Physics Analogs in Communication Models

Joseph D. Touch

USC / Information Sciences Institute (touch@isi.edu)

Abstract[1]

We describe the physics analogies used in a model of latency in computer communication and protocols. The communication model describes state imprecision as induced by message latency, and is based on physical models of particle interaction. Virtual particle effects are emulated through the use of conditional messages that reduce perceived latency while requiring additional bandwidth and active sender participation.

1: Introduction

We recently developed a model of latency in communication, based on analogies from quantum particle interaction [7]. The model uses principles of state divergence and convergence of quantum particle interaction models to describe action-at-a-distance in computer communication. In our model, communication is analogous to a field, packets are analogous to field quanta, and communicating entities (end-nodes) are analogous to particles in the field. We note that as communication rates and the amount of latent interaction increase, there is a need to model the intermediate imprecision of state that accompanies latency.

2: State imprecision

Particles (or nodes in a network) interact by forming local perceptions of remote state. Existing communication protocols (TCP/IP, OSI TP-4, etc.) are based on *point* perceptions of remote state, which are single-valued. State evolution is described by the transformation of a single point in state space to another single point. When a message is received or sent, or time passes, the perception of the state of the remote node is adjusted accordingly, in each case by moving the current state point in state space (Figure 1).

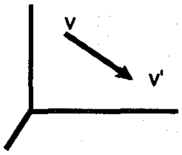

Figure 1: Point transformation model of state transition

As the latency between communicating parties increases, a point model no longer suffices to describe remote state. Perception of a distant (remote) node is best represented as a pdf (probability density function), where the actions of messages are transformations on the pdfs.

In the digital domain, these pdfs are effectively governed by the partitions within the state space, which can be represented by a set of states, denoting the possible current states that node may occupy since last heard from. Thus, we treat message reception, transmission, and time passage as separate kinds of transformations on these sets (Figure 2).

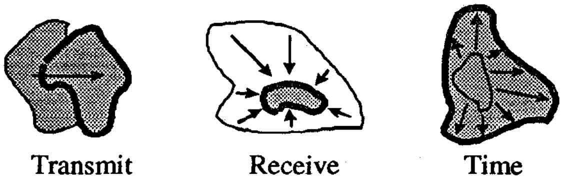

Figure 2: Volume transformation model of state transition

As time progresses, the state perception necessarily becomes less precise, increasing in entropy. When messages are received, the local perception is refined and constrained, reducing the entropy of the perception. This relationship between communication information and entropy was first noted by John von Neumann [1].

[1] This research was sponsored by the Defense Advanced Research Projects Agency through Ft. Huachuca Contract No. DABT63-91-C-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of the Army, the Defense Advanced Research Projects Agency, or the U.S. Government.

0-8186-3420-0/93 $03.00 © 1993 IEEE

The effects of time and message reception on state are well understood in protocol models. One contribution of our model is a description of the effects of message emission. Sending messages to a remote entity expands the local perception of that remote state. The combined effect of an emitted message and a subsequent received reply increases the entropy of the local perception in proportion to the number of bits sent [7].

3: Stability with state imprecision

As a system evolves over time, the set of possible states it can be in also evolves. As time progresses, this state volume (A) tends to grow larger (A') (Figure 3). Communication is an effort to prevent that expansion, and to govern the evolution of state in conjunction with time evolution (A") (Figure 3).

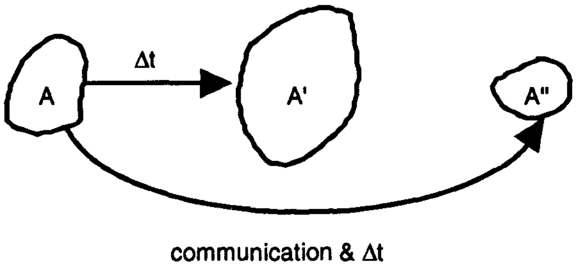

Figure 3: Evolution of state

In conventional physical systems, stability is defined as the confinement of state to a region over time (Figure 4). Because the volume of the region cannot expand over time, its entropy is also confined.

$$\underset{\Delta t > t_{min}}{\forall} A'' \subseteq A$$

Figure 4: Conventional stability as state confinement

In our model we needed to express a notion of stability that exceeded the bounds of the conventional definition. We define *entropic stability* as the *confinement of the imprecision of state over time* (Figure 5). This allows the evolution of a dynamic system to be evaluated as stable if its evolution function is deterministic.

$$\underset{\Delta t > t_{min}}{\forall} |A''| \leq |A|$$

Figure 5: *Entropic stability*

4: Sender anticipation and virtual particles

Latency induces a 'set' effect to the former 'point' model of state transitions. Message emission increases entropy by increasing the set size of the perception of remote state; this effect could not be described in the former 'point' model. High bandwidth*delay products permit increased message emission, which results in increased perception entropy. These entropy increases need to be minimized or mitigated.

Emitting a message causes the sender to update its perception of the message receiver; the new perception is the union of two perceptions - the original, denoting the state of the receiver before the message is received, and the other denoting afterwards (Figure 6). The sender thus operates under the dual assumption that a message has been received and not yet received, until it has determined otherwise via the reception of another message.

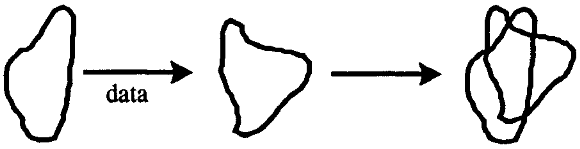

Figure 6: Message emission increases perception entropy

Using the effects of messages to determine the effects on remote state originated in 'common knowledge' [2]. In 'common knowledge', global state, global constraints, and local messages are used to infer remote state. In our model, local messages and individual remote constraints are used to infer remote state, although in a different way. We additionally require use of state sets, to model the state imprecision that latency induces.

In physical systems, these multiple paths of execution are modelled as virtual paths, also called Feynman paths. The net effect of actions on a system is the integral of these paths. We created a digital equivalent of these paths, for the purpose of evaluating a system of interrelated messages. These messages are called *guarded messages*, and correspond to the effects of virtual paths on the path integral [7].

Guarded messages mitigate the effects of message emission as entropy increase. A guarded message is a conventional message plus a prefix (guard) that denotes some subset of the receiver's state. The receiver accepts the message only if its current state is indicated by the guard. By partitioning the perception into subsets, and emitting messages to each component with suitable guards, the overall entropy of the emitted message group can be reduced, as compared to a single message (Figure 7).

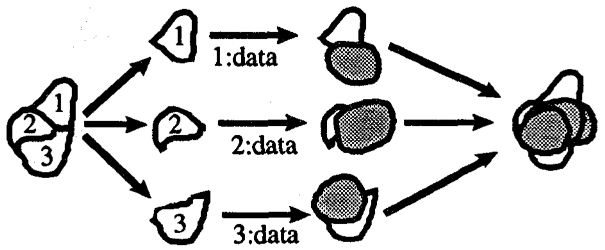

Figure 7: Guarded messages minimize entropy increases

The guards on the messages transform a single message into a group of messages with a single effect. When such a set is emitted, only one component is accepted and utilized, but the determination as to which component that will be is delayed until actual reception. This is analogous to a real particle becoming a set of virtual particles. When one virtual is realized, all others disappear (or are ignored).

5: Analogies to 'multiple futures'

Sending multiple streams of communication permits multiple remote futures to be accommodated by the source of messages in a channel. In a conventional request/response interaction, a single message is emitted by the source, and an entire round trip latency passes before the next request.

In conventional protocols, the source is prohibited from reacting nondeterministically. After the initial reply, a set of next-possible-replies can exist, but is not currently used. In our model, called Parallel Communication, the sender converts a tree of possibilities in to a sequence of messages, each predicated by the state the receiver would have been in to request it. The result is that the channel is completely filled with anticipatory messages, even though only some of these messages 'pan out' (Figure8).

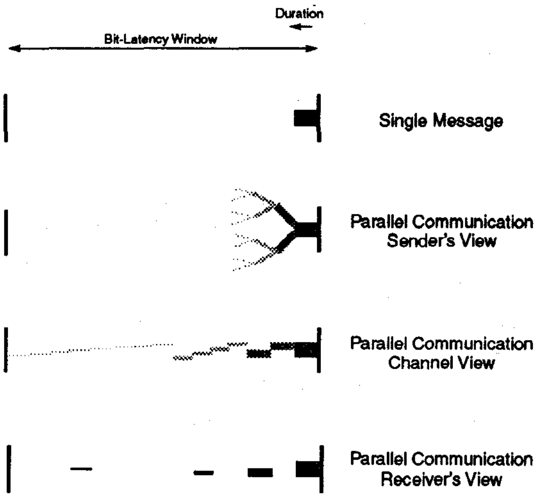

Figure 8: Parallel Communication representations

This mechanism for Parallel Communication can be implemented as an application-layer abstraction, or can also affect the utilization and management of conventional sliding-window flow control mechanisms. The result is a branching-windows mechanism, which still requires application-layer signalling (Figure 9).

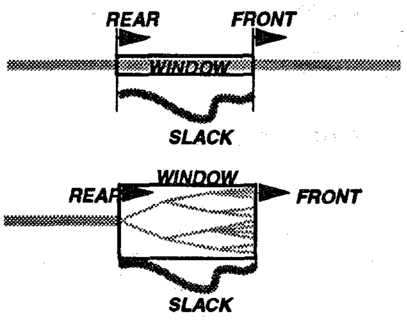

Figure 9: Sliding-windows become branching windows

6: Petri Net equivalences of 'multiple futures'

We also developed a Petri Net equivalent of this mechanism, in which the local perception of a remote participant is permitted to have sets of virtual tokens, analogous to virtual particles [7]. A virtual token becomes realized when it interacts with a real token (Figure 10).

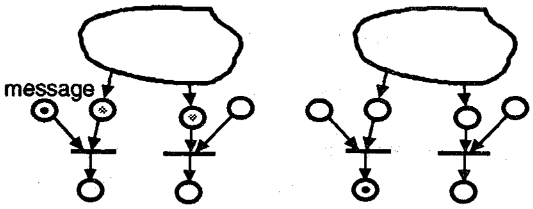

Figure 10: Virtual tokens (grey) interact with real (black) {left figure}, resulting in realization {right figure}

The equivalence is based on a transformation of basic-block-like structures. The basic blocks have single entry and exit points, and are arranged as alternates in the path of a single token (Figure 11). Each basic block can interact with remote Petri Nets by the reception of 'external' tokens, and can emit remote message tokens, but cannot accept internal tokens other than at the entry point.

The transformation involves the replication of the single, real, initial token into a pair of conjugate tokens (Figure 11). One of each of these pair proceeds down each basic block, thus permitting an overlap of the states represented by the alternates. At the exit of the basic blocks, the first token to be matched by a remote message is instantiated as a real final token (the 'hat' variables), and creates a copy token that annihilates the next token emitted from the alternate block. The structure also prohibits more than one such *virtual pair* of tokens from passing through the structure at a time.

In addition, the emitted message tokens are *guarded* by the requisite state of the system, i.e., the message exists only if the remote state is as specified. The receipt of messages indicating the remote state cause the collapse of the virtual pair, where the first token matching such a message is converted into a real token, and the other token is annihilated upon exit.

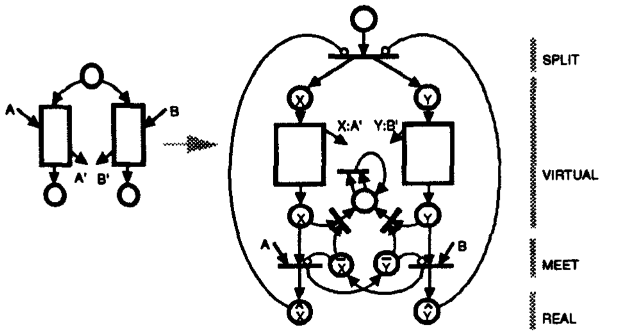

Figure 11: Petri Net transformation for virtual token pairs

The existence of a virtual pair of tokens is indicated by the structure of the transformation, and the requirement that of the basic blocks. By construction, only one of the pair ever is instantiated as a final, real token, and interaction with the pair regulates this 'collapse'.

7: A mechanism for virtual state management

From this model, we developed a mechanism to emulate the expansion of remote state into a set of possible states, and the collapses indicated by communication. We note that a model is not a mechanism, this mechanism is a granularity-equivalent realization of the model, as an example of the utility of the model.

The mechanism has two main components, a *filter* and a *pump* (Figure 12) [6]. The mechanism communicates among any time-separated request/response pair.

The *pump* manages the model of remote state as a set of messages, and implements the state expansion and collapse of Figure 2. The *converger* manages state collapse due to message reception, and the *diverger* manages state expansion due to time evolution and message emission at the sender.

The *filter* manages the reception of only those messages whose guard matches the current state of the receiver. It also manages the eventual collapse of the state, by emitting messages that more precisely specify the receiver's current state. Such messages occur after decision points in the receiver's execution of the message stream.

This mechanism can be used in any request/response interaction across a latent channel. It was first developed a model [7], and has been applied to CPU-memory opcode retrieval [6], and client/server remote file systems [8].

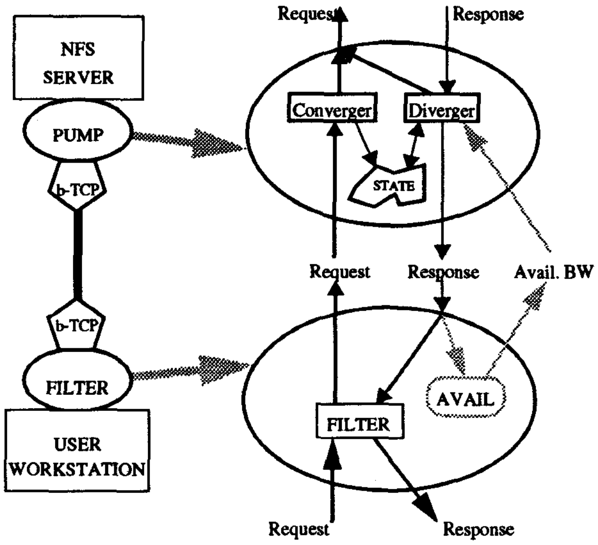

Figure 12: The *filter* and *pump* mechanisms

8: A reverse analogy seeking equivalence

During the development of the mechanism of Figure 9 for CPU-memory opcode retrieval, we noted the need for an extension to the conventional 'multiple worlds' metaphor.

'Multiple worlds' is equivalent to branching of the state space structure, denoting parallel equivalent states. When a single state becomes realized, the other states corresponding to the same point on the timeline disappear, or collapse into the single real state.

We observed the need for recursive state expansion. State sequences not only bifurcate, but they recurse. This recursion may be the equivalent of some quantum process that we are unaware of.

In particular, consider a state sequence that goes into substructures then splits a few times (Figure 13).

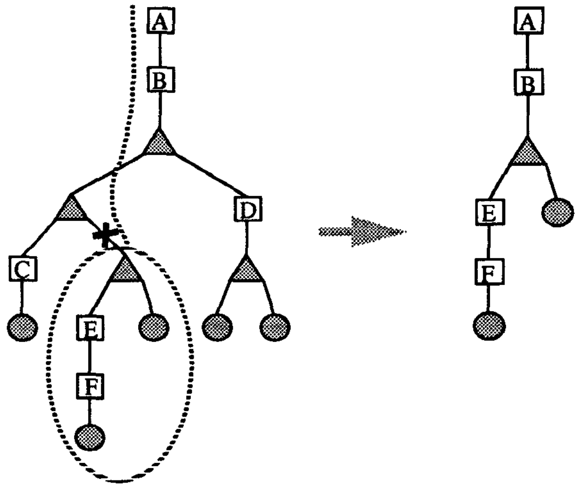

Figure 13: Substructure tree, and its 'collapse'

Such a structure would collapse in a special way that preserves all past substructuring (i.e., recursion calls), but otherwise collapses equivalently to 'multiple worlds'. We developed a data structure called a TreeStack to accommodate such structures efficiently (Figure 14) [6].

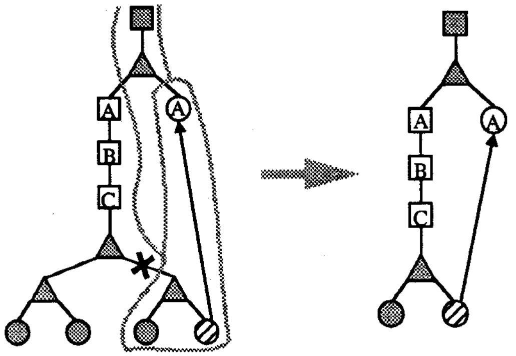

Figure 14: TreeStack 'collapse' implementation

9: Other analogies

Although there are additional analogies between Newtonian physics and telecommunications [5], there are also others involving quantum interactions which have not yet been exploited.

There is an analogy between uncertainty and bit-latency (bandwidth*delay product) in communication networks. Uncertainty is characterized by units of action, i.e., energy *time or work*time. In communication, this is just bits*time. Actions share a property with bit-latency, that of observer invariance [3]. Bit-latency describes a minimum insurmountable information separation between two remote entities.

10: Conclusions

We have developed a model of latency in communication based on analogies from physics, notably from quantum interactions. These analogies indicated characteristics of latency-limited communication, and mechanisms to circumnavigate such limitations.

The application of physics to solve communications problems is well understood; here we instead apply the conceptual analogies to solve abstract problems, rather than to address implementation issues. The notion that communication is linked to physical interaction is extrapolated to apply to communicative interaction. We believe these analogies have further potential.

11: References

[1] Hartley, R.V.L., "Transmission of Information," *Bell System Technical Journal*, V. 7, pp.535-563, (1928).

[2] Halpern, Joseph Y. and Moses, Yoram, "Knowledge and Common Knowledge in a Distributed Environment," *Symposium on Principles of Distributed Computing, ACM SIGOPS-SIGACT*, ACM Press, (Aug. 1984).

[3] Gribbin, John, *In Search of Schrödinger's Cat*, Bantam Books, Toronto, (1984).

[4] Tannenbaum, Andrew S., *Computer Networks*, Prentice-Hall, NJ, (1988).

[5] Tokoro, Mario, *Computational Field Model: Toward a New Computing Model / Methodology for Open Distributed Environment*, Tech. Report. SCSL-TR-90-006, Sony Computer Science Laboratory, Inc., Tokyo, Japan, (June 1990).

[6] Touch, Joseph D., and Farber, David J., *An Active Instruction Decoding Processor-Memory Interface*, patent applied for, Univ. of Pennsylvania, (Sept. 1991).

[7] Touch, Joseph D., *Mirage: A Model for Latency in Communication*, Ph.D. dissertation, Dept. of Computer and Information Science,. Univ. of Pennsylvania, (1992). Also available as Dept. of CIS Tech. Report MS-CIS-92-42 / DSL-11.

[8] Touch, Joseph D., "Parallel Communication," submitted to *IEEE Infocom '93*.