

Presentation for iGate: NDN Gateway for Tunneling over IP World

Nitin Kumar¹ Sudipta Halder² Soumodipta Bose³

¹ International Institute of Information Technology, Hyderabad
2021202020

² International Institute of Information Technology, Hyderabad
2021202011

³ International Institute of Information Technology, Hyderabad
2021201086

October 15, 2022

IGate: What is it?

- It is a protocol which is responsible for establishing communication between two NDN nodes in different NDN networks separated by the IP network.

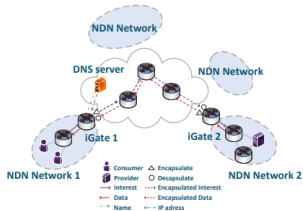


Figure 1: NDN-IP Interconnection Scenario. [1]

Challenges and Motivation

- Conversion from stateful(NDN) to stateless(IP routing) and vv.
- Gateway should reduce conversion overhead from NDN to IP and vice versa.

Architecture

To convert stateful to stateless, we use two tables which work in coherence with the already present PIT and FIB tables of NDN routers:

- ▶ Gateway Translation Table.
- ▶ Tunnel State Table.

These custom routers are renamed as iGate 1 and iGate 2 and are placed in NDN edge networks across a sea of IP network.

Gateway Translation Table

- ▶ A table that converts the requested data name into the IP address of the iGate gateway. Since names in NDN are hierarchical like DNS, it uses the longest prefix match to find the appropriate IP of IGate2 and uses the BIND [4] system for implementation. Efforts were being made to establish GTT using existing DNS [1].
- ▶ The following entry shows the conversion of named data “/bit/file” to its corresponding IP using the distributed [1].

GTT	
Name Prefix	IP address
/bit/	192.168.66.134
/bit/file	192.168.66.135
/bit/music	192.168.66.132

Figure 2: matching name in GTT [1]

Tunnel State Table

Tunnel State Table (TST)								
NDN_socket	TCP_socket	state	time stamp	Quintuple Information of tunnel				
				protocol	src_ip	dest_ip	src_port	dest_port
1	3	Green	100.125	1	192.168.66.135	192.168.66.138	10038	10038
2	4	Red	100.256	1	192.168.66.135	192.168.66.131	2345	2345
...

Figure 3: Tunnel State Table (TST) [1]

Completes the state switch between two kinds of packets in NDN and maps the data packet to the interest packet at iGate 2. It records the following information:

- ▶ **NDN_socket:** Socket ID is created when iGate 2 receives an encapsulated Interest packet from a new tunnel. Each socket has a one-to-one relationship in establishing the tunnel [1].
- ▶ **TCP_socket:** Socket ID that is created during the connecting or accepting operation. It keeps the tunnel beginning at iGate1 and ending at iGate 2 [1].

Tunnel State Table continued...

- ▶ **State:** It is used to describe the state of the tunnel to resolve various operations [1].
- ▶ **TimeStamp:** It enables iGate to know the last use time of a tunnel in real-time. If a tunnel has not been updated for a long time, it is considered lost and iGate will delete the entry saving space [1].
- ▶ **Quintuple tunnel information:** Consists of protocol, source IP, destination IP, and source port and destination port which is required for establishing a connection between the two tunnel gateways iGate1 and iGate2 [1].

Communication Process : At NDN edge 1

We will use TCP as Transport Layer to explain the entire process, for UDP it is rather straight forward as it is connection less.

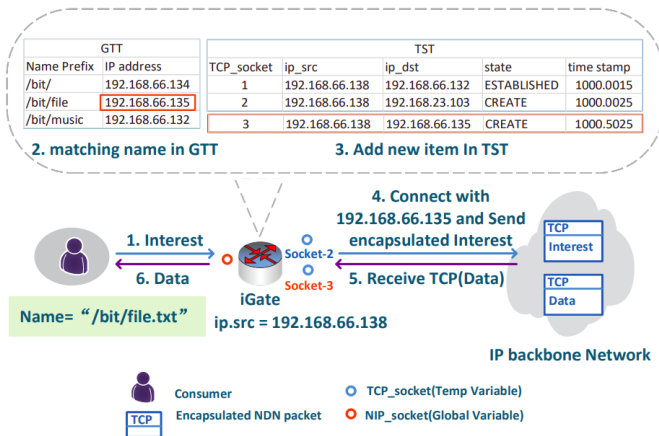


Figure 4: Communication process on Consumer [1]

Steps of Communication Process : At NDN edge 1

1. If content requested by consumer is not found in NDN edge 1, the Interest packet is forwarded to iGate 1 [1].
2. iGate 1 matches the data name in GTT to find the IP address of iGate 2 using the longest prefix match [1].
3. Create a TCP socket and add it as a new entry in TST table, if we are creating a new tunnel, or else look it up in TST table [1].
4. Send the encapsulated Interest packet through the IP network after completing the connect process with iGate 2 [1].
5. By keeping an eye on TCP sockets in TST, iGate 1 collects and decapsulates encapsulated Data packets sent by iGate 2 [1].
6. By finding the corresponding NDN socket to the TCP socket, and send the data packet back to the consumer in edge network 1 [1].

Communication Process : At NDN edge 2

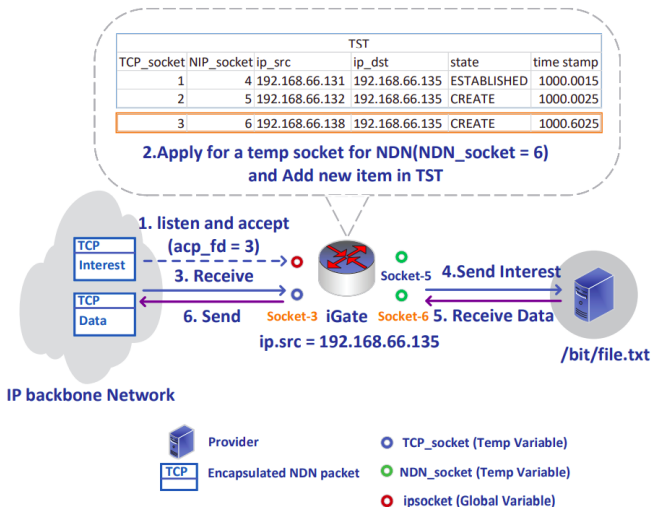


Figure 5: Communication process on Provider [1]

Steps of Communication Process : At NDN edge 2

1. At the designated port, iGate 2 allows connections via the IP socket which was known by iGate 1 [1].
2. Save the TCP socket produced by the accept procedure, then use it to apply for a temporary NDN socket in TST table [1].
3. Obtain the interest packet from IP network and decapsulate it [1].
4. Forward the interest packet to NDN edge network 2 using the temporarily created NDN socket [1].
5. Receive Data packets from the Producer from edge network 2 [1].
6. iGate 2 locates the corresponding TCP socket in TST table by the already mapped temporary NDN socket. The Data packet is encapsulated and send over the IP network [1].

Study : iGate vs PPTP

A real implementation of iGate was compared against a standard tunneling protocol that works over IP called PPTP using several factors of comparison.

Setup:

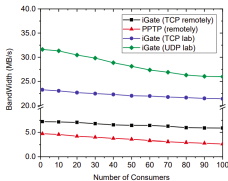
- ▶ iGate gateways were created using two custom linux kernels and deployed on either sides of the city in their own NDN edge networks and connected to each other over IP network [1].

Factors of comparison:

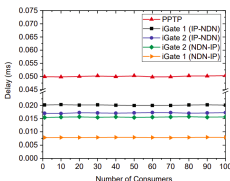
- ▶ Bandwidth (Transmission rate between two iGates).
- ▶ Conversion Delay (Processing time for Protocol conversion).
- ▶ Memory Usage (Memory required for execution).

Results : iGate vs PPTP

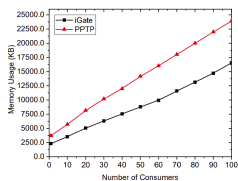
- ▶ iGate uses 85.03% less bandwidth than PPTP VPN. [1].
- ▶ The conversion delay of IP-NDN at iGate 1 is around 0.02ms and iGate 2 is around 0.016ms [1].
- ▶ The conversion delay of NDN-IP at iGate 1 is around 0.007ms and iGate 2 is around 0.015ms [1].
- ▶ iGate performs 30.30% better than PPTP in a complete communication process [1].
- ▶ Memory Usage: iGate(2304KB to 16555KB), PPTP(3271KB to 23911KB), showing 63.71% less memory usage [1].



(a) Bandwidth [1]



(b) Conversion Delay [1]



(c) Memory Usage [1]

Figure 6: Performance Evaluation [1]

Inference

- ▶ Increased bandwidth demonstrates support for high concurrent requests and shows the benefits of in-network caching [1].
- ▶ Delay of NDN-IP is less than that of IP-NDN because IP-NDN requires parsing of IP packets using self-defined communication rules compared to the simple encapsulation of NDN-IP [1].
- ▶ The low memory usage shows that iGate is lightweight and can later accommodate the introduction of other proxies and protocol gateways that are typical for an underlay approach [1].

Critiques

Assumption

- ▶ The IP address assigned to an iGate is not static which makes the GTT table entry of other iGate Nodes become inconsistent and hence needs validation.
- ▶ They have assumed that a DNS like strategy will work for getting name and IP mapping but, there are a lot of complications in doing that as this mapping will need a lot of memory (due to many to one mapping) as well as a lot of computation power, so we have proposed a solution to make it feasible.

Analysis/Results

- ▶ For Comparison of iGate with tunneling such as PPTP VPN, they didn't disabled the authentication protocol which explains the higher conversion delay in PPTP VPN.

Critiques continued...

Analysis/Results

- ▶ Performance of the iGate has been evaluated against PPTP VPN. The performance of PPTP VPN is not as per the latest VPN technologies like OpenVPN, L2TP, SSTP [5]. Hence, the bandwidth and conversion delay in PPTP VPN would not be as good as the latest VPN technologies. So, need to use the latest VPN technologies while measuring performance of iGate.

Technical Approach

- ▶ The fields such as Hop Count in NDN and IP should be synchronized before and after encapsulation/decapsulation.
- ▶ For iGate to be actually deployable in the real world it must have a mechanism to account load balancing before sending requests to other iGate.

Critiques continued...

Technical Approach

- ▶ Just like NDN, iGate also suffers from DoS attack via (Interest flooding attack).
- ▶ If requests are coming from multiple NDN for the same data then iGate has to do repeated work of encapsulation of the same data again and again, this can be avoided using iGate[Extended].
- ▶ The GTT and TST Table could reveal too much information about the data and location, using which DoS attack can be designed or Network Performance can be brought down.
- ▶ There are a lot of inconsistencies that can come due to fragmentation of encapsulated packets in the IP Network making the fragmented data object received at consumer useless.

Critiques continued...

Technical Approach

- ▶ iGate mainly focuses on integrating NDN edges into the IP world [1]. Other scenarios which need to be taken care of:
 - ▶ IP-IP communication in the NDN ocean [1].
 - ▶ NDN-IP communication in the NDN ocean [1].
 - ▶ NDN-IP communication in the IP ocean [1].
- ▶ Since the entries of Tunnel State Table (TST) are not encrypted, it is very easy for the attacker to find out the `src_ip` and `dest_ip` of each entry and hence uniquely identify each tunnel. Possible attacks can be packet injection, packet sniffing, packet capture.
TST table entries, packets passing through the tunnel should be encrypted, Inbound and outbound rules for packets should be incorporated to prevent this.
- ▶ Another attack possible on the tunnels is wormhole attack. Can be detected and prevented using AOMDV protocol.






Improvements and Extensions

- ▶ We have proposed iGate[extended] to solve most of the problems that iGate faces. It is an NDN node with the abilities of iGate and IP router.
- ▶ We have also proposed that even after finding a matching entry at GTT table, validation should be performed because the IP address given to iGates are not static.
- ▶ We have also proposed a mechanism to reduce the expected load the DNS server using the dedicated part of content store[iGate extended] for storing the mapping of name and IP address of other iGate.
- ▶ We need a conversion protocol to convert NDN packet to IP packet and IP packet to NDN packet using dual channel translation gateway [3].
- ▶ We have tried to solve the inconsistency that can come due to fragmentation of encapsulated data in IP network by giving the ability of reassembly and validation to the iGate[extended].

Improvements and Extensions continued...

- ▶ To prevent packet injection, packet sniffing, packet capture attack in the tunnel between two iGates, GTT(so that it does not reveal too much information about data and location), TST table entries should be encrypted, inbound and outbound rules for packets should be incorporated and the packets passing through the tunnel should be encrypted.
- ▶ We have also proposed a scheme to use the GTT table in such a way that it also accounts the load on other iGate nodes and request service from nodes which have less load.
- ▶ Since we have proposed the iGate[extended] to have content store it can avoid some redundant task such as encapsulating same data again and again(if multiple NDN network are requesting same data) by using its content store to cache the outgoing IP packet.

References

-  Zhu, R., Li, T., Song, T. (2021, July). iGate: NDN Gateway for Tunneling over IP World. In 2021 International Conference on Computer Communications and Networks (ICCCN) (pp. 1-9). IEEE.
-  Zhang, L., Estrin, D., Burke, J., Jacobson, V., Thornton, J. D., Smetters, D. K., ... Papadopoulos, C. (2010). Named data networking (ndn) project. Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC, 157, 158.
-  Fahrianto, F., Kamiyama, N. (2021, July). The Dual-Channel IP-to-NDN Translation Gateway. In 2021 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN) (pp. 1-2). IEEE.
-  ISC BIND Homepage, <http://www.isc.org/sw/bind/>.
-  <https://www.howtogeek.com/211329/which-is-the-best-vpn-protocol-pptp-vs.-openvpn-vs.-l2tpipsec-vs.-sstp/>.