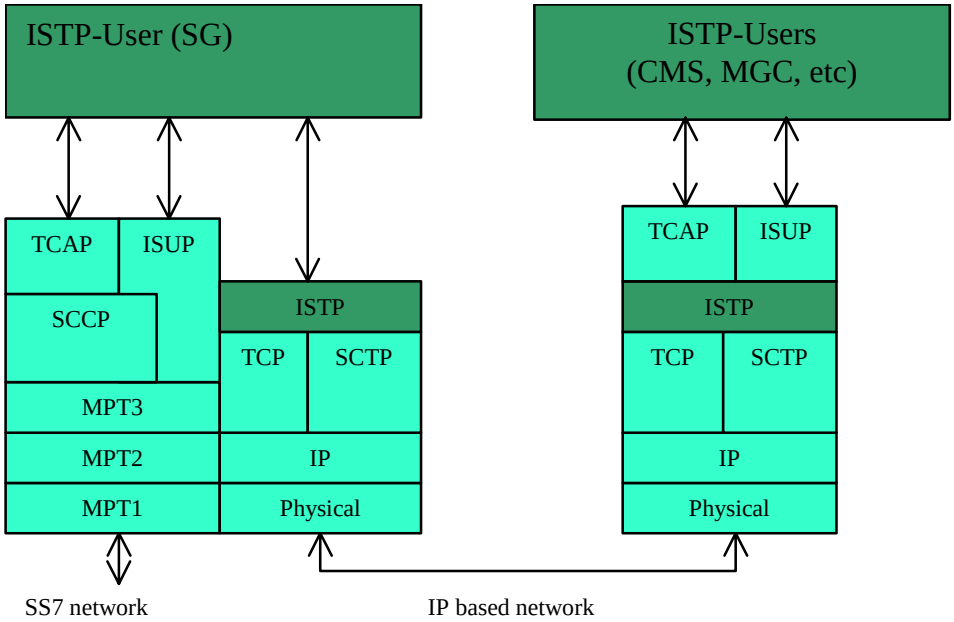


Figure 3: Protocol distribution in IPCablecom elements

6.1.2.3 General ISTP functions

The general ISTP functions are:

- Initialization.
- Registration of circuit Ids.
- Address mapping between SS7 and IP domains.
- ISUP maps based on point code and circuit identification code.
- TCAP maps based on point code and transaction ID.

- ISUP/TCAP message delivery using reliable transport.
- Maintenance operations.
- Activation/deactivation of circuit Ids with SS7 gateway.
- Error recovery (due to faults or congestion).
- SS7 signalling point or network inaccessible.
- MGC inaccessible or congested.
- CMS inaccessible or congested.
- Signalling point or link congested.

6.1.2.4 ISTP specification goals

- It **must** support cable companies' penetration into residential and business markets for multimedia services, including voice.
- It **must** support a low cost replacement strategy for PSTN switching, peripheral, and control equipment using IP based technology.
- It **must** support a network that can provide higher level features (such as multimedia) in addition to the PSTN features.
- It **must** support a transparent interface to the existing PSTN.
- It **may** use open architecture, that will support the interworking of multiple vendors' equipment in the same IPCablecom network.
- It **must** support scalable gateway architecture, allowing solutions ranging, for example, from the equivalent of a single T1 or E1 media gateway up to a system that is the equivalent of a large tandem switch supporting multiple central offices (about 40 000 trunks).
- It **must** provide an architecture that can achieve the same high degree of reliability and performance as the PSTN, while allowing for a "descoped" network (simplex connections) to support lower cost enterprise and customer premise implementations.

6.1.2.5 ISTP in decomposed IPCablecom gateway

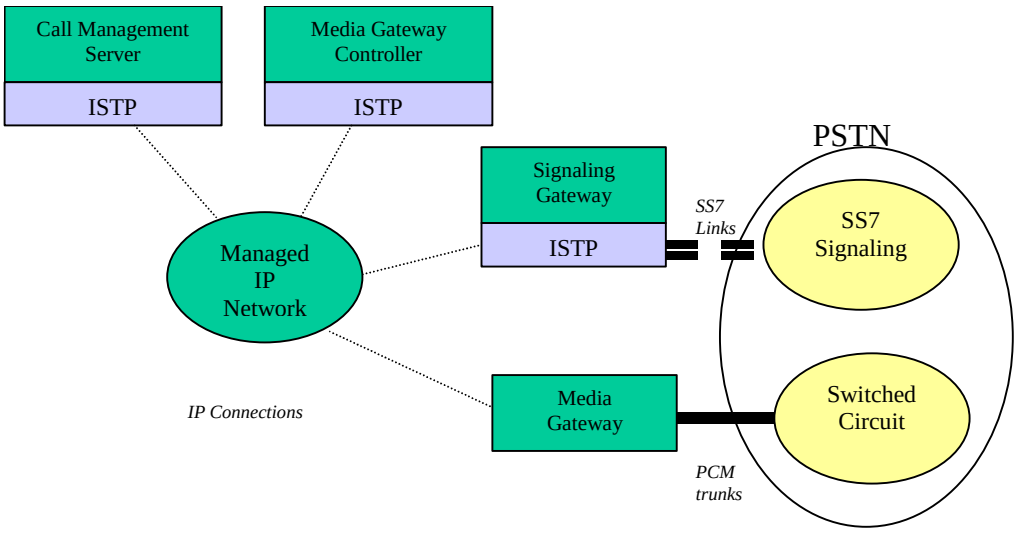


Figure 4: ISP in decomposed IPCablecom gateway

6.1.2.6 Areas beyond the scope of J.istp

- Address layer management (SNMP), security, and measurements covered in other IPCablecom Recommendations, but the SG must adhere to those requirements within the J.arch framework.
- Implementation and vendor dependent issues, such as performance, functional distribution, network configuration, etc.
- Details about CMS, MGC, and other media communication applications.

NOTE: "The specification in this annex is used in North America", means that the structure of the ISUP and TCAP is ANSI based, and some MTP3 messages in the ITU SS7 stack are not supported. Most SG implementations from vendors will implement SS7 ANSI, ITU, SS7, and a lot of variants as a matter of normal practice ISTP and IETF. In addition, later.

6.1.2.7 ISTP and IETF

- ISTP uses IP, DNS, and a lot of IETF RFCs.
- ISTP uses the SCTP for underlying transport.
- However, ISTP currently does not use other SIGTRAN protocols, since they are not RFC status.
- It has been proposed to use M3UA and possibly SUA.
- Even if M3UA and SUA were used, most of ISTP still exist to handle the many requirements for reliability, recovery, growth, software replacement, certification testing, and support for business models; the last clause of this presentation covers some areas that could be removed from ISTP if the IETF protocols were used.

6.2 J.istp: Technical requirements

6.2.1 J.istp: Framework and architecture requirements

6.2.1.1 High Level SG/ISTP requirements

- It **should** reduce the cost of deploying a SS7 based interface to the public network:
  - lower initial cost using standard interfaces with multi-vendor equipment selection; thus it **must** support certification and interoperability testing;
  - lower operation cost, using fewer point codes;
  - lower initial cost and operational cost using Managed IP network and low cost routers instead of many SS7 lines and STPs.
- It **should** allow multiple IPCablecom IP based components to share a single point code.
- It **should** allow multiple zones and domains and operators to share a SG with multiple point codes, assigning multiple point codes to a single SG.
- It **should** allow "piecewise" software replacement strategies, that allow upgrades of software without taking a IPCablecom network out of service.
- It **may** provide a system with a reliability equal to or higher than the PSTN; as a cost option, individual operators may choose to support lower reliability in some market, for example.

6.2.1.2 Support of J.arch framework service goals

ISTP and the SG **must** support J.arch/J.nsc service goals including:

- Voice or other media content conversion.
- Call control signalling.
- Quality of service control.
- Call control signalling interoperability with the existing public network.
- Media interfaces to the existing public network.
- Data transactions to public databases.
- Routing mechanisms.
- Billing.
- Operations and maintenance.
- Security.
- Privacy.

6.2.1.3 SG SS7 termination requirements

- The SG **must** appear as a normal SS7 endpoint to the connecting PSTN network.
- The SG **must** terminate physical SS7 signalling links from the PSTN (A, F links).
- The SG **must** ensure that the Gateway security is consistent with IPCablecom and SS7 network security requirements.
- The SG **must** terminate Message Transfer Part (MTP) level 1, 2 and 3.
- The SG **must** terminate SCCP, and retain global title translation tables within the Signalling gateway to support controlled update of point code translation tables.
- The SG **must** implement MTP network management functions (initialization, recovery, congestion management, and maintenance) as required for a SS7 signalling point.

6.2.1.4 Functional SG signalling requirements

- It **must** transport "raw" ISUP and TCAP messages to IPCablecom components in their native format; it may "normalize a variant SS7 message to a ITU or ANSI standard format.
- It **must** perform ISUP address mapping to support flexible mapping of Point Codes to the appropriate Media Gateway Controller (MGC). The addressed MGC will be responsible for controlling the Media Gateway, which terminates the corresponding trunks.
- It **must** perform TCAP Address Mapping to map Point Code/Global Title/SCCP Subsystem Number combinations within SS7 TCAP messages to the appropriate MGC or CMS.
- It **must** provide a mechanism for certain trusted elements (TCAP User components) within the IPCablecom network, such as Call Agents, to query external PSTN databases via TCAP messages sent over the SS7 network.
- It **must** not restrict ISUP or TCAP interfaces in the future; for example, SCP may be supported in the IPCablecom network, allowing external originated transaction queries to a IPCablecom database.

6.2.1.5 Functional SG interface requirements

- It **must** support the OSS back-office interfaces for SG configuration and capabilities.
- It **must** support SNMPv3 based MIBs and operation, administration and management of SG configuration and capabilities. The requirement for SNMP is a j.arch requirement: the description of SG related MIBs is out of the scope of the present document.
- It **must** support Managed IP network security, and maintain the "trusted" status of components within the network.
- It **must** support a highly reliable near real-time transport of messages within the Managed IP network.
- It may be bundled with the MGC and CMS components, and thus the ISTP may not be an open, certified interface as a operator option.

6.2.1.6 Architecture Zone/Domain goals

- A SG **may** appear as a single point code to the SS7 network, where it is viewed as a "signalling endpoint", or **may** appear as multiple point codes and endpoints, running on a single SG element.
- It **must** support multiple call models, vendor components, or versions in different IP based call agent components on the same network at the same time. For example:
  - it may support a CMS that handles a set of PBX "enterprise" features and one that handles a set of central office "home subscriber" features and have to route messages based on subscriber identity;
  - based on the target trunk group identity, an incoming call may be routed to a "home subscriber" MGC from vendor A, or a "home subscriber" MGC from vendor B, depending on who owns the trunk;
  - A SG **may** appear as a single point code to the SS7 network, where it is viewed as a "signalling endpoint", or **may** appear as multiple point codes and endpoints, running on a single SG element.

6.2.1.7 Reliable underlying transport

- It **must** support SS7 messages over IP on a managed IP network in a reliable and near real time fashion. It may use SCTP or TCP over a managed IP network. Note that it is the vendor and operator's responsibility to configure the selected stack and network to meet timing, reliability and security requirements for signalling.
- It **must** use explicit packet-oriented delivery (not stream-oriented).
- It **must** provide sequenced delivery of user messages within multiple streams, with an option for order-of-arrival delivery of individual user messages.
- It **may** support optional multiplexing of user messages into SCTP data-grams.
- It **must** support network-level fault tolerance through support of multi-homing.
- It **must** provide resistance to flooding and masquerade attacks.
- It **must** support data segmentation to conform to discovered path MTU size.

6.2.2 J.istp: Availability and Performance Requirements

NOTE: This clause is called "Network Reliability" in ISTP and is at the backend of the J.istp specification.

6.2.2.1 Distribution model Background

- Fundamental to understanding ISTP, but moved to back end of ITU spec from IPCablecom.
- *Elements*, also known as *IPCablecom components*, or *software components*, represent "clusters" of one to many computers sharing a common function (software code) and text-based domain address; note that the model allows multiple call agent types (such as one for different subscriber types) with different software loads, or different MGC, or other software component types.
- Nodes refer to a single computer in the managed IP network, with possibly multiple IP connections. A cluster of IP nodes with common software code is an element.

6.2.2.2 Distribution Model

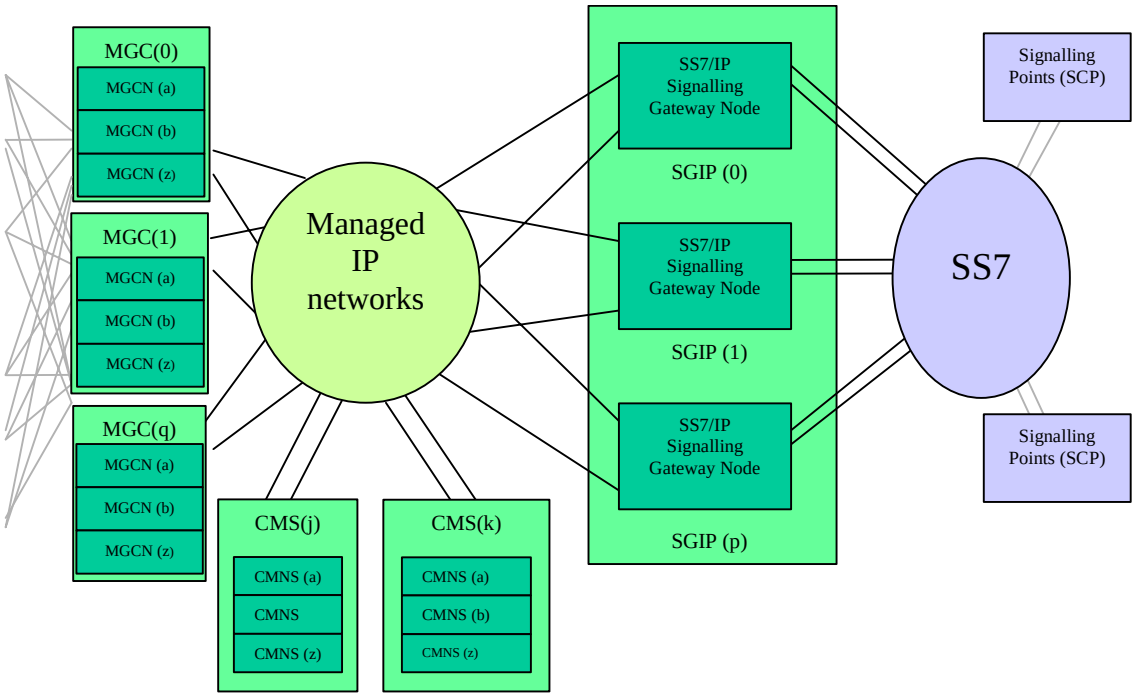


Figure 5: Distribution model

6.2.2.3 Availability

- The architecture model **must** support a network availability equal to that of the PSTN or higher (0,9999+) in a highly scalable fashion to allow for growth and replacement. Operators **may** implement lower availability targets in order to reduce costs in some markets.
- Meeting this availability objective **may** require service providers to implement several types of reliability and redundancy mechanisms in the network, such as:
  - redundant managed IP networks, with independent IP transport (WAN/LAN) and guaranteed delay and delivery times;
  - redundant independent network routers/local routers;
  - redundant connection, switching, and transport hardware;
  - n+k element node redundancy;

- no-single point of failure, including geographical power (geographical distribution).

6.2.2.4 Call Availability and Recovery

- Stable calls **must** be preserved over software replacement, and, where possible, **may** be preserved in the event of a failure.
- Calls in a set up or tear down state **may not** be preserved.
- Billing issues are not addressed in J.istp recovery.

6.2.2.5 Call Performance

- It **must** meet relevant ITU recommendations on the PSTN. This means a IPCablecom connection has the same performance requirements as a PSTN call. While the issue of performance is complex in a pure SS7 network, the mixture of IP and SS7, and the vendor-dependent breakdown of performance budget makes performance a difficult area to define precisely.
- It **must** meet user expectations of one to two seconds for set up on national communications.
- It **must** meet user exceptions of 2.5-5 seconds on international communications for total call set up.
- In order to meet these user expectations for communication set up, which consists of many messages and processes, each with their own delay budget, in many elements (as many as five) across the network, a single node **must**:
  - process critical SS7 ISUP events in under 50 ms;
  - process TCAP messages in less than 75 ms.
- This expectation for real-time transport of signalling messages across the networks of less than 50 ms delay mandates an underlying transport layer that **must** be reliable, real-time (less < 25 ms for ISUP and 75 ms for TCAP), and avoid duplicated, mis-sequenced or lost packets.

6.2.2.6 Traffic Performance

- The SG **should** scale up to a "reasonable size", which is one that does not compromise the reliability of the network by putting to many critical calls over a single component. The actual maximum size is determined by the operator based on their own network goals and implementation, so this area is left deliberately vague.
- SS7 scales by number of links and computer capacity; SS7 links should be dimensioned for no more than 40 % occupancy. As an example, with two SS7 links this is in the order of 65 000 calls per hour for a system with 13 000 trunk circuits and 65 000 subscribers. It should be noted that this is not a hard limit, but merely a reasonable number of subscribers to place on one signalling gateway with the requisite minimum of two links in the PSTN network. With more links the system may be scalable to higher numbers up to the limit supported by the ISUP circuit range.
- On the IP side similar conditions hold, except that there is no requirement for separate signalling links or channels with 40 % occupancy. However, the same rule is very good engineering practice, and the IP network should be designed with sufficient IP bandwidth redundancy to guarantee signalling in the event of a link failure.

6.2.3 J.istp: Distribution model requirements

6.2.3.1 Bi-directional mapping among components

- It **must** forward incoming ISUP messages to the ISTP-User element (ex: MGC) controlling the MG associated with the circuit identity in the message. Conversely, it **must** forward IPCablecom outgoing messages from a ISTP-User to the SG:
  - The mapping between the SG TCAP-User element (ex: CMS) for TCAP messages is much more dynamic:
    - It **must** provide a unique transaction identity for TCAP transactions originated from multiple MGCs and CMSs, mapping the individual component ID's to a SG global ID.
    - It must **map** transactions identities for incoming transactions from the PSTN to the MGC or CMS handling the subsystem identified in the TCAP query.

6.2.3.2 Numbering: MGC name to IP address

- IPCablecom elements **must** be identified by their domain name, not their IP network addresses.
- Several IP addresses **may** be associated with a domain name.
- If a message cannot be forwarded to one of the network addresses, implementations **must** retry the transmission using another address.
- It **must** support transparent relocation of software components.
- The association between a logical name (domain name) and the actual platform **must** be kept in the Domain Name Service (DNS).
- It may use IPv4 addresses (a.b.c.d.) as per IETF Recommendation RFC791; it **may** support IPv6 in the future as well.

6.2.3.3 Numbering: circuits and transactions

- It **must** map SS7 identities (CIC, OPC, and DPC) to the internal identity of the IPCablecom network (MGC, MG, and domain name).
- Transaction numbering to the SS7 network **must** adhere to SS7 transaction numbering recommendations.
- Transaction numbering **must** be dynamically allocated on demand.
- Transaction identities **must** be kept only for a length of time much greater than the longest TCAP transaction to prevent reuse of numbering.

6.2.3.4 Message distribution

- ISUP messages **must** be routed from the SS7 network to IPCablecom components (MGC) by mapping the circuit identity to an IP address associated with the corresponding element that handles the MG that handles the unique circuit identity.
- Outgoing TCAP queries from the CMS or MGC **must** be routed by mapping a transaction id to an originating IP address; TCAP responses **must** be returned in the same manner.
- Some messages are internal to ISTP and **must** be routed by IP messaging to all ISTP nodes sharing a point code. These include maintenance messages, configuration messages, and congestion messages.

6.2.3.5 Alternate mapping on failure

- It **must** re-map to redundant or alternate paths upon detection of communication failures. On failures or communication timeouts:
  - if a MGC/CMS IP communication fails, it must look up alternate IP addresses first for that MGC/CMS element;
  - if a SG IP communication fails, the MGC/CMS must look up alternate IP addresses to the SG element.
- Given the performance requirements for call set up, ISTP **must** avoid trying to use IP addresses that are known to be unavailable, that is, are out of service or have failed a heartbeat test and timed out. Timers **should** be based on the Recommendations of the TCAP retransmission timers of the interfacing network.

6.2.3.6 Relationships

- It **must** provide mechanisms to manage mapping databases and tables, which map combinational parameters from SS7 messages to target IPCablecom elements and nodes.
- It **must** provide mechanisms to manage other databases and relational parameters not addressed in this specification.

6.2.3.7 Initialization

- It **must** support a complete "cold start" initialization of all elements, communications, and dynamic data in all the nodes of the IPCablecom network, and the start up of SS7 stack and links as well.
- It **must** support a CMS, MGC or SG element initialization, which initializes all IP physical and logical communications as well as all ISTP data in the element and all of its associated nodes in the IP network.
- It **must** support a CMS, MGC or SG single node initialization, which initializes the node's IP physical and logical communications as well as ISTP data.
- It **must** support ISTP stack only initialization, which initializes ISTP data.
- It **must** support IP communication only initialization, which initializes all IP physical and logical communications, as well as all affected ISTP configuration.
- The SG **must** support SS7 managed object initialization.
- When an ISTP stack restarts, it **must** be given all necessary information (e.g. point code identity, MGC/CMS/SG lists, CIC range, IP identities); how this is achieved is left to the specific implementation.
- When a new CIC range, CIC, MGC/CMS identity, IP address or SG point code is added to the network, all ISTP nodes sharing a common point code in the MGC-SG network **must** be informed and given the new or revised mapping in a consistent fashion by the trunking gateway operations support system.
- When an element or node restarts, it **should** notify all other known ISTP nodes sharing a common point code using the SS7 network inaccessible message and the SS7 network accessible messages when it is back in service; this **must** be done in an orderly manner so that it will not flood a node or network after an outage.

6.2.3.8 Recovery

- Given the PSTN-like or higher availability requirements, the ISTP **must** recover from failures quickly and robustly.
- ISTP **must** handle fully distributed n+k node architectures, as well as interfaces to the various SS7 highly reliable network configurations:
  - at the physical level, the ISTP **must** manage two or more network level interfaces to the IP systems;
  - in the event of a failure of one of the IP interfaces it **must** switch over to another IP interface.

- If a far end IP interface on a MGC or SG fails, the ISTP **must** try a second IP address; if this fails a third **must** be tried, etc., up to the optionally provisioned limit of the IP signalling network. Before trying any IP address, ISTP **must** check its availability status. If the MGC element cannot recover a communication, the SG **must** discard the messages only after trying all MGC nodes and failing to establish communication.

- If one MGC fails, the ISTP **must** retry to a second MGC; if this fails a third **should** be tried, etc., up to the optionally provisioned limit of the IPCablecom IP signalling network. If the MGC element cannot recover the communication, the SG **must** discard the messages only after trying all MGC nodes and failing to establish communication.

- There is only one SG (comprised of possibly multiple IP nodes). If it fails, recovery is beyond the scope of the ISTP, and the MGC **must** take recovery actions.

6.2.3.9 Dynamic provisioning updates

- The ISTP internal configuration mapping relationships **must** be dynamically updated without a network restart. It may store information in a local database, or require a central distribution of data on node recovery: this is an implementation option.

- Changing a mapping relationships **must** be done in a graceful and consistent manner across the entire IPCablecom network.

- Administration of ISTP data **must** be implemented in the following way:

  - for changes to existing relationships the entire IPCablecom network **must** be changed as one consistent transaction;

  - for any change to a relationship the addressable IP nodes **must** be managed in a graceful fashion; each node need to first be disabled (put out of service), then configured, audited to verify correct configuration, and then enabled (put back in service) in a way that does not suddenly flood the network;

  - For new relationships, there is no IP node to disable, but the provision **must** also be handled as one consistent transaction, audited, and each node placed in service gracefully.

6.2.3.10 Administration

- ISTP defines some semi-permanent objects and relationship (e.g. timers) that **must** be administered by the service provider's operations staff. The mechanisms and processes used to administer this data and behaviour are currently beyond the scope of the present document.

- The operational support system for the ISTP **must** have an audit feature that will allow network management to validate a successful configuration.

- The SG **must** support SS7 managed object administration.

6.2.3.11 ISTP Security

- ISTP Message authentication **must** use current state of the art Intranet technology to ensure safe and secure transport of IP messaging.

- Further security required at ISTP and higher level is currently under study.

6.2.3.12 Maintenance

- ISTP **must** manage the IP communications owned by the particular MGC, SG, or CMS, so it can proactively skip failed IP addresses when searching for a target IP without waiting for a timeout. It **must** support the following procedures:

  - enable IP, which places the IP connect in service and allows traffic;

  - disable IP, which removes the IP connection form service;

  - wait for traffic clear on IP connection;

- restart IP connection.

- The IPCablecom OSS system **must** supply interfaces for these procedures; it is out of the present document's scope to specify the interfaces.

- Note that the ISTP does not specify element or node management, only IP communications management. These functions will by handled by the OSS and their definition is beyond the scope of the present document.

- ISTP provides no additional requirements on SS7 maintenance.

6.2.3.13 Measurement

Operational measurements will be collected. The details on these issues are currently beyond the scope of this protocol Recommendation.

6.2.3.14 Alarms

- At a minimum the ISTP **should** generate alarms whenever an IP connection fails and whenever an ISTP node restarts.

- Other alarms **may** be generated, including SS7 related alarms.

6.2.3.15 Congestion

- Congestion on the SS7 network **must** be handled as per SS7 Recommendations the interfacing PSTN network.

- The CMS and the MGC **must** handle congestion messages from the SG and meet the SS7 requirements in this area. The ISTP will only pass congestion messages to the CMS and MGC; the SG itself **must** only take SCCP/MTP level recovery actions.

- Congestion on the IP network **must** be handled; the way this is done is left up to the vendor's implementation.

6.2.3.16 SG management notification

- When the status of lower layer objects, such as IP nodes, network clusters, or subsystems, change, SG **must** report the changes to MGC; the MGC **may** take autonomous recovery actions in the IPCablecom network and may notify operators of the change event.

- The MGC **must** respond to the status changes according to the SS7 Recommendation of the interfacing network.

6.2.4 J.istp: SS7 related protocol requirements

NOTE: Much, but not all, of this clause can be replaced by M3UA and SUA if they eventually become mature RFCs.

6.2.4.1 ISTP protocol requirements

- ISTP **must** provide a message distribution function that distributes ISUP and TCAP messages to/from distributed signalling components on the IP network.

- ISTP **must** provide an encoding schema for the transport of SS7 messages over a reliable IP-based protocol.

- ISTP **must** provide a set of messages and procedures for dynamically configuring the ISTP network on the IP side.

6.2.4.2 ISTP connection requirements

- ISTP **must** establish a reliable communication path.
- ISTP **must** guarantee the prompt and sequenced delivery of the messages.
- ISTP **must** provide information about the origination of incoming messages.
- ISTP **must** retransmit messages in case of errors or timeouts.
- ISTP **must** promptly detect failures in the communication path.
- ISTP **must** close communications.
- ISTP **may** use TCP or SCTP running over IP as layer 3 and 2 transport:
  - If it uses TCP, it **must** configure the IP network and set TCP parameters in such a manner as to provide the quality of service needed for SS7 signalling.
  - If it uses SCTP, it **must** act as a SCTP server, responding to client initiations from the MGC and MG.

6.2.4.3 ISTP SS7 encoding requirements

- ISTP **must** support "raw" ISUP and TCAP messages; it **may** support messages normalized to ITU or ANSI (note: not supported by IETF Internet-Drafts M3UA v.6 and SUA v.7 (see bibliography)).

6.2.4.4 ISTP load-sharing and sequencing

- The SG **must** assign the SLS value based on the CIC or the Transaction ID for outgoing messages in order to ensure optimal SS7 performance. The MTP Level 3 uses this value to distribute the traffic evenly between available signalling link, and the SG **must** supply an even distribution of the SLS values in order to achieve balanced load on all links.

6.2.4.5 Circuit registration and activation

- The SG **must** accept requests to register ISUP circuits.
- Once an MGC is successfully registered, it **must** activate the entries in order for them to take effect and bring them into traffic:
  - It **must** allow a "a forced exclusive circuit activation" to override existing activation abruptly.
  - It **must** support a "new work activation" to stop accepting new work; this allows the gradual movement of existing traffic off a circuit while maintaining stable calls, and supports a graceful software replacement strategy.
- Only one MGC element **should** be registered on a given circuit; multiple MGC nodes **may** register a circuit to provide load distribution and redundancy.
- Redundancy **may** be achieved by having more than one MGC node within an MGC element to register with more than one SG node.
- The SG **must** deny attempts to register more than one MGC element on a given circuit.
- Since, MGC nodes do not have a unique identifier, their IP interfaces are identified by their IP addresses. The ISTP protocol **must not** differentiate between IP interfaces belonging to one MGC node, or belonging to multiple MGC nodes.
- The SG **must** validate the registration and activation information and applicability before allowing the registration.
- The SG **must** support de-registration and de-activation of circuits as well.
- The SG **must** acknowledge successful or unsuccessful response to circuit registration and activation commands.

6.2.4.6 Transaction subsystem registration

- In order to exchange TCAP messages with nodes in the SS7 network, the TCAP-User components **must** properly register with the SG as a TCAP subsystem to properly distribute messages received from the SS7 network and provide some validation of the message to the SS7 network to minimize CMS/CA conflicts.
- Once an application is successfully registered, the TCAP-User component **must** activate the entries in order for them to take effect. Only forced activation **must** be supported; there are no procedures defined for maintaining work in progress transactions with the specific CMS nodes, since TCAP transactions have a very short life, are mostly involved in call set up, not stable calls, and the implementation of new work activation messages would add unnecessary complexity to ISTP.
- Multiple TCAP-User nodes **may** be registered with the same gateway point code and SSN values, and more than one may be active at any given time.
- Only one CMS/CA element **must** be registered with a SG element for the same point code and SSN values. The SG must deny attempts to register more than one CMS/CA element on a given subsystem.
- The SG **must** validate the registration and activation information and applicability before allowing the registration.
- The SG **must** support de-registration and de-activation of subsystem assignments as well.
- The SG **must** acknowledge successful or unsuccessful response to transaction registration and activation commands.

6.2.4.7 Message failure detection and handling

The following failures **must** be detected and recovered:

- The inability of the SG to transfer a message received from the MGC or a CMS onto the SS7 network.
- The inability of the SG to transfer a message received from the SS7 network to an MGC or a CMS.
- The loss of connectivity of the SG to the SS7 network.
- The loss of connectivity between the MGC or a CMS and the SG.
- The detection of congestion on the SS7 network.
- The detection of congestion on the IP network.

6.2.4.8 Heartbeat

- ISTP **must** validate its own and the application level operation with a query response type "heartbeat" message.
- All ISTP nodes **must** send heartbeat requests on a periodic basis, and must respond to incoming heartbeat requests as soon as they are received.
- The detailed steps taken upon delayed or missing heartbeat responses are implementation dependent, but failed IP connections **should** be disabled within a time period that allows the IPCablecom network to meet its stated availability requirements.

6.2.4.9 Signalling gateway accessibility procedures

- The SG can lose access to one or more SS7 signalling points due to local SS7 link failures, remote routing failures, or maintenance activities, and it can recover or gain access as well:
  - It **must** support notification of all registered IPCablecom components when a signalling point becomes inaccessible or accessible.
  - It **must** respond to the SS7 network according to the SS7 Recommendations, and also stops transferring messages from the CMS to the affected subsystem.

- The SG can lose access to the entire SS7 network due failure of all SS7 links:
  - It **must** support notification of all registered IPCablecom components when a the SS7 network becomes inaccessible or accessible.
  - It **must** respond to the SS7 network according to the SS7 Recommendations, waiting for the MTP_Restart procedure to complete At this point it **may** resume the transfer of SS7 messages and of ISTP transfer messages.
- The SG can lose connectivity to an MGC or a CMS because of IP network or node failures, or scheduled maintenance. When the SG detects loss of connectivity to a ISTP node it **must** deactivate and de-register all circuits and subsystems with that ISTP element, and discards any subsequent SS7 messages.

6.2.4.10 SG congestion handling

- If the SG detects the congestion of a signalling point by receiving a SS7 network message, it **must** notify all registered IPCablecom nodes with the congestion level that was received; the SG **should** provide a mechanism for detection of the end of congestion.
- If the SG autonomously detects congestion of the local SS7 links for outbound traffic, it **must** notify all registered IPCablecom nodes with the congestion level that was received; the SG **should** provide a mechanism for detection of the end of a congestion status.
- If the SG detects congestion of the IP network to the MGC or the CMS node, it **must not** notify the adjacent SS7 nodes. Instead, it **must** use a four level congestion scheme as defined in MTP level 3, and discard messages based on the priority of the messages The method of detection and the measurement of congestion on the IP network is dependant on the lower layer used, and on the implementation.

6.2.4.11 IPCablecom component management procedures

IPCablecom components (ex: MGC, CMS) **must** support SS7 recommendations and standards by taking appropriate actions and sending required MTP3 messages to handle:

- signalling point inaccessible/accessible.
- SS7 network inaccessible/accessible.
- SG inaccessible/accessible.
- SS7 Network congestion/decongestion.
- IP Network congestion/decongestion.

6.2.4.12 IP usage

- Either SCTP or TCP **must** be used as an underlying transport mechanism.
- The underlying transport mechanism **must** be configured to avoid blocking, mis-sequencing, IP network congestion and sending delay.
- The underlying physically network **must** support redundancy sufficient to meet network reliability goals, which are at the option of the operator.

6.2.5 J.istp: Future work

- ISTP is currently focused on ANSI; it is not planned to upgrade the SS7 protocols requirements clause 5 and call flows to ITU.
- Instead, clause 5 and the call flows would be replaced by M3UA and SUA based text, using a fixed draft version of the IETF specification, and the other clauses upgraded to handle M3UA and SUA management. This solves the problem of the lack of RFC status, by defining a M3UA and SUA subset as an ITU specification.

- The document would then be all body, with normative and informative references: ISTP would still be a substantial document implementing the framework and operator requirements, particularly the distribution model.

7 Requirements obtained from J.tgcp

7.1 Overview

7.1.1 Scope

This clause describes the requirements obtained from reverse engineering the ITU-T J.tgcp document (J.171 [3]). The source document used has been written by the Study Group 9 of the ITU-T. The title of this file is "Revised Draft New Recommendation J.tgcp - IPCablecom Trunking Gateway Control Protocol (TGCP)", as of December 2002 [3]. The present document has been downloaded from the ITU-T web site (not public accessible).

In this clause, "it" will mean "the MG, MGC, and TGCP implemented as the MGCI". Note that there are other requirements on the MG and MGC not covered here.

7.1.2 What is J.tgcp?

J.tgcp (or J.171 [3]) is a globalized ITU-T version of the Trunking Gateway Control Protocol (TGCP) released in 1999 by Cablelabs. TGCP has been adapted by the ITU in draft J.171. It includes the following characteristics:

- It defines an API and a text based protocol in the IPCablecom Framework Architecture.
- TGCP supports IP voice over IP PSTN Trunking Gateways; it specifies the package of events and connection options for trunk connections to the PSTN only.
- TGCP appears as an annex in the document, since its trunk interfaces use MF signalling for NA operator assisted trunks; the use of MF signalling for areas outside NA is under study.
- TGCP is defined as a "profile" of the IETFRFC2805 Media Gateway Control Protocol (MGCP):
  - TGCP contains extensions and modifications to RFC2705 to support simple SS7 (clear channel no signalling) control of trunks (circuits), and control of MF Operator Service trunks.
  - TGCP contains simplifications of MGCP.
  - TGCP does not address ISDN, or R1, or R2, or ATM signalling.
  - TCCP is "stand alone" and complete in itself.

7.1.2.1 Note on MG, MGC, TGCP and MGCI

- They are not quite the same thing:
  - The media gateway (MG) is a logical component (a set of software and hardware) that is the single point of interface for bearer channel trunks to the PSTN; the MGC is a logical component (a set of software running on a computer) that manages trunk calls.
  - TGCP is a protocol implemented as a stack element running over IP and runs on several IPCablecom components (MGC and MG).
  - MGCI is an API that implements the TGCP protocol in the MG and MGC.
  - TGCP could be used in non-IPCablecom architectures.
  - In the present document, where "gateway" or "media gateway" or "MG" is used, Trunking Media Gateway is actually meant.

- J.tgcp defines requirements on TGCP the protocol, MGCI the API, and the MGC and MG as implemented in a IPCablecom network.

7.1.2.2 TGCP in the protocol stack

The TGCP protocol is highlighted in figure 6.

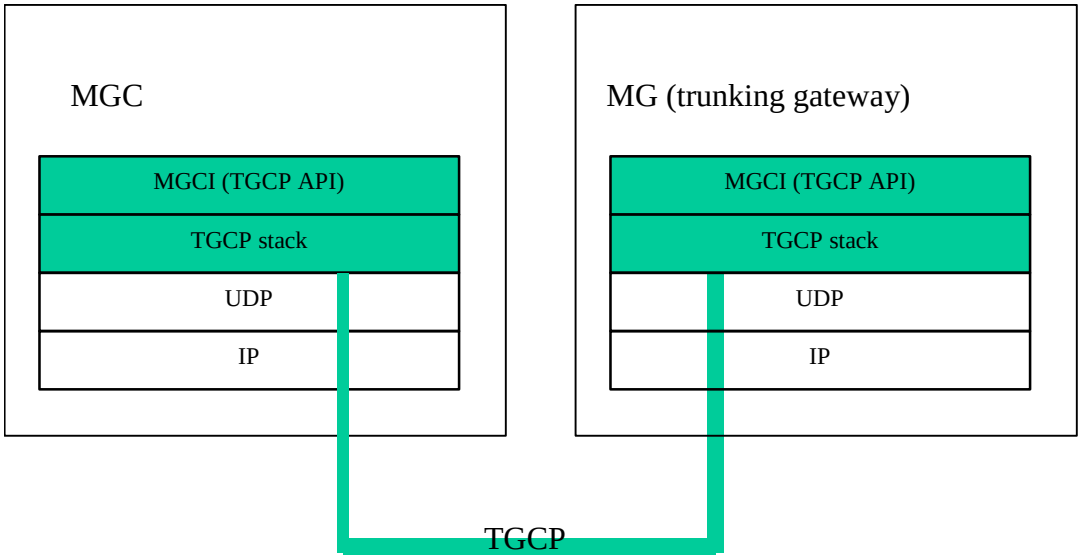


Figure 6: TGCP in the protocol stack

7.1.2.3 J.tgcp Trunking Gateway Document

The J.tgcp document contains:

- Presentation of example APIs with the name of the command, the parameters it can take and return, and the semantics of these.
- TGCP encoding of the commands and parameter based on text formats.
- Annexes for endpoint event packages.
- Appendices for:
  - Connection mode.
  - Example Command Encoding.
  - MF Terminating Protocol Package.
  - Example call flows.
  - Endpoint requirements.
  - Compatibility information.
  - Example endpoints.

7.1.2.4 First level decomposition of a Managed IP Network

See figure 1 for the first level decomposition.

7.1.2.5 Areas beyond the scope of J.tgcp

The following areas are out of the scope of the TGCP:

- Address layer management (SNMP), security, and measurements covered in other IPCablecom Recommendations, but the MG must adhere to those requirements within the J.arch framework.
- Implementation and vendor dependant issues, such as performance, functional distribution, network configuration, etc.
- Details about CMS, MGC, and other media communication applications.
- Other legacy trunk interfaces, which are not required (ex: ISDN, MF, R1, R2, etc.), or future trunk interfaces, such as BICC, which are not yet (widely) deployed.
- OSS and OA&M issues, except where support for such capabilities is noted.
- Operator or User level requirements - this focuses only on a subset of the decomposed components MGC and MG, and the interface between them.

7.2 J.tgcp: Technical requirements

7.2.1 J.tgcp: Framework and Architecture Requirements

7.2.1.1 Support of J.arch framework service goals

It **must** support J.arch/J.nsc service goals including:

- voice or other media content conversion;
- call control signalling;
- quality of service control;
- call control signalling interoperability with the existing public network;
- media interfaces to the existing public network;
- data transactions to public databases;
- routing mechanisms;
- billing;
- operations and maintenance;
- security;
- privacy.

7.2.1.2 High level control requirements

- It **must** provide:
  - Connection control.
  - Endpoint control.
  - Auditing.
  - Status reporting.
- It **must** provide a common naming convention for using these functions.

- It **must** support text based control of devices, with ASCII endpoint and connection naming.

7.2.1.3 High level IP side management requirements

It **must** support IP management procedures for:

- Notification request - instruct the MG to look for a set of events.
- Notify - notify the MGC when an looked for event occurs.
- Create connection - instruct the MG to set up an IP connection.
- Modify connection - change the nature of an existing connection.
- Delete connection - drop a connection.
- Audit endpoint - return information on an endpoint.
- Audit connection - return information on a connection.
- Restart in progress.

7.2.1.4 TGCP high level trunk side requirements

- It **must** provide control of clear channel SS7 trunk (circuits); there is no signalling information passed for these trunks.
- It **must** support requirements for operator assisted calls using legacy MF trunks:
  - *Information note*: "the specification in this annex is used in North America", means that MF Operator Services trunks are supported. This is a set of trunks supporting operator capabilities (call interrupt, billing, etc.) and *11 (911 emergency, 411 directory assistance, 611 customer service, etc.), and E911 services. In NA, both SS7 and MF in channel signalling are deployed; in Europe only SS7 is required. Note that even in NA this is being phased out as NPA relief plans are implemented, and E911 is the preferred solution in the future, but even so MF is still needed as "backup" in the event of a SS7 access failure. It is not clear if this is needed outside of NA.
  - The MF operator service part is implemented as a separate package.

7.2.1.5 Other Trunking Gateway Requirements

Trunking Gateway **must** provide:

- TGCP signalling to the MGC.
- Support for OA&M.
- Support for IPCablecom QoS.
- Support for IPCablecom Security.
- Static binding to a MGC component type; the MGC component may contain many MGC nodes, each with one or more IP addresses.
- Dynamic binding using DNS to map the name of the MGC currently controlling an endpoint to the address.
- Dynamic rebinding to alternate MGC nodes in the case of a failure of a single MGC node.

7.2.1.6 Distribution model

- A MG is physically connected to a trunk and thus is tied to the hardware.
- Its association with a MGC is flexible; one MG over time can be controlled by different MGCs; this binding can be either provisioned, controlled by a DSN server, or autonomously changed by the MG in the event of a failure of a MGC.

7.2.2 J.tgcp: MGCI API requirements

7.2.2.1 Model and naming convention

- It **must** adhere to the MGCP Connection Model:
  - MGCP assumes connection model where basic constructs are *endpoints* and *connections* participating (grouped) in a *call.*
  - Trunking gateways have one or more endpoints, e.g. one for each trunk.
  - Connections will be grouped in calls; one or more connections can belong to one call.
- Connections and calls **must** be set-up on initiative by one or several MGCs; only MGCs initiate or control connections and calls.
- An MGC **must** be identified by its domain name, not the network name or address.
- An endpoint **must** only have one MGC element associated with it at any point in time.
- Names **must** be text based.

7.2.2.2 Endpoint name

- Endpoint names/identifiers **must** have two components:
  - Domain name of the gateway managing the endpoint.
  - A local endpoint name within that domain.
- Domain name **may** be an IPv4 IP address in dotted decimal format represented as a text string, but this is discouraged.
- Trunking gateways will have one or more endpoints (one for each trunk circuit, e.g. DS-0).
- Each of the endpoints **must** be associated with a different local endpoint name.
  - Associated with the local endpoint name, **must** be the endpoint type, e.g. DS-0 or analogue access line.
  - Individual terms **must** be ASCII character strings.
  - Unique Separators, wildcards all (*), and wildcard anyone ($) **must** be supported.

7.2.2.3 Trunk name

- Trunking gateways **must** support the following basic endpoint types:
  - DS-0 ISUP trunk (clear channel with out of band SS7 signalling).
  - DS-0 MF trunk (for MF Operator Services).
- It **must** adhere to a conventions supporting hierarchical name support with/separation (for the DS-3, 1, 0), decimal channel numbers, wildcards, allowing a flexible text based manipulation of trunks at every level.

7.2.2.4 Call and Connection names

- Calls **should** be identified uniquely by the MGC, with hexadecimal a number.
- Call identifiers **must** be unique within the collection of MGCs that control the same gateways.
- Multiple connections pertaining to the same call, **must** have the same call identifier, that can be used by accounting or management schemes.
- The MGCP **must** assign a hexadecimal connection name of no more than 32 text characters to each connection.
- After breaking a connection, the same name **must not** be used for at least a period of 3 minutes.

7.2.2.5 MGC naming

- MGC **must** be named similar to endpoints name (domain name + local portion), and **must** be identified by its (logical) name not (physical) IP address for normal call handling.
- One MGC **must** be able to have multiple IP network addresses; thus network addresses **must not** be used for identification.
- For redundancy, one or more alternative network address **may** be used, and **must** be tried in the event of a failure.
- Entities **may** be moved to other platforms; where the DNS tracks the association between logical name the association **must** be held on a DNS platform, and the actual platform.
- MGC and Gateways **must** keep track of the time-to-live read from the DNS; the MGCs and MGs **must** query the DNS if the time-to-line has expired.

7.2.2.6 MGC name binding

- The "notified entity" of an endpoint is the MGC controlling that endpoint and represents a second level of name binding:
  - upon start-up, the notified entity **must** be set to a provisioned value;
  - MGC commands sent to the MG **must** contain the notified entity name for the endpoint invoked in the command;
  - the MG **must** set its notified entity dynamically to match that in the command name.
- If the "notified entity" is not set or is empty, the MG **must** default to the source address of the last connection handling command or notification request received.
- Auditing **must not** change the "notified entity".
- Each endpoint **must** have one and only one notified entity at a time.

7.2.2.7 Digit Maps

- None of the trunk types supported by the current version of the TGCP Recommendation have a need for digit maps, and digit maps therefore must not be part of the current TGCP Recommendation.

7.2.2.8 Packages

- The concept of events and signals is central to TGCP. Events and signals **must** be grouped in packages sharing a common namespace (a set of unique names).
- One or more packages **may** exist for a given endpoint-type; each endpoint-type **must** have a default package with which it is associated.
- Each endpoint-type **must** have a default package.

- Package names and event codes **must** be case insensitive strings of letters, digits, and hyphens, with the restriction that hyphens shall never be the first or last character in a name.
- Some event codes **may** need to be parameterized with additional data, which **must** be accomplished by adding the parameters between a set of parentheses. The package name **must** be separated from the event code by a slash ("/"). The package name **may** be excluded from the event name, in which case the default package name for the endpoint-type in question **must** be the default.
- Additional data **must** be added between a set of parentheses.
- The package name **must** be separated from the event code by a slash ("/").
- The package name **may** be excluded, in which case the default package name **must** be the default.
- Additional package names and event codes **may** be defined by and/or registered with IPCablecom.
- Any change to the packages defined in the present document **must** result in a change of the package name, or a change in the TGCP profile version number, or possibly both.
- Each package **must** have a package definition that defines the name of the package and the definition of each event belonging to the package:
  - The event definition **must** include the precise name of the event, i.e. the event code and a plain text definition of the event.
  - Events **must** specify if they are persistent and if they contain audible event-states.
  - Signals **must** also have their type defined.
  - Time-out signals **must** have a default time-out value defined.

7.2.2.9 Experimental Packages

- Implementers **may** define experimental packages.
- The package name of experimental packages **must** begin with the two characters "x-" or "X-".
- IPCablecom **must not** register package names that start with these two characters.
- A MG that receives a command referring to an unsupported package **must** return an error.

7.2.2.10 Wildcard support

- Package names and event codes support one wildcard notation each.
- The wildcard character "*" (asterisk) **must** be used to refer to all packages supported by the endpoint in question, and the event code "all" to all events in the package in question.
- Consequently, the package name "*" **must not** be assigned to a package, and the event code "all" **must not** be used in any package.

7.2.2.11 Events and Signals on connections

- Events and signals are by default detected and generated on endpoints, however some events and signals **may** be detected and generated on connections in addition to or instead of on an endpoint.
- In order for an event or signal to be able to be detected or generated on a connection, the definition of the event/signal **must** explicitly define that the event/signal can be detected or generated on a connection:
  - The name of the connection **must** be added to the name of the event using an "at" sign (/@) as a delimiter.
  - Wild cards for "all connections" (/*) and "current connections" (/$) and all packages all events (/all@*) are supported.

7.2.2.12 Session Description Protocol

- The MGC **must** provide MGs with description of parameters (ex: IP addresses, UDP port, RTP profile) using SDP.
- SDP descriptions **must** follow conventions as in Session Description protocol (SDP) in IETF RFC 2327 [11], however, Trunking gateways make certain simplifying assumptions:
  - SDP usage depends on type of session, as specified in the "media" parameter.
  - Currently only media type "audio" **must** be supported.
  - SDP profile: typical parameters specified **must** be supported, but non specified parameters **should not** be provided and should be ignored if received.

7.2.2.13 Gateway control functions

- An API function **must** be defined for every MGCP command.
- The MGCI function **must** take and return the same parameters as the corresponding MGCP command.
- Functions **may** be implemented, but functions that are implemented, **must** conform to the semantics specified.
- MGCI APIs **must** support connections handling and endpoint handling commands.

7.2.3 J.tgcp: Control Function Requirements

7.2.3.1 Commands

It **must** support commands to:

- Allow the MGC to notify an MG of a service request (NotificationRequest).
- Allow MGC to notify the MGC that it is ready for service (Notify).
- Allow the MGC to request a connection (CreateConnection).
- Allow the MGC to modify a connection (ModifyConnection).
- Allow the MGC to delete a connection (DeleteConnection).
- Allow the MGC to audit an endpoint (AuditEndpoint).
- Allow the MGC to audit a connection (AuditConnection).
- Allow the MG to notify the MGC that it is restarting (RestartInProgress).

7.2.3.2 Calls

Connections are grouped into "calls".

- Several connections, that **may** or **may not** belong to the same call, can terminate in the same endpoint.
- Multiple calls **may** be active on the same endpoint.

7.2.3.3 Connection mode parameter

- Each connection **must** be qualified by a mode parameter:
  - Send only.
  - Receive only.
  - Send/Receive.
  - Inactive.
  - Loop back.
  - Continuity test.
  - Network loop back.
  - Network continuity test.
- The mode parameter **must** define if the connection can send or receive packets; however, RTCP **must** be unaffected by the mode parameter.

TGCP has important properties of a transport protocol: it runs on end systems, it provides de-multiplexing. It differs from transport protocols like TCP in that it (currently) does not offer any form of reliability or a protocol-defined flow/congestion control. However, it provides the necessary hooks for adding reliability, where appropriate, and flow/congestion control. Some like to refer to this property as application-level framing (see D. Clark and D. Tennenhouse, "Architectural considerations for a new generation of protocols", SIGCOMM'90, Philadelphia). RTP so far has been mostly implemented within applications, but that has no bearing on its role. TCP is still a transport protocol even if it is implemented as part of an application rather than the operating system kernel.

RTCP is the control protocol that works in conjunction with RTP. It provides support for real-time conferencing for large groups within an internet, including source identification and support for gateways and multicast-to-unicast translators. It is standardized in RFC 1889 and RFC 1890

7.2.3.4 Audio connection modes

- Audio signals received from the endpoint **may** be sent on any connection for that endpoint whose mode is either "send only", or "send/receive.
- Handling of the audio signals received on these connections **must** also be determined by the mode parameters:
  - Audio signals received in data packets through connections in "inactive", "loop-back" or "continuity test" mode **must** be discarded.
  - Audio signals received in data packets through connections in "receive only", or "send/receive" mode **must** be mixed together and then sent to the endpoint; endpoints **may** or **may not** support mixing.
  - Audio signals originating from the endpoint **must** be transmitted over all the connections whose mode is "send only", or "send/receive".
  - Audio signals received in data packets through connections in "network loop-back" or "network continuity test" mode **must** be sent back on the connection as described below in the loop-back test requirements.

7.2.3.5 Loop-back and continuity testing

- It **must** support "loop-back" and "continuity test" modes to be used during maintenance and continuity test operations.
- New and existing connections for the endpoint **must not** be affected by connections placed in "network loopback" or "network continuity test" mode.
- Local resource constraints **may** limit the number of new connections that can be made when testing is going on.

7.2.3.6 Continuity test (COT)

- It **must** support two variations of continuity test (COT):
  - One specified for general use: the loopback test. If the originating switch sees the same tone returned (the return tone), the COT has passed. If not, the COT has failed.
  - One used in several national networks, where the go and return tones are different. When the terminating switch detects the go tone, it asserts a different return tone in the backwards direction. When the originating switch detects the return tone, the COT is passed. If the originating switch does not detects the return tone within a certain period of time, the COT has failed.
- If the mode is set to "loop-back", the gateway **must** return the incoming signal from the endpoint back into that same endpoint.
- If the mode is set to "continuity test", the gateway **must** be informed that the other end of the circuit has initiated a continuity test procedure. The gateway **must** place the circuit in the transponder mode required for dual-tone continuity tests.

7.2.3.7 Audio requirements on testing

- When a connection for an endpoint is in "loop-back" or "continuity test" mode:
  - Audio signals received on any connection for the endpoint **must not** be sent to the endpoint.
  - Audio signals received on the endpoint **must not** be sent to any connection for the endpoint.
- If the mode is set to "network loop-back", the audio signals received from the connection **must** be echoed back on the same connection. The "network loop-back" mode **should** simply operate as an RTP packet reflector.
- The "network continuity test" mode is used for continuity checking across the IP network. An endpoint-type specific signal is sent to the endpoints over the IP network, and the endpoint **must** echo the signal over the IP network after passing it through the gateway's internal equipment to verify proper operation. The signal **must** go through internal decoding and re-encoding prior to being passed back. For DS-0 endpoints the signal will be an audio signal, and the signal **must not** be passed on to a circuit connected to the endpoint, regardless of the current seizure-state of that circuit.

7.2.4 Requirements from notification request message and parameters

7.2.4.1 Notification request command

- This message **must** be used to notify a MG to be ready for service.
- Command must be applied to a unique endpoint; wildcard commands **must not** be used.
- Endpoint **must** support all signals; connections **may** support some signals, such as Continuity test and set-up MF OSS call:
  - On/off signals **must** be supported, and last until they are turned off. This **must** only happen as the result of a new command where the signal is turned off; multiple requests to turn a given on/off signal on (or off) are perfectly valid and **must not** result in any errors; once turned on, it **must not** be turned off until explicitly instructed to by the MGC, or the endpoint restarts.
  - Time-out (TO) signals **must** last until they are either cancelled (by the occurrence of an event or by not being included in a subsequent [possibly empty] list of signals), or a signal-specific period of time has passed. Time-out signals **must** have a default time-out value defined for them, which may be altered by the provisioning process. Also, the time-out period **may** be provided as a parameter to the signal.
  - Brief (BR) the duration of these signals is so short that they **must** stop on their own. If a signal stopping event occurs, or a new signalling request command is applied, a currently active BR signal will not stop. However, any pending BR signals not yet applied **must** be cancelled.

7.2.4.2 Media stream notification requirements

- If a signal applied to an endpoint results in the generation of a media stream (audio, video, etc.), the media stream **must not** be forwarded on any connection associated with that endpoint, regardless of the mode of the connection. For example, if a tone is applied to an endpoint involved in an active communication, only the party using the endpoint in question will hear the tone. However, individual signals **may** define a different behaviour.

- When a signal is applied to a connection that has received a remote connection descriptor, the media stream generated by that signal **must** be forwarded on the connection regardless of the current mode of the connection. If a remote connection descriptor has not been received, the gateway **must** return an error.

- When a (possibly empty) list of signal(s) is supplied, this list completely replaces the current list of active time-out signals. Currently active time-out signals that are not provided in the new list **must** be stopped and the new signal(s) provided will now become active. Currently active time-out signals that are provided in the new list of signals **must** remain active without interruption, thus the timer for such time-out signals will not be affected. Consequently, there is currently no way to restart the timer for a currently active time-out signal without turning the signal off first. If the time-out signal is parameterized, the original set of parameters **must** remain in effect, regardless of what values are provided subsequently. A given signal **must not** appear more than once in a signal requests.

7.2.4.3 Dynamic configuration of events

- It **must** support a dynamically configurable set of events on each endpoints and connection.

- It **must** support a list of requested events that the gateway **must** detect on the endpoint. Unless otherwise specified, events are detected on the endpoint, however, some events can be detected on a connection.

- For each event, the gateway **must** take one or more of the following actions on occurrence:

  - Notify the event immediately, together with the accumulated list of observed events.

  - Accumulate the event.

  - Ignore the event.

  - Keep signal(s) active.

  - Embedded notification request.

  - Embedded modify connection.

7.2.4.4 Default vs. dynamically set events

- It **must** support persistent and non-persistent events.

- Persistent events are events that **must** always detected on an endpoint, even if a persistent event is not included in the list of requested events occurs, the event **must** be handled as if it were a requested event.

- Persistent events **must** still be detected and notified even if an empty requested event list is received.

- Non-persistent events are those events that have to be explicitly included in the requested events list. The (possibly empty) list of requested events **must** completely replace the previous list of requested events. In addition to the persistent events, only the events specified in the requested events list **must** be detected by the endpoint. If a persistent event is included in the requested events list, the action specified **must** then replace the default action associated with the event for the life of the requested events list, after which the default action is restored. A given event **must not** appear more than once in a requested event list.

7.2.4.5 Event requests

- It **must** return an error code to the MGCP if it receives a request with an invalid action or illegal combination of actions.
- When multiple actions are specified (e.g. "Keep signal(s) active" and "Notify") the individual actions **must** be assumed to occur simultaneously.
- The generation of all "Time Out" signals **must** stop as soon as one of the requested events is detected, unless the keep signal(s) active action is associated to the specified event.
- If it is desired that time-out signal(s) continue when a looked-for event occurs, the "Keep Signal(s) Active" action **may** be used. This action has the effect of keeping all currently active time-out signal(s) active, thereby negating the default stopping of time-out signals upon the event's occurrence.

7.2.4.6 Event scripting

- If signal(s) are desired to start when a looked-for event occurs, an embedded notification request action **may** allow the MGC to set up a "mini-script" to be processed by the gateway immediately following the detection of the associated event.
- If connection modes are desired to be changed when a looked-for event occurs, the "Embedded Modify Connection" action **may** be used. The wildcard "$" **may** be used to denote "the current connection", however this notation **must not** be used outside a connection handling command - the wildcard refers to the connection in question for the connection handling command.
- The embedded modify connection action may allow the MGC to instruct the endpoint to change the connection mode of one or more connections immediately following the detection of the associated event. When a list of connection mode changes is supplied, the connection mode changes must be applied one at a time in left-to-right order. When all the connection mode changes have finished, an "operation complete" event parameterized with the name of the completed action **must** be generated. Should any of the connection mode changes fail, an "operation failure" event parameterized with the name of the failed action and connection mode change **must** be generated - the rest of the connection mode changes **must not** be attempted, and the previous successful connection mode changes in the list **must not** be changed either.
- An ignore action **may** be used to ignore an event, e.g. to prevent a persistent event from being notified. However, the synchronization between the event and an active signal **must** still occur by default.

7.2.4.7 Event request vs. signal request

- The specific definition of actions that are requested via signal Request is outside the scope of the core recommendation.
- The requested events and signal requests generally refer to the same events. In one case, the gateway is asked to detect the occurrence of the event and, in the other case, it is asked to generate it. There are only a few exceptions to this rule, notably the fax and modem tones, which can be detected but cannot be signalled. The specific events and signals that a given endpoint can detect or perform **must** be determined by the list of event packages that are supported by that endpoint.
- Each package **must** specify a list of events and signals that can be detected or applied. A gateway that is requested to detect or to apply an event belonging to a package that is not supported by the specified endpoint **must** return an error.
- When the event name is not qualified by a package name, the default package name for the endpoint **must** be assumed.
- If the event name is not registered in the signal request default package, the gateway **must** return an error.
- The MGC **may** send a notification request whose requested signal list is empty. This has the effect of stopping all active time-out signals. It can do so, for example, when tone generation, e.g. ring-back, **should** stop.

7.2.4.8 Quarantine handling

- It **must** allow the MGC to specify whether quarantined events **should** be processed or discarded.
- If the parameter is absent, the quarantined events **must** be processed.

7.2.5 Requirements from Notify message and parameters

7.2.5.1 Notify requirements

- The endpoint identifier **must** be a fully qualified endpoint name, including the domain name of the gateway, and the local part of the name **must not** use the wildcard convention.
- The notification **must** be sent to the current "notified entity" for the endpoint; the return message **may** identify the entity to which the notification is sent.
- It **must** correlate the notify message with the notification request that triggered it.
- It **must** deliver a list of events that the gateway detected and accumulated, either by the "accumulate", or "notify" action. A single notification may report a list of events that will be reported in the order in which they were detected. The list **should** only contain persistent events and events that were requested using the requested events feature of the triggering notification request. Events that were detected on a connection **must** include the name of that connection. The list **must** contain the events that were either accumulated (but not notified), and the final event that triggered the notification.

7.2.6 Requirements from create connection message and parameters

7.2.6.1 Connection definition

- A connection is defined by its attributes and the endpoints it associates with. The MGC **must** provide all the data necessary to build one of the two endpoints "view" of a connection.
- Call identities.
- They **must** at a minimum be unique within the collection of MGCs that control the same gateways.
- Connections that belong to the same call **must** share the same call-id.
- The call-id **may** be used to identify calls for reporting and accounting purposes.
- The endpoint identity **must** be specified fully by assigning a non-wildcarded value. The "all" wildcard convention must not be used.

7.2.6.2 Local connection characteristics

- The MGC **must** instruct the endpoint on the send and receive characteristics of the media connection.
- The MG **must** respond with an error if any of the local connection characteristics rules are violated. Some are dynamic, and some set by default, and all defaults **must** be modified by provisioning.
- Encoding Method: A list of literal names for the compression algorithm (encoding/decoding method) used to send and receive media on the connection **must** be specified with at least one value. The entries in the list are ordered by preference. The endpoint **must** choose exactly one of the Codecs, and the Codec **should** be chosen according to the preference indicated. If the endpoint receives any media on the connection encoded with a different encoding method, it **may** discard it. The endpoint **must** additionally indicate which of the remaining compression algorithms it is willing to support as alternatives
- Packetization Period: the packetization period in milliseconds, as defined in the SDP standard (RFC 2327), **must** be specified and with exactly one value.

- Echo Cancellation: indication whether this is used on the trunk side **may** or **may not** be sent. If omitted, the trunking gateway **must** apply echo cancellation.
- Type of Service: **may** be sent; if omitted, a default value **must** be assumed.
- Silence Suppression: an indication on whether silence suppression **should** be used or not in the send direction **may** or **may not** be sent. The parameter can have the value "on" (when silence is to be suppressed) or "off" (when silence is not to be suppressed). When the parameter is omitted, the default silence suppression **must not** be used.

7.2.6.3 Local connection options for IP security

- Secret: A seed value that **may** be used to derive end-to-end encryption keys for the RTP and RTCP security services. The secret **should** be encoded as clear-text if it only contains values in the ASCII character range 21H to 7EH. Otherwise, the secret **must** be encoded using base64 encoding. If no value is supplied, or the parameter is omitted and security services are to be used, the endpoint **must** generate a secret on its own. When a secret is supplied by the MGC, the secret **should** be used.
- RTP ciphersuite: A list of ciphersuites for RTP security in order of preference **must** be supported, ordered by preference where the first ciphersuite is the preferred choice. The endpoint **must** choose exactly one of the ciphersuites. The endpoint **must** additionally indicate which of the remaining ciphersuites it is willing to support as alternatives.
- RTCP ciphersuite: A list of ciphersuites for RTCP security in order of preference **must** be supported, ordered by preference where the first ciphersuite is the preferred choice. The endpoint **must** choose exactly one of the ciphersuites. The endpoint **must** additionally indicate which of the remaining ciphersuites it is willing to support as alternatives.

7.2.6.4 Local connection LI requirements

- It must support Lawful Intercept:
  - Connections **must** support replication and forwarding to an Electronic Surveillance Delivery Function with Call Content Connection Identifier attached.
  - Media generated by signals applied to the connection **must** be replicated regardless of the connection mode.
  - Replicated packets **must not** be included in statistics for the connection.

7.2.6.5 Remote connections

- The same media parameters **must** apply to a connection in both the send and receive direction.
- If inconsistency is detected by a gateway between the local and the remote connection, the local **must** take precedence.
- When codecs are changed during a communication, small periods of time may exist where the endpoints use different codes. The MG **may** discard any media received that is encoded with a different codec than what is specified in the local connection option.

7.2.6.6 Local connection modes

- This **must** be the same as the notify modes.
- If the command specifies a mode that the endpoint does not support, an error **must** be returned.
- If a connection has not yet received a remote connection descriptor, an error **must** be returned if the connection is attempted to be placed in any of the modes "send only", or "send/receive".

- Handling the window between sending local and receiving remote connections description **must** be done as follows:
  - if the mode was set to "receive only", the gateway **must** accept the voice signals received on the connection and transmit them through to the endpoint;
  - if the mode was set to "inactive", "loop-back", or "continuity test" the gateway **must** (as always) discard the voice signals received on the connection;
  - if the mode was set to "network loop-back" or "network continuity test" the gateway **must** perform the expected echo or response. The echoed or generated media **must** then be sent to the source of the media received 7.2.6.7 Synchronized create and notify.
- The MGC **may** command a simultaneous notification request and connection creation as a single command.
- The creation of the connection and the notification request **must** be synchronized, which means that they are both either accepted or refused.
- The call initiation notification request **must** be refused in the glare condition if the circuit is already seized. An error **must** be returned instead which informs the MGC of the glare condition.

7.2.7 Requirements from Modify Connection message and parameters

7.2.7.1 Modify connection

- It **must** be able to modify the characteristics of a gateway's "view" of a connection dynamically.
- It **must** modify information:
  - on the other "remote" end of the connection;
  - on the modes that activate or deactivate the connection;
  - on the local parameters of the connection.
- RTP address information **may** be changed by the MGC. When RTP address information is given to MG for a connection, the MG **should** only accept media streams (and RTCP) from the RTP address specified. Any media streams received from any other addresses **should** be discarded.

7.2.7.2 Synchronized create and modify connection

- Detect events parameters **may** be used by the MGC to effectively include a modify request that is executed simultaneously with the creation of the connection.
- The creation of the creation and the modification request **must** be synchronized, which means that they are both either accepted or refused.

7.2.8 Requirements from Delete Connection (from MGC) message and parameters

7.2.8.1 Delete message usage

- It **must** support deletion (termination) of a connection.
- In the general case where a connection has two ends, this command **must** be sent to both gateways involved in the connection.
- After the connection has been deleted:
  - media streams previously supported by the connection **must** no longer be available;
  - any media packets received for the old connection **must** be discarded and no new media packets for the stream sent.

7.2.8.2 Return performance data

- It **must** return status data after a connection is terminated, including:
  - Number of packets sent.
  - Number of octets sent.
  - Number of packets received.
  - Number of octets received.
  - Number of packets lost.
  - Interarrival jitter.
  - Average transmission delay.
  - It may include other data.

7.2.8.3 Synchronized notify and delete connection

- It **must** accept a notification request that is tied to and executed simultaneously with the deletion of the connection.
- The notify and the delete request **must** be synchronized, which means that they are both either accepted or refused.

7.2.9 Requirements from Delete Connection (from MG) message and parameters

7.2.9.1 Delete connection forced by MG

- The MG must support the forced termination of the connection by using a variant of the DeleteConnection command.
- The endpoint identifier, **must** be fully qualified; wildcard conventions **must not** be used.
- It **must** return a reason code for the termination.

7.2.9.2 Delete multiple or all connections

- Deletion of multiple connections, initiated from the MGC, command **must** be supported:
  - The endpoint identity **must not** use the "any of" wildcard.
  - All connections for the endpoint(s) with the CallId specified **must** be deleted. The command does not return any individual statistics or call parameters.
- Deletion of all connections that terminate in a given endpoint **must** be supported:
  - MGCs **must** take advantage of the hierarchical naming structure of endpoints to delete all the connections that belong to a group of endpoints using the "all" wildcarding convention. The "any of" wildcarding convention **must not** be used.
  - The command does not return any individual statistics or call parameters.
- After the connection has been deleted, packet network media streams previously supported by the connection **must not** be longer available. Any media packets received for the old connection must be discarded and no new media packets for the stream are sent.

7.2.9.3 Auditing

- It **must** increase system availability, by periodically "pinging" subscribers to minimize time needed to detect an outage.
- It **must** support auditing of endpoints.
- It **must** support auditing of connections.

7.2.10 Requirements from audit endpoint message and parameters

7.2.10.1 Audit endpoint

- It **must** increase system availability, by periodically "pinging" subscribers to minimize time needed to detect an outage.
- It **must** support auditing of connections.
- It **must** support auditing of endpoints.
- The "any of" wildcard convention **must not** be used for endpoint identity.

The "all of" wildcard convention **may** be used to audit a group of endpoints. If this convention is used, the gateway **must** return the list of endpoint identifiers that match the wildcard. It **must** support a maximum number of endpoints to be returned.

- It **must** return all data associated with a connection that might affect resource usage or call usage, possibly including:
  - A list of requested events.
  - A list of signal requests.
  - A request identifier.
  - The notified entity.
  - A list of comma separated connection identifiers.
  - The current value of detect events.
  - The current list of observed events for the endpoint.

- The event corresponding to the state the endpoint is in:
  - A list of protocol versions supported by the endpoint.
  - Capabilities for the endpoint similar to the local connection options parameter and including event packages and connection modes.

7.2.11 Requirements from audit connection message and parameters

7.2.11.1 Auditing connections

- It **must** support auditing of individual endpoints.
- Wildcards **must not** be used to in the endpoint identity.
- If no information was requested, and the endpoint refers to a valid endpoint, the gateway **must** check that the connection specified exists and, if so, returns a positive response.
- It **must** return all relevant data about the endpoint.
- If an connection is queried about a capability it does not understand, the information **must** be omitted from the response.

7.2.12 Requirements from restart in progress message and parameters

7.2.12.1 Restart in progress

- The MG **must** notify the MGC that an endpoint or a group of endpoints is taken out of service or is being placed back in service.
- It **must** identify the endpoints that are taken in or out of service. The "all of" wildcard convention **may** be used to apply the command to a group of endpoints, for example, all endpoints that are attached to a specified interface, or even all endpoints that are attached to a given gateway.
- The "any of" wildcard convention **must not** be used.
- It **must** support an optional "restart delay" parameter is expressed as a number of seconds. To mitigate the effects of a gateway IP address change, the MGC **may** wish to resolve the gateway's domain name by querying the DNS regardless of the time to live of a current resource record for the restarted gateway.
- A list of supported versions **may** be returned if the response indicated version incompatibility.

7.2.12.2 Restart method types

- It **must** support a "graceful" restart, and ensure that the specified endpoint(s) will be taken out of service after the specified "restart delay".
- It **must** support a "cancel-graceful", to ensure that a gateway is cancelling a previously issued "graceful" restart method for the same endpoints.
- It **must** support a "forced" restart method; indicates that the specified endpoints are taken out of service abruptly.
- It **must** support a "restart" method; indicates that service will be restored on the endpoints after the specified "restart delay".
- It **must** support a "disconnected" method indicates that the endpoint has become disconnected and is now trying to establish connectivity.

7.2.12.3 MG restart in progress response

- The MG **should** send a "graceful" or "forced" restart in progress message as a courtesy to the MGC when they are taken out of service, e.g. by being shutdown, or taken out of service by a network management system, although the MGC cannot rely on always receiving such messages.
- The MG **must** send a "restart" restart in progress message with a null delay to their MGC when they are back in service; MGC's **may** rely on receiving this message.
- The MGS **must** send a "disconnected" restart in progress message to their current "notified entity". The "restart delay" parameter **must not** be used with the "forced" restart method.
- The restart in progress message **must** be sent to the current "notified entity" for the endpoint identity in question. A default MGC, i.e. "notified entity", **must** be provisioned for each endpoint so, after a reboot, the default MGC will be the "notified entity" for each endpoint.
- MGs **must** take full advantage of wild-carding to minimize the number of restart in progress messages generated when multiple endpoints in a gateway restart and the endpoints are managed by the same MGC.

7.2.13 J.tgcp: API recovery requirements

7.2.13.1 Autonomous fault detection and recovery

- In order to implement proper call signalling, the MGC **must** keep track of the state of the endpoint.
- The MG **must** make sure that events are properly notified to the MGC.
- Hand-over conflict resolution between separate MGC's is not provided; the MGCs **must** communicate with each other, and **may** use the auditing capabilities to learn about notified entity mapping.
- Special conditions can exist when the gateway or the MGC are restarted: the gateway **may** need to be redirected to a new MGC during "fail-over" procedures; Similarly, the MGC **may** need to take special action when the gateway is taken offline, or restarted: the MGC and MG **must** support the MGC/MG endpoint and connection recovery model.

7.2.13.2 Endpoint/MGC recovery model

- An MGC **must** be identified by its domain name, not network address, for normal call handling.
- Each endpoint **must** have only one MGC associated with it: the notified entity.
- The notified entity (MGC) **must** be pre-provisioned; if empty, the last handling source address must be used.
- Responses to commands **must** always sent to the source (sending) address, despite value of notified entity.
- Endpoints must support autonomously switching between multiple IP addresses; it **must** not autonomously switch between MGCs; it must support the change MGC command.
- If MGC unavailable, endpoint **must** be unavailable until the MGC is available, or until taken over by a backup MGC.
- In the case of a takeover, the original MGC **may** take back the endpoints, or **may** become a new backup MGC.

7.2.13.3 Detection of lost association

- TGCP messages run over UDP, and hence are not guaranteed and subject to loss. In the absence of a timely response commands **must** be repeated.
- The MG **must** keep in memory a list of the responses that they sent to recent transactions, and a list of the transactions that are currently being executed. Recent is here defined by the value T_{thist} that specifies the number of seconds that responses to old transactions **must** be retained; The default value for T_{thist} **must** be 30 seconds, and **must** be administrable.

- The transaction identifiers of incoming commands **must** first be compared to the transaction identifiers of the recent responses. If a match is found, the gateway **must not** execute the transaction, but repeat the old response. If a match to a previously responded to transaction is not found, the transaction identifier of the incoming command is compared to the list of transactions that have not yet finished executing. If a match is found, the gateway **must not** execute the transaction, which is ignored - a response **must** be provided when the execution of the command is complete.

7.2.13.4 Repetition mechanism

- Repetition mechanism **must** be used to protect against:
  - Transmission errors.
  - Components failure.
  - MGC failure.
  - Fail-over (new MGC takes over seamlessly).
- A repetition algorithm **must** detect which of the four failure occur. It needs to into account the differences between the "suspicion threshold" (Max1), and the "disconnection threshold" ("Max2).

7.2.13.5 Repeat transmission algorithm

- The MG **must** always check for the presence of a new MGC.
- If a new MGC is detected, the MG **must** direct retransmissions of any outstanding commands.
- Prior to any retransmission, it **must** check the time elapsed since the sending of the initial datagram.
- If the number of retransmissions to this MGC equals a suspicion threshold threshold (Max1), the MG **may** actively query the name server.
- The MG **may** have several IP addresses for the MGC.
- If there are no more interfaces to try, then the gateway **should** contact the DNS one more time to see if any other interfaces have become available. If not, the MGC must assume the endpoint is disconnected.

7.2.13.6 Repeat algorithm timers

- The MG must support exponentially increasing timers for detecting transmission, failure over, and MGC takeovers and restarts.
- The MG must prevent retransmitted commands from being executed more than once using preset timer values; the sender and receiver must agree on these time values.
- The default value for the suspicion and disconnection values may be altered by the provisioning process.
- The provisioning process must be able to disable one or both of the suspicion threshold (Max1) and disconnect threshold (Max2) that result in DNS queries.

7.2.13.7 Race conditions

- It **must** assume the possibility of race conditions, and **must** support a quarantine list that buffers events for later processing and explicit detection of "desynchronization" the mismatched seizure-state due to glare for an endpoint.
- It **must** assume the order of commands and responses may not be maintained by the transport mechanism, and must support sequence detection and recovery.
- It **must** assume multiple gateways starting at the same time, may cause unstable operation, and must support a graceful start up.

7.2.13.8 Quarantine list accumulation and sending

- A quarantine list must accumulate and buffer events occurring while waiting in transient notification states.
- It must detect events that occur and store them in the quarantine buffer for later processing.
- When the response to the notify command is received, or when a notification request is received and executed successfully, the endpoint must send the list of events and then it must reset the list of observed events of the endpoint to a null value.
- When a new notification request is received in the "notification state", it must support "piggy-backing", placing messages in order with the oldest message first in a single packet to the source of the new notification request, regardless of the source and "notified entity" for the old and new command.

7.2.13.9 Explicit detection and transactional semantics

- The MG **must** check the condition of the endpoint before responding to a notification request to avoid race condition, and return an error if the gateway is requested to notify a "seizure" transition while the circuit is already seized, or if the gateway is requested to notify an "unseize" condition while the circuit is not seized.
- If signal definitions include conditional prerequisites, the gateway **must** return the error specified in the signal definition if the prerequisite is not met.
- The notification request **must** operate as an atomic transaction, and any changes initiated as a result of the command **must** be reverted on an unsuccessful execution.
- All pre-conditions **must** be met from the time the transaction is initiated until the transaction completes; if any of the preconditions change during the execution of the transaction, the transaction **must** fail. Furthermore, as soon as the transaction is initiated, all new events **must** be quarantined until outcome of the transaction is known, whereupon all quarantined events are then processed.

7.2.13.10 Ordering of commands, and treatment of disorder

- It **must** not assume that the underlying transport protocol guarantees the sequencing of commands sent to a gateway or an endpoint.
- It **may** want to provide consistent operation of the endpoints using these rules:
  - Commands pertaining to the different endpoints or connections **may** be sent in parallel.
  - Commands **may** be ignored on a deleted connection, but an error **must** be returned.
  - Only one outstanding notification request **may** be assumed at one time.
  - The MGC **should** individually delete all pending connections at the time of the global delete; new wild card create connection commands for endpoints **should not** be sent until a response to the wild-carded delete connection command is received.
  - Sequencing requirements for commands **must** be adhered to for embedded commands.
  - Audits commands **may not** be subject to any sequencing.
  - Restart in progress **must** always be the first command sent by an endpoint.
  - When multiple messages are piggy-backed in a single packet, the messages **must** be processed in order.
- MGs **must** follow the above rules, but the MG **must not** make any assumptions as to whether MGCs follow the rules or not, and **must** always respond to commands, regardless of whether they adhere to the above rules or not.

7.2.13.11 Restart time shifting

- In order to prevent an "avalanches" of messages after a restart, the following behaviour **must** be followed:
  - When a gateway is powered on, it **must** initiate restart timer to a random value, uniformly distributed between 0 and a provisional maximum waiting delay (MWD), e.g. 360 seconds. Care **must** be taken to avoid synchronicity of the random number generation between multiple gateways that **would** use the same algorithm.
  - The MG then waits for either the end of this timer, the reception of a command from the MGC, or the detection of a local circuit activity, such as for example an seizure transition on a trunking gateway. A pre-existing seizure condition **must** result in the generation of a seizure event.
  - When the restart timer elapses, when a command is received, or when an activity or pre-existing seizure condition is detected, the MG **must** initiate the restart procedure.

7.2.13.12 Restart execution

- Each endpoint **must** send a restart in progress command to the MGC before sending any other message.
- A MG **must** have a provisional default MGC to direct the initial restart message to.
- When the collection of endpoints in a MG is managed by more than one MGC, the above procedure **must** take full advantage of wild-carding to minimize the number of restart in progress messages generated when multiple endpoints.
- The value of the maximum waiting delay (MWD) **must** be a configuration parameter.

7.2.13.13 Disconnected endpoints management

- Endpoints can become disconnected when they fail to communicate with the MGC, and **must** follow a disconnect procedure as follows:
  - A "disconnected" timer is initialized to a random value between 0 and a provisionable "disconnected" initial waiting delay (Td_{init}), Care **must** be taken to avoid synchronicity of the random number generation between multiple gateways and endpoints.
  - The MG then **must** wait for either the end of this timer, the reception of a command from the MGC, or the detection of a local circuit activity for the endpoint.
  - In the case of local user activity, a provisional "disconnected" minimum waiting delay (Td_{min}) **must** furthermore have elapsed since the gateway became disconnected.
  - The endpoint **must** send a restart in progress command to the MGC informing it that the endpoint was disconnected; this **must** be the first command form the endpoint.
  - The endpoint **must** take full advantage of piggy-backing in achieving this. The MGC **may** then for instance decide to audit the endpoint, or simply clear all connections for the endpoint.
  - If the disconnect fails, the "disconnected" timer **must** then be doubled, subject to a provisional "disconnected" maximum waiting delay (Td_{max}), e.g. 600 seconds.
- Not additional behaviour is mandated, but vendors **may** for instance choose to provide silence, play reorder tone, or take other actions on the endpoint.

7.2.13.14 Consistent return and reason codes

- All MGCP commands receive a response, and **must** carry a consistent return code as defined in this specification that indicates the status of the command, with a ranged based categories for response acknowledgement, provisional response, successful completion, transient error, and permanent error.
- Reason-codes **must** be used by the MG when deleting a connection to inform the MGC about the reason for deleting the connection. The reason code is an integer number, defining endpoint malfunctioning, endpoint taken out of service, or loss of lower layer connectivity (e.g. downstream sync).

7.2.14 J.tgcp: Requirements from TGCP not all ready covered by the MGCI API

7.2.14.1 Consistent command structure

- All commands **must** have a consistent structure with a Command header, which for some commands **may** be followed by a session description.
- All responses **must** have a Response header, which for some commands may be followed by a session description.
- Headers and session descriptions **must** be encoded as a set of text lines, which **must** be separated either by a carriage return and line feed character or, optionally, a single line-feed character. The headers **must** be separated from the session description by an empty line.
- MGCP uses a transaction identifier with a value between 1 and 999999999 to correlate commands and responses. The transaction identifier **must** be encoded as a component of the command header and **must** be repeated as a component of the response header.

7.2.14.2 Command Header

- The command header **must** be composed of:
  - a command line identifying the requested action or verb;
  - the transaction identifier;
  - the endpoint towards which the action is requested;
  - the MGCP protocol version;
  - a set of parameter lines composed of a parameter name followed by a parameter value.
- Unless otherwise noted or dictated by other referenced standards, each component in the command header is case insensitive. This goes for verbs as well as parameters and values, and all comparisons **must** treat upper and lower case as well as combinations of these as being equal.

7.2.14.3 Command line

- The command line is composed of:
  - the name of the requested verb;
  - the identification of the transaction;
  - the name of the endpoint(s) that **should** execute the command (in notifications or restarts, the name of the endpoint(s) that is issuing the command);
  - the protocol version.

- These four items are encoded as strings of printable ASCII characters separated by white spaces, i.e. the ASCII space (0x20) or tabulation (0x09) characters. Trunking gateways **should** use exactly one ASCII space separator, however they **must** be able to parse messages with additional white space characters.

7.2.14.4 Requested verb naming

- Requested verbs are encoded as four letter upper- and/or lower-case ASCII codes, and **must** support as a minimum the following:
  - CreateConnection CRCX.
  - ModifyConnection MDCX.
  - DeleteConnection DLCX.
  - NotificationRequest RQNT.
  - Notify NTFY.
  - AuditEndpoint AUEP.
  - AuditConnection AUCX.
  - RestartInProgress RSIP.
- Comparisons must be case insensitive.
- New verbs may be defined in future versions of the protocol.
- A gateway that receives a command with an experimental verb it does not support must return an error.

7.2.14.5 Transition identity handling

- A MG **must** support two separate transaction identifier name spaces, one for sending and one for receiving transactions.
- At a minimum, transaction identifiers for commands sent to a given trunking gateway **must** be unique for the maximum lifetime of the transactions within the collection of MGCs that control that MG.
- Transaction identifiers for all commands sent from a given trunking gateway **must** be unique for the maximum lifetime of the transactions, regardless of which MGC the command is sent to.
- The transaction identifier **must** be encoded as a string of up to nine decimal digits as a value between 1 and 999999999.
- An MGCP entity **must not** reuse a transaction identifier more quickly than three minutes after completion of the previous command in which the identifier was used.

7.2.14.6 Name and protocol version coding

- The endpoint names and MGC names **must** be encoded as e-mail addresses, as defined in IETF RFC 821 [12]. In these addresses, the domain name identifies the system where the endpoint is attached, while the left side identifies a specific endpoint on that system. Both components **must** be case insensitive.
- The name of notified entities **may** be expressed with the same syntax, with the possible addition of a port number. In case the port number is omitted, the default MGCP port (2427) **must** be used.
- The protocol version **must** be coded as the keyword "MGCP" followed by a white space and the version number, which again is followed by the profile name "TGCP" and a profile version number. The version numbers **must** be composed of a major version number, a dot, and a minor version number. The major and minor version numbers are coded as decimal numbers.

- The profile version number defined by this Recommendation is 1.0:
  - The protocol version for this Recommendation **must** be encoded as: MGCP 1.0 TGCP 1.0.
  - An entity that receives a command with a protocol version it does not support, **must** respond with an error.

7.2.14.7 Parameter lines

- Parameter lines **must** be composed of a case insensitive parameter name, which in most cases is composed of a single upper-case character, followed by a colon, a white space, and the parameter value.
- Parameter names **must** adhere to the name code space defined in this specification.
- MGs and MGCs **should** always provide mandatory parameters before optional ones, however MG commands **must not** fail if this Recommendation is not followed.
- Experimental parameters **should** begin by names that begin with the string "X-" or "X+". Parameter names that start with "X+" are mandatory parameter extensions. A MG that receives a mandatory parameter extension that it cannot understand **must** respond with an error. Parameter names that start with "X-" are non critical parameter extensions. A MG that receives a non critical parameter extension that it cannot understand **may** safely ignore that parameter.
- If a parameter line is received with a forbidden parameter, or any other formatting error, the receiving entity **should** respond with the most specific error code for the error in question. Commentary text may always be provided.

7.2.14.8 Requirements on parameters in parameter line

- The following parameters **should** adhere to the name space definitions and conventions of the present document: Response Acknowledgement, RequestIdentifier, Local Connection Options, Capabilities, Connection Parameters, Reason Codes, Connection Mode, Event/Signal Name Coding, Requested events, SignalRequests, Observed events, Requested information, QuarantineHandling, DetectEvents, EventStates, RestartMethod, VersionSupported.
- In addition, the local connection attribute name **must** start with the two characters "x+", for a mandatory extension, or "x-", for a non mandatory extension.
- Extension connection parameters names are composed of the string "X-" followed by a two letters extension parameter name. MGCs that receive unrecognized extensions **must** silently ignore these extensions.

7.2.14.9 Response header

- All responses **must** have a consistent structure with a responses header.
- The response line **must** start with the response code, which is a three-digit numeric value.
- The responses **must** follow the mandatory vs. optional vs. forbidden requirements.
- Final responses **must** be identified.
- Response parameters must adhere to the name space and conventions found in the present document.

7.2.14.10 Retry at the protocol levels

- The retransmission timers **should** estimate the timer by measuring the time spent between sending a command and the return of a response. MGs **must** at a minimum implement a retransmission strategy using exponential back-off with configurable initial and maximum retransmission timer values.
- MGs gateways **should** use the algorithm implemented in TCP-IP, which uses two variables:
  - The average response delay, AAD, estimated through an exponentially smoothed average of the observed delays.
  - The average deviation, ADEV, estimated through an exponentially smoothed average of the absolute value of the difference between the observed delay and the current average.
- The retransmission timer, RTO, in TCP, is set to the sum of the average delay plus N times the average deviation, where N is a constant.

7.2.14.11 Post-retry at the protocol level

- After any retransmission, the MGCP entity **should** do the following:
  - it **should** double the estimated value of the average delay, AAD.
  - it **should** compute a random value, uniformly distributed between 0,5 AAD and AAD.
  - it **should** set the retransmission timer (RTO) to the minimum of:
    - the sum of that random value and N times the average deviation;
    - RTOmax, where the default value for RTOmax is 4 s.
- The initial value used for the retransmission timer **must** be 200 ms by default and the maximum value for the retransmission timer **must** be 4 seconds by default. These default values **may** be altered by the provisioning process.

7.2.14.12 Piggy-backing protocol

- There are cases when an MGC will want to send several messages at the same time to one or more endpoints in a gateway and vice versa. in the same UDP packets; they **must** be separated by a line of text that contains a single dot.
- The piggy-backed messages **must** be processed as if they had been received in separate datagrams, however if a message (command or response) needs to be retransmitted, the entire datagram **must** be retransmitted, not just the missing message. The individual messages in the datagram **must** be processed in order starting with the first message.
- Errors encountered in a message that was piggybacked **must not** affect any of the other messages received in that packet - each message is processed on its own.

7.2.14.13 Transaction identifier sharing

- Transaction identifiers are integer numbers in the range from 1 to 999,999,999. MGCs **may** decide to use a specific number space for each of the gateways that they manage, or to use the same number space for all gateways that belong to some arbitrary group.
- MGC's **may** decide to share the load of managing a large gateway between several independent processes; these processes **must** share the same transaction number space.

- There are multiple possible implementations of this sharing, such as having a centralized allocation of transaction identifiers, or pre-allocating non-overlapping ranges of identifiers to different processes. The implementations **must** guarantee that unique transaction identifiers are allocated to all transactions that originate from any MGC sent to a particular gateway within a period of Tt_{hist} seconds. MG's **must** be able to detect duplicate transactions by looking at the transaction identifier only.

7.2.14.14 Response acknowledgement of confirmed transactions: 3 way handshake

- A response acknowledgement parameter **must** be found in any command. It carries a set of "confirmed transaction-id ranges" for final responses received - provisional responses **must not** be confirmed.
- MGCP gateways **may** choose to delete the copies of the responses to transactions whose id is included in "confirmed transaction-id ranges" received in a message, however the fact that the transaction was executed **must** still be retained for Tt_{hist} seconds.
- When a Response Acknowledgement message is received, the response that is being acknowledged by it **may** be deleted. MGs should discard further commands from that MGC when the transaction_id falls within these ranges, and the response was issued less than Tt_{hist} seconds ago.

7.2.14.15 Transaction confirmation algorithm

- Let termnew and termold be the endpoint-name in respectively a new command, cmdnew, and some old command. cmdold. The transaction-ids to be confirmed in cmdnew should then be determined as follows:

  1) If termnew does not contain any wildcards:
     - Unconfirmed responses to old commands where termold equals termnew.
     - Optionally, one or more unconfirmed responses where termold contained the "any-of" wildcard, and the endpoint-name returned in the response was termnew.
     - Optionally, one or more unconfirmed responses where termold contained the "all" wildcard, and termnew is covered by the wildcard in termold.
     - Optionally, one or more unconfirmed responses where termold contained the "any-of" wildcard, no endpoint-name was returned, and termnew is covered by the wildcard in termold.
  2) If termnew contains the "all" wildcard:
     - Optionally, one or more unconfirmed responses where termold contained the "all" wildcard, and termnew is covered by the wildcard in termold.
  3) If termnew contains the "any of" wildcard:
     - Optionally, one or more unconfirmed responses where termold contained the "all" wildcard, and termnew is covered by the wildcard in termold if the "any of" wildcard in termnew was replaced with the "all" wildcard.

- A given response **should not** be confirmed in two separate messages.
- The "confirmed transaction-id ranges" values **should not** be used if more than Tt_{hist} seconds have elapsed since the gateway issued its last response to that MGC, or when a gateway resumes operation. In this situation, commands **should** be accepted and processed, without any test on the transaction-id.
- A response **should not** be confirmed if the response was received more than Tt_{hist} seconds ago.
- Messages that confirm responses **may** be transmitted and received in disorder. The MG **should not** retain the union of the confirmed transaction-ids received in recent commands.

7.2.14.16 Provisional response

- In some cases, transaction completion times could be significantly longer than otherwise in a network based UDP over IP. A provisional response must therefore be issued.
- If a duplicate create connection or modify connection command is received, and the transaction has not yet finished executing, a provisional response must then be sent back.
- Provisional responses must only be sent in response to a create connection or modify connection command.
- If a session description is returned by the modify connection command, the session description must be included in the provisional response here as well.
- If the transaction completes successfully, the information returned in the provisional response must be repeated in the final response. It is considered a protocol error not to repeat this information or to change any of the previously supplied information in a successful response. If the transaction fails, an error code must be returned as the information returned previously is no longer valid.
- A currently executing create or modify transaction must be cancelled if a delete connection command for the endpoint is received. In that case, a response for the cancelled transaction should still be returned automatically, and a response for the cancelled transaction must be returned if a retransmission of the cancelled transaction is detected.
- To detect endpoint failure, when a provisional response is received, the time-out period for the transaction in question must be set to a significantly higher value for this transaction. The default value of this timer must be 5 s, however the provisioning process may alter this.
- When the transaction finishes execution, the final response must be sent and the by now obsolete provisional response must be deleted. In order to ensure rapid detection of a lost final response, final responses issued after provisional responses for a transaction must be acknowledged. The endpoint must therefore indicate final responses.
- This final response must cause a response acknowledgement to be sent back to the endpoint; this response must never be acknowledged.

7.2.14.17 Security

- If unauthorized entities could use the TGCP, they would be able to set up unauthorized calls or interfere with authorized calls. Security is not provided as an integral part of TGCP. Instead TGCP **must** assume the existence of a lower layer providing the actual security.
- Security requirements and solutions for TGCP are provided in the IPCablecom Security Recommendation J.170 [13], which **should** be consulted for further information.

7.2.15 J.tgcp: Requirements from annex A, event packages

7.2.15.1 Event package overall

- Each package defines a package name for the package and event codes and definitions for each of the events in the package **must** adhere to the encoding and convention of the document.
- Unless otherwise stated, all of the events/signals are detected/applied on endpoints and audio generated by them **must not** be forwarded on any connection the endpoint may have.
- Audio generated by events/signals that are detected/applied on a connection **must** however be forwarded on the associated connection irrespective of the connection mode.

7.2.15.2 Continuity tone (COT) and fax tone

- Two types of COT **must** be supported per ITU-T Recommendation Q.724 [14]:
  - Continuity Tone 1 (co1): A tone at 2 010 Hz m. To conform with current continuity testing practice, the event **should not** be generated until the tone has been removed. The tone is of type TO - the continuity test will only be applied for the specified period of time. The provisioning process **may** alter the default value.
  - Continuity Tone 2 (co2): A tone at 1 780 Hz. To conform with current continuity testing practice, the event **should not** be generated until the tone has been removed. The tone is of type TO - the continuity test will only be applied for the specified period of time. The provisioning process **may** alter the default value.

7.2.15.3 Call tones

Events **must** be generated for:

- Fax tone: whenever a fax communication is detected - see e.g. ITU-T Recommendation T.30 [15], or V.21 [16].
- Modem tones: whenever a modem communication is detected - see e.g. ITU-T Recommendation V.8 [17].
- Reorder tone: Reorder tone, a.k.a congestion tone, as specified in ITU-T Recommendation Q.35 [18].
- Ring back tone: Audible Ring Tone as specified in ITU-T Recommendation Q.35 [18]. The definition of the tone is defined by the national characteristics of the Ring back Tone, and may be established via provisioning. The ring back signal can be applied to both an endpoint and a connection.
- Telecomm Devices for the Deaf tones (TDD): The TDD event is generated whenever a TDD communication is detected - see e.g. ITU-T Recommendation V.18 [19].

7.2.15.4 Operation failure

- The operation failure event **may** be generated when the endpoint was asked to apply one or several signals of type TO on the endpoint, and one or more of those signals failed prior to timing out. The completion report **may** carry as a parameter the name of the signal that failed.
- When the operation failure event is requested, event parameters cannot be specified. When the package name is omitted, the default package name **must** be assumed.
- The operation failure event may additionally be generated when an embedded modify connection command fails.

7.2.15.5 Long duration and media start events

- A long duration event **must** be generated when a connection has been established for more than a certain period of time. The default value is 1 hour, however this **may** be changed by the provisioning process; the event **may** be detected on a connection. When no connection is specified, the event applies to all connections for the endpoint, regardless of when the connections are created.
- A media start event **must** be generated when the first valid RTP media packet is received on the connection. This event **may** be used to synchronize a local signal, e.g. ring back, with the arrival of media from the other party. The event **may** be detected on a connection. When no connection is specified, the event applies to all connections for the endpoint, regardless of when the connections are created.

7.2.16 J.tgcp: Requirements from MF operator services

7.2.16.1 Introduction

- "MF Operator service trunk support" is a requirement in some countries (NA), not necessarily as a replacement to SS7, but as a back up to connect emergency services to the operator if there is a catastrophic failure of the link to the SS7 network; with it you still can have emergency services in the event you are cut off from the outside world.

- This issue is complicated by the fact that the Operator needs access to the phone company's OSS, and may need IPCablecom OSS access as well- these issues are beyond the scope of the present document.
- It is not clear if this or an equivalent is needed in all of Europe or none of Europe or in some countries or none.
- It is an appendix and a separate "package" and is optional.

7.2.16.2 Outgoing operator service package events/signals

It **may** support one-way outgoing MF "Feature Group D" (FGD) Operator Services as a Package, including adhering to the naming convention of the present document and handling the following events and signals:

- *Call answer (ans):* Call Answer occurs at the time of the OSS ANI request, i.e. the call **may not** necessarily have been cut-through to an operator. After Call Answer occurs, facility-hold will be established, i.e. only the OSS can now release the trunk.
- *Fax tone (ft):* The fax tone event is generated whenever a fax call is detected - see e.g. ITU-T Recommendation T.30, or V.21.
- *Long duration connection (ld):* The "long duration connection" is detected when a connection has been established for more than a certain period of time. The default value is 1 hour, however this may be changed by the provisioning process. The event **may** be detected on a connection. When no connection is specified, the event applies to all connections for the endpoint, regardless of when the connections are created.
- *Modem tones (mt):* The modem tone event is generated whenever a modem call is detected - see e.g. ITU-T Recommendation V.8.
- *Operator ringback (orbk):* This event will be generated when the OSS requests that the calling party be alerted. If the calling party is on-hook, ringing will typically be applied, where as reorder tone will typically be applied in case the calling party is off-hook.
- *Reverse make busy (rbz):* This event occurs when the OSS marks the trunk. A release event will be generated when the trunk is no longer busy.
- *Operator recall (rcl):* This signal may be applied to invoke operator recall, e.g. due to customer hook-flash to bring the operator back.
- *Release call (rel):* Release call may be signalled to the media gateway however if facility-hold is established, then the call will not be disconnected until the OSS releases it. The media gateway generates a "release call" event when the OSS is considered to have released the trunk. In this case the event may be parameterized with cause codes indicating the reason for the release:
- *Resume call (res):* This signal indicates that the other party resumed the call, i.e. the party went off-hook.
- *Release complete (rlc):* The endpoint and MGC use the release complete event/signal to confirm the call has been released and the trunk is available for another call.
- *Call set-up (sup(<addr>, <id>)):* Set-up a call to the operator service system using the address and identification information provided. The address information will be on the form.
- *Suspend call (sus):* This signal indicates that the other party suspended the call, i.e. the party went on-hook.
- *Start Wink (swk):* A media Gateway Controller can request the Media Gateway to notify it when the wink start signal occurs.
- *Telecomm Devices for the Deaf tones (TDD):* The TDD event is generated whenever a TDD call is detected - see e.g. ITU-T Recommendation V.15 [20].
- *Operation complete (oc):* The operation complete event is generated when the gateway was asked to apply one or several signals of type TO on the endpoint, and one or more of those signals completed without being stopped by the detection of a requested event such as off-hook transition or dialled digit. The completion report may carry as a parameter the name of the signal that came to the end of its live time. When the operation complete event is requested, it cannot be parameterized with any event parameters. When the package name is omitted, the default package name is assumed.

- *Operation failure (of):* In general, the operation failure event may be generated when the endpoint was asked to apply one or several signals of type TO on the endpoint, and one or more of those signals failed prior to timing out. The completion report may carry as a parameter the name of the signal that failed. When the operation failure event is requested, event parameters cannot be specified. When the package name is omitted, the default package name is assumed.

7.2.16.3 MF terminating protocol package

- It **may** support for busy-line verification (BLV) and operator interrupt (OI) on one-way incoming MF terminating trunks dedicated to BLV and OI as a package.

- When Operator Services are provided by an off-net provider, the OSS **may not** have access to subscriber databases to determine whether BLV and OI **should** be allowed or not.

- It **must** adhere to the naming conventions of the present document.

7.2.17 Future work

- The package construct is used to identify different types of trunks and IP connection: if operators require it, other trunk types can be added by creating new packages within the framework, including ISDN, R1, R2, etc. At the current time, this is not expected.

Annex A: Bibliography

- IETF Internet-Draft M3UA: "SS7 MTP3-User Adaptation Layer (M3UA)". http://www.ietf.org/internet-drafts/draft-ietf-sigtran-m3ua-11.txt
- IETF Internet-Draft SUA: "SS7 SCCP-User Adaptation Layer (SUA)". http://www.ietf.org/internet-drafts/draft-ietf-sigtran-sua-10.txt

History

Document history		
V1.1.1	February 2002	Publication