On supporting Multi-hop communication in the AllJoyn Framework for Proximity Based Networks.

Hatim Lokhandwala

Faculty Advisor: Dr. Bheemarjuna Reddy Tamma

Department of Computer Science and Engineering,
Indian Institute of Technology, Hyderabad

cs13m1018@iith.ac.in

Thesis Evaluation,
27 Jun 2016

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Outline

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Introduction to PBN and MSNP

- PBNs (Proximity Based Networks) are ad hoc networks which aim to leverage the spatial proximity of the participants / devices / users to share real-time information of a particular transaction.
- These networks are
  - highly dynamic
  - self-organizing
  - infrastructure less networks and
  - facilitate real-time communication.

PBNs

- PBNs are opportunistic networks, which are a subclass of MANETs.
- PBNs are characterized by following features:
  - Spatial Proximity of Devices
  - Multi-Hop Data Transmission (Coverage Extension).
  - Real-time Data Sharing
  - Support Extension (Heterogeneous access technologies)

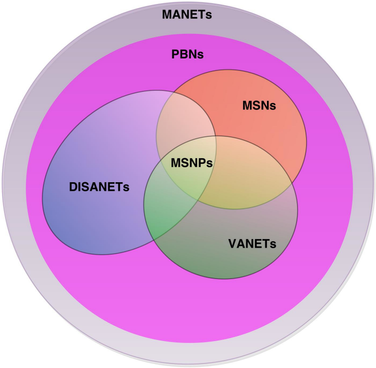


Figure 1: PBN classification and relationship

MSNP (Mobile Social Networks in Proximity)

- MSNP is a wireless Peer-to-Peer (P2P) network of spontaneously and opportunistically connected nodes and uses geo-proximity as a prime means to determine who is discoverable on the **social network.**
- MSNPs are a subclass of PBNs which are proximity based social networks.
- MSNP is the intersection of concepts from following three disciplines (Figure 2):
  - MSN (Mobile Social Networks)
  - Mobile P2P
  - Opportunistic Networks

MSNP(contd.)

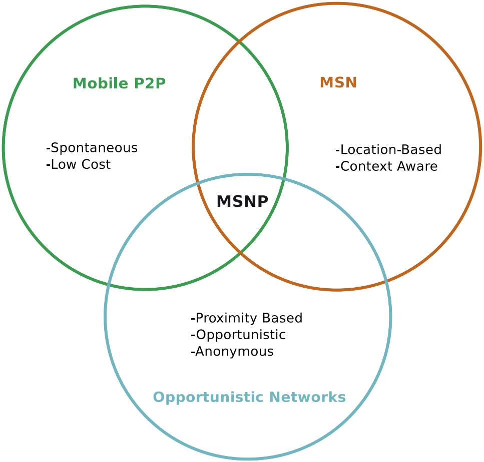


Figure 2: Relationship of MSNP with MSN, Mobile P2P & Opportunistic Networks.

Applications of PBN / MSNP

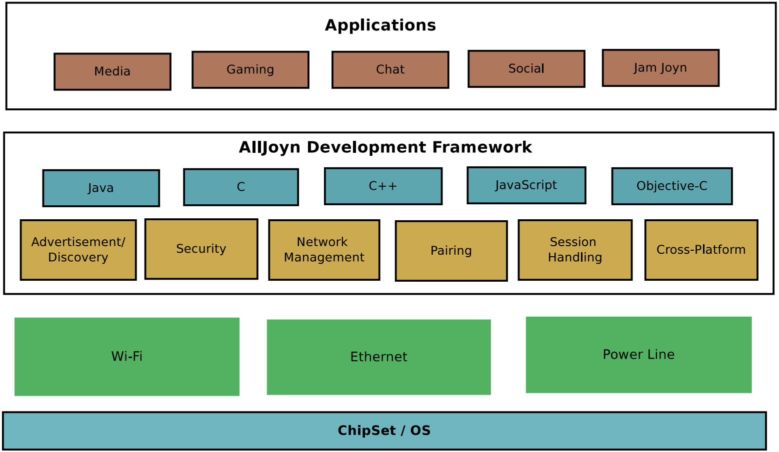
- Vehicular networks (Vehicle-to-Vehicle communication)
- E-Health
- Gaming
- Disaster networks
- Public outreach programs
- Education networks
- Mobile Social Networks

AllJoyn Framework

- Open source software framework.
- App/Device Discovery and Advertisement Mechanism.
- Heterogeneous technologies support (Wi-Fi, Ethernet, Powerline).
- Allows dynamic configuration of the network.
- Operational across multiple OSs (Linux, Android, iOS, Windows, etc.), making it platform independent.
- Support for different programming languages (C, C++, Java, Javascript, etc.).

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

AllJoyn Framework - Overview



भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Outline

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

AllJoyn Framework : Router

- AllJoyn Router (Software router) is a core module in the framework stack.
- It has dedicated sub modules for :
  1. Managing different underlying transports.
  2. AllJoyn software bus.
  3. Service advertisement & discovery.
  4. Session creation between services.
  5. Management of ongoing communication.
  6. Security mechanisms.
  7. Marshalling / Unmarshalling parameters for Remote Procedure Calls (RPC), etc.

AllJoyn Framework: Relationship between different entities

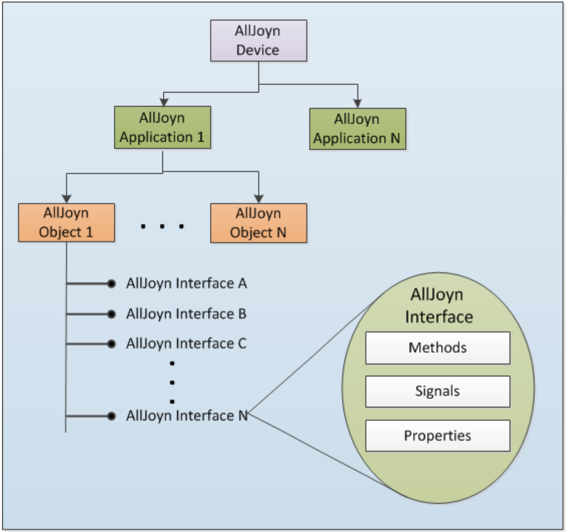


Figure 4: Relationship between different entities in the framework [14]

Outline

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Min-O-Mee Application (Leveraging the AllJoyn Framework)

- Min-O-Mee application is an PBN / MSNP based network application that facilitates participants to share Minutes-of-Meeting (MoM) in real-time in any academic or business meeting / get-together.
- It is an intended realization of the concept of MSNP leveraging the AllJoyn Framework.
- Application is based on Android platform and leverages the spatial proximity of participants in a meeting to share information.
- Application serves as a proof-of-concept and could be utilized in various real-time scenarios.

Min-O-Mee Application (Basic Design)

There will be two types of participants in a meeting.

- **Scribe :** A single participant who will create the Minutes-Of-Meeting (MoM).
- **Member(s) :** The remaining participants who would view and receive the MoM.

Scribe initiates the Min-O-Mee session and advertises a unique session name.

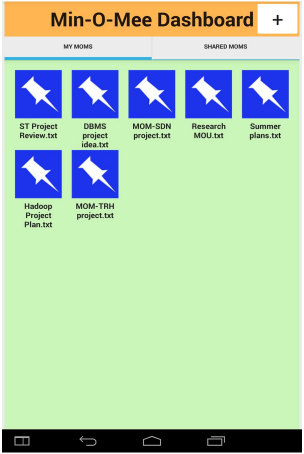


Figure 5: Application Dashboard

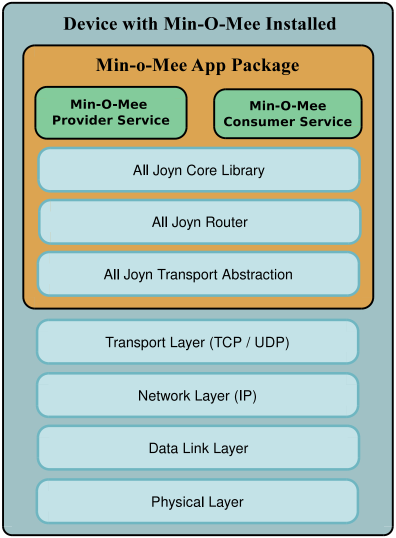


Figure 6: Application protocol stack

Stage I : Accomplished Tasks

- Literature Survey : PBN, MSNP and AllJoyn framework
- Familiarization with the theoretical concepts and the operational aspects of the AllJoyn framework.
- Comparison Analysis of the framework with other technologies.
- Development of an application prototype : Min-O-Mee leveraging the AllJoyn framework.
- Java based application variant was developed for Android phones and a C++ based variant for Raspberry PI.
- Identification of the enhancements that can be incorporated into the framework.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Current Working Model

- The current working model for devices to communicate with each other requires that Wi-Fi hotspot available on either of the devices be turned on.
- It is then the devices under same hotspot be able to discover and advertise services to each other and communicate further.
- An alternative set-up for this is when all the devices connect to a same access point.
- Clearly such set-up requires infrastructure and requires manual intervention in terms of connecting to a hotspot or an access point.
- It does not permit dynamic creation of a proximity based network or mobile social network in true sense.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Enhancements to the AllJoyn Framework

- AllJoyn framework though serves the basic purpose of ad hoc discovery and communication it still is in its nascent stage.
- We identify two enhancements that can be incorporated into the framework :
  1. Incorporation of support for short range technologies such as Bluetooth, Wi-Fi Direct and ZigBee.
  2. Multi-hop Communication mechanism in the framework

Support for Short Range Technologies (Bluetooth, Wi-Fi Direct, ZigBee)

- Inbuilt support for short range technologies in devices available these days.
- Support for these technologies would render the framework useful in a wide variety of IoT applications such as smart home networks, intelligent transport systems, etc.
- Figure 7 represents a dynamic, infrastructure less PBN / MSNP that could be set up through support for technologies such as Bluetooth and Wi-Fi Direct.
- Bluetooth and Wi-Fi Direct interfaces on these devices could communicate in an ad hoc and random fashion.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Realization of PBN / MSNP in true sense

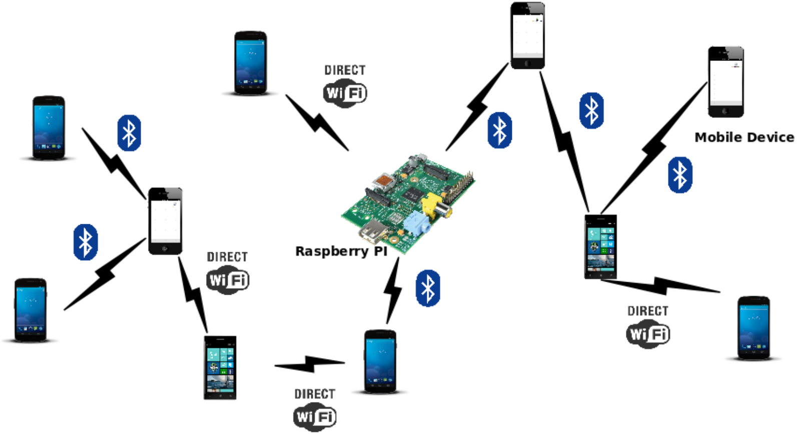


Figure 7: Support for Bluetooth / Wi-Fi Direct would facilitate dynamic PBN / MSNPs in real sense.

Realization of Extended MSNPs through Bluetooth / Wi-Fi Direct support

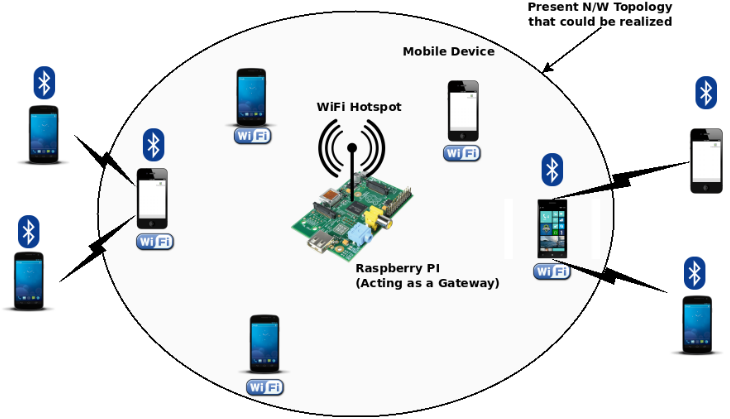


Figure 8: Support for short range transports could help to extend the network outreach from mere Wi-Fi only network

Multi-hop communication support

- AllJoyn performs some crucial functions viz., application discovery and advertisement, remote bus attachments, session management and data transfer between devices.
- However current implementation only supports communication between devices which are one hop away from each other.
- This is due to inherent drawback in the existing working model of the framework.
- The existing model as explained previously requires devices to be connected to the same access point or a hotspot.

Need for Multi-hop communication mechanism

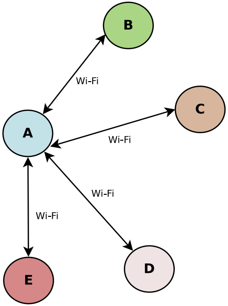


Figure 9: Topology at time t = t1

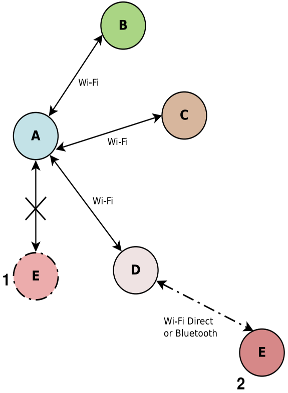


Figure 10: Topology at time t = t2

Potential Applications / Use Cases

- Below we present domains /applications which could benefit from the realization of the proposed enhancements :

1. Vehicular Communication System (VCS)
2. Disaster Networks
3. IoT applications (Smart-Homes, Smart TVs : An AllJoyn case study [11])
4. Military applications
5. Gaming
6. Proximity based social networks

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Choice of the underlying routing algorithm

- Although there exists many routing algorithms in the domain of ad hoc and dynamic networks works, we primarily choose two algorithms
  1. B.A.T.M.A.N (**B**etter **A**pproach **T**owards **M**obile **A**d Hoc **N**etworking)
  2. OLSR (**O**ptimized **L**ink **S**tate **R**outing).
- These algorithms are widely popular, thoroughly studied by the networking community.
- Extensive research on their performance in real-time test-bed on varying types of topologies (static / dynamic), network load etc. have been carried [6], [7], [8].
- Their potential to be utilised in real applications have been explored too [9], [10].

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

B.A.T.M.A.N v/s OLSR

- B.A.T.M.A.N is a lightweight protocol as it does not maintain entire routes to other destinations.
- OLSR on the other hand determines entire route towards destination.
- Topology control message in OLSR carry routes in their payload.
- This would result in a significant overhead as number of nodes in the topology increases.

B.A.T.M.A.N v/s OLSR

- B.A.T.M.A.N is a lightweight protocol as it does not maintain entire routes to other destinations.
- OLSR on the other hand determines entire route towards destination.

Performance evaluation of the 2 protocols on real-time test-bed :

- Experiments were conducted on 7 * 7 grid of Wi-Fi nodes operational at channel 6 and using Wi-Fi standard, 802.11b by [6].
- Testing of these algorithms in real-time on following metrics :
  1. Routing Overhead
  2. Throughput
  3. CPU usage
  4. Memory Consumption

B.A.T.M.A.N v/s OLSR

- **Routing Overhead :** For B.A.T.M.A.N this was observed to be 675 bytes/sec and 6375 bytes /sec for OLSR.
- **Throughput :** B.A.T.M.A.N has approximately 15% better throughput than OLSR at all inter node distances.
- **CPU Usage :** However OLSR had CPU consumption of 44%, significantly higher than the 4% CPU consumption by B.A.T.M.A.N.
- **Memory consumption :** OLSR memory requirements increased sharply beyond 30 nodes.
- Higher CPU usage, routing overhead & memory consumption in OLSR is attributed to increase in routing packet lengths with increase in number of nodes.

Choice of Routing Algorithm

- Choice of routing protocol would have direct impact on the performance of AllJoyn applications in the domains where AllJoyn framework is leveraged.
- The domains include and are not limited to, Internet of Things, proximity based network, ad hoc networks, vehicular networks, disaster networks, gaming, proximity based social networking, etc.
- In its real-time use cases, AllJoyn applications are operational across different types of devices with varying configurations.

Choice of Routing Algorithm

- Usage of OLSR as the routing protocol for multi-hop realization would result in heavy energy consumption, CPU and memory usage.
- This would be a bottleneck when AllJoyn applications are running on low powered embedded devices.
- Even if AllJoyn applications are running on mobile phones, tablets, laptops, etc., using OLSR will result in higher battery consumption.
- We therefore picked B.A.T.M.A.N routing protocol as the underlying routing protocol for realization of multi-hop communication between AllJoyn applications.

B.A.T.M.A.N routing

- Implementation of the B.A.T.M.A.N routing is obtained from the following source [5].
- The implementation is termed as B.A.T.M.A.N Advanced / batman-adv.
- batman-adv is available as kernel module with B.A.T.M.A.N routing operating on layer 2.
- Routing traffic is transported using raw Ethernet frames.
- We term the batman-adv implementation as Default approach in all further discussion.

Default approach

- Interface on which B.A.T.M.A.N routing is enabled is termed as originator.
- Every originator emits an originator message (OGM) every periodic interval, called as Originator Interval (Orig_Int).
- The originator messages are broadcast to the single-hop neighbours which further rebroadcast to their single-hop neighbours and so on.
- Since Default approach operates at layer-2, identity of originator message in the implementation is MAC address of the originator.

Default approach

- The size of originator message in implementation is 24 bytes.
- However the originator message is appended with TVLV (Type Version Length Value) container / information of 28 bytes.
- The actual originator message is of 24 bytes only and we term it as **Originator message content**.
- Size of entire message including Ethernet Header becomes 66 bytes.
- We term the entire packet of 66 bytes as **Default OGM packet.**

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Default approach

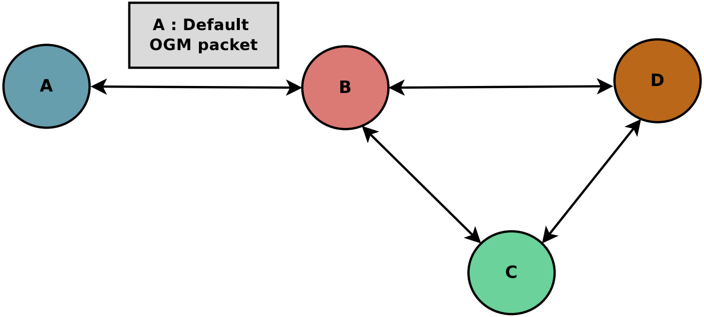


Figure 11: OGM packet : Broadcast from A

Default approach

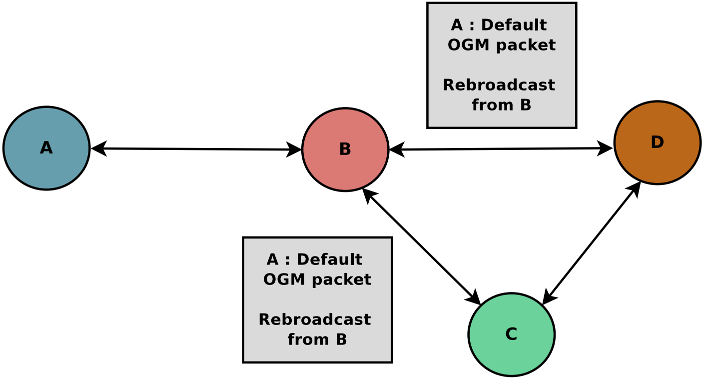


Figure 11: OGM packet : Rebroadcast from B

Default approach

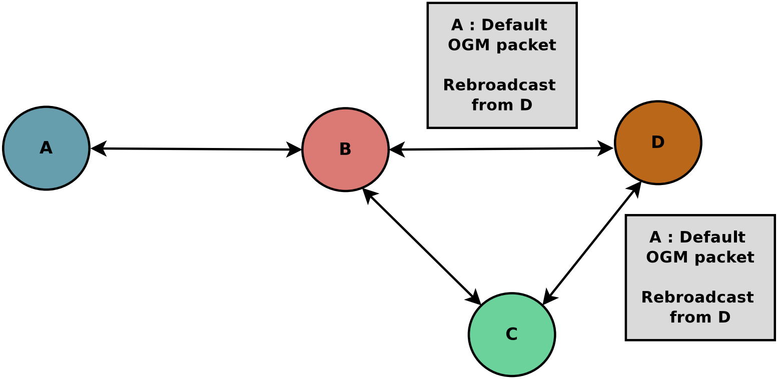


Figure 11: OGM packet : Rebroadcast from D

- Likewise C will also rebroadcast OGM packet from A.

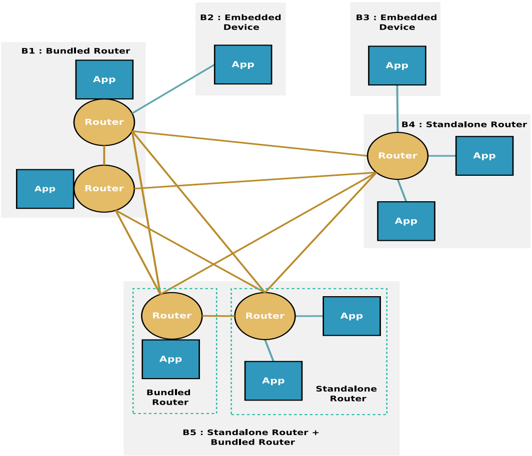


Figure 12: AllJoyn network topology and architecture illustration through devices (B1-B5).

Conventional Service Advertisement in AllJoyn

- Each application can have one or more services in it.
- The services connects to the software Bus in the AllJoyn router via Bus Attachments and requests for a unique name.
- If the request is satisfied the unique name is allocated to that service.
- The unique name allocated to the service is advertised by the AllJoyn router to other nodes [14].
- All the modificaitons in the AllJoyn framework are performed on the source code obatined from following source [13].

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Conventional Service Advertisement in AllJoyn

- Router sends IS-AT messages periodically over the IANA (Internet Assigned Numbers Authority) assigned addresses (Figure 1).
- IS-AT messages includes :
  1. Unique name of the service,
  2. Name of the interface service implements
  3. Property details of the interfaces
  4. Port at which service is running.

IPv4 Multicast group address	224.0.0.251
IPv6 Multicast group address	FF02::FB
Multicast port number	5353

Table 1: IANA assigned addresses and ports for NGNS mechanism in AllJoyn.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Approaches for Multi-Hop realization

- We design and develop 3 approaches for Multi-hop service advertisement and communication in AllJoyn using B.A.T.M.A.N.
  1. Naive approach
  2. Improved approach
  3. Ask / Reply approach

- Key challenge :
  1. AllJoyn applications, AllJoyn router and the AllJoyn library all operate at the user space
  2. B.A.T.M.A.N routing module operates in the kernel space.
  3. The applications in the kernel and the user spaces cannot communicate with each other directly.

Outline

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Naive approach

- We employ the debugfs mechanisms to let the AllJoyn framework communicate with the B.A.T.M.A.N routing module.
- debugfs is an in-kernel file system that provides means for an user space application to read data from / write data to files in debugfs.
- Kernel modules monitoring those files detect the write operation by a user space application.
- Modifications in batman_ adv implementation are performed to create a file named serviceInfo at the following path : /sys/kernel/debug/batman adv/bat0.
- This file is created in the debugfs virtual file system and it is created when the B.A.T.M.A.N module is enabled.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Naive approach

Service advertisement

- AllJoyn router would write the details about the services attached to it in the serviceInfo file in the debugfs file system.
- B.A.T.M.A.N module would obtain the written data from the serviceInfo file which it would come to know through the write call back.
- The module would form an appropriate message comprising the details of the service and advertise these details (How ?, explained later).

Naive approach : Integration Mechanism

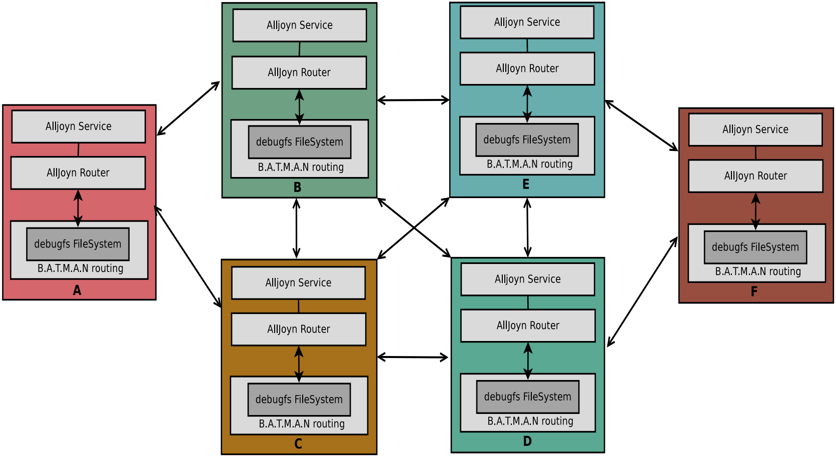


Figure 13: A network of nodes with B.A.T.M.A.N routing and AllJoyn framework integration on nodes.

Integration Architecture

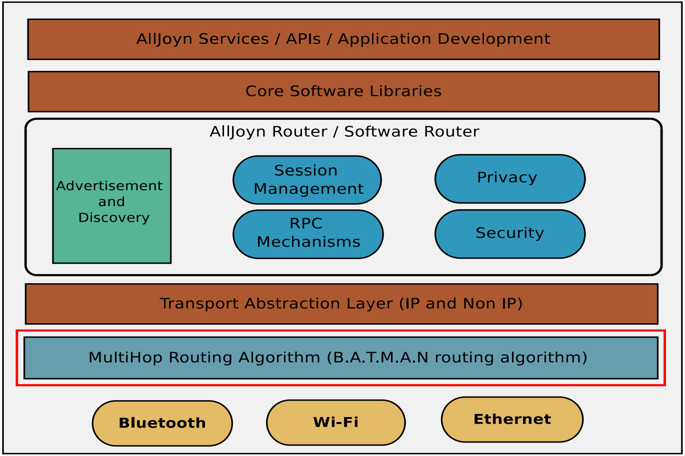


Figure 14: B.A.T.M.A.N AllJoyn Integration on a Node.

Naive approach : How services are advertised

- B.A.T.M.A.N routing module would append the AllJoyn service information to advertise the service details to other originators.
- It is appended at the end of originator message content and the TVLV information.
- AllJoyn service information comprises of the basic service details and the property details of the interface the service implements.
- Certain details in the service information are included by B.A.T.M.A.N routing also, besides those read from the 'serviceInfo' file.

Basic service details

Service Name	20 bytes
Interface Name	20 bytes
Source Ip	4 bytes
Source MAC	6 bytes
Property count	4 bytes
Port	2 bytes
protocol	1 byte

Table 2: Basic service details corresponding to a service.

isMethod	4 bytes
Name	20 bytes
Details	30 bytes

Table 3: Property details of the interface the service implements.

Total size of each of the table = 56 bytes (Structural Padding)

Naive approach

- Consider there are X services attached to the router and that the interface these services implements, each of them has Y properties.
- Number of bytes required for the advertisement of the X services would be X * (56 + Y * 56) bytes.
- However we consider that each device has a single service and interface the service implements has a single method.
- This is to account for uniform size of information be appended by routing module at each device.
- Hence the total size of the service information being advertised becomes 112 bytes.

Naive approach

- Service information is appended at the end of originator message content and Tvlv information.
- Originator message content has an additional field of 4 bytes carrying the length of service information appended.
- We term the entire packet so formed including the Ethernet header as **OGM-AllJoyn** packet (Size : 182 bytes).
- Table 4 depicts the structure of OGM-AllJoyn packet.

Ethernet Header	14 bytes
Originator message content	28 bytes
TVLV information	28 bytes
AllJoyn service information	112 bytes

Table 4: Size of different segments in the OGM-AllJoyn packet.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Naive approach

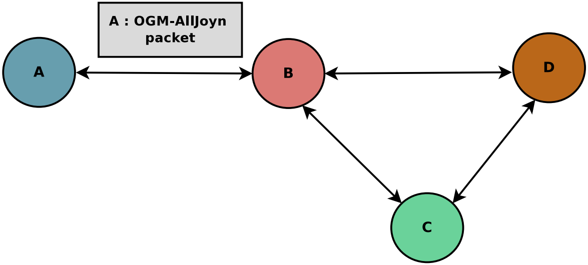


Figure 15: OGM-AllJoyn packet : Broadcast from A.

- Service information of AllJoyn service on node A is included in OGM-AllJoyn packet.
- OGM-AllJoyn packet is send every Orig_Int.
- Node B comes to know of service on node A from the OGM-AllJoyn packet.

Naive approach

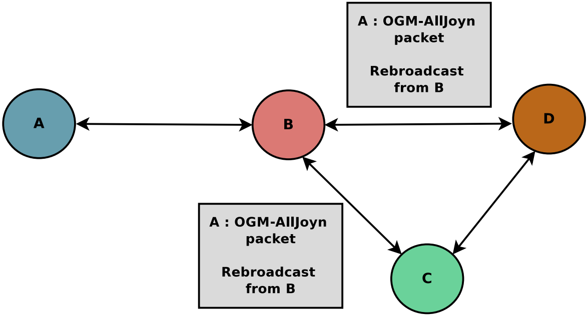


Figure 15: OGM-AllJoyn packet : Rebroadcast from B.

- Node B will rebroadcast OGM-AllJoyn packet including the service information.
- Nodes C and D will come to know of service on node A.

Naive approach

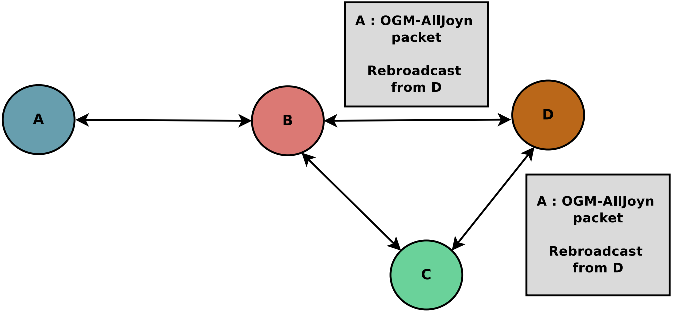


Figure 15: OGM-AllJoyn packet : Rebroadcast from D.

- Nodes C and D will rebroadcast OGM-AllJoyn packet including the service information.

Naive approach : Summary & Drawbacks

- In the Naive approach service information is appended every Orig_Int by every node.
- Service information is rebroadcast as it is by every intermediate node.
- Clearly this results in a lot of traffic / overhead from the service advertisement.
- This may increase drastically with increase in number of services on a node.
- Following two approaches : Improved and Ask / Reply approach are enhancements over the Naive approach with the goal to reduce the overhead generated from service advertisement.
- Debugfs mechanism used is same for other 2 approaches as well.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Outline

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Improved approach

- In the naive approach service information is transmitted every originator interval.
- To optimize advertisement mechanism, service information be send every x originator intervals.
- x can take value ≥ 2. (How to determine value of x ?).
- In the Improved approach we take the value of x as 2.
- This is taken based on the inferences drawn from the running the B.A.T.M.A.N routing over a real network.
- This is explained through the sample topology shown in Figure 16.

Improved approach

Consider below topology for understanding of taking value $x = 2$ for service advertisement.

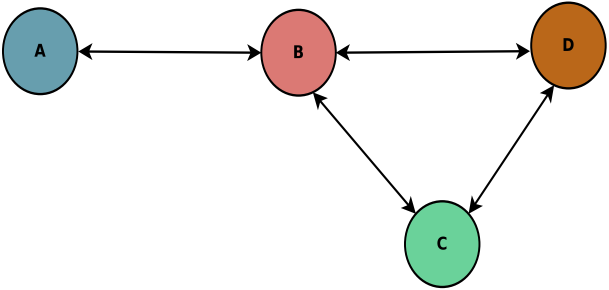


Figure 16: A network of nodes with B.A.T.M.A.N routing operational on them.

Improved approach

Single-hop

- Originator on A would receive OGM-AllJoyn packets from B.
- Originator on A would not conclude B as the next best hop towards B until 2 or more OGM-AllJoyn packets are received from B.
- This would take at least 2 originator intervals.

Multi-hop

- Originator on A would receive OGM-AllJoyn packets from C via B.
- Originator on B would not conclude B as next best hop neighbour for C until 2 or more OGM-AllJoyn packets of C are received via B.
- This would take atleast 2 originator intervals.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Improved approach

- In the first originator interval service information is not included.
- In the second originator interval service information is included.
- Interval in which service information is not included, packet so generated from it is called OGM packet 5 (Size : 70 bytes).
- This is different from Default OGM packet as it has 4 bytes field (serviceLen) in the originator message content.
- The value of field is though 0 as there is no service information.

Ethernet Header	14 bytes
Originator message content	28 bytes
TVLV information	28 bytes

Table 5: Size of different segments in the OGM packet.

Improved approach

Flow of OGM packets and OGM-AllJoyn packets in Improved approach

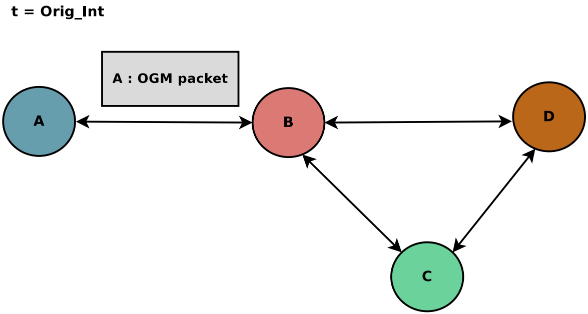


Figure 17: OGM packet : Broadcast from A.

Improved approach

Flow of OGM packets and OGM-AllJoyn packets in Improved approach

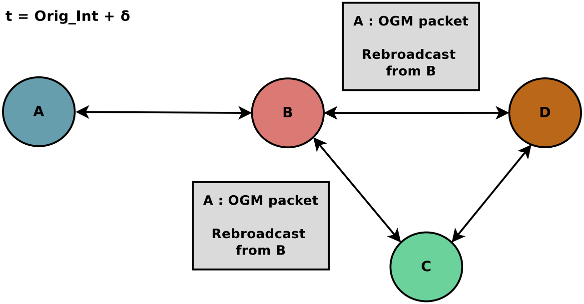


Figure 17: OGM packet : Rebroadcast from B.

Flow of OGM packets and OGM-AllJoyn packets in Improved approach

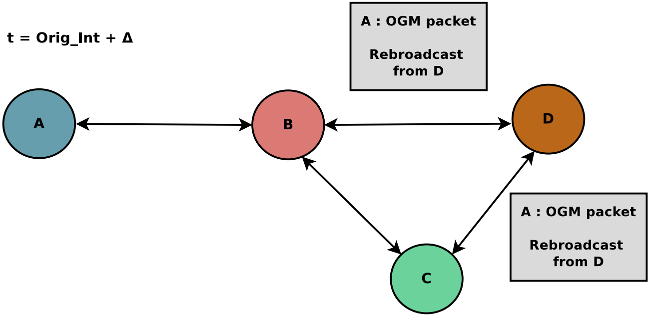


Figure 17: OGM packet : Rebroadcast from D.

Improved approach

Flow of OGM packets and OGM-AllJoyn packets in Improved approach

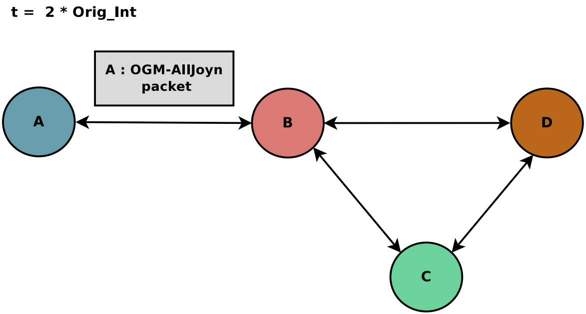


Figure 17: OGM-AllJoyn packet : Broadcast from A.

Flow of OGM packets and OGM-AllJoyn packets in Improved approach

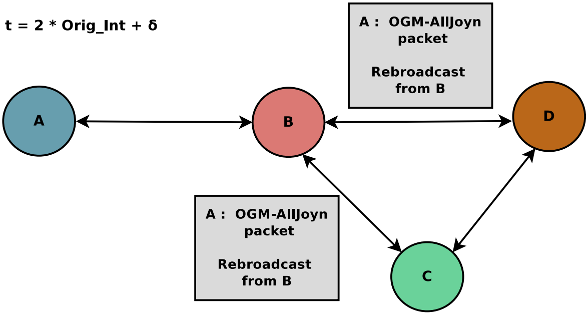


Figure 17: OGM-AllJoyn packet : Rebroadcast from B.

Flow of OGM packets and OGM-AllJoyn packets in Improved approach

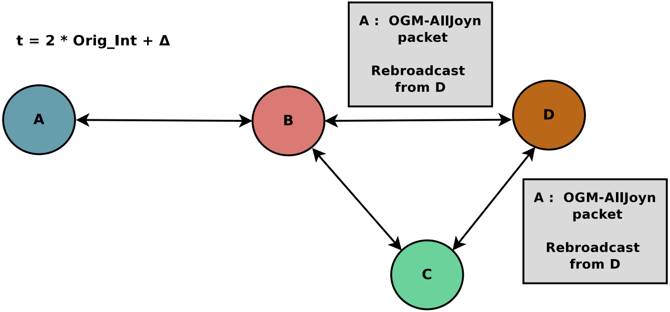


Figure 17: OGM-AllJoyn packet : Rebroadcast from D.

Improved approach : Summary & Drawbacks

- OGM packets and OGM-AllJoyn packets are send at alternating intervals starting from OGM packets.
- These are rebroadcast by the intermediate nodes without changing their contents.
- There is a reduction in service advertisement overhead due to advertisement in alternating Orig_Ints.
- However, rebroadcast of service information at the intermediate nodes may not be necessary all the times.

Outline

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Ask / Reply approach

- Every node would generate the OGM packet & OGM-AllJoyn packet in alternating Orig_Int. starting from OGM packets.
- Nodes receiving the OGM-AllJoyn packet will come to know about the service existent on another node.
- When a node first time receives the service information about the service on a certain node it is added in the originator table.
- Originator table is a hash table maintained at each node that has information related to other nodes in the network.

Ask / Reply approach

- When a node further receives the OGM-AllJoyn packet from a certain node, service information from it is compared with that in its originator table.
- If new service information is found only then service information is rebroadcast else truncated.
- Truncation of service information from the OGM-AllJoyn packet results into the formation of OGM packet.
- Originator message content & TVLV information in it is needed for the operation of B.A.T.M.A.N routing hence it cannot be truncated.

Ask / Reply approach

Flow of OGM packets and OGM-AllJoyn packets in Ask / Reply approach

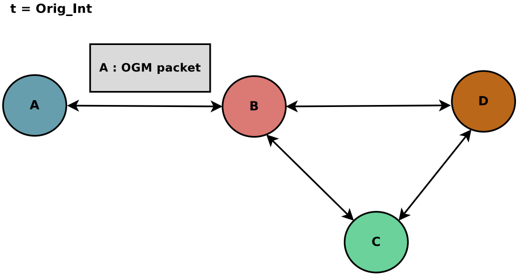


Figure 18: OGM packet : Broadcast from A.

Flow of OGM packets and OGM-AllJoyn packets in Ask / Reply approach

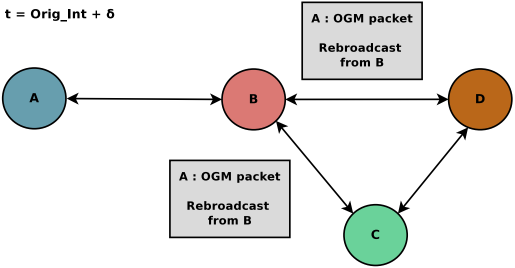


Figure 18: OGM packet : Rebroadcast from B.

Flow of OGM packets and OGM-AllJoyn packets in Ask / Reply approach

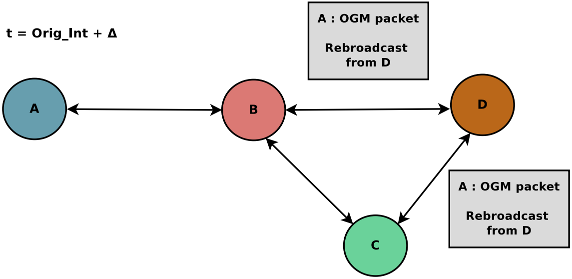


Figure 18: OGM packet : Rebroadcast from D.

Ask / Reply approach

Flow of OGM packets and OGM-AllJoyn packets in Ask / Reply approach

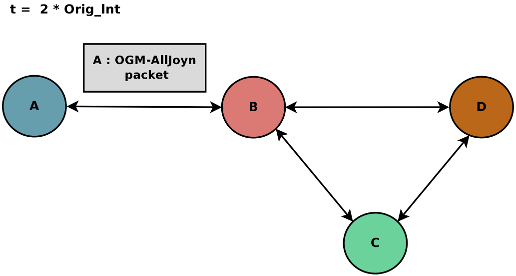


Figure 18: OGM-AllJoyn packet : Broadcast from A.

Flow of OGM packets and OGM-AllJoyn packets in Ask / Reply approach

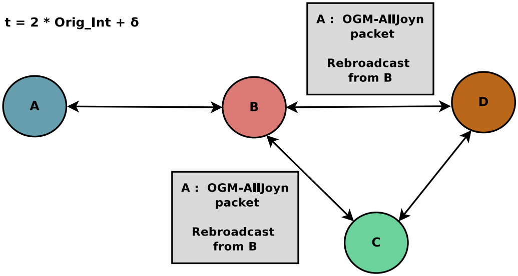


Figure 18: OGM-AllJoyn packet : Rebroadcast from B.

Ask / Reply approach

Flow of OGM packets and OGM-AllJoyn packets in Ask / Reply approach

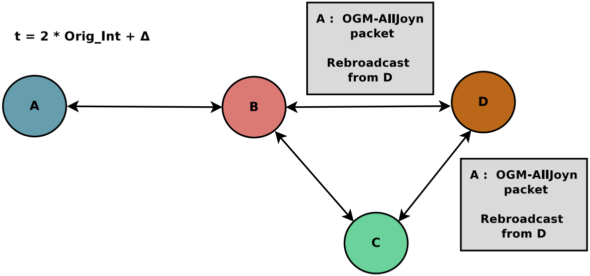


Figure 18: OGM-AllJoyn packet : Rebroadcast from D.

Ask / Reply approach

Flow of OGM packets and OGM-AllJoyn packets in Ask / Reply approach

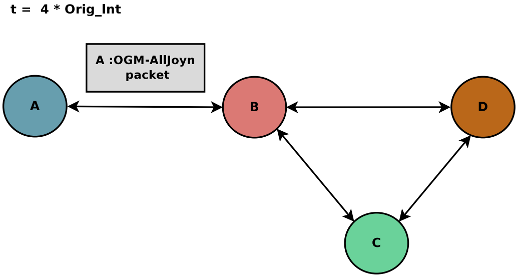


Figure 18: OGM-AllJoyn packet : Rebroadcast from D.

- OGM-AllJoyn packet broadcast from A at time t = 4 * Orig_Int.

Flow of OGM packets and OGM-AllJoyn packets in Ask / Reply approach

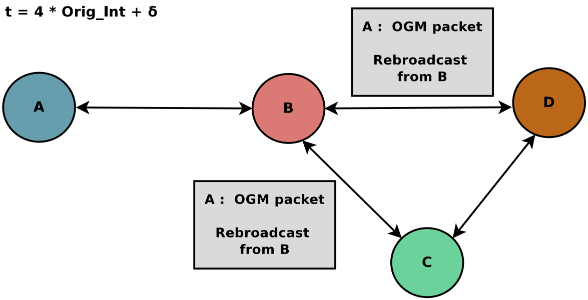


Figure 18: OGM packet : Rebroadcast from D.

- Node B already has the service information which is there in the packet.
- Node B truncates the OGM-AllJoyn packet and rebroadcast it as OGM packet.

Ask / Reply approach

Flow of OGM packets and OGM-AllJoyn packets in Ask / Reply approach

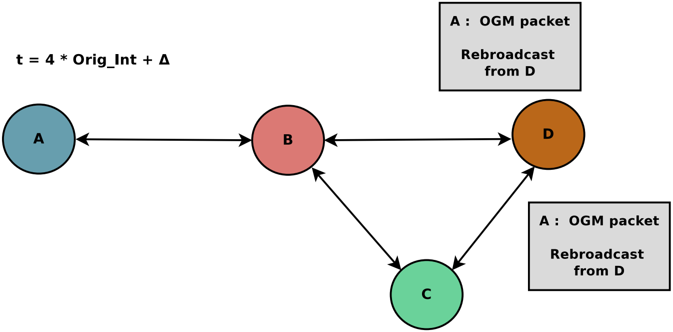


Figure 18: OGM-AllJoyn packet : Rebroadcast from D.

- Node D rebroadcasts the OGM packet.

Issues in Ask / Reply approach

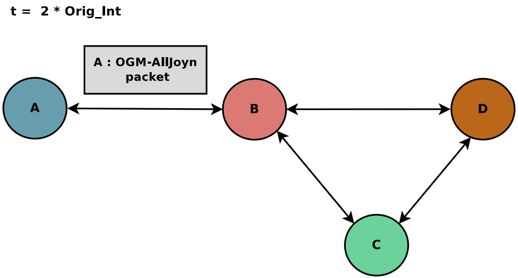


Figure 19: OGM-AllJoyn packet : Broadcast from A.

- OGM-AllJoyn packet broadcast from A at time t = 2 * Orig_Int.

Issues in Ask / Reply approach

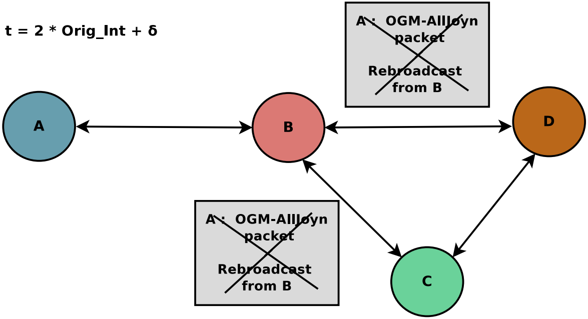


Figure 19: OGM-AllJoyn packet : Rebroadcast from B.

- Node B receives the service information for the first time.
- Node B rebroadcasts the OGM-AllJoyn packet as it is.
- However the packet is lost due to interference.

Issues in Ask / Reply approach

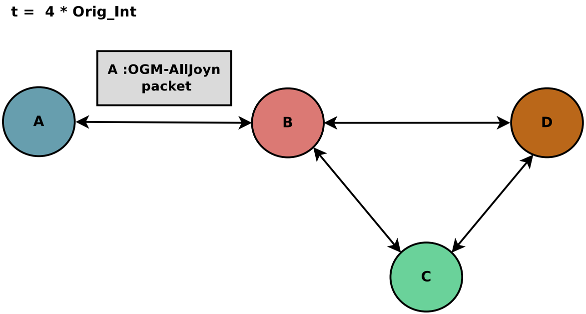


Figure 19: OGM-AllJoyn packet : Broadcast from A.

- Node A will send OGM-AllJoyn packet again at time t = 4 * Orig_Int.

Issues in Ask / Reply approach

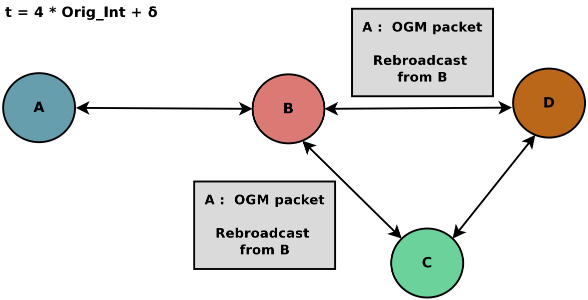


Figure 19: OGM packet : Rebroadcast from B.

- However node B already has the service information which is there in packet.
- Node B converts OGM-AllJoyn packet to OGM packet.
- Nodes C and D will not come to know of service on node A.

Issues in Ask / Reply approach

New node arriving in the network will not come to know of the services on other nodes.

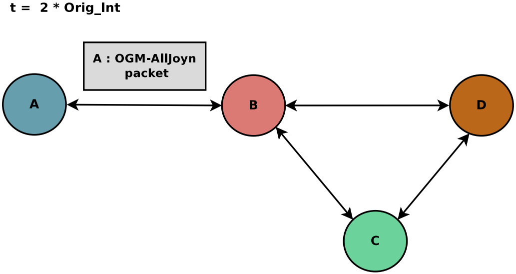


Figure 20: OGM-AllJoyn packet : Broadcast from A.

- OGM-AllJoyn packet broadcast from A at time t = 2 * Orig_Int.

Issues in Ask / Reply approach

New node arriving in the network will not come to know of the services on other nodes.

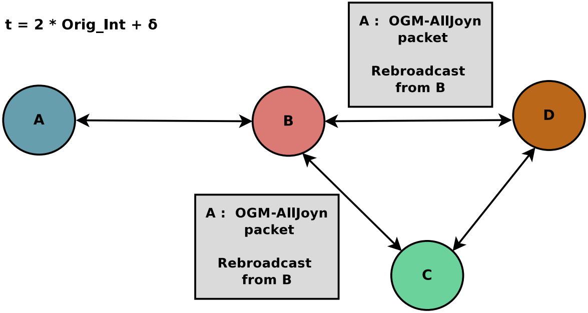


Figure 20: OGM-AllJoyn packet : Rebroadcast from B.

- Node B receives the service information for the first time.
- Node B rebroadcasts the OGM-AllJoyn packet as it is.
- Nodes C and D come to know of service on node A.

Issues in Ask / Reply approach

New node arriving in the network will not come to know of the services on other nodes.

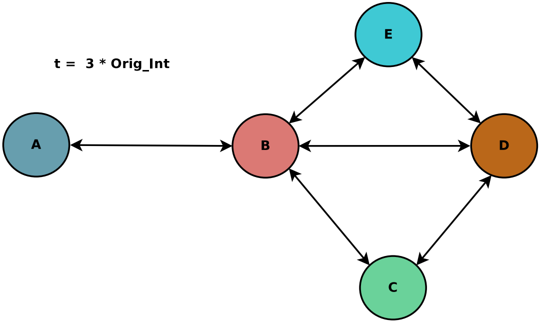


Figure 20: Arrival of a new node in the network.

- A new node E arrives into the network.

Issues in Ask / Reply approach

New node arriving in the network will not come to know of the services on other nodes.

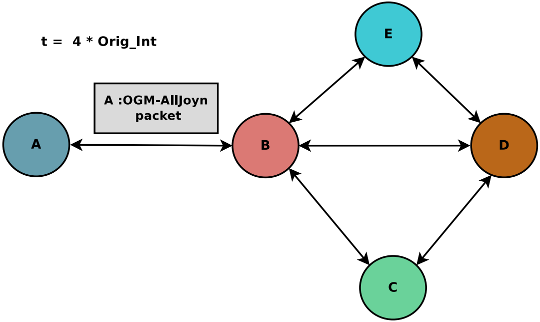


Figure 20: OGM-AllJoyn packet : Broadcast from A.

- A will send OGM-AllJoyn packet again at time t = 4 * Orig_Int.

Issues in Ask / Reply approach

New node arriving in the network will not come to know of the services on other nodes.

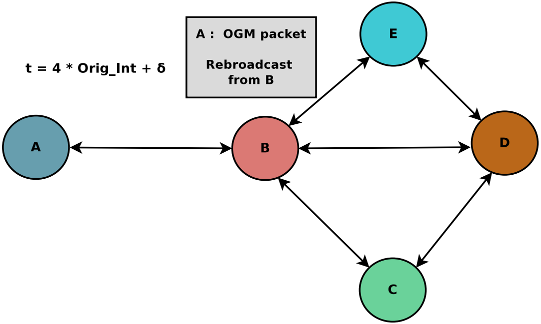


Figure 20: OGM-AllJoyn packet : Broadcast from A.

- B will not observe change in received service information.
- B will convert OGM-AllJoyn packet to OGM packet.
- New node E will not know of the service on A.

Remedy : Ask & Reply Packet

- Ask Packet is generated by a node (say X) when it wants to know of the service information on a other node (say Y).
- X comes to know of the existence of the Y through the OGM packets.
- However X does not have the service information of the service on Y.
- This is due to conversion of OGM-AllJoyn packet from Y into OGM packet by its single-hop neighbours or other intermediate nodes in the path(s) from Y to X.
- Ask packet is broadcast by a node to its neighbours.

Remedy : Ask & Reply packet

- If a neighbours has the service information corresponding to the MAC address in the Ask packet, it would generate a Reply packet.
- Reply packet would have service information corresponding to the node of which the service information is asked.
- Size of Ask packet in implementation is 22 bytes.
- Size of Reply packet in implementation is 30 bytes / 142 bytes based on whether it contains the service information in it.
- Reply packets are unicast.
- However node receiving Reply packets rebroadcast further to its neighbour.

Ask / Reply approach

Assume node D does not know of service on node A

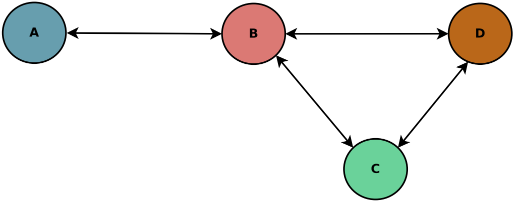


Figure 21: Topology of 4 nodes A - D.

Ask / Reply approach

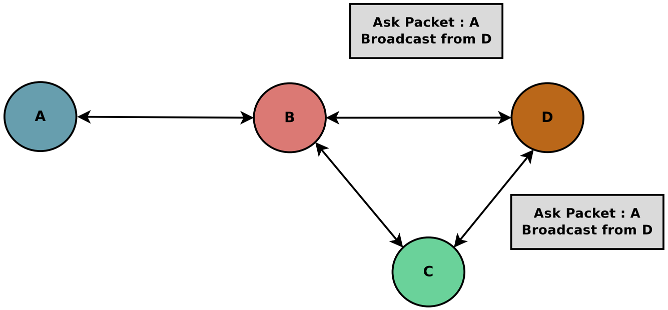


Figure 21: Ask Packet : Broadcast from B.

- Node D wil generate Ask packet asking about service on node A.
- Broadcast would be received by nodes B and C.
- Nodes C and D come to know of service on node A.

Ask / Reply approach

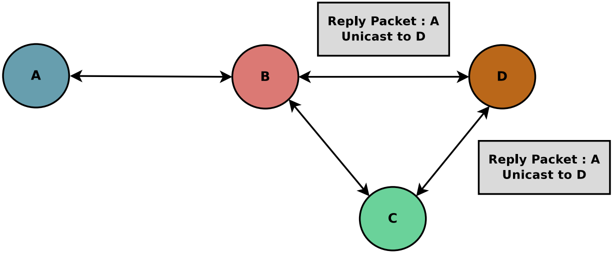


Figure 21: Reply Packets from B and C unicast to D.

- Nodes B and C unicast the Reply packet to node D.
- If they do not have service information, a Reply packet of 30 bytes would be generated.

Ask / Reply approach

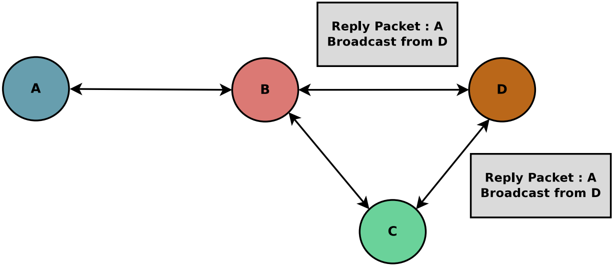


Figure 21: Reply packet : Broadcast from D.

- Node D observes a change from the received service information
- Node D will rebroadcast the Reply packet.
- This will stop when it reaches a node which does not observe a change from the received service information or TTL expires.

Ask / Reply approach : Summary & Drawbacks

- Optimization on Naive and Improved approach.
- Nodes send OGM-AllJoyn packet & OGM packet in alternate Orig_Int.
- A node will forward received OGM-AllJoyn packet as it is if it observes a change in the received service information.
- Accordingly OGM-AllJoyn packet would be converted to OGM packet.
- Has overhead of Ask & Reply packet but is a one time overhead.
- Significant less overhead from mere rebroadcast of service information.

Outline

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Multi-hop communication between services

- File named appData is created in the debugfs file system.
- It is created at the following path : /sys/kernel/debug/batman adv/bat0 when B.A.T.M.A.N routing is enabled.
- B.A.T.M.A.N routing module reads the data from the file when it is written by AllJoyn router and vice versa.
- Appropriate read / write call backs handle read / write mechanism.
- A layer 2 packet is formed by the B.A.T.MA.N routing module.
- It is routed through the B.A.T.M.A.N routing tables at the nodes.

Multi-hop communication between services

- AllJoyn router writes the application data into the 'appData' file.
- Along with it, it writes
  1. IP address & MAC address at which provider service is existent.
  2. Port at which provider service is running.
  3. Port at which consumer service is connecting to the provider service.
- B.A.T.M.A.N routing module would detect write operation by AllJoyn router on the file appData through the write callback.
- Using the application data and the other provided details it will form a packet called BATMAN-IP packet.
- Protocol field in the Ethernet header of BATMAN-IP packet is set as ETH_P_BATMAN.

Multi-hop communication between services

Assume node D does not know of service on A

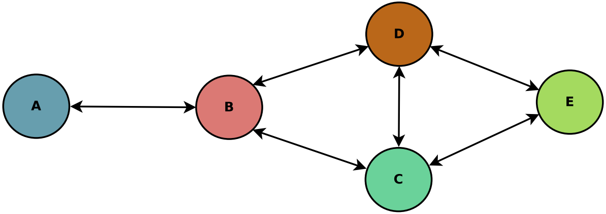


Figure 22: Topology of 5 nodes A - E.

Multi-hop communication between services

- Assume service at node A wants to communicate with that at node E.
- AllJoyn router at node A would write the data and the other details.
- BATMAN-IP packet among other fields contain destIP and destMAC.
- Here destIP = eIP and destMAC = eMAC.
- Corresponding to destMAC, routing module at node A would determine next hop.
- Here it would be node B and so Ethernet destination would be node B's MAC.

Multi-hop communication between services

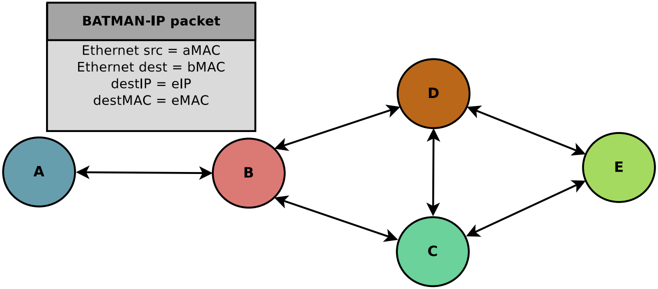


Figure 22: BATMAN-IP packet sent from A to B.

- Ethernet source is node A's MAC address and destination is node B's MAC address.

Multi-hop communication between services

- Routing module at node B would inspect the packet type.
- BATMAN-IP packets have been given a type in the implementation.
- Packet would be discard if the two values does not match.
- Then module at node B would obtain the value of destIP field (Here eIP).
- Clearly eIP != bIP.
- Next it will inspect the destMAC field (Here eMAC).
- It will then obtain next hop towards node E from its routing table.
- Let it be node D. Node B would send the BATMAN-IP packet to node D.

Multi-hop communication between services

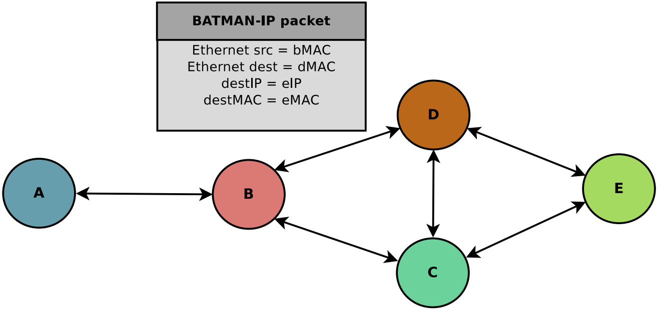


Figure 22: BATMAN-IP packet sent from B to D.

- BATMAN-IP packet originated from node A, send via node B.
- Ethernet source is node B's MAC address and destination is node E's MAC address.

Multi-hop communication between services

- Routing module at node D would do similar processing as done by module at node B.
- destIP in the received packet is eIP.
- eIP != dIP.
- It would then obtain the destMAC value.
- Module would then inspect the next hop towards node E.
- Here it is node E itself.
- BATMAN-IP will be sent to node E.

Multi-hop communication between services

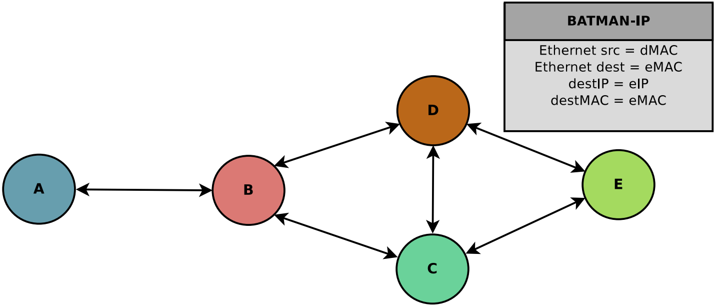


Figure 22: BATMAN-IP packet sent from D to E.

Multi-hop communication between services

- Routing module at node E would identify the packet is destined to it.
- It would write the details along with the application data to the 'appData' file.
- AllJoyn router would read the data written by the routing module.
- Details include the *destinationPort* as well.
- Router could destine the received data to the service running at that particular port.
- Communication in the reverse direction would proceed in a similar manner.

Control & Data packets in each approach

Approach	Control Packet	Size
Default approach	Default OGM packet	66 bytes
Naive approach	OGM-AllJoyn packet	182 bytes
Improved approach	OGM packet OGM-AllJoyn packet	70 bytes 182 bytes
Ask / Reply approach	OGM packet OGM-AllJoyn packet Ask packet Reply packet	70 bytes 182 bytes 22 bytes 30 / 142 bytes
Approach	**Data Packet**	**Size**
All approaches (except Default)	BATMAN-IP packet	$\leq$ 1514 bytes

Table 6: Control & Data packets in each approach.

Topologies

- Implementation approaches for B.A.T.M.A.N and AllJoyn integration are tested over real-time test-beds.
- The results are collected on a real-time test-bed with following topologies:
  1. 3 Node Topology
  2. 5 Node Topology
  3. 7 Node Topology

5 Node Topology

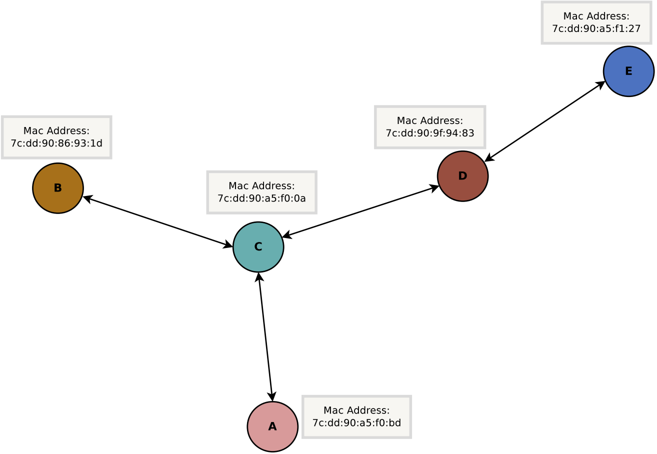


Figure 22: An ad hoc network set up between 5 nodes. Link denotes that corresponding nodes are one hop neighbour to each other.

7 Node Topology

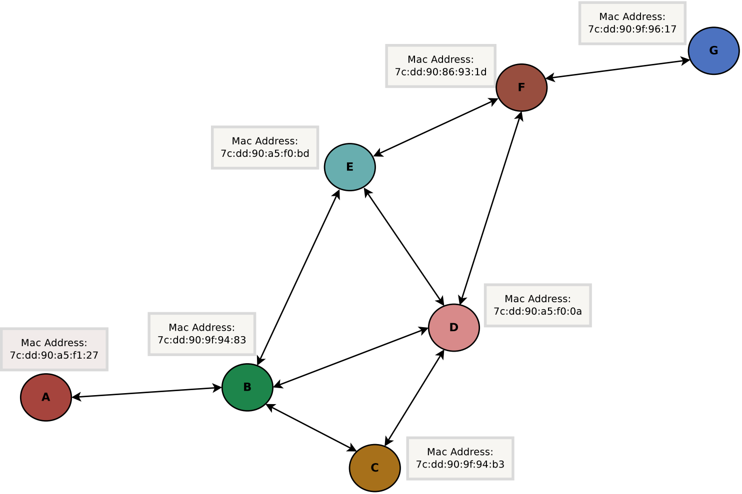


Figure 23: An ad hoc network set up between 7 nodes. Link denotes that corresponding nodes are one hop neighbour to each other.

Device Configurations

Nodes in the previous figures are one of the following:

- Linux Machine (Desktop/Laptop)
- Raspberry PI

	Raspberry PI	**Linux Machine I**	**Linux Machine II**
OS	Raspbian OS	Ubuntu 14.04	Ubuntu 15.10
CPU Architecture	ARM Cortex A7	Intel i5	Intel i5
RAM	1 GB	4 GB	2 GB
No. of Cores	Quad Core	Quad Core	Dual Core
CPU Freq.	900 MHz	2.70 GHz	2.50 GHz

Table 7: Hardware and Software specifications of the devices used in the real-time test-bed.

Wireless Network Configuration

Wi-Fi Standard	IEEE 802.11n
Wi-Fi Mode	Ad Hoc
Wi-Fi Channel	6
IP Subnet	10.10.20.0 / 24
Tx Power	20 dBm
Frequency	2.437 GHz

Table 8: Specifications of the wireless network configuration in real-time test-bed.

- External Wi-Fi adapter (ODROID Wi-Fi Module 4) is used.
- Operational specifications of the Wi-Fi network set up are shown in the above table.
- Certain links had interference from the regular Wi-Fi operating at channel 6.

Outline

Batctl

- Batctl is a debugging and configuration tool for B.A.T.M.A.N [4].
- While using it for configuration, one can add or remove interfaces using B.A.T.M.A.N.
- Change parameters such as originator interval (Orig_Int).
- It can be used for debugging as well where it has utilities such as *tcpdump*.
- In order to capture various control and data packets a packet sniffing tool is needed.
- We incorporate a module called Logger module 7 into batctl code obtained from following source [5].

Logger Module

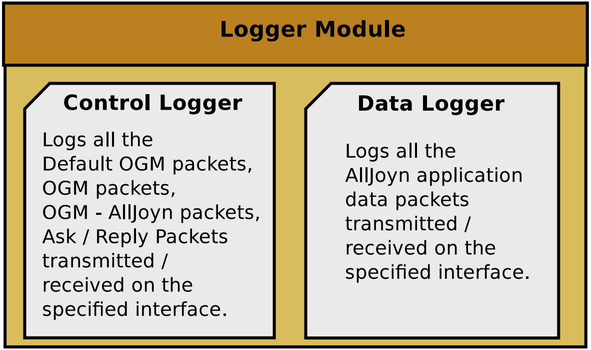


Figure 24: Logger Module incorporated into the Batctl implementation.

Logger Module

- Control Logger shown in Figure 24 has 2 parts :
    1. **controlSend logger:** Logging all the transmitted control packets.
    2. **controlReceive logger :** Logging all the transmitted control packets.
- Accordingly Data Logger also has 2 parts :
    1. **dataSend logger:** Logging all the transmitted data packets.
    2. **dataReceive logger :** Logging all the transmitted data packets.
- Different approaches have have different control packets.
- Corresponding to 4 approaches there are 4 batctl implementations (batctl - I to batctl- IV).

Outline

Service Discovery Time : Definition and Notation

Definition

Service Discovery Time is the time after which a node comes to know of the services on an another node from the time of the arrival of the node which entered later into the network among the two nodes.

- Consider any two nodes 'i' and 'j' in a particular network.
- Service discovery time for node i to know about a particular service, say 'x1' on node j is denoted as $SD[i][j_{x1}]$.

Service Discovery Time : Notation

- $SD[i][j_{x1}]$ is computed using the following equation:
  $\mathbf{SD[i][j_{x1}] = T_{iFj}[x1]}$ - $\mathbf{maxArr(arrT_i, arrT_j)}$, where
  1. $T_{iFj}[x1]$ = Time at which node 'i' first comes to know about a service 'x1' on node 'j'.
  2. $arrT_i$ = Time at which nodes 'i' arrives in the network.
  3. $maxArr(t_a,t_b)$ = A function that returns maximum of the time value between t_a and t_b.

- Extending the above equation for multiple services, following equation can be generated : $\mathbf{SD[i][j] = (SD[i][j_{x1}] + SD[i][j_{x2}] \ldots\ldots + SD[i][j_{xn}]) \,/\, n}$, where
  1. **SD[i][j]** denotes the service discovery time for node 'i' to know about all the services on node 'j' and
  2. **x1, x2 ..... xn** are the services available on node 'j'.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Service Discovery Time : Example

- Consider two nodes B and F in the 7 node topology in Figure 23.
- Let the time of arrival of node B in the network be 'x1' and that of F be 'x2' where x2 >x1.
- The time at which B comes to know of a certain service on F is z1 and time at which F comes to know of service on B is z2.
- Clearly z2 >(x+y) & z1 >(x+y).

Calculation

1. Service discovery time for B to know about service on F = (z1 - x2)
2. Service discovery time for F to know about service on B = (z2 - x2)

Network Service Discovery Time / Average Service Discovery Time

- Based on the previous notations and equations, we compute the network service discovery time or average service discovery time.
- Average service discovery time is denoted as $\mathbf{SD_N}$ and its value is determined through following steps :
  - Let V denote the set representing the devices in the network and W = V * V.
  - Compute SD[i][j] for every pair of device (i,j) ϵ W, where device i,j ϵ V.
  - Eventually we calculate $\mathbf{SD_N}$ through the formula :

$$\mathbf{SD_N} = \frac{\Sigma_{\forall (i,\, j)\ \epsilon\ \mathrm{W}}\ SD[i][j]}{|W| - |V|}$$

  - $|W|$ and $|V|$ denotes the size of set W and V respectively.

- Consider the 5 node topology shown in the Figure 22.
- Set V for the 5 node topology can be represented as V $=$ $\{$A,B,C,D,E$\}$.
- Accordingly the set W can be calculated W $=$ V * V.
- Compute SD[i][j] for every pair (i,j) ϵ W.
- Consequently $\mathbf{SD_N} = \dfrac{\boldsymbol{Z}}{\mathbf{25-5}}$ as $|W| = 25$ and $|V|$ for the 5 node topology.
- $Z = \Sigma_{\forall (i,j)\ \epsilon\ \mathrm{W}} SD[i][j]$

Control Overhead

- Control Overhead is the amount of traffic generated from the control packets.
- Different approaches would have different control overheads based on their functioning and the size of control packets.
- In the Default approach control overhead would be just from the operation of B.A.T.M.A.N routing.
- In the Naive, Improved and Ask / Reply approach, control overhead would be from the operation of B.A.T.M.A.N routing as well from the advertisement of AllJoyn service information.
- Ask / Reply approach would have additional control overheads from the generation of Ask and Reply packets.

Network Throughput

- Network Throughput is the rate of successful message delivery over a communication channel.
- Here, Network Throughput is determined by the amount of control traffic generated during the transmission of certain application data.
- Lot of control traffic would have impact on the the data traffic.
- Approach in which less control traffic is generated would yield higher network throughput.
- We define it as amount of application data transmitted in the network per unit time (Kbps / Mbps).

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Node Arrival Time

- In the experimental set up we provision the arrival of nodes in the network at different times.
- Arriving of a node in the network means the B.A.T.M.A.N routing is enabled on a certain interface on the node. (in our case wlan0).

Reason behind setting different arrival times

- If all the nodes arrive at the same time there would be no generation of Ask and Reply Packets.
- Different arrival times would help in determining the overhead from Ask / Reply packet as well as the impact on service discovery time.

Outline

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Average Service Discovery Time

- Based on the B.A.T.M.A.N approach being tested corresponding batctl implementations are run (batctl - I to batctl - IV).
- 5 experiments are conducted with every approach at different originator intervals on all the topologies.
- 'controlReceive' logger at different devices is run to capture the control packets.
- Based on the receival times logged in the 'controlReceive' logger at different nodes, value of SD[i][j] and hence SD_N is determined.
- SD_N values in different experiments for a certain approach and a particular originator interval is averaged and a final value is determined.

Average Service Discovery Time : Topology I (3 nodes)

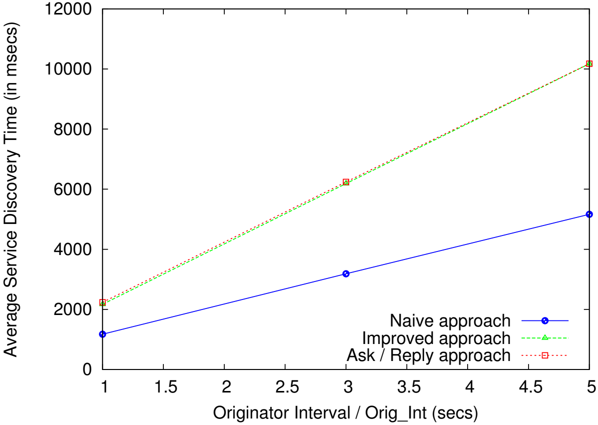


Figure 24: Average Service Discovery Time (**SD_N**) for 3 node topology at Originator interval of 1, 3 and 5 secs.

Average Service Discovery Time : Topology II (5 nodes)

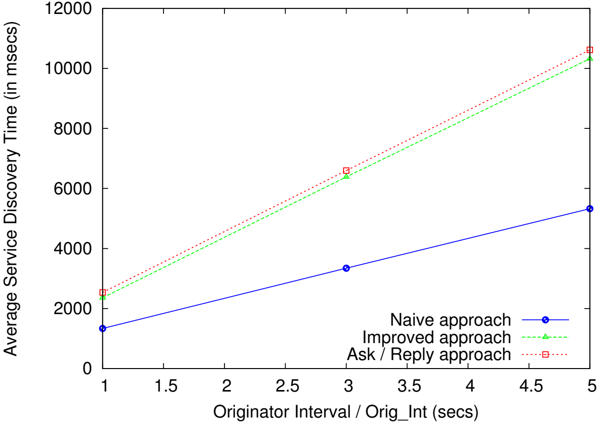


Figure 25: Average Service Discovery Time (**SD_N**) for 5 node topology at Originator interval of 1, 3 and 5 secs

Average Service Discovery Time : Topology III (7 nodes)

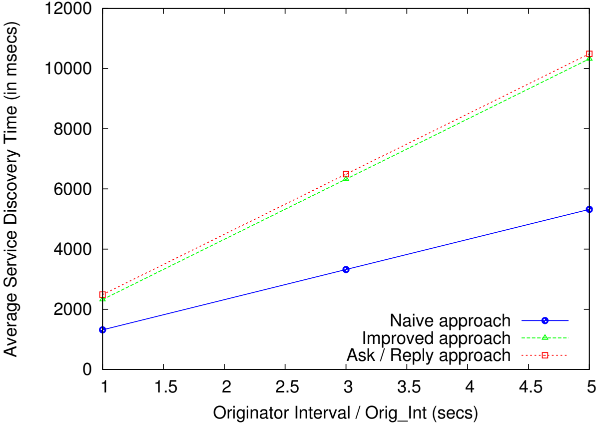


Figure 26: Average Service Discovery Time ($\mathbf{SD_N}$) for 7 node topology at Originator interval of 1, 3 and 5 secs.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Average Service Discovery Time

- SD_N is observed to be double in Improved approach in comparison to Naive approach.
- Ask / Reply approach has even higher SD_N as compared to Improved approach.
- This is due to generation of Ask / Reply packets and hence a higher SD_N.
- With the increase in Orig_Int, a proportional increase in the SD_N is observed across all approaches.
- SD_N values for 3 node topology are comparatively lower.
- This is due to smaller topology size.

Average Service Discovery Time

- Difference between SD_N values for Improved & Ask / Reply approach is too low (57 msecs) in case of 3 node topology.
- However between Improved & Ask / Reply approach difference in SD_N is observed (292 msecs) in 5 and 7 node topology compared to 3 node topology.
- Not much difference in SD_N values is observed between 5 and 7 node topologies.
- This can be attributed to the compact topology in case of 7 nodes as compared to a linear topology in case of 5 nodes.

Control Overhead

- At a time kernel module corresponding to a certain approach is loaded across all the nodes.
- 5 experiments are performed with Orig_Int of 1 sec, 3 secs and 5 secs each, with Default approach on a particular topology.
- B.A.T.M.A.N routing is made operational for a duration of 10 mins in each experiment.
- 'controlSend' logger at all the nodes is run to determine the control information being sent.
- Same exercise is repeated for all the approaches, across all the Orig_Int (1, 3 & 5 secs) and on every topology.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Control Overhead Comparison : Topology I (3 Nodes)

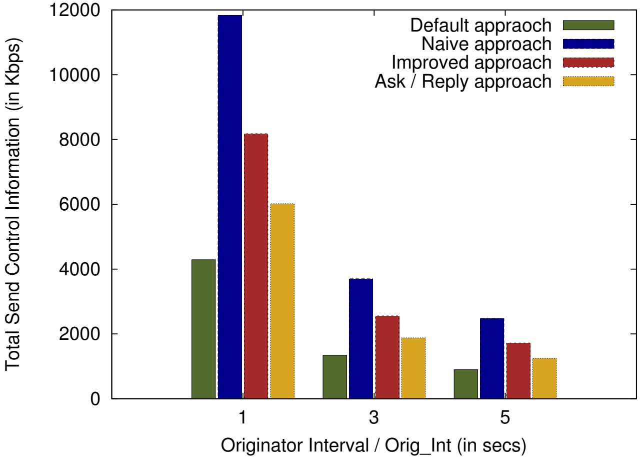


Figure 27: Control information send by the nodes in the network at Originator Interval of 1, 3 and 5 secs.

Control Overhead Comparison : Topology II (5 Nodes)

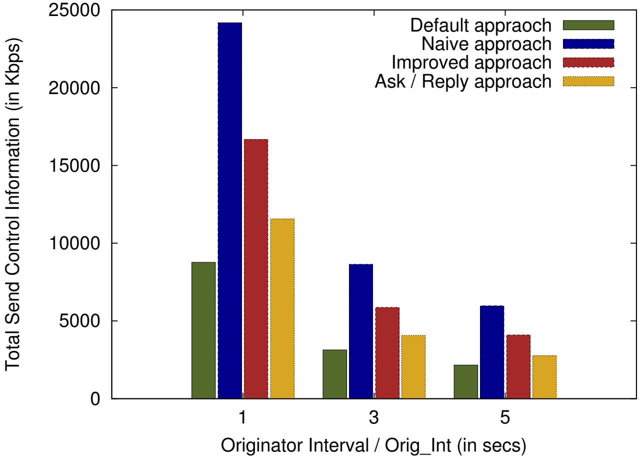


Figure 28: Control information send by the nodes in the network at Originator Interval of 1, 3 and 5 secs.

Control Overhead Comparison : Topology III (7 Nodes)

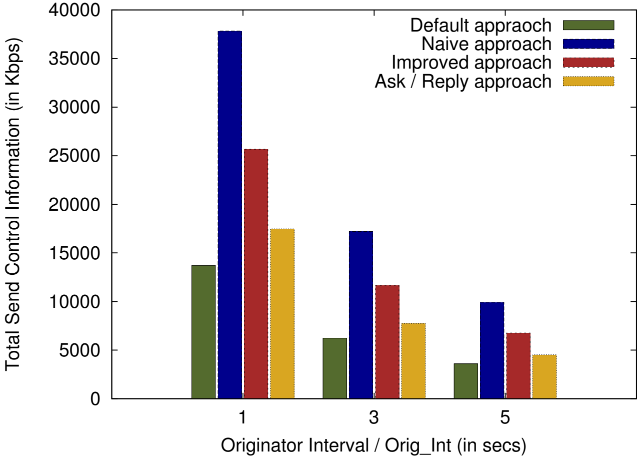


Figure 29: Control information send by the nodes in the network at Originator Interval of 1, 3 and 5 secs.

Ask and Reply Packet Overhead in Ask / Reply approach

- Topology I

Total Ask & Reply Packet send	6
Total Ask & Reply Packet send size	492 bytes

- Topology II

Total Ask & Reply Packet send	20
Total Ask & Reply Packet send size	1776 bytes

- Topology III

Total Ask & Reply Packet send	80
Total Ask & Reply Packet send size	7944 bytes

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Control Overhead

	Improved approach	Ask / Reply approach
Topology I	30.9 %	49.1 %
Topology II	31 %	52.2 %
Topology III	32.1 %	53.8 %

Table 9: % control overhead reduction in comparison to Naive approach at Orig_Int = 1 sec.

	Improved approach	Ask / Reply approach
Topology I	30.9 %	49.2 %
Topology II	32 %	52.8 %
Topology III	32.1 %	54.9 %

Table 10: % control overhead reduction in comparison to Naive approach at Orig_Int = 3 secs.

Control Overhead

- From Tables 9 & 10, it can be observed that there is significant control overhead reduction in Improved & Ask / Reply approach.
- Reduction is proportionate with increase in Orig_Int to 3 secs.
- Also the reduction increases with increase in number of nodes for each Orig_Int.
- Ask & Reply approach though has overhead of Ask & Reply packets but this is not very high.
- This can be concluded from the difference in control overhead b/w Ask / Reply & Improved approach.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Network Throughput

Topology II (5 nodes) : Following 3 flows, each of 10 MB are created.

1. C - A
2. B - E
3. D - B

Topology III (7 nodes) : Following 3 flows, each of 10 MB are created.

1. E - C
2. A - G
3. F - A

- X - Y denotes application data flow from service at X to service at Y.
- 5 experiments are conducted for each of above topology and average value is determined.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Network Throughput

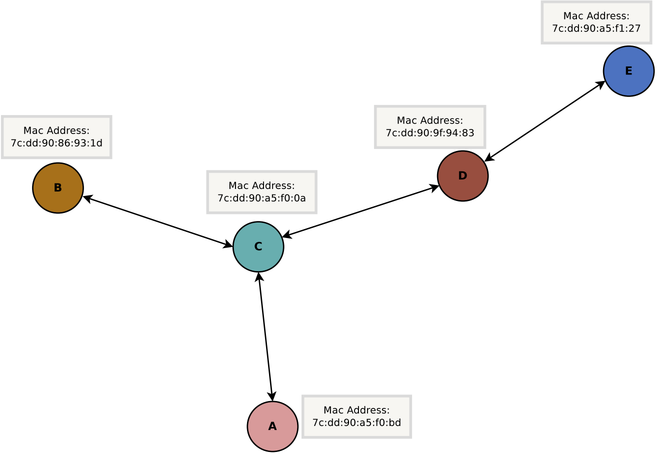


Figure 30: An ad hoc network set up between 5 nodes. Link denotes that corresponding nodes are one hop neighbour to each other.

Network Throughput

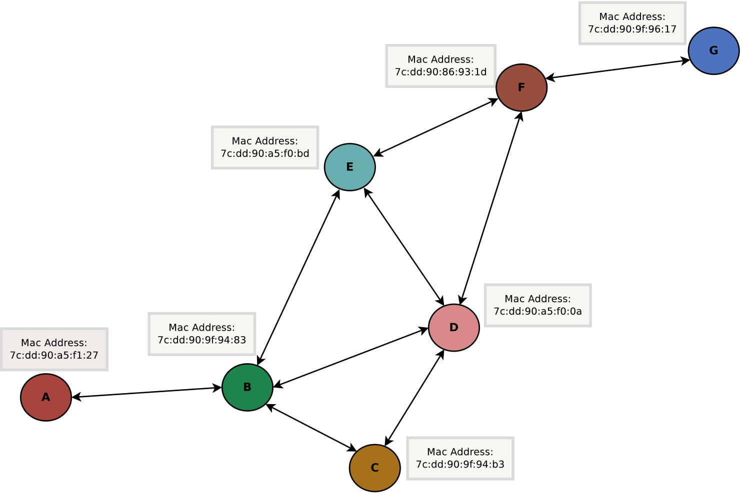


Figure 30: An ad hoc network set up between 7 nodes. Link denotes that corresponding nodes are one hop neighbour to each other.

5 Node Topology

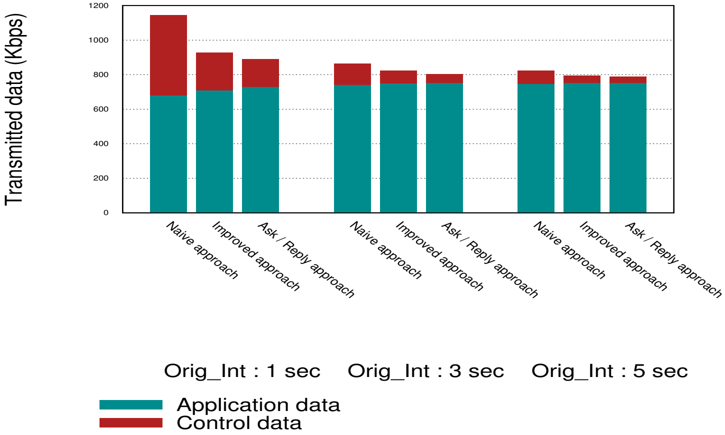


Figure 31: Network Throughput : Effect of control overhead in different approaches on transmission of application data.

Network Throughput

7 Node Topology

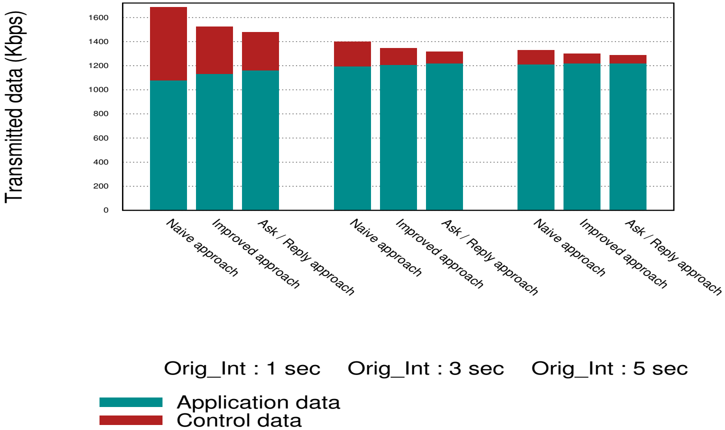


Figure 32: Network Throughput : Effect of control overhead in different approaches on transmission of application data.

Network Throughput

	Improved approach	Ask / Reply approach
Topology II	4.37 %	7.17 %
Topology III	5.15 %	7.95 %

Table 9: % Network Throughput improvement in comparison to Naive approach (Orig_Int = 1 sec)

	Improved approach	Ask / Reply approach
Topology II	.95 %	1.6 %
Topology III	1.12 %	1.9 %

Table 10: % Network Throughput improvement in comparison to Naive approach (Orig_Int = 3 sec)

Network Throughput

	Improved approach	Ask / Reply approach
Topology II	0.56 %	0.73 %
Topology III	0.71 %	0.83 %

Table 9: % Network Throughput Improvement in comparison to Naive approach (Orig_Int =5 sec)

- High control overhead results in drop of application data packets and hence a lower network throughput.
- Table 9 represents network throughput improvement in Improved & Ask / Reply approach.
- Less control overhead in this 2 approaches leads to higher network throughput.

Network Throughput

- Little higher improvement is observed in case of 7 node topology.
- This is because it has more control overhead in Naive approach.
- With increase in Orig_Int percentage improvement drops.
- This is because Naive approach too has less control overhead with increase in Orig_Int.
- Significant differences in the N/W throughput can be observed even at high Orig_Int if the network is heavily loaded.
- In that case high control traffic would impact the data transmission.

Trade Off between the metrics

- Clearly there exists a trade off between control overhead and service discovery time.
- Trade off will scale with the increase in number of nodes.
- Low service discovery time comes at the cost of high control overhead.
- Improved & Ask / Reply approach are better alternatives at low Orig_Int if low service discovery time is not a requirement.
- Higher Orig_Int and Naive approach can be a good combination too, if a little high service discovery time is acceptable.

Conclusions

- Integration approaches for realization of Multi-hop service advertisement in the AllJoyn framework.
- Multi-hop communication between AllJoyn services leveraging the BATMAN-IP packets.
- Incorporation of logger modules into the batctl implementation.
- Developed enhancements to the framework are tested on real-time test-bed of laptops, computer and Raspberry PI.
- Realization of communication between services existent on nodes single-hop or multi-hops on the test-bed.

Conclusions

- Comparison between approaches is done on identified metrics : service discovery time, control overhead & network throughput.
- Trade off between service discovery time and control overhead is observed.
- Less discovery time is obtained at the cost of high control overhead.
- Control overhead impacts application data transmission as well.
- Ask / Reply approach emerges to be a better alternative in terms of less control overhead.
- It can be leveraged in applications where less discovery time is not a requirement.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Future Work

- Multi-hop realization through other routing algorithms :
  1. Integration with other routing algorithms such as OLSR, AODV, etc.
  2. A thorough evaluation can then be performed across these implementations.
  3. Classification could be done of which integration would be a better alternative in different application scenarios of AllJoyn framework.
- Support for short range technologies such as Bluetooth, Wi-Fi Direct and ZigBee :
  1. Technologies such as Bluetooth, Wi-Fi Direct are readily available in devices these days.
  2. Support for ZigBee protocol in AllJoyn framework will lead to utilization of the benefits of both the technologies.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Visible Research Outcomes

[1] Hatim Lokhandwala, Srikant Manas Kala and Bheemarjuna Reddy Tamma, "Min-O-Mee: A Proximity Based Network Application Leveraging The AllJoyn Framework" in Proc. of IEEE CoCoNet (Special Session on Mobile Social Networks), Decemeber 2015, Trivandrum, India.

[2] To be submitted (Journal) : Hatim Lokhandwala, Srikant Manas Kala, Bheemarjuna Reddy Tamma and Antony Franklin, AllJoyn Framework: Comprehensive analysis, Application prototype, Open challenges and Realized extensions.

[3] Research proposal titled : "MAPS: A Mobile Ad hoc P2P Social Network for Disaster Scenarios with Disrupted Network Infrastructure" submitted to IMPRINT (**IMP**ACTING **R**ESEARCH IN **IN**NOVATION AND **T**ECHNOLOGY). Short-listed in initial screening round and would be submitted for the second screening round.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

References I

[1] J. Gebert and R. Fuchs, Probabilities for opportunistic networking in different scenarios in Future Network Mobile Summit (FutureNetw), 2012. IEEE, 2012, pp. 18.

[2] M. Nekovee and B. B. Bogason, Reliable and efficient information dissemination in intermittently connected vehicular ad hoc networks, in Vehicular Technology Conference, 2007. VTC2007- Spring. IEEE 65th. IEEE, 2007, pp. 24862490.

[3] D. Waitzman, S. Deering, and C. Partridge, Distance vector multicast routing protocol, 1988.

[4] M. H. Linus Lssing. Batctl Usage 2016. https://www.open-mesh.org/projects/batman-adv/wiki/Using-batctl.

[5] M. H. Linus Lssing. B.A.T.M.A.N and Batctl source code 2016. https://www.open-mesh.org/projects/open-mesh/wiki/Download.

[6] D. Johnson, N. Ntlatlapa, and C. Aichele. A simple pragmatic approach to mesh routing using BATMAN.

[7] L. Barolli, M. Ikeda, G. D. Marco, A. Durresi, and F. Xhafa. Performance Analysis of OLSR and BATMAN Protocols Considering Link Quality Parameter 307314.

[8] A. Sharma and N. Rajagopalan. A Comparative Study of BATMAN and OLSR Routing Protocols for MANETs. International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE) 2, (2013) 1317.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

References II

[9] R. Sanchez-Iborra, M. D. Cano, and J. Garcia-Haro. Performance Evaluation of BATMAN Routing Protocol for VoIP Services: A QoE Perspective. IEEE Transactions on Wireless Communications 13, (2014) 49474958.

[10] T. Honda, M. Ikeda, E. Spaho, M. Hiyama, and L. Barolli. Effect of Buildings in VANETs Communication: Performance of OLSR Protocol for Video Streaming Application 323327.

[11] A. S. Alliance. An AllJoyn case study about L.G. electronics 2016. https://allseenalliance.org/sites/default/files/case_study_lge.pdf?utm_source=case-studies&utm_medium=download&utm_content=lge&utm_campaign=Case%20Studies

[12] IETF. B.A.T.M.A.N advanced documentation 2016. https://www.open-mesh.org/projects/batman-adv/wiki/

[13] A. S. Alliance. AllJoyn source code 2016. https://allseenalliance.org/framework/download.

[14] A. S. Alliance. The AllJoyn Framework - System Description 2016. https://allseenalliance.org/framework/documentation/learn/core/system-description/.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

THANK YOU. QUERIES?