(12) STANDARD PATENT (11) Application No. **AU 2014252266 B2**
(19) AUSTRALIAN PATENT OFFICE

(54) Title
Voip bandwidth management

(51) International Patent Classification(s)
H04L 12/26 (2006.01) ***H04L 12/801*** (2013.01)

(21) Application No: **2014252266** (22) Date of Filing: **2014.03.16**

(87) WIPO No: **WO14/167431**

(30) Priority Data

(31) Number	(32) Date	(33) Country
13/859,765	**2013.04.10**	**US**

(43) Publication Date: **2014.10.16**
(44) Accepted Journal Date: **2017.09.21**

(71) Applicant(s)
Viber Media S.a.r.l.

(72) Inventor(s)
Marueli, Sunny;Shalgi, Ran

(74) Agent / Attorney
James & Wells Intellectual Property, GPO Box 1301, CANBERRA, ACT, 2601, AU

(56) Related Art
US 2011/0205889 A1

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 October 2014 (16.10.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/167431 A1

(51) International Patent Classification:
H04L 12/26 (2006.01) ***H04L 12/801*** (2013.01)

(21) International Application Number:
PCT/IB2014/059867

(22) International Filing Date:
16 March 2014 (16.03.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/859,765 10 April 2013 (10.04.2013) US

(71) Applicant: VIBER MEDIA INC. [PA/PA]; 53rd E street, MMG Tower, 16th Floor, Urbanizacion Marbella (PA).

(72) Inventors; and
(71) Applicants : MARUELI, Sunny [IL/IL]; 6 Sadot Street, 7402453 Nes Ziona (IL). **SHALGI, Ran** [IL/IL]; 21 Nizanim Street, 4528737 Hod Hasharon (IL).

(74) Agent: FOGEL, Ronny; 8 Gordon Street, 5323512 Givatayim (IL).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: VOIP BANDWIDTH MANAGEMENT

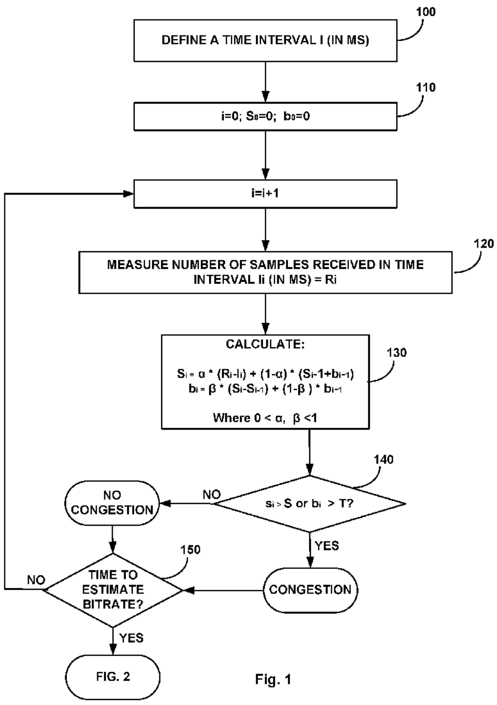
100
DEFINE A TIME INTERVAL I (IN MS)
110
i=0; S0=0; b0=0
i=i+1
120
MEASURE NUMBER OF SAMPLES RECEIVED IN TIME INTERVAL Ii (IN MS) = Ri
CALCULATE:
Si = α * (Ri-Ii) + (1-α) * (Si-1+bi-1)
bi = β * (Si-Si-1) + (1-β) * bi-1
Where 0 < α, β <1
130
140
si > S or bi > T?
NO
NO CONGESTION
YES
CONGESTION
150
TIME TO ESTIMATE BITRATE?
NO
YES
FIG. 2

Fig. 1

(57) Abstract: A computerized method of optimizing audio quality in a voice stream between a sender and a receiver VoIP applications, comprising: defining by the receiver time intervals; determining by the receiver at the end of each time interval whether congestion exists, by calculating (i) one-way-delay and (ii) trend, using double-exponential smoothing; estimating by the receiver a bandwidth available to the sender based on said calculation; sending said estimated bandwidth by the receiver to the sender; and using by the sender said bandwidth estimate as maximum allowed send rate.

WO 2014/167431 A1

Published:

— *with international search report (Art. 21(3))*

VOIP BANDWIDTH MANAGEMENT

TECHNOLOGY FIELD

[0001] The present invention relates to voice-over-Internet-Protocol (VoIP) systems, and more particularly to optimizing audio quality using adjustments of bandwidth usage.

BACKGROUND

[0002] Unlike “circuit-switched” voice, VoIP needs to work well over changing network conditions – because of competing traffic (e.g. viewing a clip on YouTube), wireless interference etc.

[0003] Some audio CODECs – e.g. Opus – support transmitting at different bitrates. In addition to using a multi-rate CODEC, one can also change the bitrate by changing the frame size and switching between CODECs.

[0004] Even with a CODEC that supports multiple bitrates (which may include changes to frame-size as above), we are still left with the problem of using the “best” bitrate given the network conditions – which requires measuring network conditions.

SUMMARY

[0005] The present invention provides a computerized method of optimizing audio quality in a voice stream between a sender and a receiver VoIP applications, comprising: defining by the receiver time intervals; determining by the receiver at the end of each time interval whether congestion exists, by calculating (i) one-way-delay and (ii) trend, using double-exponential smoothing; estimating by the receiver a bandwidth available to the sender based on said calculation; sending said estimated bandwidth by the receiver to the sender; and using by the sender said bandwidth estimate as maximum allowed send rate.

[0006] Determining whether congestion exists may comprise determining that congestion exists if the calculated one-way-delay is greater than a predefined positive constant or the calculated trend is greater than a predefined positive constant.

[0007] The method may further comprise determining congestion level based on said calculated trend value.

[0008] The method may additionally comprise determining whether bandwidth estimation should be done; and the steps of estimating, sending and using may be performed only if it is determined that bandwidth estimation should be done.

[0009] Determining whether bandwidth estimation should be done may comprise determining whether a predefined time length has elapsed since a last bandwidth estimate.

[0010] The predefined time length may be a round trip time.

[0011] Determining whether bandwidth estimation should be done may comprise determining whether congestion state has changed.

[0012] Estimating a bandwidth may comprises: a. estimating incoming bitrate; b. if there is no congestion setting the bandwidth estimate as greater than the previous estimated bandwidth; and c. if there is congestion setting the bandwidth estimate as smaller than the estimated incoming bitrate.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] For better understanding of the invention and to show how the same may be carried into effect, reference will now be made, purely by way of example, to the accompanying drawings.

[0014] With specific reference now to the drawings in detail, it is stressed that the particulars shown are by way of example and for purposes of illustrative discussion of the preferred embodiments of the present invention only, and are presented in the cause of providing what is believed to be the most useful and readily understood description of the principles and conceptual aspects of the invention. In this regard, no attempt is made to show structural details of the invention in more detail than is necessary for a fundamental understanding of the invention, the description taken with the drawings making apparent to those skilled in the art how the several forms of the invention may be embodied in practice.

In the accompanying drawings:

[0015] Fig. 1 is a flowchart outlining the congestion detection algorithm according to embodiments of the present invention; and

[0016] Fig. 2 is a flowchart outlining the bandwidth estimation algorithm according to embodiments of the present invention.

DETAILED DESCRIPTION OF EMBODIMENTS

[0017] To achieve the best audio quality in VoIP systems, we want to use the highest bitrate (for a given CODEC, using more bits/sec to encode the data should result in a more accurate representation of the input) while keeping the latency as low as possible.

[0018] We define latency as the delay, i.e. time it takes for the audio to travel between side 1's microphone and side 2's speakers. This contains two general components – the algorithmic delay (which for our purpose is all the time the audio spends in one VoIP application), which we will assume is fixed and the network delay. We will call the network delay from 1 to 2 the "one-way delay" (if we add the one way delay in the reverse direction, we get the "round trip delay"). In general, in a streaming application (e.g. viewing a clip on YouTube), it is OK for the delay to be of several seconds. However, in an interactive session (that is – a conversation), the perceived quality is strongly affected by the delay.

[0019] There are multiple components affecting the one-way-delay. The one that is handled by congestion avoidance protocols is the queuing delay. When packets arrive at a router faster than it can transmit them to the next hop, the packets are queued. The one-way-delay for the queued packet increases.

[0020] For example, say we have a “source” that sends 2 packets a second to a router that is connected to another link with a capacity of 1 packet per second. Assuming there is no queue initially, the first packet is released almost immediately by the router. However, the second packet arrives at 0.5 sec and has to wait until 1 sec to be transmitted. The next arrives at 1 sec and has to wait until 2 sec. The one-way delay due to queuing is 0 sec for the first packet, 0.5 sec for the 2nd, 1 sec for the third.

[0021] We use this increase in one-way-delay as a signal for congestion.

[0022] We assume each audio packet includes a timestamp (e.g. RTP packets) – this value usually increases by the number of samples in the previous packet. That is:

$$\text{timestamp}_i = \text{timestamp}_{i-1} + \text{samples}_{i-1}$$

Since the number of samples/sec is fixed (e.g. 8000 or 16000), this can be easily translated to seconds. For example, 480 samples at 16000 samples/sec = 30ms.

[0023] We assume the sender sends each packet “on time” – that is, in the previous example, the sender sends a packet every 30ms. In the ideal case, the packets will arrive evenly spaced (that is, every 30ms) to the receiver. However, if there is congestion we expect the reception time of packets to be larger (in the 2 packets/sec example above, the packets are sent at 0, 0.5, 1, … but are received at 0, 1, 2, … - that is, there is 0.5 sec between two packets sent, but 1 sec between 2 packets received).

[0024] In real IP networks, such as the Internet, this is not as simple as that: a random jitter is added to each packet, causing the delays to be longer or shorter than the “expected” times.

[0025] Let us call the time packet i is sent – s_i, and the time it is received r_i. We define interpacket delay as:

$$d_i = r_i - r_{i-1} - (s_i - s_{i-1})$$

If there is no congestion, we expect d_i to be 0 on average:

$E(d_i) = 0$

Congestion detection

[0026] The congestion detection algorithm according to embodiments of the present invention will now be described in conjunction with Fig. 1. To detect congestion on the receiver, the receiver's VoIP application measures the number of samples received in a predefined (step 100) fixed time interval – e.g. 120ms. If 120ms worth of packets are received (on average) in every 120ms interval, then there is no congestion. However, if in a 120ms interval less than 120ms worth of packets are received (on average), there is congestion.

[0027] Given enough samples, it is easy to detect congestion. However, we want to detect congestion quickly – while avoiding false-positives caused by jitter.

[0028] We prefer to sample in fixed length C intervals, as this simplifies the algorithm. So, let us call the 1st interval (in the example, from 0 to 120ms) I_1, the 2nd interval I_2, etc. and let us call the samples received in the i-th interval R_i (step 120), in the same units as the interval – e.g. ms. For example, using RTP, we may convert the RTP timestamps (indicating number of samples) into time units, e.g. milliseconds, since the samples rate is usually fixed, e.g. 8000 samples/sec for a "narrow band" voice call and 90,000 samples/sec for video. We have: I_i=C, for all i (in the example above, C = 120ms).

[0029] To measure congestion, we use double-exponential smoothing (step 130):

$s_i = \alpha*(R_i-I_i) + (1-\alpha)*(s_i-1+b_{i-1})$

$b_i = \beta*(s_i- s_{i-1}) + (1-\beta)*b_{i-1}$

where, s_0 and b_0 are set to some initial value (e.g. 0) (step 110) and $0 < \alpha, \beta < 1$ are constants.

s_i is the smoothed estimate of the one-way-delay (up to some constant) and should be 0 if there is no congestion. b_i is the “trend” – a positive b_i indicates an increasing delay, i.e.– congested state.

[0030] We now define congestion at the end of interval I_i as:
s_i > threshold S, for some threshold S>0 value or
b_i > threshold T, for some threshold T>0 value (step 140).
We can further define several S and/or T threshold values that will indicate the degree of congestion – e.g. none, light, normal or heavy congestion.

[0031] In step 150, if the last time congestion and bitrate were calculated indicate no need to re-estimate the available bandwidth at this time, the process goes back to step 120 to measure the number of samples received in the next time interval.

Bandwidth Estimation

[0032] The bandwidth estimation algorithm will now be described in conjunction with Fig. 2. Based on the congestion estimates, the receiver's VoIP application attempts to estimate the available bandwidth for the sender and whether the sender's VoIP application should increase or decrease its sending rate.

[0033] At time t, we estimate the incoming bitrate (for example by measuring the number of bits received in the last 1 sec) – r_t. If the network is congested, we assume the packets are transmitted as fast as possible and therefore r_t can be used as an estimate of the available bandwidth. On the other hand, if there is no congestion, the incoming rate is less than the available bitrate.

[0034] The receiver's VoIP application estimates the bandwidth available to the sender periodically, based on the most recent congestion estimate and the incoming bitrate. Let us say the bandwidth is estimated at times t_i, and the result is At_i. The initial bandwidth At_0 can be estimated, for example, from the incoming bitrate during an initial predefined time. Alternatively the initial bitrate may be fixed by a standard or negotiated as part of the initial handshake, or determined in any other method known in the art.

[0035] Let us call the incoming bitrate estimate at time t_i - rt_i (step 200).

[0036] If there is no congestion, we want to increase the available bandwidth estimate (step 220). For example:

if $r_{ti} > 2*A_{ti-1}$ we can set $A_{ti} = 2/3 * r_{ti}$.

Otherwise, Ati may be increased by multiplication with a constant factor: $A_{ti} = C * A_{t-1}$ (step 230), where C > 1 or by adding a constant: $A_{ti} = C + A_{t-1}$, where C > 1.

[0037] Another exemplary option for increasing the available bandwidth is to remember the last available bandwidth estimate A_{ti-1} and attempt to return to it quickly after a period of congestion (e.g. set the new estimate to be at least ½ of the last available estimate).

[0038] It should be noted that in some cases the bitrate may not be increased. For example:

- A maximal bitrate may be defined;
- The incoming bitrate from peer is (significantly) lower than the current estimate.

[0039] Alternatively, if there is congestion and assuming that the current incoming bitrate is our best bandwidth estimate, we want to reduce the bitrate (step 230) in order to resolve the congestion.

[0040] If there is light congestion, we may estimate At_i as follows:

$A_{ti} = \min(A_{ti-1}, r_{ti})$ (step 240)

[0041] At higher congestion levels, we may multiply the incoming bitrate by a constant (D<1) to allow for the delay to decrease (step 250). If the sender sends at full-speed, the delay will not decrease. In contrast, if the sender uses e.g. 80% of the available bandwidth, it means it is “catching up” – the queues are draining. Other methods of decreasing the estimated bandwidth may be used.

[0042] The receiver's VoIP application sends the estimates to the sender's VoIP application which will use them as the maximal send rate allowed.

[0043] The last thing we need to specify is how to determine the times to update the estimate.

Several exemplary options are:

1. Periodic update. This can be done every C (e.g. 120ms) or more. For example, we may re-estimate the bandwidth every 1 sec. Alternatively RTCP may be used for sending the periodic updates.

2. When the congestion state shows a change – e.g. there is a change from no congestion to congestion.

3. Since it will take a round trip (receiver sends estimate to sender, then has to wait for first data with changed bitrate to arrive), every time the receiver sends an estimate, it sets up a timer for the next estimate to take place, of e.g. RTT + ε or (1+ ε)*RTT, where RTT is an estimate of the round-trip time and ε is some constant.

[0044] RTT may be measured in a number of methods such as, for example:

1. The sender's VoIP application may send an “ack” packet for the change (we assume every packet following that packet is affected by the change).

2. Each media packet may contain an encoding of the current send rate (for example, a CODEC may support 256 different send rates – the first byte of the encoded stream may be the “mode” used). The receiver can then identify when the send rate has changed.

3. Explicit RTT packets may be sent.

4. RTT may be computed from RTCP.

[0045] As an enhancement, if congestion changes significantly – an immediate bitrate estimate may be triggered.

[0046] The present invention may be implemented in various combinations of software, hardware, or firmware.

CLAIMS

1. A computerized method of optimizing audio quality in a voice stream of Voice over Internet Protocol (VoIP) applications between a sender and a receiver, comprising:

defining, by said receiver, a plurality of time intervals for said voice stream;

determining, by said receiver, at an end of each of said plurality of said defined time interval whether congestion exists, by:

(i) calculating a one-way-delay for each of a first plurality of received audio packets in said voice stream; and

(ii) using double-exponential smoothing to calculate a one-way-delay trend of said first plurality of received audio packets;

estimating, by said receiver, a bandwidth available to said sender based on a result of said one-way-delay trend calculation;

sending, by said receiver, said estimated bandwidth to said sender; and

using, by said sender, said estimated bandwidth received by said sender as maximum allowed send rate for a second plurality of audio packets said voice stream of said VoIP applications.

2. The method of claim 1, wherein said determining whether congestion exists comprises determining that congestion exists if said calculated one-way-delay is greater than a first predefined positive constant or said calculated one-way-delay trend is greater than a second predefined positive constant.

3. The method of claim 2, further comprising determining congestion level based on a value of said calculated one-way-delay trend.

4. The method of claim 1, additionally comprising a step of determining whether a bandwidth estimation should be done; and wherein said estimating, sending and using are performed only if it is determined that said bandwidth estimation should be done.

5. The method of claim 4, wherein said determining whether said bandwidth estimation should be done comprises determining whether a predefined time length has elapsed since a previous bandwidth estimation was done.

6. The method of claim 5, wherein said predefined time length is a round trip time.

7. The method of claim 4, wherein said determining whether said bandwidth estimation should be done comprises determining whether congestion state has changed.

8. The method of claim 1, wherein said bandwidth estimation comprises:

a. estimating an incoming bitrate of said voice stream;

b. if it is determined there is no congestion, setting said estimation of said bandwidth at a value greater than said estimated incoming bitrate; and

c. if it is determined there is congestion, setting said estimation of said bandwidth at a value smaller than said estimated incoming bitrate.

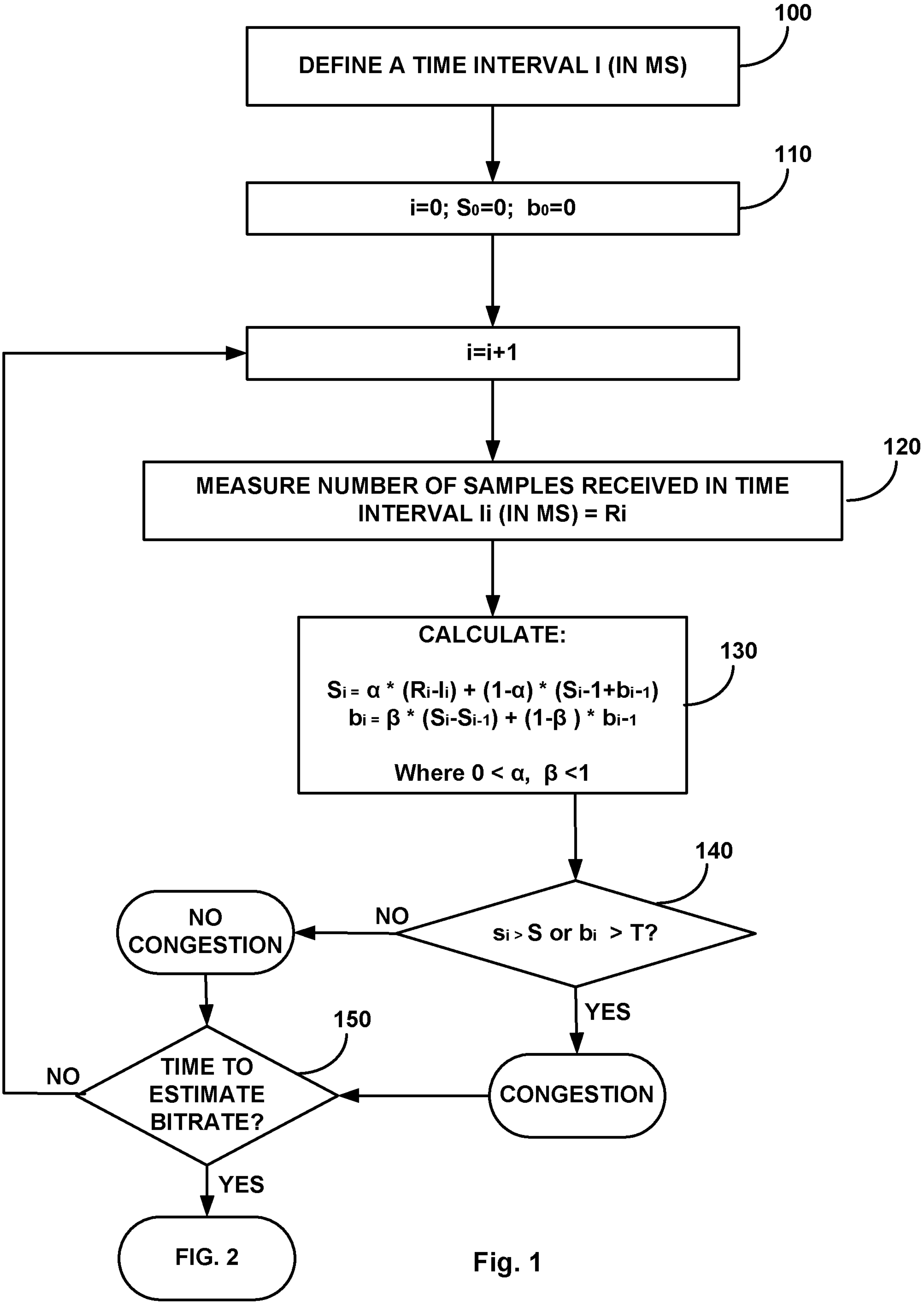
100
DEFINE A TIME INTERVAL I (IN MS)
110
i=0; S_0=0; b_0=0
i=i+1
120
MEASURE NUMBER OF SAMPLES RECEIVED IN TIME INTERVAL Ii (IN MS) = Ri
CALCULATE:
130
$S_i = \alpha * (R_i-I_i) + (1-\alpha) * (S_i-1+b_{i-1})$
$b_i = \beta * (S_i-S_{i-1}) + (1-\beta) * b_{i-1}$
Where 0 < α, β <1
140
$s_i > S$ or b_i > T?
NO
NO CONGESTION
YES
CONGESTION
150
TIME TO ESTIMATE BITRATE?
NO
YES
FIG. 2

Fig. 1

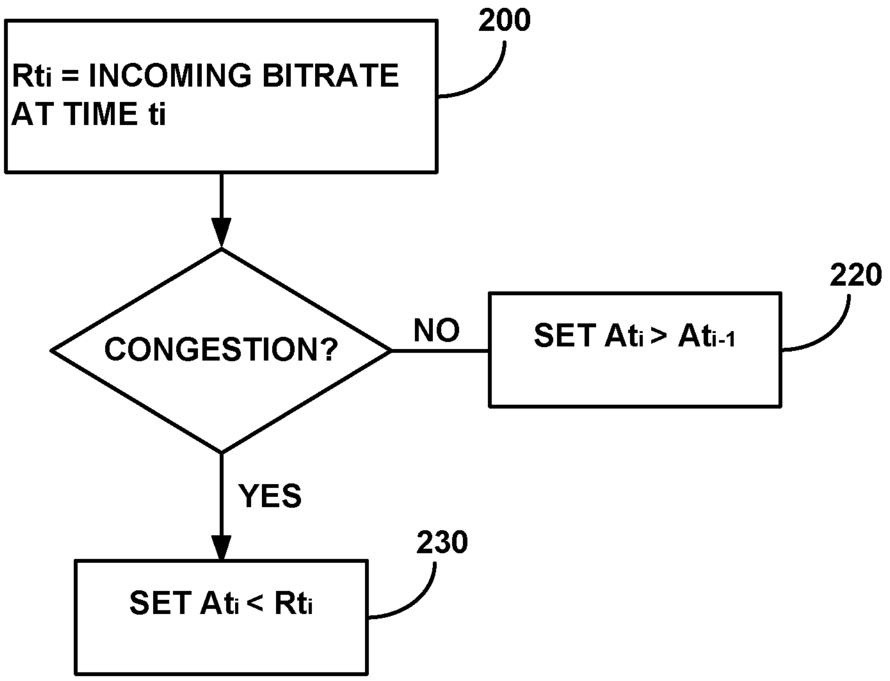
200
Rti = INCOMING BITRATE AT TIME ti
CONGESTION?
NO
SET Ati > Ati-1
220
YES
SET Ati < Rti
230

Fig. 2