University of Glasgow

VIA VERITAS VITA

Call me from Anywhere

Remote Procedure Calling meets *Named Data Networking*

Jeremy.Singer@glasgow.ac.uk

what is a remote procedure call?

<networking, programming>

(RPC) A protocol which allows a program running on one host to cause code to be executed on another host without the programmer needing to explicitly code for this.

Java RMI example

```
import java.rmi.Naming;

public class RmiClient {
    public static void main(String args[]) throws Exception {
        RmiServerIntf server = (RmiServerIntf)Naming.lookup("//localhost
/RmiServer");
        System.out.println(server.getMessage());
    }
}
```

RESTful web service example

```
response = requests.get (
'https://samoa.dcs.gla.ac.uk/events/rest/Event/recentlychanged'
)
```

Summary of state-of-the-art RPC

- specify ***endpoint***
- specify ***function***
- package up parameters and send
- wait for result
- unpackage result and continue execution

What is NDN?

- refer to data by **name**, rather than **location**
- data *sinks* request data with **interest packets**
- data *sources* respond with **data packets**

alternative definition

*the internet becomes a gigantic distributed **key/value** store*

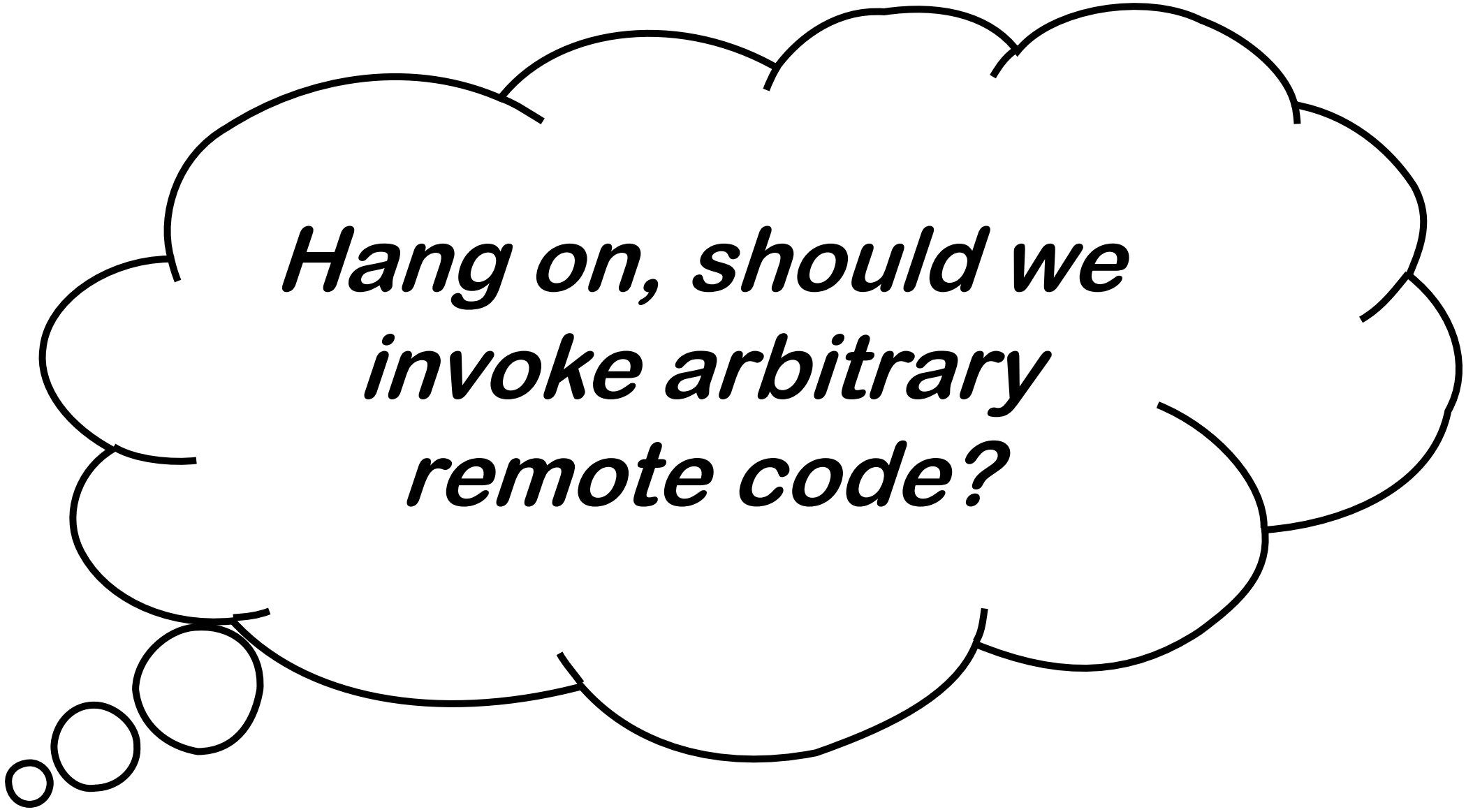
How can we layer RPC over NDN?

- just specify function **name** - no need for endpoint
- add **params** (package as part of function call) and make NDN request
- remote key **lookup**
- value found … function is **invoked** (somewhere?)
- **result** returned to caller
- potential **caching** of pure function invocation results

In Python?

- a *dynamic* language - so no need for types of params or return value
- although **`mypy`** does allow types, which could be helpful

- possible calling models
  - direct - any name not known locally could be looked up remotely
  - via a specific NDN binding - a library that does the lookup for us (more intentional)


Hang on, should we
invoke arbitrary
remote code?

invoking arbitrary remote code

possible mitigations …

- code signing - NDN can do some of this security for us?
- types of params / returns, maybe with asserts?
- static guarantees - e.g. like Dafny
- dynamic testing - e.g. like QuickCheck
- code sandboxing
- redundancy with N-versioning

NFaaS: Named Function as a Service

Michał Król
University College London
m.krol@ucl.ac.uk

Ioannis Psaras
University College London
i.psaras@ucl.ac.uk

ABSTRACT

In the past, the Information-centric networking (ICN) community has focused on issues mainly pertaining to
livery (*e.g.,* routing and forwarding scalabi
and in-network caching). However, to keep
architectural trends the wider area of futu
there is a pressing need to support edge/f
ments, where cloud functionality is availa
where the data is generated and needs pro

With this goal in mind, we propose *Nam*
(NFaaS), a framework that extends the Na
architecture to support in-network functio
to existing works, NFaaSbuilds on very ligh
for dynamic execution of custom code. F
loaded and run by any node in the networ
between nodes according to user demand
moving functions a first-class challenge. I
Store component, which is responsible no
tions, but also for making decisions on v
locally. NFaaSincludes a routing protocol a

functionality to be incorporated. Powerful end-user devices and new applications (*e.g.,* augmented reality [1]) demand minimum service delay, while the Internet of Things (IoT) [2] generates huge

RICE: Remote Method Invocation in ICN

Michał Król
UCL
m.krol@ucl.ac.uk

Karim Habak
Georgia Tech
karim.habak@gatech.edu

David Oran
Network Systems Research & Design
daveoran@orandom.net

Dirk Kutscher
Huawei
dirk.kutscher@huawei.com

Ioannis Psaras
UCL
i.psaras@ucl.ac.uk

ABSTRACT

Information Centric Networking has been proposed as a new network layer for the Internet, capable of encompassing the full range of networking facilities provided by the current IP architecture. In addition to the obvious content-fetching use cases which have been the subject of a large body of work, ICN has also shown promise as a substrate to effectively support remote computation, both pure functional programming (as exemplified by Named Function Networking) and more general remote invocation models such as RPC and web transactions. Providing a unified remote computation capability in ICN presents some unique challenges, among which are timer management, client authorization, and binding to state held by servers, while maintaining the advantages of ICN proto-

1 INTRODUCTION

Much of today's network traffic consists of data sent for processing to the cloud and web-servers exchanging high volumes of dynamically generated content. While today's ICN networks can deal efficiently with static data delivery, they have difficulty handling service/function invocation [24]. In view of these limitations, multiple works have recently tried to extend ICN's capabilities to deal with dynamic content.

Notable among these efforts, Named Function Networking (NFN) [29] and Named Function as a Service (NFaaS) [17] extend ICN's named data access model to a remote function invocation capability, enabling consumers to request the network to execute functions remotely. In NFN [29], for instance, function invocation corresponds

Next steps

- prototype implementation (Charles is working on this!)
- end-to-end evaluation on simple case study
- package up as a Python library or similar

https://github.com/Coffee-Powered/NDN/blob/main/prog_5/src/remote.cpp

```
Name name = interest.getName();
string content;

// Extract the proc name from name.
string proc = name.get(1).toUri();
cout << "Received request to call: " << proc << endl;


if(strcmp(proc.c_str(), "HelloWorld") == 0) {
    cout << "Calling " << proc << ".." << endl;
    hello_world();
    content = string("Procedure called.");
}
```

Next steps

- prototype implementation (Charles is working on this!)
- end-to-end evaluation on simple case study
- package up as a Python library or similar