LANCASTER UNIVERSITY

Stealth Distributed Hash Table

Unleashing the Real Potential of Peer-to-Peer

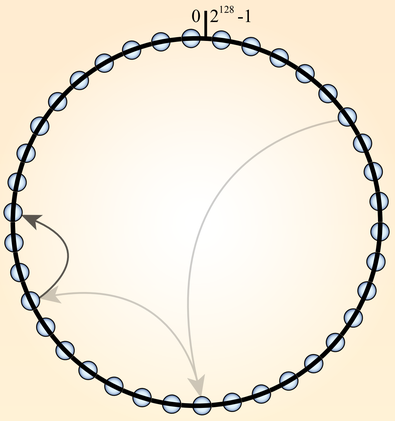
Andrew Brampton, Andrew MacQuire, Idris Rai, Nicholas Race and Laurent Mathy

Distributed Hash Tables (**DHTs**) are a form of **decentralised**, **scalable** and **resilient** peer-to-peer network. They are often used for applications such as *distributed storage*, *multicast*, *load-balancing and file sharing*

Most DHTs assume nodes are **homogeneous**, but in reality:

- **Underpowered** nodes **slow down** message forwarding
- **Malicious** nodes can **alter** messages in transit
- **Unstable** nodes cause **churn**

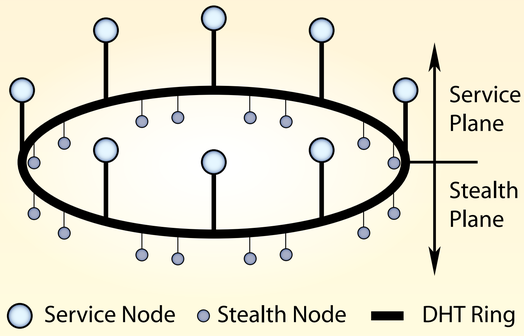
(Churn refers to the overhead incurred due to nodes rapidly joining and leaving the network; something which causes many DHT implementations to simply break down!)



Many DHTs use a ring structure, where messages are routed continually closer to their destination in a 128 bit circular address space

Stealth DHTs help to solve these problems through the use of two node types

(The assignment of these roles is application dependent and is in no way prescribed or constrained by the Stealth DHT itself)



Service Nodes

- Can perform all operations supported by the DHT
- Ideally **highly capable** and **reliable** machines

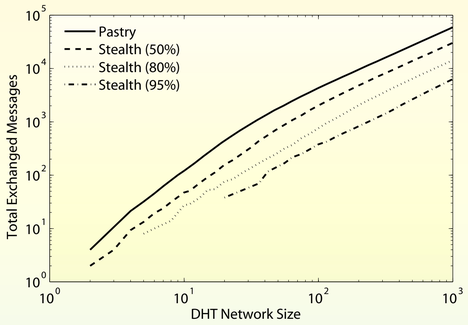
Stealth Nodes

- Prevented from storing keys and forwarding messages
- Likely to be **low powered** and **unreliable** machines

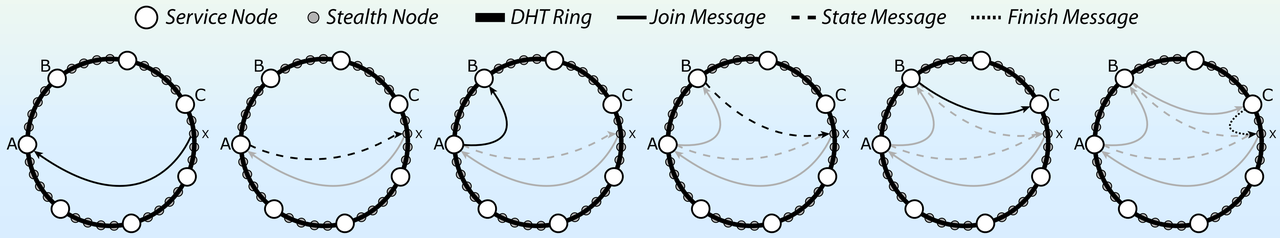
Differentiation is achieved through the use of a *lightweight join process* for stealth nodes

Stealth nodes **never announce their presence** in the DHT, meaning they never appear in any routing tables

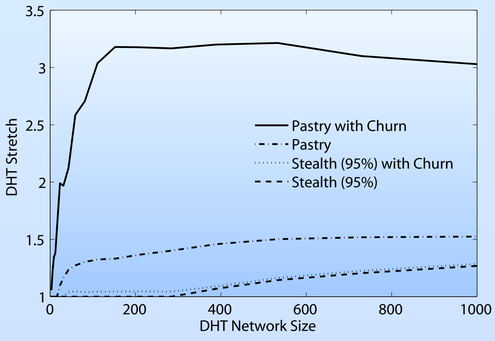
So when a Stealth node joins or leaves,
no routing data need change

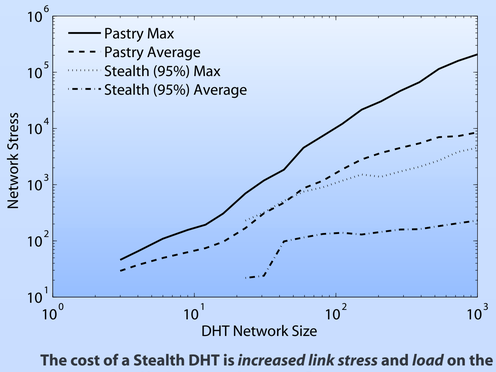


The figure above shows the total number of messages exchanged during the join procedure for Pastry in comparison with Stealth DHTs with varying percentages of stealth nodes.



The figure above illustrates how a Stealth node **x** gathers state and joins the DHT. Unlike generic DHTs, this is not followed by an announcement of the node's presence on the network.





The cost of a Stealth DHT is *increased link stress* and *load* on the service nodes. Under churn, however, Pastry generates so many maintenance messages that Stealth DHTs remain the better option

• Performance

Regardless of churn, Stealth DHTs offer improved performance over generic DHTs in many standard metrics (*hop count, stretch, load balancing etc.*)

• Security

With an appropriate authentication mechanism, a Stealth DHT can ensure that **only Service nodes route data**, eliminating problems such as *sniffing*, *man-in-the-middle* and *poisoning attacks*

• Control

Through the service nodes, a network provider can regulate all the content available on the network, thus aiding *Digital Rights Management* (**DRM**). Stealth nodes are allowed to push content into the network *only if they are trusted*, thus returning control to the owner of the system