1- Methodology

Existing event matching algorithms are inadequate to meet the requirement of disseminating live content in the cloud-based IPTV surveillance platform, since their multi-hop routing techniques and coarse-grained partitioning techniques lead to a low matching throughput, and their upload capacities do not scale well.

1- To achieve the goal of efficient matching, to group the subscriptions into clusters, which are managed by a group of parallel servers in cloud, and events can be matched against their corresponding clusters using exact match approach. (multi domain)
2- The clustering method can be extended to consider the statistical distribution of subscriptions which can dynamically identify appropriate number of clusters in different subscription distributions at any time to reduce the matching/evaluation cost as well as to make the searching faster. It will adaptively adjust the scale of servers according to the changing performance of servers and rebalances the workloads of servers through migrating clusters. When migrating clusters, we will make sure to synchronize the status of servers to keep correct and continuous event matching.
3- A technique, to ensure elastic event matching service.
4- cloud hosting
5- The steps are as follows:

1. The Matching Process

Using the content-based method, the matching has definite steps. First, a new user subscribes to the service. Each subscription is described by a group of attributes. Each attribute has three fields which are; attribute name, value range and geographical position. Attribute name is one of the attributes used to describe a feature of the video that is to be published. Value range defines the lower and the upper bounds of the interested values. Geographical position is an ID number to refer to subscriber's geographical position *(Nguyen & Huh, 2011)*.

When an event notification is sent by a publisher, the message is holding a list of tuples. Each tuple has three fields which are attribute name, value, and geographical position. Attribute name is one of the attributes used to describe a feature of the video that is to be published. Value is the value of the attribute that is a description of a feature of the video to be published. Geographical position is an ID number to refer to publisher's geographical position *(Nguyen & Huh, 2011).*

A message matches a subscription by examining the value of an attribute of the video to be published. If the value is within the value range of the subscription of a subscriber, then there is a match for this attribute *(Nguyen & Huh, 2011).* A minimal number of matched attributes is to be defined to consider that the video matches a subscription.

2. An Example of Matching

In order to do the matching two steps are needed to happen. First, the subscription process must include efficient details about the interested events. An example of subscription description CD is:

Time: morning, afternoon, evening
Number of people involved: from 1 to 3
Place: kitchen, bedroom, outdoors
Action: falling, injured, screaming, running

Second, when any user is publishing a video, he/she has to fill in a form of description. Again the description is tuples of pairs. Each tuple contains a pair. Each pair contains a name and a value. An example of video description CD is:

Time: morning
Number of people involved: 1
Place: kitchen
Action: running

The matching will be done based on the matching values of the attributes. When the similarity is above a minimal function, determined during the implementation phase, the video will be sent accordingly.

3. The Subscription Process

To push the UDP or TCP packets over an IP network, the content based description is integrated. The critical forwarding function, that is responsible to find the matches, is to be presented only at some major routers, which are called the edge routers, and leave the other routers to perform the traditional forwarding. Moreover, the description of a video surveillance is maintained as CD (Content Descriptor) that stores the tuples of description of the video surveillance. In addition, there are various RP (Rendezvous Point) and every RP is responsible for storing a group of CDs (Chen, et al. 2012).

In order to transfer the video surveillance from the publisher to a receiver the following steps take place. First, the subscriber sends a subscription request including the description of the interested attributes of video surveillance, which is called the CD, to an edge router. **The edge router stores two tables a table of CDs and a table of ST (Subscription Table) and is responsible to make two steps. First, the edge router checks its table of CDs to know which RP is responsible for this CD and sends the CD to the RP. The second step is to add the new subscription associated with its CD in the ST table**. Finally, **the RP, according to the received message, adds the new subscription to its group of subscribers and sends a confirmation message to the edge router** (Chen, et al. 2012). This process is illustrated in the following figure.

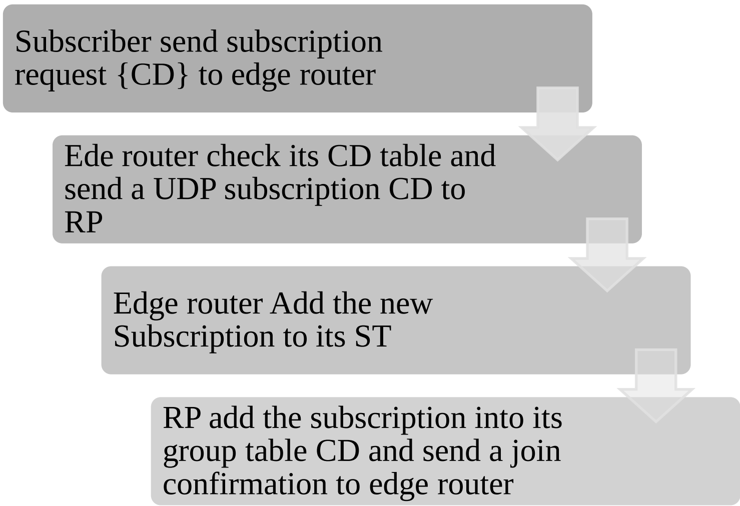

Figure 1 Subscription Process (Chen, et al. 2012).

4. The Publishing Process

When a new video surveillance is to be published the following steps happen. First, the publisher sends the video associated with its CD to the nearest edge router. This message is called publish notification. Then, the edge router checks its table of CDs to identify the responsible RP of this video based on its CD. The RP accordingly checks its table of CD that refers to the group of subscribers interested in this video, and sends a message to the interested edge routers. The edge routers according to their CD, sends the video to the interested subscribers *(Chen, et al. 2012).* This process is illustrated in the following figure.

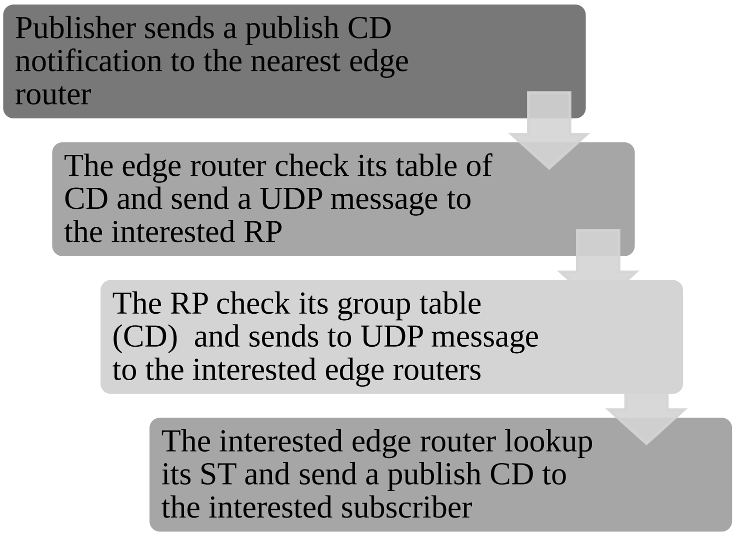

Figure 2 Publishing Process (Chen, et al. 2012).

5. The Multidimensional Domains Objective

To satisfy the fourth objective, the ad-hoc network is divided to multiple domains to avoid the scan of all RPs to find the responsible one. The IP multicasting is to be done in specific domains rather than the whole network. To enable exchanging the data among several domains, **a third type of routers is to be setup which is the boundary routers** *(Chen, et al. 2012)*. The following figure illustrates the idea.

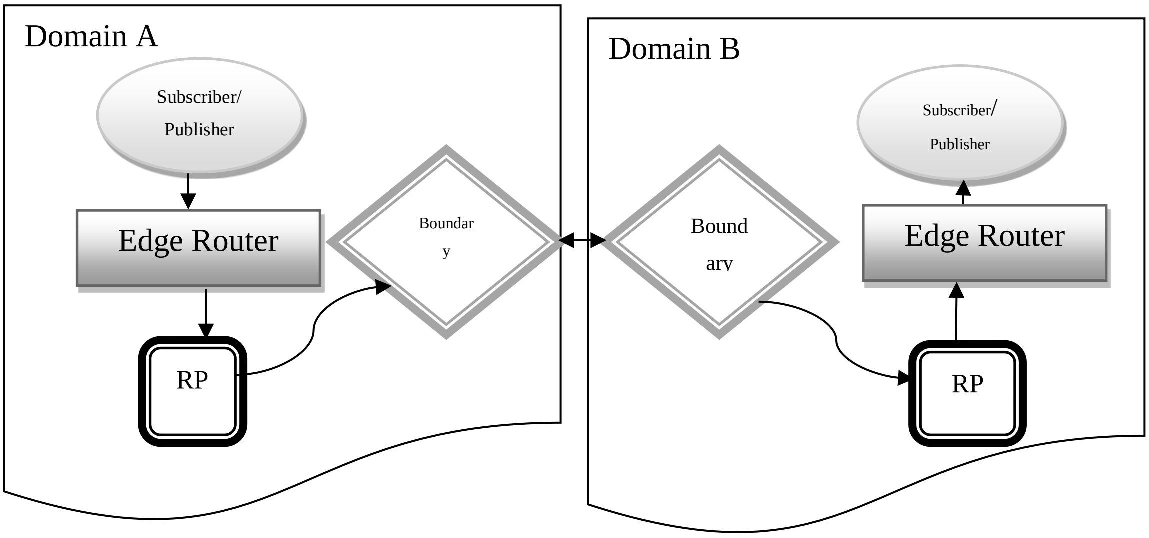

Figure 3 Multi Domain Processing (Chen, et al. 2012).

The Main Components of the Implementation are as follows::

1. Cloud hosting of the core application that provides the business layer of the application. It is the NDN Service Handler. It is the server side.
2. Mobile Application that is going to be used by the users to subscribe and to publish their videos. It is the client side.

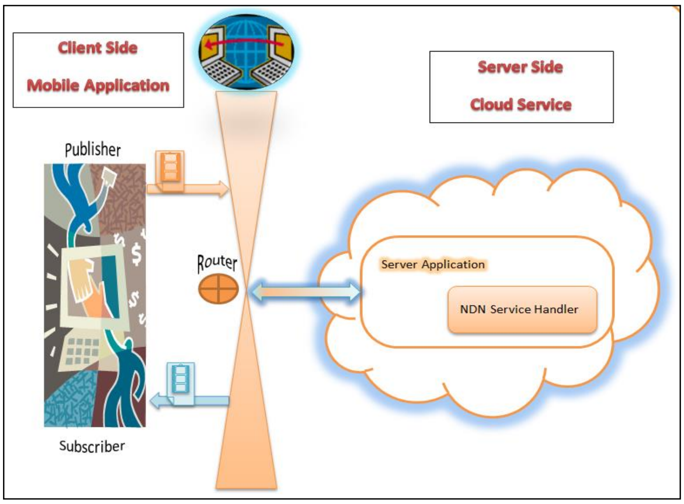

Figure 4the Main Components of the Implementation

The NDN Service handler is responsible for three functions which are:

- Forward Interest and Data packet
- Maintain basic data structures like the Content Store (CS), the Pending Interest Table (PIT), the Forwarding Information Base (FIB)
- Implements the packet processing logic (NFD Team, 2014).

To evaluate the performance of the proposed event mating algorithm, we will use synthetic and real-world workload and perform experiments by using CloudSim (http://www.cloudbus.org/cloudsim/) or **OpenStack** (http://www.openstack.org/) test-bed. We will also compare the performance of the algorithm with existing approaches in terms of evaluation/matching cost, scalability with respect to the number of subscriptions as well as the number of events,

Methodology in Research:

The Used Technology

CCN (Content-Centric Network)

CCN, which is described earlier, provides content requirements of the users. It aims to change content to be an entity and is based on the content mode of Publish/Subscribe model. Consequently, it offers loosely coupling of the subscriber and the publisher, and flexibility for the users to subscribe to the interested information. It is suitable for large amount of information (Chen, Arumaithurai, Fu, & Ramakrishnan, 2012)

NDN (Named Data Networking)

NDN is a popular implementation of CCN. Example of NDN is content streaming. A chief component of NDN is the broad use of caching at each hop of the network. Consequently, NDN requires that to process every NDN router packet, the content is requested and stored in the cache to be used for the following requests (Chen, et al. 2012).

To Communicate in NDN a subscriber sends an *Interest* packet, that carries a name that specify the wanted data. For example, a subscriber may request /videos/morning.mpg. A router remembers the interface of the requester, IP protocol for example, and then forwards the Interest packet by searching the name in its *FIB (Forwarding Information Base)*. When content is published, a *Data* packet is sent, which carries both the name and the content of the data, together with a signature by the producer's key. This *Data* packet traces in reverse the path created by the Interest packet back to the subscriber. Interest packets are *routed* towards data producers based on the names (Zhang, Estrin, Burke, Jacobson, Thornton, Smetters, ... & Yeh, (2010).

The major components of the NDN architecture are listed below:

1. Names

The naming scheme evolve independently from the network. NDN design adopts hierarchically *structured* names, e.g., a video produced by Ali may have the name /Ali/videos/Morning.mpg, where '/' indicates a boundary between name components. This hierarchical structure represents relationships between pieces of data and allows routing to scale. This hierarchical structure is based on the hierarchical structure of IP addresses (Zhang, et al. 2010).

2. Routing and Forwarding

NDN routes and forwards packets on names to eliminate four problems based on using the IP architecture, which are: address space exhaustion, NAT traversal, mobility, and scalable address management. There is no address exhaustion problem since the namespace is boundless. There is no NAT traversal problem since a host does not expose its address to offer content. Mobility, which needs changing addresses in IP, no longer breaks communication because data names need no changing. Finally, address assignment and management is not needed in local networks. Routing can be done similar to IP routing but instead of broadcasting IP prefixes, *name prefixes* are used. This broadcasting is propagated through the network via a routing protocol, and every router builds its FIB based on received routing broadcasting. NDN inherently supports multipath routing. The first Data coming back will satisfy the Interest and be cached locally; later arriving copies will be discarded (Zhang, et al. 2010).

3. Caching

When an NDN router receives an Interest, an NDN router first checks the Content Store. If there is a data whose name falls under the Interest's name, the data will be sent back as a response. The Content Store, is the buffer memory in a router. Both IP routers and NDN routers buffer data packets. NDN routers are able to reuse the data since they are identified by persistent names (Zhang, et al. 2010).

4. Pending Interest Table (PIT)

The PIT holds the arrival interfaces of Interests that have been forwarded but are still waiting for matching Data. This information is needed to send data to their subscriber. PIT entries need to be timed out quickly to maximize the usage of the PIT(Zhang, et al. 2010).

5. Transport

The NDN architecture has no separate transport layer. It moves the functions of transport protocols up into applications. Multiplexing and demultiplexing among application processes is done directly using names at the NDN layer (Zhang, et al. 2010).

NFD (NDN Forwarding Daemon)

NFD is based on NDN. NFD is intended to provide modularity and extensibility to enable easy experiments with new protocol features, algorithms, and applications for NDN (Afanasyev, Shi, Zhang, Zhang, Moiseenko, Yu, ... & Team, N. F. D. 2014).
NFD contains the following modules:

1. NDN Library, Faces, Tables ,Core, and Tools

It includes all the classes and services of NDN including the faces class to provide abstraction on top of transport layer. The included tables are: Content Store (CS), the Pending Interest Table (PIT), and the Forwarding Information Base (FIB) (Afanasyev, 2014).

2. Forwarding

Implements basic packet processing pathways, which interact with Faces, Tables, and Strategies. Strategies is a major part of the forwarding module. It implements a framework to support different forwarding strategies in the form of forwarding pipelines (Afanasyev, 2014).

3. RIB Management

It manages the RIB (Routing Information Base). The RIB may be updated by different parties in different ways, including various routing protocols by system admins. It processes all requests to generate a consistent forwarding table, and syncs it up with NFD's FIB, which contains only the minimal information needed for forwarding decisions (Afanasyev, 2014).

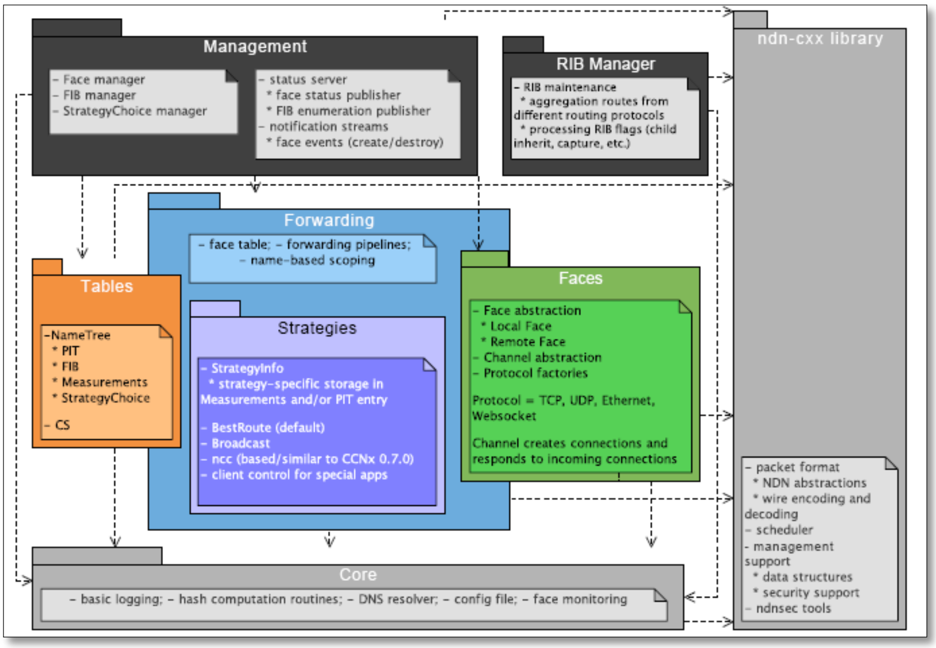

Figure 5: NFD main modules

In order to process data in NFD, many steps are to be done. the steps are:

1. Packets arrive at NFD via Faces. "\Face" is a simplification of \interface". It can be a physical interface such as operating directly over Ethernet, or an overlay tunnel such as operating as an overlay above TCP or UDP. In addition, it can communicate via a UNIX-domain socket. The Face reads incoming stream via operating system API, gets rid of link-layer headers and delivers network layer packets (Afanasyev, 2014).
2. A network layer packet, which may be an Interest or Data, is processed by forwarding pipelines. NFD's is stateful, and the processing is differentiated based on the packet itself and the forwarding state, which is stored in tables(Afanasyev, 2014).
3. If the packet is an interest.
   - It is inserted to the Interest table (PIT). If the Interest is not a duplicate, a lookup is performed on the (CS). If there is a matching Data packet in CS, that Data packet is returned to the requester. Otherwise, the Interest needs to be forwarded.
   - A forwarding strategy is followed based on names. A match is made based on name is performed on the Strategy Choice table, which contains strategy configuration. The strategy defines whether, when, and where to forward the Interest. in the meanwhile, it gets the routing information from the Forwarding Information Base (FIB)
   - The Interest goes through a few more steps in forwarding pipelines, and then it is passed to the Face. The Face adds link-layer header and sends the link-layer packets as outgoing stream or datagrams via operating system API (Afanasyev, 2014).
4. if the packet is Data
   - The Interest table is checked to see if there are matching PIT entries. All matched entries are selected for further processing. If there is no matching, it is dropped. Otherwise, the Data is stored to the Content Store.
   - Finally, the Data is sent to all requesters, recorded in downstream records of the PIT entries. The same process of sending an Interest is done to send a Data via a Face (Afanasyev, 2014).

The Details of proposed Implementation

The Newly Added Features

The implementation of this research is based on NFD. The enhanced added features are:

1. Working specifically on video streams. NFD is a general framework that can work with text and small video streams that do not exceed 4000KB. Programming modification are done to make it work with large video files with no limitation to size,
2. Making the whole framework hosted on Cloud and doing the required modifications.
3. Adjusting the matching algorithm to be based on a group of dynamic attributes of the video file rather than using the name which is provided by NFD
4. Adding multi domain feature to NFD framework to enhance performance.
5. Dynamically adapt the system to fit the number of current users
6. Creating an Android application to be used by the end user to publish and subscribe to video surveillance and events.

The main components

Consequently, the main components of the proposed system are:

1. Clouding using *NFD Component*
2. IP Multi-casting using *jNDN Component*
3. Video Surveillance Files Transferring using *jNDN Component*
4. Publish/Subscriber Model using *NDNClient Component*
5. Matching Contents Model using *NDNClient Component*
6. Mobile Application using *NDNClientAndroid Component* to work specifically for this implementation.

The components are communicating in order which is illustrated in the following figure. The NFD Component communicate jNDN which communicate NDN Client which communicate NDNClientAndriod component and the vice versa.

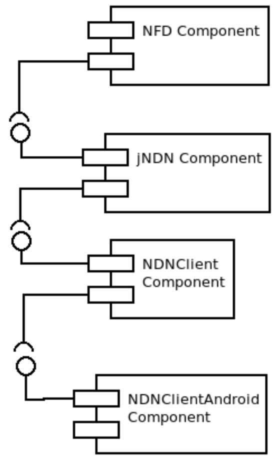

Figure 6: The order of communication of components in the proposed system

Description of Main Components:

1. Clouding, using NFD Component based on the NDN protocol, the NFD Server is hosted on a Linux Ubuntu Operating System. The clouding is hosting software as a Service. The cloud server handles all the tables and forwarding services such as FIB, RIB, SC..etc. It will create a Face using IP or URL to access and get the services of host server. Open-stack clouding provider is to be used.
2. IP Multi-casting: datagrams are to be sent to a group of interested Subscribers in a single transmission Using jNDN Component and using TCP or UDP
3. Video Surveillance Files Transferring: Using jNDN Component, The Resources which are Video Surveillance Files are be transferred.
4. Publisher/Subscriber Model: Using NDNClient Component, a Publisher will be able to register his own "Video file" on the NFD server. The associated specific prefix name, description, and specific list of content are to be registered as well. The Subscriber sends an interest (subscribtion) using a prefix name and matching is made based on contents of the attributes. Server will allocate prefix name and matched contents attributes, then will let the routers doing their jobs, which is sending and receiving the required data packets between the Publishers and the Subscribers. Prefix names create subdomains in the network to enhance performance. the prefix name represent the mobile network oprator.
5. Matching Content Model: Using NDNClient Component a matching video has at least 50% of the attributes identical to the publisher's attributes. Publishers can send one or more Contents Attributes, each content has a Name and a single value. eg.: Event = Regular, Time = 12:30, Persons = 3, Place = Riyadh. On the other hand, Subscribers can send one or more Contents Attributes to be evaluated. Each each Contents Attribute have a single or a range of values. eg.: Event = Blood, struggling, Some Properties Stolen; Time = from 10:00 to 15:00; Persons = from 1 to 5; Place = Riyadh, Jeddah, Dammam. For each published video file, there will be four obligatory attributes to fill which are Event, Time, Place, and number of Persons. The user can add more attributes dynamically.
6. Mobile Application: Using NDNClientAndroid Component and Android Operating System, the Mobile Application is operated. It has with three main screens which are: Settings, Publisher, Subscriber. The publisher screen is used to add the attributes and their values as well as the video file to be sent. The subscriber screen is to be used to enter the attributes of interested videos and a range of values for each attribute. The settings screen is to define user preferences. For example, it is used to choose the default upload and download folders in the mobile device, and to define the default domain which is the mobile network operator.