Semantics and Deviation Aware Content Request

Lijun Dong
Futurewei Technologies Inc.
10180 Telesis Ct, Suite 220
San Diego, CA, U.S.A
lijun.dong@huawei.com

Richard Li
Futurewei Technologies Inc.
2330 Central Expressway
Santa Clara, CA, U.S.A
renwei.li@huawei.com

***Abstract*— When a content is published in the Internet, it is usually associated with many semantics information, which can be used to identify and discover the content. The current Internet protocols do not support the semantics aware content request very efficiently. By taking advantage of the recently developed Big IP Protocol (BPP), which brings the intelligence into the network while still maintaining the traditional communication paradigm, we propose in this paper a novel semantics and deviation aware content request scheme. It considers the possibility that a consumer may be able to accept the semantics information of the received content having certain deviation from his/her requirements. The paper proposes the structures of BPP enabled content request and content messages, as well as the detailed procedures taken by the network nodes to process those messages. The performance analysis shows that the latency experienced by the consumer is reduced tremendously compared to the traditional approach, with the extra overhead to be reasonable small in different settings and scenarios.**

Keywords*— *Big IP Protocol*; *in-network intelligence*; *in-network programmability*; *semantics aware*; *content request*; *deviation aware.

I. Introduction

In the current Internet, content request involves many steps before the content reaches the consumer, i.e. the consumer finds out the URI of the content as well as the IP address of the content server, the consumer initiates a content request to the content server (could be a surrogate server deployed by Content Delivery Network Provider) with TCP/UDP transport layer protocol being used, in the last the content server returns the content back to the consumer. The first step may bring many obstacles to the consumer, since the consumer needs to discover the URI of the content satisfying the semantics requirements, as well as to resolve the location of the content where it is hosted.

The semantics information [1] summarizes basic properties about data, which can help identifying the data, make discovering and working with particular instances of data easier. For example, a user wants to get the traffic video at 8:00AM 01/06/2019 on Route 15 exit 1 in San Diego, with the semantics requirements in mind, which include that the type of the content is traffic video, the location is Route 15 exit 1 in San Diego and the time is 8:00AM 01/06/2019. The consumer initially does not have any idea of where to get the content, which requires the first step to be carried out. A consumer may be able to tolerate when he/she does not get the exact data being requested, given the cost may be lower as pointed out in [3]. For example, the consumer can accept a temperature data sensed 1 hour ago, instead of the current up-to-date data, given he/she would pay less for the service because a cached copy from an in-network router may be returned. As another example, the consumer can accept a traffic video data that does not exactly match the location semantics requirement, e.g. instead of Route 15 exit 1, a traffic video with location semantics information of Route 15 exit 2 is also acceptable.

With the steps illustrated at the beginning of the paper, the content request always goes to the content server without considering the opportunity that there might be a cached matching content located nearer to the consumer. Thus the current Internet architecture was not designed to make the content request with semantics requirement very efficient.

In the last decade, the Information-Centric Networking (ICN) has emerged and been extensively researched to design the Internet to center around information retrieval. One of the most outstanding architectures is Named Data Networking (NDN) [4][5]. The NDN architecture has its success in facilitating the content request by changing the IP-based routing foundation to content name based routing. However, it is yet another overlay solution with a great deal of overhead in the routing table construction, pending request recording and matching. The state maintenance is in the scale of content number instead of network devices.

Recently, Big IP Protocol (BPP) [6] has been proposed as an evolutional extension to the current IP packet to bring the intelligence of user experience, continuity and awareness of services into the network. The BPP framework brings minimum changes to the current Internet protocols. The authors proposed the idea of creating a block of information about the IP packet. This information is carried and processed en-route from the source to the destination. Fig. 1 shows the unified framework of the future IP packet, which could enable user-oriented, context-aware, intelligent services provided by the Internet. The "Commands" could describe how the routers treat the packet as it traverses the network. The "Metadata" contains data about the packet, e.g. geo-coordinates, classification tags, identity metadata, accounting information, as well as the contextual information about the user, the application, etc. And it can also allow IP packet to maintain customized statistics about the flow on intermediate hops. The in-network node intelligence is naturally embedded and supported by the BPP framework. The current Internet's inefficiency in content request that was described earlier can be well addressed with the in-network intelligence and programmability enabled by the BPP framework. In this

978-1-5386-1478-5/18/$31.00 ©2018 IEEE

paper, we intend to provide the solutions to support the semantics aware content request with much less latency experienced by the consumer and reasonable overhead that may be brought into the network.

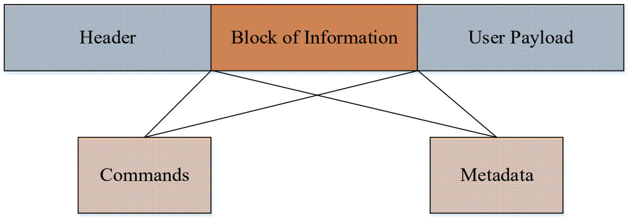


Fig. 1. Unified framework for future IP packet

The rest of the paper is arranged as followings: Section II proposes the design on BPP content request and content messages, as well as the procedures for in-network nodes to process those new types of BPP packets. Section III gives a thorough analysis on the extra overhead and consumer's experienced latency. Section IV concludes the paper.

II. SEMANTICS AND DEVIATION AWARECOTNENT REQUEST

To make the content request more efficient and leverage the cached content in the network, it is proposed that a consumer' request can have the form as shown in TABLE I.

TABLE I. CONSUMER'S REQUIREMENT ON CONTENT REQUEST

Number of Metadata Information to be Matched
Type
Keyword (*deviation degree)*
Location (*deviation degree*)
TimeStamp (*deviation degree*)
SizeLimit (*deviation degree*)
Other semantics information ...

In the Metadata block of a BPP content request message, a consumer is allowed to specify the semantics information of the desired content to be discovered. Thus the *Number of Semantics Information to be Matched* field is to indicate how many semantics information needs to be satisfied by the returned content.

The *Type*, *Location*, *TimeStamp*, *Keyword*, *SizeLimit* fields are the types of semantics information that are considered to commonly exist in consumers' requests [2]. A consumer's request such as traffic video at 8:00AM 01/06/2019 on Route 15 Exit 1 in San Diego can be formulated with three sets of semantics information to be matched, i.e. the *Type* field is set to traffic video, the *Location* field is set to Route 15 Exit 1 in San Diego, the *TimeStamp* field is set to 8:00AM 01/06/2019.

- The *Type* field can be used to specify the type of the content with defined and acknowledged urgency, e.g. traffic video, temperature, vehicle speed, etc. We assume that from the *Type* field, the network node is able to quantify the urgency of the requested data. For example, the vehicle speed data may have higher priority over the temperature data when the packets forwarding need to be scheduled at the network nodes.
- The *Location* field is used to designate the location semantics information where the data was generated by the physical device. The location could be a physical address, geolocation, latitude/longitude, a thing that the physical device is attached to (e.g. road, person, window, light, etc.).
- The *TimeStamp* field is used to designate the time or time period when the data was stamped/generated. If the *TimeStamp* field is not specified, then the most up-to-date content needs to be returned.
- The *Keyword* field is used to specify any keywords that need to be matched by the content.
- The *SizeLimit* field is used to designate the lower bound of the data size that the consumer requires. For example, the consumer may require that the traffic video size cannot be smaller than 100M to ensure the video resolution. The *SizeLimit* field is optional. The request message may also include any other semantics information that needs to be matched based on the consumer's demands, which can be followed after the *SizeLimit* field.

The *deviation degree* has been proposed for the *Keyword*, *Location*, *TimeStamp*, and *SizeLimit* semantics information. It means that the consumer is willing to tolerate some level of inaccuracy/deviation if he/she can be charged less compared to the cost for retrieving the exactly matching data. We assume a charging model, in which a consumer is charged by the number of routers forwarding the request message until a matching content is returned. As a result, it is very likely that a consumer could be charged less if he/she can tolerate some level of deviation from the desired *Keyword*, *Location*, *TimeStamp* and *SizeLimit.*

The *deviation degree* of *Keyword* could be defined as Levenshtein distance [7] of two sequences. The Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other. Another example to define the deviation degree of keyword is to use Jaro–Winkler similarity, which calculates the similarity between two strings. The Jaro–Winkler similarity [8][9] uses a prefix scale which gives more favorable ratings to strings that match from the beginning for a set prefix length. Given two strings s_1 and s_2, it is defined as:

$$JW(s_1,s_2) = sim_{Jaro}(s_1,s_2) + \quad (1)$$
$$l * p * (2 - sim_{Jaro}(s_1,s_2))$$

where, $sim_{Jaro}(s_1,s_2)$ is the Jaro similarity for two strings s_1 and s_2. l is the length of common prefix at the start of the string up to a maximum of four characters. p is a constant scaling factor for how much the score is adjusted upwards for having common prefixes. p should not exceed 0.25, otherwise the distance can become larger than 1. The standard value for this constant in Winkler's work is $p = 0.1$.

The Jaro Similarity of two strings s_1 and s_2 is defined as

$$sim_{Jaro}(s_1, s_2) = \begin{cases} 0, & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right), & \text{otherwise} \end{cases} \tag{2}$$

where: $|s_1|$ and $|s_2|$ is the length of the string s_1 and s_2 respectively; m is the number of matching characters; t is half the number of transpositions. Each character of s_1 is compared with all its matching characters in s_2. The number of matching (but different sequence order) characters divided by 2 defines the number of transpositions.

The *deviation degree* of *Location* could the physical distance between the required location and the location where the actually returned data was generated, which can have various units/formats (miles, number of exits, etc).

The *deviation degree* of *TimeStamp* could be the difference between the required time stamp and the time stamp of the actually returned data, which can be represented in seconds, minutes, hours, and days.

The *deviation degree* of *SizeLimit* could be the difference between the required size and the size of the actually returned data, which can be represented in bytes.

A matching data must satisfy (we consider the common semantics information, i.e. *Type*, *Location*, *TimeStamp* and *Keyword*):

(1) *Type* is the same;

(2) The distance between the *Location* in the request message (denoted as $loc_{desired}$) and the location semantics information of the matched content (denoted as $loc_{matched}$) is shorter than the *deviation degree* of the *Location* field (denoted as dd_{loc}), i.e. $diff_{loc} = distance(loc_{desired}, loc_{matched}) < dd_{loc}$;

(3) If the *TimeStamp* field exists in the request message, the absolute value of the difference between the *TimeStamp* in the request message (denoted as $t_{desired}$) and the time matadata information of the matched content (denoted as $t_{matched}$) is smaller than the *deviation degree* of the *TimeStamp* field (denoted as dd_{time}), i.e. $diff_{time} = |t_{desired} - t_{matched}| < dd_{time}$.

(4) If the *Keyword* field exists in the request message, the Jaro–Winkler similarity of the specified keyword in the request (denoted as $keyword_{desired}$) and the keyword of the matched content (denoted as $keyword_{matched}$) is smaller than the *deviation degree* of the *keyword* field (denoted as $dd_{keyword}$), i.e. $JW(keyword_{desired}, keyword_{matched}) = JW_{keyword} < dd_{keyword}$;

A. BPP Content Request Message Design and Processing Procedure

In the Metadata block of the BPP content request message, the proposed consumer's requirement to discover a matched content can be added. The source address in the BPP content request message is set to be the consumer's IP address, while the destination address is set to be the content server, which is the original producer or host of the content.

In the Command block of the BPP content request message, it is proposed to add the following commands to be executed by the BPP enabled network nodes:

1. Accept a cached content and stop forwarding the message to destination if the conditions (1) to (4) satisfy.
2. If the first action is not taken, forward the content request to a nearby node within certain distance (e.g. direct neighbor), if the node notifies the semantics information of a locally stored content to the current forwarding router, which satisfies the conditions (1) to (4).
3. Keep forwarding the message towards the destination IP address if the first two actions are not taken.
4. Record the content request in order to collect statistics, such as the popularity or importance of a content, which is out of scope of this paper.

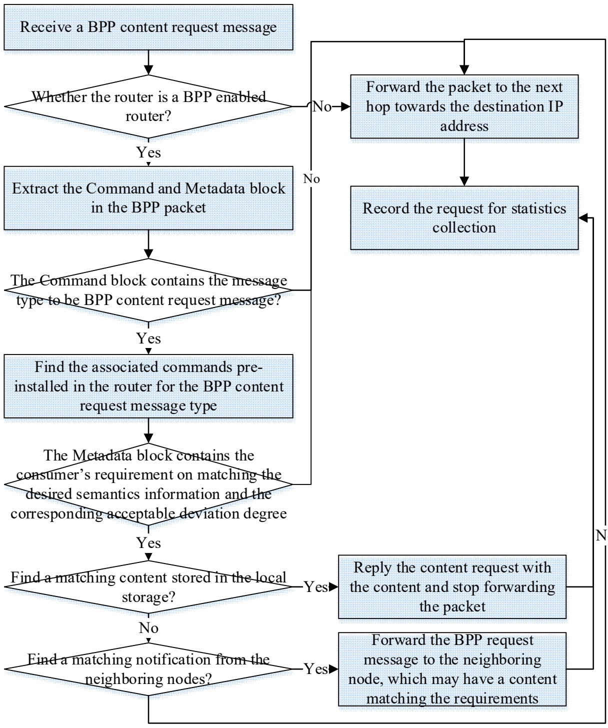

Receive a BPP content request message
Whether the router is a BPP enabled router?
No
Forward the packet to the next hop towards the destination IP address
Yes
Extract the Command and Metadata block in the BPP packet
No
Record the request for statistics collection
The Command block contains the message type to be BPP content request message?
Yes
Find the associated commands pre-installed in the router for the BPP content request message type
The Metadata block contains the consumer's requirement on matching the desired semantics information and the corresponding acceptable deviation degree
Yes
No
Find a matching content stored in the local storage?
Yes
Reply the content request with the content and stop forwarding the packet
No
Find a matching notification from the neighboring nodes?
Yes
Forward the BPP request message to the neighboring node, which may have a content matching the requirements

Fig. 2. Message flow of processing a BPP content request message

Since the commands for this type of content requests can be considered consistent and need to be executed by the en-route BPP enabled routers, it is not necessary to carry the commands each time a content request is launched by a consumer. It is suggested that the commands can be pre-defined and installed in the BPP enabled routers and are associated with the message type of the BPP content request. In the other words, the message type needs to be carried in the Command block to indicate it is a BPP content request message, then the BPP enabled routers are able to execute the corresponding commands.

When a traditional router receives the BPP content request message, it simply forwards the packet towards the destination and would not process any of the BPP blocks. The above procedure is illustrated in Fig. 2.

B. BPP Content Message Design and Processing Procedure

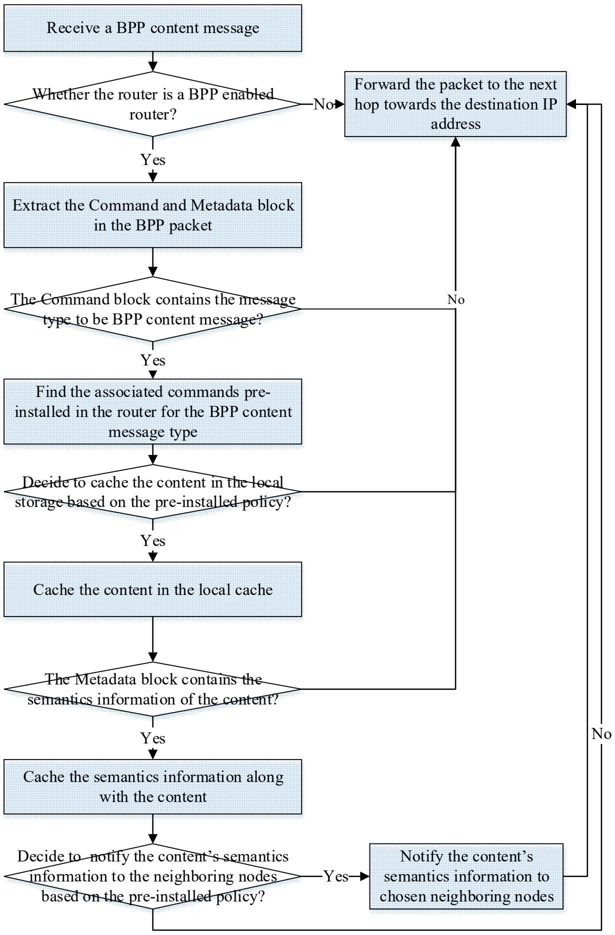
In the Metadata block of the BPP content message, the semantics information associated with this content can be added. The source address in the BPP content message is set to be the IP address of the node, which replies the request and returns the content, while the destination address is set to be the consumer's IP address.

In the Command block of the BPP content message, it is proposed to add the following commands to be executed by BPP enabled network nodes:

- Cache the content with certain policy designated by the source of the content. The policy could be based on the urgency or popularity of the content.
- Notify the semantics of the content to the neighboring nodes with certain policy designated by the source of the content. The policy may be based on the number of received requests on this particular content from the history record/statistics.
- On the other hand, the Command block of the BPP content message can only include the message type, such that each BPP enabled router on the path towards the consumer can execute the pre-installed commands independently according to its own polices on content caching and content semantics notification.

When a content is cached by the BPP enabled router in its local storage, the corresponding semantics information contained in the Metadata block of the message is also copied and cached.

When a semantics notification message is received by a BPP enabled router, the semantics information is recorded with the source of the notification message. In other words, it is indicated that the content with the semantics information can be found in the neighboring node where the notification message comes from. The above procedure is illustrated in Fig. 3.



Receive a BPP content message
Whether the router is a BPP enabled router?
No
Forward the packet to the next hop towards the destination IP address
Yes
Extract the Command and Metadata block in the BPP packet
The Command block contains the message type to be BPP content message?
No
Yes
Find the associated commands pre-installed in the router for the BPP content message type
Decide to cache the content in the local storage based on the pre-installed policy?
Yes
Cache the content in the local cache
The Metadata block contains the semantics information of the content?
Yes
No
Cache the semantics information along with the content
Decide to notify the content's semantics information to the neighboring nodes based on the pre-installed policy?
Yes
Notify the content's semantics information to chosen neighboring nodes

Fig. 3. Message flow of processing a BPP content message

III. Performance Analysis

In this section, we analyze the expected latency reduction experienced by the consumer due to the proposed semantics and deviation aware content request scheme. We consider the scenario as shown in Fig. 4, the consumer makes semantics aware content request to the original content server. It is not necessary to assume all network nodes to be BPP enabled, the traditional routers only need to forward the packets without any in-network processing on the packets, which does not affect our analysis. In order to make the analysis easier to understand, we assume that the network nodes are all BPP enabled. Each intermediate router along the path between the consumer and the content server has averagely m number of direct neighbors, which do not interconnect with each other. In other words, the direct neighbors of an intermediate router are not direct neighbors of any other intermediate routers as shown in Fig. 4.

The parameters that are needed for the analysis are described in TABLE II.

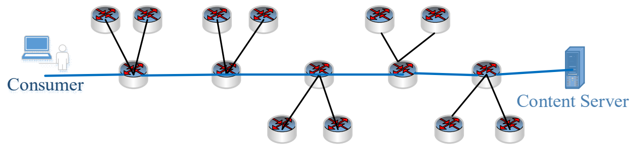


Fig. 4. Network scenario

TABLE II. PARAMETERS

Parameter	Description
m	The average number of neighboring routers of a BPP enabled router, other than the previous hop along the path from the consumer towards the original content server. In order to simply the analysis, we assume each router in the considered network is BPP enabled router. Thus each hop that the BPP request message traverses is able to process the packet.
ps	The probability that a BPP enabled router contains a content that satisfies the consumer's content request requirement with consideration of the acceptable deviation degree.
$p(n)$	The probability that the BPP content request message is replied by a matched content. n is the number of hops between the consumer and the router that replies the request message.
$D(n)$	The latency that is experienced by the consumer when the content request has been forwarded nth hop from the consumer and is replied by the node with a satisfying content. It includes two scenarios: (1) it is forwarded to the nth hop node along the path from the consumer towards the original content server; (2) it is forwarded to one of the direct neighbors of $(n-1)$th hop node along the path from the consumer towards the original content server.
tr	The average delay that is introduced when a BPP content request message is processed by one router.
tc	The average delay that is introduced when a BPP content message is forwarded by one router, which is fixed in the analysis without considering the content size effect on the delay, which is set to be much larger than tr.
$Overhead$	The extra number of bytes that are added to achieve the proposed semantics and deviation aware content request, which include the BPP block introduced each time a content request is sent and a content is returned, as well as the semantics information is notified to direct neighboring nodes each time a content is cached in a network node.
ln	The number of bytes that is needed to carry semantics information in the BPP content message or to notify the neighboring node.
$lbpp$	The number of bytes that is needed to carry semantics aware content request with deviation degree specified.
N	The total number of hops between the consumer and the original content server.
$Latency$	The average latency that would be experienced by consumer for the content request.

Equation (3) defines the probability that a BPP enabled router has a cached data satisfying that the *Location*, *TimeStamp* and *Keyword* are within the corresponding acceptable *deviation degree* specified by the consumer.

$$\begin{aligned} ps = p_{cached} * p(diff_{loc} < dd_{loc}) \\ * p(diff_{time} < dd_{time}) \\ * p(JW_{keyword} < dd_{keyword}) \end{aligned} \tag{3}$$

First we calculate the probability that the BPP content request message is replied by a matched content by a router, which is n number of hops away from the consumer. It falls into two scenarios: (1) It is replied by an intermediate router which is n number of hops away from the consumer. (2) It is replied by a neighboring router of the intermediate router, which is (n-1) number of hops away from the consumer. Both scenarios would generate the same latency that will be experienced by the consumer. The formulations to calculate each of the probabilities are shown in Equation (4)-(9). The last hop is the original content server, which hosts the content with 100% certainty.

$$p(1) = ps \tag{4}$$

$$p(2) = (1-ps) * (1-(1-ps)^m) \tag{5}$$

$$p(3) = (1-ps)^{m+1} * (1-(1-ps)^m) \tag{6}$$

$$p(4) = (1-ps)^{2m+1} * (1-(1-ps)^m) \tag{7}$$

$$\dots\dots\dots$$

$$p(n) = (1-ps)^{(n-2)*m+1} * (1-(1-ps)^m) \tag{8}$$

$$p(N) = 1 \tag{9}$$

The latency that is experienced by the consumer when the content request has been forwarded n th hop from the consumer and is replied by the node with a satisfying content is shown in Equation (10). As a result, the average latency that will be experienced by the consumer with the proposed semantics and deviation awarecontent request is shown in Equation (11).

$$D(n) = n * (tr + tc) \tag{10}$$

$$Latency = \frac{\sum_{n=1}^{N} D(n) * p(n)}{N} \tag{11}$$

The extra overhead is generated due to the BPP block introduced to the content request and content messages, and the semantics notification messages sent to direct neighboring nodes each time a content is cached, which is shown in Equation (12).

$$\begin{aligned} Overhead = \frac{\sum_{n=1}^{N} p(n) * n * (lbpp + \ln)}{N} \\ + ps * (m+1) * ln * N \end{aligned} \tag{12}$$

A. Content Matching Probability

Fig. 5 shows the comparison of the average latency that is experienced by the consumer when the content matching probability changes. It is noticeable that the semantics and deviation aware scheme introduces significantly smaller latency compared to the traditional content request approach as described in the beginning of the paper. In the traditional procedure, the cached copy of satisfying content cannot be used to reply the content request, due to the opaqueness of the content request message to the network nodes, while in the proposed scheme, the network nodes can leverage the in-network caching, programmability, processing capabilities to reply the content request faster and more efficiently. Fig. 6 shows a zoomed in view of the latency in the semantics and deviation aware scheme. The latency experienced by the consumer consistently decreases when the probability of finding a matched content in the network nodes increases. However, we also observe that when the matching probability changes from 0.1 to 1, the latency does not decrease very dramatically. It may be valid to allocate a reasonable size of storage to accommodate the cached content in order to achieve the similar amount of reduction in the latency performance.

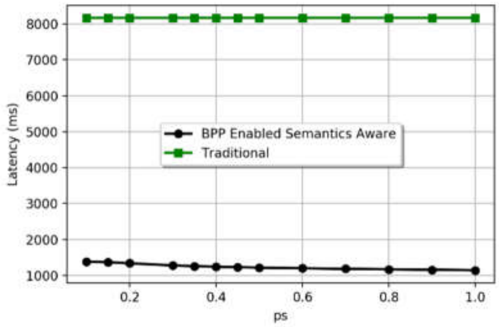


Fig. 5. Latency comparison vs. *ps*

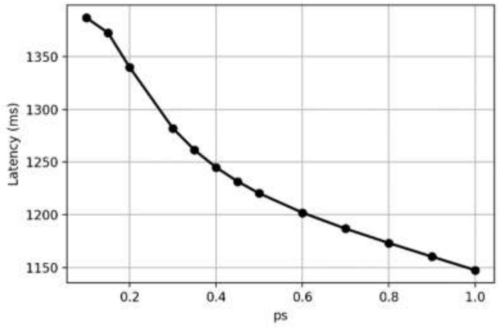


Fig. 6. Latency of proposed semantics and deviation aware scheme vs. *ps*

Fig. 7 shows the extra overhead that is introduced by the proposed scheme, which includes two parts, one is introduced by the BPP block extension in the content request and content messages, another part is introduced by the semantics notifications among neighboring network nodes. The first part introduced by the BPP block extension decreases when the matching probability increases as shown in Fig. 8. The reason is that with higher matching probability, the network nodes that are within short distance from the consumer are more likely to reply the content request message, thus the content also travels much smaller number of hop in the network before it reaches the consumer. On the hand, from Fig. 7 we can see that the extra overhead introduced by the BPP block extension is quite small compared to that introduced by the semantics notification. More number of content copies being cached in the network nodes, proportionally more semantics notifications need to be sent among neighboring nodes.

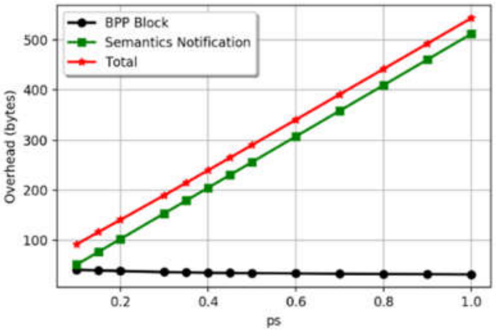


Fig. 7. Extra Overhead comparison vs. *ps*

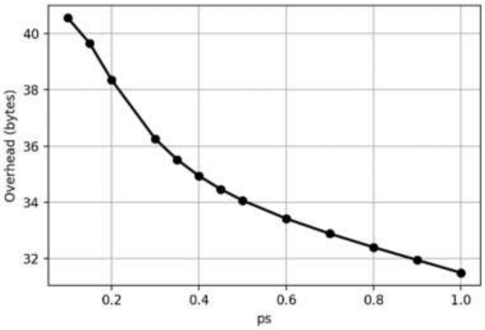


Fig. 8. Extra overhead introduced by BPP block vs. *ps*

B. Number of Adjacent Neighbors

When the number of adjacent neighbors varies, we again achieve the significant reduction in the latency experienced by the consumer in the proposed scheme, which is not shown again in this section due to the limit of the paper length. It is shown in Fig. 9 that the latency of the proposed scheme decreases when the number of adjacent neighbors increases, because it is more promising that the content request may be replied by one of the neighboring network nodes with a cached content. On the other hand, as shown in Fig. 10 the extra overhead introduced by the BPP block extension decreases due to the same reason.

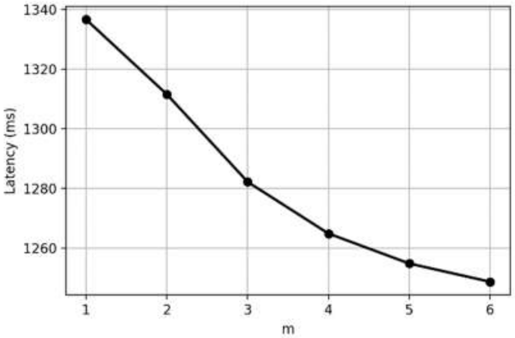


Fig. 9. Latency of proposed semantics and deviation aware scheme vs. *m*

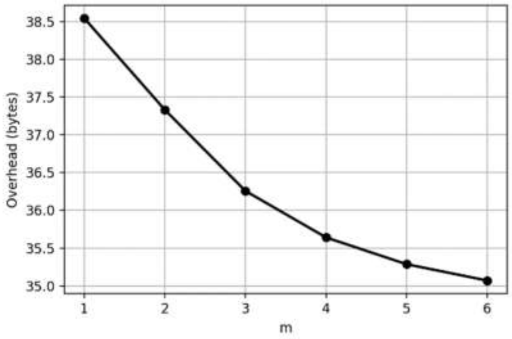


Fig. 10. Extra overhead introduced by BPP block vs. *m*

C. *Distance from the Consumer to the Content Server*

Fig. 11 shows the comparison of the average latency that is experienced by the consumer when the distance from the consumer to the content server varies. We observe that the latency performance improvement of the proposed scheme is very impressive when the distance is large.

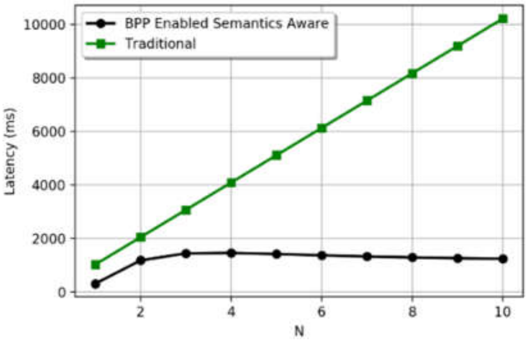


Fig. 11. Latency comparison vs. *N*

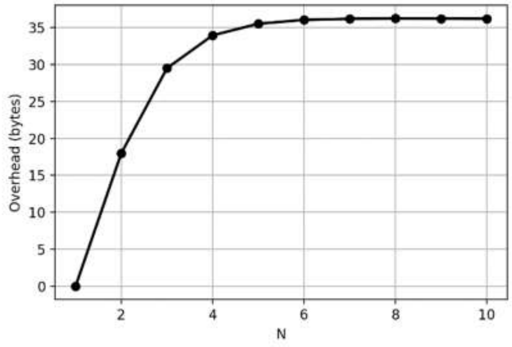


Fig. 12. Extra overhead introduced by BPP block vs. *N*

The extra overhead introduced by the BPP block extension increases along with the larger distance from the consumer to the content server. However, when the distance becomes even longer (i.e. larger than 6), the extra overhead stays constant, which further strengthens the proposed scheme in most common content request scenarios where the consumer is usually quite far away from the original content server (i.e. the overhead won't grow as the distance gets bigger).

IV. Conclusions

The paper presents a novel semantics aware content request based on a new evolutional and unified Internet framework, named Big IP Protocol (BPP). It considers the likelihood that a consumer may be able to accept certain degree of deviation on the semantics information of the returned content from the exact requirement. The BPP protocol provides the extensions to the current IP packet to carry the semantics requirement and the deviation degree in the content request message, such that the network nodes is enabled with the intelligence to do the semantics based matching on its locally cached content. When a content stored in an intermediate network node is matched to the semantics requirement within the deviation degree, the content can be used to reply the consumer's request, resulting in significant reduced latency. The paper analyzes the latency reduction that can be achieved by the proposed scheme compared to the traditional approach. On the other hand, the extra overhead that may be introduced by the proposed scheme is also quantified to show that it is negligible relative to the improvement on the latency.

References

[1] A. Sheth, "Data Semantics: what, where and how?" 6th IFIP Working Conference on Data Semantics, Atlanta, GA.

[2] P. Barnaghi, W. Wang, L. Dong, C. Wang, "A linked-data model for semantic sensor streams," GreenCom 2013.

[3] L. Dong, G. Wang, "Consumer Oriented IoT Data Discovery and Retrieval in Information Centric Networks," IEEE PIMRC, 2017.

[4] NDN, the Named Data Networking project, http://www.named-data.net/.

[5] V. Jacobson et al., "Networking named content," ACM CoNEXT Conference, 2009.

[6] R. Li, A. Clemm, U. Chunduri, L. Dong, K. Makhijani, "A New Framework and Protocol for Future Networking Applications," ACM Sigcomm NEAT workshop, 2018.

[7] F. Papadimitriou, "Modelling spatial landscape complexity using the Levenshtein algorithm," Ecological Informatics 4: 48–55, 2009.

[8] M. A. Jaro, "Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida," Journal of the American Statistical Association 84:414-420, 1989.

[9] M. A. Jaro, "Probabilistic linkage of large public health data files, " Statistics in Medicine 14: 491-498, 1995.