Component Based Channel Assignment in Single Radio, Multi-channel Ad-Hoc Networks

Presenter: Sandeep Kakumanu

Authors: Ramanuja Vedantham, Sandeep Kakumanu, Sriram Lakshmanan and Raghupathy Sivakumar

GNAN Research Group

School of Electrical and Computer Engineering

Georgia Institute of Technology

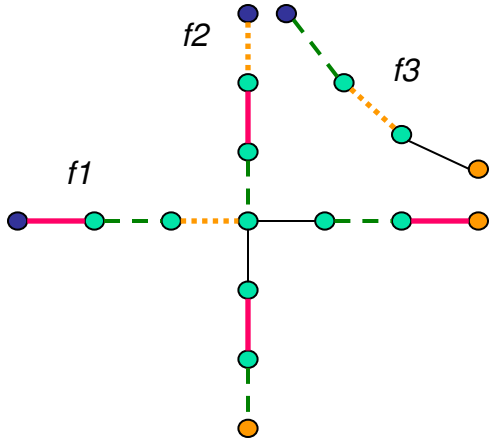
Preview: Context

- Channel assignment in multi-channel wireless networks can increase achievable throughput
  - Interference, varying channel characteristics, poor end-to-end characteristics
- Multi-channel, multi-hop wireless networks with single radio
- ☞ Channel Assignment: For each node, which channel should we operate at any given point in time?
- Granularity of assignment
  - Packet: Channel assignment on a per-packet basis [DCA'00]
  - Link: Channel assignment on a per-link basis [MMAC'04,SSCH'04]
  - Flow: All packets in a flow are sent along the same channel [MCP'05]
  - ☞ Component: Channel assignment on a component basis

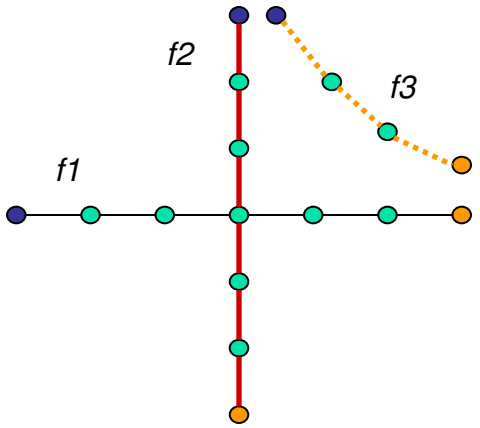
What is the ideal granularity for doing channel assignment, and how to achieve it?

Background: Link and Flow Based

- Link based channel assignment
  - Different links in the flow graph can operate on any of the available channels.
  - Different links in a flow can potentially be assigned to different channels.

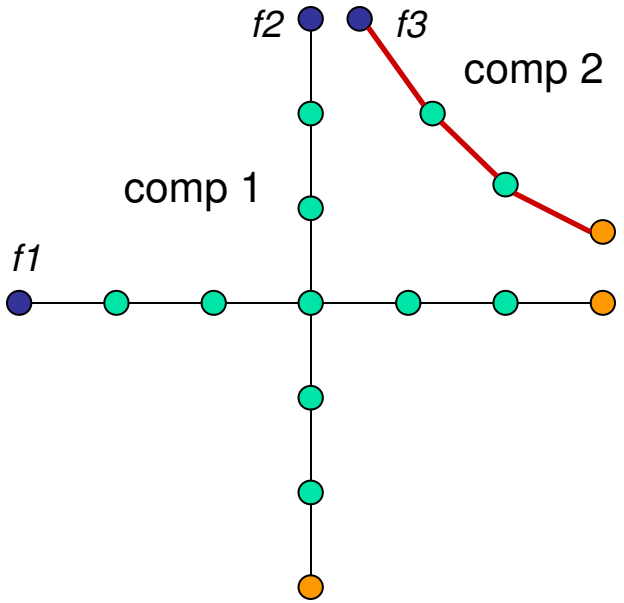


- Flow based channel assignment
  - Different flows in the flow graph can operate on any of the available channels.
  - All links in a single flow operate on the same channel.



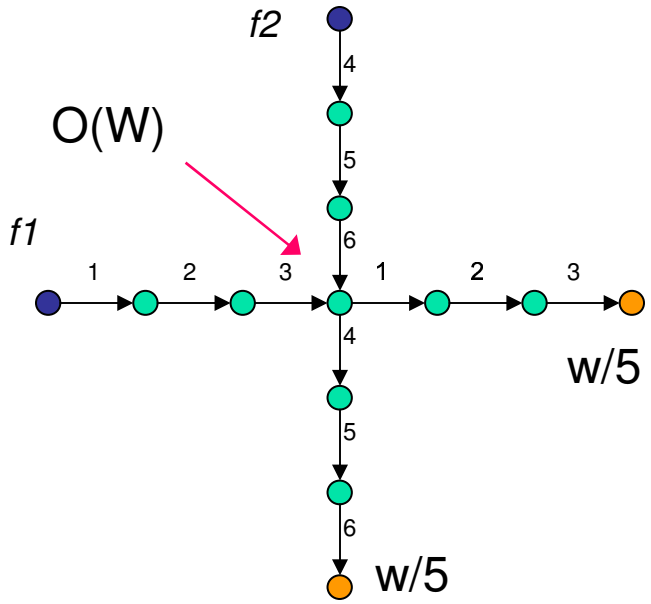
Component based Channel Assignment

- We introduce a new model for channel assignment known as Component-Based:
  - All links in a connected component induced by the underlying flow graph operate in a single channel.
  - However, different connected components can potentially operate on different channels.
  - Leverage the presence of multiple channels to increase spatial reuse at the granularity of a component.
  - Although the component based model looks simple, we show that this model can have equal if not better performance over link and flow based approaches.

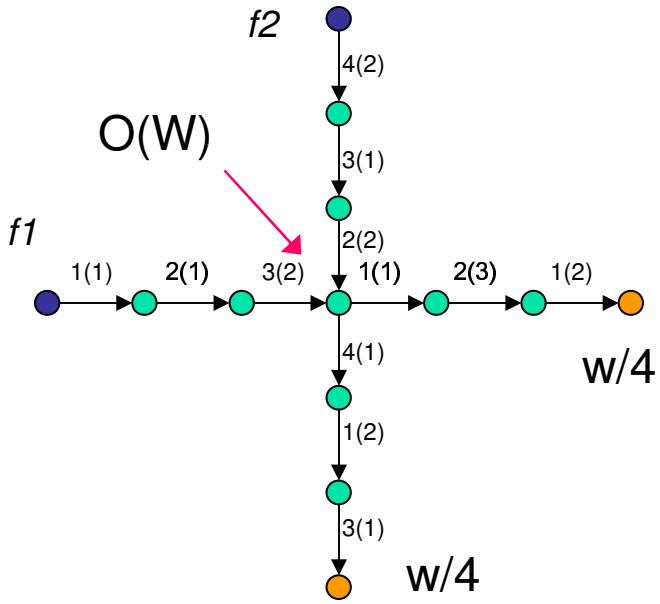


Motivation: Logical Reasoning

- Single Radio Bottleneck.
- Capacity under an ideal scheduling scheme.



Component based

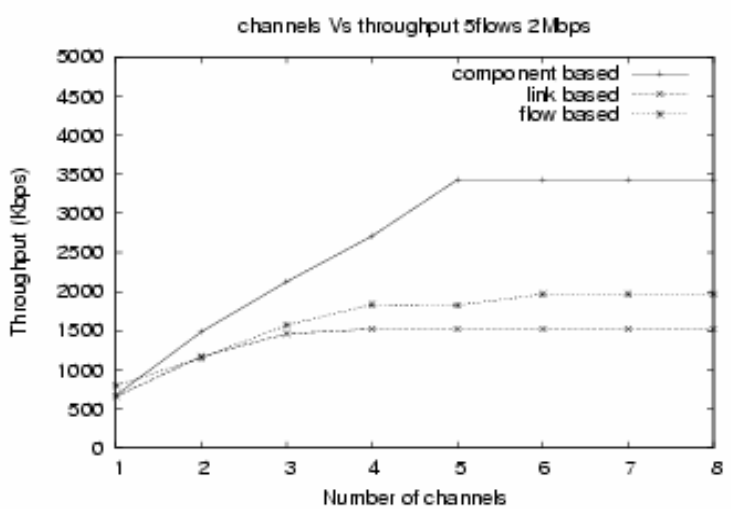


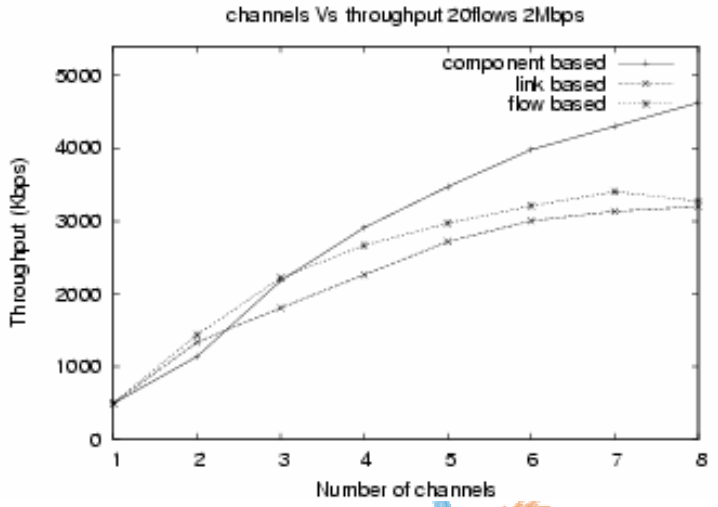
Link based

Toy Topology

Motivation: Quantitative Results

- Performance in a random network using simulations
- NS2 simulations
  - 100 nodes in 750mx750m square
  - Transmission range: 250m
  - Channel data rate: 2 Mbps
- From graphs 1, 2
  - Component based shows minimal degradation in throughput
  - Flow and link based approaches saturate
    - *Switching delay*
    - *Lack of synchronization*
    - *Head of line blocking*





Motivation: Practical Considerations

- Hardware/ MAC changes
  - Link and flow based assignment require changes to MAC layer [MMAC'04].
  - Need for customized wireless cards to support new MAC layer functionality.
- Switching delay
  - Link and flow based require switching at intersecting links or flows.
  - Hardware switching delay: 80-100 μs [Herzel'03].
  - With software overheads it can be higher.
- Synchronization requirement
  - When a common node serving two links (or flows) switches to another channel
    - Sender/receiver for new link should be on the same channel.
    - Sender of old link should not transmit for the duration of time spent in the other channel.
- Scheduling overheads
  - Common node informs the switching schedule to neighboring nodes .

Motivation: Analytical Results - Bounds

Variable	Description
W	Capacity of single channel
F	Total number of Flows
c	Number of Channels
Δ	Maximum number of contending flows
Γ	Maximum number of intersecting flows

(i) Notation

Type	Condition	Link LB	Link UB	Comp LB	Comp UB
NC	N/A	O(WF)	O(WF)	O(WF)	O(WF)
C	$\Delta \leq c$	O(WF)	O(WF)	O(WF)	O(WF)
C	$\Delta > c$	O(WFc/Δ)	O(W(c+F- Δ))	O(WFc/ Δ)	O(W(c+F- Δ)
I (NC)	N/A	O(WF/ Γ)	O(W(1+F- Γ))	O(WF/ Γ)	O(W(1+F- Γ))
I and C	$\Delta \leq c + \Gamma - 1$	O(WF/ Γ)	O(W(1+F- Γ))	O(WF/(Γ+ Δ))	O(W(1+F- Γ))
I and C	$\Delta > c + \Gamma - 1$	O(WF/ Γ)	O(W(C+F- Γ))	O(WF/(Γ+ Δ))	O(W(C+F- Γ))

(ii) Theoretical Upper and Lower Bounds

Centralized Approach

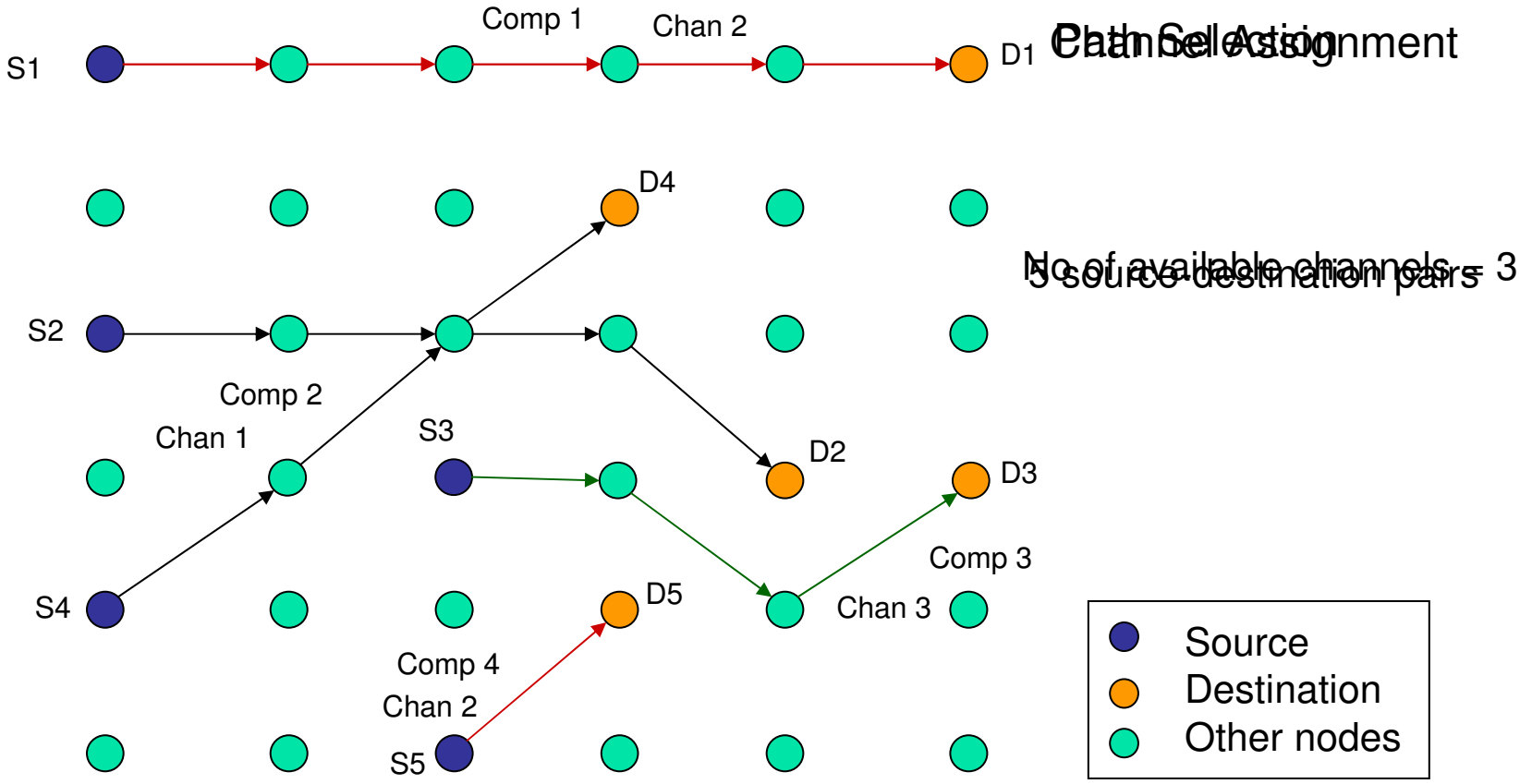
- Greedy centralized approach to do component-based assignment efficiently.
- Based on insights from theoretical analysis.
  - Capacity is inversely proportional to number of intersections.
  - Capacity is inversely proportional to level of contention.
- Algorithm has two phases:
  - Phase 1: Path selection – minimize the number of intersections in the network and form components.
  - Phase 2: Channel Assignment – minimize the contention level among different components.

Centralized Approach (...contd)

- Path Selection: Given S-D pairs, find the flow graph, component set
  - Compute k shortest paths for each S-D pair.
  - Cost of the path, $w(i)$ = sum of the weights of each node, i (node weight = 1 initially).
  - Path with the least cost is chosen.
  - Update weights for any chosen path to $w(i) = w(i) + \alpha$.

- Channel assignment: Given component set, determine the channel assignment
  - Compute total contention for a component: sum of pair-wise contention.
  - Compute channel contention: number of nodes assigned to that channel.
  - Choose component with maximum total contention
    - Assign to a channel with least channel contention.
    - Update channel contention level corresponding to the assigned channel.

Centralized Approach (Example)



Distributed Approach

- Distributed realization of the centralized algorithm.

- Path and channel selection are performed in an integrated fashion (8 phases) :
  1. Pre-preparation
  2. Route Request Broadcast
  3. Route Request Update
  4. Channel Selection
  5. Route Reply Propagation
  6. Component Update
  7. Route Maintenance
  8. Flow Termination

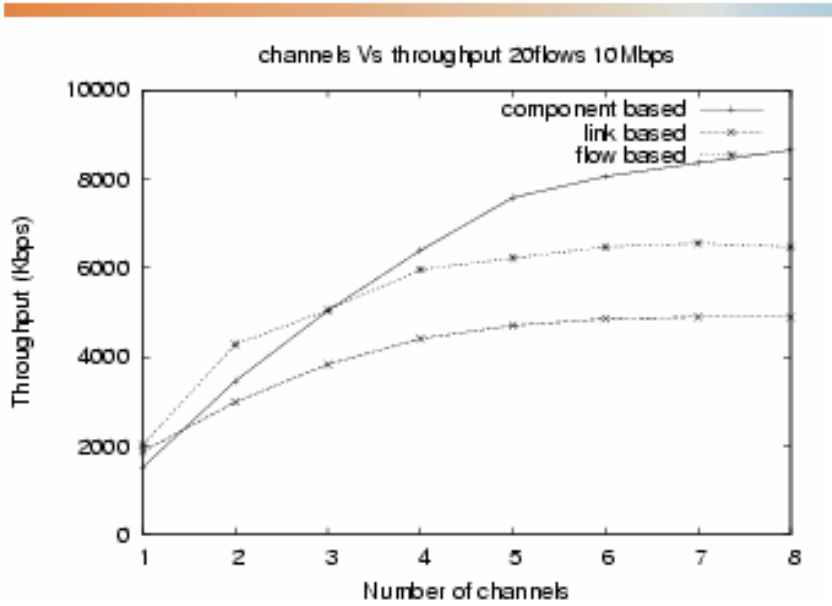
Distributed Approach (contd ...)

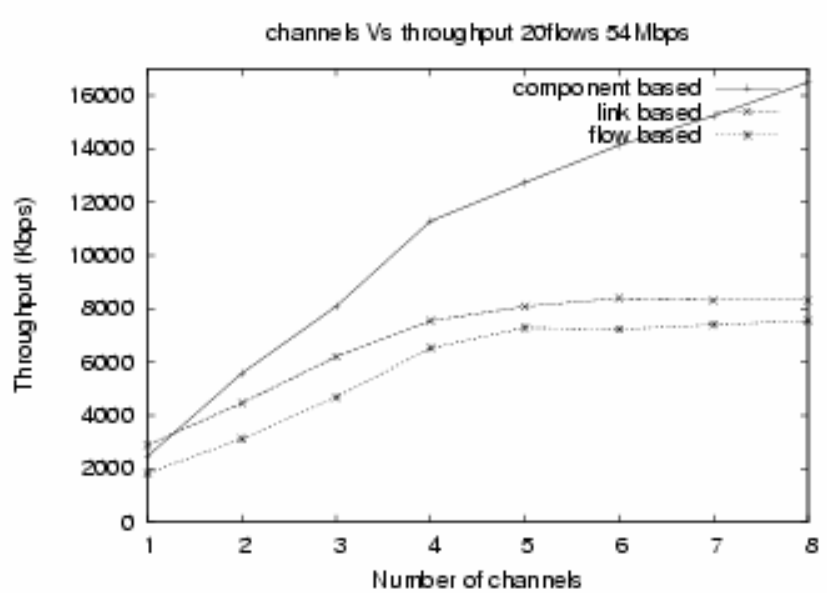
- Route Request Broadcast:
  - $RREQ()$ on all active channels by source.
- Channel Selection
  - Destination waits for some T_{RREQ} seconds or k RREQ() messages.
  - Destination selects path with minimum congestion and also decides the channel.
  - Path selection and channel assignment in centralized algorithm are preformed on for each S-D pair when required.
- Route Reply Propagation
  - Route Reply is sent on old active channel of receiving node.
  - As Route Reply propagates nodes in the route update new component information.
- Component Update
  - Update channel and component information for nodes.
  - Component broadcast by node, forwarding RREP(), to update channel and component information of other nodes in existing component.

Simulation Results: Setup

- Setup
  - NS2 simulator
  - 750mx750m grid with 100 nodes
  - Number of orthogonal channels: 1 to 8
  - Data rate: 10 Mbps, 54 Mbps
  - 20 flows, CBR over UDP
  - Switching delay: 100 μs
  - Routing protocol: Distributed algorithm for component, DSR for link and flow
  - Flow: MCP, Link: MMAC
- Metric
  - Throughput (Kbps)
  - Delay (sec)

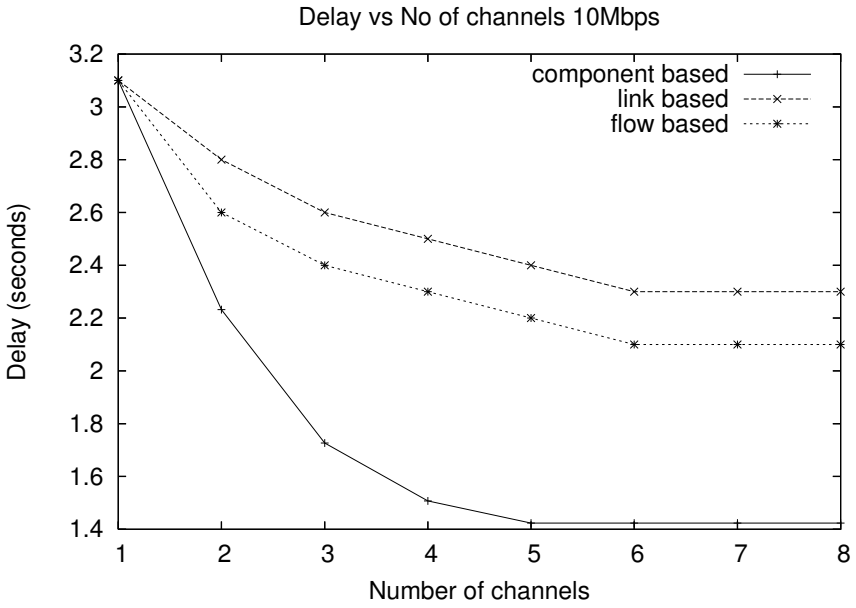
Simulation Results: Channel Rate

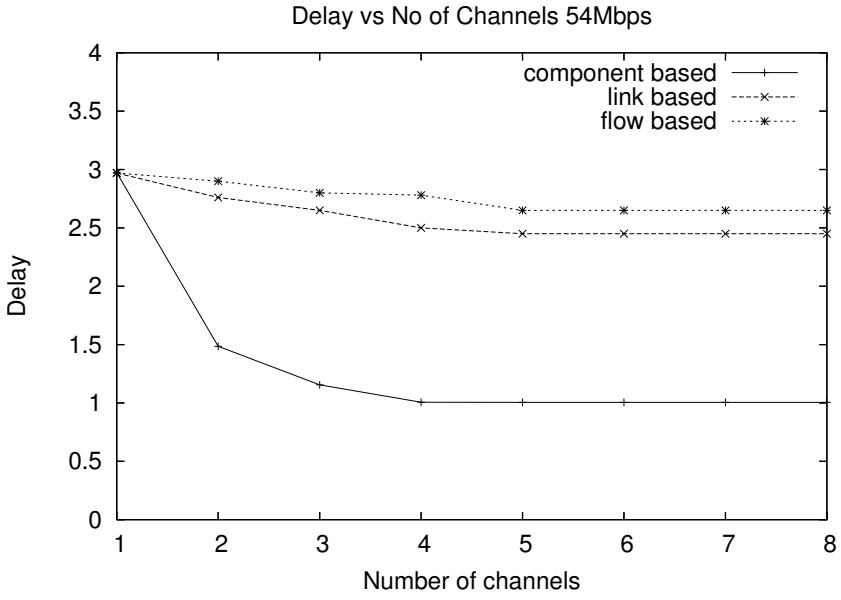




- Component based throughput increases with increasing number of channels (for both data rates)
  - Fewer intersections
  - No switching delay
  - No synchronization requirements
  - No scheduling overheads

Simulation Results: Delay





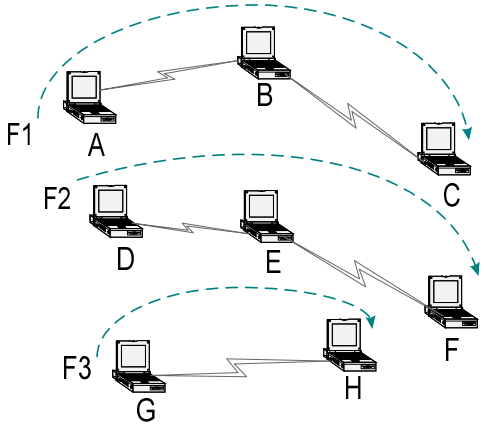
- Component based has decreasing end-to-end delay with increasing number of channels
  - No synchronization requirements
  - No head of line blocking
  - Decrease in intersections, contention

Testbed

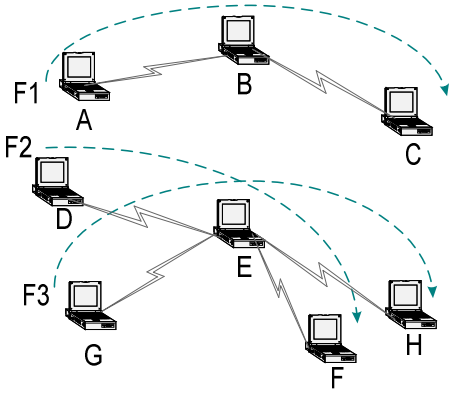
Setup

- 8 IBM and Dell laptops
- Lucent Ornico & Intel Pro wireless 2200 802.11b/g Wifi cards
- 3 laptops have FC 4 linux
- 5 laptops run on windows XP
- Ftp application

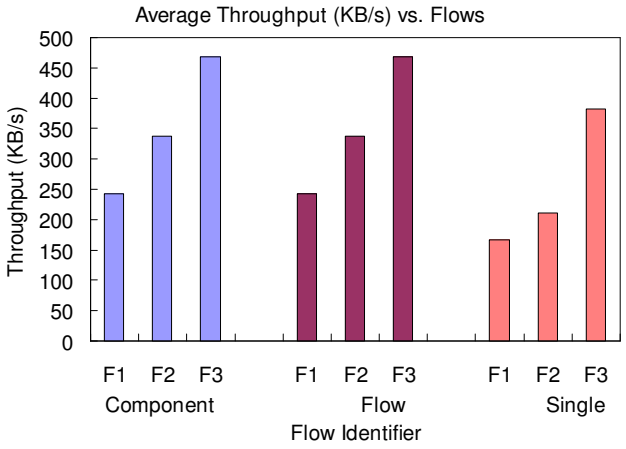
Testbed

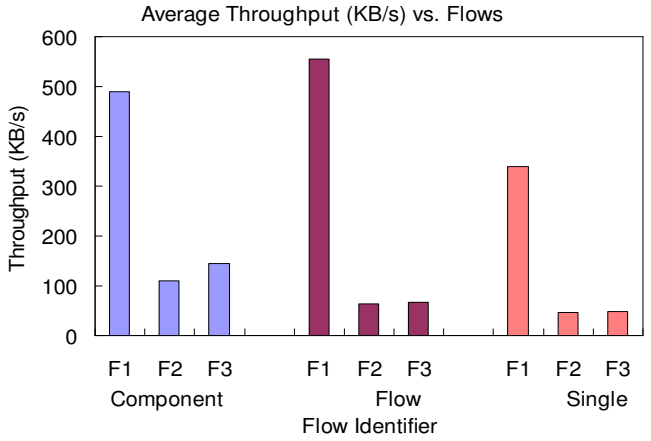


Topology 1



Topology 2





Summary

- Multiple channel usage does not automatically imply good performance in a single radio setting.
- Practical considerations greatly impact the performance of the type of channel assignment
  - Switching delay
  - Synchronization
  - Scheduling overheads
- Component based assignment performs well in most scenarios.
- Proposed centralized and distributed algorithms to perform efficient channel assignment in component-based.

Thank You.