Information-Centric Networking: A Natural Design for Social Network Applications

Bertrand Mathieu, Patrick Truong, Wei You, and Jean-François Peltier, Orange Labs

ABSTRACT

Millions of people now use online social network applications such as Twitter, Facebook, and Google+. This kind of application offers the freedom for end users to easily share contents on the Internet. In parallel with the expansion of OSNs, a new networking paradigm emerges, the *information-centric networking* approach, where the focus is on the content the user wishes to obtain instead of the server that provides this content: a content-based approach instead of a host-centric one. Looking at the ICN concept, and more precisely the CCN (which stands for content-centric networking) solution in this article (based on Interest-Data messages), and at OSNs behavior (users subscribing to contents posted by another user in case of Twitter, or the notion of friends receiving contents from others friends in Facebook), we advocate that the CCN paradigm perfectly fits with the OSN applications behavior and can help to save network load. In this article, we describe how OSN applications can run with the CCN solution and the evaluation we have performed highlights the benefit of using such a CCN approach compared to the current classical IP-based delivery and to the CDN-based solution, for the use-case of Twitter. Depending on the CCN nodes location and the cache efficiency, the network load can be significantly reduced and the response time to get the content can be faster up to 60 percent.

INTRODUCTION

Social networking has been a growing trend for a few years, and millions of people interact now with each other every day across online social websites such as Facebook, Twitter, and YouTube (which is not only a video repository, but can also be considered as a social platform). The recent uprisings in the Arab world have shown the impact of these media on our daily lives. Social networking is redefining the way we use the Internet, shifting away from browsing information to online consuming and sharing all types of content, including user-generated content. With the fast viral spread of information in social networks, traditional end-to-end communications tend to disappear to make way for one-to-many or many-to-many dissemination and retrieval of content. Content distribution is thus influenced by the multiple interactions between social media users. However, all the current workarounds for content diffusion such as content delivery networks (CDNs) or peer-to-peer networks push content to the network edges to improve quality of user experience, but they still rely on host-to-host IP principles and do not take into account social semantics of transferred content for optimizing content routing or caching.

The popularity of social networking applications is one of the numerous reasons that have fostered information-centric networking (ICN) in the research community as a new paradigm for Internet architecture to directly route content based on users' interests. ICN focuses on finding and delivering information (i.e., the content a user wants) instead of maintaining end-to-end communications between hosts (i.e., where to find the content the user wants). A naming scheme uniquely identifies content at the network level, making it possible to use dynamic content caching so that queries for content in ICN are generally routed to the most efficient location in the network. As interexchanges in social networks mainly consist of contents such as photos or videos (users' posts can also be viewed as pieces of content), we believe that leveraging ICN as the core principle for content distribution can improve network performance in terms of minimal demand for transmission bandwidth, higher availability, and minimal delay.

This article is organized as follows. We present online social networking (OSN) applications and related work about analysis of social graphs to better serve user requests in OSN. After introducing the ICN paradigm, we show how an ICN architecture can increase performance of content distribution to cope with the huge number of users in OSN applications. We also illustrate our discussion with an evaluation based on Twitter and the content-centric networking (CCN) architecture proposed by Jacobson *et al.* [1] as an implementation of ICN. Finally, we conclude our article and outline our future work.

 0163-6804/12/$25.00 © 2012 IEEE

Online Social Networking Applications

In this section, we present the main features of the world leader OSN applications (i.e., Facebook and Twitter), give some figures showing their impact on network traffic, and finally present how the network delivery of such contents can be optimized.

Social networking derives from human society, which consists of grouping individuals or organizations related to each other through specific connections such as friendship, common interest, and cultural expectations, classmate relationships, business-to-business exchanges, and so on. With the popularity of Internet, the concept has naturally grown in scale to interconnect millions of users behind their computers so that people around the world can discuss online together. The most famous online social networking services are Facebook and Twitter, and recently Google has also proposed its own application, Google+, which combines and redefines the main features of Facebook and Twitter.

Facebook allows users to publish their personal information so that they can find other users sharing the same interests (friends, colleagues, etc.). Interactions between users are complex, including keeping up with friends (message publication on an user's wall), connecting people through interest-based groups, and multimedia document sharing. The social network also proposes numerous optional features (referred to as applications), such as photo and video sharing or RSS feed readers, which represent a set of personalized views of the user's Facebook page.

The social network Twitter is a microblogging service that allows users to send and read short text messages (called tweets) of up to 140 characters. Users can subscribe to (or follow) other users' tweets. Subscribers to a user's tweets are known as followers. Twitter is not only a way to connect and interact with others, but it is also more and more used for searching and sharing information in real time. Breaking news is often broadcast on the spot first on Twitter, and thanks to users' retweets or comments, information is spread very rapidly and virally through the social network and then over other communication media.

Nowadays, Internet is becoming a large repository of contents whose main providers are music, video or webTV streaming services. According to Cisco [2], video represents 40 percent of today's traffic, and will reach 62 percent by the end of 2015. The landscape of Internet usage is also being reshaped by the increased phenomenon of online social networks (OSNs), which has led to social sharing and bookmarking. The majority of websites or mobile applications do indeed integrate "like" or "share" buttons to let users share content they are viewing with their friends through the two main social platforms, Facebook and Twitter. A recent study by ShareThis [3] has shown that sharing activities on the web represent more than 10 percent of all Internet traffic. Facebook dominates social sharing with 38 percent. This is compared to 17 percent for Twitter and emails. As a consequence, OSNs represent a new powerful means of disseminating and finding content over Internet. An important part of accesses to contents on the Web comes then from OSNs. As an example, either people watch videos directly on the social networking site (in this case, users' videos are hosted in the OSN storage servers), or they watch videos through YouTube (or another third party streaming service) links shared by their friends. Over 150 years' worth of YouTube video is thus daily watched on Facebook [4].

Facebook and Twitter, as massively distributed applications, rely on content delivery networks (CDNs) in order to reduce the load on their own servers and to accelerate user requests that directly access data caches of CDNs. To scale with the growing rate of users, some studies recently propose having an in-depth understanding of the characteristics of the social network graph on a large scale to help CDNs optimize the content distribution. In [5], based on an auto-regressive prediction method, Twitter's trending topics are analyzed and used as harbingers to anticipate the popularity of some content for optimizing the content replication and replica placement problems in CDNs. Starting from the fact that the circle of a user's contacts in an OSN generally contains a densely connected core delimited by a geographically restricted area, the authors of [6] focus on geographic information extracted from OSN user interactions to efficiently cache content in CDNs. Following the same motivation, [7] proposes a framework to optimize content delivery in ad-hoc mobile social networks by extracting time-based usage patterns as users usually connect on their mobiles in specific periods of time.

Generally speaking, the emergence of new content consuming applications leveraged by the rise of OSNs redefines interdependencies between network entities, which depend now more on semantics of pieces of content and their social relationships, rather than physical endpoints addressed by the underlying IP protocol. One possible solution to efficiently map all the relevant interactions inside the resulting social graph is to use the new paradigm, ICN, which has been proposed in the past few years to envision a redesign of Internet architecture by placing content at the heart of network transactions. With Twitter as a use case, we then show in the following sections how an ICN approach can improve content exchanges between users in an OSN.

The emergence of new content consuming applications leveraged by the rise of OSNs redefines interdependencies between network entities, which depend now more on semantics of pieces of content and their social relationships, rather than physical endpoints addressed by the underlying IP protocol.

Information-Centric Networking

This section aims to present the main concepts of the ICN approach before detailing them with the CCN example.

Concepts

An ICN [8] is a set of interconnected pieces of information, also called content, information, or data objects, which are addressed by names for routing and managed by applications or services

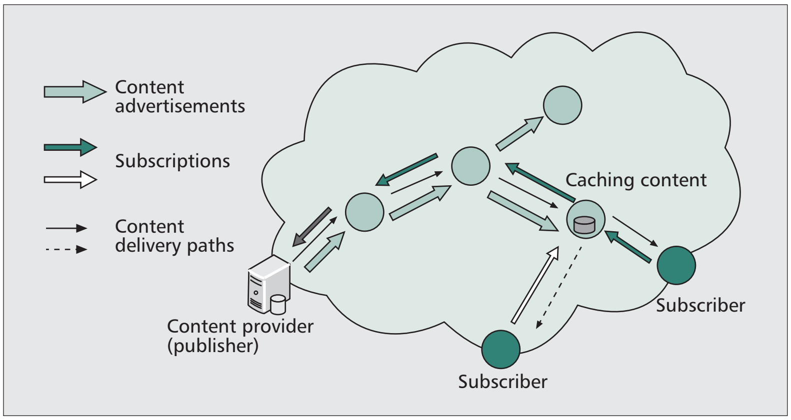

Figure 1. *Information centric networks.*

at the higher level. They can be of any type, including web applications, static or user-generated content, real time media streams, and more complex interactive multimedia communications, etc. The principal concern of the network is to disseminate, find and deliver information rather than the reachability of end-hosts and the maintenance of conversations between them.

The communication paradigm within ICN is different from what it is with IP. Current IP architectures revolve around a host-based conversation model (i.e., a communication is established between two hosts before any content is transferred), and the delivery of data in the network follows a source-driven approach (i.e., the path is set up from the sender to the receiver). In ICN, the user requests content without knowledge of the host that can provide it, and the communication follows a receiver-driven principle (i.e., the path is set up by the receiver to the provider), and the data follows the reverse path. The network is then in charge of doing the mapping between the requested content and where it can be found (Fig. 1). The match of requested content rather than the findability of the endpoint that provides it thus dictates the establishment of a communication in ICN.

To be efficient, one important aspect of ICN is naming. Content should be named in such a way as to be independent of the location of the node where the content can be found, which is the main objective of ICN (to separate naming and location). Indeed, it avoids receiving a message like "Error 404: File not found" like we get in HTTP if the server is down (being the initial server or a server replicating the content). ICN also includes a native caching function in the network, in such a way that nodes can cache the contents passing through it for a while (depending on the cache size and replacement algorithm) and deliver them to requesting users. Via this caching mechanism, the content is replicated, and the delivery probability of the content to the end user is increased.

Decoupling naming from location also allows native support of mobility or multicast in ICN. Indeed, when users move, they are connected to another node in the ICN network, but since no IP address is used for the routing, it is transparent, as opposed to IP, where the address should be changed. For multicast, as soon as one user has requested a given content, one node can cache it and then deliver it for subsequent requests for the same content. It then naturally create a multicast-like delivery.

Introduction to CCN: A Pioneer ICN Solution

One of the well-known ICN networking solutions is CCN [1, 9], designed by the Palo Alto Research Center (PARC).

In CCN, the end user sends an Interest message for the content in which she/he is interested. This Interest message is only identified by the content name. A Data message is returned back to her/him as a response. This matched Data packet is identified by the same content name. Thus, the traditional IP forwarding table, routing with IP addresses, is no longer suitable and should be adapted to forward data, resolving the content names instead of the IP addresses. This adapted forwarding table is named the forwarding information base (FIB) in CCN and contains the identifiers of the content (not the IP addresses) as well as the outgoing interfaces to which to forward the message. Indeed, as the network does not deal with any notion of destination, the next node is not referred to the next hop identified with an IP address but referred with only the outgoing interface information, which connects the current node to its neighbors, which may have the right contents or knowledge about how to propagate the Interest message to potential sources. Routing data per interface then leads to hop-by-hop routing, without knowledge of the final destination of the packet.

The pending interest table (PIT) is a new component in the CCN node not present in IP routers. When each CCN node receives an Interest packet for a given content, it keeps track of it and of the incoming interface on which it receives the Interest, before consulting the FIB table and forwarding the Interest to the next hop(s). When this CCN node receives the response (the Data message matching the Interest message), it looks up in the PIT the interface(s) information through which the matching Interest(s) came and forwards the content through all the matching interface(s). After forwarding of the Data packet, the entry for this content is removed from the PIT. In case of multiple interests in the same content, the CCN node will forward only the first Interest message once, but keeps track of all the interfaces from which it received such interest messages in order to forward/duplicate the corresponding Data packet to all the interfaces when it gets the response. Doing so, the CCN network then naturally offers a native multicast function.

Finally, the last component in a CCN node is the content store (CS), which is a cache for contents. Received Data packets will be cached locally in the CS. If an Interest message is received for content already cached in the CS, the CCN node will just deliver it from the CS

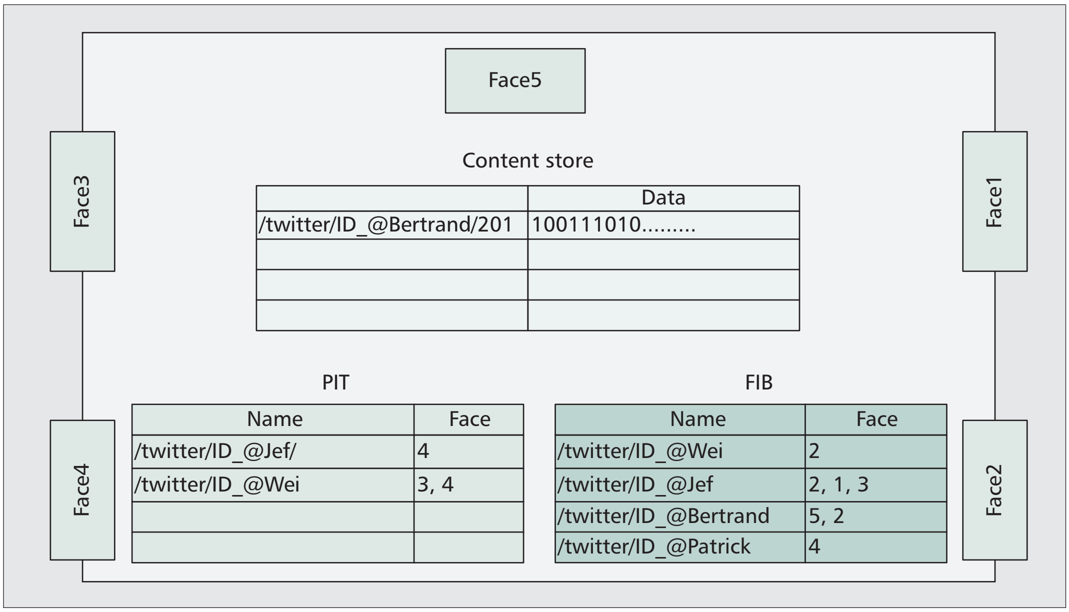

Figure 2. *Example of one CCN node with twitter traffic.*

To be efficient, one important aspect of ICN is the naming. Content should be named in such a way as to be independent of the location of the node where the content may be found, which is the main objective of ICN (to separate naming and location).

without filling the PIT table and without forwarding the Interest message upstream. The larger the CS, the more contents can be cached.

There are several studies related to the cache function in CCN (e.g., the location of caches in the network, the cache size for efficiency, the replacement strategy [10, 11]). With this feature, the CCN network then naturally offers a caching function in order to reduce the time to deliver some contents (popular contents) and also to reduce the load on the network.

For the scalability of the CCN network, a hierarchical naming structure is proposed, which allows the aggregation of entries in the FIB based on the content name. Indeed, in order to reduce the size of the FIB, the entry can be aggregated based on content provider name (i.e., twitter.com or facebook.com) or application name prefix (i.e., *#topic#* name for Twitter), or even ISP prefix.

Since in the PIT the Interest and content messages should be exactly matched, it is not difficult to imagine that the number of contents in CCN might be very huge. However, since an entry in the PIT is removed as soon as the corresponding Data is coming, the size is smaller. Furthermore, some ongoing research works investigate how to avoid having a huge PIT table and improving the scalability of the CCN proposal, for example, making the PIT distributed in order to match today's memory technologies limits, or implementing a space-efficient architecture based on a Bloom filter in order to greatly reduce the table size.

The three main functional components of a CCN node are depicted in Fig. 2 with the example of Twitter traffic (which we describe in detail in the next section): the FIB to find the appropriate interface(s) to which arriving Interest packets should be forwarded, the content store aimed at caching contents, and the PIT to keep track of the inbound interfaces of received Interest packets.

Information-Centric Social Networks

This section presents how OSN applications can benefit from an ICN network to deliver content to end users in an efficient way, from both the network point of view (reduction of the network load) and the end user point of view (improvement of the latency to get the content). After a description of the approach, the evaluations we have performed are presented.

Information-Centric Architecture for OSNs

As previously seen, OSN applications can send data to few or many users (e.g., followers). For Twitter, for instance, sending one tweet to 1000 followers can be considered as a mix of multicast delivery and caching. ICN networks, having such behavior, could really help in the delivery of tweets while caching contents on the path and providing it to requesters. The same method can apply for the delivery of Facebook or Google+ data (i.e., update of the wall with new photos or announcements and sending to friends).

Figure 3 schematizes the three content delivery architectures for Twitter we took into consideration and shows the evolution from the client/server IP network, via the CDN network, to the CCN network. For the client/server IP architecture, the end user fetches the content directly from the server. For instance, in the case of Twitter, one end user, connected in France to the Orange network, will regularly (every 90 s) poll the Twitter server to be aware of the latest tweets by people she/he follows. This IP traffic then passes through several networks from the Orange access network in France toward the local Twitter site hosting the server. In order to reduce the network load as well as improve the response time for end users, service providers (Twitter, Facebook, etc.) now use CDN net-

With CCN, the benefit of native added-value functions, such as caching, multicast, and multipath in the network, applies at many levels, enabling the network load to be greatly reduced.

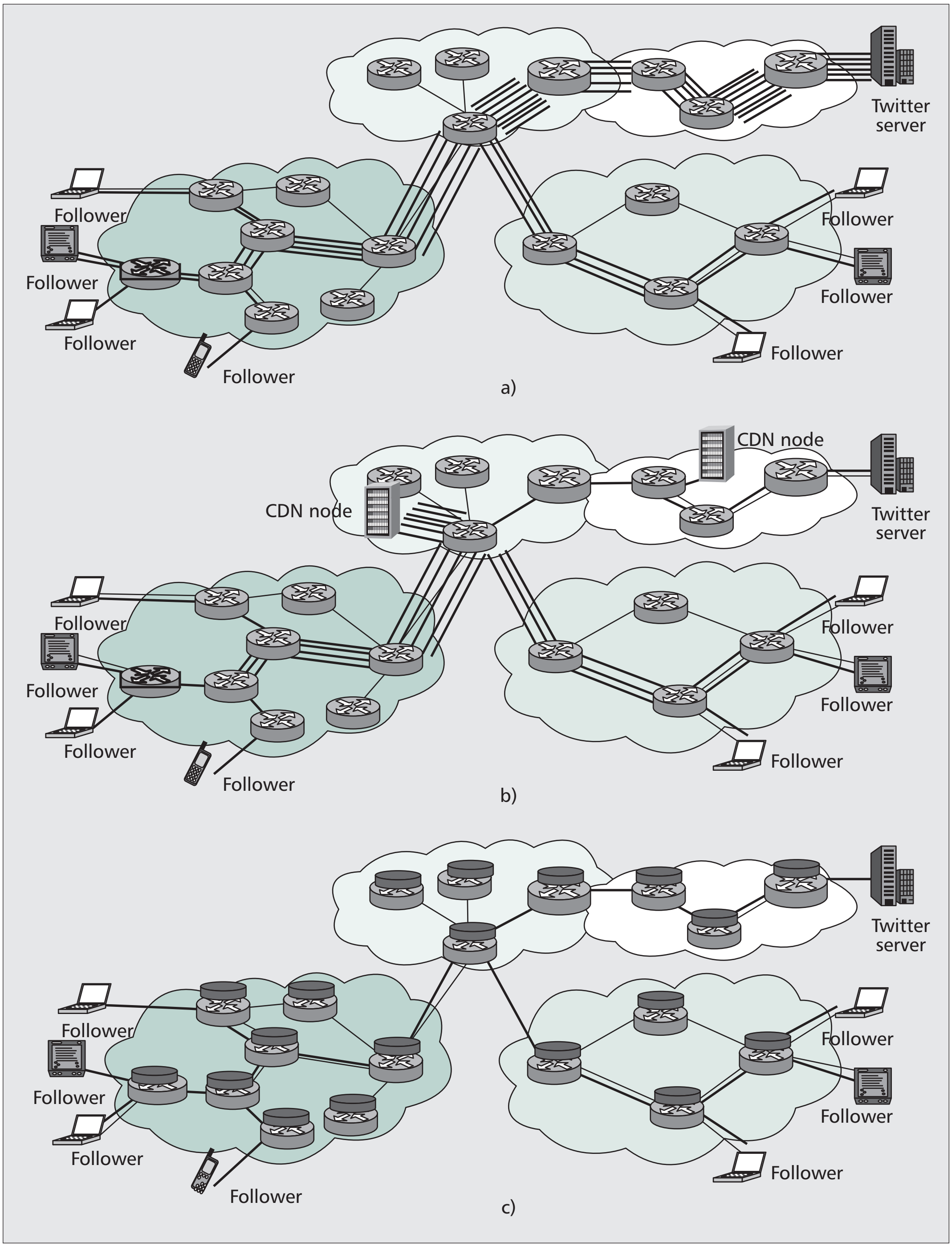

Figure 3. *a) IP-based; b) CDN-based; c) CCN-based network infrastructures.*

works. The main idea of CDN is to have the most popular contents available in those nodes (in a push-based manner or dynamically upon request) that can provide a shorter response time since they are closer to the end users. The dominant CDN players (Akamai, Limelight, etc.) are not network operators but third-party providers and deploy their CDN infrastructure outside of the network of a network operator, as an overlay network. For our use case, the request from one Twitter user in France should still pass trough the whole Orange network, go to the closer CDN node operated by the CDN provider (possibly via other intermediate networks), and come back. Compared to IP, the network traffic is then reduced from the CDN node to the Twitter server, but not from the end users to CDN node. The last architecture is the CCN one, where nodes in the path can cache and forward content to subsequent end users requesting the same contents. Then for popular contents, several end users in the same network vicinity can request the same content (e.g., many followers of Twitter users or a popular video announced on a social network), and the CCN node can deliver it by itself if it has it in cache. With CCN, the benefit of cache function in the network applies at many levels, enabling the network load to be greatly reduced, starting from the operator network. For live content it is similar since the CCN node will receive the content only once from the upstream node/server and deliver it downstream to all interested entities. There is no need for all end users to fetch it from the original server or the CDN node.

For the use case of the Twitter application, in the current architecture the servers are useful to provide a unique identifier *ID_@user* for each user. This naming is crucial for this application; thus, we should also provide it efficiently. In our CCN-based architecture for Twitter, we also rely on those central servers to propose a naming scheme, but in accordance with the CCN naming. Typically, each Twitter user, considered an object in CCN, is then named with the identifier */twitter/ID_@user* (the prefix *twitter* is used to ensure global uniqueness of names from all other information-centric services). Based on this naming for users, each message from the user can now be named */twitter/ID_@user/timestamp*$_{msg}$ with a suffix indicating the date of the message publication.

After a user gets her/his CCN-like Twitter name prefix, her/his followers use this identifier to send an Interest message to the CCN node and then get automatically updated with the user's new tweets. Let us take an example, as illustrated in Figs. 4 and 2. User *Patrick* sends an Interest message to follow *Jef*, one message to follow *Bertrand*, and another to follow *Wei*. We suppose that end user *Patrick* is connected to the CCN node via *face4*. The CCN node receiving the interest message will first check in its CS if it has the content in cache. For the interest related to *Bertrand*, it has the content and will deliver it itself to *Patrick*, by sending back the content via face4. For the interest related to *Jef*, it is not in the CS; the CCN node then looks into its FIB table to know where to forward the interest message to: it has to forward it out of its *face2*, *face1*, and *face3*. For the interest related to *Wei*, it has to forward it out of its *face2*. The interest messages forwarded by the node (i.e., related to *Jef* and *Wei*) are indicated in the PIT table with the associated incoming face (i.e., *face4*), corresponding to the interface from which *Patrick*'s traffic is coming. Later, after reception of the requested content, the CCN node will forward the Data message back to *Patrick* via this interface. For interest messages issued by *David* or *Elena*, wishing to follow *Patrick*, the CCN node will forward them out of its *face4*, as specified in the FIB.

Based on the hierarchical naming in CCN, starting from that name prefix, only one identical Interest can be generated to request a user's subsequent tweets. As CCN is designed, every received Data message corresponds to an Interest message; thus, for receiving the next messages, the application should send a new Interest after reception of the Data message. It is the responsibility of the application to define its own strategy in terms of Interest transmission, in order to cope with updated information, user mobility, and so on. As CCN routers may keep the initial Interests longer in the PIT, another advantage of the CCN solution would be a possible removal of the current 90 s polling from the

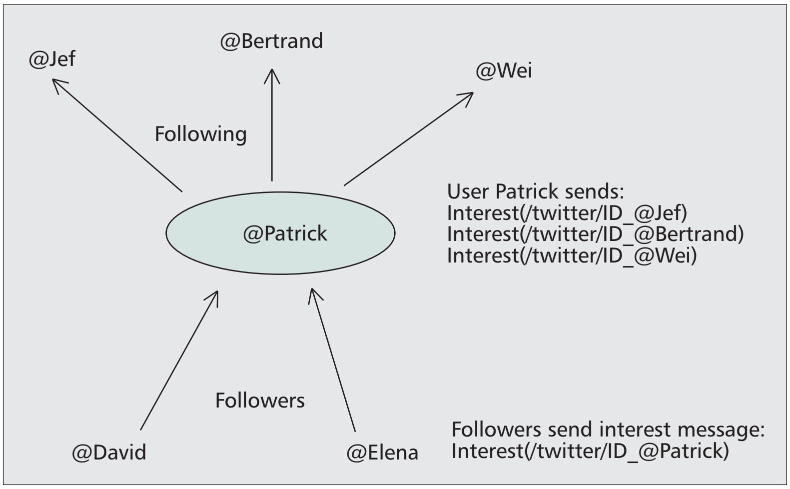

Figure 4. *Users' requests for receiving tweets.*

clients to check for updates on tweets of people they follow.

This new CCN architecture, with the native in-network functions, has many advantages for the applications. With the native caching feature in CCN, all the tweets the user can write will additionally be disseminated in different caching locations in the network as soon as her/his followers start to update their Twitter timeline.

Through its native multicast capability, the CCN design would also allow users to create and manage groups of their contacts, which is a feature currently missing in Twitter. (Contrary to most OSNs such as Facebook or Google+, Twitter does not currently allow sending private tweets within a restricted circle of followers.)

EVALUATION RESULTS

Using the CCN-based architecture for tweet delivery, we can intuitively imagine that the network load can be largely reduced depending on the numbers of followers/friends for a given content. We then performed some evaluations to prove the interest of a CCN delivery, compared to the current network architectures (as deployed today), depending on the number of users of the OSN application.

In our simulations, we took the example of the Twitter application and users in France connected to the Orange domestic network. In our assumption, the Twitter server is outside of the Orange domestic network. We also considered the use of a content delivery network (CDN) with one server in the United States and one at the peering point with Orange. To compare with the ICN solution, we suppose the CDN will not deliver only heavy contents such as videos and photos as it is used now but also tweets. To be effective the servers need to have a lot of contents; this prevents having CDN nodes close to end users. In the ICN solution, the community created between the followed and the followers is automatically handled in the router's tables, whereas in the case of a CDN, it has to be provisioned by hand or using some defined choices, hence creating a trade-off between a poor hit ratio if the servers are close to end users and the need for huge databases if they are in the core

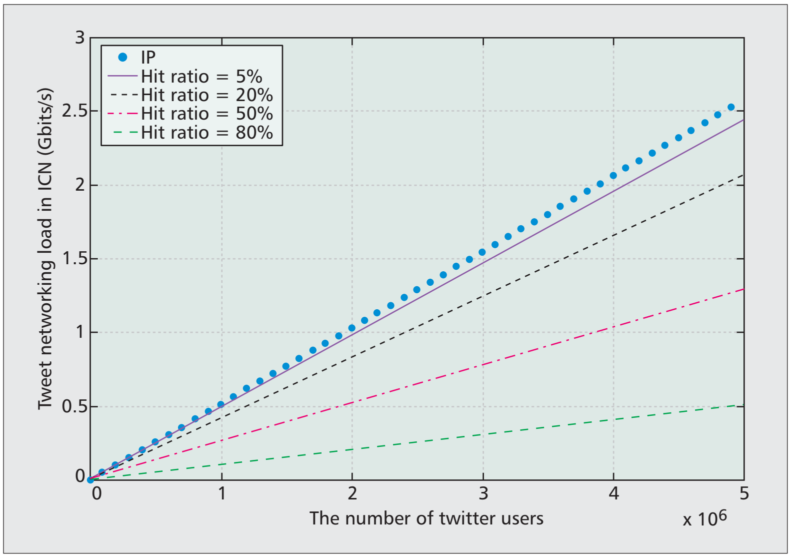

Figure 5. *Comparison of traffic load.*

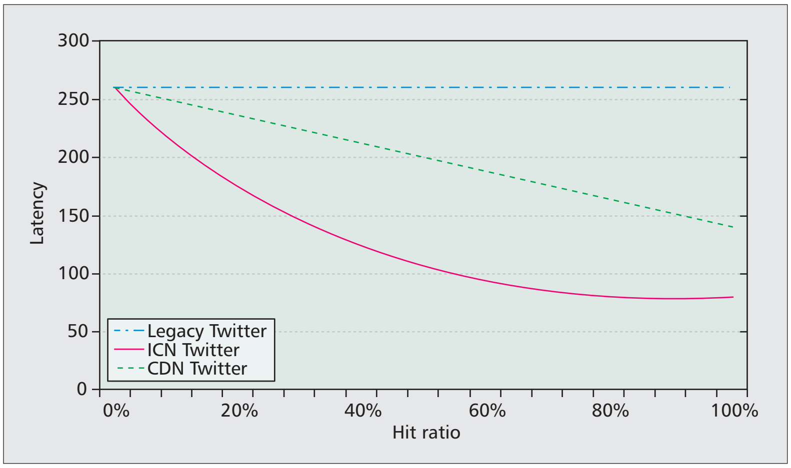

Figure 6. *Comparison of latency.*

network. For the study we used a two-level CDN architecture, with contents stored in the United States for the first level and at the peering points for the second level. The latency to reach those nodes would be an intermediate value between what it would be to reach servers in France and to reach servers in the United States. For the operator network topology, we assumed a hierarchical three-level network topology, with caching facility at each level for the CCN architecture.

The parameters we used to define the Twitter service behavior are mainly taken from [12, 13]. Typically, we took an average length of 120 characters for each tweet, an average number of tweets sent by users at 0.97 per day, the number of Twitter users in France at about 3 million (in our evaluations, the number of clients varied between 0 and 5 million). We took the number of followers for a given group, following a curve similar to a power law distribution (as analyzed in [14]). For the CCN behavior, we do not think it is realistic that ICN has the ability to cache (e.g., store content of CCN nodes) and deliver 100 percent of Twitter content, so we make different evaluations having different cache hit ratios, from low-efficiency caching systems (e.g., 5 percent hit ratio) up to an efficient system (80 percent of data provided by the ICN caches).

The current 3 million French Twitter users generate around 1250 tweets/s. The average number of characters each tweet has is around 120; thus, every second there is around 1.2 Mb traffic introduced by Twitter users. Among all tweet messages, 3 percent contain a video or an image link. A *middle-quality* video is usually coded as 512 kb/s. Having an average 2 min duration for one video, it gives about 60 Mb of traffic. A compressed image is usually 50 kbytes. Tweets that contain videos or images are usually more popular than pure-text tweets (e.g., private messages). If we take into account this behavior, 1250 tweets/s corresponds to about 1.13 Gbs/s networking load. Having an ICN network to deliver such traffic, a low hit ratio in ICN caches does not really help to save bandwidth; but as the hit ratio grows, the volume of data in the network greatly decreases. For instance, with 50 percent hit ratio, we can save half of the traffic in the network, that is, about 0.55 Gbs/s of networking load (Fig. 5). This saving is just for one application (Twitter); it will increase similarly as other applications are delivered in such a way. Having an ICN solution then helps to greatly reduce the traffic in the domestic network for the operator as well as at the peering point, and also consequently the cost.

We also estimated the latency between the end users and the provider of content in the three network architectures depicted in Fig. 3. Figure 6 shows the latency for a given hit ratio of the following use cases:

- Legacy Twitter represents the latency measured from France to the Twitter servers in the United States to get the content.
- CDN Twitter is the latency to get the content with a two-level CDN, with the request being handled by either the CDN servers operating for Twitter (e.g., Akamai) in the case of a cache hit or the Twitter server in the case of a cache miss.
- ICN Twitter is for the latency we get with the ICN network in France, being a three-level hierarchical network with CCN nodes on all three levels.

The ICN Twitter architecture gives the best results, showing that caching near customers is the most efficient even with a low hit ratio. Depending on where the content is cached in the network tree, the latency can be reduced from 20 to 60 percent (hit ratio of 50 percent).

Conclusion

Resulting from our evaluations, using ICN for social networking applications is advantageous for the network operator as expected but also for the end users QoE (Quality of Experience) since the content is closer to end-users and can be delivered more rapidly. Furthermore the social links derived from the OSN have a direct influence on the network delivery efficiency and perfectly fit with ICN features, such as the native multicast-like function.

The ICN paradigm is then a promising solution for social networking applications. In our future work, we plan to develop/adapt an OSN application on top of the CCN solution, to better suit native ICN functions and make large-scale tests on a real network (e.g., the PlanetLab environment) to compare with the current IP infrastructure. We also plan to take into account OSN social links in the ICN network, in order to optimize the routing and delivery of contents.

References

[1] V. Jacobson *et al.*, "Networking Named Content," *Proc. ACM CoNEXT 2009*, Dec. 2009.

[2] "Cisco Visual Networking Index: Forecast and Methodology, 2010–2015," http://tinyurl.com/3p7v28.

[3] "The Law of Sharing," http://blog.sharethis.com/2011/07/07/the-law-ofsharing/.

[4] "Youtube — Share and Share Alike: We've Acquired Flick," http://youtubeglobal. blogspot.com/2011/01/share-and-share-like-weve-acquired.html.

[5] S. Agarwal and S. Agarwal, "Social Networks as Internet Barometers for Optimizing Content Delivery Networks," *Proc. 2009 IEEE 3rd Int'l. Symp. Advanced Networks and Telecommunication Systems (ANTS)*, Dec. 2009.

[6] S. Scellato *et al.*, "Track Globally, Deliver Locally: Improving Content Delivery Networks by Tracking Geographic Social Cascades," *Proc. WWW 2011*, Mar. 2011.

[7] F. Nazir *et al.*, "Time Critical Content Delivery Using Predictable Patterns in Mobile Social Networks," *Proc. IEEE Computational Science and Engineering 2009*, Aug. 2009.

[8] B. Ahlgren et al., "A Survey of Information-Centric Networking (draft)," Information-Centric Networking, no. 10492, 2011, available: http://drops.dagstuhl.de/opus/volltexte/2011/2941.

[9] L. Zhang *et al.*, "Named Data Networking (NDN) Project," Oct. 2010, available: http://www.named-data.net/ndn-proj.pdf.

[10] I. Psaras *et al.*, "Modelling and Evaluation of CCN-Caching Trees," Networking 2011, ser. Lecture Notes in Computer Science, vol. 6640, Springer Berlin/Heidelberg, 2011, pp. 78–91.

[11] G. Carofiglio *et al.*, "Modeling Data Transfer in Content Centric Networking," *Proc. 23rd Int'l. Teletrafic Congress (ITC)*, 2011.

[12] "Twitter Finally Reveals All Its Secret Stats," http://www.businessinsider.com/twitter-stats-2010-4.

[13] "State of the Twittersphere," http://blog.hubspot.com/Portals/249/sotwitter09.pdf.

[14] M. Cha *et al.*, "Measuring User Influence in Twitter: the Million Follower Fallacy," *Proc. 4th Int'l. AAAI Conf. Weblogs and Social (ICWSM 10)*, May 2010.

Biographies

Bertrand Mathieu [SM] (bertrand2.mathieu@orange.com) has been a senior researcher at France Telecom, Orange Labs since 1994. He received a Diploma of Engineering in Toulon, an M.Sc. degree from the University of Marseille, and a Ph.D. degree from the University Pierre et Marie Curie, Paris. His research activities are related to dynamic overlay networks, P2P networks, and information-centric networking. He has contributed to several national and European projects, and published more than 30 papers in international conferences, journals, and books. He is a member of several conferences' Technical Program Committees and an SEE Senior Member.

Wei You (wei.you@orange.com) is a Ph.D. student with Orange Labs and Telecom Bretagne. He obtained his Bachelor's degree at XiDian University in Xi'an, China, and his Diploma of Engineering in Brest, France. His thesis research is about ICN networking, and more specifically on the performance and quality of service in CCN networks.

Patrick Truong (patrick.truong@orange.com) joined France Telecom, Orange Labs in 2004 after getting his engineering degree in telecommunications from Ecole Nationale Supérieure des Télécommunications in Paris. Since 2004 he has worked as a security engineer on network intrusion detection based on traffic analysis and in the data streaming domain applied to network security to get his Ph.D. degree from the University Pierre et Marie Curie in 2010. Since 2010, he is focusing his research activity on information-centric networking with a contribution to the European SAIL project.

Jean-François Peltier (jeanfrancois.peltier@orange.com) is a senior researcher at Orange since 2005. He graduated from ESIEE Paris in 1988. He started his career as a network software engineer for Alcatel-Thomson and then developed real-time embedded software at Lucent Technologies. He has been responsible for system tests of DWDM equipment in Algety-Corvis in Lannion before joining Orange Labs. He has been evaluating CDN solutions and WebTV service. He is now working on information-centric networking, focusing on search, naming, and name resolution.