Keith Douglas
Statistics Canada[1]
philosopher.animal@gmail.com

Title: “Prolegomena to any Future Metaphysics of Networking”

Abstract: The hierarchical nature of both Mario Bunge’s metaphysics and the OSI model of computer networks suggests a point of contact between these two areas of human endeavour. This paper examines Bunge’s metaphysics in the light of how it applies to computer networking and three areas of apparent conflict are discussed.

---

[1] Institutional affiliation is for identification purposes only. My work in this paper is not a product of my employment there.

Prolegomena to any Future Metaphysics of Networking

Introduction

Understanding computer networks is an increasingly important practical problem from many perspectives: e.g.: security (Cross 2007), social justice (Servon 2002), and so forth. However, as a result of these practical considerations, the philosophically inclined may also come across a rich source of problems in metaphysics. The present author was struck when reflecting on the selected theme for the 2009 NA-CAP conference that both the OSI model of computer networking and Mario Bunge's metaphysics[2] (e.g., Bunge 1977) appear to involve a leveled structure of reality with different features at each level. However, Bunge's metaphysics of levels allows for 5 levels of reality which, needless to say, do not correspond at all to the OSI networking layers. In this paper I discuss the difficulties of applying Bunge's metaphysics in the area of computer networking. To this end, I first introduce briefly the OSI 7 layer model. Second, I discuss the transmission of a very simple request over an Internet style network. This will serve also to examine in a little greater detail several of the layers of the OSI model. Third, I will introduce Bunge's metaphysics of levels and examine the status of the OSI layers are in its light. I introduce three areas of difficulties with understanding the metaphysics of networks in Bungeian terms. Fourth, I will briefly summarize to close the paper.

Section 1: Introducing the OSI Model

The OSI model of networking is a seven layer hierarchy, as depicted in figure 1 below.

---

[2] This paper can be read as an exercise in using Bunge's metaphysics to analyze a novel situation. It is thus dedicated to his 90th year of life and his 50th year of work in science oriented metaphysics.

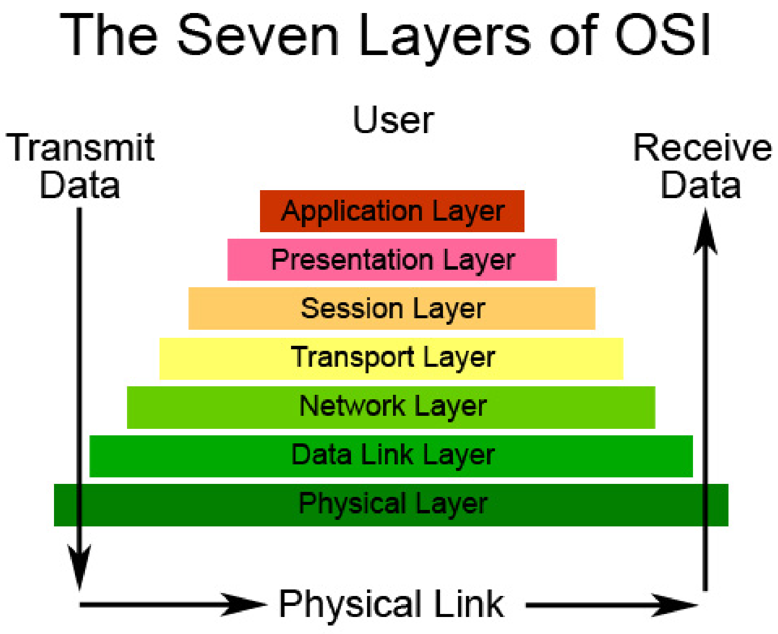

Figure 1: The OSI model (as adapted from http://catalyst.washington.edu/help/computing_fundamentals/networking/osi.html)

A brief summary of each layer follows, drawing upon Chapman, Johnson, Graziani, *et al* (2005). Starting at the top of the p age, the 7th layer is thus "Application", the closest to the user. This layer is also called the top of the hierarchy precisely for this reason. Moreover, unless you have dealt with matters like setting up a router, this layer is probably all you have ever discussed or seen; this is the layer for the "ends" of the communication. For example, HTTP, as we shall see, runs at this layer. Layer 6 (Presentation) standardizes various sorts of network wrappers, like some encryption protocols and file formats. Layer 5 (Session) can be viewed as a sort of "operator" for the network. Layer 4 (Transport, which is why TCP [Transmission Control Protocol] is at this layer) segments and reassembles data into "manageable" pieces, keeping them in an appropriate order, etc. Layer 3 (network layer, where IP [Internet Protocol] lives) provides path selection (do I go here rather than there, use this network rather than that, etc.) Layer 2 (Data link) concerns itself with physical addressing, error handling of a fairly low level, flow control, etc. Layer 1 (Physical) specifies electrical, mechanical, etc. specifications: for example, what voltage to use on a cable.

Section 2: Transmission Over Internet

In this section I discuss briefly how TCP/IP (“Internet”) style networking works to introduce some ideas important in what follows. To fix our ideas, examine figure 2. This is a networking setup which is about as simple as it can be (many other complications could be introduced) and yet illustrate all the points at issue. Assume (for the time being) that all the connections in the diagram are “category 5” (Cat5) Ethernet or similar cabling and that the switch and router are each just that[3]. I make the simplifying assumptions also that the hosts have only one ethernet port, the entire network is an TCP/IP (with IPv4 only) network, that routing is static, that IP addresses are also static and bound to one interface only and that DNS is not involved. The task in this section is thus to trace through the OSI model to answer the question: “What happens when Host 1 enters into its web browser “http://192.168.0.2/page.html[4]?” A more detailed answer to the question can be found in a typical introductory textbook of networking principles

---

[3] The reader who is familiar with home-style switches and routers might be confused by the fact that many of these low-end devices combine routers and switches into one box. Adding additional layer 2 switches to the mix do not complicate the situation too much.

[4] The simplest case is not the case with “no page”, i.e. http://192.168.0.2/, because the page fetched then is decided directly by the web server configuration - usually an index.html page, an error condition (not found, permission denied, etc.) or the like.

(e.g., Chapman, Johnson, Graziani, *et al* 2005). (Note: In this diagram “incoming” is from the perspective of the initial request from Host 1.)

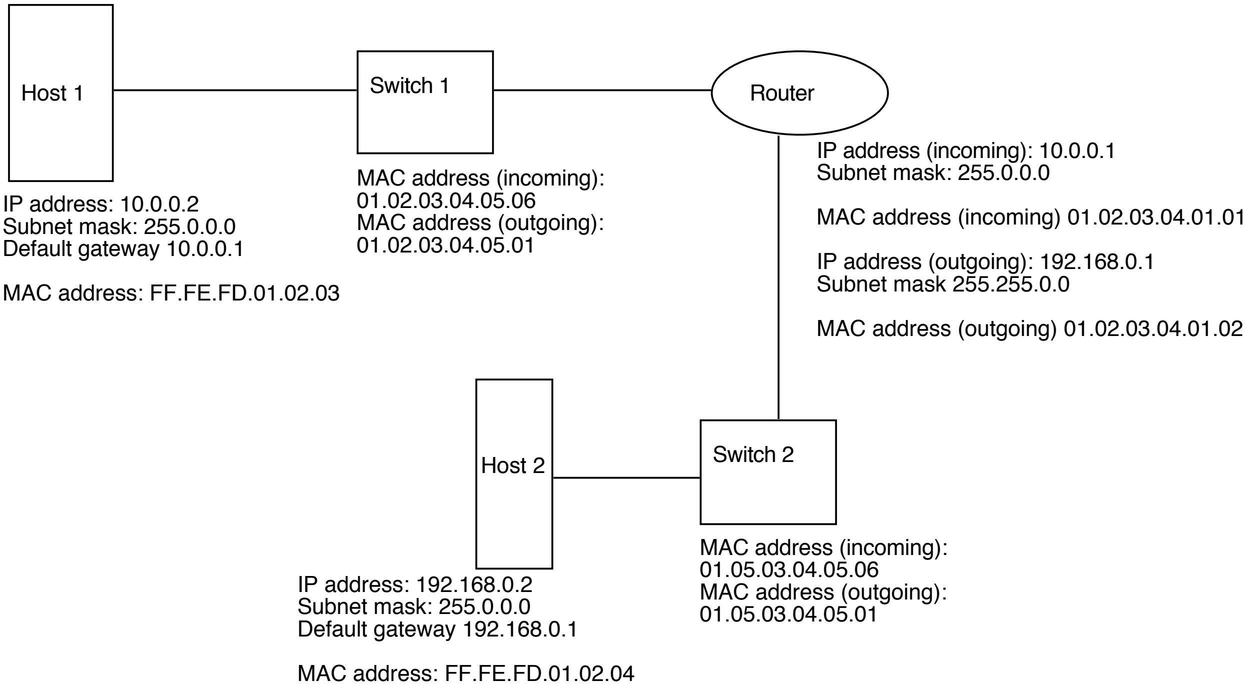

Figure 2: A simple TCP/IP style network

The user entering this request interacts (clicks on a destination bar, types, etc.) with an application (like Safari, Firefox or Internet Explorer) to start the “process” off. First, the web browser converts the request into a HTTP GET request so that a data structure that looks something like the text in figure 3 is formed. This is layer 7: the details do not matter so much; the important feature to realize the user entered string is transformed into another[5].

---

[5] It has been suggested, semi-tongue in cheek, from time to time, that the user interaction is an “eighth layer” in the model. This suggestion has merit for reasons which I cannot discuss in this paper due to space and time constraints.

```
GET /page.html HTTP/1.1[CRLF]
Host: 192.168.0.2[CRLF]
Connection: close[CRLF]
User-Agent: Safari[CRLF]
Accept-Encoding: gzip[CRLF]
Accept-Charset: ISO-8859-1,UTF-8;q=0.7,*;q=0.7[CRLF]
Cache-Control: no[CRLF]
Accept-Language: de,en;q=0.7,en-us;q=0.3[CRLF]
Referer: http://10.0.0.2/[CRLF]
[CRLF]
```

Figure 3: HTTP GET request structure. `[CRLF]` indicates two nonprintable control characters.

I will ignore layer 5 and 6; they are unimportant for our discussion (and would simply raise the same issues as I will come to discuss anyway). Layer 4 is where things get interesting again. This is the first layer that deals directly with what might be called data-transport issues (as opposed to application issues - this in spite of the name of layer 7). At this layer, the data is segmented - one can only transmit so much at a time. The example above would probably be only in one segment, so the literal (usual) sense of segmentation would not apply. However, again, (meta)data is added. TCP (in this case, an operating system feature) prepends 20 bytes of header onto the data from the upper layers. These include source and destination port, a sequence number (so that the messages passed can be placed in the original order if they arrive out of order: think of it like sending a letter one page per envelope; you'd better number the pages if your recipient is to read the letter!), a header length field, reserved bits, code bits, the window (size), checksum, a so-called "urgent" field, and an options field. Some of these fields are less than one byte long, which is why so many additional data fields can be stored in the header. These allow negotiation of speed, flow control, reliability checking, etc. Note however there's no actual network source or destination available yet. This is the job of the next two layers. Layer 3 (also in this case an OS feature), IP handles what is called logical addressing. This are the addresses of the form originally put in by the user (192.168.0.2, in our case). Our segment is encapsulated into a packet. Another header is prepended, most importantly for our purposes consisting of source and destination IP addresses. However, it is important to realize that these are only 8 bytes out of the entire header. 12 additional bytes for fixed fields, plus variable length options and padding (to ensure that the header is always a multiple of 32 bits in size) area are also always present. Once this is done the destination address is looked at to determine how the packet should be sent. In our case, the host has only one network interface, so the decision is simple. The layer 2 device to receive the packet is thus the ethernet interface. Here the packet has data prepended and appended to form a frame. These will include MAC (media access control: i.e, the so-called physical) address information about the layer 2 device to send to. This will either be switch 1 if the information is available from earlier communication, or a broadcast address, if it is not. At this point networking textbooks talk as if the frame is then converted to a signal and put on the cable in the appropriate physical fashion. Of course, only the second is true; what really happens is that the internal signals are converted to the appropriate electrical (in this case, since we're talking cat5 cabling) signals for Ethernet. At this point, matters become "purely" a matter of electrical engineering (and then, if the cable engineers were up to par, physics) until the signal starts to arrive at the switch, some milliseconds later.

Switches, at least simple layer 2 ones (like low end consumer market ones), are hardware devices with no software control. The patterns of incoming signals determine which outgoing port to send a frame onto. Note that a new frame is created as a new destination is put in place. This destination is dependent on whether the *switching table*[6] contains the destination address or not. Like in the case of host 1, it can either have seen the destination or not and will either forward appropriately or broadcast the frame to all its outgoing ports. (How it maintains the switching table is not important to us.)

When this frame arrives at the ethernet interface of the router, it recognizes (like the switch) that the frame is for it, and “removes” the frame header and footer and performs the necessary tests to ensure that it is “in good shape”. The packet is left, which is passed up to the layer 3 aspect of the router. This is normally regarded as being a software layer, as it has configuration options (even on consumer quality routers). More expensive routers (e.g., by Cisco) even have a scripting language to set up how routing is to be performed, etc. The routing software examines the packet (again checking for well-formedness[7]) and looks at the header to determine its destination. A routing table is

---

[6] It is this table and its supported and supporting features that distinguishes a switch from a hub. A hub always broadcasts.

[7] Philosophers who are not in agreement with John Searle (e.g., 1982 [1981]) on syntax/semantics as it pertains to computing systems or philosophically unaware computing professionals might well call this a “semantics” check as it has absolutely nothing to do with making sure electrical timing is appropriate, etc and instead makes sure that a length field corresponds to the actual length of a bit string, etc.

invoked. This can be as simple as one network destination for all traffic or arbitrarily[8] many. In our case, the example works with as little as two entries in the table, one for each network (10.0.0.0/8 and 192.168.0.0/16) the router is connected to. This is where the router gets its name: it has (stored within it) the route between these two networks. Subsequently a new packet is assembled and then a new frame and so on. (The router behaves from this time on much like Host 1.) The signal goes out, and the situation repeats itself. When a frame arrives at Host 2, it is “unwrapped” and the data is passed up layer by layer, eventually as a data structure to a web server program (we hope!) like Apache or IIS. Then, the server tries to fulfill the request (or send a 404 [not found] page, etc.) and the same sort of scenario we have seen plays itself out “in reverse” (Host 2 “sends” the output down the OSI layers, a frame gets send out as a series of electrical signals, etc.).

Section 3: Bunge’s Metaphysics of Levels and Comments on Networking

In this section, I introduce the important features of Bunge’s (1977) metaphysics of levels. As starting point, he suggests that the universe is ontologically monistic in the sense that there is only one stuff (matter) but pluralistic in the sense that assemblies of matter (a) exist and (b) have properties (here as types, rather than tokens, or any resultant property would count) their components lack. These sorts of properties are

---

[8] I do not mean “arbitrary” in the logician’s sense. Cisco’s IOS or at least the memory size or processing capacity of the router will practically limit the size of the routing table, but in principle there’s no intrinsic limitation to the size of routing tables.

called emergent[9]. Since for Bunge a law is an objective relation between properties (or their changes) of systems, the way to then *discover* emergent properties is to look at the laws of the systems in question. Now that we've seen the gist of Bunge's metaphysics in brief outline, let's look at some problems with interpreting the layers of the OSI model as layers of reality somewhere in the hierarchy. I raise families of three such concerns (there are no doubt others). I shall introduce additional features of Bunge's metaphysics as we go along to give further substance to the discussion.

The first of my objections concerns emergent properties. Since these are known through their reconstruction in linguistic and logical terms, i.e. as law statements and systems of same, we must examine any theories of networking to see whether or not these contain reference[10] to properties which the theories would hold to be emergent. "Simply" look at what properties components lack and the things they compose have. For Bunge a theory (see Bunge 1983; cf. van Dalen 2000) is a set of propositions closed under an entailment relation. This cannot be done as easily as it can in (say) physics, where there are full fledged theories; there are (as far as I know) no such general theories of networks. Textbooks of networking involve principles: sentences of

---

[9] Note that this is a purely ontological characterization, unlike the Broadian notions (rightly, in Bunge's view as well) criticized by Kim (1995 [1989]) and others as a species of obscurantism. Bunge has dedicated much of a more recent work (2003) to discussing how we come to know emergent properties and how they are formed (e.g. how shape arises out of shapelessness).

[10] I am going to ignore Bunge's (1974a, 1974b) theory of reference (and semantics generally) in what follows and assume we can "pick out" the properties referenced by predicates in any appropriate statement/proposition (and, *a fortiori*, in any theory).

the form “set this configuration if you want this outcome” which are not (straightforwardly) propositions, or isolated statements taken from the basic sciences involved (usually physics), or sometimes, engineering. For example, in Tanenbaum’s classic textbook (1996) we have a discussion of the Nyquist sampling theorem and its associated premisses and conclusions (i.e., a theory of sampling is presented) but no interconnection between this physical layer issue and higher layers. A substantial investigation of why there are no theories would take me too far afield and into more epistemology than I want to address, but my guess is that it has to do with the **technological** nature of the discussion, so **rules** rather than laws are important.

The next area of potential metaphysical conflict between the OSI model and Bunge’s model of reality concerns the nature of the two hierarchies themselves and the nature of the composition relations “binding” their layers together. Bunge postulates (at least in his 1977) that the sciences show that the world exhibits the partial hierarchy shown.

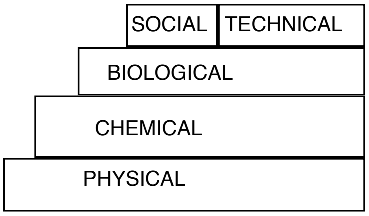

Figure 4: Bunge’s Levels of Reality

Here we run into an immediate problem. This objection centers around his theory of things.

```
“*** Signoff: Tyburn (Quit: It's only a stream of
electrons. I know where I can get more. )”
```

This sign-off message from IRC (taken from dalnet #macintosh on 2008-08-24) came up fortuitously as I was writing an early draft of this paper. It illustrates (largely in jest) the idea of placing computer networks into Bunge's layer 1 as it views computer networking purely in terms of what might be viewed as its most basic constituents. On the other hand, his technical level would seem appropriate since they are artefacts built on purpose by humans. But since his levels are supposed to be levels of composition (each sort of *thing* [see his 1977, pp. 110-162 for his theory of things], is composed of things at the lower level) this cannot be quite right. In this case, "composed by" is instead true: biological systems are responsible for the construction of both items above them but artefacts do not at the very least have gregarious animals as their *only* component in the same way a social system (e.g. a card game club) does. But unlike societies, which would be annihilated if the animals which compose them were all to perish, their artefacts would at least for a time being continue to operate the same as before. If, as planned, Internet survived a nuclear war, for example, signals (at least in the electrical engineering sense) would still pass through the network at various layers etc. until it all fell apart due to lack of maintenance. It seems at least *prima facie* plausible to suggest that this would still involve any emergent properties as well. Hence Internet would still be a computer network for a while.

But several problems can be raised about this idea. if we think about network "objects" as being composed out of bit sequences (however understood), then lower layers are the ones with the emergent properties, not the reverse (at least at first glance). Why? Because a frame has more bits than a packet.

The first assumption required to support the paradoxical (but prima facie true) claim that items at the lower levels of the networking hierarchy emerge out of the higher ones is the also very intuitive idea that packets, frames, etc. are composed out of "zeros and ones" as the saying goes. This notion requires sharpening in order to get a precise hold on the usual claims in texts and networking discussions. I introduce some stipulations to that end.

Let us call three states[11] of a concrete system 0, 1 and ε. Then a sequence (temporally or spatially juxtaposed, represented by an association operator $\oplus$) of such states is called a *bit string* when and only when the following laws apply:

BS1) $x \oplus (y \oplus z) = (x \oplus y) \oplus z$

BS2) $x \oplus \varepsilon = \varepsilon \oplus x = x$

BS3) $x \oplus y \neq y \oplus x$

BS4) $0 \neq 1 \neq \varepsilon$

(Bit strings thus form a semigroup on the sequence of states.) When (and only when) the sequence of states is arrayed spatially (i.e. we can think of the association operation as spatial juxtaposition), we call the items in which the sequence takes place a *register*.

---

[11] It is rather crucial that one does not interpret these strings as forming numerals (see Clements 1994). I only adopt the labels 0 and 1 to abide by the usual misleading conventional notation.

Similarly, a *stream* is a bit string where the association operator is temporal. A *register of size n* for each n is (intuitively) a register with a fixed bound of size n. These obey the following laws:

RGN1) $x \oplus (y \oplus 1) = (x \oplus y) \oplus 1$ iff length (x) + length (y) < n, undefined otherwise.

RGN2) $x \oplus (y \oplus 0) = (x \oplus y) \oplus 0$ iff length (x) + length (y) < n, undefined otherwise.

RGN3) $x \oplus \varepsilon = \varepsilon \oplus x = x$

RGN4) $x \oplus y \neq y \oplus x$

RGN5) $0 \neq 1 \neq \varepsilon$

Now let us apply these definitions to a concrete case: the various "network objects" such as packets and frames are thus bit strings that we can feign exist at once. Or so the usual understanding is: they are actually arrayed in: first a 1, then a 0, then a 1 etc. (for example) arrive at a register in some networking hardware (e.g. an ethernet interface). Since real materials do not allow infinite registers, the network object has to fit in a register of a fixed size n for some n. This is also a reason for thinking of the network objects in the first place: otherwise one has simply a sequence of state transitions of arbitrary length (taken at some fixed location: another problem). Another factual assumption that is crucially underlying the notions introduced is that one can distinguish three states of the system. But these states are characterized functionally, by the laws I have stated, not in material terms (and, *a fortiori* in the "language" of any specific science). They are indeed multiply realizable (to use the jargon from philosophy of mind): not only can the same bit stream be propagated or stored in electronic form, but also in photonic media (e.g. fibre optics). Indeed, it can even be encoded in different manners which are then "understood" by different devices. For example, in so-called "non-return to zero" two voltage levels are the physical states of the device. However, the "functional" states of the device are whether or not the previous relevant interval included a high or low voltage in addition to the current physical state. But this can be replaced by "Manchester" encoding with a different way of mapping physical states on to the two states 0, 1. In some of these there is a further state which corresponds to no signal at all; in others a sufficiently long duration of one of the voltages is regarded “semantically” as the state ε. That is, some bit string 0000000000...0 (or 111111111...1) actually corresponds to the “empty” signal, one which does not ever get interpreted as a frame or packet, etc. Note also that packets, etc. are either “contents” of registers or streams, depending on their stage in the “process”. Of more on this later.

On the other hand, the claim that the networking hierarchy is actually inverted with respect to emergence does not hold at layer 1 (that is, frames do not compose the objects at layer one so the hierarchy is not merely inverted, instead at first blush it looks like it could be written top to bottom as 1-7-6-5-4-3-2). Here, in our “theories” of networks there are no networking objects at all, merely voltage specifications, photonic equivalents, etc. Or, alternatively, a device taken purely as a level 1 device, appears to be purely a physical thing and one with no interesting emergent properties not found in physics (unlike, say, a sufficiently large quantity of water, which gains viscosity, etc.; it has some of these like, say, hardness and shape). The fabled ones and zeroes are in fact not *things* at all: they aren't composed in the sense of mereology out of more basic things also at layer one. They appear to be rather a member from the classes of properties referenced in the theoretical reconstruction (or rather, would map perfectly if our theories were completely true)[12]. That is, 0 is such and such a voltage pattern, 1 another, etc, or a light amplitude pattern, etc., etc. Summarizing this detailed section, then, there are two related difficulties: one is that bit strings are not apparently composed out of *things* and one is if they are composed, then items towards the bottom of the hierarchy are the ones which get built up of ones from higher up!

Our third and final area of difficulty with Bunge's metaphysics and networks concerns another pair of premisses we need in order to draw the paradoxical conclusion about the nature of networking objects discussed above. This postulate is simply that the packets and so on themselves are real and not convenient fictions or theoretical intermediaries. (Some readers may have noticed this in the previous difficulty as well.) This involves two considerations. One is simple Bunge-style scientific realism; another is a bit more subtle. Even if one takes a less than strongly realist attitude towards the packets, etc. one has to account for their persistence conditions, at least at first glance, which do seem strange, even relational. Let's look at these. In our example, the frame is assembled in the ethernet card of a workstation computer. This means that at some point there is a register of some fixed size containing an appropriate sized bit string.

---

[12] One other possibility is that the 0 and 1 are actually events, whence networking objects are processes or extended events. Bunge lacks a mereology of events, so this interesting idea is a nonstarter as far as the current project is concerned. However, see Douglas (2001) and Simons (1987) for some thoughts in this area.

However, this “thing”, if indeed we can speak that way, is then sent onto the cable. What does this involve? In our example, the transmission is serial, as it is the case in most networking physical layers. So, one bit is transmitted at a time, subsequently the frame does not exist “on” the cable until a certain delay has elapsed. Similarly, the frame will start to arrive at the destination interface piecemeal. If a frame is a spatially oriented bit string of a certain length, then there is sometimes never any frame on the cable at all: only parts of it, taken intuitively. We seem to be required to say that the interaction is what matters and that existence of packets, etc. is merely relational, or perhaps also a functional kind. On the other hand, we could say that the bit string itself is the thing, and it undergoes some sort of change. This seems to be a particularly acute case of the question of persistence through time. Bunge’s view (1977) is that strictly speaking there is no such thing, only persistence in some respect or other. (That everything changes does not entail that everything changes in the same respect to the same degree or at the same relative rate: this is in fact what makes clocks possible at all.) This has difficulties too, as what persists in these cases is not obvious. Moreover, if a piece of networking hardware sends out a layer n object it seems to presuppose that it is going to arrive at a suitable area for decoding: if there is no reception the object ceases to exist or perhaps never existed at all. A suggestion to get out of this difficulty would be that a frame is a frame due to its origin (cf. Kripke 1980 [1972]), but this requires further investigation. Bunge does not discuss essentiality of origin in any great detail, so I leave that for another time as well.

Section 4: Conclusion

I have discussed Bunge's metaphysics in the light of computer networking and raised some questions that a metaphysics sensitive to the nature of computational artefacts should try to answer. We have seen how networking objects do not fit clearly into any metaphysical category, how understanding relations of emergence in networking is difficult, how a mild realism runs into difficulties and how understanding identity conditions for networking objects is difficult. At some other time I will hopefully propose some answers.

Bibliography

Bunge, Mario. 1974a. *Sense and Reference*. (Volume 1 of *Treatise on Basic Philosophy*). Dortrecht: Reidel.

Bunge, Mario. 1974b. *Interpretation and Truth*. (Volume 2 of *Treatise on Basic Philosophy*). Dortrecht: Reidel.

Bunge, Mario. 1977. *The Furniture of the World.* (Volume 3 of *Treatise on Basic Philosophy*). Dortrecht: Reidel.

Bunge, Mario. 1983. *Exploring the World.* (Volume 5 of *Treatise on Basic Philosophy*). Dortrecht: Reidel.

Bunge, Mario. 2003. *Emergence and Convergence: Qualitative Novelty and the Unity of Knowledge*. Toronto: University of Toronto Press.

Chapman, Bill; Johnson, Allan; Graziani, Rick; Horn, Elaine; Large, Andrew and Rufi, Antoon. 2005. *Cisco Networking Academy Program: CCNA 1 and 2 Companion Guide*. Revised Third Edition. Indianapolis: Cisco Press.

Clements, Alan. 1994. *68000 Family Assembly Languages*. Boston: PWS Publishing Company.

Cross, Michael. 2007. *Developer's Guide to Web Application Security*. Rockland: Syngress.

Douglas, Keith. 2001. *A Special Davidsonian Theory of Events.* Unpublished MA Thesis, Department of Philosophy, University of British Columbia.

Kim, Jaegwon. 1995 (1989). "Supervenience as a Philosophical Concept". Reprinted in Kim, Jaegwon. (ed.) 1995. *Supervenience and Mind: Selected Philosophical Essays.* New York: Cambridge University Press.

Kripke, Saul. 1980 (1972). *Naming and Necessity.* Cambridge: Harvard University Press.

Searle, John. 1981 (1980). “Minds, Brains and Programs”. Reprinted in Hofstadter, Douglas and Dennett, Daniel. (eds.) 1981. *The Mind’s I: Fantasies and Reflections on Self and Soul*. New York: Bantam Books.

Servon, Lisa. 2001. *Bridging the Digital Divide: Technology, Community and Public Policy*. Malden: Blackwell.

Simons, Peter. 1987. *Parts: An Study in Ontology*. Cambridge: Oxford University Press.

Tanenbaum, Andrew. 1996. *Computer Networks* (3e). Upper Saddle River: Prentice Hall.

van Dalen, Dirk. 2000. *Logic and Structure* (3e). New York: Springer.