INTERNATIONAL TELECOMMUNICATION UNION

ITU-T H.248.11

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(11/2002)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Communication procedures

Gateway control protocol: Media gateway overload control package

ITU-T Recommendation H.248.11

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	H.230–H.239
Communication procedures	**H.240–H.259**
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
SYSTEMS AND TERMINAL EQUIPMENT FOR AUDIOVISUAL SERVICES	H.300–H.399
SUPPLEMENTARY SERVICES FOR MULTIMEDIA	H.450–H.499
MOBILITY AND COLLABORATION PROCEDURES	
Overview of Mobility and Collaboration, definitions, protocols and procedures	H.500–H.509
Mobility for H-Series multimedia systems and services	H.510–H.519
Mobile multimedia collaboration applications and services	H.520–H.529
Security for mobile multimedia systems and services	H.530–H.539
Security for mobile multimedia collaboration applications and services	H.540–H.549
Mobility interworking procedures	H.550–H.559
Mobile multimedia collaboration inter-working procedures	H.560–H.569

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation H.248.11

Gateway control protocol: Media gateway overload control package

Summary

This Recommendation describes a package for Media Gateway (MG) Overload Control for use with the H.248.1 Gateway Control Protocol. It serves to protect an MG from processing overload that prevents the timely execution of H.248.1 transactions.

In summary, in this Recommendation, overload protection is achieved as follows:

1) An MG (or virtual MG) detects that it is in overload and notifies its Media Gateway Controller (MGC) of that fact whenever it receives an ADD command.

2) The MGC adaptively throttles the rate it sets up calls using that MG (or virtual MG) to maximize the MG's effective throughput whilst bounding its response times. It does this by throttling the rate at which transactions that set-up new calls or that new call legs are sent to the overloaded MG, so as to cause the rate of overload notifications the MGC receives from the overloaded MG (or virtual MG) to converge to a suitably low level.

A separate instance of the overload control shall be initiated at an MGC for each of its dependent MGs (or virtual MGs) that is overloaded. These separate instances should run independently (that is, they do not explicitly exchange information). Their overload control parameters shall be separately configurable, for example, by means of a proprietary management interface, or the use of SNMP to invoke configuration functions.

The most general overload scenario the control can handle is where one or more MGCs are jointly overloading a single MG that has several virtual MGs (virtual MG 'i' interacting only with MGC "i"). The control does not need to know how many MGCs are causing the MG to be overloaded, nor what the MG capacity is. Informative reference [1] provides a full explanation of one way this can be achieved, and informative reference [2] provides further material on designing overload controls.

The overload control is largely specified by saying how it shall behave, but not how it should be implemented to achieve that behaviour. This has two important consequences.

As a first consequence, part of the package (see 8.5) defines a set of overload scenarios, and any fully-compliant implementation of the package must automatically (i.e. without the need for operator intervention to adjust parameter values from one overload scenario to another) satisfy all the requirements for each of the scenarios.

As a second consequence, not all configurable parameters can be known to this package since they depend upon specific implementations of the control. Nevertheless there is a requirement that the implementation shall provide a means by which an operator can change all parameters which affect the performance of the control. See clause 9 for the Management Requirements associated with this package. It is expected that they will be realised by a proprietary management interface, or the use of SNMP.

Source

ITU-T Recommendation H.248.11 was prepared by ITU-T Study Group 16 (2001-2004) and approved under the WTSA Resolution 1 procedure on 29 November 2002.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

ITU-T Recommendation H.248.11

Gateway control protocol: Media gateway overload control package

1 Scope

This Recommendation describes a package for the H.248.1 gateway protocol related to Media Gateway Overload Control. With the root termination implementing this package, a gateway is expected to report MG_Overload events to a Media Gateway Controller.

The following MGC/MG characteristics are required for this package to work effectively:

1) The applications consist of individual point-to-point calls between two end users, and of calls involving several additional call legs (for example, a conference call where an MG hosts a conference bridge).
2) Point-to-point call set-up using an MG typically involves no more than two ADD commands requested by its MGC.

An example of the applications covered by such an MGC is real-time conversational voice transported using TDM, ATM, IP or Frame Relay.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

2.1 Normative references

– ITU-T Recommendation Q.543 (1993), *Digital Exchange Performance Design Objectives*.

– ITU-T Recommendation Q.714 (2001), *Signalling connection control part procedures*.

2.2 Informative references

[1] Oftel, PNO-ISC Information Document 015 ISUP Overload Controls, Ref. PD 6673:2001.

[2] WHITEHEAD (M. J.) and WILLIAMS (P. M.): Adaptive Network Overload Controls, *BT Technology Journal*, Vol. 20, No. 3, July 2002.

3 Definitions

This Recommendation defines the following terms:

3.1 call: In this package, "Call" means a generic term, meaningful only to MGCs, related the establishment of a path to carry user data between end users via an MG. Normally a qualifier is necessary to make clear the aspect being considered, e.g. call attempt.

3.2 call attempt: In this package, "Call Attempt" means an attempt to set up a call.

3.3 load: In this package, "Load" means the total number of "call attempts" in a given interval of time (i.e. offered load).

This definition is motivated by alignment with performance objectives of ITU-T Rec. Q.543.

3.4 MG overload and MG capacity: In this package, "MG Overload" means that the MG (or virtual MG) is close to being unable to respond to MGC transactions in a sufficiently timely manner to avoid the calling customer abandoning the call in set-up.

This definition is motivated by the need to maximize the calls/second handled by the MG subject to keeping most MG response times short enough to prevent customers from abandoning a call before it is established end-to-end between the calling and the called customers.

In this package, "MG Capacity" means the calling rate (calls/second) at which the MG detects it is in MG Overload. Thus the MG Capacity depends on the specific sequence of H.248.1 commands (ADD, MODIFY, etc.) which implement the calls, as well as on the processing power of the MG.

The precise way MG Overload is detected is likely to be implementation-specific, since different suppliers' switches are likely to have different architectures. Overload detection could, for example, be based on the use of processor occupancy thresholds, or queue thresholds or internal delay thresholds. Each overload detection scheme would have an associated set of configurable parameters, which however cannot in general be known to this package. Despite this, an operator will need to be able to configure an MG's overload detection scheme to ensure suitably short response times under overload. It is therefore necessary for the MG/MGC implementation to provide a means for an operator to configure such parameters via, for example, a proprietary management interface, or the use of SNMP, not the H.248.1 interface. Clause 9 gives the management requirements for this package.

There may be MG resources (e.g. codecs, ATM bandwidth, tone generators) whose congestion, although preventing a call from being set up, does not prevent timely response of the MG to MGC transactions. Congestion of those resources must not cause MGC overload control (as defined in this package) to be invoked. The reasons for this are that:

a) congestion of such resources does not (by definition) prevent timely response to MGC transactions; and

b) it is probably more appropriate to flag such congestion via existing H.248.1 error messages in order to trigger a capacity upgrade of the congested resource.

3.5 leaky bucket: This package requires the MGC to restrict the rate it admits calls to an MG which has reported that it is congested. Any one of the following 3 types of leaky bucket is acceptable for use as a call restrictor with this package, provided its implementation can be configured to cover the required range of admitted rates implied by the set of overload control scenarios given in 8.5. See clause 9 for an approximate performance analysis.

Type 1 Leaky Bucket. This is a count which decreases by a configurable LeakAmount every LeakInterval (subject to not falling below 0) and which increases by a configurable SplashAmount at each call arrival (subject to not exceeding the configurable MaximumFill count). A call which arrives to find the count less than or equal to the {MaximumFill – SplashAmount} is admitted (and the count is incremented by SplashAmount); otherwise the call is rejected and the count is not incremented. The maximum sustained admitted rate is approximately (LeakAmount/SplashAmount)/LeakInterval provided that LeakAmount $\leq$ MaximumFill. The LeakInterval is adaptively changed by the overload control.

Type 2 Leaky Bucket. This is a count that is decremented by (Now – TimeOfLastDecrement) × LeakAmount/LeakInterval at each call arrival instant (subject to not falling below 0). It is then incremented by SplashAmount if the count is less than or equal to the {MaximumFill – SplashAmount} and the call is admitted; otherwise the call is rejected and the count is not incremented. The maximum sustained admitted rate is (LeakAmount/SplashAmount)/LeakInterval provided that LeakAmount $\leq$ MaximumFill. The LeakInterval is adaptively changed by the overload control.

Type 3 Leaky Bucket. This is a count that is decremented by LeakAmount every LeakInterval (subject to not falling below 0). LeakInterval is an operator-configurable fixed period; LeakAmount is adaptively changed by the overload control. A call which arrives to find the count less than or equal to the {MaximumFill – SplashAmount} is admitted (and the count is incremented by SplashAmount); otherwise the call is rejected and the count is not incremented. The maximum sustained admitted rate is (LeakAmount/SplashAmount)/LeakInterval provided that LeakAmount ≤ MaximumFill.

For all three bucket types, SplashAmount and LeakAmount shall not exceed MaximumFill.

The use of leaky bucket restrictors is widespread in telephony switches. They are simple to implement, and are preferable to proportional discard because they bound the maximum sustained admitted rate irrespective of the offered rate of calls.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations:

ATM Asynchronous Transfer Mode

IP Internet Protocol

MG Media Gateway

MGC Media Gateway Controller

OCP Overload Control Package

SCCP Signalling Connection Control Part

SNMP Simple Network Management Protocol

TDM Time Division Multiplex

5 Overload Control Package

Package Name: OCP

PackageID: ocp, 0x0051

Description:

The package makes it possible for an MG (or virtual MG) to control its load so that it can process in a timely manner all transactions received from its MGC whilst maximising the effective throughput (in calls/second) of the overloaded MG.

The event in this package may be provisioned in the MG.

The event in this package may only be applied to the Root termination.

Version: 1

Extends: None

5.1 Properties

None.

5.2 Events

5.2.1 MG_Overload

Event name: MG_Overload

EventID: mg_overload, (0x0001)

Description:

This event occurs only when the MG (or virtual MG) receives an ADD command from an MGC and the MG has determined it is overloaded. The event is ordered by the MGC or provisioned.

EventsDescriptor Parameters:

None

ObservedEvetsDescriptor Parameters:

None

6 Signals

None.

7 Statistics

None.

8 Procedures

8.1 Actions at overloaded MG (or virtual MG)

An MG (or virtual MG) shall be capable of detecting when it is in overload ("MG Overload" is defined in 3.4).

An overloaded MG (or overloaded virtual MG) that receives an ADD command from an MGC shall

a) continue normal processing of that transaction; and

b) as soon as possible, notify the MGC that it is overloaded (by sending a Notify Request with event "MG_Overload" to the MGC).

NOTE – For the applications listed in the Scope of this Recommendation, there will usually be just two ADD commands from the MGC per call. So there will usually be at most two such MG_Overload notifications per call, and they will occur early in the sequence of transactions exchanged between the MG and its MGC. Moreover, the rate at which the overloaded MG returns such indications will converge to a configurable low level (the TargetMG_OverloadRate, see 8.2.3) – less than one per second. Therefore, sending "MG_Overload" transactions should not impose a significant processing or transmission overhead.

8.2 Actions at an MGC

8.2.1 Control initiation at an MGC

Control at an MGC shall be activated (as quickly as possible) towards an MG when:

a) the rate of MG_Overloads per second the MGC receives from the MG exceeds the TargetMG_OverloadRate (see 8.2.3); and

b) no control is currently activated from the MGC to the MG.

When control is activated the bucket count shall be initialised to an operator-configurable value (InitialFill).

NOTE 1 – This enables the operator to limit the initial burst of admitted calls, by setting the InitialFill of the bucket close to or equal to its maximum.

For type 1 and type 2 leaky bucket restrictors, the LeakInterval shall be initialised to an operator-configurable value (InitialLeakInterval) when control is activated.

For type 3 leaky bucket restrictors, the LeakAmount shall be initialised to an operator-configurable initial value (InitialLeakAmount) when control is activated.

The range of values for the initial restriction level (i.e. InitialLeakInterval for a type 1 or type 2 bucket, and InitialLeakAmount for a type 3 bucket) shall allow an operator to set the initial admitted rate to be as low or high as is required.

NOTE 2 – This allows the control to respond rapidly to the initial surge of calls in a likely worst-case overload event.

8.2.2 Call restriction method at an MGC

To choose which calls to block when overload control has been initiated for an MG, the MGC control instance shall use one of the 3 types of Leaky Bucket restrictors defined in 3.5, or a leaky bucket implementation that is equivalent to one of those 3 types.

NOTE – "Equivalent" means "admits exactly the same calls and rejects exactly the same calls as", when offered an arbitrary sequence of call arrival instants.

If an overload control instance at an MGC admits a call set-up request, then it must admit all subsequent Call Control signalling messages, and consequential H.248.1 transactions, relating to that call. Under MG overload this gives priority to subsequent call processing for an admitted call over new calls.

8.2.3 Adaptation of admitted calling rate at an MGC

The overload control instance at an MGC protecting an MG (or virtual MG) shall adapt the rate it admits calls to the MG (the "admitted rate") so as to cause the rate it receives "MG_Overload" transactions to converge to an operator-configurable TargetMG_OverloadRate (defined as MG_Overloads/second).

NOTE 1 – This requirement ensures that the total calls/second offered to the overloaded MG, possibly from many MGCs, will converge close to the overloaded MG's capacity, provided that the (per MGC control instance) TargetMG_OverloadRate is small (i.e. less than 1 per second). A full explanation may be found in [1] and [2]. Those references show that such convergence occurs for a wide range of MG capacities, and a wide range in the number of MGCs.

NOTE 2 – It is important to note that controlling the reject rate in this way means that, in the steady-state, the share of the overloaded MG's capacity that an MGC with an active control gets is in direct proportion to the value of its TargetMG_OverloadRate. In particular, if all MGCs have equal TargetMG_OverloadRates then the overloaded MG's capacity will be divided equally among them. See references [1] and [2].

NOTE 3 – It is desirable that changes to the admitted rate at an MGC should become progressively larger (or that changes be made more frequently) as the MG_Overload rate it detects departs further from its TargetMG_Overload Rate. This is to ensure a rapid response to sudden changes (increases or decreases) in the offered calling rate, or MG capacity, or number of MGCs causing the MG to be in overload, whilst still maintaining control stability.

NOTE 4 – For type 1 and type 2 leaky bucket restrictors, the number of admitted calls/second is changed by adjusting the LeakInterval. For type 3 leaky bucket restrictors, the number of admitted calls/second is changed by adjusting the LeakAmount.

NOTE 5 – Under constant offered load, convergence close to the steady-state means that the number of admitted calls/second offered to an overloaded MG varies by no more than a small fraction (10 to 20%) of the MG's calls/second capacity. This implies that the control variable (i.e. the LeakInterval for type 1 or 2 buckets, and the LeakAmount for type 3 buckets) needs to be capable of fairly small changes. For example, for a type 1 or 2 bucket, the LeakInterval will need to be able to change by as little as 1/5 of the minimum

required LeakInterval in order to keep variation in offered rate to within 20% of the MG's capacity. See 9.1 for an analysis of the control variable range and granularity that are required by the range of scenarios specified in 8.5.

If an MGC has several active overload control instances, one per overloaded dependent MG (or virtual MG), then they shall operate completely independently.

8.2.4 Termination of control at an MGC

Control of load towards an MG (or virtual MG) shall be terminated at an MGC only when both

a) the rate the MG sends MG_Overloads to the MGC; and

b) the rate the MGC leaky bucket restrictor rejects calls to the MG

have been zero for a sufficiently long period (the TerminationPendingPeriod, measured in seconds) to indicate that the overload has abated.

NOTE – This is essential to prevent the control repeatedly and rapidly ending and restarting (at its initial, potentially low, admitted rate) where an MG is only slightly overloaded.

8.2.5 Use of priorities at an MGC

H.248.1 provides an optional 16 priority levels for contexts, numbered from 0 (the lowest priority) to 15 (the highest), and an optional emergency indicator for contexts. This Recommendation regards the emergency indicator as an additional priority level taking precedence over the 16 priority levels, giving an expanded set of priority levels.

The use of priorities specified here is based on the SCCP traffic limitation mechanism described in 2.6/Q.714 and 5.2.4/Q.714.

The basic idea is that the control at an MGC should, in the steady state, admit (and hence offer to the MG) a calling rate close to the MG's capacity by rejecting as many of the lowest priority calls as is required to drive the MG_Overload rate close to the TargetMG_OverloadRate; and if that is not possible, by rejecting all lowest priority calls and some (possibly all) of the next highest priority calls, and so on. Eventually, it will reach a steady state in which all calls with priorities 0, 1, … , P-1 are rejected (for some priority level P), some of the priority P calls are rejected and none of the higher priority calls (if any) are rejected. Priority P is called the HighestControlledPriorityLevel. (Capitalized P is used to distinguish it from lower-case p used later in this clause to denote an arbitrary priority level.)

This idea is illustrated in Figure 1, which plots the total calls/second admitted by a single MGC and offered to an MG as a function of time. The figure assumes that calls with priority levels 0, 1 and 2 are jointly overloading an MG. Their offered calling rates (before restriction at the MGC) are displayed in a stacked way against the vertical axis. Also indicated on that axis is the MG's calls/second capacity. In this example, it lies between the priority 2 offered rate and the sum of the priority 1 and 2 offered rates. In the steady-state, it should accordingly reject all priority 0 calls, roughly half the priority 1 calls, and none of the priority 2 calls.

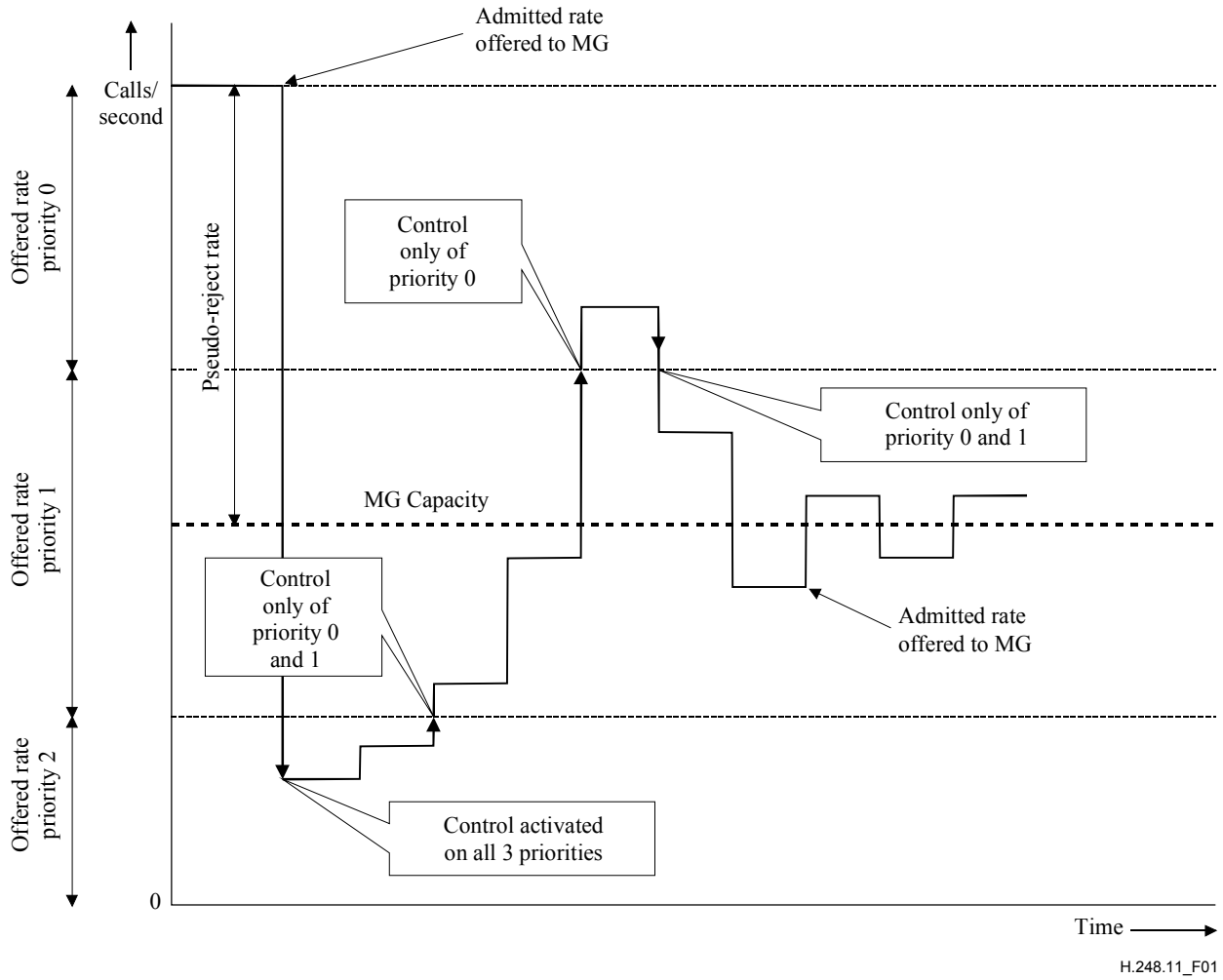

Figure 1/H.248.11 – Example of control using priorities

In this example, when the overload starts, the MG_Overload rate (which approximately equals the offered calling rate minus the MG's capacity) is large enough to activate overload control at the MGC. The control starts with a HighestControlledPriorityLevel initialized to a (configurable) InitialHighestControlledPriorityLevel, which is set to 2 in this example. The associated leaky bucket monitor is initialized and activated. Any call set-up request of priority less than the HighestControlledPriorityLevel is rejected by the control. Call set-up requests with priority equal to the HighestControlledPriorityLevel are offered to the active leaky bucket restrictor to determine whether they should be admitted or not.

In this example, the control over-restricts initially (so no MG_Overload notifications are returned by the MG), and the admitted rate of the restrictor adaptively increases up to its maximum, at which point all priority 2 calls are being admitted, but the MG_Overload rate is still too low. So the HighestControlledPriorityLevel is reduced by 1 and the associated leaky bucket restrictor has its admitted rate and count initialized (to their respective configurable minimum and maximum values to correspond to maximal priority 1 restriction).

The restrictor continues (in this example) to adaptively increase its admitted rate (with the HighestControlledPriorityLevel held at 1) and eventually it reaches its configurable maximum, at which point the control reduces the HighestControlledPriorityLevel to 0, and reinitializes and reactives the leaky bucket restrictor as before.

At this point, in the example, the control is under-restricting and it receives MG_Overload notifications. These cause the MGC overload control instance to increase restriction by adaptively reducing the leaky bucket admitted rate to its minimum (representing maximal restriction of priority

0 calls). At that point, the MG_Overload rate is still too high, so the HighestControlledPriorityLevel is increased to 1 and the leaky bucket has its admitted rate and count reinitialized (to their respective configurable maxima to correspond to minimal priority 1 restriction).

Thereafter, in this example, the control keeps the HighestControlledPriorityLevel constant at 1 (thus rejecting all priority 0 calls and some of the priority 1 calls) and adapts the leaky bucket's admitted rate so as to cause the MG_Overload rate to converge close to its target.

To summarize the use of priority levels:

a) the control aims to adapt its admitted rate so that the MG_Overload rate converges close to the TargetMG_OverloadRate;

b) it always rejects lower priority calls before higher ones;

c) it has a configurable InitialHighestControlledPriorityLevel to invoke effective control quickly without having to adapt through intermediate priority levels first; and

d) only one leaky bucket restrictor is needed however many priority levels are in use.

The above example served to explain the ideas underlying the following text that states the package's requirements.

The HighestControlledPriorityLevel shall take integer values in the range MinimumHighestControlledPriorityLevel to MaximumHighestControlledPriorityLevel (both shall be operator configurable).

When control is activated at an MGC towards a specific MG, then the HighestControlledPriorityLevel shall be set equal to the value of the (operator-configurable) InitialHighestControlledPriorityLevel, and a leaky bucket restrictor shall be activated and initialized. If the restrictor is of type 1 or 2, then its count shall be set to InitialFill, and its LeakInterval shall be set to InitialLeakInterval. If the restrictor is of type 3, then its count shall be set to InitialFill, and its LeakAmount shall be set to InitialLeakAmount.

When a call set-up request of priority "p" arrives at an active MGC control, then it shall be dealt with as follows:

a) if p < HighestControlledPriorityLevel, the call is rejected;

b) if p = HighestControlledPriorityLevel, the call is offered to the active leaky bucket restrictor to determine whether it is admitted or not;

c) if p > HighestControlledPriorityLevel, the call is admitted.

For each value of HighestControlledPriorityLevel, the control shall adapt the leaky bucket's admitted rate so as to cause the rate it receives MG_Overloads to converge towards the TargetMG_OverloadRate.

The control shall adapt the HighestControlledPriorityLevel so as to cause the rate its MGC receives MG_Overloads to converge towards the TargetMG_OverloadRate. Specifically:

a) Increasing the HighestControlledPriorityLevel. If the MG_Overload rate exceeds the TargetMG_OverloadRate (i.e. the control is under-restricting) and the leaky bucket's admitted rate equals its (configurable) minimum rate (so it is maximally restricting calls of priority level HighestControlledPriorityLevel), then the value of the HighestControlledPriorityLevel shall be increased (subject to not exceeding the MaximumHighestControlledPriorityLevel), the count of the leaky bucket shall be set to MaximumFill, and the LeakInterval (respectively LeakAmount) shall be set to its (configurable) MinimumLeakInterval (respectively MaximumLeakAmount) corresponding to minimal restriction at the new HighestControlledPriorityLevel;

b) Decreasing the HighestControlledPriorityLevel. If the MG_Overload rate is less than the TargetMG_OverloadRate (i.e. the control is over-restricting) and the leaky bucket's

admitted rate equals its (configurable) maximum rate (so it is minimally restricting calls of priority level HighestControlledPriorityLevel), then the value of the HighestControlledPriorityLevel shall be reduced (subject to not falling below the MinimumHighestControlledPriorityLevel), the count of the leaky bucket shall be set to its (configurable) MaximumFill, and the LeakInterval (respectively LeakAmount) shall be set to its (configurable) MaximumLeakInterval (respectively MinimumLeakAmount) corresponding to maximal restriction at the new HighestControlledPriorityLevel.

8.3 Bounding MG response times in the steady-state

It shall be possible to configure:

a) MG overload detection; and

b) MGC restriction

so that the 95th percentile of the response times to call set-up requests whose H.248.1 transactions are handled at an overloaded MG (or virtual MG) shall not exceed a time TargetMG_ResponseTime milliseconds in the steady-state attained by the control.

The response time is defined as the time from the arrival of a call set-up request until the response (see Figure 2). (For example, for ISUP the response time is from the arrival of the IAM until the associated IAM is forwarded or ACM or REL returned.)

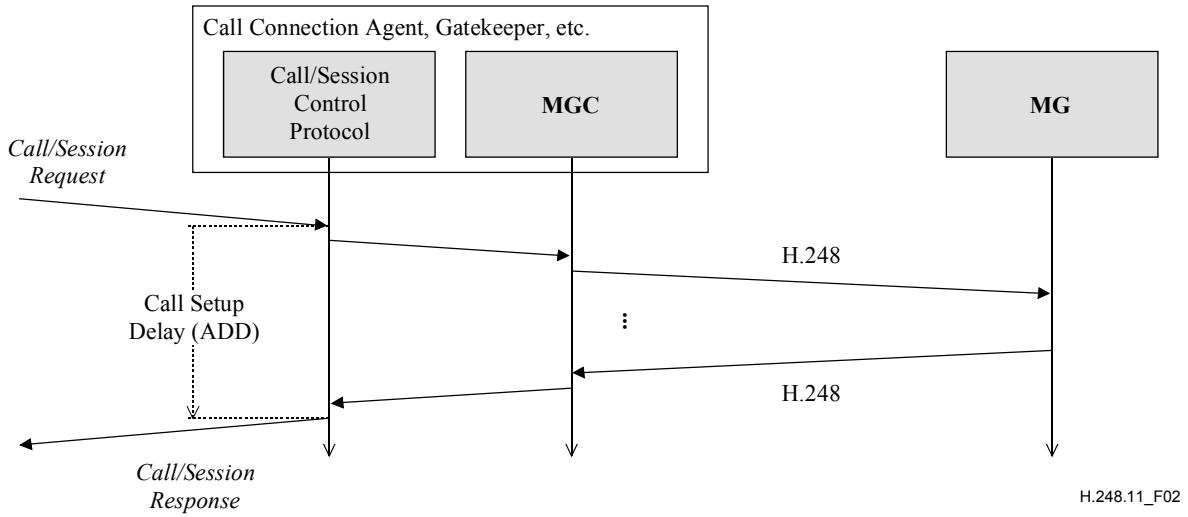

Figure 2/H.248.11 – Response time

NOTE 1 – The TargetMG_ResponseTime should be short enough to reduce the likelihood of customers abandoning calls during set-up. The suggested range is from 0 milliseconds to 100 milliseconds.

NOTE 2 – This requirement serves a dual purpose: it ensures that, when the control has settled down to a steady state, customers do not abandon a call before it is set-up due to long call set-up times (this is necessary to limit customer-initiated repeat attempts). This requirement also limits the round-trip delay from MGC to overloaded MG to MGC, which improves the stability of the overload control.

8.4 Bounding offered rates to MG during initial overload transient

It shall be possible to configure MG overload detection and MGC restriction, so that during the initial transient response of the control (i.e. prior to the steady state being reached) the total calling rate admitted by the active MGC restrictors and offered to the overloaded MG does not much exceed the number of calls/sec at which it detects an MG overload, as measured in consecutive short measurement periods (for example, of duration 1 second) during the initial transient response.

The admitted calling rate (in calls/second) is defined to be the number of call-setups admitted in the measurement period.

NOTE – This requirement ensures that the overload control reacts fast enough to prevent the load offered to the overloaded MG from dangerously exceeding the MG's capacity.

8.5 Range of overload scenarios

This package has been designed for overload scenarios in real-time speech services characterized by:

- number of MGCs overloading an MG in the range 1 to 10;
- a "step increase" in the total load offered to the MGCs (and destined for the overloaded MG) from 0 to 5 times the calls/second capacity of the MG, the load then remaining at that high level for 20 minutes;
- a fast "ramp increase" to 5 times the calls/second capacity of the MG over a period of duration 20 seconds in the total load offered to the MGCs (and destined for the overloaded MG) followed by a slower ramp decrease over an period of duration 10 minutes;
- any distribution of the total offered load among MGCs;
- overloaded MG capacities in the call load range 50 calls/second to 500 calls/second.

NOTE – The control may also be applicable to other overload scenarios.

9 Management requirements for H.248.11

This clause collects together all the management requirements for the package defined in this Recommendation. The bulk of the requirements relate to providing an operator of an MG/MGC with the means to configure all overload control parameters used at an MGC and an MG. It is expected that these requirements will be realized either by a proprietary management interface, or by the use of SNMP.

An important peculiarity of H.248.11 is that not all the overload control parameters can be known in advance. This is because the package does not specify in complete detail how the control should be implemented. Rather, it specifies its behaviour, for example, that an MGC shall adapt the admitted rate it offers to an MG so that the rate of MG_Overload notifications/second it receives from that MG converges to the TargetMG_OverloadRate. Consequently, some of the parameters can only be known when an implementation for the control is finalised.

9.1 An approximate performance analysis of leaky bucket restrictors

This clause gives an approximate performance analysis of the type 1 and type 3 of leaky buckets allowed by H.248.11. It should help an implementor of the control to decide:

a) which bucket type to use; and

b) what factors affect the range and granularity required by the various configurable parameters.

Consider the general scenario in which N MGCs (numbered 1 … N) are jointly overloading a single MG of capacity C calls/second.

Define the following notation:

γ_i = number of calls/second admitted by the active control at MGC i, and hence offered to the MG

m_i = MaximumFill at MGC i

l_i = LeakAmount at MGC $i \leq m_i$

s_i = SplashAmount at MGC $i \le m_i$

t_i = LeakInterval at MGC i (seconds)

Over a large number of leak intervals, the average rate at which a bucket's fill increases is given by $\gamma_i s_i$ per second, since every admitted call causes the fill to increase by the SplashAmount. Also, in every leak interval, the bucket fill leaks by l_i (or less, depending upon the fill level at the leak instant). Consequently, the true average rate at which the bucket fill decreases is $\le \frac{l_i}{t_i}$ per second. Hence, we have the following upper bound on the long-run admitted rate:

$$\gamma_i \le \frac{l_i}{s_i t_i}$$

We now simplify things by assuming that the calls/second offered to MGC i (before restriction) is large enough for the maximum sustained admitted rate to be attained. In that case, a simulated comparison of the true admitted rate with the upper bound reveals that the bound is less than $1.1\gamma_i$ for 75% of randomly chosen buckets. We therefore accept as an approximation that

$$\gamma_i = \frac{l_i}{s_i t_i}$$

Now assume that all N MGCs have the same value of TargetRejectRate. Then it follows [1] and [2] that each MGC gets an equal share of the overloaded MG's calls/second capacity. That is,

$$\gamma_i = \frac{l_i}{s_i t_i} = \frac{C}{N}$$

provided that the value of MaximumFill is greater than or equal to the LeakAmount: $m_i \ge l_i$.

This is true for each MGC and holds whether the bucket is of type 1 or 3.

There are now two cases to analyse. Case 1 has a type 1 leaky bucket at MGC i whose control variable is the LeakInterval, and case 2 has a bucket at MGC i whose control variable is the LeakAmount (i.e. a type 3 bucket). In each case, we deduce the expressions for the relevant configurable parameters.

Case 1: **Control Variable = LeakInterval**

For this case, in the steady-state, the LeakInterval is given by:

$$t_i = \frac{l_i}{s_i} \cdot \frac{N}{C}$$

We require that a small change dt_i in the LeakInterval, shall result in a suitably small absolute change in the admitted rate. That is:

$$\left| d\gamma_i \right| \le f \cdot \frac{C}{N}$$

where f is a fraction (say 20%). This will hold if

$$\left| \frac{d\gamma_i}{dt_i} \right| \cdot dt_i = \left| \frac{-l_i}{s_i t_i^2} \right| \cdot \left| dt_i \right| \le f \cdot \frac{C}{N}$$

That is, if

$$\left| dt_i \right| \le f \cdot t_i$$

From this we can deduce the minimal necessary change in the LeakInterval:

$$\min |dt_i| = f \cdot \min t_i = f \cdot \frac{l_i}{s_i} \cdot \frac{\min N}{\max C} = f \cdot \frac{l_i}{s_i} \cdot \frac{1}{500}$$

It makes sense to take this quantity equal to the shortest accurate clock interval available at MGC i, which we denote by τ_i. Typically τ_i will lie in the range 10 to 100 milliseconds. So, we get the following simple formulae for the required LeakInterval granularity, and the minimum and maximum required values for the LeakInterval:

$$\min |dt_i| = \tau_i$$

$$\min t_i = \frac{l_i}{s_i} \cdot \frac{\min N}{\max C} = \frac{l_i}{s_i} \cdot \frac{1}{500} = \frac{\tau_i}{f}$$

$$\max t_i = \frac{l_i}{s_i} \cdot \frac{\max N}{\min C} = \frac{l_i}{s_i} \cdot \frac{10}{50} = 100 \cdot \frac{\tau_i}{f}$$

(The numerical values in these equations come from 8.5.)

Finally, since the MaximumFill must not be less than the LeakAmount, i.e. $m_i \geq l_i$, we deduce that the maximum number of calls that can be admitted per LeakInterval, namely $\frac{m_i}{s_i}$, must satisfy the inequality:

$$\frac{m_i}{s_i} \geq \frac{l_i}{s_i} = 500 \cdot \frac{\tau_i}{f}$$

To illustrate these results, we assume that $f = 0.2$ and tabulate 4 cases.

Table 1/H.248.11 – Case 1: Control variable = LeakInterval

Shortest clock period τ_i (seconds)	**LeakInterval granularity $\|dt_i\|$ (seconds)**	**Min LeakInterval min t_i (seconds)**	**Max LeakInterval max t_i (seconds)**	**Max calls admitted per LeakInterval $\max \frac{l_i}{s_i}$**
0.002	0.002	0.01	1	5
0.01	0.01	0.05	5	25
0.05	0.05	0.25	25	125
0.1	0.1	0.5	50	250

The table reveals a potentially serious problem for larger values of the clock period. The bucket will tend to admit a large burst of calls at the start of each LeakInterval for the longer clock periods (50 and 100 milliseconds), irrespective of the duration of the leak interval. (Even with a clock period of 10 milliseconds, a high offered calling rate will lead to a burst of 25 calls at the beginning of each leak interval). That could lead to very bursty call arrivals at the overloaded MG, which might reduce the effectiveness of the overload control.

Case 2: **Control Variable = LeakAmount**

For this case, in the steady-state, the LeakAmount is given by:

$$l_i = t_i s_i \cdot \frac{C}{N}$$

We require that a small change dl_i in the LeakAmount, shall result in a small absolute change in the admitted rate. That is

$$\left|d\gamma_i\right| \le f \cdot \frac{C}{N}$$

where f is a fraction (say 20%). This will hold if

$$\left|\frac{d\gamma_i}{dl_i}\right| \cdot dl_i = \left|\frac{1}{s_i t_i}\right| \cdot \left|dl_i\right| \le f \cdot \frac{C}{N}$$

That is, if

$$\left|dl_i\right| \le f \cdot l_i$$

From this we can deduce the minimal necessary change in the LeakAmount (the LeakAmount granularity):

$$\min\left|dl_i\right| = f \cdot \min l_i = f s_i t_i \cdot \frac{\min C}{\max N} = f s_i t_i \cdot \frac{50}{10} = 5 f s_i t_i$$

Taking the LeakInterval t_i as equal to the shortest clock period τ_i at MGC i, we get the following simple formulae for the required LeakAmount granularity, and the minimum and maximum required values of LeakAmount:

$$\min\left|dl_i\right| = 5 f s_i \tau_i$$

$$\min l_i = s_i \tau_i \cdot \frac{\min C}{\max N} = s_i \tau_i \cdot \frac{50}{10} = 5 s_i \tau_i$$

$$\max l_i = s_i \tau_i \cdot \frac{\max C}{\min N} = s_i \tau_i \cdot \frac{500}{1} = 500 s_i \tau_i$$

Finally, since the MaximumFill must not be less than the LeakAmount, i.e. $m_i \ge l_i$, we deduce that the maximum number of calls that can be admitted per LeakInterval, namely $\frac{m_i}{s_i}$, must satisfy the inequality:

$$\frac{m_i}{s_i} \ge \max \frac{l_i}{s_i} = 500 \tau_i$$

To illustrate these results, we assume that $f = 0.2$, (somewhat arbitrarily) choose the SplashAmount $s_i = 100$, and tabulate 4 cases.

Table 2/H.248.11 – Case 2: Control variable = LeakAmount

Shortest clock period τ_i (seconds)	LeakAmount granularity $\|dl_i\|$ (seconds)	Min LeakAmount min l_i (seconds)	Max LeakAmount min l_i (seconds)	Max calls admitted per LeakInterval $\max \frac{l_i}{s_i}$
0.002	0.2	1	100	1
0.01	1	5	500	5
0.05	5	25	2500	25
0.1	10	50	5000	50

Table 2 reveals an improvement upon the type 1 leaky bucket (compare with Table 1). Namely, the maximum number of calls admitted per LeakInterval is now 1/5 that of the type 1 leaky bucket. For example, for a clock period of 50 milliseconds the type 3 bucket admits at most 25 calls per burst (and only for the highest required admitted rates), whereas, for the same clock period, the type 1 bucket admits at most 125 calls per LeakInterval, over the whole range of admitted rates.

The main conclusion from the foregoing analysis is that a control implementor should do this kind of performance study (and more!) when designing an implementation of the H.248.11 control.

Another conclusion is that the availability of short clock periods (ideally < 50 milliseconds) is an important part in meeting the requirements of H.248.11.

A third conclusion is that type 3 buckets may have significant advantages over type 1 buckets, since they do not rely totally on short clock periods to achieve the required control variable granularity.

9.2 Configuration of leaky bucket restrictors at MGC

If a type 1 or a type 2 leaky bucket is used, then the parameters MaximumFill, LeakAmount and SplashAmount shall be operator-configurable by means of a proprietary management interface, or by means of the use of SNMP. The implementor shall ensure that the range and granularity of each of these parameters are configurable and sufficient to meet H.248.11 requirements over the complete range of overload scenarios defined in this Recommendation.

If a type 3 leaky bucket is used, then the parameters MaximumFill, LeakInterval and SplashAmount shall be operator-configurable by means of a proprietary management interface, or the use of SNMP. The implementor shall ensure that the range and granularity of each of these parameters are configurable and sufficient to meet H.248.11 requirements over the complete range of overload scenarios defined in this Recommendation.

If a leaky bucket restrictor equivalent to one of 3 types is employed by an implementation of the overload control, then the parameters equivalent to MaximumFill, LeakAmount, LeakInterval, and SplashAmount shall be operator-configurable by means of a proprietary management interface, or through the use of SNMP. The implementor shall ensure that the range and granularity of each of these parameters are configurable and sufficient to meet H.248.11 requirements over the complete range of overload scenarios defined in this Recommendation.

NOTE – "Equivalent" means "admits exactly the same calls and rejects exactly the same calls as", when offered an arbitrary sequence of call arrival instants.

If prioritized restrictors are provided, then the InitialHighestControlledPriorityLevel, MinimumHighestControlledPriorityLevel and MaximumHighestControlledPriorityLevel shall be operator-configurable either by means of a proprietary management interface, or through the use of SNMP.

9.3 Configuration of proprietary parameters relating to overload detection at an MG

Implementations shall provide a set of parameters relating to detection of MG Overload which (together with the configurable parameters relating to call restriction) enable an operator to configure MG overload detection so that the MG requirements specified in 8.3 are met. These (implementation-specific) parameters shall be operator-configurable either via a proprietary management interface, or through the use of SNMP.

9.4 Configuration of proprietary parameters relating to control activation at an MGC

If the control uses a type 1 or type 2 leaky bucket restrictor, then the InitialLeakInterval (seconds) and InitialFill applied at an MGC when an overload control instance is activated shall be configurable by the operator via a proprietary management interface, or through the use of SNMP. The implementation shall ensure that the range and granularity of these parameters are configurable and sufficient to meet H.248.11 requirements over the complete range of overload scenarios defined in this Recommendation.

If the control uses a type 3 leaky bucket restrictor, then the InitialLeakAmount and InitialFill applied at an MGC when an overload control instance is activated shall be configurable by the operator via a proprietary management interface, or through the use of SNMP. The vendor shall ensure that the range and granularity of these parameters are configurable and sufficient to meet H.248.11 requirements over the complete range of overload scenarios defined in this Recommendation.

Implementations shall provide a means for an operator to configure any other parameters relating to control activation, whose adjustment may be necessary to meet the requirements of this Recommendation. The implementation shall specify the configurable range and configurable granularity of each such parameter.

The (implementation-specific) parameters relating to control activation shall be operator-configurable by means of a proprietary management interface, or the use of SNMP.

9.5 Configuration of proprietary parameters relating to adaptation of admitted rate at an MGC

The TargetMG_OverloadRate (MG_Overloads/second) shall be configurable by the operator via a proprietary management interface, or through the use of SNMP. The TargetMG_OverloadRate may take any value in the set 0 to 1 in 0.1 steps. The default value of the TargetMG_OverloadRate shall be 0.5 MG_Overloads per second.

Implementations shall provide a means for an operator to configure any other parameters relating to the adaptation of the admitted rate (calls/second) at an MGC, whose adjustment may be necessary to meet the requirements of this Recommendation. The vendor shall specify the range and granularity of each such parameter.

The parameters used by the (implementation-specific) admission-rate adaptation scheme shall be operator-configurable by means of a proprietary management interface, or through the use of SNMP.

9.6 Configuration of control termination at an MGC

The TerminationPendingPeriod (measured in seconds) shall be operator-configurable by means of a proprietary management interface, or the use of SNMP. The TerminationPendingPeriod may take any value in the set 0 seconds to 300 seconds in 1 second steps. The default value of the TerminationPendingPeriod shall be 120 seconds.

The vendor shall provide a means for an operator to configure any other parameters relating to control termination at an MGC, whose adjustment may be necessary to meet the requirements of

this Recommendation. Implementations shall specify the range and granularity of each such parameter.

The parameters used by the (vendor-specific) admission-rate adaptation scheme shall be operator-configurable by means of a proprietary management interface, or through the use of SNMP.

9.7 MGC statistics

When an MGC initiates overload control (see 8.2.1) towards an MG (or virtual MG) output, the following information shall be recorded by the MGC for subsequent retrieval via a proprietary interface or the use of SNMP: date, time, MGC identity, MG identity.

When an MGC terminates overload control (see 8.2.4) towards an MG (or virtual MG) output, the following information shall be recorded by the MGC for subsequent retrieval via a proprietary interface or the use of SNMP: date, time, MGC identity, MG identity, total calls offered to, and rejected by, the restrictor over the whole duration of the overload.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2003