

CAINE: A Context-Aware Information-centric Network Ecosystem

George Kamel, Ning Wang, Vassilios Vassilakis, Zhili Sun, Pirabakaran Navaratnam, University of Surrey
 Chonggang Wang, Lijun Dong, InterDigital Communications LLC
 Rahim Tafazolli, University of Surrey

Abstract—Information-centric networking (ICN) is an emerging networking paradigm that places *content identifiers* rather than *host identifiers* at the core of the mechanisms and protocols used to deliver content to end-users. Such a paradigm allows routers enhanced with content-awareness to play a direct role in the routing and resolution of content requests from users, without any knowledge of the specific locations of hosted content. However, to facilitate good network traffic engineering and satisfactory user QoS, content routers need to exchange advanced network knowledge to assist them with their resolution decisions. In order to maintain the location-independency tenet of ICNs, such knowledge (known as *context information*) needs to be independent of the locations of servers. To this end, we propose CAINE — Context-Aware Information-centric Network Ecosystem — which enables context-based operations to be intrinsically supported by the underlying ICN routing and resolution functions. Our approach has been designed to maintain the location-independence philosophy of ICNs by associating context information directly to content rather than to the physical entities such as servers and network elements in the content ecosystem, while ensuring scalability. Through simulation, we show that based on such location-independent context information, CAINE is able to facilitate traffic engineering in the network, while not posing a significant control signalling burden on the network.

I. INTRODUCTION

Information-centric networking (ICN) is an emerging networking paradigm that places *content identifiers* rather than *host identifiers* at the core of the mechanisms and protocols used to deliver content to end-users. It essentially makes content *location independent*, thus inherently supporting features such as multicast and user mobility, and potentially enhancing content delivery performance for networks and end-users alike. For this reason, ICN is becoming increasingly appealing to network operators and manufacturers, who are investing in ICN research to bring it to the level of maturity needed for wide-scale commercial rollout. However, this is still a long way off, with many challenges yet to address satisfactorily such as content naming, security, and routing and resolution system scalability [1].

ICNs play a direct role in the routing and resolution of content requests from users, supporting fine-grained content access and distribution, with the ability to handle both complexity and uncertainty. This is done without reliance on any dedicated domain name system (DNS) like entity sitting ‘outside’ the network. To facilitate this role, content routers (CRs), i.e. routers enhanced with content awareness, need to possess advanced network knowledge — known as *context information* — such as content availability, content popularity,

content server load, and end-to-end path conditions for content delivery. This context information may then be used by CRs to make routing and resolution decisions that fulfil the traffic engineering requirements of the network and the quality-of-service (QoS) requirements of users.

Although existing ICN schemes support context information dissemination and traffic engineering to some degree, these are done by explicitly associating context information to physical elements in the ecosystem, but not intrinsically to the content itself. As such the current practice of using context information in ICN environments still fails to support location independence in terms of context awareness. For example, in the data-oriented network architecture (DONA) [2] and content-ubiquitous resolution and delivery infrastructure for next-generation services (CURLING) [3] approaches in which resolution is carried out by dedicated handlers, only server load information is captured; path load information is not, as it is technically challenging/unsuitable to do so given their centralized architectures. In the named-data networking (NDN) scheme [4], content routers periodically flood user requests for chunks of different contents towards all potential content sources. The routers will then learn the best interface to use for all near-future content requests based on the time taken to receive the requested chunks. However, such an approach allows only the least delay path to be gauged without really obtaining concrete information about the distance to the servers, their loads, and the bandwidth available along each potential path.

In this article we introduce CAINE — Context-Aware Information-centric Network Ecosystem — which enables advanced context-based operations to be intrinsically supported by the underlying ICN routing and resolution functions. Our approach has been designed to maintain the location-independent philosophy of ICNs by associating context information directly to content rather than to the explicit servers hosting it or indeed the network elements delivering it. Regardless of this, our proposed scheme ensures that location-independent context information is exchanged efficiently among CRs, affording more frequent context information exchanges. This, in turn, reduces context information staleness and improves the performance of content delivery for both the network and the users.

II. CAINE FRAMEWORK

In any information-centric network, CRs need to be able to handle content publication and resolution messages, manage

content record entries and forwarding states, and correctly handle the transmission of content data itself towards the clients. In addition to these basic ICN primitives, CAINE also needs to be able to handle the exchange and processing of context information which we integrate with content publication.

Fig. 1 shows the architecture required to be implemented within each CR. Such an architecture enables three main functionalities related to the life cycle of a content:

- 1) *Content publication (box 1)*: Content servers send out two types of content publication messages (elaborated upon later): an original-content publication (OCP) message whenever a new content is uploaded to a server, and a pseudo-content publication (PCP) message sent at regular intervals to update the context information related to content information a server has previously published;
- 2) *Content resolution (boxes 2 and 3)*: Content clients submit content requests to their first-hop CR. This CR then determines the best next hop *towards* a content source hosting the requested content. The resolution routing decision is then carried out hop-by-hop at each CR until a content server is reached. The resolution decision made by each CR is based on the context knowledge each CR has accumulated from the PCP messages they have received;
- 3) *Content delivery (box 4)*: Content delivery is made along the reverse of the resolution path based on forwarding states created during the associated content resolution phase.

Each CR contains two main content management tables: a *content record table (CRT)* and a *content forwarding table (CFT)*. The CRT contains information associating content IDs with one or more network interfaces through which content requests can be routed towards a content node holding the requested content. In addition, each content item may have network or server context information indicating some QoS metrics related to reaching the given content from the node at which the CRT resides. In the example given in Fig. 1, three types of context information are shown:

- 1) *D*: the distance (number of CR-hops) from the content router to the content source;
- 2) *P*: the bandwidth available on the path between the content router and the content source, which is given by the minimum link bandwidth along the path;
- 3) *R*: the resources available at the content server, for example, the number of additional connections it can support.

The presence of context information in the CRT departs from the approach taken by NDN [4], which specifies in their forwarding information base (FIB, the equivalent of our CRT), only content ID (specifically, content prefix) and content resolution interface fields. A content publication manager (CPM) interfaces with the CRT to add or modify content records in response to different types of publication messages (expounded upon later) it receives from other content-aware nodes. In addition, a content resolution manager (CRM) also interfaces with the CRT in order to perform look-ups on the

next-hop interface to which to forward content requests. The CPM communicates with a PCP message processor (PMP) to perform operations on pseudo-content publication messages and determine the content IDs to which the received messages pertain.

Finally, the CFT (equivalent to NDN's *Pending Interest Table, PIT*) contains forwarding states related to ongoing sessions. It maintains associations between content IDs and outgoing next-hop interface(s) through which to forward received content towards the relevant client(s). The CRT will interface with the CFT to install forwarding states in response to content requests it receives.

CAINE is designed to facilitate accurate decision-making during content resolution, so as to ensure that traffic load is well-balanced across the network. Contrary to many existing ICN approaches, we take a more distributed approach in which all CRs within a domain are empowered with knowledge to help them make accurate content resolution decisions. We follow a gossip-style approach to content publication and resolution, essentially coupling together the physical signalling routes of content resolution and corresponding content delivery.

The main focus and novelty of our approach lies in the content publication process which is performed in two stages, and which together serves to facilitate that of *context-aware* content resolution. The following sections elaborate on the two content publication processes. In a later section, we explain how the information disseminated during content publication is used to intelligently resolve content requests in such a way that ensures well-balanced network load.

III. ORIGINAL-CONTENT PUBLICATION

An original-content publication (OCP) message is sent by a server whenever a new content is uploaded to or created at it to make CRs in the same domain aware of the *presence* of a new piece of content, as well as the direction towards it. An OCP message is encapsulated within an IP packet, and contains two fields: '*Message Type*,' and '*New Content ID*.' The '*Message Type*' field simply specifies that this message is an 'OCP' message, whereas the '*New Content ID*' field contains the ID of the new content being published. A server identifier is not sent in the OCP, thus completely decoupling content identifiers from server identities.

When a CR receives an OCP message, it will first confirm receipt by sending an acknowledgement to the previous-hop CR. It will then proceed to create a new content record in the CRT, filling in the content ID and the content resolution interface, which is the interface through which the OCP was received; the context information is not filled until the next PCP message is sent by the server, since the aim of an OCP is to make CRs aware of merely the presence of and direction towards the content in the network. To ensure that *all* CRs are made aware of the presence of the new content, a simple dissemination mechanism is employed in which each CR forwards the OCP messages it receives across all of its interfaces, except the one on which the message originally arrived. This allows the CRs to forward requests for content

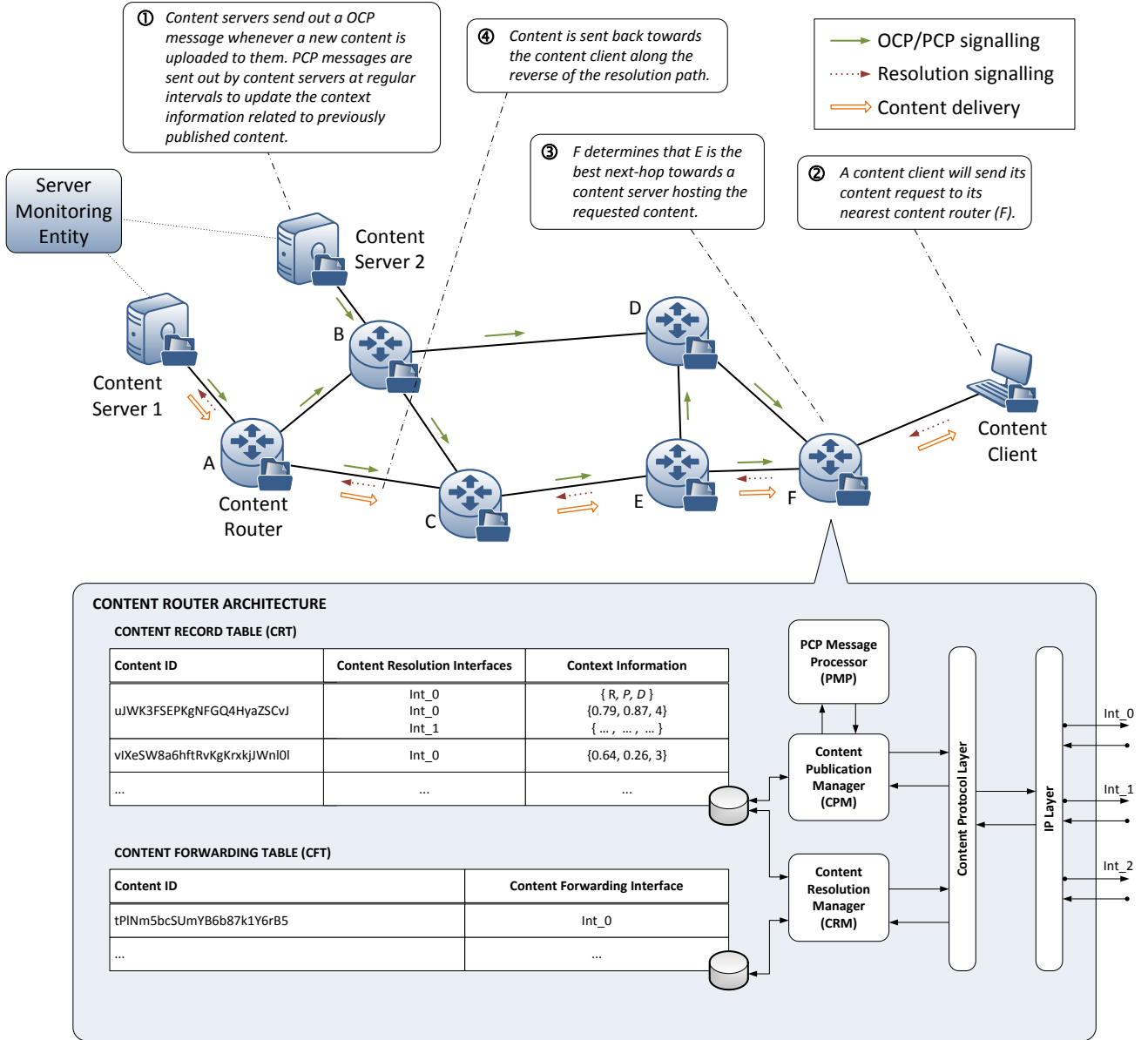


Fig. 1. CAINE network and router architecture.

along the correct interface leading to the content source. In the case where multiple sources along multiple network interfaces exist for a particular requested content, the CR must make a decision on the interface to use, *i.e.*, the direction in which to perform content resolution for the incoming request. To facilitate this decision, we propose the use of a special *pseudo-content publication* detailed in the following section.

IV. PSEUDO-CONTENT PUBLICATION

Pseudo-content publication (PCP) messages are sent periodically to update CRs with the latest context information (server and network state) related to the contents hosted at a server. The aim of the PCP is to disseminate knowledge about the current network conditions to the CRs, without revealing explicit information about servers' identities. This information empowers CRs to make decisions about routing content requests to the best available content source.

The PCP message contains three fields: '*Message Type*,' '*Context Information*,' and '*Bloom Filter*.' The '*Message Type*' simply indicates that the message is a 'PCP'. The '*Context Information*' field contains network and server context associated with the content hosted at that server. Specifically, this context information relates to the server resource availability, R , path bandwidth, P , and the distance, D , in CR-hops to the source hosting the requested content.

The last field in the PCP is a Bloom filter [5], which is a probabilistic data structure that allows for a set of elements to be represented by a single space-efficient bit string. Computationally-efficient logic-based set membership queries can then be performed on it to determine if an element is a member of the set it represents. In our case, the set of elements represented by the Bloom filter is the set of IDs of contents hosted at the server that generated the PCP. Set membership

queries are performed by CRs using prior knowledge of the content IDs gained through OCP dissemination. Therefore, with both the OCP and PCP messages, the CRs can build up next-hop routing knowledge for each content, together with their associated context, without ever having exposed to the CRs any form of server identifier.

The rest of this section details the way in which the Bloom filters are constructed and elements tested for membership, the strategy for disseminating PCP messages throughout the network, and the way in which the PCP messages are processed at CRs.

A. Bloom Filter Construction

In order to produce the Bloom filter message, each element, $e_i, 1 \leq i \leq n$, in the set, S , is hashed k times using k independent hash functions. The resulting Bloom filter bit array representing S is formed of m bits, which is given by [6]

$$m \geq \frac{n \log_2(1/p_f)}{\ln(2)} \quad (1)$$

where p_f is the false-positive probability, *i.e.*, the probability that a test for element membership of element $e_j \notin S$ is positive when it should be negative.

In Fig. 2 we show a simple example of the construction of a Bloom filter by a content server, and the testing of elements for membership by the CRs. In this example, the Bloom filter, B , is constructed by performing the bitwise OR operation on three elements, e_1 , e_2 , and e_3 . The content server then places the resulting Bloom filter, B in a PCP packet together with the related context information, C . Each content router receiving the PCP checks, in turn, each content ID within its CRT for membership in B . This is achieved by performing a *bitwise AND* operation with B . In the example, the result of the bitwise AND operation of e_1 and B yields e_1 , indicating that e_1 is a member element of the Bloom filter. If some other element e_6 were to be tested for membership in B , the result would be negative, since the bitwise AND operation of e_6 with B does not yield e_6 .

From [6] we know that the optimal number of hash functions, k , is given by

$$k = \log_2 \left(\frac{1}{p_f} \right) \quad (2)$$

whereby *optimal* implies that m is minimized subject to meeting the target false-positive probability, p_f .

From Eqs. 1 and 2, we can observe two key characteristics of Bloom filters that make them ideally suited to our application:

- 1) The size of a Bloom filter is independent of the size of the elements. This means it is possible to use very long content IDs without increasing the size of the Bloom filter. For instance, to create a Bloom filter for 50,000 content IDs, and given that the optimal number of hash functions is used, then to achieve a false-positive probability of 2%, approximately 8 bits per element will need to be used, giving a total Bloom filter size of approximately 50 kB. If the size of each content ID

is assumed to be 256-bits, then without Bloom filters, conveying information about 50,000 IDs would require approximately 1.5 MB of space.

- 2) The number of hash functions used has a bearing on the computational complexity of the Bloom filter, since the number determines the bits that need to be read to test for membership. From Eq. 1 we can deduce that the optimal number of hashes grows only linearly with the number of bits per element, b , where b is given by the ratio m/n .

In the case where a false-positive occurs, we anticipate that this won't have an adverse effect on the performance of our proposed mechanism, for reasons which we will explain in the coming sections.

B. PCP Dissemination Strategy

Once a PCP message has been constructed by a content server, it needs to be disseminated to other CRs within the network. Since the PCP message is building up a distance-vector-type view of network context such as available path bandwidth and path length, each CR is required to forward the PCP message across all of its interfaces, as in the case of the dissemination of OCP messages. In the process of dissemination, context information on the network side will be added and updated within the PCP at each CR hop. This is done as an 'offline' background process such that it does not interfere with the routing and resolution efficiency of CRs. To avoid potential routing loops, PCPs are forwarded using the *split-horizon rule* [7], *i.e.* CRs forward PCPs along all of its interfaces except the one on which it was originally received.

An important issue to consider is the frequency with which PCP messages are disseminated, as stale context information reduces the accuracy of content resolution decisions. Since PCP messages relate not only server resources but also network path bandwidth, the frequency with which servers send out PCP messages also needs to take into account the dynamics of link bandwidth availabilities. However, from experiments carried out based on real network traffic traces as well as real traces of user requests to YouTube servers [8], we found there is a strong correlation between the available server resources and the link conditions of the network. Therefore, such strong correlation between the two metrics validates a mechanism to determine the most suitable update frequency based purely on the server resource availability.

Since our aim is not necessarily to achieve perfectly load-balanced servers, but rather to avoid servers from becoming overloaded, we propose a PCP dissemination frequency based on a non-linear set of triggers. With such triggers, PCP messages would be sent more often when server load is high, and less often when it is low. At times when the server load fluctuates little, hence not crossing any triggers, a PCP message may still be disseminated after a given time has elapsed from when the last one was sent. The purpose of these time-driven PCPs is to ensure that new contents recently published by a server using an OCP have some context attached to them.

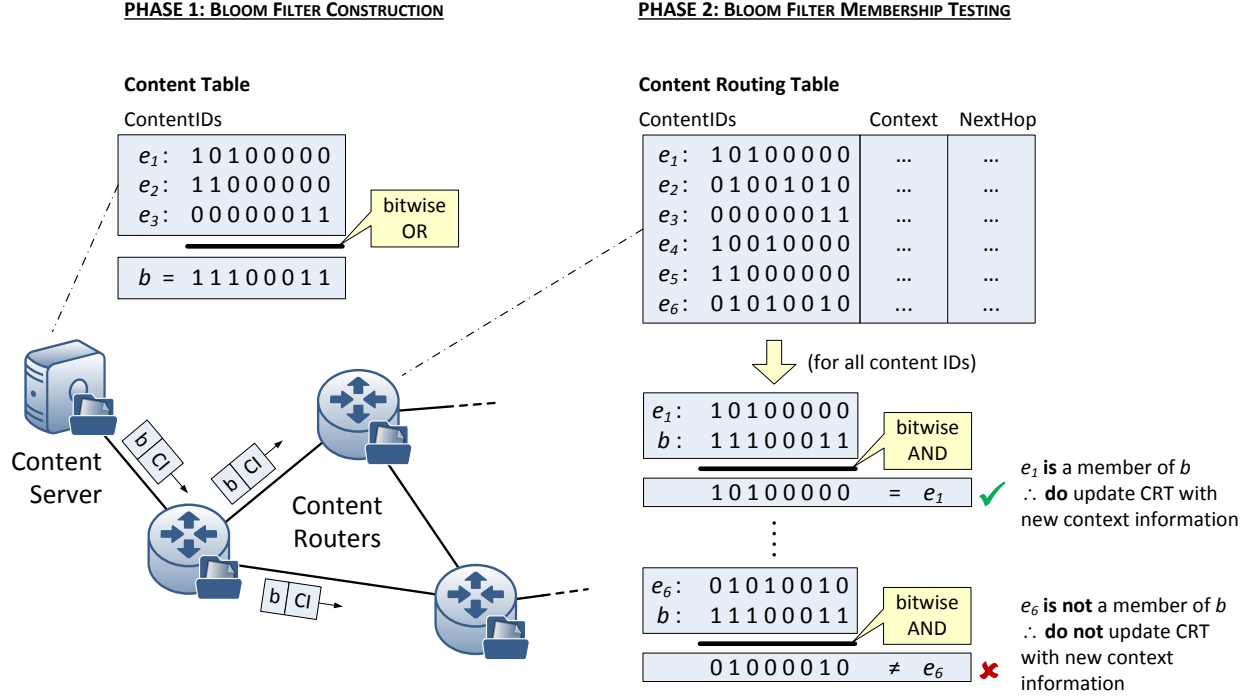


Fig. 2. Bloom filter construction and element membership testing.

C. PCP Processing Strategy

When a CR receives a PCP, it updates the context information contained within the PCP, determines which entries in its CRT to update with the updated context information, and then forwards the PCP to the next-hop CR(s). The full PCP message processing algorithm is shown in Fig. 3.

In order to determine which entries in its CRT to update with the new context information, the CR checks in turn each content ID, $e_i, 1 \leq i \leq N_{crt}$, within its CRT for membership within the Bloom filter, B . This is done in the manner described previously and as illustrated in Fig. 2. If e_i is found to be a member of B and the ID of the network interface on which the PCP message was received matches one of the Interface field entries in the CRT for the given content ID, then the context information related to that content ID and interface is updated with the new context information.

To minimize the effect of Bloom filter false-positives, the PCP message processing algorithm checks not only for membership of the content ID in the Bloom filter; it also checks for membership of the identifier of the network interface on which the PCP message was received in the given content ID's CRT interface list. Thus, if a false-positive does occur, it will affect only the *accuracy* of the context information of that particular content ID's interface information. As a result, the content resolution process (discussed in the next section) will still be able to route content requests towards *one* of the available sources of the requested content, although the routing decision may be suboptimal.

Once a CR has extracted the relevant information from the PCP message, the context information contained therein is updated before being forwarded to the next-hop CR(s). For example, the distance metric is incremented by one,

```

1:  $B \leftarrow \text{PCP.BloomFilter}$ 
2:  $C \leftarrow \text{PCP.ContextInfo}$ 
3:  $I \leftarrow \text{IP.GetRcvInterface(PCP)}$ 
4:  $C.D \leftarrow C.D + 1$  {update hop count}
5:  $C.P \leftarrow \min(C.P, I.BW)$  {update path bandwidth}
6: for each row in CRT do
7:    $E \leftarrow \text{row.ContentID}$  {content ID element to hash}
8:    $E_{\text{hashed}} \leftarrow E$  {initialize hashed element}
9:   for each HashFn in B.HashFns do
10:     $E_{\text{hashed}} \leftarrow \text{HashFn}(E)$ 
11:   end for
12:   check  $\leftarrow (B.\text{BitString} \mid E_{\text{hashed}})$ 
13:   if (check =  $E_{\text{hashed}}$ ) and ( $I \in \text{row.Interfaces}$ ) then
14:     CRT.UpdateContent(row.ID,  $I, C$ )
15:   end if
16: end for
17: PCP.ContextInfo  $\leftarrow C$ 

```

Fig. 3. Pseudocode for processing a Publish primitive at a CR.

whereas the path bandwidth is updated with the measured bandwidth of the link through which the PCP message was received, and is updated only if the locally measured link bandwidth is less than the overall path bandwidth¹. The server resource availability information is not changed along the PCP dissemination paths.

¹The bandwidth of a link is reverse estimated by a CR by passively measuring the rate of data it receives and subtracting this from the total supported data rate.

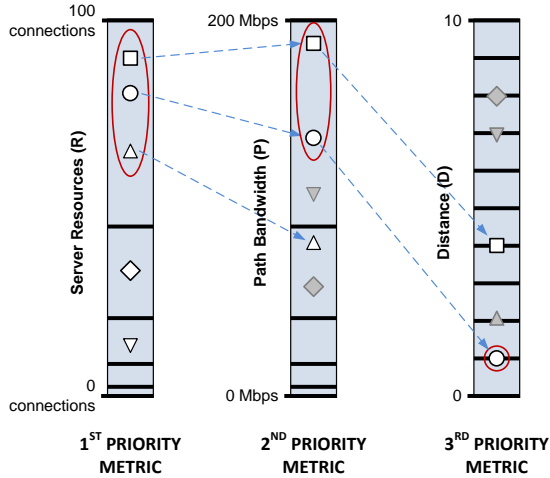


Fig. 4. Example of metric prioritisation for content resolution.

V. CONTENT RESOLUTION

When a content router receives a content request and there is more than one interface in its CRT attached to that content ID, the CR must make a decision about the ‘best’ interface to use to forward the content request. To make this decision, the CR prioritizes the various context metrics, and performs tie-breaker tests on each priority metric in turn, as illustrated in Fig. 4. If the values of a given metric are equal, or lie within a certain pre-determined range from each other, those interfaces qualify for the next lower-priority round of selection. For this particular work, server resources, R , is given the highest priority, followed by path bandwidth, P , and then finally the distance, D . In order to allow all context metrics to be given consideration, we use non-linear ranges, such that higher QoS metric values have looser ranges, and vice versa.

In the example shown in Fig. 4, for a given content ID there are five interfaces from which to choose. In the first priority metric selection, the algorithm qualifies to the next round of selection the three interfaces towards servers having the highest resource availability and that are within the non-linear bounds. In the second priority, the algorithm qualifies out of the three interfaces the two having the highest path bandwidth to the content source and that are within the non-linear bounds. Finally, for the third priority metric, out of the two interfaces that qualified to the third stage of selection, the one with the least number of hops towards the content server is selected as the *best* interface.

Once a best next-hop along which to forward the content request is determined, the CR will install a forwarding state in the CFT to indicate the interface through which to send the content data towards the content client.

VI. EVALUATION

The performance of CAINE was evaluated by means of computer simulation using the GÉANT topology as the reference topology [9], a pan-European point-of-presence (PoP) data network for the research and education community. Specifically, we used the topology of the year 2004, which consists of 23 PoP nodes, and 74 high capacity interconnecting

network links. This choice of topology was influenced primarily by the availability of real path congestion measurements extending over four months conducted within the TOolbox for Traffic Engineering Methods (TOTEM) project [10], hence ensuring that the modelling is realistic. Five content servers with equal connection capacity were deployed at various locations within the network, each hosting 10,000 contents selected randomly from a pool of 25,000 possible contents, except for the 1,000 most popular contents, which were hosted by *all* servers.

The rate and length of the videos were made to follow the measured trends reported by Cheng *et al.* in [11]. User request patterns were synthetically generated for a 24-hour period based on the characteristics of real YouTube request traces collected by Zink *et al.* [8]. This synthesis was achieved by observing the mean request rate, λ in each 15-minute interval, Δ_i , and generating Poisson-distributed sets of content requests with different factors, β , of the *set* of mean content request rates. The popularity of the contents followed a Zipf distribution with a shape parameter, α of 1.0. Finally, the performance of the CAINE scheme is compared against a context-unaware scheme, i.e. one that is agnostic to the server resource and path bandwidth availabilities, and that forwards all content requests along the shortest path towards the nearest source that holds the requested object.

Fig. 5 shows some key traffic engineering performance metrics of CAINE based on our simulation of content request events over a 24-hour period. To illustrate CAINE’s load-balancing efficacy, we captured in Fig. 5(a) the proportion of time for which the most heavily loaded content server was saturated during the 24-hour period. At low content request rate factors, β , content servers never reached saturation, but with increasing β , content servers suffer from significantly longer duration of saturation in the context-unaware case than in the context-aware case. At $\beta = 2.5$, the most heavily loaded server was saturated for 29% of the time under the context-unaware scheme, and only 9% of the time under the context-aware scheme. We can also glean some insight from the degree of server load imbalance across the five content servers, which we define to be the mean statistical range of server loads across the 24-hour period that was simulated. It was found that under low β , the context-aware approach had a load imbalance of 13%, which is marginally greater than that of the context-unaware approach. This is due to the fact that at low content request rates, server utilizations are relatively low, resulting in reduced frequency of PCP updates, and hence reduced freshness of context information and less optimal resolution decisions. However, with increasing values of β , as server utilizations reach more critical levels and PCP exchanges become more frequent, the load imbalance of the context-aware approach tended towards 10%, whereas that of the context-unaware approach increased to as much as 33%.

One of the benefits of CAINE’s load balancing efficacy is its optimal network utilisation, reflected by its higher ratio of successfully resolved content requests in comparison to the context-unaware approach, as shown in Fig. 5(b). At high β , the resolved content request ratio of the context-aware scheme was as much as 11% more than the context-unaware

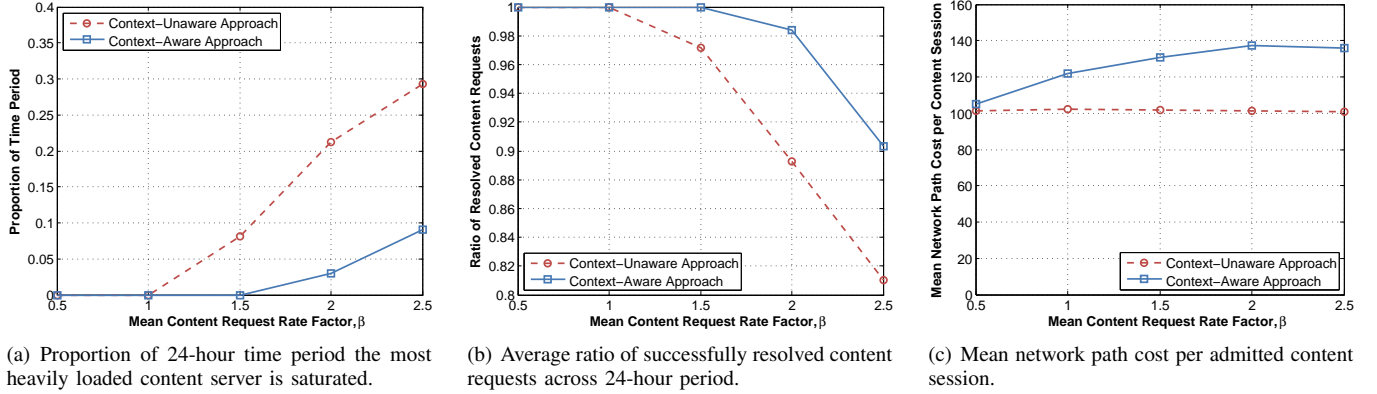


Fig. 5. Simulation results.

scheme. This is due to CAINE’s ability to use alternative paths to the shortest path, as well as alternative servers with more resources available. However, the tradeoff of CAINE’s increased network utilisation is higher network path cost per content session, which we computed based on the sum of the costs of the individual links traversed along a content delivery path. The cost of using a link is given by a piecewise function defined in [12], which is dependent on the link’s utilization, and increases exponentially with it. Therefore, it essentially captures together both the link utilization and path length metrics. From Fig. 5(c) it can be seen that at the lowest simulated β , the network path cost of the context-aware scheme is only marginally greater than the context-unaware scheme, but can be as much as 35% more with higher β .

With regular PCP exchanges being at the heart of CAINE’s operation, we looked at the mean frequency with which PCP messages are exchanged by servers at different values of β . The PCP messaging frequency was observed to increase exponentially with increasing β , from PCPs being exchanged every 14 minutes at $\beta = 0.5$, to every 20 seconds at $\beta = 2.5$. However, such high rate of messages occurs only when the content servers approach saturation. Furthermore, such messages are approximately only 10-KB in size for advertising 10,000 contents, and are processed efficiently at CRs as a background (‘offline’) process.

VII. CONCLUSION

To facilitate efficient use of network resources in ICNs, we have proposed a novel context-aware ICN-based scheme called CAINE in which location-independent context information is efficiently published to the network to facilitate better decisions during content resolution. Bloom filters are periodically constructed to efficiently convey the IDs of contents hosted at servers, to which up-to-date context information is then attached and disseminated to the content routers. Such a mechanism avoids the need to reveal to the network explicit condition information associated with physical elements within the ecosystem, thereby upholding the key ICN principle of location-independence.

Through simulation, we have shown that CAINE can achieve optimized network utilization and effective load-balancing between servers, particularly when their utilizations

are at critical levels, albeit at the cost of increased network path cost. Furthermore, the use of Bloom filters ensures that such messages do not pose a significant overhead to the network, from the perspectives of both transmission and content router processing.

REFERENCES

- [1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, “A survey of information-centric networking,” *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26–36, 2012.
- [2] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, “A data-oriented (and beyond) network architecture,” *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 181–192, Aug. 2007.
- [3] G. Pavlou, N. Wang, W. Chai, and I. Psaras, “Internet-scale content mediation in information-centric networks,” *Annals of Telecommunications*, vol. 68, no. 3–4, pp. 167–177, 2013.
- [4] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking named content,” in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, ser. CoNEXT ’09. New York, NY, USA: ACM, 2009, pp. 1–12.
- [5] B. H. Bloom, “Space/time trade-offs in hash coding with allowable errors,” *Commun. ACM*, vol. 13, no. 7, pp. 422–426, Jul. 1970.
- [6] A. Broder and M. Mitzenmacher, “Network applications of bloom filters: A survey,” *Internet Mathematics*, pp. 636–646, 2002.
- [7] C. Hedrick, “Routing Information Protocol,” RFC 1058 (Historic), Internet Engineering Task Force, Jun. 1988, updated by RFCs 1388, 1723. [Online]. Available: <http://www.ietf.org/rfc/rfc1058.txt>
- [8] M. Zink, K. Suh, Y. Gu, and J. Kurose, “Characteristics of YouTube network traffic at a campus network - Measurements, models, and implications,” *Comput. Netw.*, vol. 53, no. 4, pp. 501–514, Mar. 2009.
- [9] GEANT project home. [Online]. Available: <http://www.geant.net/>
- [10] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, “Providing public intradomain traffic matrices to the research community,” *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 83–86, Jan. 2006.
- [11] X. Cheng, J. Liu, and C. Dale, “Understanding the characteristics of Internet short video sharing: A YouTube-based measurement study,” *Multimedia, IEEE Transactions on*, vol. 15, no. 5, pp. 1184–1194, Aug. 2013.
- [12] B. Fortz and M. Thorup, “Optimizing OSPF/IS-IS weights in a changing world,” *IEEE J. Sel. A. Commun.*, vol. 20, no. 4, pp. 756–767, Sep. 2006.

BIOGRAPHIES

George Kamel (g.kamel@surrey.ac.uk) is a Research Fellow at the Institute for Communication Systems (ICS), University of Surrey, UK. He received his M.Eng. (Honours) in Telecommunications Engineering and his Ph.D. in Mobile Communications from King’s College London, UK in 2005 and

2010, respectively. He is currently involved in research efforts at the 5G Innovation Centre (5GIC), University of Surrey, UK, and has been involved in a number of EU, EPSRC and industry-funded projects. His research interests include information-centric networking, context-aware network management, and cellular traffic offloading.

Ning Wang (n.wang@surrey.ac.uk) is a Reader at the Institute for Communication Systems (ICS), University of Surrey, UK. He received his B.Eng (Honours) from Changchun University of Science and Technology, P.R. China in 1996, his M.Eng. from Nanyang Technological University, Singapore in 2000, and his Ph.D. from the University of Surrey, UK in 2004. He currently leads the work area on content, user and network context at the 5G Innovation Centre (5GIC), University of Surrey, UK. His research interests include information-centric networking, context-aware network management, and mobile content delivery.

Vassilios Vassilakis (vv274@cl.cam.ac.uk) is a Research Associate at the Computer Laboratory, University of Cambridge, UK. He received his Ph.D. in Electrical and Computer Engineering from the University of Patras, Greece in 2011. From 2011 to 2013 he was with the Network Convergence Laboratory, University of Essex, UK where he conducted research on information-centric networking, and from 2013 to 2015 he was with the Institute for Communication Systems (ICS), University of Surrey, UK, where he conducted research on 5G wireless networks. He has been involved in a number of EU and industry-funded R&D projects. His research interests are in the areas of future Internet technologies and next-generation wireless networks.

Zhili Sun (z.sun@surrey.ac.uk) is a Professor at the Institute for Communication Systems (ICS), University of Surrey, UK, and has been with the university since 1993. He received his B.Sc. in Mathematics from Nanjing University, China in 1982, and his Ph.D. in Computer Science from Lancaster University, UK. He worked as a Postdoctoral Research Fellow at Queen Mary, University of London, UK. He has been Principle Investigator in many projects funded by EU, ESA, EPSRC and industry, and has published over 195 papers and three books. His research interests include wireless sensor networks, mobile ad hoc networks, satellite networks, mobile operating systems, future Internet protocols, and security.

Pirabakaran Navaratnam (piraba.navaratnam@rssi.co.uk) is an experienced Research Engineer with a background in mobile communication technologies and Internet of Things. He received his B.Sc.Eng. in Electronic and Telecommunications Engineering from the University of Moratuwa, Sri Lanka, and his Ph.D. in Mobile Communications from the University of Surrey, UK. He worked as a Research Fellow at the Institute for Communication Systems (ICS), University of Surrey, UK. He is currently working for RSSB, UK on European research and innovation programmes. His research interests include machine-to-machine communication, resource management, information management systems, and future Internet architecture design.

Chonggang Wang (chonggang.wang@interdigital.com) is currently a Member of Technical Staff at InterDigital Communications, USA where his focus is on Internet of Things (IoT) R&D activities including technology development and standardization. He received his Ph.D. from Beijing University of Posts and Telecommunications (BUPT), China in 2002. He is the founding Editor-in-Chief of IEEE Internet of Things Journal, and is on the editorial board for several journals including IEEE Transactions on Big Data and IEEE Access. He is an IEEE ComSoc Distinguished Lecturer (2015-2016). His research interests include IoT, mobile communication and computing, and big data management and analytics.

Lijun Dong (lijun.dong@interdigital.com) is currently a Staff Engineer at InterDigital Communications, USA. She received her Ph.D in Electrical and Computer Engineering from Rutgers University, USA in 2011. Her research interests include machine-to-machine communication, Internet of Things, and information-centric networking.

Rahim Tafazolli (r.tafazolli@surrey.ac.uk) is a Professor and the Director of the Institute for Communication Systems (ICS) and the 5G Innovation Centre (5GIC), University of Surrey, UK. He has published more than 500 research papers in refereed journals, international conferences, and as invited speaker. He is the editor of two book volumes published by Wiley in 2004 and 2006, entitled "Technologies for the Wireless Future". In April 2011 he was appointed a Fellow of the Wireless World Research Forum (WWRF), in recognition of his personal contribution to the wireless world.