Policy control framework-based algorithm for uncertain QoS harmonization in IP-based live video surveillance communications

Abhishek Mishra¹

Received: 20 October 2015 / Accepted: 11 May 2016 / Published online: 4 June 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract There are around 15 million surveillance camera installed in US and about 10 % of it is IP based. This number of IP-based surveillance is constantly growing as it gives an opportunity for smart real time surveillance. When the number of camera increases, it create a challenge to smart surveillance both in terms of cost and efficiency. Cost—as to monitor so many views, there is a need to have enough people. Efficiency—due to overload of monitoring, there are chances that important tracking is missed. Intelligent surveillance helps in avoiding many such issues, however, there is a constant threat that due to lack of appropriate policy control framework, tracking may be affected due to inherent network issues or human glitches. The paper presents a novel way, where a policy control framework is deployed for IP-based surveillance which will help in efficient monitoring. The paper studies the QoS pattern where such a system is deployed and shows that the utility function used for such tracking follows a linear uncertainty distribution pattern. The paper also concentrates on illustrating the algorithm for finding the cameras where the QoS needs to be harmonized. It shows how to make the algorithm opportunistic, so that the number of messages in the network can be reduced while tweaking the QoS.

Keywords Uncertain sources · PCRF · Uncertain prioritization · 3GPP · Video surveillance · IP · Mobile · Uncertain distribution

Introduction

IP-based video surveillance [1] is going to witness a large growth in the coming days. Also, video download as a service and video surveillance is expected to reach \$42 billion by 2019 [2]. City of London in 2013, quoted that there were more than 91,000 surveillance cameras [3]. This figure is increasing for efficient monitoring.

So far, trained security personnel who are capable of monitoring a modest number of incoming video streams, do the video review. As the volume of video streams grow, the monitoring abilities of security personal further fatigues. Adding more security personnel is expensive. Also, there is a sheer disadvantage that accompany human analyzing of video data.

The important aspect is the emergence of intelligent surveillance [4]. This technique enables the tracking to a new dimension, where the video is not a mere stream, but, it can at the same time identify the subject.

Background

Intelligent surveillance

In the past, there had been lot of successful studies for automated tracking. A subject can be tracked by using an Eigen tracker [5,6]. For example in Fig. 1, a subject (in oval) is selected which can be effectively tracked.

Figure 1a–e shows, the subject (in oval) which is constantly moving is being tracked by the system, despite the motion [6]. It has been proved that in spite of significant occlusions, changing background and multiple interacting, the subject can be tracked successfully [5].

In order to extend the area, multiple cameras are needed. A common subject can be tracked through multiple cam-

✉ Abhishek Mishra
siteofabhishek@gmail.com;
Abhishek.mishra2@redknee.com

¹ Redknee Technologies, Mfar-Greenheart Bld, 4th Floor,
Manyata Embassy Business Park, ORR, Nagawara,
Bangalore 560045, India

Fig. 1 Subject (in oval) is selected and tracked

Fig. 2 Subject tracking technique

eras, in dynamic scenes and can be analyzed without any issues [7]. Techniques like histograms and spatial relationships [7] have been highly successful in doing such analysis.

In a realistic scenario, when multiple cameras are installed, there is very little or no overlap between the view of the two cameras. The technique used for the fast tracking [8] uses a template library which has the appearance of the subject during the initial stage as well as the current ones. This template is having different variation of pose and brightness changes.

Figure 2 contains the technique which is used while tracking of a subject. Here, the selected frame is send for a match with the template.

The technique used for tracking subject in [9] is referred in this paper. It works as follows: consider a case, where function $F(I_x, I_y, t)$ is an image region in which I_x and I_y

represents the coordinates of I as a vector on x -axis and y -axis at a given time t . Let T be the template library containing occurrences of I as shown in the Eq. (1). For a given instance ∂ , the matching of the subject [9] at a location L is given by optimization equation (2) mentioned below:

$$T = \{T_{i,j}\}_{i=1,2,\dots,n}\{j=1,2,\dots,k\}. \quad (1)$$

$$\text{Sub}_L = \max_{i=1}^n \sum_{j=1}^k \{F(I_{x+\partial x}, I_{y+\partial y}, t + \partial t), T_{i,j}\}. \quad (2)$$

The optimization equation of (2) require to model the “shift” at instance ∂ [9]. This technique converges faster, if the approximation is crudely programmed in “shift”, for example, if there is a larger step size value, the technique will converge faster. In such a case, the solution may lack accuracy. The results will be of higher accuracy, if smaller value is selected, however, it will be obtained at the expense of an increase in computational complexity.

When an object is under observation, the template library of this object is refreshed during the handoffs among the cameras.

Assuming that the object selected is of 120×80 pixel and the template library consists of 15 such templates at a time ∂ . If this is 8 bit grayscale image, it amounts to $120 \times 80 \times 15 \times 8 = 1152$ Kbits bandwidth. Therefore, the bandwidth differs based on the variation of the template at a given instance ∂ and object selected.

Modern broadband infrastructure

Different standardization bodies have proposed different ways to stream the traffic. 3GPP [12] has proposed a convergent way which enables different access to route the data through a common core. This architecture enables a user to move from one access (e.g. WiFi) to another (e.g. 4G) without any break of existing session. The core is efficient enough to continue the service.

Another feature of this architecture is that it gives an operator the possibility to set the policy and QoS of the session based on different conditions and attributes using the PCRF. Figure 3 contains one such convergent architecture proposed by 3GPP.

Applicability of uncertainty theory

The case of live subject tracking is an ideal case where uncertainty theory [10] can be applied. This is due to the case, that the technique for tracking mentioned in Eq. (2), depends largely on the template size, which varies dynamically. Based on the scene changes or background occultation, the template size can increase to a different degree. Similarly, the template size can decrease too, if the system optimizes the unrelated coordinates from the template.

“Uncertainty distribution” [11] can be treated as a component based on this theory, which works as a basic tool to specify uncertain variables. In cases where QoS of a given setup is not certain, it becomes necessary to model such an environment, which can give the idea to the network planner on how to effectively write policies. “Calculations and results” of this paper gives an example of the use of the uncertainty distribution.

Fig. 3 Simultaneous IP-CAN session connection using multiple accesses

Identification of nearest neighbor

The nearest neighbors (NN) problem is researched deeply in many fields of vision [15] and geographical information systems (GIS) [16]. There is an availability of vast literature in this area. Normally, the nearest neighbor identification scheme can be divided into two types: the k -nearest neighbor searching scheme, where the idea is to find the k -nearest neighbor for a given point p . The other is the ε range neighbor scheme, where all the nearest neighbor within the distance ε of the point p are identified.

“Calculations and results” of this paper illustrates that the scheme of ε range neighbor fits for the use case requirements mentioned in paper. The section also introduces algorithm for finding such neighbor in an opportunistic way, so that it leads to reduction of signaling messages in the network.

Experiment description

Requirements identified

Following are the use cases which are identified:

- A. A subject to be tracked is moving fast. An operator monitoring from one of the cameras can select a subject that can be further tracked using the above mentioned intelligent schemes, through other cameras via which it is being viewed, without facing the network glitches and QoS. This would enable real-time tracking of the subject. The uncertainty in QoS, cannot be solved by merely provisioning a statically high QoS at all the camera nodes and it will be an un-necessary waste, as at many times the subject tracking may not be required at that camera.

each with /128 address, which are viewed as an application to the surveillance center. Therefore, this scheme helps in reducing the address overhead to the operator, at the same time can increase the number of cameras in any given zone. Figure 4 also illustrates that the subject selected on a camera in zone 1 is a fast moving car carrying unauthorized people. Important thing to note is that out of three camera zones, only two camera zones are able to identify the subject which is selected. Finally, the car is tracked by a camera in zone 3.

Figure 4 shows the 3GPP-defined architecture for communicating with the application server. At this point, it is important to note that there is a difference between a typical IP sessions and IP-based surveillance. In the IP-based surveillance the subject is always selected by the application server.

Calculations and results

Call procedures

Operators has to do policy control in their network to do make the flows streamline. An efficient policy and QoS control is done using a PCRF which is connected with the gateway with the standard Gx [13] interface and with application server with Rx [14] interface. This is shown in the Fig. 4.

Figure 5 illustrates the call flow for setting up the priority and QoS for the subject selected. The steps in Fig. 5, which are in “bold yellow” are the new steps, introduced as part of the experiment. The following steps are executed in order to do this operation:

- A.1. IP tunnel is created between the router(s) and the gateway. The prefix length of /64 is assigned.
- A.2. The gateway sends the CCR-I for each of the /64 prefix assignment towards the PCRF.
- A.3. PCRF evaluates the policy and provisions the default QoS for the session.
- A.4. The IP tunnel is modified with the installed QoS.
- A.5. The cameras start sending the data which they are viewing. There are N cameras in each zone and have /128 derived address from the /64 prefix assigned to its router.
- A.6. Subject-1 is selected on Cam-1 in zone 1.
- A.7. Application server sends the App-id as Subject-1 and Cam-1's /128 address.
- A.8. PCRF sends AAA-I as the binding is successful.
- A.9. PCRF binds the session with the information provided. It sends RAR message towards the gateway to prioritize the session from Cam-1 and install a new QoS.
- A.10. Gateway does the installation of the rules.
- A.11. IP tunnel is modified and flows from Cam-1 are given the priority and QoS supplied by the PCRF.

The subject now moves away from zone 1. Zone 2 cameras are not able to identify the subject. The subject is further observed in zone 3 in Cam-5. Now, it is important to reset the priority and QoS of Cam-1 in zone 1 and prioritize the viewing of Cam-5 in zone 3.

Figure 6 has an illustration for this scenarios. The steps in Fig. 6, which are in bold yellow are the one introduced as part of this experiment:

- B.1. Subject-1 moves to zone 3 from zone 1.
- B.2. Application server sends the AAR-I with prioritize the flows for Cam-5 in zone 3 and the Subject-id.
- B.3. PCRF sends AAA-I as the binding is successful.
- B.4. Based on the Subject-id, the PCRF sends the RAR to reset the QoS and priority for Cam-1 in zone 1.
- B.5. Gateway does the installation of the rules.
- B.6. IP tunnel is modified and flows from Cam-1 are set to default QoS and priority.
- B.7. PCRF sends RAR towards the gateway to prioritize the session from Cam-5 and install a new QoS.
- B.8. Gateway does the installation of the rules.
- B.9. IP tunnel is modified and flows from Cam-5 in zone 3 are given the priority and QoS supplied by the PCRF.

Distribution of QoS uncertainty

The Eigen tracking scheme, discussed in “Intelligent surveillance” for processing the fast moving subject is an application, where the QoS demands varies based on the template library profile, which is dynamically changing. This creates an uncertainty in the QoS needs. Furthermore, there will be a gap between the camera perceived QoS and the network/service provided QoS at two different timestamps.

Due to dynamic change in QoS demands, there is a need for a mechanism which will assess the camera-perceived QoS at different timestamps. In order to do so, a utility function is to be defined, which is not be merely based on network characteristics. This utility function provides a measure for the bearer performance.

Definition wise, utility functions are mathematical modeling techniques that is typically used to model the relative preferences of UE at different timestamps. They also map the order of various outcome of the events by assigning a simple scalars to each outcome.

While dealing the uncertainty, one needs to note that the QoS demands of the camera which were once in priority, but since now the subject has moved out of its coverage needs to be reset also, as shown in Fig. 6. Keeping these things in mind utility function needs to be defined.

A resource metric, $r_i(k)$ is defined, which represents the QoS demands of camera i for k instance of time. We define a loss factor f_i^k as under:

Fig. 5 Call flow for QoS and priority setting when subject is in zone 1

$$f_i^k = \max \left(1 - \frac{r_i^{\text{Req}}(k)}{r_i^{\text{Curr}}(k)}, 0 \right), \quad (3)$$

where, $r_i^{\text{Req}}(k)$ and $r_i^{\text{Curr}}(k)$ represent the required and current QoS respectively.

For a difference, $\mu_i^k = \frac{r_i^{\text{Req}}(k) - r_i^{\text{Curr}}(k)}{r_i^{\text{Curr}}(k)}$, the utility function U_i^k is defined as under:

$$U_i^k = \begin{cases} \max \left(1 - \gamma_{k,i}^+ \mu_i^k, 0 \right), & r_i^{\text{Req}}(k) < r_i^{\text{Curr}}(k) \\ 1, & r_i^{\text{Req}}(k) = r_i^{\text{Curr}}(k) \\ \max \left(1 + \gamma_{k,i}^- \mu_i^k, 0 \right), & r_i^{\text{Req}}(k) > r_i^{\text{Curr}}(k) \end{cases} \quad (4)$$

Here $\gamma_{k,i}^\pm$ represents the degree of utility reduction. $\gamma_{k,i}^+$ means the reduction in utility function U_i^k . $\gamma_{k,i}^-$ is the same in the

reverse direction. $\gamma_{k,i}^+ > \gamma_{k,i}^-$ means that the utility is decreasing strongly when the template size is increasing.

As an example, the value of $\gamma_{k,i}^+$ is chosen as 1 and that of $\gamma_{k,i}^-$ is 0.5. Based on the varying template size discussed in “Intelligent surveillance section”, the utility function value is studied in Table 1.

Considering a case, where, $r_i^{\text{Req}} = r - g, r - g - 1, \dots, r, \dots, r + g$. In such a scenario, the equation (4) reduces to,

$$U_g = \begin{cases} 0, & g \geq 2 \\ 1 - 0.5g, & 0 < g < 2 \\ 1, & g = 0 \end{cases} \quad (5)$$

Fig. 6 Call flow for QoS and priority setting when subject moves to zone 3

Figure 7, shows the plot of U'_g which is $(1 - U_g)$. This shows, that it follows the pattern for linear uncertainty distribution [10].

Camera selection algorithm for QoS determination

The vehicle under observation is a fast moving subject and hence requires the QoS of the cameras through which it passes dynamically adjusted, so that the image tracking technique discussed in Intelligent surveillance section can be implemented with ease.

Figure 8a consists of two subjects (S1 and S2) which are to be tracked and cameras of those paths only needs to have an upgraded QoS. Figure 8b consists of the graph constructed based on the path followed. The camera zones are mentioned as “Z” in the figure.

Once, the vehicle leaves the camera site, it needs to be reset to the normal QoS.

As an example of the algorithm mentioned in this section is shown in Fig. 9. Figure 9 shows that subject S1 at time T_a is under the influence of camera zone Z4 and Z5. At the time T_b it is under the influence of camera zone Z5, Z12 and Z6. Between two consecutive times, T_a and T_b , the camera zone, i.e. Z5, needs to retain the same QoS. It is very important to mark the camera zones which are to retain the same QoS, as it would reduce the number of signaling messages which are exchanged between the policy control and the network for QoS changes.

Based on Fig. 8a, the adjacency matrix of different camera zones Z_i , based on the street map is shown in (6). The zones which are having the value one, shows the neighbor zones.

Table 1 Pattern of utility function

r_i^{Curr}	r_i^{Req}	U_i^{k+}
(a) $r_i^{\text{Curr}} < r_i^{\text{Req}}$		
560	1152	0
560	768	0.628571
560	691.2	0.765714
560	614.4	0.902857
560	570	0.982143
r_i^{Curr}	r_i^{Req}	U_i^{k-}
(b) $r_i^{\text{Curr}} > r_i^{\text{Req}}$		
1152	560	0.691667
768	560	0.864583
691.2	560	0.905093
614.4	560	0.946875
570	560	0.989474

Fig. 7 Linear uncertainty distribution pattern for the utility

	z1	z2	z3	z4	z5	z6	z7	z8	z9	z10	z11	z12	z13	z14
z1	0	1	1	0	0	0	0	0	0	1	0	0	0	0
z2	1	0	1	0	0	0	0	0	0	0	0	0	0	0
z3	0	1	0	1	0	0	0	0	0	0	0	0	0	0
z4	0	0	1	0	1	0	0	0	0	0	1	0	0	0
z5	0	0	0	1	0	1	0	0	0	0	1	1	0	0
z6	0	0	0	0	1	0	1	0	0	0	0	1	0	1
z7	0	0	0	0	0	1	0	1	0	0	0	0	1	0
z8	0	0	0	0	0	0	1	0	1	0	0	0	1	0
z9	0	0	0	0	0	0	1	1	0	1	0	0	1	0
z10	1	0	0	0	0	0	0	0	1	0	1	0	0	0
z11	0	0	0	1	1	0	0	0	0	1	0	0	0	0
z12	0	0	0	0	1	1	1	0	0	0	0	0	0	0
z13	0	0	0	0	0	0	1	1	0	0	0	0	0	1
z14	0	0	0	0	0	0	1	0	0	0	0	0	1	0

(6)

There are different algorithms proposed to find the nearby cameras using the nearest neighbor. However, while finding the nearest neighbor, the important thing here is to find the

cameras it that do not need QoS upgrade, as their QoS is already in the desired QCI and bandwidth.

In typical cases, the adjacency vector-based solution is used. In such cases, for each camera, there is a vector which has information about the every other camera as shown in the matrix (6). This is a very expensive data structure, as there is a space of $\Theta(N^2)$ needed. The searching of nearest camera and excluding the camera's which do not require QoS manipulation requires a mere vector XOR operation, and this requires $O(N)$ operation at most.

There is a need for a better algorithm than the above mentioned adjacency vector based one, which can take more realistic parameter (e.g. subject's varying speed), and requires lesser space.

The algorithm proposed is based on the following: we select a universal point omega (Ω), from where the distance, $d(\Omega, k)$ is computed for a query camera k . Let C be the set containing all the cameras. This set can be divided into three sets as under:

$$\begin{cases} C1 = \{k \in C : d(\Omega, k) \leq (R - \lambda)\} \\ C2 = \{k \in C : d(\Omega, k) \geq (R + \lambda)\} \\ C3 = \{k \in C : |d(\Omega, k) - R| \leq \lambda\} \end{cases} \quad (7)$$

Here, for a given R and λ , $C1$ contains the cameras which are in neighbor of k , $C2$ contains the camera which are not the neighbor of k and $C3$ contains the cameras which are again neighbor of k , but do not require the QoS upgrade as they already have the desired QCI and bandwidth. As an example in Fig. 9, at the time T_b , $C1$ contains $Z12$ and $Z6$, $C2$ contains $Z4$ and $C3$ contains $Z5$. Since $Z5$ was $Z4$'s neighbor at time T_a , hence, it do not require the QoS setting again.

Algorithm construct proposes a tree like structure, where $C3$ is added as a common map to $C1$ and $C2$. This will ensure that a binary tree is constructed and while searching the tree, in a specific node read, the common cameras are identified along with the new cameras. This will help in setting the QoS of the cameras which are not in "common map". In the algorithm construct given below, the node consists of "common map" and "data map". The "data map" contains the camera, which require the QoS calibration. On the other hand, the "common map" contains the camera which are common to more than one nodes.

Furthermore, in each C_j , there is an index value R_{LO} and R_{HI} , which marks the lowest and highest bound of distance value [17]. These index values will be used to optimize the search of the neighbor cameras. For example, while querying for camera q for a given radius r , it is not important to look for a C_j , till, $[d(\Omega, q) \leq R_{LO} - r]$, or $[d(\Omega, q) \geq R_{HI} + r]$.

In field, it is important to identify the value of R and λ at the run time. So, at a given time, based on the speed of the subject, these values will be identified: R will be computed based on

(a)

(b)

Fig. 8 **a** City map path taken by different subjects; **b** camera zone graph constructed based on path followed

speed and time. For example, if the subject is moving at a speed on 60 km/h, then the radius R can be 32 m if the T_a is chosen for every 2 s basis. Similarly, λ is chosen as the difference of $|T_a - T_b|$. The time utilization pattern can be

computed based on the subject's observation or as decided by the operator. In any of the cases, the algorithm effectively reduces the number of camera's which requires the QoS to be tweaked with.

Algorithm Construct:**Construct(C, R)**

```

1. {
2.   if (C == NULL) exit;
3.   Select q from C[j--];
4.   {
5.     Node* lNode = new Node();
6.     lNode->p = Ω;
7.     lNode->R = median(d(Ω, i), ∀i ∈ C);
8.     if (|d(Ω, i) - R| ≤ λ)
9.     {
10.      lNode->commonMap ← q;
11.    }
12.    if ((d(Ω, q) < (R))
13.    {
14.      if (lNode->commonMap == NULL)
15.      {
16.        lNode->dataMap ← q;
17.      }
18.      lNode->left = Construct(C, R);
19.    }
20.    elseif (d(Ω, q) ≥ (R))
21.    {
22.      if (lNode->commonMap == NULL)
23.      {
24.        lNode->dataMap ← q;
25.      }
26.      lNode->right = Construct(C, R);
27.    }
28.  }
29. }

```

Analysis of algorithm construct

The algorithm construct works on the concept of obtaining median, which requires N time for N elements present. Furthermore, it requires a constant time η for computing the common cameras which are appended to the either side of the tree. Hence, in terms of recurrence, this can be mentioned as:

$$T(N) = 2T\left(\frac{N}{2}\right) + N + \eta. \quad (8)$$

Since, $T(1) = c$ (a constant) and taking $N = 2^k$, summation of k levels lead to

$$\begin{aligned}
T(N) &= kcN + cN + k\eta \\
T(N) &= Nc \log N + cN + \eta \log N \\
\sim T(N) &= \Theta(N \log N).
\end{aligned}$$

Algorithm Search:**Search(q, r, Node)**

```

1. {
2.   if (d(Ω, q) < r)
3.   {
4.     Selected_Nodes_Map = Append(Node)
5.   }
6.   elseif (d(Ω, q) + r > Node->R)
7.   {
8.     if (max(RLO - d(Ω, q), 0) ≤ r)
9.       Search(q, r, Node->right);
10.  }
11.  elseif (d(Ω, q) - r ≤ Node->R)
12.  {
13.    if (max(d(Ω, q) - RHI, 0) ≤ r)
14.      Search(q, r, Node->left);
15.  }
16. }

```

Analysis of algorithm search

The algorithm search works on the concept of the searching a binary tree, which in usual cases yields a performance of $\Theta(\log n + 1)$. The search algorithm proposed for the camera search also is under the same range.

Impacts of motion variation

Now, let us consider a case, where the speed of the subject is varying and this requires the nearest neighbor radius is also varying [18]. In such a case, there is a tolerance, which needs to be taken into account for the Eq. (7).

In such a case, the radius of C1, C2 and C3 will be impacted. Let there be a factor defined, $0 < \beta < 1$ by which the radius varies. In such a case, if the subject's speed is more in time T_b than T_a , then to find the q in the radius R will modify to $d(\Omega, q) - \beta R$. On the other hand if the subject's speed is more at time T_a than T_b then to find q in the radius R will modify to $(1 + 1 - \beta)R$. So, in this case, to find q , in radius R , will modify to $(2 - \beta)R - d(\Omega, q)$.

If $T_a[\text{speed}] < T_b[\text{speed}]$

$$\begin{cases}
C1 = \{k \in C : d(\Omega, k) \leq (\beta R - \lambda)\} \\
C2 = \{k \in C : d(\Omega, k) \geq (\beta R + \lambda)\} \\
C3 = \{k \in C : |d(\Omega, k) - \beta R| \leq \lambda\}
\end{cases} \quad (9)$$

If $T_a[\text{speed}] > T_b[\text{speed}]$

$$\begin{cases}
C1 = \{k \in C : d(\Omega, k) \leq ((2 - \beta)R - \lambda)\} \\
C2 = \{k \in C : d(\Omega, k) \geq ((2 - \beta)R + \lambda)\} \\
C3 = \{k \in C : |d(\Omega, k) - (2 - \beta)R| \leq \lambda\}
\end{cases} \quad (10)$$

Fig. 9 Camera zone selection for subject S1 at different time T

Fig. 10 Tolerance captured in the algorithm due to speed variation of the subject

Analysis of the experiment

In normal 3GPP communication world, the iterations are done based on subscriber identification (e.g. MSISDN) or IP addresses. However, while using this architecture for IP based video surveillance, one thing to note is that the subscriber identification or IP addresses may be insufficient to effectively do the policy and QoS control.

In Fig. 6, on receiving the AAR, to modify the policy and QoS characteristics for the Cam-5 in zone 3, the PCRF have to first reset the policy and QoS characteristics for the Cam-1 in zone 1. Thing to note is that the zones 1 and 3 do not share the same IP prefixes. In such a case, the PCRF have to use the Subject-id sent in the AAR to match, if any other cameras in any other zone have been prioritized.

Hence, PCRF has to maintain additional data structure to keep a track of subject.

Furthermore, in 3GPP defined network, the primary or default bearer which is set when the camera is powered-on, have a non-guaranteed bit rate (non-GBR) QoS class indicator (QCI) [12]. This means that there is a potential threat of data to be lost, as the maximum bit rate (MBR) value at a time can be zero also, since the GBR does not apply for such bearers. To avoid this data loss, on setting the priority, an additional dedicated or secondary bearer will be setup which has a GBR QCI. This helps in avoiding the cluttering of the media which is being sent.

Important thing here to note is, that GBR QCI consumes more power and network resources, so, it is advisable to modify the QCI as soon as the requirement is over. The call procedures defined in the “Calculations and results”, proposes this.

Furthermore, the use of the uncertainty distribution, works as a sufficient and necessary condition for the given utility

function about the kind of uncertainty distribution which it follows.

The algorithm proposed reduces the number of RAR messages sent in the network, as it identifies the cameras which already has the desired QoS.

On doing the comparison of adjacency vector based scheme with the tree based scheme, we obtain that the tree based scheme performs better than the adjacency vector based scheme. Also, the tree based scheme can take different real-time parameter like subject's motion into account while defining the nearest cameras. Figure 10 captures the plot of tolerance in algorithm due to speed variation at different time.

Benefits

Installing the bandwidth statically for a large number of cameras is a resource wastage for an operator in cases where the views are not a priority at a specific time. Hence, to adapt the QoS dynamically when there is a need is the key of the day. Also, scheduling the camera with the automated machine learning techniques helps the operator in pin-pointing the tasks required without involving much of the human errors. The paper presents a solution where policy and QoS issues can be eliminated while tracking of a moving subject. Setting of correct priority of the view makes it distinct from the others. It becomes very important to effectively view the subject when the network becomes congested. Also, streamlining the view in such scenarios becomes quite important as there are too many cameras to be observed. The algorithm proposed also reduces the number of RAR in the network by identifying the cameras which already have the desired QoS.

Conclusion

IP-based remote surveillance proposes a challenge to view a given subject in real time. These challenges become graver as there are inherent QoS issues with the IP network. PCRF-based surveillance promotes a state-of-the-art solution as it rightly fits into standard 3GPP promoted network. The solution rightly computes the uncertainty in the QoS at different point of times and overcomes it by using the algorithm proposed in the paper.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. SYS-CON Media (2013) Research and markets, global video surveillance systems and services market report 2013–2018. <http://www.sys-con.com/node/2874993>
2. Video Surveillance and VSaaS Market (2013) Global industry analysis, size, share, growth, trends and forecast, 2013–2019, Dublin. <http://www.heraldonline.com/2013/11/14/5405828/research-and-markets-video-surveillance.html>
3. Bloomberg.com (2013) Surveillance cameras sought by cities. <http://www.bloomberg.com/news/2013-04-29/surveillance-cameras-sought-by-cities-after-boston-bombs.html>
4. IMS Research (2013) IMS research video surveillance trends for 2013. http://www.securitymarketintelligence.com/press_releases/Video_Surveillance_Trends_for_2013
5. Black MJ, Jepson AD (1996) Eigen tracking: robust matching and tracking of articulated objects using a view-based representation. Technical report: RBCV-TR-96–50. Department of Computer Science, University of Toronto
6. Shetty A, Roy SD, Chaudhuri S (2008) Importance sampling-based probabilistic Eigen tracker. In: Proceedings of national conference on computer vision, pattern recognition, image processing and graphics, Gandhinagar
7. Hu W, Tan T, Wang L, Maybank S (2004) A survey on visual surveillance of object motion and behaviors. *IEEE Trans Syst Man Cybern* 34(3):334–352
8. Lucas BD, Kanade T (1981) An iterative image registration technique with an application to stereo vision. International joint conference on artificial intelligence, pp 674–679
9. Tsagkatakis G, Savakis A (2009) A random projections model for object tracking under variable pose and multi-camera views. International conference on distributed smart cameras (ICDSC)
10. Liu B (2014) Theory and practice of uncertain programming, 4th edn. Springer, Berlin
11. Peng ZX, Iwamura K (2010) A sufficient and necessary condition of uncertainty distribution. <http://orsc.edu.cn/online/090305.pdf>
12. TS 23.401v13.2.0 (2015) General packet radio service (GPRS) enhancements for evolved universal terrestrial radio access network (E-UTRAN) access, 3GPP
13. TS 29.212 V12.5.0 (2014) 3rd generation partnership project. Technical specification group core network and terminals. Policy and charging control over Gx reference point, 3GPP
14. TS 29.214 V12.5.0 (2014) 3rd generation partnership project. Technical specification group core network and terminals. Policy and charging control over Rx reference point, 3GPP
15. Shakhnarovich G, Darrell T, Indyk P (2006) Nearest-neighbor methods in learning and vision. MIT Press, Cambridge
16. Gao P, Smith TR (1989) Space efficient hierarchical structures: relatively addressed compact quadrees for GISs. *Image Vis Comput* 7(3):173–177
17. Tzi-cker C (1994) Content-based image indexing. In: Proceedings of the 20th international conference on very large data bases, pp 582–593, September 12–15
18. Chen JY, Bouman CA, Dalton JC (2000) Hierarchical browsing and search of large image databases. *IEEE Trans Image Process* 9(3):442–455