A Wireless Flit-Based OpNET Model

Michael Buettner & Dirk Grunwald
Department of Computer Science
University of Colorado at Boulder
Boulder, CO 80309

CU-CS-1011-06 May 2006

University of Colorado at Boulder

Technical Report CU-CS-1011-06
Department of Computer Science
Campus Box 430
University of Colorado
Boulder, Colorado 80309

A Wireless Flit-Based OpNET Model

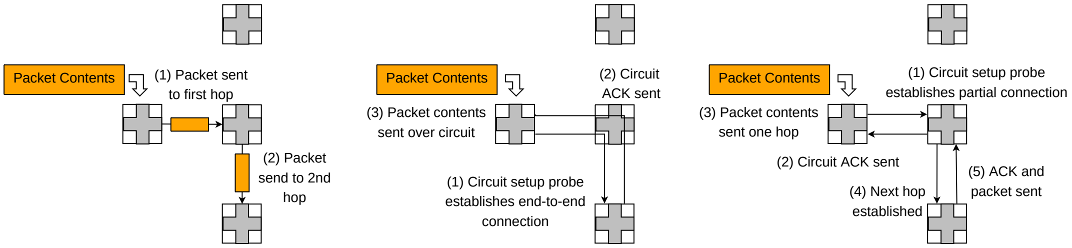
Michael Buettner & Dirk Grunwald
Department of Computer Science
University of Colorado at Boulder
Boulder, CO 80309

May 2006

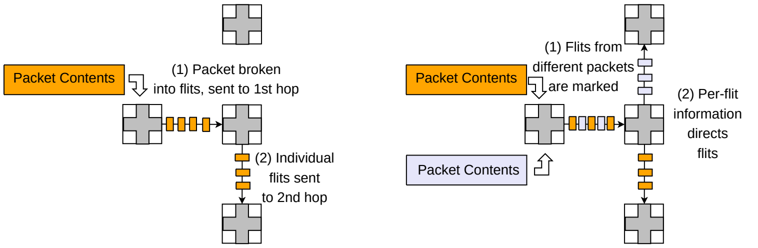
Abstract

Recent evaluation of *ad hoc* networks has shown that current techniques do not scale well with either network size or desired throughput. We argue that the greatest opportunity for improvement is in the integration and coupling of different network layers. To this end, we are building a “relay network” design in which end-to-end path resources are pre-allocated to avoid the cost of channel acquisition and contention at every hop. The routing, congestion control, and MAC functions are coupled – contention informs the routing and packet admission decisions, and demand determines link capacity allocation. We propose using spatial frequency division multiple access (SFDMA) to reduce interference and propose a forwarding layer using label-switching to allow a combination of SFDMA as well as statistical multiplexing. Normally, SFDMA methods are difficult to allocate due to a paucity of channels; we provision our network using Orthogonal Frequency Division Multiple Access (OFDMA) channels, avoiding re-synchronization delays on each packet.

One option for such networks is to use *wireless flits* rather than full packets for store-and-forward communication. This report describes an OpNET model that implements both those wireless flits and the traditional store-and-forward SFDMA networks.



(a) Packet Switched (b) Circuit Switched (c) Cut-Through Switching



(d) Flit-based (e) Flit-based with multiple packets

Figure 1: Switching Methods

1 Introduction

Wireless networks are one example of the more general class of *multi-stage interconnection networks*. These networks use intermediate switches to route messages from a source to a destination. In multicomputer systems, such networks typicaly have a regular structure [1, 2] resulting in different performance for different traffic distributions. To a first order, the interaction of *topology* and the *traffic distribution* of the network have the primary influence on network performance [3].

The *switching method* also influences performance and particularly latency. Figure 1 show schematic diagrams illustrating common switching techniques. The most common technique, shown in Figure 1(a), is packet switching. Individual switch nodes (shown with a cross in the diagram) contain buffer space. Messages are broken in fixed or variable sized "packets" and each packet is transmitted in sequence. Each packet contains addressing information and each packet can be individual switched. Each packet is received completely and then forwarded; this increases the latency a packet proportional to the number of hops or switches the message traverses. Classical Ethernet and wireless Ethernet (*e.g.* 802.11) are examples of packet switched networks.

The Internet is based on packet switching; earlier telephony networks and some computer networks were based on *circuit switching*. This process, shown in Figure 1(b), involves three steps. The first is a *circuit establishment* phase where a probe is used to reserve the entire path to be traversed by a packet. Signaling is then used to initiate the packet transfer which is then transmitted end-to-end without buffering. Circuit switching is felt to be more complex than packet switching because it requires *deadlock free* paths – each circuit setup holds and reserves resources and thus the possability of deadlock is very real. One alternative that eliminates this problem is *virtual cut-through swithing*, shown in Figure 1(c). In this method, a circuit establishment is attempted, but may terminate before reaching the final destination; the scenario in Figure 1(c) shows the situation when the second hop link is not available. Virtual cut-through switching effectively provides the benefit of circuit switching (low latency) under low utiization and the benefit of packet switching (simplicity and high throughput) during periods of high utilization.

So-called “wormhole networks” or flit-based networks [4], shown in Figure 1(d), offer advantages similar to virtual cut through routing with simplified implementation and and typically simpler signalling constraints and lower latency. In this switch design, a message is divided into *flits*, which are the smallest unit of communication exchange. For example, in a multi-computer network, a flit might be eight bits. Each interconnection step contains additional signaling lines used for flow control, such as “clear-to-send” (CTS) and “ready to send” (RTS). Except for the head and tail flits, all of the flits of a single packet are treated identically. The head flit is injected into the network and is used to establish a path through the network. Flits use buffer resources at intermediate nodes and use explicit flow control to limit the number of flits to insure there are no buffer overflows. The tail flit deallocates the buffer resources that was allocated when the head flit traversed the switch node. Because messages are divided into small flits, messages can be “pipelined” through the network – this reduces latency because the head flit is close to the destination when the last link becomes available. It also simplifies signaling because flow-control operates on shorter links.

However, as with circuit switching, flit switching requires a deadlock free path. The complexity of deadlock free routing can be simplified by introducing *virtual channels* in a flit network [5]. In this switch, each message is assigned a “virtual channel” between each pair of switches; in essence, this means that each switch subdivides the available buffer space between the different messages. This allows flits form the different messages to traverse the link, as shown in Figure 1(e). The sender and receiver must agree on a method to distinguish the flits of different messages and the flow control mechanism must be aware of the buffer space used. On common mechanism for this is include additional forward and reverse control signals. For example, the CTS/RTS control lines become c-wide fields for a switch with 2^c virtual channels; this increases the per-flit signaling overhead but are efficient when a varying number of virtual channels are used. Alternative methods use round-robin scheduling of virtual channels; these methods are less efficient when a varying number of virtual channels are in use.

There are also variants of these switching methods. For example, ATM networks use “cell switch,” which is similar to both packet switching and flit switching. In a cell switched network, a message is decomposed in many smaller “cells” (typically larger than a flit – ATM used 54 byte cells). Like packet switching, individual cells can be dropped during congestion and thus each cell contains addressing information and can be independently switched or routed. Like flit switching, the individual cells making up a message can be “pipelined” through the network.

1.1 Switching options for wireless networks

Wireless networks pose a number of challenges for switching designs. Most wireless systems can not transmit and receive simultanously in the same band because the transmission drowns out reception because receivers are designed to accomodate the ≈100-150db attenuation experienced in most wireless networks. This limits the ability to use circuit-switching or flit-switching unless there are ways to isolate the interference between transmitters and receivers. Several such technologies exist - for example many first-responder networks employ sophisticated relay networks to increase the range of emergency telecommunications. These networks use static frequency allocations in the radio switches; we intend to extend this using Orthogonal Frequency Division Multiplexing (OFDM), a modulation technique that uses a large number of “subcarriers” allocated across a specific frequency range [6].

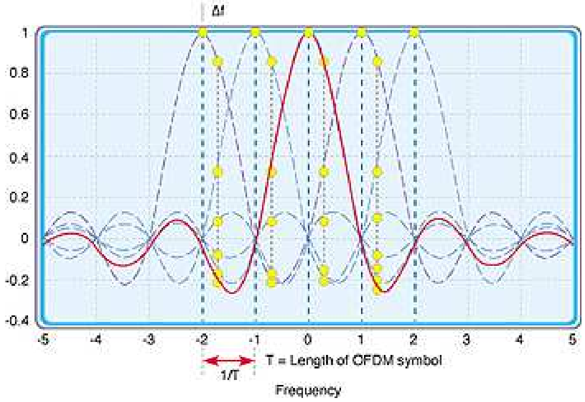


Figure 2: OFDM Waveform

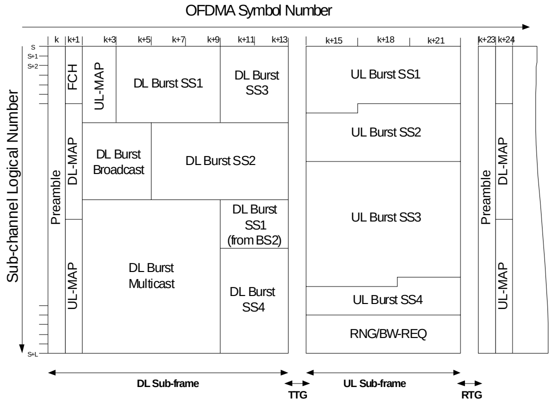


Figure 3: OFDMA Frame Format In 802.16

We are interested in building a *low latency* network using spatial frequency division multiple access (SFDMA) protocols. We want to use SFDMA to reduce the need to contend for a channel and use flit-based networks to reduce latency. We expect the network will have *engineering channel allocations*, and thus the basic switch organization will not require dynamic deadlock free routing.

In order to use multiple frequencies and simultaneously transmit and receive, we plan on using OFDM. A sample waveform for OFDM is shown in Figure 2. OFDM is used in the 802.11a physical layer as well as the 802.16-2004 specification. At a high level, OFDM waveforms have the benefit of reducing interference from multipath. Each sub-carrier is modulated at a fairly low rate, and the aggregate number of subcarriers provide high throughput. For example the 802.11a PHY has a total of 52 carriers with sub-carrier spacing of 312.5KHz. Since each sub-carrier is modulated at 312.5KHz, the symbol time is 3.2 μseconds. In indoor environments, the delay spread is typically 100 nanoseconds, meaning that the slowly-modulating subcarrier is more resistant to inter-symbol interference.

Most OFDM implementations devote multiple subcarriers to “pilot tones” that are used to transmit well-known symbols; these pilot tones are used to estimate the channel transfer function and correct frequency specific fading. By combining forward error correction across the different subcarriers, the resulting channel provides reliable, high bandwidth services.

As mentioned, OFDM is a well understood technology and has been used in a number of products, including the 802.11a and 802.16 wireless networks, digital audio broadcasting (DAB), high performance ship-to-ship wireless networks and powerline networking systems. We plan on using variants of OFDM for our prototype environment; OFDM provides the ability to have a large number of subcarriers that can provide greater flexibility in evaluating “frequency switching”.

The specific variant of OFDM we are most likely to use will be similar to OFDMA, or *Orthogonal Frequency Division Multiple Access*, which is used in 802.16 networks for efficient media control and allocation. In this scheme, illustrated in Figure 3 [1], individual subscribers to a wireless network (“subscriber stations”, indicated by SS in the diagram) are assigned specific frequency-time communication blocks. The 802.16 MAC layer may use either a full-duplex or time-division duplex (half duplex) transmission schedule. During download from the base-station to the subscriber station, frames for each subscriber are included in a single “superframe”; each station receives the beginning of the frame to determine the transmission schedule and then selectively acquires

---

[1] This figure was excerpted from [7]

information from its specified subcarriers. On the uplink, individual subscriber stations transmit their frames using tightly synchronized clocks; individual subscribers transmit for fixed durations and multiple subscriber stations may use the same subcarriers during the total uplink phase. Subscriber stations may use individual subcarriers for different purposes; for example, some commercially offered WiMAX (802.16) equipment allocates subcarriers to phone voice lines or VOIP services (thus providing specific bandwidth reservations) while the remaining subcarriers are used for best-effort internet service. Our use of OFDM will be similar, but will have the additional complexity of requiring tight frequency synchronization through the mesh or alternate solutions to avoid the need for such frequency synchronization.

Commercial implementations of 802.16 currently use a 256-subcarrier OFDM waveform and several inexpensive subscriber station chipsets are available [8]. Future systems will deploy 512, 1024 or 2048 subcarrier solutions [2]. The 802.16 draft specification provides support for mesh networks, but this meshing is implemented using standard store-and-forward techniques. Traffic is relayed from one station to another, demodulated and then transmitted to other relay stations or the final subscriber stations. Our proposal to combine OFDM with "radio wormhole" switches should reduce the latency through a mesh node by 1-3 orders of magnitude.

2 Implementation of Wireless Flits In OPNET

We needed to develop a simulation environment to evaluate the use of wireless flit-based networks. This section describes the model constructed for the OPNET simulation environment, its configuration and parameters.

2.1 Modeling OFDM subcarriers as individual data streams

Wireless communication in OPNET is achieved using the radio transmitter and receiver modules provided by OPNET. These modules allow for the configuration of some number of channels. The main parameters that define a channel, and so define the communication potential between a transmitter and receiver, are the bandwidth and minimum frequency, the data rate, and the modulation of the channel. The modules can have any number of channels, all of which are used independently. When a packet is sent to a particular channel of a transmitter module, OPNET iterates through the receiver modules found on other nodes and evaluates if the packet is received through the use of a 13 stage Radio Transceiver Pipeline. With this, if a receiver module is found with a channel configuration that matches that of the transmitter channel, the receiver is considered as a candidate for reception of the packet. However, if the transmitter channel overlaps a receiver channel only partially, the receiver will register the packet as noise and will not receive the packet. In addition, the data rate and modulation parameters must match or the packet will be viewed as noise by the receiver. If the three above parameters match, the transmit power of the channel, the distance separating the transmitter and receiver, error correction codes, and other related factors determine if the packet is received successfully. For further details, see the OPNET documentation pertaining to the Radio Transceiver Pipeline.

To model the subchannels of OFDM, the FLIT_RELAY model was developed that has 48 channels configured for the transmitter and receiver modules. For each of these channels, the bandwidth and minimum frequencies of the transmitter and receiver match. The modulation is QAM-64 and the data rate is 1.125 mb/s for all channels. With this, packets sent on channel 0 can be received on channel 0 of the receiver, but will not interfere with any of the other channels.

---

[2] In OFDM, the number of subcarriers are powers of two since demodulation is performed using FFT's.

With the 48 subchannels statically configured, the user must configure a communication channel between nodes by specifying which of the subchannels compose the channel. The subchannels assigned to a particular channel on the transmitter must match the subchannels assigned to that channel for the receiver. The model supports up to 10 channels, each of which is made up of some number of subchannels, and the total bandwidth and data rate of the channel is determined by the number of subchannels that compose it. The subchannel sets that represent the channels must be disjoint, and the union of the sets should include all 48 subchannels so as to make use of all of the available bandwidth. The node parameters that determine the subchannel assignments are the `Chan_*_Subchans` parameters, where `*` is a an integer 0 through 9, and the value is a comma delimited list of some number of subchannels 0 through 47.

In the FLIT_RELAY node, the FDM mechanism is modeled as an addition to the standard 802.11 MAC seen in the OPNET wireless nodes, not as a replacement. This was done to allow for broadcast messages in the network such as those for ad-hoc routing and RSVP. If the user intends to make use of the broadcast channel, they must determine how much bandwidth they are allowing for the broadcast channel, and possibly account for this usage by using less than all 48 of the subchannels when configuring the FDM mechanism.

2.2 Packet Forwarding

When a packet is generated by a node, it must be determined if the packet should be sent over the broadcast channel or if the FDM mechanism should instead be used. In addition, if the FDM mechanism is used the output channel that the packet should be sent on must be determined. For this, the user configures the model by setting the `Transmit_Chan_*_Destinations` parameters, where * is a channel. For each channel, the parameter is a comma delimited list of IP addresses, and packets destined for these addresses will then be sent out the associated channel. IP addresses are used as labels as they can be easily defined by the user for each node, and their usage removes the level of indirection introduced when assigning IP addresses to MPLS style labels. As the label size is user defined, the label overhead of various labels can be modeled while the actual switching decision is determined by the IP address.

The decision to send the packet out the broadcast channel or an FDM channel is made by the `relay_ip_arp_v4` process of the node model. This process is identical to the `ip_arp_v4` process distributed with OPNET with two modifications. First, when a packet is passed down the stack from the IP process the destination address is extracted and the `Transmit_Chan_*_Destinations` parameters are iterated through, checking if the destination is listed for any of the channels. If so, the packet is sent to the FDM process, otherwise it is sent out the broadcast channel. Hence, it is of the utmost importance that every destination is listed in one of the `Transmit_Chan_*_Destinations` parameters. The second modification to the `ip_arp_v4` process allows packets received by the FDM channels to be passed up the stack. This modification effectively merges the packet streams coming up the stack from the standard 802.11 MAC and the FDM mechanism.

When a packet is sent to the FDM mechanism, the output channel is determined using the `Transmit_Chan_*_Destinations` parameters as discussed above. As a channel is made up of some number of subchannels, the packet is sent out the associated subchannel that has the smallest current queue size. Packets are sent out the subchannel after a delay determined by the `Initial_Delay` parameter. This allows the user to model the time needed to determine the output channel.

When a packet is received at an intermediate node, the `Receive_on_Channels` parameter is used to determine if the packet should be processed. This is a list of channels that are assigned to the node for reception. Packet transmissions overheard on non-assigned channels will be ignored. If the channel on which the packet is received is listed in the `Receive_on_Channels` parameter, and the packet is destined for the node, it is passed up the stack. If the node is not the destination, the packet must be forwarded to the next hop.

When forwarding packets, one of two techniques is used to determine the output channel for the packet; label switching or frequency mapping. With label switching, each packet carries a label that is used to determine its path through the network. The label may be an IP address, an MPLS label, or a virtual channel ID. With label switching, statistical multiplexing can be used to allow the sharing of a single channel between flows. However, the additional overhead introduced by the labels, as well as the overhead introduced by switching apparatus complexity, may not be acceptable in all situations.

Frequency mapping on the other hand, simply takes packets received on one channel and translates them to an output channel. The benefits of this approach are that the switch can be far simpler and there is no label overhead. Unfortunately, flows cannot be multiplexed over a single channel, which in some cases results in a scarcity of available frequencies.

In the FLIT_RELAY model, the boolean switch parameter `Use_Labels` determines which of the two techniques is used. With label switching, the parameters `Relay_Chan_*_Destinations` are used. These function similarly to the `Transmit_Chan_*_Destinations` parameters discussed above. The values are lists of destination IP addresses that should be forwarded out a particular channel. When a packet is received, the destination address is examined and the packet is sent out whichever channel has this IP listed in its `Relay_Chan_*_Destinations` parameter. When using label switching, overhead must be added to each packet to model the label. The size of this overhead is set using the `Label_Sizeparameter,` and this overhead is added to every packet by the originating node when `Use_Labels` is true.

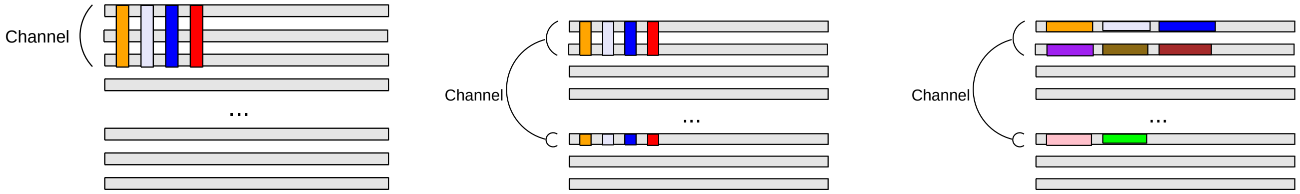
If frequency mapping is used instead, the `Relay_Chan_*_Destinationsparameters` are ignored, and the `Receive_*_Out_Channel` parameters are used. These are one to one mappings where * is the channel the packet was received on and the value is the output channel for all packets received on *. When using frequency mapping, the label overhead is not added to packets. Regardless of the forwarding technique used, the packet is sent after a delay determined by the `Translation_Delay` parameter.

2.3 Flits

While our preliminary simulations suggest that far lower packet latency can be achieved with our relay technique than with convential 802.11 communication, even greater gains can be seen with the introduction of flits. The intuition behind flits is that a packet can be broken into small pieces called flits, and these flits can then be pipelined through a multi-hop network. This pipelining of flits further reduces packet latency in comparison to a simple store and forward approach as the different nodes along a multi-hop path can be processing parts of the packet simultaneously.

In our model, the user can model both flit based and simple store and forward communication. The `Use_Flits` parameter is a boolean switch that determines which approach is used. If flits are used, the user defines the flit size using the `Flit_Size` parameter. With this, when a packet originates at a node it is broken into a number of flits of `Flit_Size` bits. If the packet cannot be evenly divided, the last flit is padded so all flits are of equal size. The flits are then spread across the subchannels of the output channel, with flits being output to the subchannel with the smallest queue size. In addition, if label switching is used the label overhead is applied to each flit, so the total size is determined by the `Flit_Size` and `Label_Size` parameters.

The transmission of flits is identical to the transmission of packets from the viewpoint of relaying nodes. Flits are simply relayed on the appropriate channel, as when using packets. However, the destination node must reassemble the packet which is then sent up the stack after it is complete. For this, a number of packet buffers



(a) Aggregated adjacent channel (b) Aggregated non-adjacent channel (c) Independent non-adjacent channel

Figure 4: Possible Channel Models

are used that are populated when flits are received at the destination node. The buffers keep track of the number of flits received for each packet, and when all flits are received for a particular packet the packet is sent up the stack.

2.4 Model Trade-offs and Discussion

There are two points of configuration that must be closely understood to assure accurate modeling and best performance. These issues concern the usage of subchannels and the effect of flit size.

With OFDM, a communication channel can be viewed as a single channel composed of a number of subchannels, or as a set of independent subchannels. This is illustrated in Figure 4. In (a) and (b), flits are modulated across all subchannels while in (c), entire flits are sent across individual subchannels. However, the channel utilization of these two techniques is only equivalent when the number of flits is a multiple of the number of subchannels. For example, if we have one flit and two subchannels, using independent subchannels will not effectively utilize the channel as only one subchannel would be used and the other would stand idle. In contrast, modulating across both subchannels would fully utilize the channel and the packet will take half as long to transmit. As load increases the channel utilization of the two techniques converge.

In most cases, modulating across multiple subchannels makes best use of the medium. Unfortunately, modeling this in OPNET is not always possible using the standard radio communication model. In OPNET, a packet can be sent to a single output channel. This is a fundamental modeling concept in OPNET. To model modulation of packets across a number of OFDM subchannels it is necessary to aggregate subchannels into a single channel. For example, in (a) the three independent subchannels would be replaced by a single channel with a data rate and bandwidth equal to the aggregate of the three subchannels. In OFDM, however, packets can be modulated over a number of subchannels that are not adjacent in frequency space, as seen in (b). The OPNET channel model does not allow this, as the interference model would be compromised. To illustrate, if we are to aggregate the subchannels of (b) into a single channel, we must be able to set a single minimum frequency and bandwidth value for the channel. This entails that the bandwidth must span across all subchannels, and so interfere with the intermediate, non-assigned, subchannels, or it must span across only the upper two subchannels resulting in the absence of realistic interference for the lower subchannel. Neither of these is acceptable, and this led to the usage of 48 independent subchannels.

As the OPNET channel model does not allow for the aggregation of non-adjacent subchannels into a single channel, it is necessary to assure the highest channel utilization when using independent subchannels. This can be achieved by transmitting very small flits across the independent subchannels, which more evenly spreads the packet across the subchannels. However, every packet is then split into a large number of flits, with a

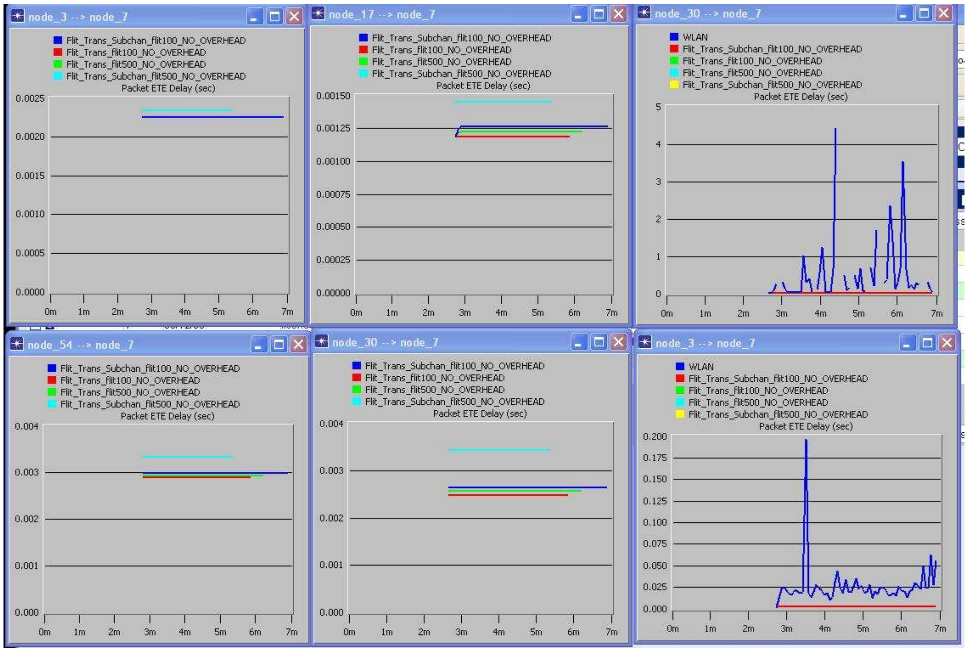


Figure 5: OpNET Performance

proportional increase in simulation time. When using independent subchannels, a smaller flit size means better channel utilization at the expense of longer simulation time. This trade-off must be considered by the user.

There is an additional factor that must be considered when deciding on the flit size. When using labels or virtual circuits, every flit will incur the label overhead. As flit size is reduced to avoid inefficiencies in subchannel usage the label overhead begins to contribute significantly to the total flit size.

3 Some performance studies using the simulator

To demonstrate the effect of varying flit size when using independent subchannels and subchannel aggregation, a series of simulations was conducted for a seven node wireless multi-hop network. For the simulations, no label overhead was applied. In Figure 5, we show end to end delay results when using aggregated subchannels and independent subchannels while varying the flit size. In the simulations, a traffic flow of 10 kb/s exists from each of the nodes to node seven, and also a 10 kb/s flow from node seven to each of the other nodes. The packet size for all simulations was 1500 bytes. The longest path in the network was three hops, and consists of the path $30 \rightarrow 54 \rightarrow 17 \rightarrow 7$. The end to end delay results are shown for the traffic flows originating at each of the nodes on this path. Additionally, results are shown for node three which is a one hop neighbor of node seven and is not responsible for relaying any traffic.

As can be seen when looking at the four leftmost graphs in Figure 5, the simulation results labeled Flit_Trans, where subchannels are aggregated, show little variation when using different flit sizes. In contrast, the results when independent subchannels are used, labeled Flit_Trans_Subchans, show wide variation when using different flit sizes and this discrepency increases with path length. With a flit size of 500 bits the simulation using independent subchannels performs far worse than any other case, while using a flit size of 100 bits approximates the results seen when using aggregated subchannels.

In addition to comparing independent subchannels and subchannel aggregation, we performed experiments for the same network using 802.11g and AODV. In the simulation, AODV used the same routes as were used for the relay simulation. The results for this can be seen in the rightmost graphs of Figure 5. While there is an appreciable difference between the two methods of subchannel usage, this difference pales in comparison to the difference between these techniques and conventional ad-hoc networking. At best, the end to end delay seen for the 802.11g network is nearly an order of magnitude greater than for our relay network; this is seen for the single hop path from node three to node seven. When looking at the results for the multi-hop case, node 30 to node seven, the results are many orders of magnitude greater. In this case, discontinuities are seen which indicate that packets were lost.

4 Discussion & Conclusions

These preliminary simulations suggest that for best results when using the relay model, it is desirable to aggregate the subchannels when they are adjacent in frequency space, and use the smallest flit size that (simulation) time allows when they are non-adjacent. From the comparison to 802.11g, the initial findings suggest that an OFDM based relay network could provide significantly better performance than current 802.11 based solutions. While these results are preliminary, they are promising and encourage further research.

References

[1] Howard J. Siegel. Interconnection networks for simd machines. *IEEE Computer*, 15(1):57–65, June 1979.

[2] C. Kruskal and M. Snir. The performance of multistage interconnection nets for multiprocessing. *IEEE Transactions on Computers*, 32(12):1091–1098, 1983.

[3] Daniel A. Reed and Dirk C. Grunwald. The performance of multicomputer interconnection networks. *IEEE Computer*, 20(6):63–73, June 1987.

[4] William J. Dally and Charles L. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. *IEEE Transactions on Computers*, C-36(5):547–553, May 1987.

[5] William J. Dally. Virtual-channel flow control. In *ISCA '90: Proceedings of the 17th annual international symposium on Computer Architecture*, pages 60–68, New York, NY, USA, 1990. ACM Press.

[6] Ove Edfors, Magnus Sandell, Jan-Jaap van de Beek, Daniel Landstrom, and Franl Sjoberg. An introduction to orthogonal frequency-division multiplexing. Technical report, Lulea University of Technology, Sept 1996.

[7] Hassan Yaghoobi. Scalabel ofdma physical layer in 802.16 wirelessman. *Intel Technology Journal*, 8(3):201–212, 2004.

[8] Intel Corp. *Intel ProWireless 5116 Chipset Product Brief*, Apr 2005. Available at `http://www.intel.com/network/connectivity/products/wireless/prowireless_5116.htm`.