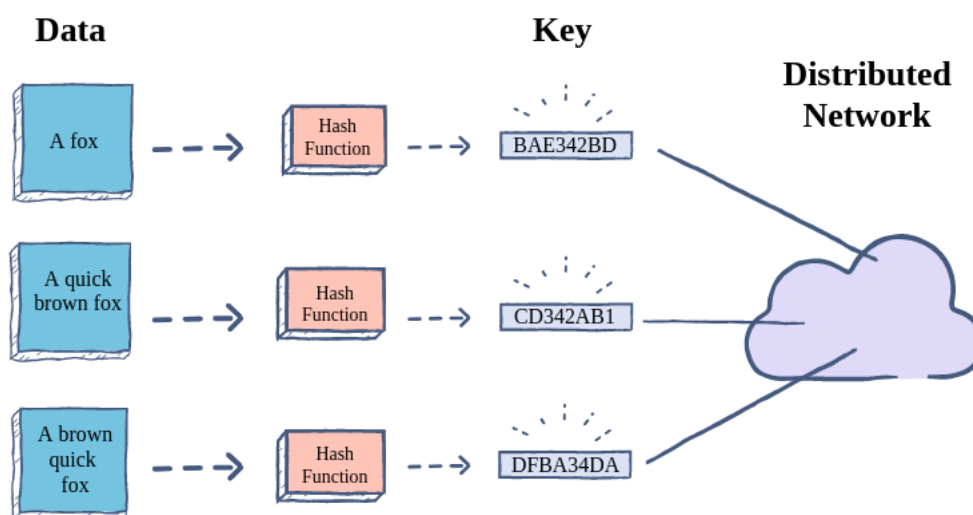


## Problem 1: Distributed Hash Table

### 1. Định nghĩa bảng Hash

Bảng băm phân tán (Distributed Hash Table - DHT) là một lớp các hệ thống phân tán không tập trung, cung cấp dịch vụ tra cứu tương tự như một bảng băm: các cặp (khóa, giá trị) được lưu trữ trong DHT, và bất kỳ nút mạng tham gia nào cũng có thể lấy được giá trị liên kết với một khóa cho trước một cách hiệu quả. Nhiệm vụ lưu trữ ánh xạ từ khóa tới giá trị được phân tán giữa các nút, bằng cách đó sẽ giảm bớt lỗi nếu có thay đổi trong một tập hợp các nút tham gia. Điều này cho phép sử dụng DHT cho một số lượng cực lớn các nút mạng và xử lý việc vào, ra, và lỗi các nút mạng một cách liên tục.



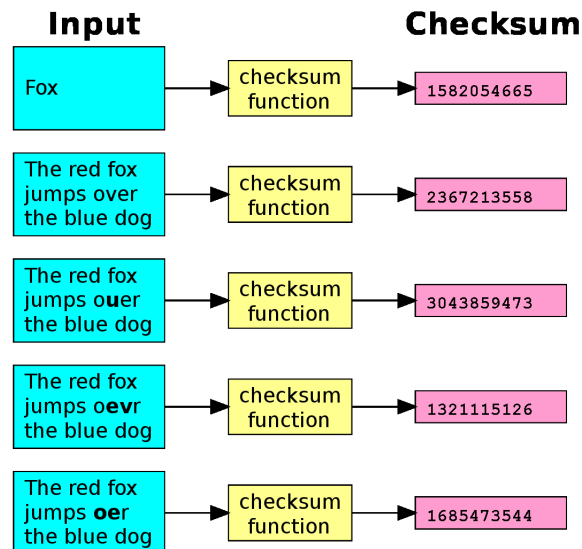
### 2. Ứng dụng của DHT

DHT tạo nên cơ sở hạ tầng cho việc xây dựng các dịch vụ phức tạp hơn, chẳng hạn như các hệ thống file phân tán, chia sẻ file trong mạng đồng đẳng, hệ thống phân phối nội dung (content distribution), web cache có tính hợp tác, multicast, anycast, dịch vụ tên miền và instant messaging. Các mạng phân tán nổi tiếng sử dụng DHT bao gồm máy theo dõi phân tán của BitTorrent, mạng eDonkey, mạng bot Storm, YaCy, và Coral Content Distribution Network.

## Problem 2: Checksum

### 1. Checksum dùng để làm gì ?

Giá trị tổng kiểm (Checksum) là một giá trị tính toán dùng để gắn vào một gói dữ liệu, dùng để dò tìm các lỗi trong các segment đã được truyền. Tính toán checksum của segment đã nhận, nếu không bằng với giá trị trong trường checksum bên gửi đặt thì đã có lỗi xảy ra.



Hình 1: Input to checksum

### 2. Ví dụ tính checksum 16-bit

Giả sử ta có 3 chuỗi 16-bit như sau:

0110 0110 0110 0000  
0101 0101 0101 0101  
1000 1111 0000 1100

Tổng của 2 chuỗi 16-bit đầu tiên là:

0110 0110 0110 0000  
0101 0101 0101 0101  
1011 1011 1011 0101

Cộng chuỗi 16-bit thứ 3 vào tổng vừa tính được:

```
0101 0101 0101 0101
1000 1111 0000 1100
0100 1010 1100 0010
```

Lưu ý rằng trong phép cộng cuối cùng có xảy ra overflow và đã được xử lí.

*Checksum* = 0101 0101 0101 0101

### Problem 3: Rdt

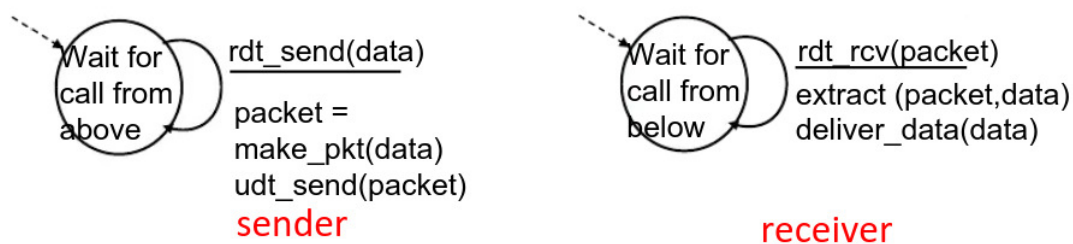
#### 1. Vì sao phải dùng các nguyên lí truyền đáng tin cậy ?

- Đảm bảo dữ liệu được gửi đến thành công;
- Dữ liệu được gửi đến đúng thứ tự và không trùng lặp;
- Hạn chế mất mát dữ liệu;
- Mở rộng tốc độ truyền tải

#### 2. So sánh các phiên bản RDT

- **RDT 1.0** Đường truyền lí tưởng

Giả định rằng kênh truyền bên dưới tuyệt đối không xảy ra lỗi bit và mất gói tin.



Hình 2: Bên nhận và bên gửi Rdt 1.0

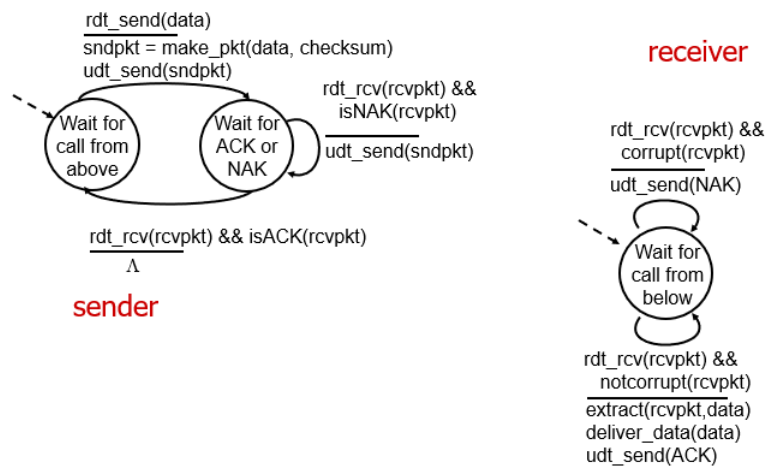
- **RDT 2.0** Kênh truyền có thể xảy ra lỗi bit

Sử dụng các cơ chế kiểm tra lỗi: *Checksum*

Cách khắc phục khi nhận ra lỗi:

- Acknowledgement(ACKs): bên nhận báo cho bên gửi đã nhận được dữ liệu
- Negativeacknowledgement(NAKs): bên nhận báo gói tin bị lỗi
- Bên gửi sẽ gửi lại gói tin khi nhận NAK

**So với rdt1.0, rdt2.0 có cơ chế phản hồi ACK, NAK để nhận dạng lỗi**



**Hình 3:** FSM bên nhận và bên gửi của Rdt 2.0

### • RDT 2.1 Phương thức cải tiến của Rdt 2.0

Rdt 2.0 cung cấp ACK và NAK để giải quyết trường hợp gói tin bị gián đoạn. Tuy nhiên trong trường hợp chính ACK hoặc NAK bị gián đoạn thì Rdt 2.0 rất có thể sẽ gặp lỗi. Rdt 2.1 giải quyết vấn đề này bằng Sequence number

### • RDT 3.0 Kênh truyền với lỗi và mất mát

Bên gửi đợi một khoảng thời gian hợp lí cho ACK

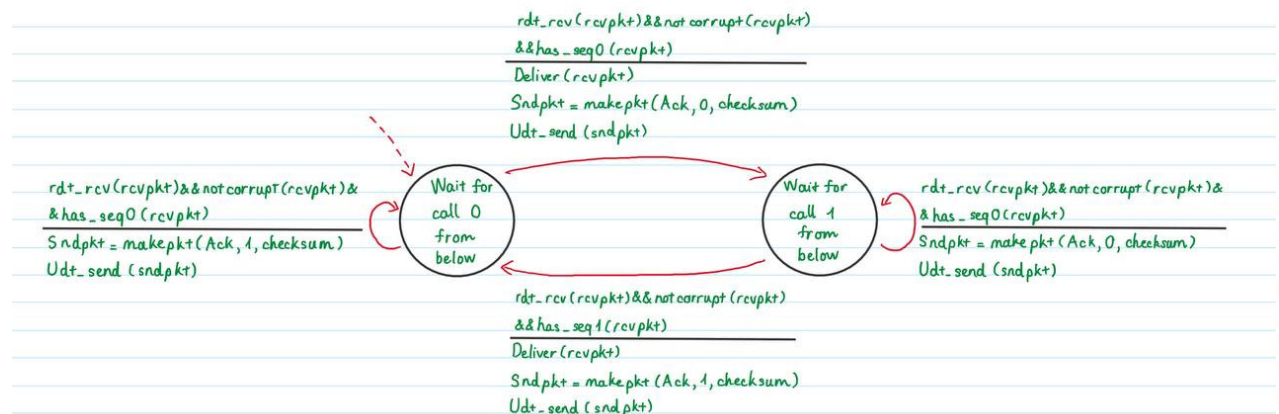
Gửi lại nếu không nhận được ACK không khoảng thời gian này

Nếu gói tin (hay ACK) bị trễ (không mất)

- Gửi lại có thể trùng, phải đánh số thứ tự
- Bên nhận phải xác định thứ tự của gói tin đã ACK

Yêu cầu **đếm thời gian**

## Problem 4: FSM của rdt 3.0 bên nhận



Hình 4: Reliable Data Transfer (3.0): Receiver FSM

## Contributors and Reference

Bảng 1: Contributors

Problem	Contributors
1	Kim Yến
2	Trâm Anh
3	Hoàng Tân, Gia Khang
4	Trâm Anh

[https://en.wikipedia.org/wiki/Distributed\\_hash\\_table](https://en.wikipedia.org/wiki/Distributed_hash_table)

<https://hazelcast.com/glossary/distributed-hash-table>

<https://en.wikipedia.org/wiki/Checksum>

[http://www2.ic.uff.br/~michael/kr1999/3-transport/3\\_040-principles\\_rdt.htm](http://www2.ic.uff.br/~michael/kr1999/3-transport/3_040-principles_rdt.htm)