UNIVERSITY OF SUSSEX

foundations of software systems

RaptorQ Codes

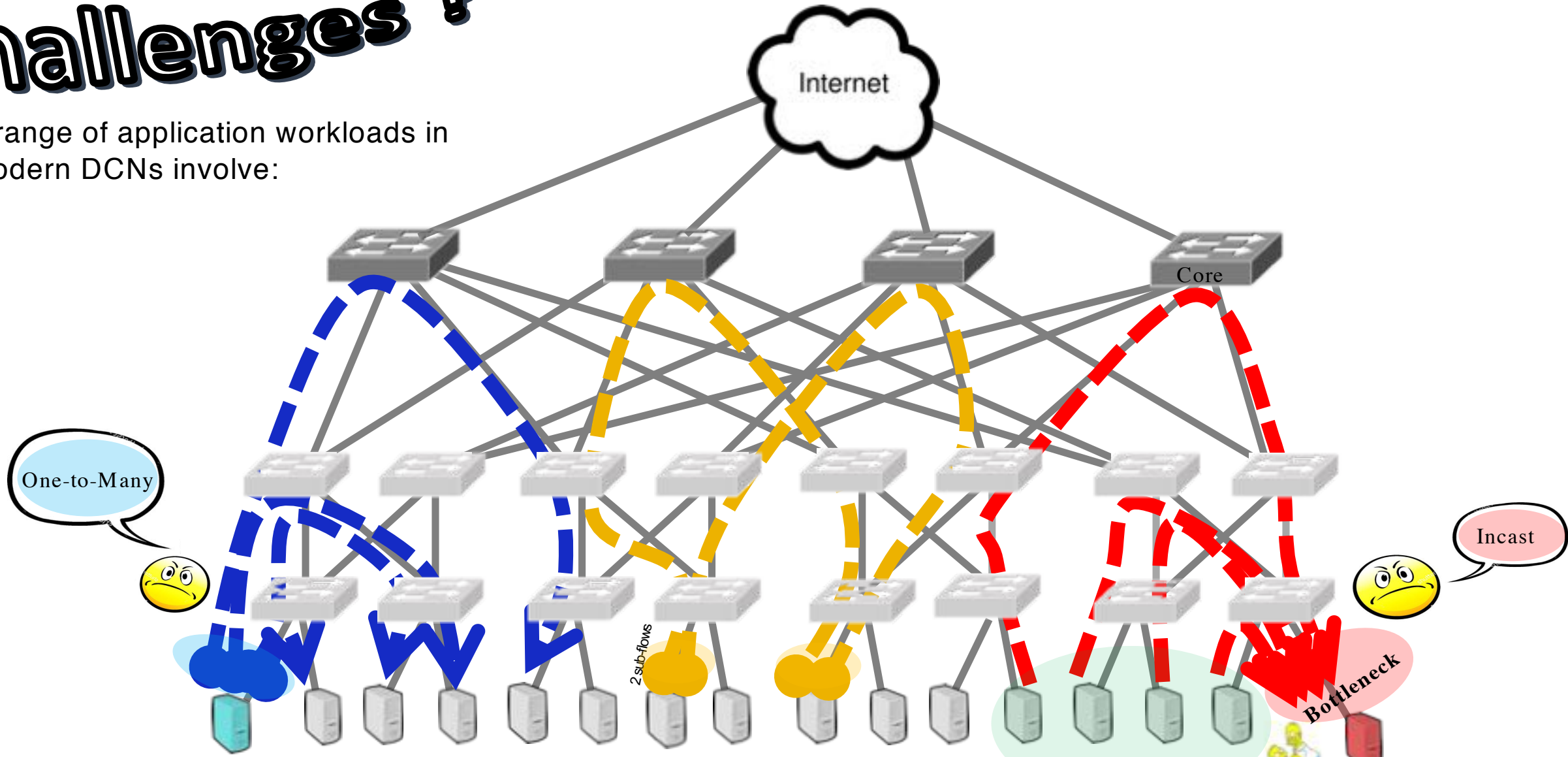
Polyraptor: Embracing Path and Data Redundancy in Data Centres for Efficient Data Transport

Mohammed Alasmar

A joint work with: George Parisis & Jon Crowcroft

Challenges ?

- A range of application workloads in modern DCNs involve:



✓ Replicating copies of data to multiple servers (when required) e.g., GFS

✓ Improving resource utilisation (per-packet Vs ~~per-flow ECMP~~)

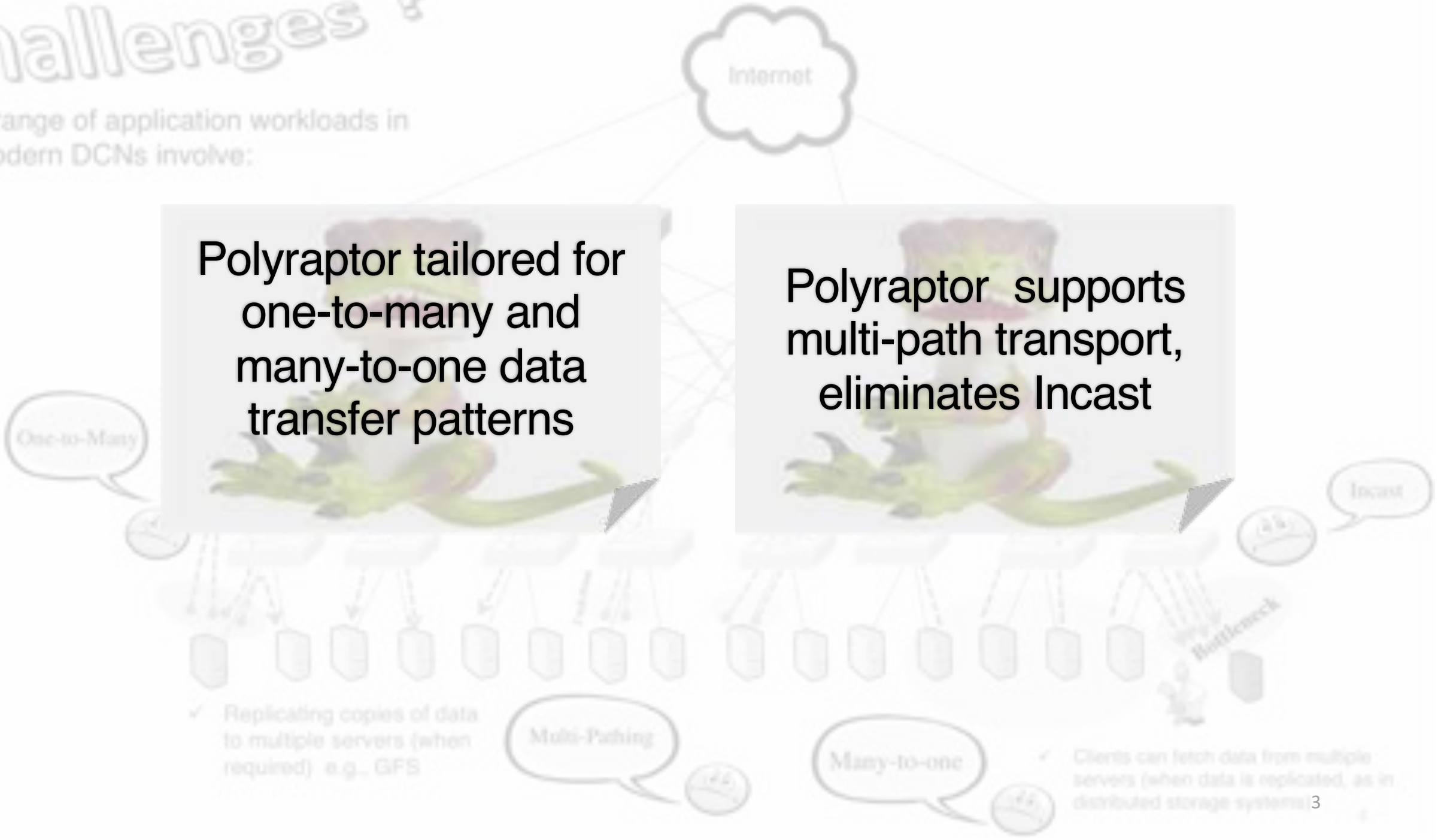
✓ Clients can fetch data from multiple servers (when data is replicated, as in distributed storage systems)

Challenges ?

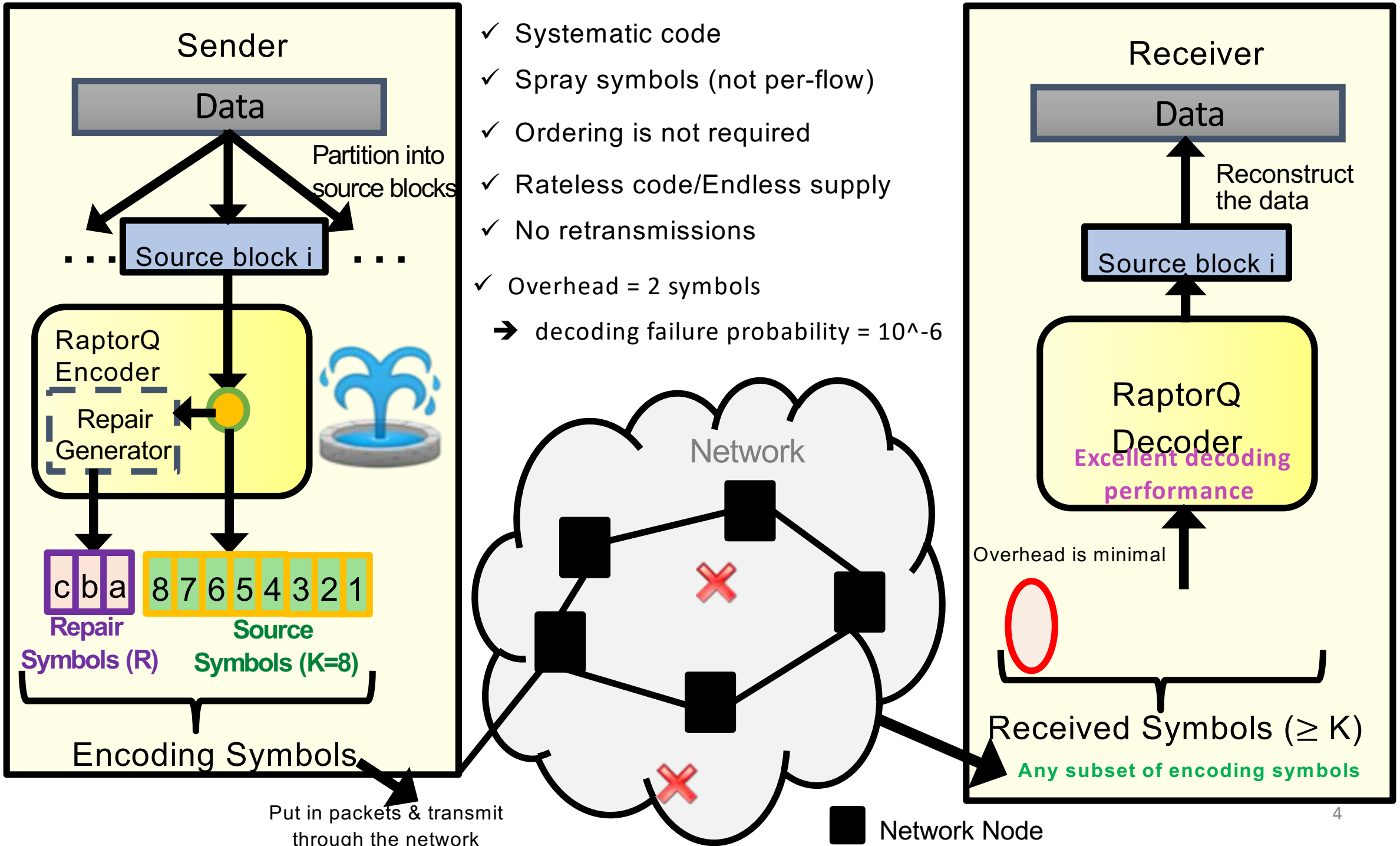
- A range of application workloads in modern DCNs involve:

Polyraptor tailored for one-to-many and many-to-one data transfer patterns

Polyraptor supports multi-path transport, eliminates Incast

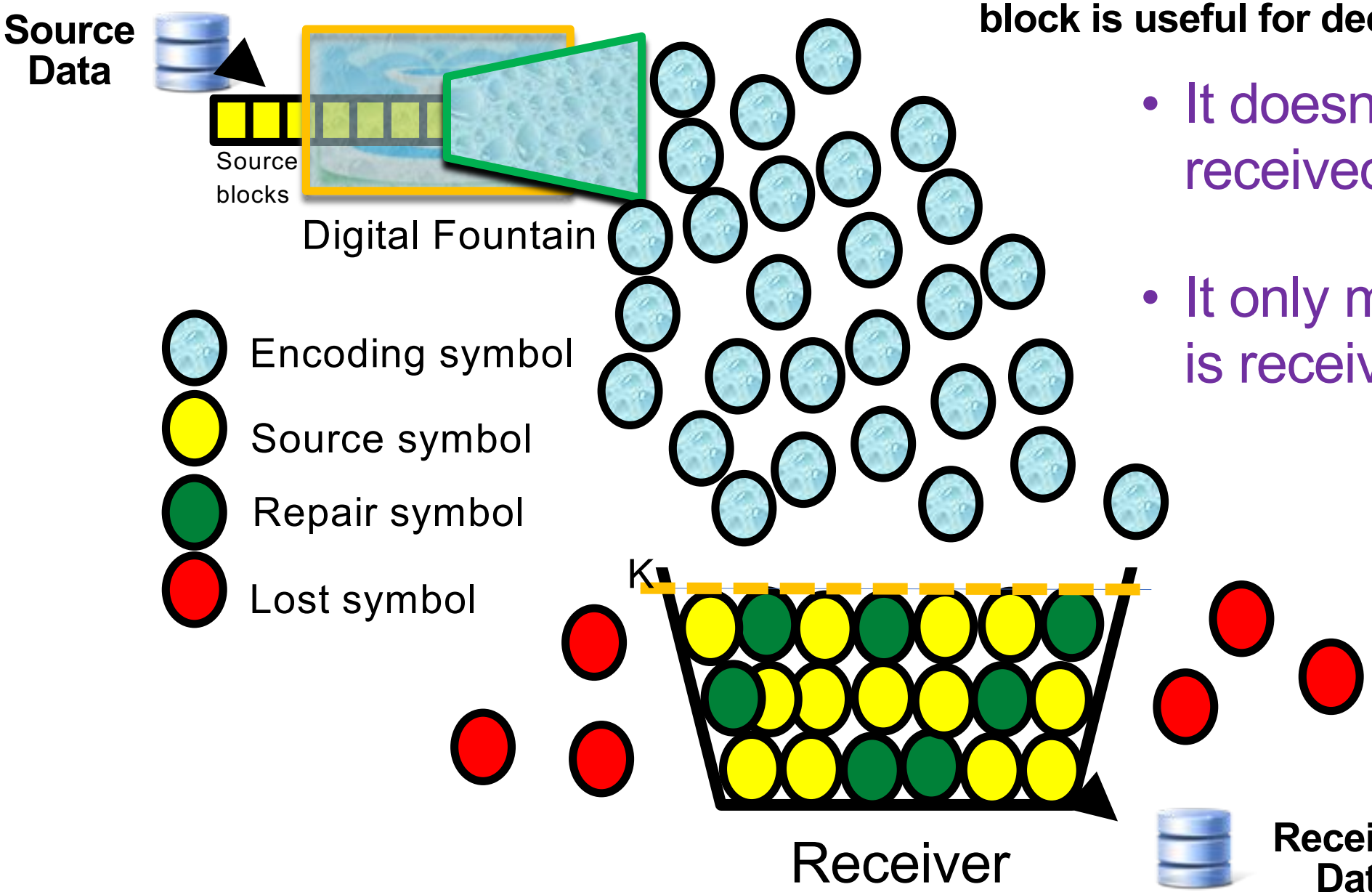


Fountain coding (RaptorQ)



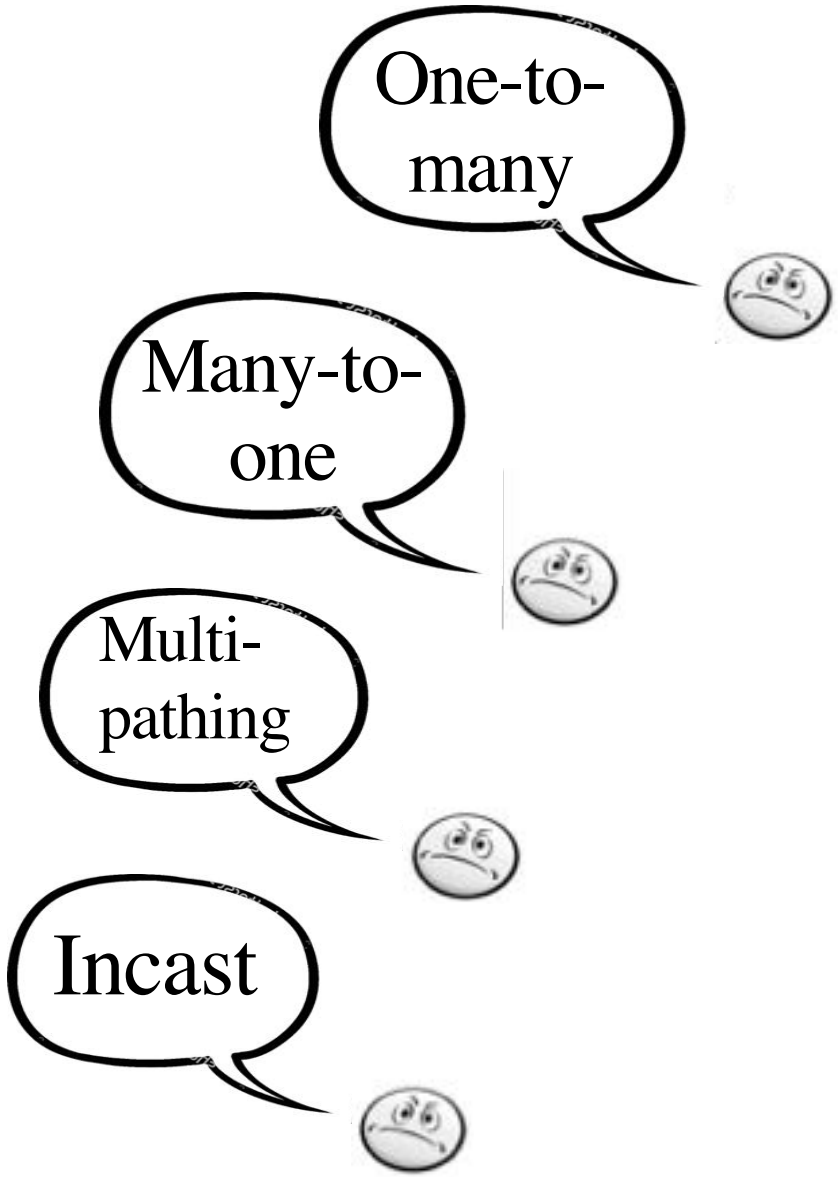
Fountain coding (RaptorQ)

- **Any subset of encoding symbols whose size is equal or slightly larger than the source block is useful for decoding**



- It doesn't matter what is received or lost
- It only matters that enough is received

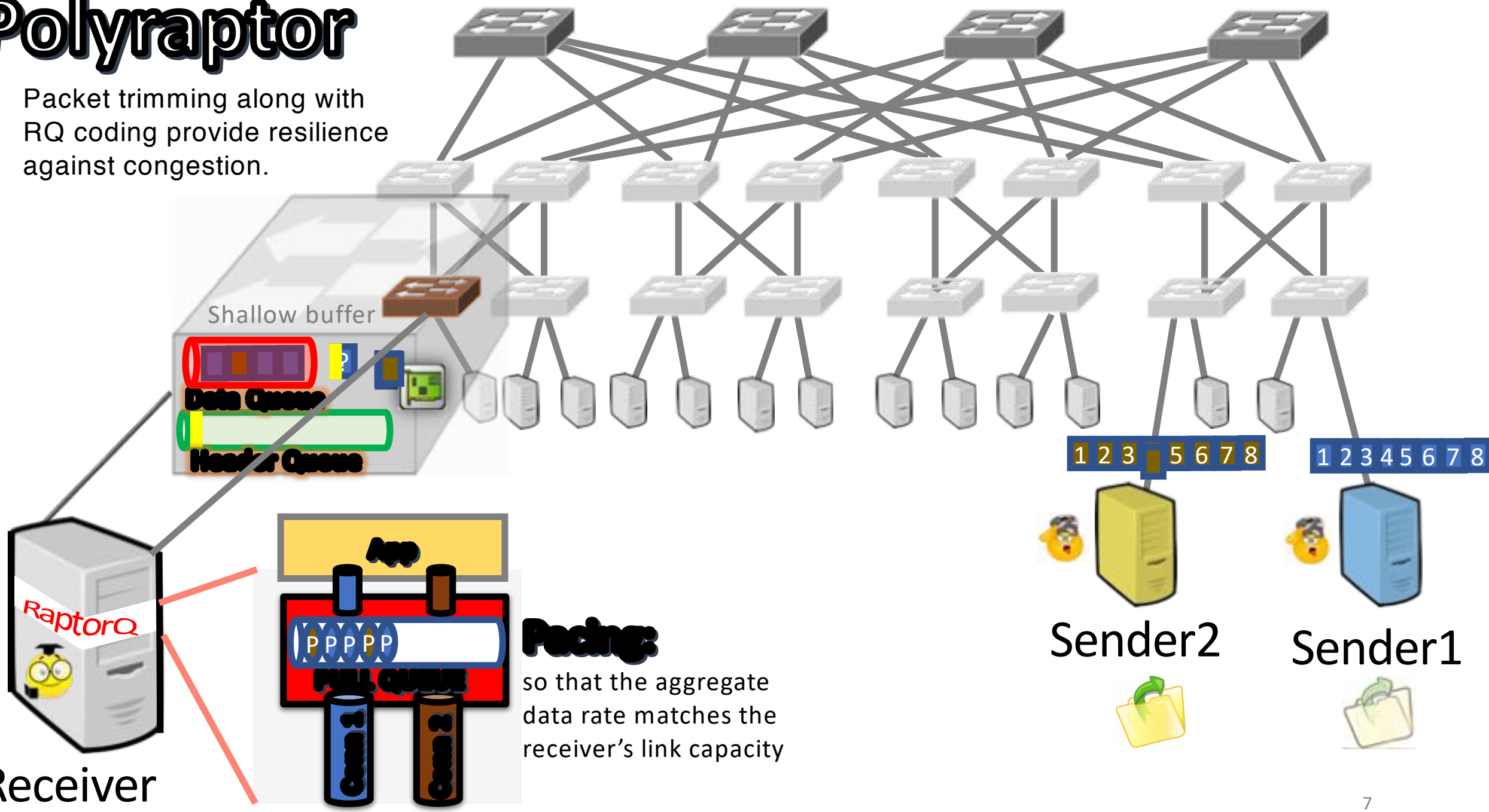
Polyraptor



Receiver-driven approach

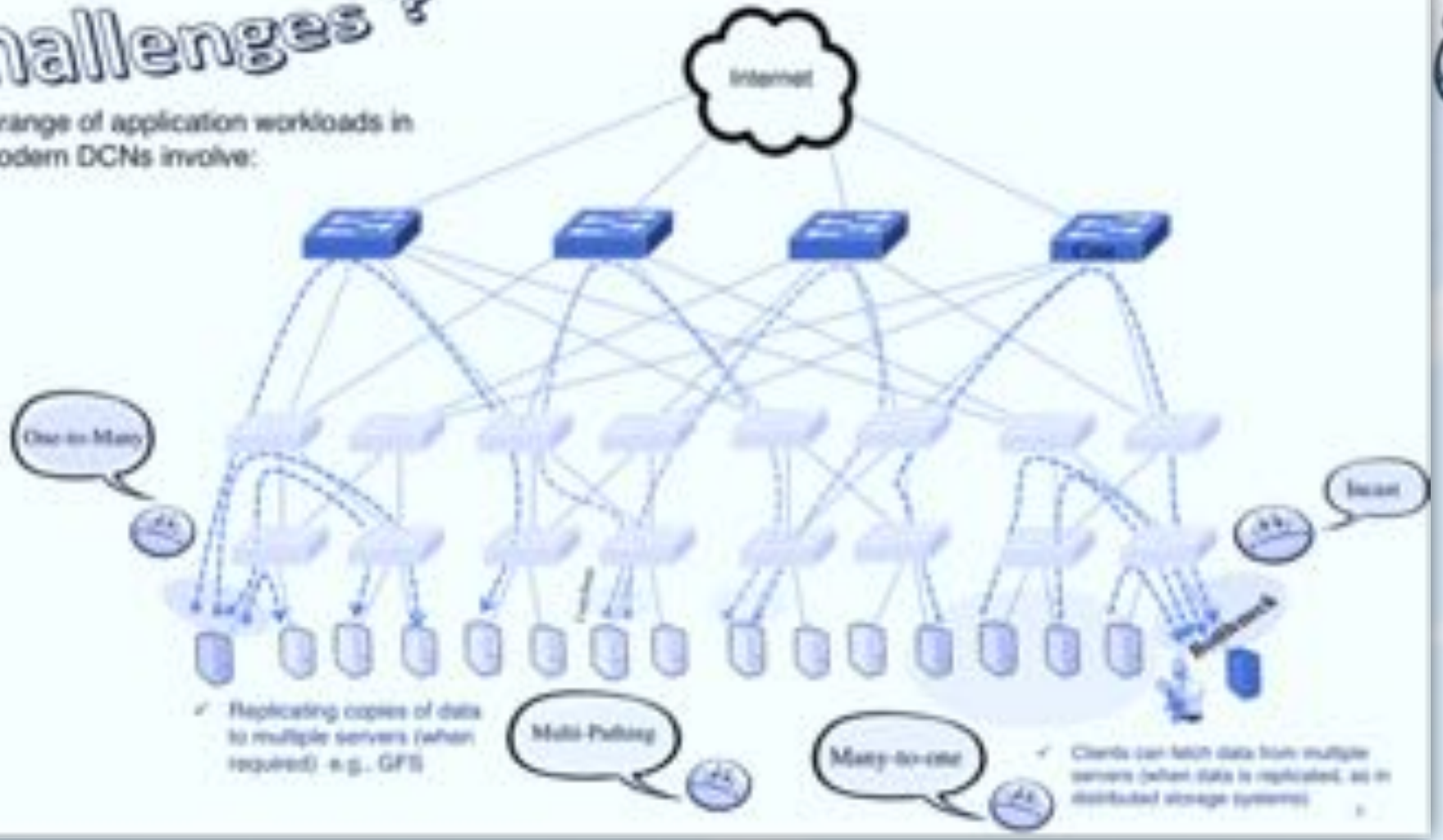
Polyraptor

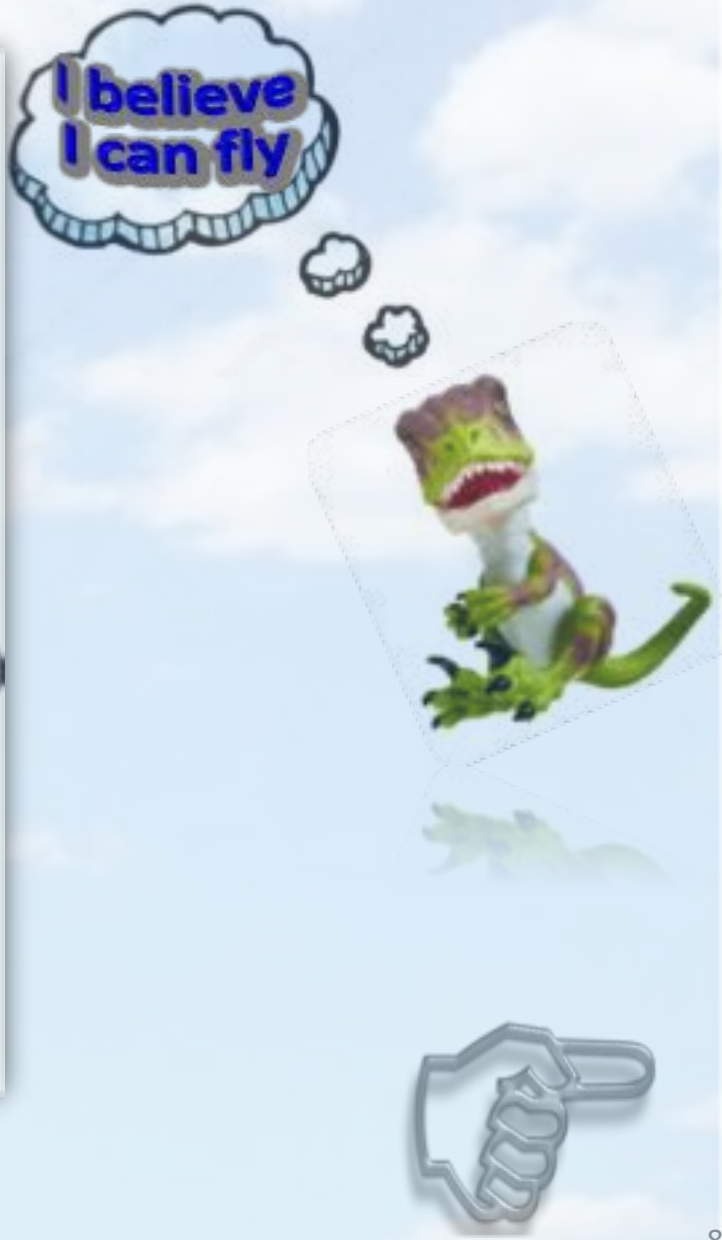
- Packet trimming along with RQ coding provide resilience against congestion.



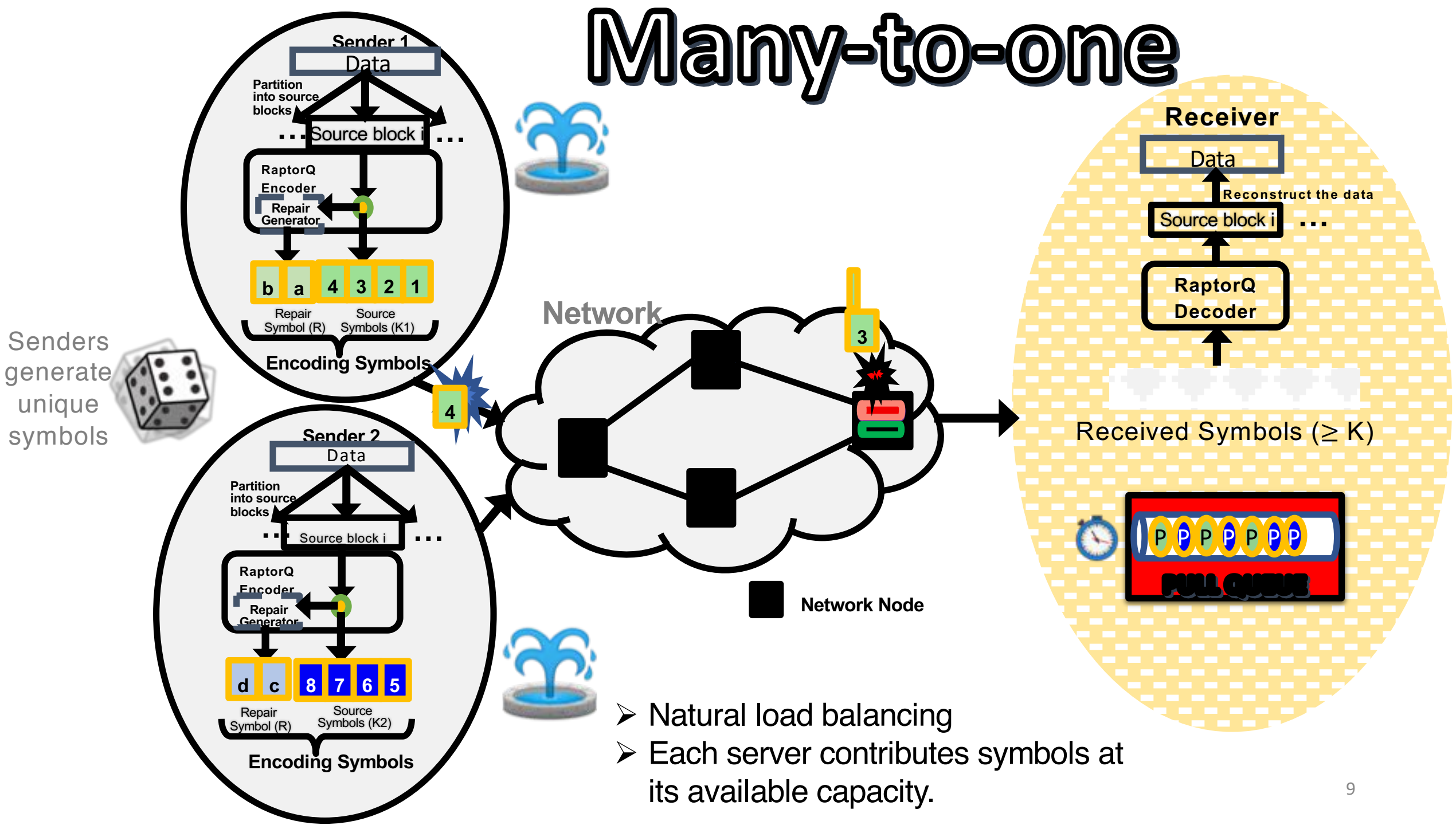
Challenges ?

- A range of application workloads in modern DCNs involve:



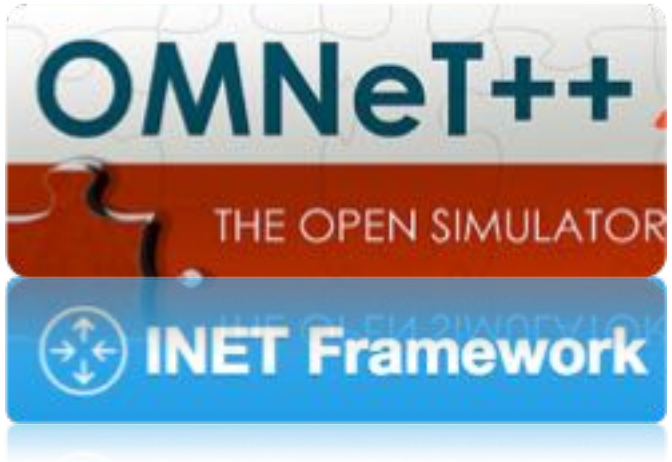


Many-to-one



- Natural load balancing
- Each server contributes symbols at its available capacity.

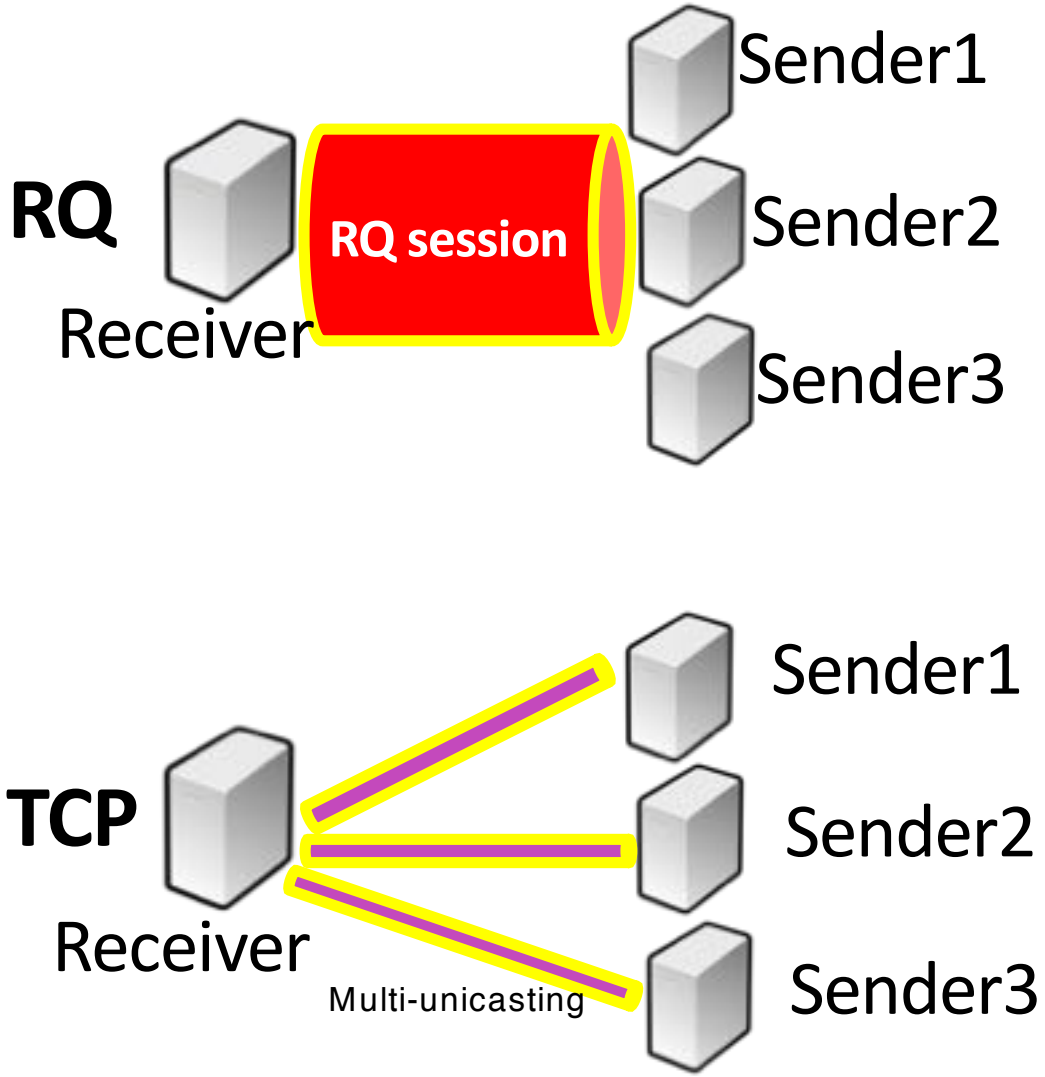
Simulation

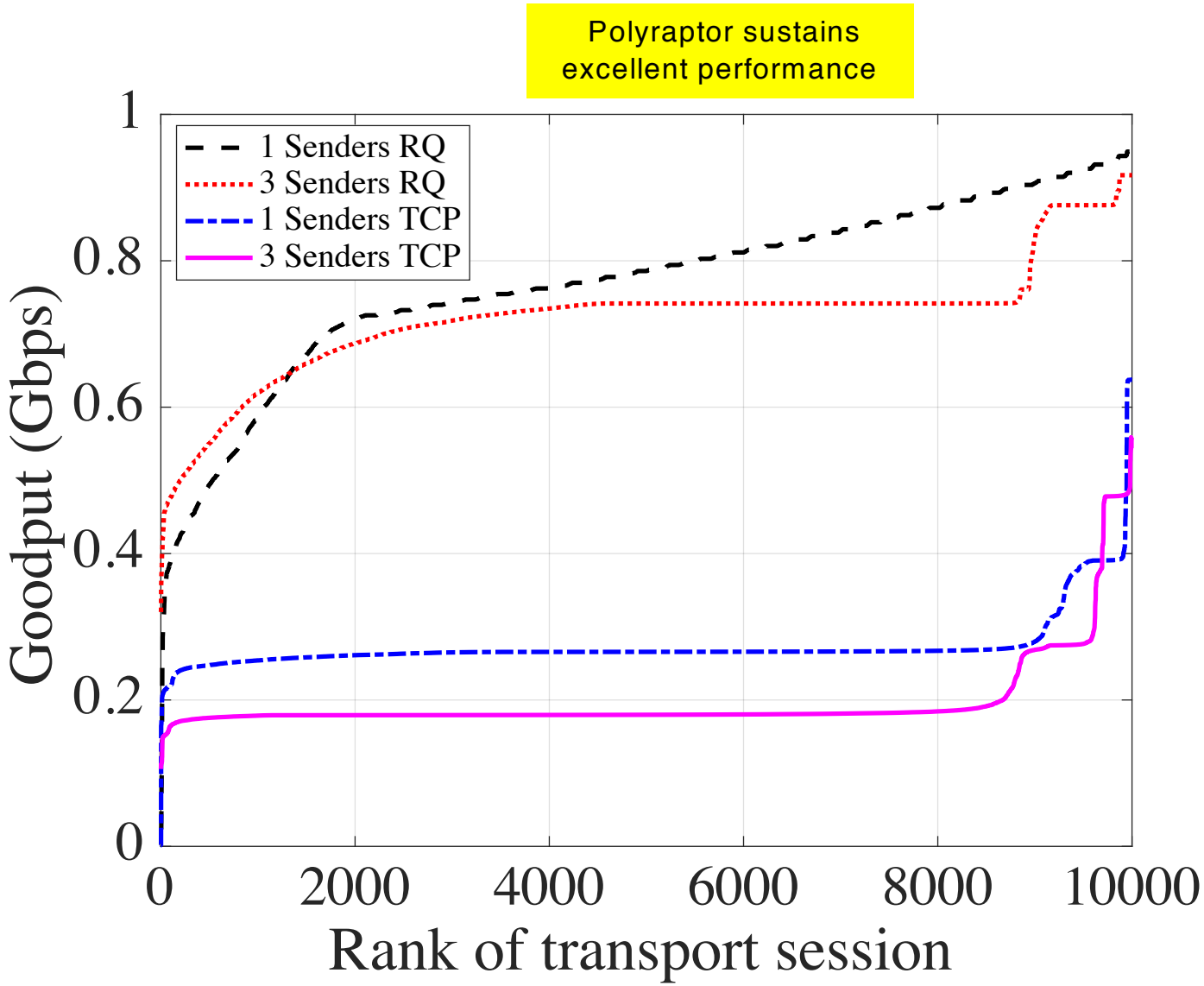


- ✓ FatTree topology: 250 servers
- ✓ 1Gbps link speed & 10μs link delay
- ✓ 10,000 Short sessions (4 MB each), arrival times follow a Poisson process with $\lambda = 2560$ session/sec
- ✓ 20% (2000) Long sessions (background traffic).
- ✓ Traffic matrix: permutation & random.
- ✓ A distributed storage scenario with 1 and 3 replicas.

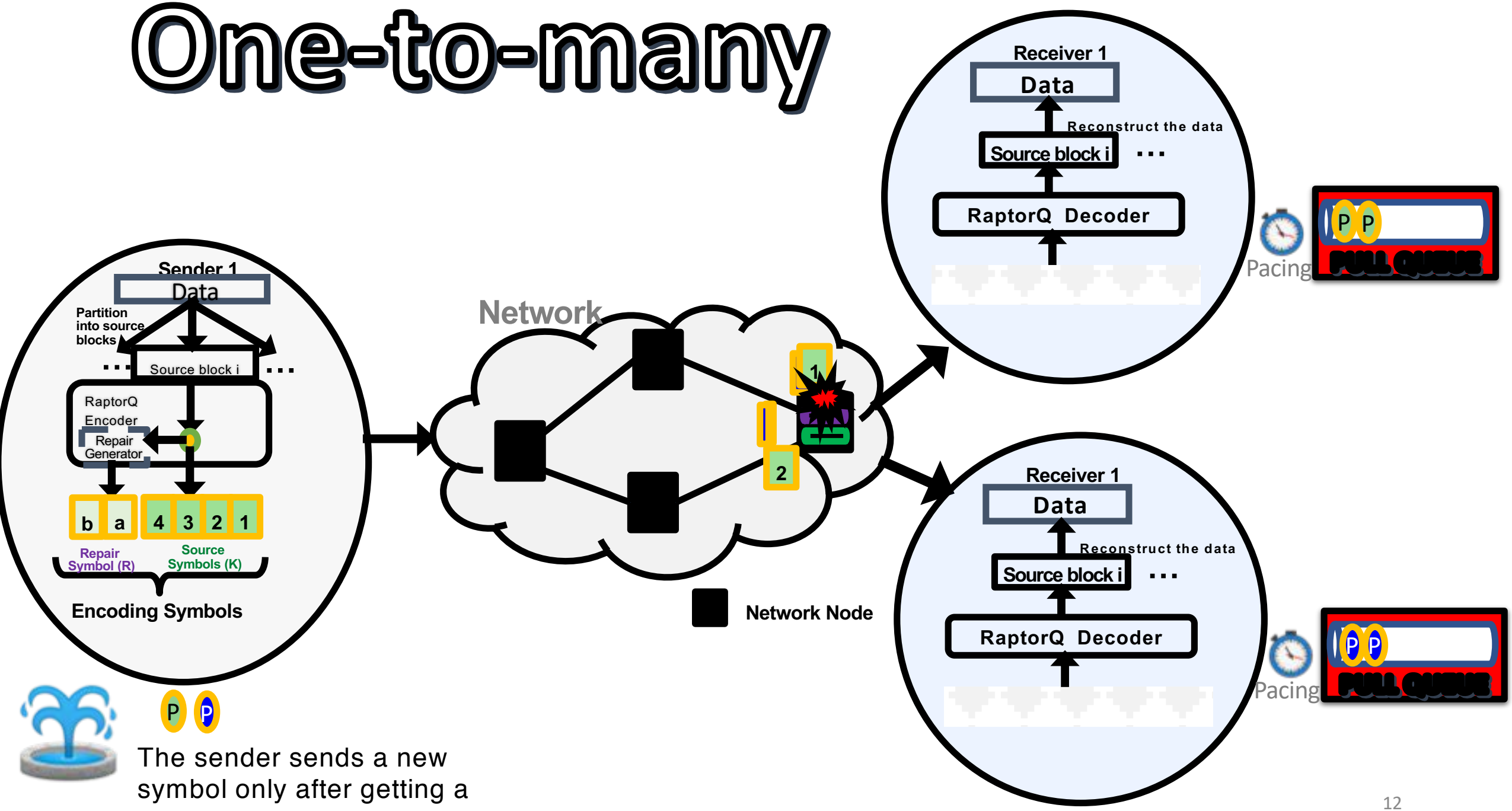
Many-to-one

A client fetches data from 1 and 3 replica servers



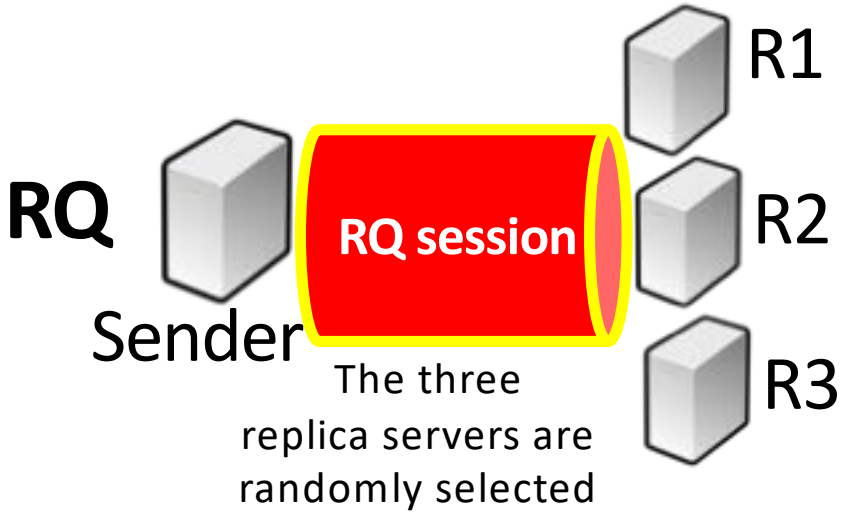


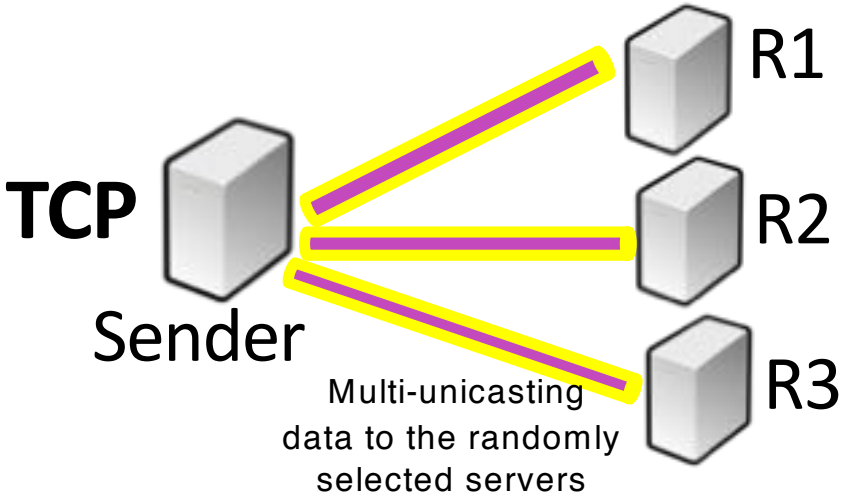
One-to-many



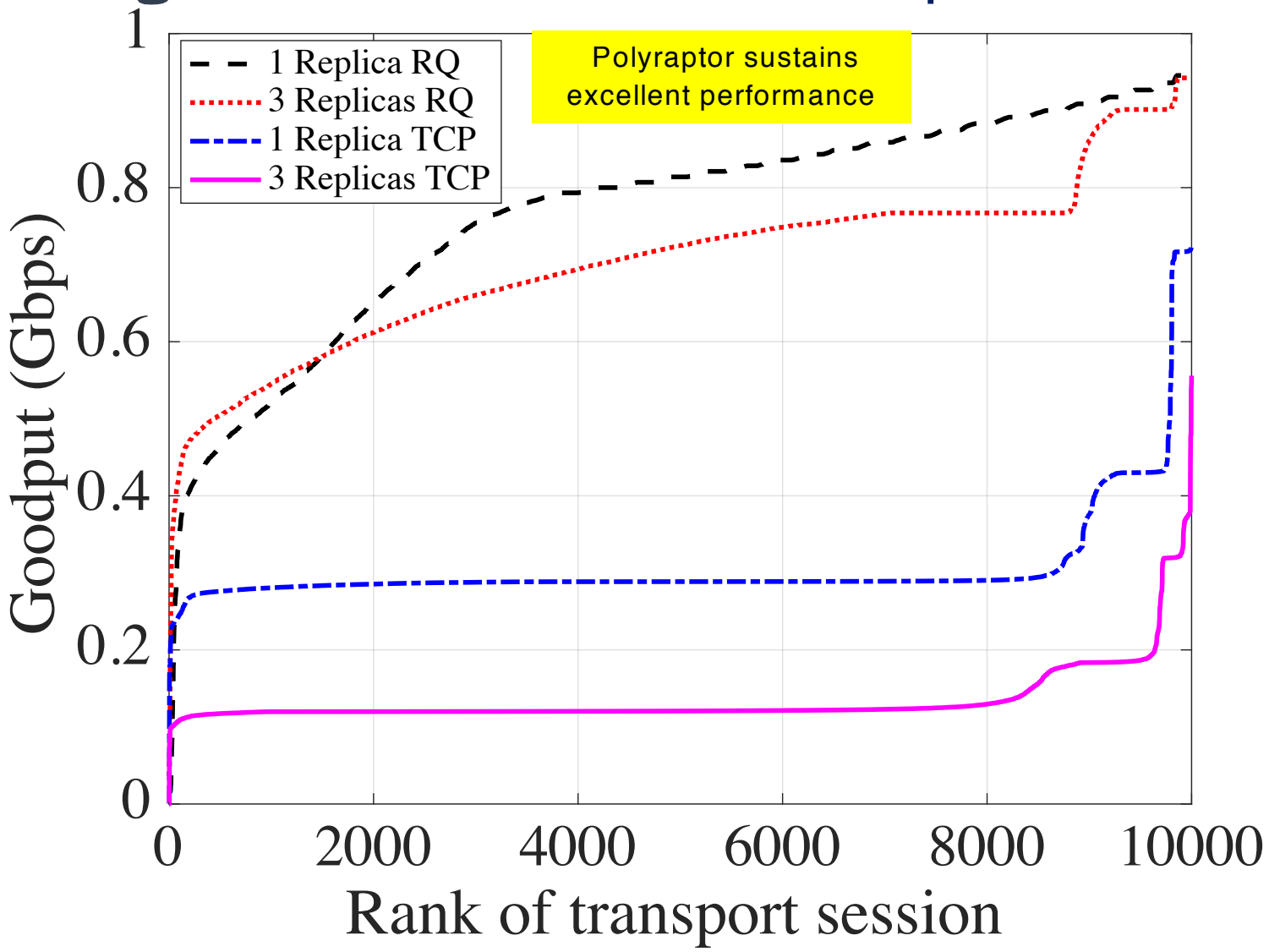
The sender sends a new symbol only after getting a request form all receivers.

One-to-many



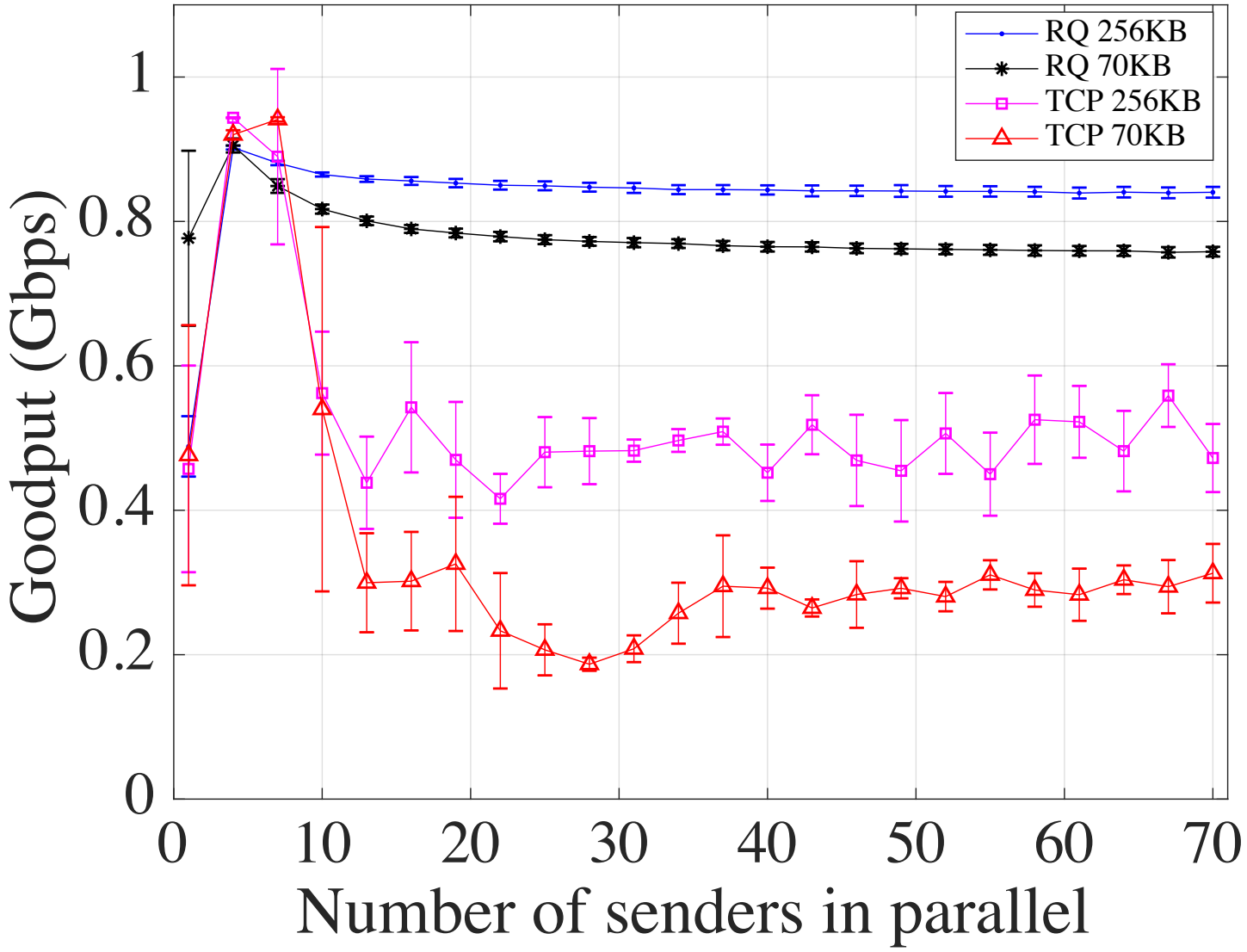


A distributed storage scenario with 1 and 3 replicas.



Incast

Polyraptor eliminates incast



Polyraptor: Embracing Path and Data Redundancy in Data Centres for Efficient Data Transport

Mohammed Alasmar
Department of Informatics
University of Sussex
M.Alasmar@sussex.ac.uk

George Parisis
Department of Informatics
University of Sussex
G.Parisis@sussex.ac.uk

Jon Crowcroft
Computer Laboratory
University of Cambridge
Jon.Crowcroft@cl.cam.ac.uk

ABSTRACT

In this paper, we introduce Polyraptor, a novel data transport protocol that uses RaptorQ (RQ) codes and is tailored for one-to-many and many-to-one data transfer patterns, which are extremely common in modern data centres. Polyraptor builds on previous work on fountain coding-based transport and provides excellent performance, by exploiting native support for multicasting in data centres and data resilience provided by data replication.

1 INTRODUCTION

Data centres support the provision of core Internet services,

2 DESIGN

Polyraptor employs a receiver-driven communication model, where receivers actively manage the rate at which encoding symbols arrive (effectively providing flow and congestion control), by explicitly requesting symbols from senders. RQ codes are rateless and systematic; encoding symbols consist of the source symbols (i.e. original data fragments), along with a potentially very large number of repair symbols. In Polyraptor, source symbols are sent at the beginning of a session, followed by repair symbols, as required by receivers. In the absence of loss, source symbols are immediately passed to the application without inducing any penalty in terms of

Conclusion