

IMPACT OF CONTENT-CENTRIC NETWORKING ON LARGE-SCALE SCIENTIFIC APPLICATIONS

Hiroyuki Ohsaki

Department of Information Networking
Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
E-mail: oosaki@ist.osaka-u.ac.jp

ABSTRACT

Data-centric networking, which takes the data transferred in a network rather than the point-to-point communication between hosts as the first-class, is recently getting increased attention [?, ?, ?]. One of notable features of data-centric networking is reusing data at a network-level by transferring data based on the identifier assigned to every content. In data-centric networking, a unique identifier is assigned to every content. In IP network, an identifier (i.e., IP address) is assigned to every host. On the other hand, in data-centric network, an identifier (i.e., content identifier) is assigned to every content. Since a content is transferred based on its identifier in data-centric networking, contents that have the same identifier are treated as the same wherever the contents may exist. Therefore, in data-centric networking, a distribution of contents like CDN (Contents Delivery Network) [?] is possible, which reduces the data transfer delay as well as the amount of traffic transferred in a network.

1. INTRODUCTION

Data-centric networking, which takes the data transferred in a network rather than the point-to-point communication between hosts as the first-class, is recently getting increased attention [?, ?, ?]. One of notable features of data-centric networking is reusing data at a network-level by transferring data based on the identifier assigned to every content. In data-centric networking, a unique identifier is assigned to every content. In IP network, an identifier (i.e., IP address) is assigned to every host. On the other hand, in data-centric network, an identifier (i.e., content identifier) is assigned to every content. Since a content is transferred based on its identifier in data-centric networking, contents that have the same identifier are treated as the same wherever the contents may exist. Therefore, in data-centric networking, a distribution of contents like CDN (Contents Delivery Network) [?] is possible, which reduces the data transfer delay as well as the amount of traffic transferred in a network.

Data-centric networking has been actively studied in the

literature, and there are several data-centric network architecture proposals such as DONA(Data-Oriented Network Architecture) [?], CCN (Content-Centric networking) [?], and NDN (Named-Data Networking) [?].

A representative example of data-centric networking is CCN (Content-Centric Networking) [?], which routes packets within a network based on their content identifiers. CCN adopts a request-and-response communication model. CCN routes a request packet called *Interest packet* from a user to sources by longest-prefix matching of the identifier of the requested content. If the Interest packet is successfully delivered to the source, the content packet called *Data packet* is sent back to the user by reversely traversing the path of the Interest packet.

From an application viewpoint, CCN can be seen as a network-level realization of a distributed (*key, value*)-database. Namely, every content is stored in the network in the form of (*content identifier, content*). An application requests a content named as *content identifier* to the network. The CCN network looks for the corresponding content in the network, and it then delivers the found content to the application.

CCN has therefore high affinity with applications, which can be implemented with a distributed (key, value)-databased. Hence, it is expected that those data-intensive applications can be easily and efficiently realized with CCN. CCN solves the drawbacks of the conventional IP network, and it provides availability, security, and location independence to applications [?]. Applications running on CCN should be benefited those advantages of CCN.

There have been a number of architectural studies of CCN in the literature. However, the impact of CCN on several types of network-based applications has not been well understood. For instance, there have been several discussions on CCN applications such as data sharing and voice over CCN. However, the implication of CCN on, in particular, large-scale applications such as scientific computing has not been understood. Also, design methodology of large-scale applications on CCN has been left almost untouched.

In this paper, we therefore discuss the impact of data-

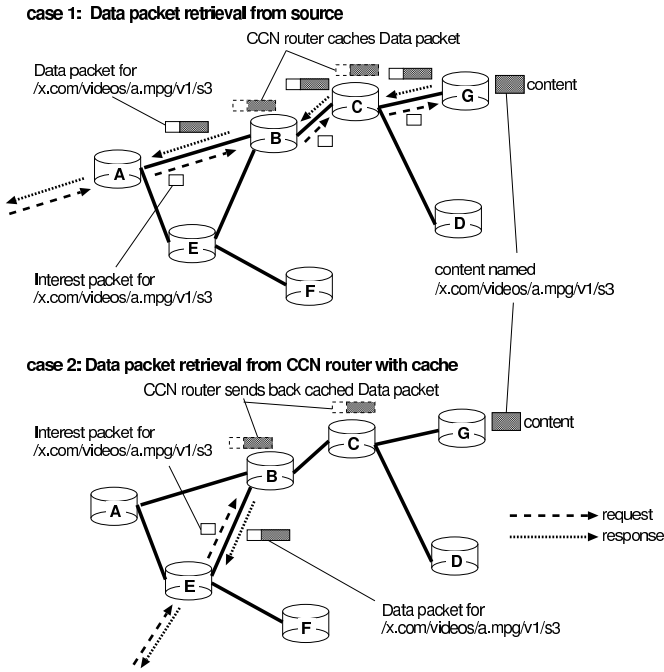


Fig. 1: Routing of Interest and Data packets in CCN.

centric networking, on large-scale scientific applications. First, focusing on CCN as a representative data-centric networking, we discuss advantages and disadvantages of CCN for large-scale scientific applications. We also discuss practical issues and challenges for running large-scale scientific applications on CCN. Second, we present two possible large-scale scientific applications (i.e., a remote procedure call and a distributed filesystem), which could be benefited the advantages of CCN.

The rest of this paper is organized as follows. In Section 2, we explain the overview of content delivery in CCN. In Section ??, we explain DCC, which improves the efficiency of Content Store usage. In Section ??, we evaluate the effectiveness of DCC by simulations. Finally, in Section ??, we conclude this paper and discuss future works.

2. CCN (CONTENT-CENTRIC NETWORKING)

A representative example of data-centric networking is CCN (Content-Centric Networking). CCN adopts a request-and-response communication model. CCN routes an Interest packet from a user to sources by longest-prefix matching of the identifier of the requested content. If the Interest packet is successfully delivered to the source, the content is sent back to the user by reversely traversing the path of the Interest packet (see Fig. 1).

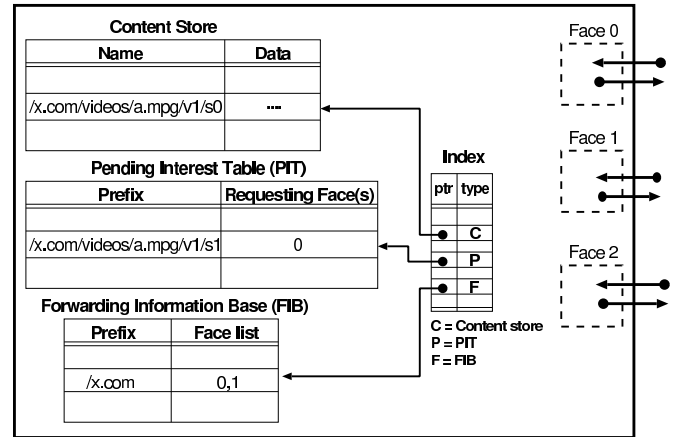


Fig. 2: The structure of CCN router [?]

2.1. CCN router structure

A CCN router forwards a packet through a *face*, which corresponds to the interface in an IP router. The CCN router usually has multiple faces. The CCN router forwards an arriving Interest packet through one or more faces, which are expected to be directed to sources.

A CCN router forwards packets using three data structures: *Forwarding Information Base (FIB)*, *Content Store*, and *Pending Interest Table (PIT)* (see Fig. 2). Forwarding Information Base (FIB) corresponds to the routing table of an IP router. FIB is used to forward the received Interest packet toward the potential sources of the requested content. Each FIB entry is composed of the prefix of a content identifier and one or more faces through which the Interest packet should be sent. Content Store is a buffer memory for caching Data packets that arrived at the CCN router. Each entry of Content Store has a content identifier and the Data packet. Pending Interest Table (PIT) is a table for recording unresolved Interest packets. PIT is used to avoid duplicate transmission of Interest packets in the network, and to determine the face through which Data packets should be sent back. Since PIT is soft-state, each entry of PIT is automatically removed after a certain period of time.

2.2. Routing of Interest and Data packets

In what follows, we explain how CCN routes Interest packets and Data packets using these three data structures.

Basically, Interest packets are routed toward the potential sources based on longest-prefix matching of the content identifier with FIB entries.

Using PIT, the Data packet is sent back to the user by reversely traversing the path of the Interest packet. Each entry of PIT works as a trail of *breadcrumb*, and the Data packet is

sent back by reversely following the PIT in all routers along the path. Each CCN router along the path stores the Data packet in its Content Store, and deletes the corresponding entry from its PIT.

On the contrary, if a Data packet corresponding to the Interest packet was found in Content Store of a CCN router, the CCN router discards the Interest packet and sends the Data packet in Content Store back. This realizes reuse of contents at a network-level.

If the requested content is recorded in PIT of a CCN router, the CCN router discards the Interest packet. Namely, CCN avoids duplicate transmission of the Interest packet.

3. IMPLICATIONS OF CCN TO LARGE-SCALE SCIENTIFIC APPLICATIONS

3.1. Advantages

CCN 上の大規模科学技術アプリケーションを、CCN の持つ以下のような利点を活かすことができるだろう。

- 高可用性

CCN は、CDN のようなコンテンツ配送を行うため、コンテンツの高可用性が実現される。CCN では、自然な形で、同一のコンテンツの複製をネットワーク中に配置することが可能であり、コンテンツの複製の少なくとも一つがアクセス可能であればコンテンツを取得することができる。さらに、CCN ルータがコンテンツをキャッシュするため、コンテンツのリポジトリが一時的に故障した場合でもコンテンツを取得することができる。

CCN 上で動作するアプリケーションは、ネットワークレベルで提供されるコンテンツの高可用性のメリットを享受できる。IP ネットワーク上のアプリケーションでは、従来、アプリケーションレベルでデータもしくはサービスの可用性を高める工夫が不可欠であった。

ネットワーク中に存在するコンテンツ複製数を M 、利用者からリポジトリまでの平均ホップ数を H 、各ノード (ルータおよびリポジトリ) の故障率を等しく λ 、各 CCN ルータにおけるキャッシュのヒット率を p とする。簡単のため、平均ホップ数が 0 より大きい整数の場合を考える。コンテンツの複製、ルータ、リポジトリがそれぞれすべて独立であると仮定すれば、コンテンツの可用性 A は次式で与えられる。

$$A = 1 - (1 - a)^M$$

ここで a はあるコンテンツの複製の可用性であり、次式で与えられる。

$$\begin{aligned} a &= \sum_{i=1}^{H-1} (1 - \lambda)^i (1 - p)^{i-1} p + (1 - \lambda)^H (1 - p)^{H-1} \\ &= \frac{\lambda (1 - \lambda)^H (1 - p)^H + p (1 - \lambda)}{\lambda + p (1 - \lambda)} \end{aligned}$$

- 高効率性

CCN では、ルータがネットワーク内でコンテンツのキャッシュを行うため、コンテンツの配送遅延が短縮されるとともに、ネットワーク中を転送されるトラフィック量の削減が期待できる。CCN では、すべてのコンテンツは、それぞれのコンテンツに固有の識別子で区別されるため、ネットワークレベルでのコンテンツの再利用が行われる。CCN ルータは、要求パケットに対応するデータパケットをコンテンツストア内にキャッシュしていれば、要求パケットをそれ以上中継することとなるデータパケットを返送する。

CCN 上で動作するアプリケーションは、ネットワーク中でコンテンツのキャッシュが行われるため、小さなコンテンツ配送遅延と高いスループットの実現が期待できる。

ノード間の平均遅延を τ とする。CCN ルータにおける処理遅延およびキューイング遅延が無視できるほど小さいと仮定すれば、コンテンツ配送遅延 D は次式で与えられる。

$$\begin{aligned} D &= 2 \left(\sum_{i=1}^{H-1} i (1 - p)^{i-1} p + H (1 - p)^{H-1} \right) \\ &= \frac{2(1 - (1 - p)^H)}{p} \end{aligned}$$

ルータがキャッシュを行わない場合のコンテンツ配送遅延が $2H$ である。ルータがキャッシュを行わない場合の、ネットワーク全体のスループットを \tilde{T} とする。リトルの定理より、CCN のスループット T は次式で与えられる。

$$\begin{aligned} T &= \tilde{T} \frac{H}{D} \\ &= \frac{p H \tilde{T}}{1 - (1 - p)^H} \end{aligned}$$

3.2. Disadvantages

CCN 上の大規模科学技術アプリケーションを、CCN の持つ以下のような欠点の影響を受けるだろう。

- アプリケーション再設計の必要

CCN は、従来の IP ネットワークとはまったく異なるネットワークアーキテクチャであるため、CCN の API は、IP のそれとはまったく異なる。このため、IP ネットワーク上で動作しているアプリケーションを、そのまま単純に CCN 上に動作させることは容易ではない。例えば、従来のネットワークアプリケーションが利用している BSD ソケット API はホスト中心のネットワークアーキテクチャを前提としており、データ中心のネットワークのための新しいネットワーク API が設計される必要がある。

CCN上で、大規模科学技術アプリケーションを動作させるためには、CCNの特性を活かすようなアプリケーションを設計する必要がある。IPからCCNへの移行には相応のコストが必要であろう。

- ネットワーク処理の複雑さ

CCNは、IPネットワークよりも複雑なネットワークアーキテクチャである。例えば、IPルータは、到着したパケットをそれぞれ独立に処理するステートレスな処理を行うが、CCNルータはPITテーブルを用いたステートフルな処理を行う。

CCNのスケーラビリティをどのように実現するかは今後の課題である。現在、CCNxのような、アプリケーションレベルのCCN実装は存在し、小規模なCCNネットワークの実証実験はすでに行われている。しかし、CCNネットワークをさらに大規模化した時に、どのような問題が発生するか、それらの問題をどのように解決すれば良いかは今後の研究が必要である。

4. EXAMPLE APPLICATIONS

4.1. CCN-based remote procedure call

CCNを用いて、軽量で、準最適な、大規模並列計算のためのリモートプロシージャコールの実現が可能であろう。

1章で述べたように、アプリケーションの視点からは、CCNは、ネットワークレベルで実現された(key, value)型の分散データベースと見ることができる。プロシージャ名と引数をkeyとし、プロシージャの実行結果をvalueとみなす。アプリケーションは、プロシージャ名と引数をコンテンツ識別子に埋め込んで要求パケットを送出する。CCNは、コンテンツ識別子のプロシージャ名をもとに要求パケットをリモートサーバにルーティングする。要求パケットを受信したリモートサーバは、プロシージャの実行結果をデータパケットとしてアプリケーションに返送する。このような方法により、リモートプロシージャコールを非常に自然な形で実現できる。

第一に、CCN上でリモートプロシージャコールを実現する場合は、CCN自体が持つ高可用性と位置独立性が非常に大きなメリットとなる。つまり、CCN上でリモートプロシージャコールを実現する場合には、ホスト中心(リモートプロシージャを提供するリモートサーバ)ではなく、サービス中心(いずれかのリモートサーバ上で実行されるプロシージャ)となる。つまり、IPネットワーク上でリモートプロシージャコールを実現する時のように、プロシージャを実行するリモートサーバを指定する必要はなく、単にプロシージャ名を要求パケットに指定するだけでよい。アプリケーションがサービス名を指定して要求パケットを送出すれば、CCNネットワークがサービスを提供する(コンテンツ識別子に対応する資源を保有する)リモートサーバを発見する。つまり、複製されたコンテンツを保有する複数のリポジトリが、同一のサービスを提供する複数のリモートサーバに対応する。リモートプロ

シージャコールでは、通常、あるプロシージャを実行するリモートサーバを一つ発見すれば十分である。

第二に、CCNによるネットワーク中のコンテンツキャッシングにより、リモートプロシージャコールのキャッシングが自動的に実現できる。前述のように、プロシージャ名と引数をCCNのコンテンツ識別子に埋め込む場合、このプロシージャの実行結果は、データパケットとして経路上のCCNルータにキャッシュされる。従って、同一のプロシージャ名および引数に対するリモートプロシージャコールが発生した場合、リモートサーバに要求パケットを届ける必要はなく、CCNによって、CCNルータにキャッシュされているプロシージャの実行結果が返送される。

4.2. CCN-based distributed filesystem

CCNを用いて、高信頼で、高効率な、大規模分散ファイルシステムの実現が可能であろう。

1章で述べたように、アプリケーションの視点からは、CCNは、ネットワークレベルで実現された(key, value)型の分散データベースと見ることができる。ファイルのパス名をkeyとし、ファイルの実体もしくはディレクトリエントリをvalueとみなす。アプリケーションは、ファイルのパス名をコンテンツ識別子に埋め込んで要求パケットを送出する。CCNは、コンテンツ識別子をもとに要求パケットをリポジトリ(分散ファイルシステム)にルーティングする。要求パケットを受信したリポジトリは、ファイルの実体をデータパケットとしてアプリケーションに返送する。このような方法により、分散ファイルシステムを簡単に実現できる。

第一に、リモートプロシージャコールの場合と同様に、CCN上で分散ファイルシステムを実現する場合は、CCN自体が持つ高可用性と位置独立性が非常に大きなメリットとなる。つまり、CCN上で分散ファイルシステムを実現する場合には、分散ファイルシステムを構成する個々のノードの台数や物理的な位置を考慮する必要がない。つまり、IPネットワーク上で分散ファイルシステムを実現する時のように、各ノードが保有しているファイルの集合を管理する必要はなく、単にファイルのパス名を要求パケットに指定するだけでよい。

第二に、これもリモートプロシージャコールの場合と同様に、ネットワーク中でのファイルのキャッシングが自動的に実現できる。あるファイルのパス名を要求パケットによって要求した場合、対応するファイルの実体は、データパケットとして経路上のCCNルータにキャッシュされる。従って、ファイルのパス名に対するアクセスが発生した場合、CCNルータにキャッシュされているファイルの実体が返送される。

ただし、CCN上で分散ファイルシステムを実現するためには、ファイルの揮発性をどのように扱うかを考える必要がある。分散ファイルシステムでは、一般に、あるパス名に対応するファイルの実体は変更される可能性がある。単純に、ファイルのパス名をコンテンツ識別子にマッピングする方法では、ファイル(コンテンツ)の複製の間で不整合が発生してしまう。この問題への解決法として、例え

ば、(1) 不揮発性の分散ファイルシステム、(2) インダイレクトな分散ファイルシステム、の二種類のアプローチがある。

不揮発性の分散ファイルシステムでは、いったん生成されたファイルに対する更新を許可しないことにより、ファイルのパス名とファイルの実体との不整合性を解消する。つまり、ファイルは生成もしくは削除のみが可能であり、あるパス名に対応するファイルの実体は不変とする。分散ファイルシステムとしての利便性は低下するが、実現が容易であり、データセントリックネットワークの利点である可用性・効率性のメリットを最大限に活かすことができる。

インダイレクトな分散ファイルシステムでは、ファイルのパス名を、直接ファイルの実体にマッピングするのではなく、ファイルの実体のコンテンツ識別子を表すメタコンテンツにマッピングする。つまり、ファイルの実体の変更されるごとに、それぞれのファイルに異なるコンテンツ識別子を割り当てる。ファイルのパス名に対応するコンテンツは、ファイルの実体を表すコンテンツ識別子とする。このように、ファイルのパス名から、ファイルの実体までの解決を二段階にすることにより、非永続的なファイルを柔軟に扱うことが可能となる。当然のことながら、メタコンテンツはネットワーク中でキャッシュすることはできない。

5. CONCLUSION

been clarified or understood

For instance, data-centric networking could...

- Simplify the /design and implementation/ of scientific applications - Improve both /computing and networking efficiency/ of scientific applications - Improve /scalability/ of scientific applications because of simplicity and efficiency - Improve /availability/ of scientific applications under non-negligible failures