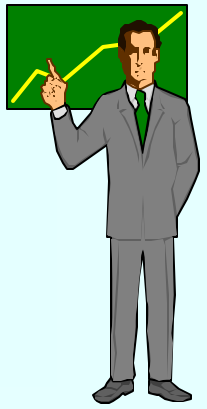
Alternative Approaches to Secure Multicast

Yair Amir, Baruch Awerbuch,
Theo Schlossnagle, Jonathan Stanton

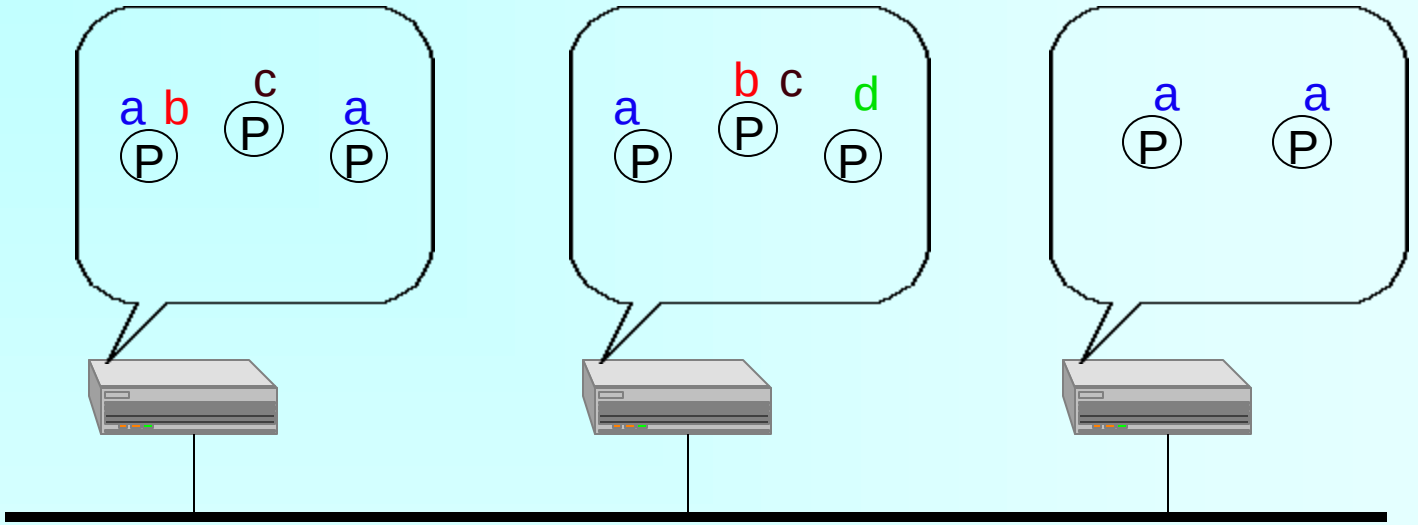
The Center for Networking and Distributed Systems

Johns Hopkins University

http://www.cnds.jhu.edu



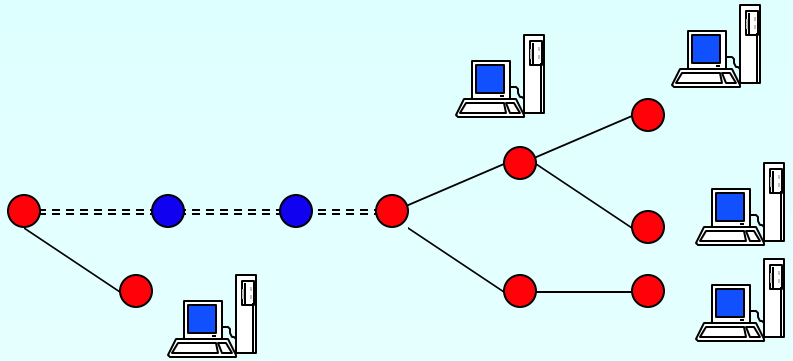
The Group Communication Model



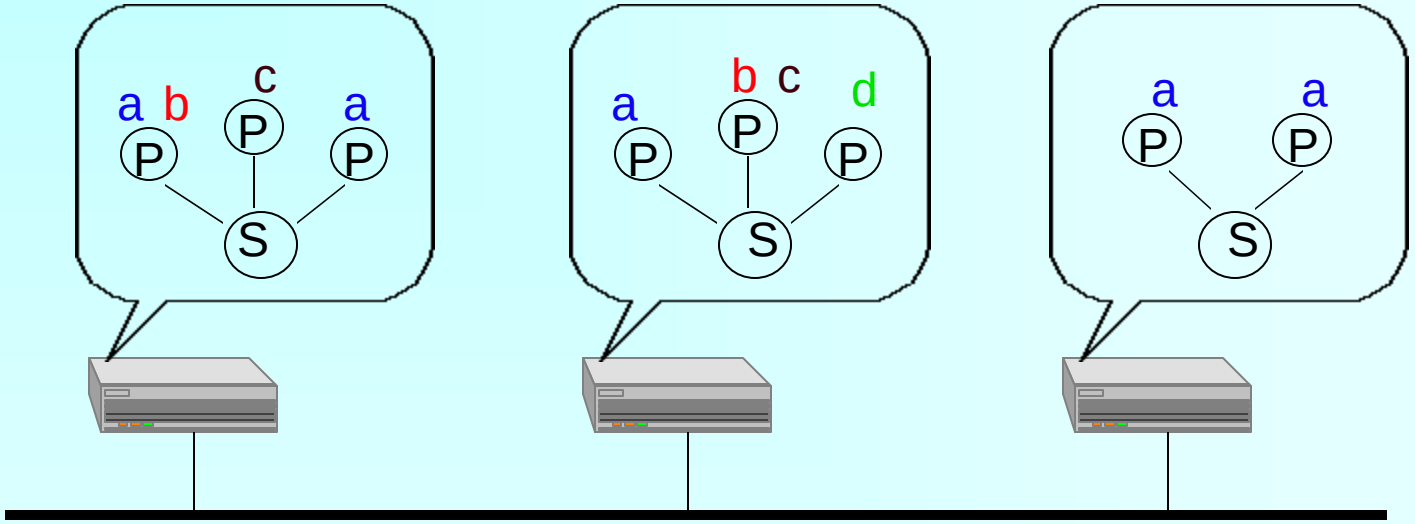
- Ordering (Unordered, FIFO, Causal, Agreed).
- Delivery guarantees (Unreliable, Reliable, Safe/Stable).
- Open groups versus close groups.
- Failure model (Omission, Fail-stop, Crash & Recovery, Network Partitions).
- Multiple groups.

IP Multicast

- Multicast extension to IP.
- IP multicast provides some of the traditional group communication services, specifically: unreliable, unordered, best-effort multicast, fully supporting the various failure models.
- No accurate membership.
- Utilizes network routers to store state of IP Multicast groups that span attached networks.

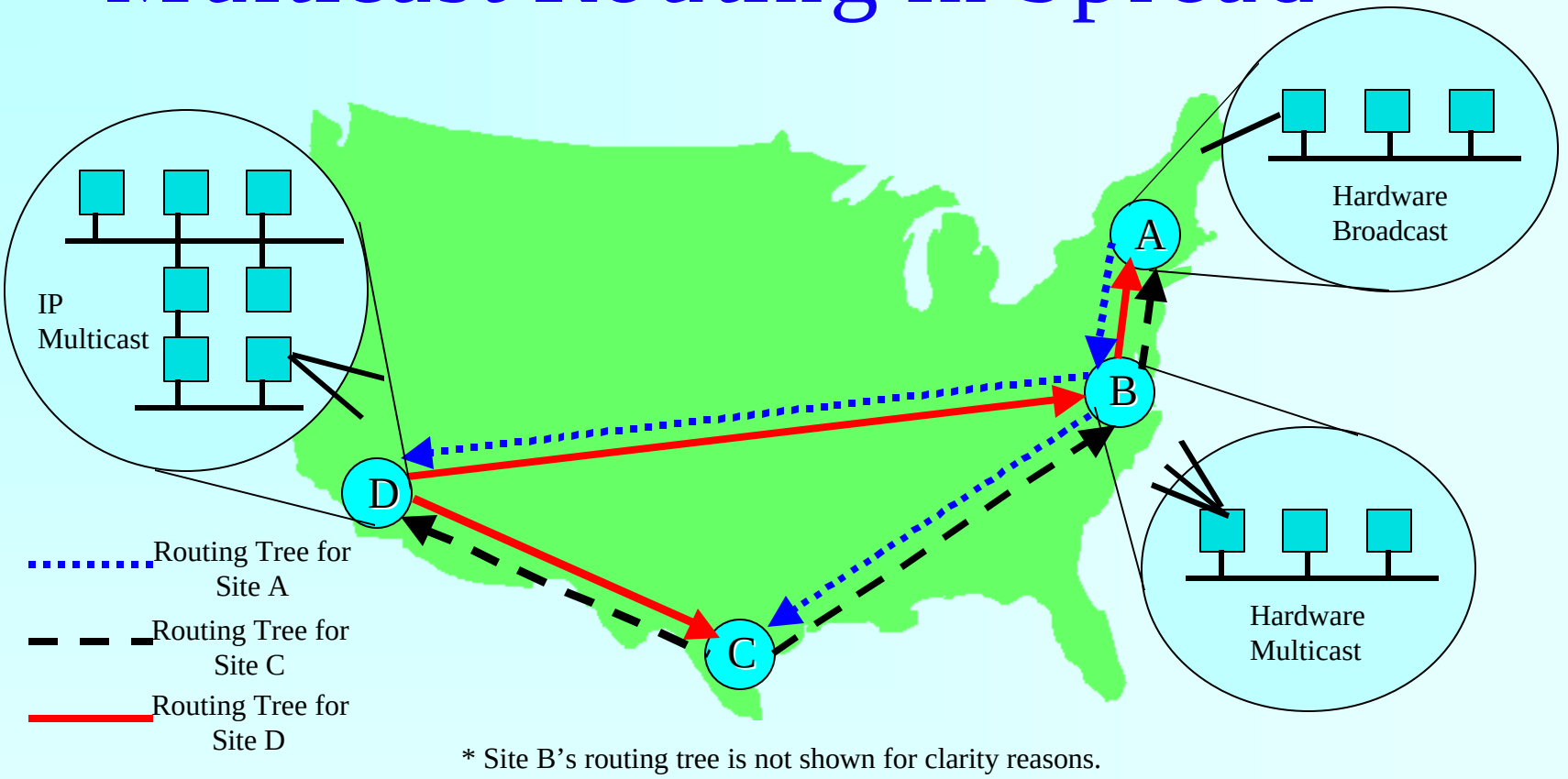


Spread: A Wide and Local Area Group Communication System



- Spread daemons are located in various parts of the network.
- Clients may connect to the nearest daemon.
- Network structure is transparent to the client processes.
- Open group semantics.

Multicast Routing in Spread



* Site B's routing tree is not shown for clarity reasons.

- Optimized routing tree for each multicast source.
- Network routers are not involved in group maintenance and multicast routing calculations.
- Network routers perform only insecure unicast routing.

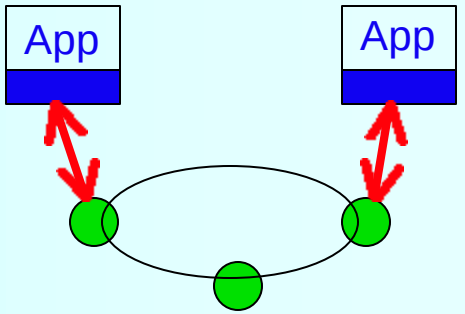
Advantages of Alternative approaches

- Not require changes to standard Internet Protocols.
- Security goes hand in hand with high availability and fault tolerance.
  - The group communication paradigm worked well for providing these, so it could work well for multicast security.
- Exploit advantages of the daemon model
  - Already are exploited for fault tolerant issues.
  - Not likely to be achieved in the Internet router level.

Secure Spread

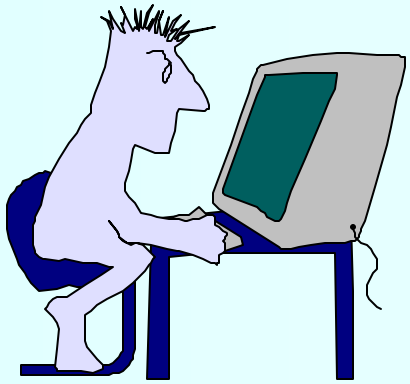
- Requirements
  - **Encryption, Signatures, Freshness.**
  - Immediate usability (without changes to network hardware or protocols, or operating systems).
  - High performance.
  - Allows administrative control of policy (e.g. group access control, how fast keys are refreshed, key size).
- Approaches
  - Spread Daemons are not trusted.
  - Spread Daemons are trusted.

Trust Tradeoffs



- Advantages in trusted daemons
  - High performance.
  - Can be reduced to two-party key agreement (client/daemon) aside of the core daemon group.
  - Better semantics (guaranteeing virtual synchrony after a membership change).
- Advantages in untrusted daemons
  - Daemons are not a security concern.
  - Better separation of security and networking code.

The Spread API



```
main( int argc, char **argv)
{
    mailbox     my_mbox;

    my_mbox = SP_connect( daemon, user);
    SP_join( my_mbox, “Hackers” );

    SP_multicast( my_mbox, AGREED, “Hackers”,
                  “Here is a new exploit in sendmail”);

    SP_recv( my_mbox,  Message, sender);
}
```

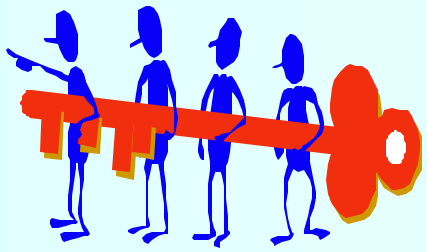
Secure Spread API

- Messages can be encrypted by setting a new ENCRYPT flag in SP_multicast.
- Messages can be signed by using a new call SP_signed_multicast(…, sign_group) .
- Messages are received by calling the same SP_recv which automatically decipher encrypted messages and verify signed messages.
- Freshness of clients can be cryptographically verified.

Key Agreement Algorithms

- Based on Cliques work from ISI:

  Tsudik et al, ICDCS98.



- Important features:
  - Decentralized.
  - Two message latency for join or leave of a group member.
  - Everyone contributes to the key.
  - Everyone can prove they took part in the generation of the key.

Experimental Results

- Large performance hit in the number of group joins and leaves possible per second (from 60 to 80 in generic Spread to less than 2).
- High computational cost for generating keys.
- High use of data on disk (is this more secure somehow?).