A Unified Packet Core Network Architecture and Drone Prototype for ID/Locator Separation

Shoushou Ren, Yongtao Zhang, Shihui Hu
2012, Network Technology Lab, Huawei Technologies Co., Ltd., Beijing, China
E-mail: {renshoushou, zhangyongtao3, hushihui}@huawei.com

Abstract—**In recent years, new Internet applications are raising restrict requirements and great challenges to the Internet. The current IP-based Internet architecture cannot accommodate well with these applications and cannot meet people's demands as before. This is mainly caused by the overloading of Internet protocol (IP) address semantics, namely, an IP address represents not only the location but also the identity of a host. To address this problem, researchers have proposed to replace the IP namespace with separation of namespaces for identities and locators. In this paper, we propose a Unified Packet Core (UPC) network architecture based on Identity Oriented Network (ION) to realize the separation of identities and locators. The UPC architecture also provides a unified access network gateway, which can support hosts to access Internet with multiple Radio Access Technologies. Further, a drone prototype implementation is also designed and described for the validation of the UPC architecture. The prototype realizes the ID-based connection between a moving drone and a fixed stationary endpoint. It is also verified that the ID-based connection can be kept continuous even when the drone moves across different gateways. The prototype shows that the basic idea of ID/Locator separation is a feasible and positive evolution of the current Internet architecture.**

Keywords- ID/Locator separation; Unified Packet Core; identifier; locator; handover.

I. INTRODUCTION

The current Internet architecture which has been built on top of the Internet Protocol (IP) was designed for a very different environment from today's networks. Early versions of the Internet Protocol were designed in the 1970's, at which time the primary application of Internet was a very rudimentary form of messages, like email. After that the landscape of networks has changed dramatically with the development of Internet technologies, and many of the initial Internet architecture tenets have changed too. The decade of 90's and early 2000's witnessed the coming of the mobile era, in which cellular networks have gradually adopted IP as the underlying protocol and merged with the Internet in the 3rd Generation and Long Term Evolution (LTE). From then on, the concept of mobility is well ingrained into the Internet functionality and mobile user equipment (UE) has become a common platform to connect people through rich mobile applications. Today, the 5th Generation Internet is already on the way to be realized, expected to bring more convenience to people's life.

These dramatic changes of Internet are now breeding more and more new applications such as Micro-message, Virtual Reality, Augment Reality [1][2][3], massive Internet of Things [4][5], etc. As these applications are becoming much more sophisticated than ever, more restrict requirements are also being raised and challenging the current Internet. The current Internet architecture, which is IP-based, cannot accommodate well with these applications and cannot meet people's demands as they were expected to in terms of three main aspects.

The first main aspect is the *growing mobility connectivity*. In recent years, communication behavior is swiftly shifting from PC based fixed computing to smartphone and tablets based mobile computing, and mobile data traffic has witnessed an explosion growing [6]. When a UE moves frequently from one place to another, the accessing gateway may also change consequently, which leads to frequently changes of IP address and brings severe problems. *a)* One problem is session interruption caused by frequently changes of IP address. This will further lead to severe packet loss, high latency, and finally cause great impacts on the quality of user experience. In the worst case, it may even interrupt the whole communication and UEs may lost each other. *b)* Another problem caused by the frequently change of IP address is the rapid expansion of routing tables, which brings great pressure on routers with terms of CPU and RAM. The huge size of routing tables can also costs a long time to converge and thus brings great network latency. Both of these issues challenge the scalability of Internet [7][8], while the existing solutions cannot solve these problems well. For example, the GPRS Tunneling Protocol (GTP) needs an anchor at a high position of Internet, bringing traffic roundabout and extra latency [9]. The Distributed Mobility Management (DMM) employs a mobility anchor to allow a mobile node to remain reachable after it has moved to a different network, which cannot overcome the short comings completely [10]. The Mobile IPv6 has the problem of Triangle Routing and high latency [11]. *c)* How to access to heterogeneous networks also remains challenge brought by the growing mobility. Since radio frequency resources are limited, Mobile Network Operators (MNOs) will have to bear more costs due to an increase in the number of cells per unit area. It is foreseen that the future 5G mobile network will become heterogeneous with multi-RATs (Radio Access Technologies) environment, where the existing different RAT cells and wireless LAN networks will be integrated and used [12]. Thus, mobility across heterogeneous access methods, for example from WLAN to LTE/5G network, must be supported. Besides, the IP address of a mobile UE or a fixed end host is now strictly managed by one specific MNO for commercial reasons. The mobility among different

MNOs in heterogeneous networks also remains a challenging issue in the current IP-based Internet.

The second aspect is the *inter-connecting between different applications* on mobile UEs. In most cases, the IP-based connection between two mobile UEs is absolutely driven by mobile applications (like Facebook, Micro-Message, etc.), which are monopolized by different Over-The-Top(OTT) service providers. Since UEs can only be identified by identifiers of different applications on the Application Layer rather than the IP address on the Network Layer, individual UE cannot identify and communicate with each other across different OTT service. For example, a Micro-message user cannot communicate with a Facebook user because they cannot even “see” each other on the Internet. Moreover, when a UE’s IP address changes along with its location when it moves, the TCP session/socket with other hosts will be broken down.

The third aspect is the *scale of everything connected*. The past few years has witnessed the rapid development of Internet of Things (IoT) and many new technologies have emerged to realize various IoT applications. Consequently, a massive number of IoT devices are being connected to the Internet progressively, presenting great challenges to the scalability of the Internet by demanding more IP addresses and more space in routing tables. The new features of IoT devices, like various types and complicated access environment, also propose more restrict requirements for the conventional Internet. Moreover, it is also a cruel issue with increased complexity when dealing with non-IP packets generated by some tiny IoT objects.

A common consensus is that these problems are mainly caused by the overloading of Internet protocol (IP) address semantics [13]. That is, an IP address represents not only the location but also the identity of an end host. Therefore, several new schemes, such as the Host Identity Protocol (HIP) [14][15][16] and the Locator/ID Separation Protocol (LISP) [17], have been proposed to replace the IP namespace in today’s Internet with a locator namespace and an identity namespace. In these schemes, a locator namespace consists of *locators* that represent the attachment point of hosts in the network, while the identity namespace consists of *identifiers* (ID), also known as endpoint identities (EIDs) that represent unique identities of hosts. When IDs are separated from their network attachment position information, packets destined for IDs are generally forwarded with the default routing method by using the locators as IPs. By decoupling an identifier from its locator, changes of a host’s location become transparent to the upper layers above TCP/UDP.

Consider the communication in the ID/Locator separation network between two end hosts, which are called ID hosts. Each host only needs to know the other’s ID before the connection is established, since only the ID can tell each other *who* the correspondent host is. While the locator is only used for packet forwarding in the Internet and it may change according to different access gateways. Thus, we call this kind of communication/connection as an ID-based communication/connection. In this paper, we propose a new network architecture called Unified Packet Core (UPC), based on the idea of Identity Oriented networks (IONs) [18]. The UPC architecture provides a unified access network gateway, which can support hosts to access Internet with multiple RATs, including 5G, LTE, WLAN, etc. The UPC architecture can also support ID-based communication between ID hosts. Further, we realize a drone prototype to verify the UPC architecture. In this prototype, a drone and a ground station are used as ID hosts. Each of them is with a unique and fixed ID, while their locators can change according to the access gateways. Our prototype ensures that the drone can establish an ID-based connection with the remote ground station. Moreover, when the drone accesses different gateways, the ID-based connection between the drone and the ground station is continuously maintained even when the drone’s locator changes.

The rest of this paper is structured as follows. In Section II, we summary some related works of the ID/Locator separation networks. Then, we introduce the basic framework of the UPC architecture in Section III. Section IV describes the topology of the drone prototype and the main entities in the prototype. Some detail designs are also presented in this section, including the ID packet format, packet encapsulation and decapsulation. In Section V, we show the handover process of the prototype in detail. At last, this paper is concluded in Section VI.

II. Related Works

Over the past several years, considerable efforts have been made on investigating solutions for the overloading of IP address semantics. Many protocols or architectures have been proposed based on the idea of ID/locator separation and some of them are briefly introduced below.

A. Host Identity Protocol

The Host Identity Protocol (HIP) is a famous protocol which aims to split the locator and the endpoint identifier roles of the IP addresses. HIP uses host identifiers at the host identity layer and IP addresses at the network layer. The identity layer is inserted between the transport layer and the network layer as a shim layer, Briefly, the HIP architecture proposes an alternative to the dual use of IP addresses as "locators" (routing labels) and "identifiers" (endpoint, or host, identifiers). In HIP, public cryptographic keys, of a public/private key pair, are used as host identifiers, to which higher layer protocols are bound instead of an IP address. By using public keys (and their representations) as host identifiers, dynamic changes to IP address sets can be directly authenticated between hosts, and if desired, strong authentication between hosts at the TCP/IP stack level can be obtained [14][15].

HIP does not change the architectural principles of the socket interface and the inserted identity layer is transparent to applications. In addition since it is based in public key identifiers it relies on well-known and proven security mechanisms that provide authentication, confidentiality and message integrity. However, the used cryptographic algorithms, especially those based on asymmetric key pairs, costs much in terms of CPU. HIP may impact user experience when CPU and battery power are limited in mobile devices.

B. *The Locator/Identifier Separation Protocol*

The Locator/Identifier Separation Protocol (LISP) was originally designed and developed to solve the scalability problem of the routing system proposed by the Routing and Addressing Workshop of Internet Advisory Board. LISP is a network protocol that separates the conventional IP addresses into two new numbering spaces: Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). It provides a set of functions for routers to exchange information used to map from EIDs that are not globally routable to RLOCs. It also defines a mechanism for these LISP routers to encapsulate IP packets addressed with EIDs for transmission across a network infrastructure that uses RLOCs for routing and forwarding.

Three main entities are designed in LISP, namely Ingress Tunnel Routers (ITRs), Egress Tunnel Routers (ETRs), and a mapping system. When an end host in the local LISP site needs to contact a remote end host, it sends a normal (IPv4 or IPv6) packet with the destination EID as destination address. This packet is intercepted by one of the site's ITRs. To forward the packet, the ITR first needs to obtain at least one of the RLOCs of the destination ETR from the mapping system. Then the ITR encapsulates the packet with a LISP header and sends out the packet. The LISP header contains the locator of the ITR and the destination ETR. When the destination ETR receives the packet, it strips the LISP header and forwards it to the destination end host [17].

LISP can be incrementally deployed in the current Internet, while no changes are required to either host protocol stacks or to the "core" of the Internet infrastructure. However, the EID in LISP is actually still IP address. Thus, LISP doesn't works as well as expected with terms of the mobility issue, even though there were relative drafts have been proposed to deal with it.

C. *Identity Oriented Network*

In order to meet the aforementioned deficiencies and inefficiencies of the current architecture based on IP, our colleagues proposed the Identity Oriented Networks (IONs) based on the idea of Identity and Location separation [18].

Since the concept behind ION is applicable to any underlying network infrastructure, it was proposed to work in a backward compatible manner with the current Internet and didn't intend to change the IP infrastructure. The basic idea of ION is to insert a naming/identifier sub-layer in the protocol stack, generally as an over-layer of IP stack. The ION framework is briefly described in Figure 2 and the details are out of scope for this paper. Since identity and locators are separated, ION expands network layer concept to accommodate ID in the following manner.

- *ID layer* is a distributed function responsible for ID management and authentication services.
- *Mapping system:* An ID/location resolution system is introduced which maintains mappings between a host and its location.

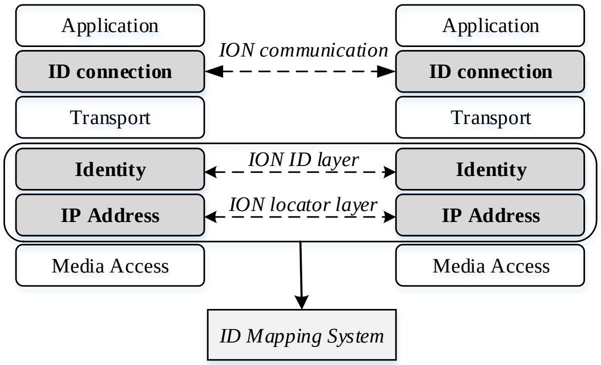

Figure 1. Brief framework of ION.

- *ID based connection:* In order to inter-connect two endpoints independent of network address an ID aware socket connection.

The ID oriented architecture relies on defining the ID of the user and a mapping or binding to the location of the user in order to forward traffic. The ID namespace comprises the whole IPv4 and IPv6 address space to enable it to interoperate with traditional applications, while newer applications can use the newly defined ID. ION architecture enhances traditional network layer with identity awareness. Some advantages ION scheme include: a) communication of non-IP devices such as IoT, b) a smoother and seamless location agnostic mobility and c) cross-silo communication across applications working with same network entities..

III. THE UPC ARCHITECTURE

To fill the gap between the conventional IP-based Internet and the requirements for future networks aforementioned in Section I, we design a Unified Packet Core (UPC) architecture in this section, based on the framework of ION. Compared to the Evolved Packet Core (EPC), the UPC architecture can support ID-based communication by nature and provide a unified core network which allows ID hosts to access the core network via multiple RATs.

Figure 2 shows the overview of the UPC architecture, which consists of mainly three new components, namely the Universal Access Gateway (UAG), the ID-Locator Mapping System (ILMS) and the Inter-Operation Gateway (IOG).

The UAG is the edge access gateway of the UPC architecture. It is extended from the ION gateway and can support multiple radio access technologies, as well as the wired access. The UAG is in charge of locator assignment, locator registration and packets encapsulation/decapsulation. Specifically, when an ID host, such as the drone in the following prototype, is online and tries to access to a UAG for the first time, the UAG assigns to it an IP address as locator. Then the UAG registers the ID/Locator mapping item of the ID host to the mapping system and caches the item until the host leaves. Moreover, the UAG can also perform packet forwarding function as a legacy gateway if a conventional IP host accesses.

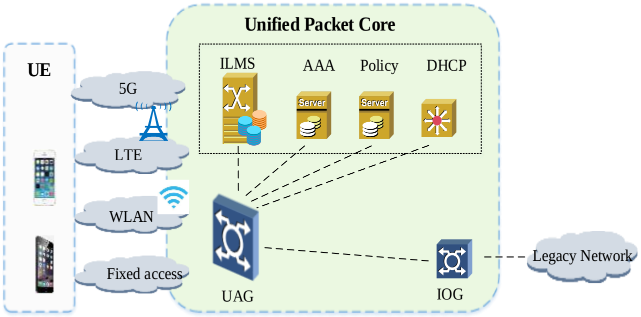

Figure 2. Brief architecture of UPC.

The ILMS, which is an extension of the ID mapping system in ION, is another core entity of the UPC. It stores all the ID/Locator mapping items that have been registered. Once an ID host is assigned a locater by its access UAG, the ID/Locator item will be registered or updated to the ILMS. If an ID host wants to communicate with other ID hosts, the accessing UAG can retrieve the demanded locator from its local cache or from the ILMS. The IMLS is designed as a distributed system and independent from all access networks. It helps to realize seamless mobility among heterogeneous networks.

The IoG is a gateway that helps to connect UPC with the legacy network. Besides, the UPC also provide other legacy service such as AAA (Authentication, Authorization and Accounting) service, DHCP (Dynamic Host Configuration Protocol) service (mainly for locator assignment in UPC), Policy routing, etc.

The protocol stack of UPC is shown in Figure 3, which is exactly the same with ION. An ID sub-layer is inserted in the legacy protocol stack, generally as an over-layer of IP stack. On the users' side, hosts are aware of the separation of ID and Locator, and each host is assigned with a global ID. The ID is the only identifies that can represent a host, rather than the legacy IP address, which can only represent the where the host is.

The UPC architecture which can support ID-based communication is a feasible solution to the aforementioned problems. Firstly, with ID/Locator separation, the network are no longer in charge of the mobility management. When a mobile UE moves, the network need not to know WHO the end host is, while it only cares about WHERE the packets should be forwarded according to the UE's locator. Traffic anchors no longer exist in mobility scenario and traffic roundabout can be avoided. Secondly, the core network is decoupled with the access network completely in UPC, which allows seamless roaming in heterogeneous networks. Thirdly, the global ID of hosts enables that an ID host is always on line and reachable at any time. Furthermore, the design of ID can also accommodate well with massive IoT objects. Last but not the least, inserting ID layer also gives new possibilities to change the upper layer by having ID aware applications above the ID layer.

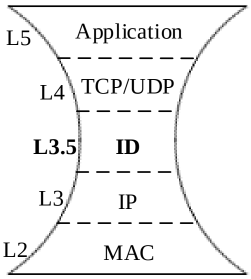

Figure 3. Prototol stack of UPC.

Note that though it is easier to understand and accept the ID/Locator separation protocol stack as designed in Figure 3, there are other options. For example, the ID oriented architecture may also reside directly on the L2 level alternatively, in which case the mapping is between the ID and the L2 layer MAC address. The evolved architecture using ID oriented networks aims at using the IP addressing, but inserting an Identity layer. This also gives new possibilities to change the upper layer by having ID aware applications above the ID layer. Besides, IDs of end hosts may be set before leaving the factories or assigned after that by some organizations and this is out of the scope of this paper.

IV. Design of the Drone Prototype

A. Topology

The topology of our drone prototype is depicted in Figure 4, which mainly consists of following five kinds of entities:

- *Drone*: The drone is an ID host with a unique and fixed ID. When it accesses a UAG, a locator will be assigned, which is used to locate where it is. The drone is equipped with a camera for shooting real-time video when flying across different UAGs. It is controlled by a ground station and the video will be transmitted to the ground station via ID-based communication. In this prototype, we use the IPv6 addresses those are with prefix *2F00* as IDs for convenience.
- *UAGs*: Three UAGs are deployed in our prototype and the drone flies randomly in the area covered by the three UAGs.
- *Access Point (AP)*: Traditional APs. The drone access to a UAG via an AP. Only one AP is deployed under each UAG for the case of layer-3 handover [19][20], which will be further explained in the next section.
- *Ground Station (GS)*: the GS, which is also an ID host, is the controller of the drone. It receives and displays the video shot by the drone.
- *ID-Locator Mapping System (ILMS)*: the ID mapping system.
- This prototype aims to achieve the following goals: 1) Realize an ID-based communication between two ID hosts: the drone and the remote GS; 2) When the drone's locator changes while roaming across different UAGs, the ID-based communication could be kept continuous.

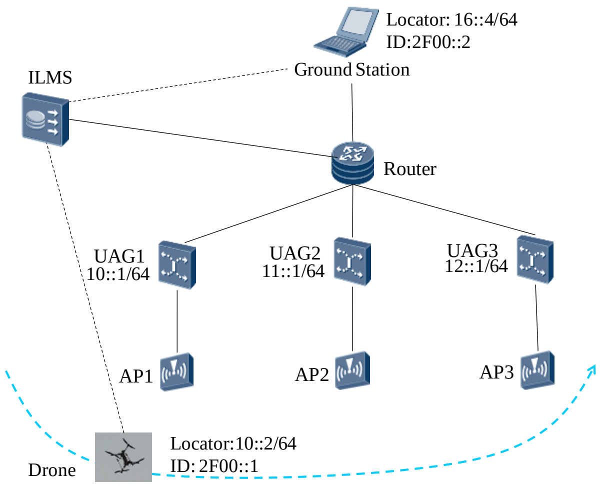

Figure 4. Topology of the drone prototype.

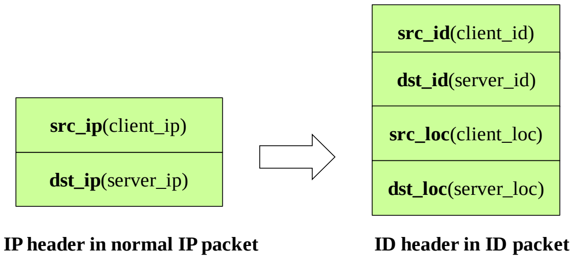

Figure 5. Changes of IP header in ID packet.

B. *Packet Format*

Packets in an ID-based communication are called ID packets in this prototype, while the traditional packets are called IP packets. As is shown in Figure 5, the format of ID packets in UPC changes accordingly with the protocol stack. The main change in ID packet lies in the IP-layer header. The tuple <*src_ip*, *dst_ip*> in a normal IP packet is replaced by a new header of tuple <*src_id*, *dst_id*, *src_loc*, *dst_loc*> in the ID packet. In this prototype, the IP address in the normal IP packets has the same meaning with the locator in ID packets.

C. *Packet Encapsulation and Decapsulation*

In this subsection, we take the drone as an example to show how the ID packets are encapsulated and decapsulated.

● *Packet Encapsulation*

The main encapsulation process of packets in an ID-based communication is depicted in Figure 6.

When a normal packet is generated by the TCP layer at the drone, it will be first checked by an *is_ID()* function to determine whether it belongs to an ID-based communication according to its *src_ip* and *dst_ip*, which can be found in the five-tuple of TCP sockets. If the *src_ip* or *dst_ip* is with IPv6 prefix *2F00*, the packet will be further encapsulated into an ID packet by the *id_out()* function with following steps. *1)* If the *2F00* prefix is detected by the *id_out()* function, the drone firstly tries to get the locator of the GS from local cache, i.e., its own cache and the UAG's cache. *2)* If fails, a request will be sent to the ILMS for the retrieval of GS's locator according to its ID. *3)* Then, the normal packet will be encapsulated according to the format shown in Figure 5. Specifically, the drone's locator, i.e., the *src_loc*, is assigned when it accesses a UAG. The *dst_loc* is retrieved from caches or from the ILMS. Since we use the ipv6 address with prefix *2F00* as id, the *src_id* in id packet is the same with *src_ip* in the normal IP packet, and the *dst_id* in ID packet is the same with *dst_ip* in the normal IP packet. *4)* Now, the original packet has been encapsulated to an ID packet at Layer 3 and it will be sent as normal packets to Layer 2 and then sent out.

Otherwise, If the *2F00* prefix is not detected by the *id_out()*, the packet will be encapsulated as normal IP packet and sent out.

At last, the encapsulated ID packet or normal IP packet will be sent to the access AP and UAG. The access UAG just treats the locator as the normal IP and forwards all packets as usual according to the routing table.

● *Packet Decapsulation*

The decapsulation process of ID packets is shown in Figure 7. Once a packet is received by the hardware of the drone, it will be sent to the IP layer and checked by the *is_ID()* function to determine whether it is an ID packet or not. If the packet is a normal IP packet, it will be sent to the TCP layer directly. Otherwise, it will be treated as an ID packet and further decapsulated by the *id_in()* function. The *id_in()* function strips the locator header, i.e., the *src_loc* and *dst_loc* fields. Then the stripped packet will be further handled as a normal packet.

It should be noted that in this prototype, the ID hosts (i.e., the drone and the GS) are designed to be aware of ID/locator separation. The locator header of ID packets is encapsulated and decapsulated at the drone for realization convenience. In fact, the ID/locator separation network can also be designed as that the hosts are completely unaware of ID/locator separation, in which way the process of packet encapsulation and decapsulation will be embedded into gateways rather than end hosts.

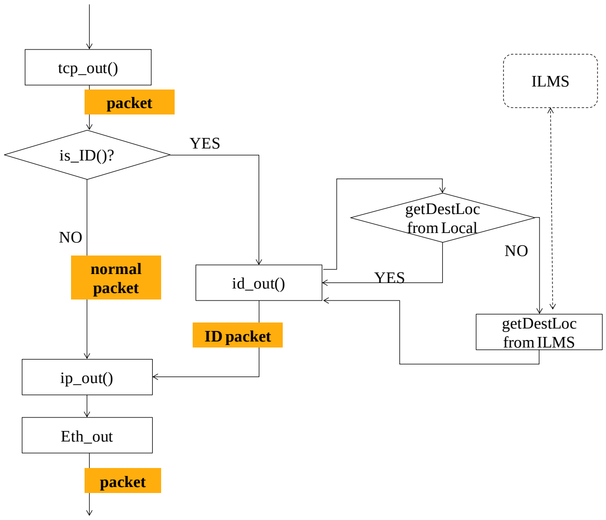

Figure 6. Packet encapsulation process.

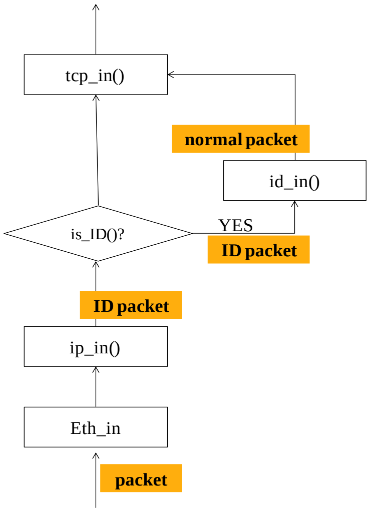

Figure 7. Packet decapsulation process.

V. Handover with Continuous ID-based Connection

In this prototype, we mainly aims to prove that the UPC architecture can realized seamless mobility without session being interrupted when the drone's locator changes. Figure 8 shows the change of data flow in the handover process. The drone was firstly connected to the GS via UAG1, as the dashed green line shows. When the drone moves out of the coverage of AP1, the handover will be handled. When the handover is finished, the drone communicates with the GS via UAG2, which is depicted in the red solid line. During the handover, UAG1 caches the packets those are destined to the drone but still in fly. These packets will further be forwarded to the drone vial UAG2 with an IP tunnel, as the blue dashed line shows.

A. *Handover with single Network Interface Card (NIC)*

In the first experiment, the drone is equipped with one NIC. When the drone moves out of the range of its access AP, a handover process must be handled. Since the layer-2 handover does not lead to changes of locator, we only consider the layer-3 handover in this prototype. Only one AP is deployed under each UAG for convenience of layer-3 handover. When the drone flies across different APs, its locator changes and a layer-3 handover will be activated.

The detail handover process is shown in Figure 9.

Step 0: the drone, with ID *2F00::1* and locator *10::2* assigned by UAG1, has already established an ID-connection with the GS, whose id is *2F00::2*.

Step 1: Once the signal strength of current AP is lower than a threshold, the handover process will be activated. Then the drone sends a handover notification to UAG1. Upon receiving the notification, UAG1 will send a confirmation to the drone.

Step 2: UAG 1starts to cache packets with *dst_loc* or *dst_ip* equals to the drone's old locator *10::2*.

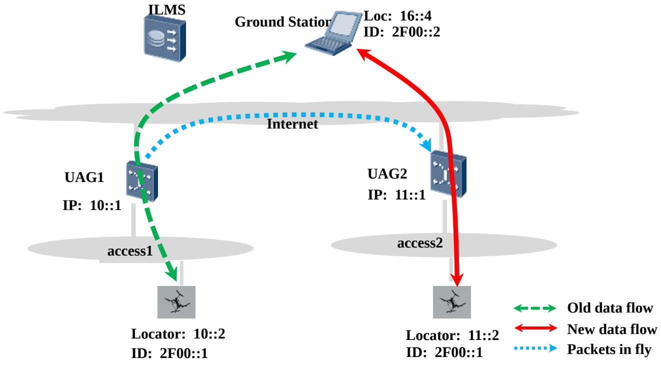

Figure 8. Data flow during the handover.

Step 3: After receiving the confirmation from UAG1, the drone disconnects from the AP under UAG1 and starts to send probe requests on other channels. If success, a new AP will be selected, say AP 2 under UAG2. The drone tries to connect with AP2. If success, the drone will get a new locator *11::2*, which is assigned by UAG2.

Step 4: Then the drone uses the new locator to notify the ILMS and the GS that its locator has changed from *10::2* to *11::2*. The ILMS and the GS then update their mapping item related to ID *2F00::1* and return the confirmation to the drone. At the same time, the drone will also send its new locator to UAG1, notifying UAG1 that it has successfully finished the handover and requests for the cached packets. Upon receiving the notification, UAG1 also sends a confirmation to the drone.

Step 5: With the same ID *2F00::1* and the new locator *11::2*, the drone continues the old ID-based connection with the GS. The packets in fly will also be tunneled to the drone according to its new locator via UAG2.

Though the NIC must change its working channel, which brings interruption on the Physical layer, the session on upper layer is not interrupted. From the view of the GS, the

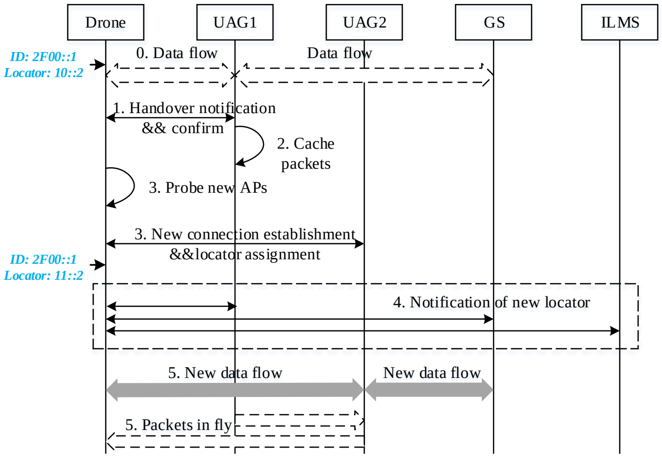

Figure 9. Handover process with single NIC.

corresponding node in the ID-based connection is always the drone with ID 2F00::1 during the handover process. Thus, change of the drone's locator is transparent to the upper layers including TCP/IP, and the ID-based connection can be kept continuous.

B. *Seamless Handover with two NICs*

To improve the performance of the handover, we then equip the drone with two NICs for another experiment. Two NICs on the drone take turns to perform the function of data transfer and probe. The process is detailed in Figure 10.

Step 0: Initially, the drone is using NIC_1 as the data card to associate with AP1. Locator (*10::2*) of the drone is associated with NIC_1. At this time, NIC_2 is idle and ready for probing other available APs around.

Step 1: When the signal strength of the current AP is lower than the threshold, NIC_2 is waked up to probe new APs.

Step 2: Once a new available AP (say AP2 under UAG2) is discovered and selected, the drone tries to establish new connection with AP2 via NIC2, and NIC2 will be assigned a new locator *11::2* by UAG2 (via AP2).

Step 3: Notifications will be sent to both ILMS and the GS, informing that the current valid locator of the drone is *11::2*. Note that at this time, the drone has two different locators and one unique ID *2F00::1*.

Step 4: The subsequent ID packets, destined to ID *2F00::1*, from the GS to the drone will be forwarded with the destination of new locator *11::2* via UAG2, then finally to the drone. Besides, the connection between NIC_1 and AP1 will be held for a period of time for receiving the packets in fly, which are still forwarded with the old locator *10::2*.

Step 5: At last, the drone disconnects with AP1. Now, NIC2 is in charge of data transfer instead of NIC1, which is idle and waiting for the next handover.

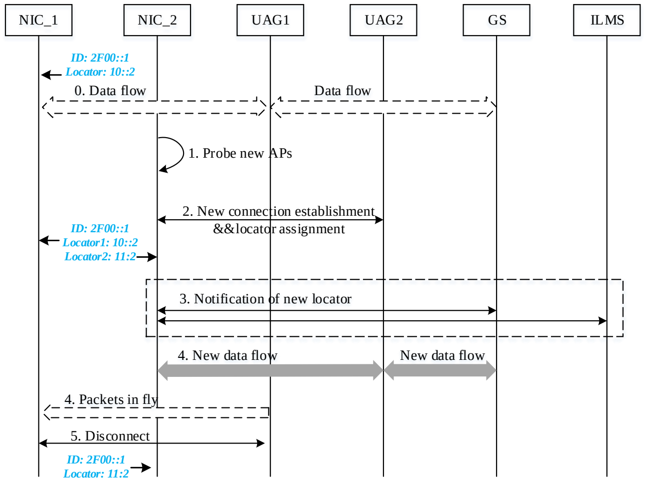

Figure 10. Handover process with two NICs.

During the handover process, since the end hosts of the ID-based connection are always ID 2F00::1 and ID 2F00::2, change of the drone's locator is transparent to the upper layers above including TCP/IP, and the ID-based connection can be kept continuous.

VI. CONCLUSION

In this paper, we proposed a Unified Packet Core architecture based on the Identity Oriented Networks, in which ID is designed as the only identifier of hosts, while the locator is only used for routing and packet forwarding. A relative prototype was also designed to realize the ID-based communication. Some protocol principles are also presented to define the format, as well as encapsulation/decapsulation of id packets. The prototype was proven to be able to support ID-based communication and seamless roaming of ID hosts. As the basic idea of ID/Locator separation is now widely accepted by researchers and Internet organizations such as IETF. This paper shows that this basic idea is a feasible and positive evolution of the current IP-based Internet Protocol.

In the future, there still remains many important issue to be dealt with before the universal deployment of the ID/Locator network architecture. The ID/Locator mapping system, which is at the heart of the ID/Locator network architecture, is the first issue to be considered. An ideal mapping system should be high reliable and with high efficiency. How to design such a mapping system still remains an important problem. Secondly, in order to realize the interoperation among different ID/Locator solutions, a generic control plane is also necessary to be design. Last but not the least, the publication and management of identifiers also needs to be considered carefully. All these issues will be further investigated in our future work.

REFERENCES

[1] S. Ren and Y. Zhang, "A ID/Locator Separation Prototype Using Drone for Future Network," The Tenth International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ), 2017.

[2] Digi-Capital, "Augmented/Virtual Reality to hit $150 billion disrupting mobile by 2020".

[3] "Next Generation Protocols – Market Drivers and Key Scenarios," ETSI White Paper No. 17, http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp17_Next_Generation_Protocols_v01.pdf

[4] C. Perera, C. H. Liu, S. Jayawardena, and M. Chen, "A survey on internet of things from industrial market perspective," IEEE Access, vol. 2, pp. 1660-1679, 2014.

[5] A. Al-Fuqaha, M.Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347-2376, 2015.

[6] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update," 2016–2021 White Paper.

[7] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering," IETF Internet Standard, RFC 3272, May 2002.

[8] "BGP Routing Table Analysis Reports," http://thyme.apnic.net/.

[9] "3GPP TS 29.060 General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface," 2013.

[10] H. Chan, Ed., X. Wei, J. Lee, S. Jeon, A. Petrescu, et al., "Distributed Mobility Anchoring," IETF Internet Draft, July 2017.

[11] R. Koodli, Ed., "Fast Handovers for Mobile IPv6", IETF Internet Standard, RFC4086, July 2005.

[12] "NGMN 5G White Paper," 2015.

[13] D. Meyer, L. Zhang, and K. Fall, "Report from the IAB Workshop on Routing and addressing," IETF Internet Standard, RFC4984, September 2007.

[14] R. Moskowitz and P. Nikander, "Host Identity Protocol (HIP) Architecture," IETF Internet Standard, RFC 4423, May 2006.

[15] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, "Host Identity Protocol, IETF Internet Standard," RFC5201, April 2008.

[16] T. R Henderson, J. M. Ahrenholz, and J. H. Kim, "Experience with the host identity protocol for secure host mobility and multihoming," In IEEE Wireless Communications and Networking, pp. 2120-2125, 2003.

[17] D. Farinacci, V. Fuller, D. Meyer, and D.Lewis, "The Locator/ID Separation Protocol (LISP)," IETF Internet Standard, RFC6830, January 2013.

[18] https://www.iaria.org/conferences2016/filesAICT16/AICT_Keynote_May_2016_Padma_V5.0.pdf

[19] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," IETF RFC 3775, June 2004.

[20] R. Koodli, "Fast Handovers for Mobile IPv6," IETF RFC 4068, July 2005.