

Interest Reduction and PIT Minimization in Content Centric Networks

Aadil Zia Khan

Lahore University of Management Sciences, Lahore, Pakistan

Abstract—Content Centric Networking aspires to a more efficient use of the Internet through in-path caching, multi-homing, and provisions for state maintenance and intelligent forwarding at the CCN routers. However, these benefits of CCN's communication model come at the cost of large Pending Interest Table (PIT) sizes and Interest traffic overhead. Reducing PIT size is essential since larger memory sizes have an associated cost of slower access speeds, which would become a bottleneck in high speed networks. Similarly, Interest traffic may lead to upload capacity getting filled up which would be inefficient as well as problematic in case of traffics having bidirectional data transfers such as video conferencing. Our contribution in this paper is threefold. Firstly, we reduce PIT size by eliminating the need for maintaining PIT entries at all routers. We include the return path in the packets and maintain PIT entries at the egress routers only. Further, we use Persistent Interests (PIs), where one Interest suffices for retrieving multiple data segments, in order to reduce PIT entries at the egress routers as well as to reduce Interest overhead. This is especially useful for live and interactive traffic types where packet sizes are small leading to a large number of pipelined Interests at any given time. Lastly, since using PIs affects CCN's original transport model, we address the affected aspects, namely congestion and flow control and multi path content retrieval. For our congestion scheme, we show that it achieves max-min fairness.

Keywords—Content centric networks; congestion control; scalability

I. INTRODUCTION

Content Centric Networking aims for a more efficient use of the Internet. It follows a pull based communication model - the content is split into multiple segments and the subscriber sends out a separate Interest packet for each segment. Unlike the current Internet, routing is based on content names as opposed to end hosts' IP addresses. The approach followed by Content Centric Networks (CCNs) has many benefits - in-path caching, multi-homing, and provisions for state maintenance and intelligent forwarding at the CCN routers, to name a few. However, CCN does have a problem of a large memory footprint and overhead traffic that needs to be addressed to get the maximum benefit from CCN's pull based communication.

A. Memory Footprint of the PIT

PIT as a core module of the CCN architecture allows many benefits in particular, such as multicasting, forward and reverse path symmetry, and filtering of unsolicited and duplicate packets. However, CCN derives these benefits at the cost of high memory requirements. To put this statement in context and see how much of a problem large memory sizes can be, let us first look at the different memory types studied for CCNs.

Off chip SRAMs can store up to 210Mb of data and have an access time of 4ns. SRAMs have been shown to not follow Moore's Law. RDRAMs, on the other hand, can store up to 2Gb of data but have an access time of 15ns. DRAM has an access time of 55ns, but it can store up to 10GB [1]. Assuming each PIT entry size is 100 bytes; SRAMs can only contain around 262.5K entries, whereas RDRAMs can have 2.5M entries. Can 100 bytes for a PIT entry be considered a realistic figure? In [2], for two different data sets, authors give an average URL length of 18 and 53 characters. Note that in CCN, the lengths may be longer if you also include AS names in the content name for ease of routing. Also, in the future, if different languages are to be incorporated in the content names, Unicode encoding would be required instead of ASCII so in terms of bytes the length would double.

Let's examine the above figures in the context of the PIT. The PIT would have a large memory footprint. In [2] authors observe from trace of a 20Gbps access link that the corresponding PIT had 1.5M entries. This figure is expected to go up as bandwidths and the number of links increase. The number of PIT entries in a router is roughly on the order of the following:

$$(N_{\text{port}} \times \text{BW} \times \text{RTT}) / \text{Size}_{\text{packet}} \quad (1)$$

N_{port} is the number of ports that the switch has, BW is the bandwidth, RTT is the total round trip propagation delay, and $\text{Size}_{\text{packet}}$ is the size of data packets.

The contribution of each flow to the PIT size may be large in case of pipelined interests for live and interactive traffic where packet sizes are small. For example, in VoCCN, each client sends out around 50 interests per second [3]. Unlike live traffic, for traffic types where chunk sizes can be large, such as file sharing, the number of Interests can be kept small if we keep large segment sizes and allow packet fragmentation [4]. However large packet sizes have performance issues in face of packet loss. So we believe that the number of pipelined interests could be large in non-live traffic too.

Fitting PIT in the SRAM would be difficult and slow memory access makes DRAM or RDRAM infeasible for CCN routers. Note that multiple memory accesses would be required for a PIT entry lookup. Another benefit of reducing PIT and FIB sizes would be a reduction in overall costs and energy requirements since fast access memories are expensive (\$27/MB for SRAM as compared to \$0.27/MB for RDRAM) and consume more power (0.12W/MB for SRAM as compared to 0.027W/MB for RDRAM) [1].

B. Overhead of the Interest Traffic

The one Interest packet per Data packet communication model adopted by CCN causes other overhead issues as well. Firstly, frequent insertions and deletions increase CPU load and lead to a decrease in efficiency, causing increase in download times [5]. Secondly, high upload bandwidth usage is an outcome if packet sizes are small and bitrates are high. For 9Mbps videos with packet size of 1500bytes, the number of Interests generated per second would be 750, requiring 0.6Mbps upload capacity assuming 100 byte Interest packet. For video conferencing, assuming 1Mbps streaming rate and average 500 bytes packets [6], there will be around 250 packets per second, requiring 0.2Mbps upload capacity. If we consider an IPTV solution, it is possible that multiple channels are retrieved simultaneously for smooth channel transitioning [7]. In this case the upload rate would increase by a factor equal to the number of channels simultaneously pre-fetched. Constraining upload capacity usage would be especially important in cases where upload is used for data transmission as is the case in video conferencing or in cases where links have asynchronous upload and download capacities. Also, lost Interests due to a congested upstream link would affect the stream quality even if the downstream link had sufficient capacity available.

We believe it is essential to revisit the original architecture with these issues in mind so as to ensure that the architecture does not become a bottleneck in the future. In this regard, we reduce PIT size, upload bandwidth usage, and insertion/deletion operations by using PIs where necessary, and completely eliminating PIT entries where possible. To this end, we also address congestion and flow control and multipath forwarding issues that arise due to a departure from CCN's original pull based approach to a push based one.

Keeping in mind increasing bandwidths and an ever increasing share of long flow streaming traffic, we believe the benefits of this work would remain even in the long term.

Rest of the paper is organized as follows. First of all, we examine the related work in section 2. Our approach for PIT size reduction by including the return path inside the packets is proposed in section 3. Our PI based communication model, together with max-min fair congestion control is described in section 4. In section 5, we give a summary of our proposed changes to the CCN. Evaluation results are given in section 6. We conclude and describe our future direction in section 7.

II. RELATED WORK

PIT size reduction is recognized as an important goal in CCN research. State of the art focuses on the data structure and hardware aspect of PIT. We believe that an approach which can reduce or eliminate the dependence on PIT without losing the benefits PIT brings is not only ideal but achievable as well.

Recent works try to reduce PIT sizes by using counting bloom filters [8], mapping bloom filters [9], and encoded named prefix tree [2]. Distributing PIT over multiple cards within a router has also been explored [8]. Using fingerprints, instead of full names, at the core routers, has been proposed in [10] to reduce the PIT sizes. Our approach can be used together

with all of these approaches to further reduce the memory requirements.

Authors in [11] explore Interest aggregation by using Interest Sets. Unlike them, we use PIs to reduce PIT size as well as Interest traffic. The problem with Interest aggregation is that a single Interest loss would affect several data packets. Also, in case of traffics having a large number of pipelined Interests, for instance HiDef IPTV, we would have large Interest Sets which would introduce great complexity at the router. This complexity would be because a large Interest Set would lead to a traffic spike. Also if an Interest Set has several Interests, the router would have to check each corresponding data packet in its Content Store which would require several memory lookups. Multipath forwarding would also be an issue since an Interest Set can only be forwarded to one source. Unlike their work, we also address multipath forwarding and congestion and flow control issues arising due to doing away with the one interest per data packet model. Similarly, authors in [12] also use Interest aggregation, however at the router, these Interests are internally translated into multiple PIT entries thus the PIT size is not reduced and only the Interest traffic is reduced.

A different approach to reducing PIT sizes is taken in [13], where RTT awareness at the routers is used to reduce the residence time of PIT entries that have not been responded to.

In our congestion control scheme, the routers inform of the paths bottleneck fair share bandwidth. This approach has been studied in the context of Internet in works such as RCP[14] and XCP[15] where the returning data packet carries the bottleneck information. This bottleneck information is echoed back to the sender so that transmission rates may be adjusted accordingly. Compared to the approaches studied in the context of the current Internet, our approach would react quickly to changing network conditions for two reasons. Firstly, since CCN's breadcrumb approach eliminates forward and reverse path asymmetry, instead of echoing the rate information in Data packets, the Interest packets can be used to convey the bottleneck rate. Secondly, as we show in section IV, in CCN we can keep count of the exact number of flows as well as the flows bottlenecked on that router, thus enabling sharing of residual capacity amongst the bottlenecked flows only.

PIs have been proposed before [16]. Unlike our work, they do not address issues, such as congestion and flow control and multipath forwarding, arising due to a push based approach.

III. PIT REDUCTION USING STATEFUL PACKETS

PIT performs three functions. Firstly, it eliminates forward and reverse path asymmetry. Secondly, it allows multicasting. When an Interest is received at a router but there is already a corresponding PIT entry, this implies a multicast scenario. The router will drop the Interest and update the PIT entry. When the corresponding data packet is received, copies will be made and forwarded on all the faces at which the Interests were received. Thirdly, PIT prevents unnecessary forwarding. When an unsolicited data packet is received which does not have a PIT entry, it would be dropped. Likewise if a duplicate Interest is received (having the same nonce) on a different face, it would be ignored and no further action would be taken. Our

approach allows these benefits to be achieved even if we eliminate the PIT entries at the non-egress routers. We now describe our approach followed by a discussion of how the above mentioned benefits can still be achieved.

A. Interest Packet Transmission

We propose specifying the return path in the Interest packet's header using a PATH tag. When an Interest is received at an ingress router, instead of making an entry in the PIT, the ID of the node as well as the face at which the Interest was received is entered in the label header. The packet is then forwarded towards the egress router. Instead of making an entry in the PIT, each intermediate node appends only the ID of the face at which the Interest was received to the PATH tag.

When the Interest reaches the egress node, one option would be to append the face id to the PATH tag and forward the packet without making any entry in the PIT. While this would lead to complete elimination of the PIT, thus saving memory, it would also lead to the loss of benefits mentioned above that PIT brought. Thus we propose maintaining a PIT entry only at the egress routers. This PIT entry would also include the complete reverse path from the egress to the ingress routers within that AS. PATH tag would then be cleared before forwarding the packet into the next Autonomous System (AS).

In case multiple Interests, having the same ingress node ID in the PATH tag, are received at an egress router, the PIT entries would be combined to allow multicasting. To do this, the PATH tag saved in the PIT entry would be updated so that last entry corresponding to the ingress ID and the face includes all the relevant face IDs.

B. Data Packet Transmission

When a Data packet is received at the egress router, the PIT would be consulted and the reverse path inserted in the packet's PATH tag. Based on this field, each downstream router would determine the face at which the packet needs to be forwarded. In this manner, the packet would either reach the egress router of the downstream AS or the content requester.

C. Discussion

As shown in Fig.1, suppose an Interest packet is received at node R1 face 1. It would be forwarded to the egress node R4 as per our approach. When the Interest reaches the egress node, the PATH tag shows for each intermediate node the faces at which this Interest was received. The PIT entry would include this return path. Suppose another Interest is received at R1 face 2. This Interest would also be forwarded towards R4. However, from the PIT entry, R4 would infer that this is the same Interest but for a different face. So the entry would be updated. In the figure 1, <R1:1,2> implies that the returning Data packet has to be forwarded to node R1 faces 1 and 2.

The above approach ensures that the benefits of PIT are not lost as follows. When a Data packet is received at the egress router, the PIT would be consulted and incase it is unsolicited, it would be dropped. Otherwise, the packet would follow the reverse path specified in the PIT.

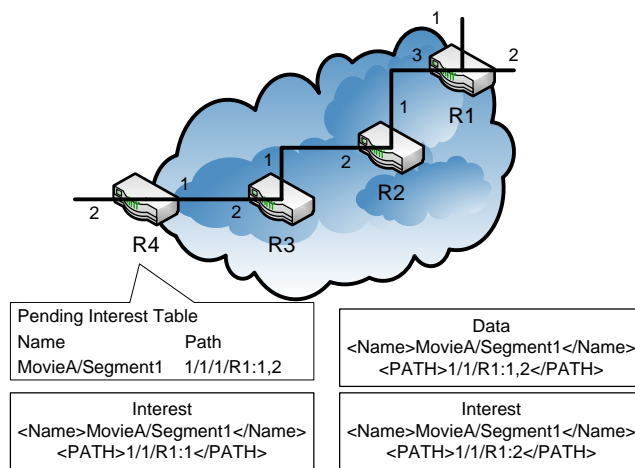


Fig. 1. PIT and Packets Structure.

For multicasting, note that if an Interest packet is received at the egress router and there is already a corresponding entry (or it is a duplicate Interest as determined by the nonce), the Interest would be dropped and the PIT entry updated. This would ensure multicasting at the gateway link. In case multiple Interests were received at an ingress router, to prevent duplicate data packets flowing from the egress to the same ingress node, the egress node would infer from its PIT entry if an Interest had been received at multiple faces of the same ingress router. This information would then be conveyed to the ingress router using the packet's PATH tag as already explained. This approach would reduce link stress [19] at the egress link and also on the path from the egress to the ingress router. However, link stress would not be eliminated in a scenario where two copies of the same data flow from the egress gateway to two different ingress gateways having some overlap in their paths. This is a tradeoff. We believe that minimizing memory requirements of CCN routers is important and link stress reduction at the gateway links is sufficient because link stress in the intra-AS paths would not be so problematic because load balancing approaches will be in place.

IV. INTEREST REDUCTION USING PERSISTENT INTERESTS

The approach discussed in the previous section reduces PIT entries at the non-egress nodes. The proposed approach would work well for static content, such as file sharing, where the data chunks already exist, can be large, and each chunk can be requested as required and from whatever source desired. Note that packet fragmentation [4] would allow Data packet sizes larger than the link MTU.

For live and interactive traffic, the Data packets have to be small leading to many pipelined Interests. VoCCN forwards 50 interests per second - the rate going higher if we also include video data. If we could replace these pipelined Interests with one Interest, we would save upload bandwidth as well as reduce the number of entries in the egress router's PIT. Also, decrease in the number of read write operations on the PIT would reduce processor load improving performance especially for video traffic. For this purpose, we adopt the concept of PIs to aggregate the Interests belonging to the same flow.

A. Session Management

To initiate content retrieval, the end user would send out a PI registration packet, which would also include the rate at which the application seeks to receive data. The receiving router would make a corresponding entry in the PIT and forward the PI packet to the next upstream router towards the content store. Once the transmission has been completed, a tear down message would be transmitted by the end user, indicating session closure.

During the session, the user would periodically send PI keep-alive messages to keep the session alive as well as to inform the content source of the allowed transmission for congestion and flow control. In case a deadline for a PI packet is missed, or a deregistering message is received, the node would remove the corresponding PIT entry.

B. Congestion and Flow Control

Similar to approaches like XCP and RCP, for congestion control, we propose an approach where each router maintains fair share bandwidth and the sender is informed of the bottleneck's fair share.

The content subscriber sends out a PI packet including the allowed bitrate specified in CURBOTRATE tag in the Interest packet. The receiving router would compare its own bottleneck fair share with value specified in the packet's CURBOTRATE tag, and update the latter if it is less before forwarding. Each router calculates the fair share bandwidth (FS) for each of its links using the following equation:

$$FS_t = (\text{TotalCap}/N) + (\text{ResidualCap}_{t-1}/N) \quad (2)$$

In the above equation, TotalCap refers to the link capacity, while ResidualCap_{t-1} refers to the total link capacity minus the capacity used in the previous measurement cycle. N is the number of flows passing through that link. N is basically the number of PIT entries of type PI for that link.

When the Content Source receives the Interest, it would forward the data stream at CURBOTRATE.

We observe in our experiments that the convergence to min max fair share for each flow would be slow if we use the above equation for sharing residual capacity among flows. For this reason we propose an improved residual capacity sharing approach based on the observation that state maintenance inherent in the CCN's architecture [18] can enable us to share the residual capacity among bottlenecked flows only instead of sharing among all flows.

For this purpose, we introduce a PREVBOTRATE tag in the Interest packet which would include the path bottleneck for the previous cycle, and the router would also maintain a count of bottlenecked flows (BOTFLOWCOUNT) for each link. The PIT entry would include a flag variable specifying if that flow is bottlenecked on that router or not. Each time a Persistent Interest is sent, the sender would reflect the CURBOTRATE value so that it becomes the PREVBOTRATE value in the next PI keep alive packet. Now when a PI packet is received, the router would check if the value in the PREVBOTRATE tag is greater than or equal to the FS. If it is, it would imply that the flow is being bottlenecked on that link. PIT entry would be

checked and if the bottlenecked flow flag is not set, the flag would be set and BOTFLOWCOUNT would be incremented. If the flag is already set, it implies that the flow has already been counted in the BOTFLOWCOUNT so no action would be taken. As opposed to this if a PI packet is received and PREVBOTRATE tag is less than the FS, it would mean that the flow is bottlenecked upstream. In this case if the PIT entry's flag is set, it would be set to false and the BOTFLOWCOUNT value would be decremented. If the flag is not set, no further action would be taken. The packet would then be forwarded upstream after modifying the CURBOTRATE tag if needed as per the approach described above. In this approach the equation for calculating the FS would be as follows.

$$FS_t = (\text{TotalCap}/N_{\text{all}}) + (\text{ResidualCap}_{t-1}/N_{\text{bot}}) \quad (3)$$

In the above equation, N_{all} is the number of flows passing through that link while N_{bot} is the number of flows bottlenecked on that link.

This approach requires PIT entries in all routers. We believe this is acceptable because firstly, multiple entries belonging to the same flow would be replaced by a single entry, and secondly this would allow multicasting on all nodes which would be especially important for reducing link stress in live and interactive traffic types. It should be noted that PI would only be used in live/interactive streaming types where the either the encoder can modify bitrate based on network feedback or the content has multiple copies encoded at different discrete rates and the corresponding rate can be sent based on the feedback. Thus for a multicast scenario, if multiple requests are received, based on different bottleneck rates, it is possible that different bitrates may be requested. Thus they have to be treated as different flows. For future work, we plan to address these issues and explore other issues arising when using this approach for multimedia streaming.

C. Multipath Download

PI approach would change the way multipath retrieval is done in CCNs. We handle this by splitting the session into multiple streams if needed and using regular expressions in the PIT and PI packets. For instance if the content can be obtained from 4 different faces, the segment numbers requested at face would be as the expression below and corresponding entries would be made in the PIT.

$$\text{segment}_{id} \% 4 == \text{face}_{id} \quad (4)$$

For example, if the face id is 2 and the total number of faces is 4, the above expression would basically be requesting only those segment numbers which when divided by 4 would return a remainder of 2 and these segment would only be requested from the content source reachable through face 2. Other regular expressions may also be used.

V. CHANGES TO CCN

Our approach eliminates the need for maintaining PIT at the non gateway routers for traffic types similar to file transfer. In this regard we need to modify the way Interest and Data packets are forwarded and PIT entries are made. For intra-AS transmission purposes, we include a PATH field in the Interest and Data packets to include the path to be followed from the

egress to the ingress node. PIT entries also need to be modified so that each entry also includes the paths that need to be followed by the returning packets.

For live and interactive traffic, our approach replaces multiple PIT entries belonging to the same flow with a single entry. We use PIs so that content stream may be retrieved using a single Interest. Since this would affect how congestion control is handled in CCN, we also propose a router specified bottleneck rate based rate control. For this purpose, we require each router to include local information regarding each link's fair share bandwidth, number of flows on that link, number of bottlenecked flows, and residual capacity available from the previous measurement cycle. We also require Interest packets to include fields for return path's current bottleneck fair share rate as well as the bottleneck fair share in the previous transmission cycle.

Lastly, multipath retrieval would be different if we use PIs. For this purpose, we propose splitting the stream and using regular expressions to match streams with the requests in PIT entries and Interest packets.

VI. EXPERIMENTAL EVALUATION

To make a case and see how much PIT size reduction can be achieved by using PIs, we took dataset from CAIDA¹ related to four 10Gbps links which measured the packets per second as well as the number of flows per second flowing through the links during a one hour period once every month. The former corresponds to the number of PIT entries using CCN's original approach while flows per second roughly translate to the number of PIT entries using our PI based approach. Dividing them gives us the reduction factor. For the readings provided, we get a maximum 6.56 fold PIT reduction with an average of 2.4 fold assuming a 100ms RTT. If we assume a 200ms RTT, the number increases to a maximum of 13.13 fold PIT reduction with an average of 4.89 fold reduction.

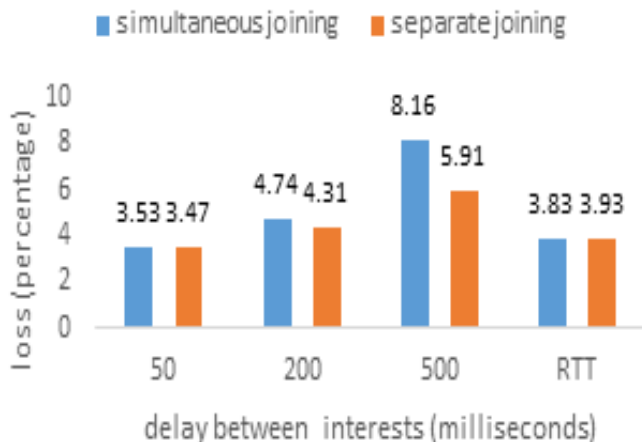


Fig. 2. PI Frequency vs Packet Loss.

For evaluation purposes, we implemented our PI approach in ndnSim[17]. First of all we studied the effects of frequency of PI keep alive messages on packet loss. We chose to measure packet loss because it would be representative of how quickly the sending rate would be updated in face of changing network conditions. We used a dumbbell topology consisting of four subscribers requesting content from four different sources sharing a bottleneck link having 1mbps capacity. All other links also had 1mbps capacity. Router queue limit was set to 10 packets. The RTTs for the four subscribers were 142ms, 62ms, 242ms, and 44ms respectively. The duration was 10 seconds and after 5 seconds, the bandwidth of node1's access link fell to 80kbps. For experimentation, we used two different scenarios. In the first scenario, each node joined at the same time, while in the second, the nodes joined at 0, 1, 2, and 3 seconds respectively. Results are shown in Fig. 2.

From the Fig. 2 we observe that the best results are obtained if the bottleneck probing PI packet is sent more than once per RTT. However, for our scenario, sending the probe once per RTT also gives comparable results. For less frequent probes, the sender would be slow to react, leading to a higher loss rate. Also, as shown in the figure, loss rate is lower for the scenario in which all the nodes join at different times. This is because in our scenario each node joined after a delay of one second after the previous nodes joining. This gave sufficient time for the network to converge to the fair share before the next node joins.

For the simultaneous joining scenario, we also studied the throughput each node got so as to measure fairness. For this purpose, we used a dumbbell topology where each link had 10Mbps capacity and RTT was set to 200ms for each node. Three nodes subscribed to a continuous stream whereas the fourth node followed an on-off transmission pattern where the on period varied from 500ms to 1.5s. The off periods also varied in a similar manner. The duration was 10 seconds. Fig. 3 shows the average throughput each node got. The slight difference is because of packet loss.

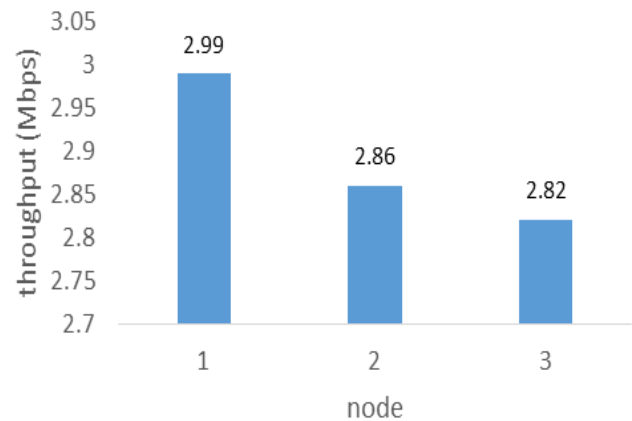


Fig. 3. Fairness.

¹ http://www.caida.org/data/passive/trace_stats/

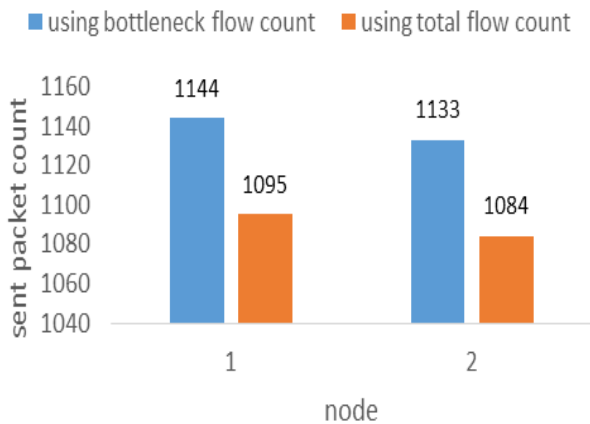


Fig. 4. Residual Capacity Sharing among Bottlenecked Flows.

Lastly, we explored the total number of packets transmitted in the scenario where the residual capacity was shared between all flows and compared it with the scenario where the residual capacity was shared only between the flows bottlenecked on that link. Sent packet count is used because it shows how quickly the sender converges to the fair share in case some flow's bottleneck shifts from the bottlenecked link to some upstream link. For this purpose we used the same dumbbell topology as above, however the the duration was 3 seconds. Also after one second, the bottleneck for node 3 and 4 shifts upstream and drops to 80kbps so that the original bottleneck link now has excess residual capacity which can be shared between flows belonging to nodes 1 and 2. As shown in the Fig. 4, more packets are transmitted if we distribute the residual capacity between the bottlenecked flows only.

VII. CONCLUSION AND FUTURE WORK

In this paper, we focused on PIT size reduction. For this purpose we proposed maintaining the return path inside the packets so as to eliminate the need for PIT in non egress routers. For traffic types where the number of pipelined Interests would be large, such as in multimedia streaming and video conferencing, we proposed using PIs where one Interests would be sufficient to retrieve the entire stream. We also proposed a congestion control approach to work with PIs. Our congestion control approach has been presented as a proof of concept in order to show the feasibility of the Persistent Interest approach.

For future work, we intend to focus on the congestion control aspect and experiment on larger topologies and in greater detail. We also plan to apply our fair share bottleneck probing approach for bit rate selection in rate adaptive video streams.

REFERENCES

- [1] D. Perino, and M. Varvello, "A reality check for content centric networking," in ACM Sigcomm Workshop on Information-Centric Networking (ICN) 2011.
- [2] H. Dai, B. Liu, Y. Chen, and Y. Wang, "On pending interest table in named data networking," In Proceedings of the eighth ACM/IEEE symposium on Architectures for networking and communications systems (ANCS) 2012.
- [3] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard, "Voccn: Voice over content centric networks," in Workshop on Re-architecting the Internet (ReArch) 2009.
- [4] A. Narayanan, N. Nathan, D. Oran, and G. Tsudik, "Secure Fragmentation for Content-Centric Networks," CCNxCon 2013.
- [5] M. Wahlisch, T. C. Schmidt, and M. Vahlenkamp, "Bulk of Interest: Performance Measurement of Content-Centric Routing," in Proc. of ACM SIGCOMM, Poster Session, 2012.
- [6] Implementing QoS Solutions for H.323 Video Conferencing over IP. <http://www.cisco.com/c/en/us/support/docs/quality-of-service-qos/qos-packet-marking/21662-video-qos.html>
- [7] D.A.G. Manzato, and N.L.S. da Fonseca, "A survey of channel switching schemes for IPTV," Communications Magazine, IEEE , vol.51, no.8, pp.120,127, August 2013.
- [8] W. You, B. Mathieu, P. Truong, J.-F. Peltier, and G. Simon, "DiPIT: a distributed Bloom-Filter based PIT table for CCN nodes," In ICCCN, 2012.
- [9] Z. Li, K. Liu, Y. Zhao and Y. Ma, "MaPIT: An Enhanced Pending Interest Table for NDN With Mapping Bloom Filter," in IEEE Communications Letters, vol. 18, no. 11, pp. 1915-1918, Nov. 2014.
- [10] H. Yuan and P. Crowley, "Scalable Pending Interest Table design: From principles to practice," IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, Toronto, ON, 2014, pp. 2049-2057.
- [11] X. Jiang, and J. Bi, "Interest Set Mechanism to Improve the Transport of Named Data Networking," in Proceedings of SIGCOMM, 2013.
- [12] D. Byun, B.-J. Lee, and M.-W. Jang, "Adaptive flow control via interest aggregation in CCN," in Proc. IEEE ICC 2013.
- [13] Y. Tan, Q. Li, Y. Jiang and S. Xia, "RAPIT: RTT-Aware Pending Interest Table for Content Centric Networking," 2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC), Nanjing, 2015, pp. 1-8.
- [14] N. Dukkupati, M. Kobayashi, R. Zhang-Shen, and N. McKeown, "Processor sharing ows in the Internet," Thirteenth International Workshop on Quality of Service, 2005.
- [15] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," Computer Communication Review, vol.32, no. 4, pp. 89-102, 2002.
- [16] C. Tsilopoulos and G. Xylomenos, "Supporting Diverse Traffic Types in Information Centric Networks," In ACM SIGCOMM Workshop on Information Centric Networking (ICN 2011), 2011.
- [17] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN, Technical Report NDN-0005, 2012.
- [18] A. Z. Khan, S. Baqai and F. R. Dogar, "QoS aware path selection in content centric networks," 2012 IEEE International Conference on Communications (ICC), Ottawa, ON, 2012, pp. 2645-2649.
- [19] A. Z. Khan and S. Baqai, "A Case for Reducing Link Stress in a Multimedia Streaming Service Backbone," The IEEE symposium on Computers and Communications, Riccione, 2010, pp. 945-947.