Networked Surfaces: A New Concept in Mobile Networking

James Scott* Frank Hoffmann* Mike Addlesee† Glenford Mapp† Andy Hopper*†
*Laboratory for Communications Engineering, †AT&T Laboratories Cambridge
Cambridge University Engineering Department
{jws22,fh215}@eng.cam.ac.uk {mda,gem,hopper}@uk.research.att.com

Abstract

Networked Surfaces are surfaces which provide network connectivity to specially augmented objects, when these objects are physically placed on top of the surface. When an object (e.g. a notebook computer) connects, a handshaking protocol assigns functions such as data or power transmission to the various conducting paths that are established.

This paper describes the position occupied by this concept in the world of networking, presents an overview of the technology used in its realisation, describes the current prototype implementation, and outlines the potential implications of its introduction.

Keywords: Mobile Networking, Ubiquitous Computing, Sentient Computing

1. Introduction

The Networked Surfaces Project at the Laboratory for Communications Engineering (LCE) was started in October 1998. Initial exploration focused on a floor-based network for wearable computer users, but the focus has quickly shifted towards a desk environment. As of September 2000, a prototype surface has been built and tests are underway to explore the limits of the technology.

This paper compares Networked Surfaces to other networking technologies, outlines the technologies used in their realisation, describes the current prototype implementation, and discusses implications of their use.

This section describes the Networked Surface concept, its position in the networking world, the advantages it exhibits over other networks, and an example application, the “Networked Desk”.

1.1. The Networked Surface concept

We describe a surface (such as a desk or floor) as “Networked”, if a suitably augmented object can acquire *connectivity* to a data and/or power infrastructure, simply by being in *physical contact* with that surface.

“Connectivity” is achieved by providing a number of electrically independent paths, or “links”, between the surface and object, which are allocated to “functions” such as data transmission or power. Different objects may require different functions, and the functions which are available to objects may differ from surface to surface.

By “physical contact”, we mean that the object may occupy any position and orientation on the surface. This flexibility is achieved using a special layout of conducting “pads” on the surface and object. When the object touches the surface, pairs of surface and object pads provide communications channels. While current research focuses on direct electrical conduction between the pads, other means of communication such as capacitive coupling and induction may be explored in future.

1.2. Networking for Mobile Users

Mobile networking can be achieved in two ways, either with physical wired/wireless networking, or using “virtual networking”.

Virtual networking can free users from carrying mobile devices, if computing hardware becomes so prevalent that a user will find adequate facilities wherever they might go. Technologies like VNC [11] allow users to access their working environments irrespective of location. However, such a prevalence of computing facilities is unlikely in the foreseeable future, so we shall restrict our attention to real-world (i.e. “wired” and “wireless”) networking.

Wired networking requires a physical medium for network traffic, which must be attached to each node. Available bandwidth may either be shared between many connections or dedicated to a particular connection. Also, being wired for networking means that another (or the same) wire could easily provide power to the device.

However, the need to be physically attached to the network leads to a lack of mobility. One solution to this problem is the use of disconnection-friendly file systems [5], using caching and other techniques to facilitate useful work whilst in a disconnected (and therefore mobile) state.

Wireless networking stands at the other end of the spectrum. Bandwidth is inherently shared inside "cells", which are delimited by the range of the transmitted signals. Mobility is of course "free" within a cell, but there is the problem of handover if the mobile devices cross cell boundaries.

Mobile devices also need batteries to power them; these must be charged from time to time, causing a loss of mobility during these periods.

1.3. Characteristics of Networked Surfaces

Networked Surfaces attempt to combine the best qualities of both wired and wireless networking, for indoor environments. On one hand mobility is effectively provided, because the use, transport, and connection of wiring is not required. On the other hand, wired-style connectivity and bandwidth are present, and power may be provided to charge batteries.

Networked Surfaces allow objects to be networked in a simple, yet powerful manner. The network is flexible in that many different types of objects may be used on the same surface, and convenient in that objects are easily attached to and detached from the network. They are upgradeable, as new types of data bus may be added without replacing the physical surface. The bandwidth available is also easily increased; a surface can always be provided with more data busses.

The use of such a generic interface as a surface lends itself to interoperability, but without the possibility of misconnection. This is in contrast to wired networks, where the user is either forced to carry many types of cabling (if different cables are used for different functions, such as RS-232 data and Ethernet data), or, if a single generic cable type is used, the user may misconnect the system causing lack of function at best and damage to hardware at worst. On the surface, the object "asks" for the function it requires electronically, thus creating a situation where no wiring is required, and where the user may remain ignorant of minutiae such as the networking protocols being used.

Providing power over the surface requires the use of current protection hardware, as shorts are inevitable in a situation where a spilt drink can temporarily connect any "wire" on the network to any other. However, doing so frees the user from worrying about battery levels; if an object is habitually stored on a Networked Surface, there will rarely be insufficient charge for useful mobile operation.

1.4. Target environments for Networked Surfaces

In order to explore the scope of usefulness of Networked Surfaces, devices are grouped into five categories, namely fixed incommunicado devices, fixed networked devices, mobile devices, wearable devices, and devices for outdoor use. The applicability of Networked Surfaces for each category is examined below.

Fixed incommunicado devices include clocks and other normally isolated appliances. The Networked Surface can provide power for them, allowing those which use batteries to never need replacements, and those which use mains power to be free of the wires connecting them. Some use can also be found for networking such devices in a low-bandwidth sense, for example a clock could also show up-to-date news or share prices. Alternatives to the Networked Surface include power-line networking [12], and radio-based Piconet [1] and Bluetooth [2].

Fixed networked devices such as computer peripherals, computers themselves and telephones can use Networked Surfaces for networking and power; instead of using normal wiring and wired networks such as Ethernet. The disadvantage lies in the expense, which will always be more than a simple wire, however, the Networked Surface provides more scope for mobility than a wire, which tethers an object to a certain space.

Surfaces are also useful for mobile devices, in order to communicate (e.g. Palm Pilot) or just to recharge their batteries (e.g. mobile phone). The disadvantage is that the surface does not give full spatial mobile communications, unlike a radio-based system.

For devices that are "wearable" [10], it may actually be an inconvenience to take the device off and place it on the surface, for recharging or communications, so a radio solution is more suitable, or a solution such as Personal Area Networks (PAN) [17].

For devices used outdoors, the Networked Surface is impractical for networking due to the difficulty of networking a useful portion of the environment, and the problems inherent in exposing conductive pads to rain, etc. The Networked Surface is therefore primarily considered as an indoor technology.

1.5. The Networked Desk

One appealing use of Networked Surfaces is as Networked Desks. In today's computer-enabled workspaces, there is an abundance of wire on and around desks, causing clutter and making rearrangement of the working environment difficult. By providing networking and power through the desk itself, most of these wires could be eliminated.

Such a desk could simultaneously provide computer-peripheral connections (e.g. replacing USB wiring), computer-computer connections (e.g. replacing ethernet wiring), power connections, and even connections for telephones. Note that it is only the wiring that is replaced; the protocols themselves can still be used when appropriate, so existing hardware need not be extensively modified to become Networked Surface capable.

For power, it is of course infeasible to supply mains voltages due to the likelihood of shorts. However, many devices do not require mains-level voltages; a separate transformer is often used to reduce this to, say, 12V DC. The output of such transformers may be safely carried over a Networked Surface, with current limiting hardware to safely detect and handle shorts.

A "desk" of particular interest might be one in a conference room, where it is particularly important to have a multi-user, multi-purpose, easy to operate, and uncluttered environment.

2 Technology

This section explains the overall architecture of the networked surface implementation, and discusses the technologies that have been used in their design, from the physical surface itself through to data transport concerns.

2.1. Overall Architecture and Nomenclature

A Networked Surface has many hundreds or thousands of electrical "pads", only some of which will be in use at any one time. This poses a huge control problem - in order to detect new devices, each pad will have to be monitored closely. Using a single entity to handle this would limit scalability, therefore a distributed approach is taken.

The surface hardware is structured as follows. The surface pads are grouped into "tiles", each of which has a "tile controller". This is responsible for handshaking new connections by communicating through the surface pads with "object controllers" on the object. These controllers are only active during the handshaking phase; after an object is patched through to its required "functions", the controllers can no longer participate.

The functions, which may be data or power-related, are provided as busses on the surface. The data protocols used must therefore be compatible with a multi-drop bus architecture. Provision of the functions, i.e. driving power lines and bus-mastering data lines, is performed by the "surface manager", a single entity which communicates with all connected objects on the surface; this manager also bridges between the surface and the Internet at large. A standard PC may be a suitable surface manager.

On the object side, the "object manager" provides the object with access to the data busses and/or power, and exchanges control traffic with its surface counterpart. In reality, the object pad controller and object manager may be implemented as a single unit (since there is no scalability problem for an object).

This architecture is illustrated in Figure 1.

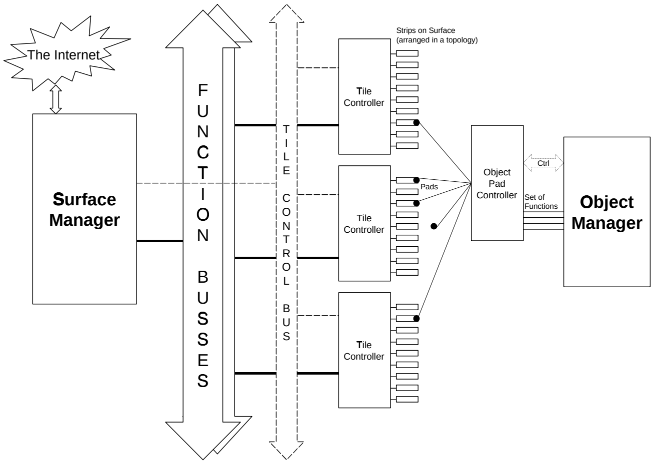

Figure 1. Networked Surface Architecture

2.2. Topology

In order for an object to establish a network connection, a number of independent electrical conducting paths, or "links", are required. The layout of pads on the surface and object to achieve this is known as the "topology". Exactly how many links are required depends on the type of network, and whether the provision of power is desired. The simplest useful connection types require two links, namely ground and either half-duplex data or power. More complicated connections may have six or more links, for large data busses etc.

A topology must guarantee that for any position and orientation of the object on the surface, enough links will be present to allow a connection to be made. Furthermore, a topology should support both simple (two link) and more complicated (six or more link) objects.

2.2.1. Design. There are many possible choices for the topology, including hexagonal, grid-like, and brickwork-like arrangements of surface pads.

The chosen topology is one where rectangular strips cover the surface, and a small number of circular pads, themselves arranged in a circle, form the object. The gaps, or "margins", between the strips are chosen to be strictly bigger than the size of the object pads.

This topology, which is illustrated in Figure 2, has the following properties:

- It guarantees no shorts between surface pads, thus making full use of of all strips available under a particular object.

- It minimises the footprint required of an object, since the object pads are small and their spanning area is bounded.
- It is geometrically simple and therefore easy to manufacture.
- It is extensible to different link requirements, since for any given surface of this style and any given number of links required, it is possible to produce an object footprint that will guarantee that number of links, so long as the surface strips are wide enough (the topology relies on the fact that at most two "columns" of strips are reachable by an object at any one time).

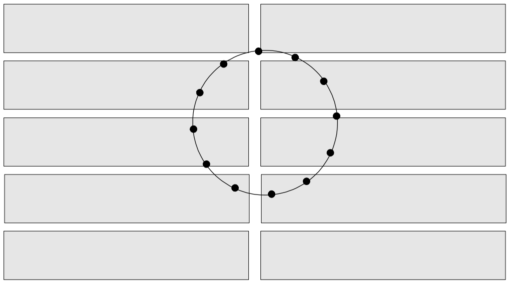

Figure 2. Example layout for chosen topology

2.2.2. Results. In order to get actual workable dimensions for the chosen topology, a simulation was used to find the minimum size of object footprint required for particular numbers of links. These tests were done for a surface with the characteristics outlined below.

- Width of strip (including margin): 2.08 cm. This is chosen for ease of manufacture (24 strips per 50cm tile), without loss of generality (the object footprint size becomes the variable).
- Length of strip (including margin): 12.5 cm This is chosen for ease of manufacture, and is irrelevant to the results below, so long as an object never touches pads from more than two "columns".
- Size of margin: 0.3 cm. This limits the size of the object pads.

2.3. Object Connection and Disconnection

New objects are discovered and connected using a "handshaking protocol". When disconnection occurs, the pads involved must return to a state ready for the next handshake.

Links Required	Object Pads Required	Diameter of Object (cm)
2	5	2.64
3	9	4.62
4	12	6.78
5	16	8.82
6	19	10.80

Table 1. Minimal object specifications for the surface described above

2.3.1. The Handshaking Protocol. The handshaking protocol operates upon the pads which are currently unconnected; such a pad is said to be in "handshaking" mode. When a connection is established, the pads it uses are said to be in "connected" mode, and do not participate in further handshaking.

One difficulty with handshaking is that the object is not commonly grounded with the surface, so simple transmission on one pad at a time will not work. To achieve a common ground, a strategy of "grounding by consensus" is used, whereby the object takes the average voltage level of its pads, using a resistor network, to be the ground value. This is illustrated in Figure 3.

Using this effect, the links are discovered as follows. Tile controllers periodically sends beacon messages along each surface pad, while holding other ("handshaking" mode) pads around that pad to ground. The object controller uses comparison against the consensual ground to detect this beacon on one of its pads, and replies with a "function request". After a further exchange of identification data, both controllers put their respective pads into a "standby" mode.

When that surface pad is next due for a handshaking beacon, the tile controller instead sends a "standby beacon", incorporating data so that the object knows that link still "belongs" to it. The object can use this mode to request further information from the tile controller, or to wait for enough links to become available for a connection, before finally transmitting a "connect" message to each pad.

One of the connected links will be the function "ground", thereby commonly grounding the surface and object so that future communication need not use "grouding by consensus".

2.3.2. Disconnection. Once the "connect" message is issued, the pad controllers have no means of communicating with one another; the connection is handed off to the surface and object managers, which must therefore perform the task of disconnection detection. When disconnection is detected, the managers cause the relevant pads to be re-inserted into the handshaking protocol by sending control messages to their respective pad controllers.

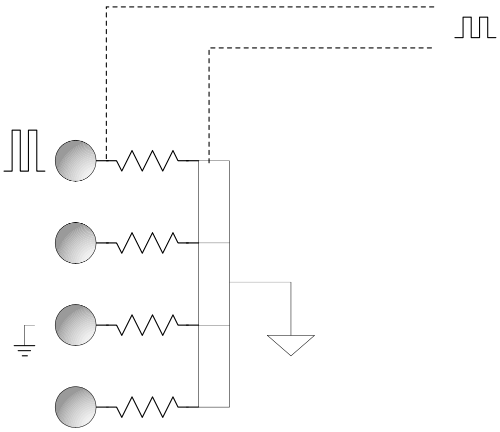

Figure 3. Resistor Network used for "grounding by consensus" in the Handshaking Protocol

However, in order to identify the correct surface pads, the surface manager must know the mapping between objects and the surface pads they are using. This is achieved by having the object collate this information during handshaking using the "standby" mode to make appropriate queries, and by mandating the object manager to send said information to the surface manager immediately on connection.

Furthermore, it is noted that an incomplete handshake, or a malicious object, could result in the pad data transmission never taking place, and therefore result in pads being "stuck" in a connected mode, but actually doing no useful work. This is solved by causing the tile controllers to set time-outs on connected pads, bringing them automatically back into the handshaking system, unless the surface manager explicitly instructs them not to, which the surface manager would do only on receiving the pad information from the object. Thus, the handshaking protocol is made robust.

There remains the problem of what constitutes a "disconnection", and how this is detected. Disconnection is defined as one or more of the pairs of previously connected pads losing connectivity, for example if the object were either picked up or moved. Disconnection detection can take place using one of two methods. Firstly, a hardware-based method may be able to tell if, for example, an object has stopped drawing current along a "power" line.

Secondly, objects may be required to transmit periodic "heartbeat" packets on data busses, failing which the surface may conclude that the object has disconnected. Similarly, an object may disconnect its pads if it has not heard the surface's "heartbeat" in a particular period. Note that erroneous disconnection (due to network traffic causing the heartbeat to be lost, etc) is not too costly, since a new connection will be immediately re-established.

2.4. The Surface Architecture

Once the tile and object controllers have finished their negotiations, a connection is set up between the surface and object managers, which perform the actual networking. The networks available are provided as multi-drop data busses, including I^2C [8], which handles data rates up to 3.4Mbit/sec, or a custom high-speed multi-drop bus based on the BLVDS [3] transmission scheme. Support can also be provided for phone jacks or simple serial interfaces such as RS-232 (e.g. for PDA synchronisation). However, the latter is not multi-drop compatible, requiring the surface to ensure that only one connection could be made to that interface at a time. Alternatively, some buffering and retransmission by hardware on the object side could allow serial data to be carried over, say, an I^2C bus.

The use of a multi-drop architecture for data seems to sacrifice one of the features of wired-style networking, that available bandwidth can be provided in a dedicated fashion to many devices simultaneously, and not shared. However, this property is instead achieved by employing multiple busses on the surface.

The provision of more than one data bus gives rise to many interesting properties making use of asymmetries between the busses. For instance, different busses may span different areas of the surface, perhaps with more busses on high-usage areas; this would result in partitioning of bandwidth by location. Another option is to have different bus flavours running simultaneously, for example the I^2C bus and the BLVDS bus described above; this gives more powerful devices higher bandwidths, whereas slower devices are only required to support a simple low-rate bus.

A further interesting possibility is the use of busses to enforce Quality of Service guarantees on the surface, by allocating devices to busses according to bandwidth requirement. Finally, the busses may be used to perform different functions, in particular a bus can be dedicated as an "arbitration channel" for other busses, allowing reservation of bandwidth out-of-band without interrupting the flow of data, and allowing control messages such as the "heartbeats" to also not impinge on the data flow.

2.5. Data transport

While simpler devices on the surface may have their own bespoke transport-level protocols, such as the Palm Pilot's RS-232 "Hot-Sync", more complicated devices will usually be built on top of a TCP/IP stack on the object. The surface therefore needs to provide support for TCP/IP.

However, the Networked Surface is peculiar in that disconnection and reconnection can occur often. As such, unmodified TCP/IP may well be a bad choice, due to its inefficiency in conditions of high packet loss, which it assumes to be due to congestion [9]. On a Surface, congestion is highly unlikely, as there is no routing in broadcast situations such as the multi-drop busses, and therefore no bottleneck routers at which congestion can occur.

To make TCP/IP disconnection-aware, there are several strategies that could be employed, the most obvious being to re-write the TCP code to include a new "disconnected" state, in which the transfer is effectively frozen until re-connection moves the TCP session back into another state. However, this requires re-implementation of TCP on every object that the Surface would like to support, which is not ideal.

A different approach, and the one taken in this project, is to make the Surface link-layer "smart". This is achieved by buffering one or more recent TCP packets for each TCP connection, and when re-connection occurs, immediately re-sending those packets on TCP's behalf, in order to "kick-start" the connection. In lieu of this impetus, TCP simply waits for a timeout, which due to exponential backoff techniques may not occur until seconds or minutes after reconnection.

2.6. Power

The provision of power across Networked Surfaces is one of the most compelling reasons for their use, in that it removes the ever-present need to plug in existing mobile devices from time to time. However, such provision also immediately brings up many human safety issues. However, while humans need protection, the networking circuitry in the surface and objects is actually much more sensitive; a surface which protects this circuitry would intrinsically be safe for use by humans.

As previously discussed, power can be in many cases be provided at a low DC voltage level, since many portable devices only require these voltages. Current requirements for these devices can further be reduced by noticing that time-to-recharge is much less of an issue, since networking the surfaces they are placed on would result in being "plugged in" a higher percentage of the time. However, even at low voltages, shorts can damage circuitry, unless current-limited.

To prevent this, configurable current-limiting and short-detecting hardware can be used on all pads, both surface and object. Such hardware immediately disconnects a pad on detection of a current higher than reasonable for the appropriate link, and re-enables the pad when the short is removed, allowing the connection to be re-established.

3. Implementation

This section discusses the current state of the project, and outlines the metrics by which the performance of the network can be measured.

3.1. Description of prototype

The prototype Networked Surface includes physical implementations of the tile and object controllers, and the surface and object managers referred to in Figure 1. A photograph of the prototype is shown in Figure 4.

The prototype tile and object controllers use PIC microprocessors to execute the handshaking and tile control protocols, which interface with FPGAs controlling the switching of the functions assigned to pads on the surface. Both these technologies lend themselves to quick prototyping and ease of reprogramming, facilitating rapid development of the system.

The object manager function can be fulfilled by the object controller for simple objects, for example those requiring only RS-232 data connectivity, or objects just requiring power. For more complex objects such as notepad computers, they themselves can fulfill the role of object manager.

The surface manager role is performed by a software driver in a PC with a custom PCI card, which includes FIFO buffers for incoming and outgoing data on all busses, implemented in an FPGA. Again, this solution is designed to offer maximum flexibility.

3.2. Metrics for evaluation

Tests on the characteristics and limits of Networked Surfaces are currently underway. The parameters which are being explored, and which will define the potential of this technology, are discussed briefly below.

The maximum bandwidth available to devices on the surface will be limited by the characteristics of the electrical conducting path, which includes a wide strip of metal on the surface, and conduction through touching surfaces rather than permanently connected wires.

The minimum connection establishment time will determine many aspects about the usability of the surface, if it were to be low, then the user of an object on the surface need not worry about accidentally moving the object slightly, but if it were to be high, the penalty for slight movement may break quality of service guarantees.

Another parameter is how closely the objects on the surface can be placed. Whereas the surface can accommodate a single object at any position and orientation, as soon as one object is using pads on the surface, this creates one or more dead zones where a second object would not acquire enough pads to make a full connection. This may lead to a maximum "density" of objects on the surface.

Finally, the maximum power that can be provided through the surface, while protecting the circuitry and that of the objects on it, will determine its usefulness in freeing

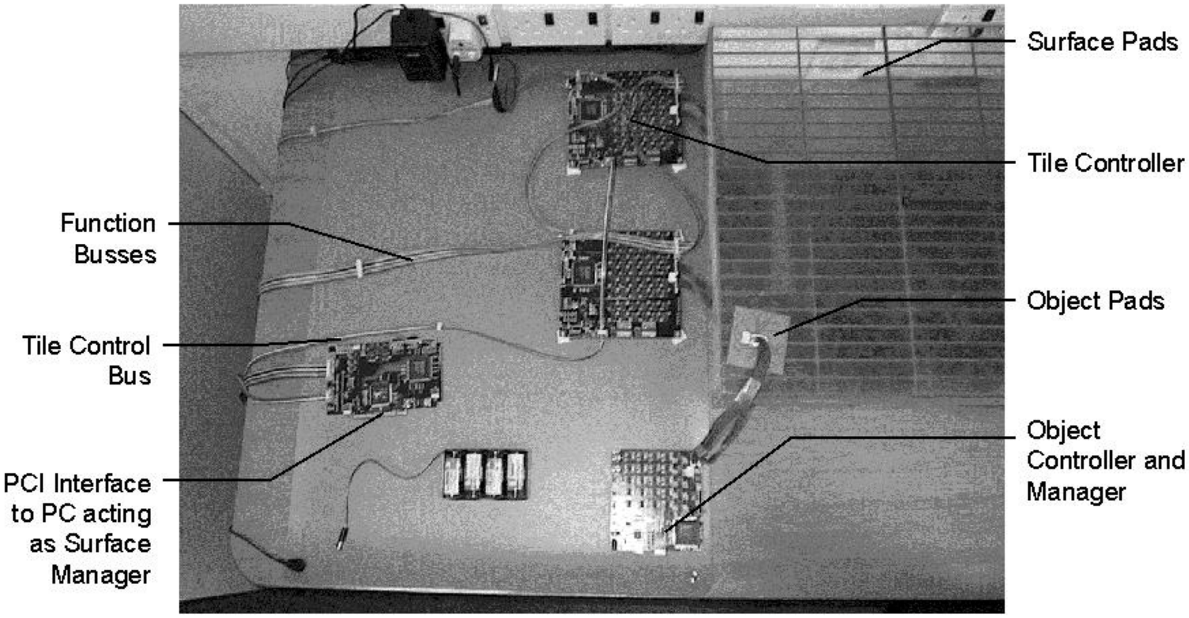

Figure 4. Photograph of prototype Networked Surface

users from power cabling for seldom-moved objects, and from the need to recharge battery-powered mobile objects.

4. Implications

This section discusses the implications of using Networked Surfaces, with reference to the fields of Ubiquitous Computing [16] and Sentient Computing [7].

4.1. Ubiquitous networking

The Networked Surface uses an interface very familiar to humans, namely touch. This is in line with the goals of Ubiquitous Computing, i.e. it facilitates a move away from interaction using mice and keyboards with monolithic computers, and towards the embedding of computers in everyday environments, such as desks and floors, using simple intuitive interfaces.

As mentioned before, the “touch” interface frees users from having to know about cabling, bus types, power requirements, and so on. The object, being aware of its capabilities and requirements, can choose for itself from a “menu” of functions that a particular surface can provide, completely transparently to the user.

Other work focusing on embedding computing into everyday environments includes that by Gellersen *et al.* on the MediaCup [6], which showcases embedded awareness in a coffee cup scenario, and by Cooperstock *et al.* [4] on a reactive videoconferencing environment, as well as the Piconet project already mentioned.

The authors regard such work as complementary, for instance the MediaCup and various elements of the videoconferencing environment may be made Networked Surface capable, allowing them convenient high-bandwidth networking, and also the ability to add other facets of awareness such as location, discussed below.

4.2. Sentient surfaces

Sentient Computing involves providing computers with data about the environment them. With the data available to the Networked Surface, the location and orientation of all objects sitting on the surface can be tracked, and this information can be made available to interested applications.

This information would be used to make devices behave more intelligently, for example, if a keyboard were picked up and put down closer to a different monitor, then the system might infer that the user wants input from that keyboard to be re-directed to the computer associated with the second monitor.

Other indoor location technologies include the Active Badge and Active Bat systems developed at AT&T Labs [14, 15], and the Locust Swarm system by Starner *et al.* [13]. The Networked Surface is set apart from these by the fact that it is primarily a network, with location information as a useful “side effect”.

5. Conclusions

Networked Surfaces are a new network medium, sharing qualities of both wired and wireless technologies. This paper has described the position occupied by this technology in the world of networking, presented an overview of the technology and the prototype implementation constructed, and described the potential implications of their introduction.

6. Acknowledgements

The authors would like to thank everyone at the LCE and many people at AT&T Laboratories Cambridge for helpful discussions and comments.

James Scott is pursuing a PhD, funded by the Schiff Foundation of Cambridge, and by AT&T Laboratories Cambridge.

Frank Hoffmann is also pursuing a PhD, funded by AT&T Laboratories Cambridge.

Mike Addlesee and Glenford Mapp are research engineers at AT&T Laboratories Cambridge, where Andy Hopper is Managing Director. Andy is also Professor of Communications Engineering at the Engineering Department of the University of Cambridge.

References

[1] F. Bennett, D. Clarke, J. B. Evans, A. Hopper, A. Jones, and D. Leask. Piconet - embedded mobile networking. *IEEE Personal Communications*, 4(5):8–15, October 1997.

[2] Bluetooth. A low-cost short-range radio networking solution. http://www.bluetooth.com/.

[3] BLVDS. A high speed multi-drop bus signalling architecture. http://www.national.com/appinfo/lvds/.

[4] J. R. Cooperstock, S. S. Fels, W. Buxton, and K. C. Smith. Reactive environments. *Communications of the ACM*, 40(9):65–73, September 1997.

[5] M. E. Fiuczynski and D. Grove. A programming methodology for disconnected operation. Technical report, Department of Computer Science and Engineering, University of Washington, March 1994.

[6] H.-W. Gellersen, M. Beigl, and H. Krull. The MediaCup: Awareness technology embedded in an everyday object. In *1st Int. Sym. Handheld and Ubiquitous Computing (HUC99)*, 1999.

[7] A. Hopper. Sentient computing. Royal Society Clifford Paterson Lecture 1999.

[8] I^{2}C. A networking solution for integrated circuits. http://www-us2.semiconductors.philips.com/i2c/.

[9] A. Kumar. Comparative performance analysis of versions of TCP in a local network with a lossy link. *IEEE/ACM Transactions on Networking*, 6(4):485–498, 1998.

[10] S. Mann. ‘Smart Clothing’: Wearable multimedia computing and ‘personal imaging’ to restore the technological balance between people and their environments. In *Proceedings of ACM Multimedia ’96*. ACM, 1996.

[11] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper. Virtual Network Computing. *IEEE Internet Computing*, 2(1):33–38, February 1998.

[12] SmartHome. Networked home products using X10 technology. http://www.smarthome.com/.

[13] T. Starner, D. Kirsh, and S. Assefa. The Locust Swarm: An environmentally-powered, networkless location and messaging system. In *Proceedings of the International Symposium on Wearable Computing ’97*. IEEE Computer Society, 1997.

[14] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The Active Badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, 1992.

[15] A. Ward, A. Jones, and A. Hopper. A new location technique for the Active Office. *IEEE Personal Communications*, 4(5):42–47, October 1997.

[16] M. Weiser. Some computer science problems in ubiquitous computing. *Communications of the ACM*, 36(7):75–84, July 1993.

[17] T. G. Zimmerman. Personal Area Networks: Near-field intrabody communication. *IBM Systems Journal*, 35(3,4):609–617, 1996.