Computer Communications 33 (2010) 655–666

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Networking in E-textiles

Z. Nakad [a,*], M. Jones [b], T. Martin [b], W. Fawaz [a]

[a] Lebanese American University, Electrical and Computer Engineering Department, P.O. Box 36, Byblos, Lebanon
[b] Virginia Polytechnic Institute and State University, Bradley Department of Electrical and Computer Engineering, Blacksburg, VA 24060, USA

ARTICLE INFO

Article history:
Received 14 July 2008
Received in revised form 22 July 2009
Accepted 12 November 2009
Available online 4 December 2009

Keywords:
E-textiles
Fault-tolerance
Interconnection networks
Distributed ubiquitous systems

ABSTRACT

The abundance of fabrics in our life offers immense possibilities for ubiquitous electronic integration. Electronic textiles (e-textiles) represent the merging of fabric and electronic technologies to be used in wearable and large-scale applications. The communication requirements of the computing elements of these systems will be presented along with a fault tolerant solution. The operation of the communication scheme will be studied analytically and compared to simulation results. The graceful deterioration of the performance with increasing faults will also be reported. Multiple e-textile application examples utilizing the networking scheme will be presented.

1. Introduction

The merge of textile and electronic technologies creates a novel method in deploying electronics in a pervasive and non-intrusive fashion. Electronic textiles (e-textiles) represent this merge and are defined as a set of sensors, processors, and/or actuators embedded in a fabric backplane and interacting to fulfill a specific task. Communication and power lines are integrated in the fabric as well. The terms e-textiles and electronic textiles will be used interchangeably in the text.

This paper focuses on the interconnection network created to serve e-textiles. E-textile applications are predicted to exist in two different formats. A wearable format that integrates e-textiles in regular clothes or apparel and a large-scale format that integrates e-textiles in large fabrics; carpets, wallpapers, tents, and camouflage nettings are some representative examples. The requirements of these systems vary with the specific application.

The communication between the different processing elements in a fabric presents challenges and opportunities that create a unique design situation. Flaws expected during both the manufacturing of the fabric and the deployment of an e-textile in the field (tears) call for the implementation of a fault-tolerant scheme to deal with both permanent and transient faults. E-textiles deployed in the field are also expected to stay active for a long time while operating on batteries compelling low-power consumption considerations in both the design of the elements and the communication scheme.

The network to be implemented in the fabric must have a low-power operation cost, and be able to route around faults. Processing nodes with a "sleep mode" are also to be supported by such a network to alleviate power consumption. A hardware prototype, an acoustic beamformer, was constructed to test the networking scheme and the feasibility of working on e-textile applications. A case study for this application is presented in [1]. This prototype will be referred to in the discussions in this paper. This system consists of four stand-alone beamforming clusters; the independent results of the clusters are communicated to other clusters to create a more accurate result.

The implemented interconnection network is based on the Token Grid [2] network and is modified to fulfill all the requirements of e-textiles. This paper presents the theory of operation of this network (Textile Token Grid), explains the fault-tolerant scheme, and presents simulation and analytical results, that show the gradual degradation of performance with increasing faults. The addition of a "transverse" dimension is also presented. This dimension helps in the performance of the fault-tolerance scheme and also in interconnecting different swatches of fabric that will form the overall e-textile. A shirt for example is an integration of several separate swatches of fabric. The main focus of the networking scheme is providing connections between nodes on the e-textile when they are awake (around 2–5% of the time, depending on the application) while also routing around faults and "sleeping" nodes.

This paper is organized as follows: Section 2 provides background information on electronic textiles and the Token Grid network, the reasons for deciding on this specific networking scheme are underscored. Section 3 presents the variations and the implementation of the new networking scheme. Section 4

* Corresponding author. Tel.: +961 9 547254/2166; fax: +961 9 547256.
E-mail addresses: znakad@lau.edu.lb (Z. Nakad), mtj@vt.edu (M. Jones), tlmartin@vt.edu (T. Martin), wfawaz@lau.edu.lb (W. Fawaz).

0140-3664/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2009.11.011

describes the operation of the network in optimal and faulty conditions, the operation with sleeping nodes is also presented. Section 5 provides an overview of the simulation environment used in presenting the results. Section 6 discusses the analytical method used to predict the performance of the network. Section 7 presents several scenarios that were used to test the networking scheme along with the analytical and simulation results for the different scenarios. Section 8 concludes the presented work.

2. Background

Electronic textiles promise the enhancement and introduction of a plethora of applications. The creation of these systems requires the implementation of a communication scheme to connect the several interacting components. This communication scheme has to abide by the promise of e-textiles in providing a fault tolerant and low-power system. Testing a networking system requires a simulation environment especially when testing on a large number of nodes. The following will provide background information on extant e-textile projects, the base networking system used, and the simulation environment utilized in testing the overall system.

2.1. Electronic textiles

A plethora of electronic textile applications are found in the market and in the research community, a representative sample will be provided in this section. Three examples from The Virginia Tech E-Textiles Laboratory are the glove project [3], the Acoustic Beamformer [1,4], and the motion-sensing pants [5]. The glove detects the motion of the user's fingers and acts as an unobtrusive keyboard for a wearable computer. The Acoustic Beamformer is a more complex project that integrates multiple processors and sensors. The beamformer is a large-area system, Fig. 1, that is deployed in the field to locate passing vehicles by their audio signature. The Acoustic Beamformer is used as the test subject for the communication network that is presented in this paper. The motion-sensing pants introduce the idea of repeatable swatches of fabric that form a large communication background on which smart sensors and processors communicate arising a great need for a reliable communication network [6]. The Acoustic Beamformer is one of the few applications in e-textiles where the processing is done on the fabric and the results are reported to the outside. As more processing power is added to such applications especially with multiple nodes; a communication scheme is required to interconnect these nodes. This paper presents one approach to connecting multiple processing nodes in an e-textile environment.

Fig. 1. The Acoustic Beamformer with four nodes in a field test.

Reliably connecting electronic components to the fabric is one of the challenges of e-textiles. The authors in [16] present the design and implementation issues of e-Textile Attached Gadgets (e-TAGs). This approach allows for modules that are easily attached, removed, replaced, and interchanged in an e-textile implementation. This approach was used in [17] to create an e-textile that automatically collects physiological and contextual information for medical analysis. Modeling an e-textile is another important aspect in the design phase that was also studied and a simulation environment was created to help in modeling new implementations. This paper discusses the portions pertaining to the communication between processing nodes on an e-textile while [18] provides a wider view of the simulation environment.

The GTWM (Georgia Tech Wearable Computer) [7] is another project that utilizes e-textiles to locate a bullet hole on a soldier. The LifeShirt [8] includes sensors embedded in its fabric to measure the breathing rate. These are just few examples of applications that rely on the fabric for sensing and providing a simple connection between elements. No processing is done in the e-textile in these cases. The authors in [19] provided a survey of the novel e-textile architecture while showing multiple trends tackled in several research groups. Challenges and opportunities for e-textiles were studied in [20] where the authors divided the challenges for designers and CAD tool developers into three areas: the need for a new model of computation, reconfigurability and adaptability with low computational overhead, and device and technology challenges. These form the key aspects that an e-textile implementation has to focus on. [21] addresses modeling of computation, communication, and failure in e-textiles; the performance of two techniques, namely, code migration and remote execution is investigated. This paper discusses the operation of a fault tolerant network that deals with both node and link failures and supports "sleeping" nodes to help in power utilization. This network is based on the Token Grid introduced in the next section.

2.2. Token grid

A communication network servicing an e-textile system should be easily implemented in a fabric backplane, communicate inner network information, provide scalability, and offer fault-tolerance. In an e-textile system, the number of the nodes to be connected would not be known apriori and that number is expected to change throughout the lifetime of the system. Several network schemes can be applied such as hypercube or tree-type architectures. The node degree, number of connections at the node, increases linearly with an increase in the dimension of a hypercube [9]. Given the fixed node degree in fabric, this renders architectures similar to the hypercube unsuitable for e-textiles. A tree-type architecture relies heavily on specific nodes for connections between different branches, which does not map well to the faulty environments of e-textile applications. The fixed degree, fault-tolerance [10], and reasonable scalability of the Token Grid are the primary attractive features [2,11]. A token traversing the network can be used to keep information about the topology and the state of the nodes, another benefit of the Token Grid shown in Fig. 2. This network matches the inherent *X*–*Y* layout of a fabric facilitating its implementation on a fabric backplane.

Each node in the Token Grid is connected to two token rings (Fig. 2).This architecture offers a significant advantage in throughput as compared to common rings or bus networks, its throughput increases with the number of nodes [12]. This increase in throughput is a significant benefit to systems in which the number of nodes on the grid is not known apriori.

The network interface of each node on this network has two configurations as shown in Fig. 2. In the DR (Double Ring) configuration (all nodes except (1, 2)), the rings converging at this node

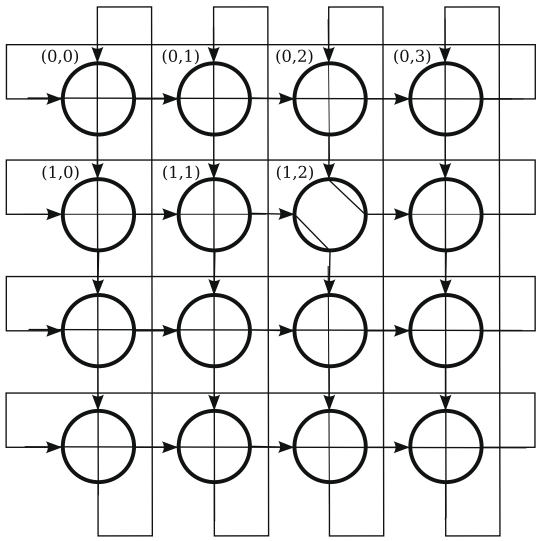


Fig. 2. Row 1 and column 2 of this Token Grid are merged at Nodes (1, 2) (based on information from [2]).

are separate. The second configuration is the SR (Single Ring) configuration (node(1, 2)), the rings converging at this node are merged and operate as a single ring. Special tokens are released in such a situation; namely the row/column and column/row tokens. The merge operation has been altered for the Textile Token Grid and will be discussed in Section 3.

2.3. Simulation

The entire operation of the system, from the communication scheme to the sensor behavior and the processing done on the nodes is simulated with Ptolemy [13]. An e-textile system is constituted from different interacting components ranging from complex processing nodes to simple acoustic sensors. All of the components placed on the fabric can be simulated as Ptolemy actors and their interactions as one of the models of computation. Significant aspects of the physical world can be added to the simulation; for example, the sound of a passing large vehicle.

The simulation environment was created at first to simulate the operation of the Acoustic Beamformer [1,14]. The audio signature of a passing vehicle, and the analog audio signals received at the separate microphones of the beamformer were simulated using the CT (Continuous Time) domain. The processing inside the processing nodes was simulated using the DE (Discrete Event) domain as the processing of the beamformer and the communication control are both tied to a timer interrupt. The physical layer of the communication scheme in both the hardware prototype and the simulation is based on a serial protocol that mimics RS-232. Other than the Ptolemy actors the simulated code was kept as close as possible to the hardware code to preserve the correctness of the simulation. The presented work focuses on the operation of the networking scheme and presents a comparison between simulation and analytical results for different applications.

3. Textile Token Grid

This section discusses a networking scheme designed for e-textile applications. The scheme is based on the Token Grid described in Section 2.2. The Token Grid was modified to provide better support for electronic textile applications. The number of hops a token has to traverse is directly related to the number of nodes on the ring. In the case of large-area e-textiles, a large number of nodes in a ring can slow the communication on the ring. Fabric swatches will be connected to form large or wearable fabrics as shown in Fig. 3; the networks on these different fabrics must be connected. The addition of a third dimension to the networking scheme to address both of these issues.

A new fault-tolerance scheme was created for the Textile Token Grid for two main reasons. First, the Token Grid assumes no faults on the communication links. Second, the Token Grid assumes that the processing nodes will have the ability to forward data in the case of a failure. Both of these assumptions do not apply to e-textiles. The implemented fault tolerant scheme is discussed in Section 4.3. The Textile Token Grid also supports low-power operation by routing around "sleeping node", this avoids waking these nodes up unless necessary. This section focuses on the addition of the transverse dimension. The other aspects are discussed later.

If a node wishes to communicate on a node on its row or column it has to wait for the token of that ring, grab the token, send the packet to the destination and release the token. The other nodes on the ring would receive the data packet, check if they are the destination and consume the packet if so or forward it down the ring otherwise. If a node wishes to communicate with a node on neither its row nor column ring, then it has to send a "request a merge" to a node where the row of the source intersects the column of the destination, or the column of the source intersects the row of the destination. When a node receives such a request it waits to grab the other token on its rings; after both tokens are captured the node enters the merged state. The merged node sends a row/column token around its row, any node on the row can send to a destination on the merged node's column. When the row/column token is captured, a column/row token is circulated around the column, and the nodes on that column can send to nodes on the row. At the reception of the column/row token at the originating node, the merged state is stopped and regular row and column tokens are released.

4. Network operation

The implementation of a fault tolerant communication scheme is a must for electronic textile systems. The following section discusses the fault tolerant scheme used in this architecture. Detection of errors and propagating fault information are the first aspects presented. Utilization of this information in routing data is then discussed.

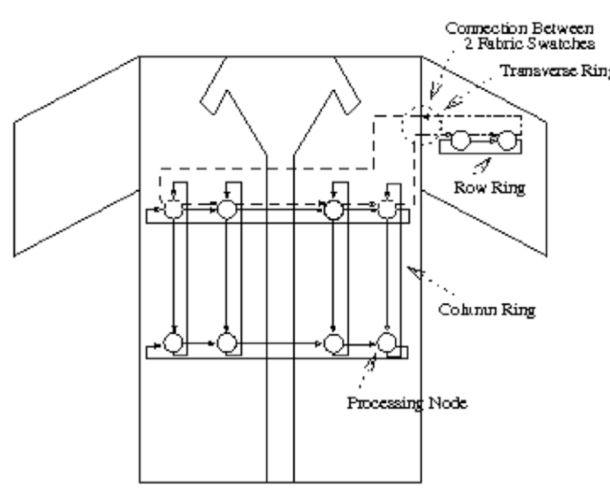


Fig. 3. The transverse dimension can be used to join processing nodes on different fabric swatches.

4.1. Error detection

Each node is connected to three different rings (row, column, and transverse), each providing a possibility for a link error. Each node will keep three separate Timeout Counters pertaining to each link it is connected to. These counters are reset to zero at the reception of any data (tokens, data packets, and error packets). Fig. 4 shows a specific example, Node 1 does not receive anything from its ring and after the timeout period it creates an error packet and sends it down the ring. Node 1 will also update a flag that states that the row it belongs to is in error. When a node receives an error packet it forwards that packet down the same ring and updates the flag stating that the ring is in error.

4.2. Error information propagation

Using the process discussed above, each node on a ring with an error has a flag that reflects that information. This information is to be propagated to other nodes in the network. If the error is on the row ring, then the column token that passes through each of these nodes will be updated to reflect that the specific row ring has an error. This has no propagation cost because the token is already traversing the whole column ring. Each node will have to keep another variable that pertains to errors. This variable will carry the information of errors on other rings in the network; three variables will be needed reflecting the information about the other rows, columns, and transverses in the network.

4.3. The fault tolerant scheme

This subsection demonstrates the operation of the fault tolerant scheme. Communication between nodes on the same row or column are dealt with directly by the token traversing the significant ring. When the destination lies on a different row and a different column then a merging process will be needed as seen with node (1, 2) in Fig. 2.

The first case to consider is a link fault on the destination row as shown in Fig. 5 where Node 0 is trying to reach Node 5. This discussion assumes that all of the significant error information has been transmitted and received. From the error information, Node 0 will only request Node 1 to go to a merged state. Node 1 will receive the request for the merge through the row token. At the reception of the row/column Token, Node 0 will release the data packet that will be forwarded by Node 1 to Node 5.

A more complex case occurs when there is an error on both Row 0 and Row 1, as depicted in Fig. 6(a), with the same transmission from Node 0 to Node 5. In this case, both nodes in position to offer a useful merge have errors on their significant rings. The data packet is forwarded on a "Wrong Route", the column in this case. The node that receives this data packet should save it in its data queue for processing at the reception of a token as if it is a newly created packet. In the presented case, the data packet first reaches Node 4. At that node, the destination (Node 5) is not reachable because of the link error and the data packet is forwarded on the "Wrong Route" again, down the column. Node 8 then requests Node 9 for a merge; which enables the data packet to reach Node 5 through the column ring. Fig. 6(b) depicts a case where a node is trying

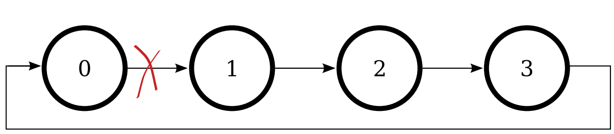


Fig. 4. Node 1 has an error at the link directly to its "left".

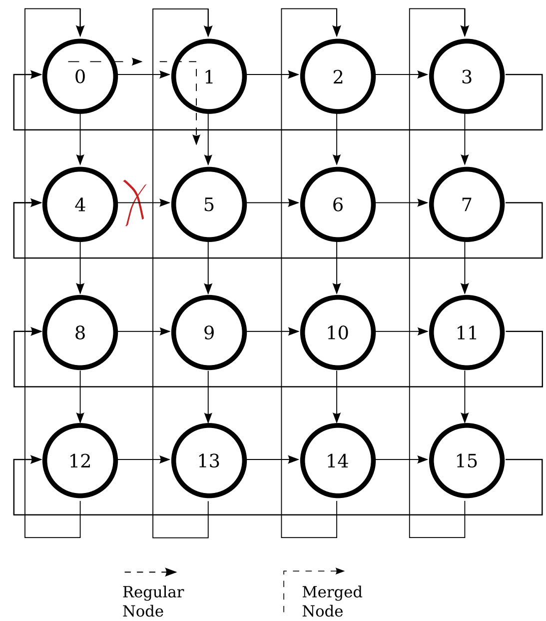


Fig. 5. A link error on the destination row in the transmission from Node 0 to Node 5 limits the merge options.

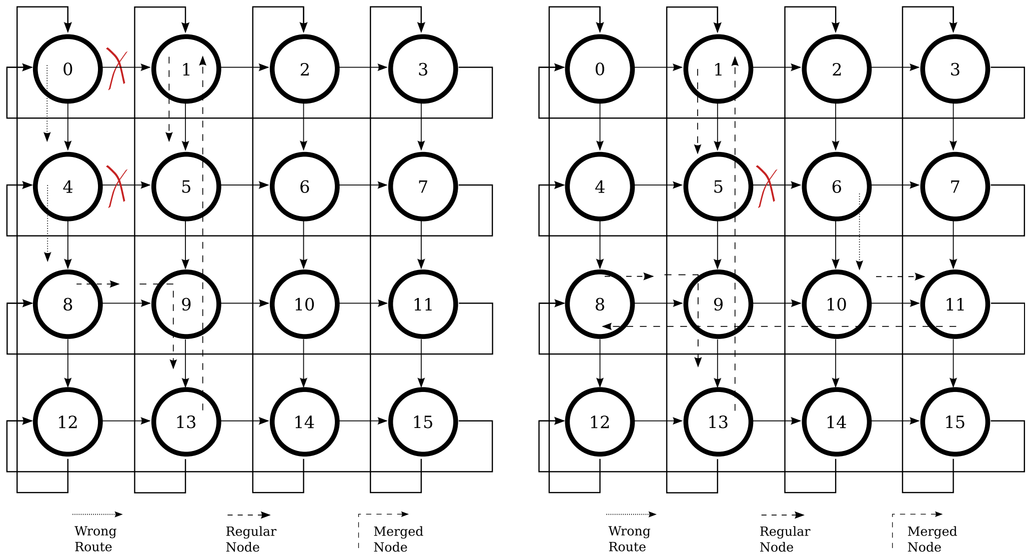
to send to another node on its row with an error on the row. The details of this algorithm and its pseudo codes are presented in [14].

5. Simulation

Ptolemy [13] was chosen as the tool to simulate the presented networking scheme because it is designed to simulate heterogeneous and embedded systems. The created simulation environment dealt with simulating the networking aspect along with the processing in the nodes depending on the specific application while also providing a power utilization estimate [14,18]. The Acoustic Beamfomer [1] acted as the case study for the simulation environment. This section will describe the different aspects of the simulator and will demonstrate its general usability. With Ptolemy it is possible to simulate the interaction with the outside world and the response created from the e-textile system. The presented discussion will focus on the simulation pertaining to the communication scheme, a full description of the simulation environment is presented in [14,18].

The simulator environment simulates different aspects of an e-textile system and the physical world it interacts with. The three main parts of this environment are: Physical World, Interrupt Handling Core, Power Simulation. The Physical World as the name implies deals with simulating the physical world that is sensed or interacted with by the e-textile prototype. Simulating a received audio signal is an example of a physical aspect that would be simulated using the CT domain (Continuous Time) in Ptolemy. Simulating faults on the fabric is another aspect to be simulated to test the operation of the system under varying faults that can cause loss of power or communication this type of simulation is created in the DE domain (Discrete Event). The Interrupt Handling Core simulates the operation of an embedded system that utilizes interrupt driven processing. This simulator is managed by the DE domain and provides it operational states to the power simulator. The Power Simulation entity collects the power state information from the core simulator and provides power consumption values for the whole system.

Fig. 15 shows the simulator of a node in the Acoustic Beamformer; the simulation of the full operation of this node from



(a) Link errors on Row 0 and Row 1 prevent a regular merging scheme and force a "Wrong Route."

(b) The path of a data packet from Node 6 to Node 5 with an error on Row 1.

Fig. 6. Data packet paths with link errors.

processing the acoustic signals received to the communication to the other nodes are implemented here. Fig. 16 shows an example of connecting 32 nodes together utilizing Ptolemy providing the ability of testing the functioning of the networking scheme in different settings.

6. Analysis and enumeration of transmission costs

Enumerating the cost to send a packet from a source to a destination is needed to provide any study on the performance of the network under varying fault schemes. Given the networking scheme used, the location of the destination has a direct effect on the amount of processing and forwarding required from the network and the nodes in question. The following section studies the different costs attributed to sending a data packet from a fixed node to a set of different destinations. Two costs will be presented, the first represents the cost of transmission for the source node, the second represents the number of hops in the network to reach the destination.

The simplest case is sending to a node on the same ring. The first step is to wait to capture the token TW_{token}. The time needed to send this packet is TS_{data}. After sending, the node can release the token. The operation of the network requires that a token is always rotated around the ring; thus, the extra cost of sending the data packet is the parameter, TS_{data}. Table 1 presents a set of variables that will be used in reporting measures of network performance.

The operation of the networking scheme is that it sends a row/column token when a node first merges. This token traverses the row ring allowing nodes sending to the merged column to send. When the merged node receives this token, it releases a column/row token on the column ring to allow nodes on the column to send on the row ring. This provides a smaller waiting time for a node sending from the row to the column and that is reflected in Table 1.

Table 1
Nomenclatures and variables used in the discussion.

Variable name	Value	Explanation
BW	–	Bandwidth of the communication links
TS_{token}	$Size_{token}/BW$	Time required to transmit a token
TW_{token}		Time required to wait for reception of token
min[a]	TS_{token}	(min) Preceding node is sending the token
max[b]	$(N-1)TS_{token}$	(max) Following node is sending the token
Average	$(N/2)TS_{token}$	
TS_{data}	$Size_{data}/BW$	Time required to transmit a data packet
$TW_{merged\text{-}token}$	$TW_{token}+$	Token carrying request to reach merging node
(sending with	$TW_{token}+$	Time required to capture the other token
row_column)	$TW_{rc\text{-}token}$[c]	Time required to wait for row_column token
$TW_{merged\text{-}token}$	$TW_{token}+$	Token carrying request to reach merging node
(sending with	$TW_{Token}+$	Time required to capture the other token
column_row)	$TW_{rc\text{-}token}+$	Time required to wait for row_column token
		Time required to wait for column_row token
$TW_{merged\text{-}token}$	$2(TW_{token})+$	Assuming equal distribution between
(average)	$1.5(TW_{token})$	rc and cr merged tokens

[a] The Node processing times and the link latencies are considered negligible to simplify the discussion. The token is assumed to not be held in the current node.
[b] Considering there are no other data packet transmissions (light load).
[c] rc-token row column token, cr-token: column row token. $TW_{token} = TW_{rc\text{-}token} = TW_{cr\text{-}token}$.

Table 2 provides the costs of communication as perceived by the sending node. When the destination node is on the same ring, the specific location on the ring is not related to the cost of sending to the source. The source node waits to capture the ring token, sends the data packet, and then releases the token. This operation is the same if the destination is the next node in the ring or the farthest. The increase of cost in sending to a farther node is manifest in the number of hops traversed, this cost is reported in Table 3.

A last point to investigate is the operation of this networking scheme in a large network, or the deterioration of the networking performance with an increasing number of nodes in the network. As mentioned earlier the connection cost includes two costs, the cost to the sending node and the cost in number of hops to reach the destination. The cost to the sending node as shown in Table 2 depends on TW_{token} or $TW_{merged\text{-}token}$. According to Table 1 $TW_{merged\text{-}token}$ is a function of TW_{token} which is dependent on N (the number of nodes in a ring). The cost to the sending node increases with increasing the number of nodes in the ring and not the number of nodes in the network. In a square network, this increase is dependent on $\sqrt{N^2} = N$ where N^2 is the number of nodes in the network. The cost on the sending node is not affected in the presence of a transverse connection because in both cases the node waits for TW_{token} before sending.

The cost in terms of hop count is different. If there are no transverse connections, the connection cost is an average of $N/2$ for same ring connections and N for a different row and column connections. The communication cost also grows in order of N, the number of nodes in the ring. In the case of transverse connections, the connection to another grid is a constant factor increase over a regular connection inside the grid and thus is dependent on N. For a network with multiple grids, the maximum communication cost will be that of traversing all of the transverse connections. Let M be the number of nodes in the network and N the number of nodes on a ring. Assume all of the grids are square and of the same size. The number of grids in the network is $\frac{M}{N^2}$, therefore the maximum connection cost is $O(\frac{M}{N^2} * N)$ (N represents the order of increase for a transverse connection).

There are two cases where the network scales, one in which the physical size stays fixed and the density increases, and one in which the physical size increases. In the case of a wearable fabric, the number of grids cannot be increased indefinitely as the number of nodes is increased. In this case the factor $\frac{M}{N^2}$ is $O(1)$ and the overall communication cost is $O(N)$. In a large-scale application the number of grids in the network increases with the number of nodes and thus the communication cost is $O\left(\frac{M}{N^2} * N\right) = O(M/N)$.

The cost of communication varies with the location of the fault and the specific communication path under study. The next section will provide an analytical method to predict the general response of the network and will compare the analytical results with the simulated values.

7. Results

The following section presents the results collected from simulating the networking scheme under varying fault occurrences, loads, and sleeping nodes. The networking scheme was tested by using a model of communication based on either a general communication setup or an e-textile application setup. The simulation results obtained are compared with our predicted analytical results to prove that the networking scheme operates as expected. The results also depict the degradation in performance with additional faults.

Table 2
Enumerated communication timing costs (to the sending node).

Type	Dest.	Total Time	Extra cost	Explanation
Ring token[a]	–	$TW_{token}+$	–	Time waiting for token
		TS_{token}		Time to send that token
Data packet	Same ring	$TW_{token}+$		Time waiting for token
		$TS_{data}+$	TS_{data}	Time sending the data
		TS_{token}		Time sending the token
Data packet	Different row & Column	TW_{token}		Waiting (request merge)
		$TW_{merged\text{-}token}$	$TW_{merged\text{-}token}$	Time waiting for merged
		$+TS_{data}+$	$+TS_{data}$	Time sending the data
		$TS_{merged\text{-}token}$		Time sending the merged
Data packet	Different grid[b]	$TW_{token}+$		Waiting for transverse or
		$TS_{data}+$	TS_{data}	Row Token[c]. Send data
		TS_{token}		Send token

[a] The value of TW_{token} assumes no data transmissions from other nodes on the ring.
[b] The cost to the node here is loss than the cost to send The packet. The packet will have to be re-routed once it reaches the other Grid.
[c] If the node is on the wrong transverse ring, forward on the Row ring until it reaches the correct link.

Table 3
Enumerated number of hops (cost to the network)

Type	Destination	Total cost	Explanation
Ring token	–	1	One hop source to destination
Data packet		1 (min)	Destination is next node
	Same ring	$N-1$ (max)	Destination is previous node
		$N/2$ (average)	
Data packet	Different row & column	2 (min)	One row & one column off
		N (average)	
		$2(N-1)$ (max)	$(N-1)$ rows & $(N-1)$ clmns off
Data packet	Different grid (correct Tran.)	l (min), $(N/2)$ (max)	To reach the other grid
		+ cost of	Routed as a regular
		data packet	Packet
Data packet	Different grid (wrong Tran.)	1 (min), $(N/2)$ (max) +	To reach the other Tran
		l (min), $(N/2)$ (max) +	To reaclh the other grid
		Cost data packet	Regular data packet

7.1. General communication setup

This section will provide a study of a simple 16 node (4 Rows, 4 Columns) setup with no transverse dimension. Node 0 is used to send a data packet to each node in the network (Node 0–15). At the reception of a data packet the receiving node sends a reply back to Node 0. Node 0 waits for these replies before sending a new request. The whole process is finished when Node 0 receives the response from Node 15. The time consumed to finish this sequence of communication is then reported. The time is reported as an increment of a timed counter representing the smallest granularity of operation in the node. Each node decides on transmission of data at an increment of this counter. This reported value covers both significant metrics described in Tables 2 and 3. The simulated communication scheme required 211 increments of the counter to finish with no link errors in the network.

Using the obtained values and the enumerations provided in Section 6 an analytical explanation is now provided. A few assumptions are made to simplify the analysis. The first is considering the cost of physically sending a packet to be negligible (TS_{token} or TS_{data}) and adding this cost to the hop count. The second is considering the hop cost to be equivalent to the duration of one increment of the timed counter. The physical implementation used in recording this data was attempted to be close to this assumption to ease the discussion. The cases used to carry this study are shown in Fig. 7.

Each data communication on the network can be to a destination on the same ring, a different row and column ring, or a "Wrong Route". Sending on the same ring requires a TW_{token} according to Table 2 and an average of $N/2$ hops according to Table 3, overall time ($TW_{token} + (N/2)$) counter iterations. Using the stated assumption that $TW_{token} = (N/2)$, the cost of sending on the ring (X) in this case is:

$$X = \frac{N}{2} + \frac{N}{2} = N = 4 \tag{1}$$

Sending a data packet to a destination on a different row and column requires the use of a merge. The sending Node has to wait for $TW_{merged\text{-}token}$ and the connection will require N hops on average. The cost of sending with a merge (Y) is:

$$Y = TW_{merged\text{-}token} + N = 3.5(TW_{token}) + N = 11 \tag{2}$$

The last communication type to consider is the "Wrong Route". This route forces an additional TW_{token} to wait for the token in the forced direction, then one hop to the corresponding ring, another TW_{token} awaiting the processing at the data queue and finally a merged communication. The total cost of the connection (W) is (where $Y1$ is the merge cost, which will be explained in Section 7.2):

$$W = 2(TW_{token}) + 1(hop) + Y1 = 5 + Y1 \tag{3}$$

Eqs. (1)–(3) will be used to analyze the communication in the networks shown in Fig. 7. In case (a), Node 0 sends requests to all the nodes on the network, and these requests will be sent to seven nodes that belong directly to the rings that Node 0 belongs to: Nodes 0, 1, 2, 3, 4, 8, and 12. The other nine nodes in the network require merge communications. The same case exists for the responses to these requests. The total communication cost for this scheme is:

$$(a) 7(X) + 9(Y) + 7(X) + 9(Y) = 254 \tag{4}$$

This analytical approach is the basis of all the analytical results that are presented in the following discussions.

7.2. Fault introduction

The following section introduces faults and reports the delay value in multiple connection schemes. The fault tolerant scheme declares a node unreachable when both its row and column ring are in error, such cases will arise with multiple faults. The graceful degradation in the performance of the network with increasing the number of faults will be represented in both the predicted and the simulated results. Fig. 8 shows five trials and the simulated counter value. The two cases presented in Fig. 7(b) and (c) will be used to predict the performance of the networking scheme.

Trials 1 and 5 have a significantly more severe impact on the operation of the network. Both of these cases have the link error on a ring that Node 0 (the source of all requests and the destination of all replies in the communication scheme) belongs to. There are two effects to that situation. Node 0 is creating all the requests and thus all requests destined to a node on the ring with the error will require a "Wrong Route", which is usually an expensive operation. When Node 0 is the destination of all the responses, because this node is on a ring with a link error, every response has to be dealt with using the fault tolerant scheme (almost always more expensive than regular operation or at the same cost). When the error is on a ring that Node 0 does not belong to, such as Trials 2, 3, and 4, the fault tolerant scheme is used in a substantially smaller portion of the communication and thus the cost is less.

For case (b) in Fig. 7, Node 0 sends to the same seven nodes directly on the rings. The six nodes 9, 10, 11, 13, 14, and 15 are again reached using a merged connection. Nodes 5–7 are also reached through merged connections, but Node 0 has to wait for its row token to send the merging request; in the other cases it can send either using the row token or the column token. This increases the initial waiting period. In this case there is probability of 0.5 that

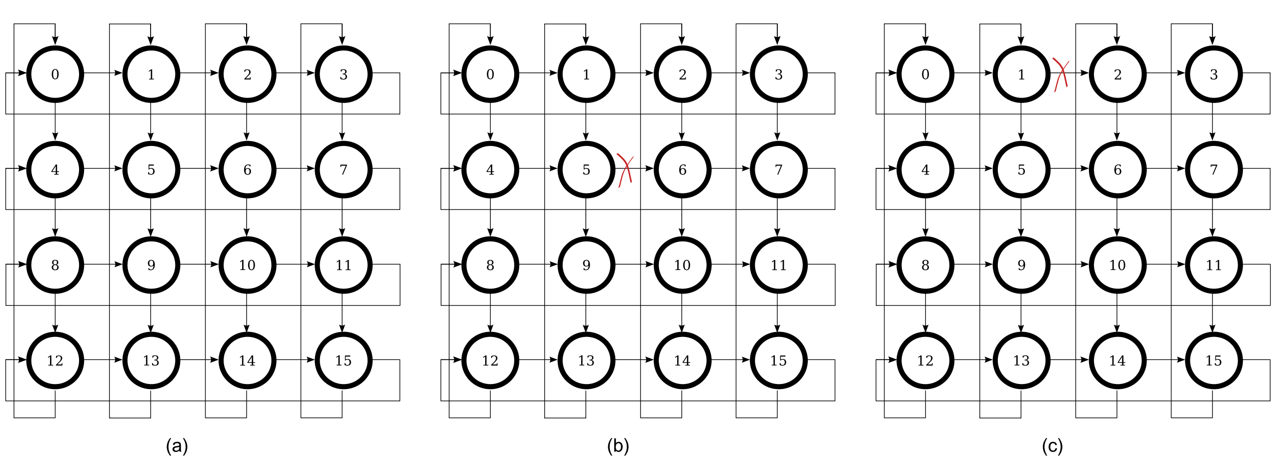


Fig. 7. The general cases for communication with one link errors are shown in (b) and (c). The link error in (c) belongs to the same ring as Node 0.

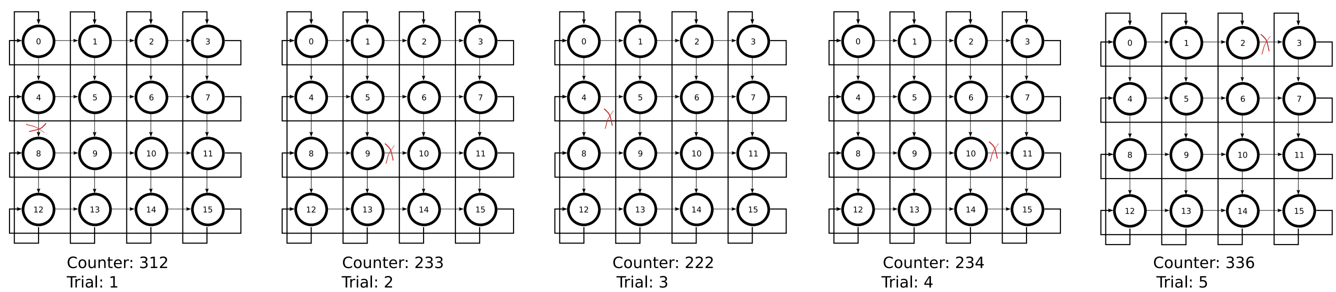


Fig. 8. Five trials of one link failures and their respective cost of communication.

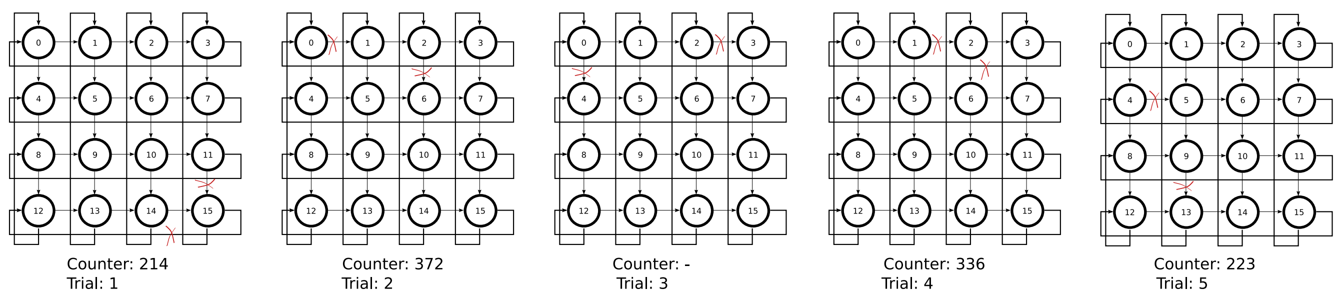


Fig. 9. Five trials of two link failures and their respective cost of communication.

Table 4
Summary of simulation values (in counter increments).

Case	Mean	Standard deviation	95% Confidence Interval	
			Lower bound	Upper bound
1 Fault not on ring[1]	228.40	5.941	221.04	235.77
1 Fault on ring	321.80	14.37	303.95	339.64
2 Faults not on ring	267	5.29	258.51	275.48
2 Faults on ring	392.5	26.85	355.23	429.76
3 Faults not on ring	302	1	300.39	303.60
3 Faults on ring	459	8.36	447.38	470.61

[1] Ring denotes a ring directly attached to Node 0, in this example (Row 0 or Column 0).

it gets the wrong token first and thus has to wait for another TW_{token}. Thus Nodes 5–7 will have an increment of $0.5(TW_{token})$, the merge cost Y will be represented by $Y1$ in this case where $Y1 = Y + 0.5(TW_{token})$. The seven nodes on the direct rings send back on those rings. Nodes 9, 10, 11, 13, 14, 15 communicate using regular merging, while Nodes 5–7 require a "Wrong Route" to reach Row 2 and then a merge to reach Node 0. The communication cost for this case is:

$$(b)14(X) + 12(Y) + 3(Y1) + 3(W) = 14(X) + 15(Y) + 3\left(\frac{TW_{token}}{2}\right) + 3(5 + Y + 1) = 275 \tag{5}$$

For case (c), Node 0 sends to Nodes 0, 4, 8, 12 directly on the ring, and to the nine merging nodes using regular merges with the addition of the extra waiting period because there is only one merge that is possible. For Nodes 1–3, a "Wrong Route" is used; this route reaches Node 4 and merges are instantiated from there. The responses in this case follow the same communication routes as the requests and thus the total cost for this case is:

$$(c)2(4(X) + 9(Y1) + 3(W)) = 2(4(X) + 12(Y1) + 15) = 350 \tag{6}$$

The values calculated using (4)–(6) are representative of the values collected from the simulator and reported in Fig. 8. These calculations offer a way to estimate the cost of communications and to understand the differences in the simulated data. The discrepancies in the values are attributed to the assumptions taken and probabilities inherent in creating these enumerations. From the calculated and recorded results, it can be seen that schemes requiring "Wrong Routes" can increase the cost of communication significantly.

The same general communication scheme was tested with two link errors in the network. Fig. 9 presents a portion of the trials tested and the cost values for each case. The implemented fault tolerant scheme assumes that if a node has an error on its row and column then that node is unreachable, which is why there is no result for case 3 in Fig. 9. When the fault is on one of the rings of Node 0, the cost increases by a large amount, as shown by Trials 2 and 4. Since both faults in Trial 4 belong to one ring, then the effect is the same as that of one link error on that ring. Another point

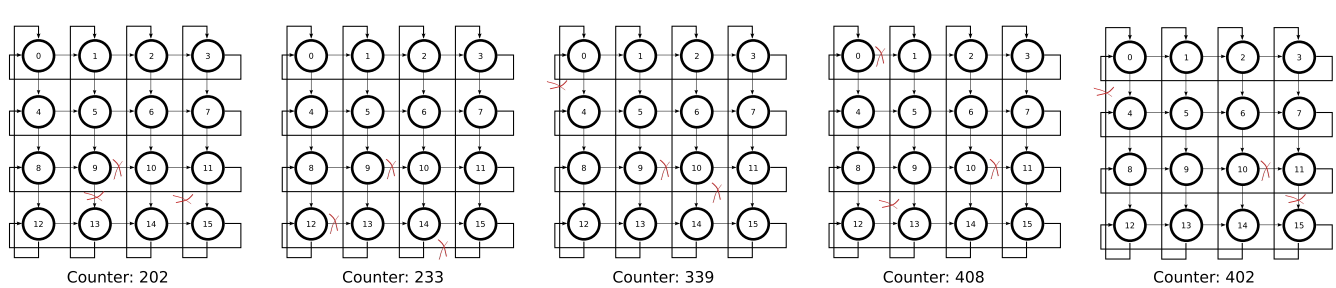


Fig. 10. Five trials of three link failures and their respective cost of communication.

to be added in this discussion is that the sender will discard any request for a node with an error on its row and column, Node 4 in Trial 5, for example. This saves on the overall time needed because that request is never sent. In the case that one error is on a ring connected to Node 0, for example, Row 0 in Trial 2, Node 0 will not learn of the column error on Column 2, and thus will try to send to Node 2 without discarding the data packet. Node 0 in this case sends to Node 4 using a "Wrong Route"; Node 4 discards the data packet, but Node 0 waits for a timeout period before it realizes that the connection does not exist. In our case that timeout was set to 50 counter cycles. Following this analytically, the total cost is (which compares favorably to the cost of 372 obtained by simulation):

$$8(X) + 15(Y1) + 7(W) + 50 = 381 \quad (7)$$

Table 4 provides a summary of the simulation values collected for different fault numbers. The reported values manifest the graceful degradation of the networking performance with the addition of faults (Fig. 10).

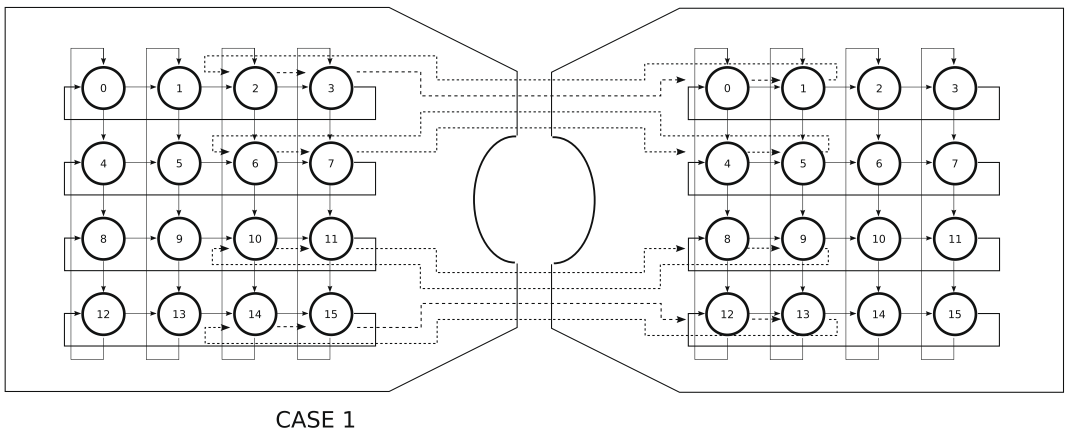


Fig. 11. The transverse links are used to connect both networks on the front and the back of the vest.

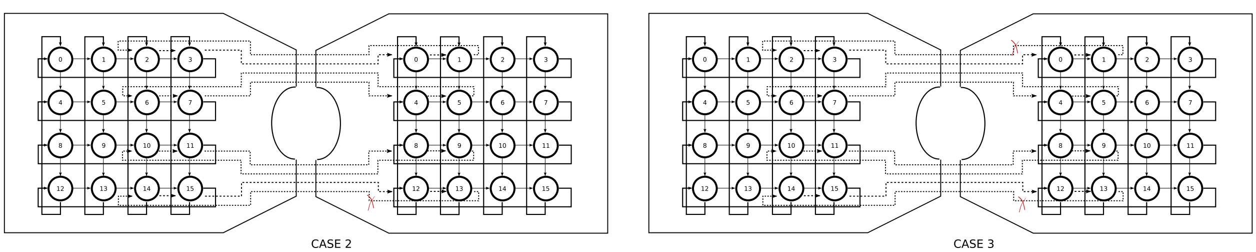


Fig. 12. The two cases provide means to study the effect of an error on the used transverse links.

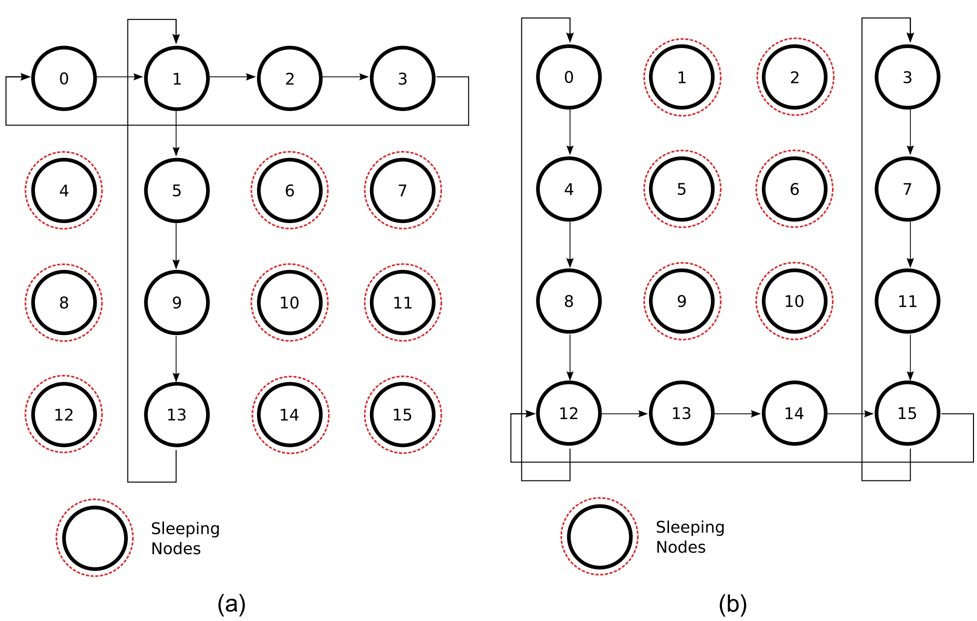


Fig. 13. The 16-Node Beamformer with the insignificant nodes in a dormant state.

Case	Simulation Result	Analytical Result	Difference	Description
1	34	36	2	16-Node Beamformer
2	81	98	17	Shirt Case 2
3	92	100	8	Shirt Case 1
4	94	108	14	Shirt Case 3
5	211	254	43	GCP No errors
6	222	275	53	Gcp 1 error
7	233	275	42	Gcp 1 error
8	234	275	41	Gcp 1 error
9	312	350	38	Gcp 1 error
10	336	350	14	Gcp 1 error
11	372	381	9	Gcp 2 errors
GCP: General Communication Protocol				

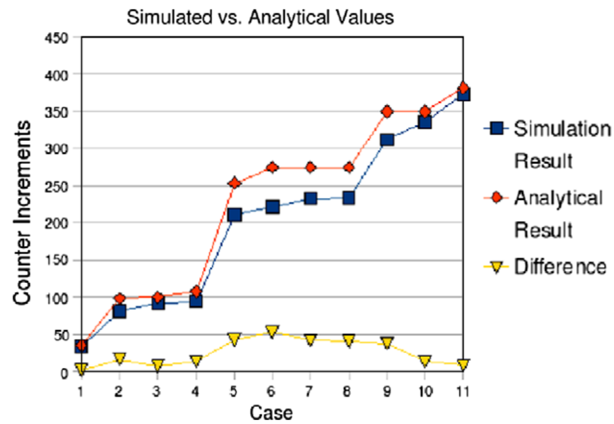


Fig. 14. There is a general trend of increase in difference between the simulation and the calculated analytical results as the value of the counter increases.

7.3. Shirt network

This section will provide data pertaining to the effects of the transverse dimension on the communication cost. The expected increase in communication cost using these links is verified analytically and by simulation. These networks are connected by the transverse dimension across the front and back textiles is shown in Fig. 11. The application protocol is as follows: Node 0 on Grid 0 (front) sends requests and waits for replies from three nodes on the front and one node on the back. Three nodes were picked at random from the front grid (Nodes 8, 14, 15) and a fourth on the back grid (Node 15). Case 1 provides the cost of sending over the transverse link. Two more cases are presented to study the effect of link failures on the transverse links. The first, Case 2 in Fig. 12, provides an error on the link used by the reply to reach Grid 0. The second, Case 3 in Fig. 12, provides errors on the both of the links used by the request and the reply and thus forces a "Wrong Route" in both cases.

For Case 1, the analytical methods provides the following cost:

$$2(X+2(Y)+2(X)+5+Y)=2(3(X)+3(Y)+5)=100 \qquad (8)$$

In Case 2, Node 12 (Back) realizes that it cannot use its transverse link and thus forces a "Wrong Route" on the column, then Node 0 (Back) will route the data to Node 2, at a cost of X and a $W=(4+5)=9$. The increase on Case 1 is waiting and one hop = 3. There is a decrease from the merge to a ring connection (Y–X–W). Then the cost should be 98. This is one of the rare cases where a "Wrong Route" results in less cost than a regular route. In Case 3, the analytical methods provides a cost of 108. The simulated values are: Case 1: 92, Case 2: 81, Case 3: 94. This test presented the increase in the communication cost by the introduction of transverse connections.

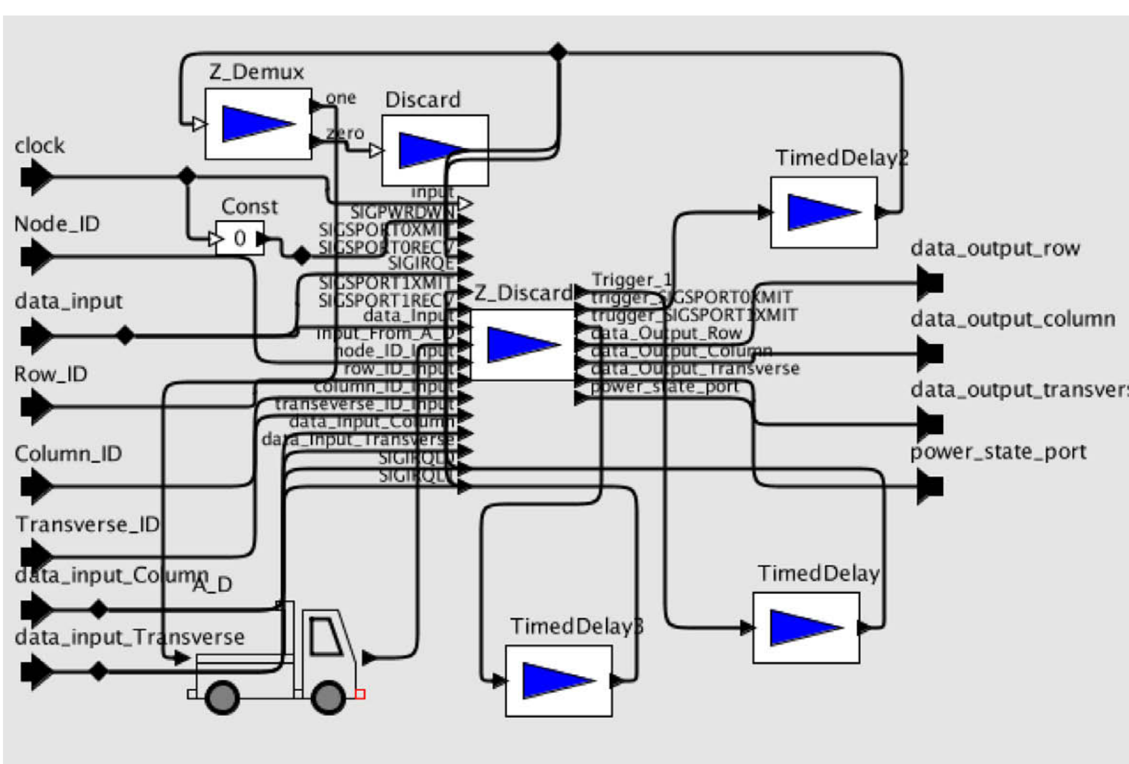


Fig. 15. The node simulator in Ptolemy, this node connects using ports to other nodes in the network and interacts with the acoustic data simulator (truck).

7.4. Sleeping Nodes on a 16-Node Acoustic Beamformer

Inactive nodes in a network can move to a dormant state to lower the power consumption of the whole network. These sleeping nodes can affect the communication cost since their effect is the same as two link failures. There is no difference between a sleeping node and a failed node to the routing scheme. The application protocol can be chosen in a way to route around sleeping nodes to save power or it can wake nodes in the needed path to save time.

The application protocol provided in this case is the following, where every ring in the 16-Node system represents an Acoustic Beamformer Array [1]. The "master" on each ring will ask for data from the other nodes and compute a location for the vehicle. The location data can be verified by checking the data from another ring; in this case the scheme uses a ring at a perpendicular to the original ring. The communication can thus be constricted to a row and a column in the system. For specific cases it is assumed that the data is redundant and only the two specific rings need to be awake in the system. If Node 1 is the node in common between the two specific rings, the network will be as shown in Fig. 13(a).

Node 0 is considered the master of Row 0 and Node 13 the master of Column 1. Both masters send requests for the three other nodes in their ring. When Node 13 receives all needed replies, it sends its value to Node 0. When the response from Node 13 is received by Node 0 a cycle of the operation is finished. The analytical cost is 36 while the simulated cost is 34. The gain in using the sleeping nodes in this case is attributed to power consumption. According to the values in [15] the average power consumed in a node is 201.42 mW. In this specific example, 9 nodes were asleep and the power consumption is decreased by 1812.78 mW.

Another application scheme is provided in Fig. 13(b), Nodes 0, 13, and 3 are masters. The application proceeds as mentioned earlier with Node 13 and Node 3 sending their information to Node 0. This application is processed under two routing circumstances, the first considers the sleeping nodes when communicating between Nodes 3 and 0 and thus routes around the sleeping nodes (Nodes 1 and 2) using "Wrong Routes" and a merge. The second wakes up Nodes 1 and 2 with a token and sends its value directly to Node 0. This application will show the effects of using the fault-tolerance routing on time and power consumption. The simulation values collected were 63 for the first case and 58 for the second case. There is a decrease in time consumed when the sleeping nodes are woken up but at an increase in the power consumption. This presents a trade-off between power and time consumption.

The implemented networking scheme was discussed in the previous section. This networking scheme provides connections while dealing with faults and sleeping nodes with gracefull degradation in performance. The simulation results displayed the effects of different situations on the networking scheme. An analytical method was provided to estimate the communication cost for specific application protocols. The comparison between the simulated and the analytical results proved the predictability of the networking scheme. This analytical method introduces more errors as it analyzes a longer time as can be seen in Fig. 14 representing the simulation times versus the analytical times predicted.

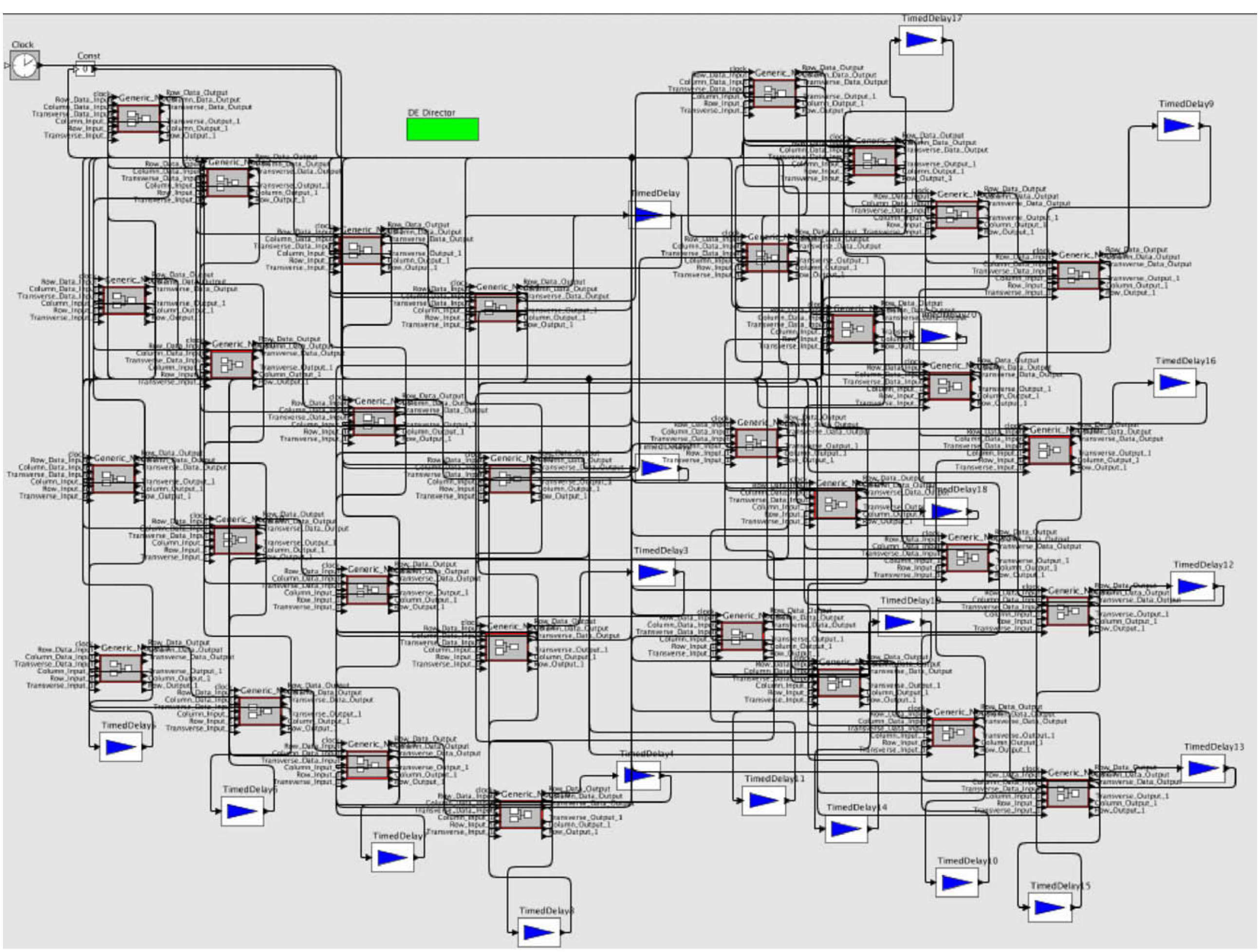


Fig. 16. Simulation of 32 nodes on two separate grids joined with transverse links.

8. Conclusion

The durability, fault-tolerance, and ease of deployment of electronic textiles promise upcoming widespread use of this technology. The interconnections in e-textiles are of utmost importance. The networking scheme in an active fabric controls its fault-tolerance, productivity and even functionality. This paper introduced and studied the Textile Token Grid which is a Token Grid network modified to map better to e-textile applications. The Textile Token Grid is a fault tolerant, power-aware, predictable and scalable network.

The presented networking scheme provides the needed support for e-textile application by providing a predictable and manageable method to connect multiple nodes in such a system. The system interacts with both faults and sleeping nodes (low-power consumption) by re-routing around trouble areas while providing graceful degradation of performance. The operation of the network was presented along with several example applications that prove its support for e-textiles and its adaptability to multiple fabric situations, including an interesting trade-off between power and time consumption highlighted in one of these examples. This paper produced a method to empirically predict the operation of the network with and without faults along with the support for sleeping nodes. The empirical values mapped well to the simulation data proving the predictability of this network and validity of the analysis of its operation.

The previous discussions manifest the appropriateness of the Textile Token Grid as a networking scheme for e-textile applications. The author plans to extend this networking architecture by arranging its functionality and extending its applicability to other research areas.

Acknowledgment

DARPA contract F30602-00-2-0548, NSF Grant CCR-0219809.

References

[1] Zahi Nakad, Mark Jones, Tom Martin, Using electronic textiles to implement and Acoustic Beamforming array: a case study, Elsevier Pervasive and Mobile Computing 3 (5) (2007) 241–246.
[2] T.D. Todd, The Token Grid: Multidimensional Media Access For Local and Metropolitan Networks, in: Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '92.
[3] J. Edmison, M. Jones, Z. Nakad, T. Martin, Using piezoelectric materials for wearable electronic textiles. In: Proceedings of the Sixth International Symposium on Wearable Computers, pp. 41–48, 2002.
[4] Z. Nakad, M. Jones, T. Martin, Communications in electronic textile systems, in: The 2003 International Conference on Communications in Computing (CIC 2003), June 2003.
[5] Tom Martin, Thurmon Lockhart, Mark Jones, Josh Edmison, Electronic Textiles for in Situ Biomechanical Measurements, in: 24th Army Science Conference, December 2004.
[6] Zahi Nakad, Mark Jones, Thomas Martin, Fault-Tolerant Networks for Electronic Textiles, The 2004 International Conference on Communications in Computing (CIC 2004), Las Vegas, June 2004.
[7] B. Firoozbakhsh, N. Jayant, S. Park, S. Jayraman, Wireless communication of vital signs using the Georgia Tech Wearable Motherboard, in: 2000 IEEE International Conference on Multimedia and Expo, vol. 3, pp. 1253–1256, 2000.
[8] VivoMetrics, Continuous Ambulatory Monitoring, Lifeshirt System, Available from <http://www.vivometrics.com/site/system.html>.
[9] K. Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill Series in Computer Science, 1993.
[10] E.L. Hahne, T.D. Todd, Fault-Tolerant Multimesh Networks, Global Telecommunications Conference, GLOBECOM '92.
[11] T.D. Todd, The Token Grid Network, IEEE/ACM Transactions on Networking, June 1994.
[12] T.D. Todd, E.L. Hahne, Multiaccess Mesh (Multimesh) Networks, IEEE/ACM Transactions on Networking, April 1997.
[13] Ptolemy Project, Heterogeneous modeling and design, 2003, [Online], Available from <http://ptolemy.eecs.berkeley.edu>.
[14] Zahi Nakad, Architectures for e-Textiles, Ph.D. Dissertation, Virginia Polytechnic Institute and State University, 2003.
[15] Tanwir Sheikh, Modeling of Power Consumption and Fault Tolerance for Electronic Textiles, Master's Thesis, Virginia Polytechnic Institute and State University, 2003.
[16] D. Lehn, C. Neely, K. Schoonover, T. Martin, M. Jones, e-TAGS: e-Textile Attached Gadgets, Communication Networks and Distributed Systems Modeling and Simulation Conference, January 2004.
[17] J. Edmison, D. Lehn, M. Jones, T. Martin, E-textile Based Automatic Activity Diary for Medical Annotation and Analysis International Workshop on Wearable and Implantable Body Sensor Networks, 2006. BSN 2006.
[18] Thomas Martin, Mark Jones, Joshua Edmison, Tanwir Sheikh, Zahi Nakad, Modeling and simulating electronic textile applications, in: Proceedings of the ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES '04), Washington, DC, pp. 10–19.
[19] D. Marculescu, R. Marculescu, N.H. Zamora, P. Stanley-Marbell, P.K. Kholsa, S. Park, S. Jayaraman, S. Jung, C. Lauterbach, W. Weber, T. Kirstein, D. Cottet, J. Grzyb, G. Troster, M. Jones, T. Martin, Z. Nakad, Electronic textiles: a platform for pervasive computing, Proceedings of the IEEE 91 (12) (2003).
[20] D. Marculescu, R. Marculescu, P. Khosla, Challenges and opportunities in electronic textiles modeling and optimization, in: Proceedings of the 39th annual Design Automation Conference New Orleans, Louisiana, USA 2002.
[21] P.S. Stanley-Marbell, D. Marculescu, R. Marculescu, P.K. Khosla, Modeling analysis and self-management of electronic textiles, IEEE Transactions on Computers 52 (8) (2003).