Journal of Experimental Psychology:
Human Perception and Performance
2003, Vol. 29, No. 1, 3–18

Copyright 2003 by the American Psychological Association, Inc.
0096-1523/03/$12.00 DOI: 10.1037/0096-1523.29.1.3

# A Central Capacity Sharing Model of Dual-Task Performance

Michael Tombu and Pierre Jolicœur
University of Waterloo

The authors present the central capacity sharing (CCS) model and derive equations describing its behaviors to explain results from dual-task situations. The predictions of the CCS model are contrasted with those of the central bottleneck model. The CCS model predicts all of the hallmark effects of the psychological refractory period (PRP) paradigm: −1 slope of the PRP effect at short stimulus onset asynchronies (SOAs), underadditivity of precentral Task 2 manipulations, additivity of central or postcentral Task 2 manipulations with SOA, and carry forward to Task 2 of Task 1 precentral or central manipulations at short SOAs. The CCS model also predicts that Task 1 response times increase with decreasing SOA. The model is a viable alternative to the central bottleneck model.

When people are required to perform two speeded tasks in rapid succession, it is generally found that response times (RTs) on the second task become increasingly long as the stimulus onset asynchrony (SOA—the temporal gap between the presentation of the two stimuli) decreases (Pashler, 1994a). In contrast, Task 1 performance is usually not as strongly affected by SOA. The observed slowing of Task 2 responses with decreasing SOA is known as the psychological refractory period, or PRP, effect (Telford, 1931). Another signature of the PRP effect is that the slope of the Task 2 RT by SOA function often approaches −1 at sufficiently short SOAs (Pashler, 1994a).

Several theories have been proposed as possible accounts of the PRP effect. These theories can generally be placed into three broad categories: structural bottleneck theories (Pashler, 1994a; Welford, 1952), strategic bottleneck theories (Meyer & Kieras, 1997a, 1997b), and capacity sharing theories (e.g., Kahneman, 1973; McLeod, 1977; Navon & Gopher, 1979; Navon & Miller, 2002). In addition, Logan and Gordon (2001) recently proposed a cascaded model that includes elements of both structural and capacity sharing theories.

## Bottleneck Accounts of the PRP Effect

Bottleneck theories propose that some processing needed to perform each task requires access to one or more processors that can only act on one input at a time (Pashler, 1994a). If both tasks require one of these processors simultaneously, then only one can get access to it. While this processor is busy with one task, processing for the other task must be suspended until the processor is free. We refer to processors that can only operate on one task at a time as bottleneck processors or bottleneck stages.

In the general bottleneck model, it is hypothesized that certain stages are bottleneck stages. The specific bottleneck model that we are interested in postulates that bottleneck stages are responsible for response selection and decision making (McCann & Johnston, 1992; Welford, 1952). It is further hypothesized that early processing responsible for stimulus identification and late processing responsible for response execution can act on several stimuli simultaneously and can proceed simultaneously with bottleneck processing. In other words, stimulus identification and response execution can operate in parallel, whereas processors at or around response selection must operate on stimuli serially.

Figure 1 shows how a bottleneck model accounts for the PRP effect. At long SOAs (Figure 1B), Task 1 has finished bottleneck processing by the time Task 2 requires it, so Task 2 processing is not postponed by Task 1. However, at short SOAs (Figure 1A), Task 1 still occupies the bottleneck processor when Task 2 is ready to use it. As a result, Task 2 bottleneck processing is postponed until Task 1 releases the bottleneck. As SOA is decreased further, Task 2 must wait longer for central (bottleneck) stages of Task 1 to finish, which causes RTs for the second task to increase with decreasing SOA. At a sufficiently short SOA, a further decrease in SOA of 10 ms results in a 10-ms increase in Task 2 RTs. Therefore, bottleneck models predict the observed −1 slope of the Task 2 RT by SOA function. Because Task 1 always gets access to the bottleneck processor as soon as it requires it, manipulations of SOA are predicted to have no effect on Task 1 RTs.

## Adaptive Executive Control Model (Meyer & Kieras, 1997a, 1997b)

In their adaptive executive control model (Meyer & Kieras, 1997a, 1997b), Meyer and Kieras postulated that the PRP effect is due to bottlenecks in information processing but that these bottlenecks are peripheral or strategic in nature. In principle, under the right set of conditions, Meyer and Kieras and colleagues claimed that it should be possible for two tasks to have "virtually perfect time sharing" (Schumacher et al., 2001, p. 102). In most PRP

Michael Tombu and Pierre Jolicœur, Department of Psychology, University of Waterloo, Waterloo, Ontario, Canada.

During the writing of this article, Michael Tombu was supported by a doctoral fellowship from the Natural Sciences and Engineering Research Council of Canada (NSERC), and Pierre Jolicœur was funded by Research Grant 795-2000 from the NSERC.

Correspondence concerning this article should be addressed to Michael Tombu, Department of Psychology, University of Waterloo, 200 University Avenue West, Waterloo, Ontario N2L 3G1, Canada, or to Pierre Jolicœur, who is now at Départment de Psychologie, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Quebec H3C 3J7, Canada. E-mail: mt2tombu@cgl.uwaterloo.ca, pjolicoe@cgl.uwaterloo.ca, or pierre.jolicoeur@umontreal.ca

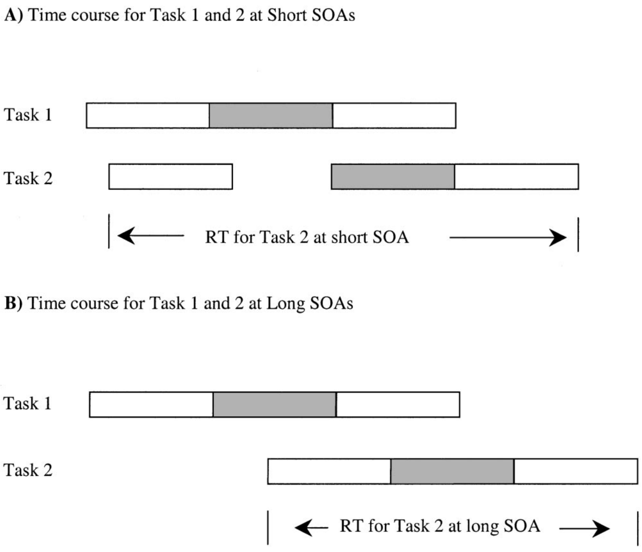
A) Time course for Task 1 and 2 at Short SOAs
Task 1
Task 2
RT for Task 2 at short SOA
B) Time course for Task 1 and 2 at Long SOAs
Task 1
Task 2
RT for Task 2 at long SOA

*Figure 1.* A bottleneck account of Task 2 slowing with decreasing stimulus onset asynchrony (SOA) in the psychological refractory period paradigm. Shaded regions represent bottleneck processing, whereas unshaded regions represent stages that can process information from different tasks in parallel. (A) At short SOAs, Task 1 bottleneck processing is not complete when Task 2 requires it. This causes Task 2 to be postponed until Task 1 completes bottleneck processing. (B) This postponement does not occur at long SOAs, because Task 1 has completed bottleneck processing by the time Task 2 requires it. As a result, Task 2 response times (RTs) are faster at long SOAs than at short ones. A further decrease of 10 ms in the short SOA condition would result in Task 2 having to spend an additional 10 ms waiting for access to bottleneck processing, hence the experimentally observed −1 slope of the RT2 by SOA function at short SOAs.

studies however, it is claimed that one or more of these conditions is violated, resulting in either the implementation of a strategic bottleneck to control response order or a peripheral bottleneck (both tasks require the same input or output processors). In the Meyer and Kieras framework, strategic bottlenecks can be invoked at any point in the information-processing stream, so the locus of a bottleneck can sometimes be early in processing, whereas at other times it may be late. This freedom allows this model to account for a wide range of findings. However, this model is still in effect a bottleneck model. Under circumstances in which bottlenecks are predicted to be encountered, this model often makes the same core predictions as nonstrategic (i.e., structural) bottleneck models.

## Executive Control of the Theory of Visual Attention (Logan & Gordon, 2001)

Logan and Gordon (2001) proposed the executive control of the theory of visual attention, or ECTVA theory, to account for a wide range of findings including those from the PRP literature. ECTVA takes Bundesen's (1990) theory of visual attention (TVA), which determines how a stimulus in the environment is categorized, and combines it with a response selection mechanism based on the exemplar-based random walk (EBRW) model of Nosofsky and Palmeri (1997) and an executive that controls TVA and EBRW. The executive sets the parameters of TVA. There are six parameters in ECTVA. To perform categorizations for Task 1 and Task 2, TVA is run twice. During the first run, Task 1 categorizations are given high priority and Task 2 categorizations are given low priority. This configuration is reversed the second time TVA is run to give Task 2 categorizations priority. In this sense ECTVA is a serial model. However, when a task is given low priority, it still has the ability to affect RTs, and this feature gives ECTVA the power to explain crosstalk (Task 2 stimulus or response characteristics influencing Task 1 RTs and accuracy) that is sometimes observed in the dual-task literature (Hommel, 1998; Logan & Gordon, 2001; Logan & Schulkind, 2000). In this sense ECTVA is a capacity sharing model. The ability of the low-priority task to influence processing of the high-priority task at sufficiently short SOAs (by introducing additional possible categorizations), but not at long SOAs, makes this model different from traditional bottleneck models and more like a capacity sharing model. However, the work done on Task 2 when it has low priority (i.e., the first time TVA is run) is for the most part lost when TVA is run the second time. This is because before TVA is run the second time, the counters in EBRW are reset to a (small) percentage of their value. As a result, most of the work done on Task 2 during the first run of TVA is wiped out.

Given the dual nature of ECTVA, the methods introduced in the present article may not be sufficient to distinguish between ECTVA and the central capacity sharing model we describe in this article. However, the central capacity sharing model is a viable alternative to the ECTVA model. In ECTVA, work done on Task 2 when Task 1 has priority is for the most part lost when the counters in EBRW are reset. However, in the central capacity sharing model we describe in this article, work done on Task 2 when Tasks 1 and 2 share capacity is not lost.

## Capacity Sharing Accounts of the PRP Effect

A second class of models proposes that processing mediating performance in Task 1 and Task 2 occurs in parallel but that there is a limited amount of processing capacity (Kahneman, 1973; McLeod, 1977). Navon and Miller (2002) recently presented a model with the same core assumptions as those discussed here.[1] Because resources are limited, Task 1 and Task 2 must share the available processing capacity. As the SOA between the Task 1 and Task 2 stimuli is decreased, there is an increase in the duration of a period of processing during which capacity is shared. This capacity limitation may not apply to all processing required to perform the tasks but may instead be restricted to certain central operations (Kahneman, 1973; Posner & Boies, 1971). It is often assumed that capacity can be allocated voluntarily (McLeod, 1977), although characteristics of the tasks may also affect the allocation policy. Determining exactly which task characteristics affect capacity allocation is an empirical issue that will need to be resolved at a later time (although in a later section we consider some possibilities). In many early PRP studies, the instructions given to participants stressed the importance of Task 1. As a result, full (or very close to full) capacity could have been allocated to Task 1, in which case capacity models mimic bottleneck models.

[1] The present article was developed independently of that of Navon and Miller (2002).

In a subsequent section of the article, we develop the equations for the central capacity sharing model at long and short SOAs for both RT1 and RT2, and we show that the central capacity sharing model can mimic the predictions of bottleneck models.

## Multiple Resource Models (Navon & Gopher, 1979)

Navon and Gopher (1979) extended the general capacity sharing model by postulating that there are multiple types of resources that a task may require. Tasks will interfere with each other to the extent that they require common limited resources. In principle, if two tasks do not require common limited resources, it should be possible to perform them concurrently, with no dual-task interference. In theory, this may explain why it has been possible to demonstrate "virtually perfect time sharing" (Schumacher et al., 2001), although we provide an alternative account in a later section. In the model that we are proposing, we have assumed that there is only one, central resource; but the model could be extended to incorporate the idea of multiple resources.

## Recent Attempts to Differentiate Between All-or-None Bottleneck Models and Capacity Sharing Models

To differentiate between bottleneck and capacity sharing models, Pashler and Johnston (1989) looked for situations in which the models predicted different outcomes. They claimed that one such situation occurred when the contrast of the Task 2 stimulus was manipulated. In the critical experiment (Pashler & Johnston, 1989, Experiment 1), participants performed a two-alternative pitch (high vs. low) discrimination task in which they made manual responses. This was followed at varying SOAs by a three-alternative discrimination task based on letter identity (A vs. B vs. C) on each trial. Three SOAs were used: 50, 100, and 400 ms. Pashler and Johnston manipulated contrast by presenting the letter (Task 2) in white on a black background (high contrast, easy condition) on half of the trials, whereas the letter was presented in gray on a black background (low contrast, hard condition) on the other half of the trials. According to a bottleneck account of the PRP effect, the effect of this manipulation should be underadditive with decreasing SOA. The logic for this prediction is illustrated in Figure 2. At short SOAs, because the contrast manipulation affects early, prebottleneck stages of processing, the effect of the manipulation can be absorbed into the period during which Task 2 is waiting for Task 1 to finish bottleneck processing. As a result, bottleneck models predict that there will be no effect of a contrast manipulation at sufficiently short SOAs. However, at long SOAs, Task 2 does not spend any time waiting for Task 1 because Task 1 will have released the bottleneck by the time Task 2 requires it. As a result, the full effect of the contrast manipulation will be seen at long SOAs.

Pashler and Johnston (1989) claimed that capacity sharing models would predict an overadditive (with decreasing SOA) effect of a Task 2 manipulation, regardless of the locus of this manipulation (see Pashler & Johnston, 1989, pp. 41–42). They claimed that this follows from the assumption that at short SOAs, Task 2 does not have full access to capacity, because it is sharing with Task 1. As seen graphically in Figure 3, it will take more time at short SOAs for the system to accomplish the additional work required because of the contrast manipulation than it would at long SOAs in which full capacity is allocated to Task 2. This prediction only follows if it is assumed that capacity sharing occurs for all processing required to perform the tasks. In a subsequent section we show that a different class of capacity sharing models makes the same prediction as bottleneck models for this experimental manipulation. When these predictions were tested empirically, the results were clearly underadditive. Pashler and Johnston rejected the capacity sharing model and favored a bottleneck explanation of the PRP effect.

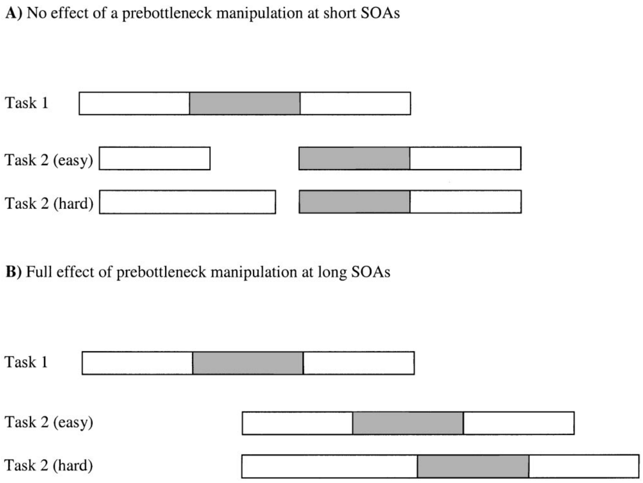
A) No effect of a prebottleneck manipulation at short SOAs
Task 1
Task 2 (easy)
Task 2 (hard)
B) Full effect of prebottleneck manipulation at long SOAs
Task 1
Task 2 (easy)
Task 2 (hard)

*Figure 2.* A bottleneck account of the observed underadditivity of a Task 2 prebottleneck manipulation with decreasing stimulus onset asynchrony (SOA). Shaded regions represent bottleneck processing, whereas unshaded regions represent stages that can process information from different tasks in parallel. (A) At short SOAs, increasing the duration of a prebottleneck stage can be absorbed into the time Task 2 must spend waiting for access to bottleneck processing. (B) At long SOAs, however, this cannot occur because Task 1 has completed bottleneck processing by the time Task 2 requires it. Hence there is no opportunity for the increase in the Task 2 prebottleneck processing to get absorbed into the time Task 2 would have spent waiting for Task 1 to complete central processing.

Pashler (1994b) also argued against capacity sharing models. In this study, participants were presented with a tone and a letter on every trial to which they were required to make speeded manual responses. The SOAs between the tone and the letter were −1,000, −500, 0, +500, and +1,000 ms. Participants were free to respond to the stimuli in any order with both stimuli given equal priority. Pashler was specifically interested in what would occur at the 0-ms SOA. He argued that if dual-task performance is best characterized by a bottleneck model, then three possible response patterns could be observed. In the first pattern, which is diagrammed in Figure 4A, participants perform bottleneck processing on the tone first, then the letter. In the second pattern, which is diagrammed on Figure 4B, the letter enters bottleneck processing first, followed by the tone. In the third pattern, depicted on Figure 4C, one or the other task gains access to the bottleneck stage first, but its response is delayed until the response for the second task has been determined, at which point both responses are executed. Pashler suggested that two types of response patterns should emerge: one type for participants who delay the response for one task until the response for the other task is known (spike results, participants who used Pattern C), and another type that does not (double-ridge

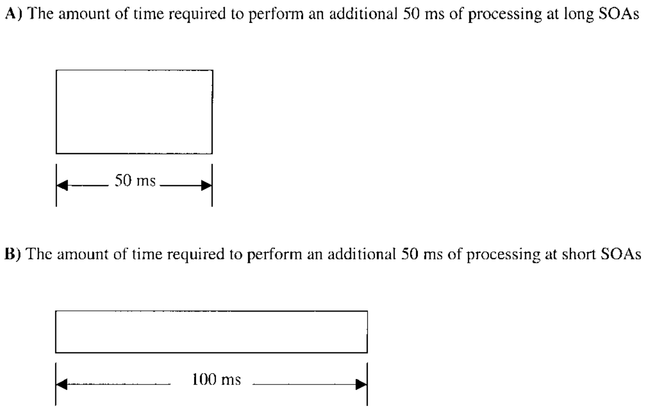
A) The amount of time required to perform an additional 50 ms of processing at long SOAs
50 ms
B) The amount of time required to perform an additional 50 ms of processing at short SOAs
100 ms

*Figure 3.* Pashler and Johnston's (1989) explanation of why capacity sharing models predict overadditivity of a contrast manipulation with decreasing stimulus onset asynchrony (SOA). In this illustration, we have assumed that, at short SOAs, Task 1 and Task 2 share capacity evenly (50/50). (A) Because at long SOAs, Task 2 is not sharing capacity, a 50-ms manipulation will have a 50-ms effect. (B) However, at short SOAs, Task 2 is sharing capacity and will only have access to half the total capacity. This means that it will take twice as long, or 100 ms, to perform the additional work.

results, for participants who used both Patterns A and B). This set of patterns predicts a specific distribution of interresponse intervals (IRIs—the temporal interval between the two responses). Specifically, spike results should have very tight IRI distributions centered on 0 ms, whereas double-ridge results should have bimodal distributions, centered on either side of 0 ms, with very few IRIs of about 0 ms.

Pashler (1994b) argued that if capacity sharing underlies dual-task performance, then a different pattern should be observed. He hypothesized that if both tasks have about equal amounts of the available capacity, then both tasks should be finishing at approximately the same time. However, by random variations in processing time, one task will finish before the other, which should produce a broad distribution of IRIs with a mean at 0 ms.

Six participants produced spike RT distributions, whereas 17 participants produced double-ridged distributions. One participant showed a pattern suggesting a mixture of spiked and double-ridged distributions. Pashler (1994b) interpreted these findings as supporting a bottleneck account of the PRP effect. In later sections, we argue that this test is not as diagnostic as argued by Pashler.

## Central Capacity Sharing Model

Given the above, it would seem that capacity sharing models of the PRP effect fail to account for important empirical evidence. In this section we describe a central capacity sharing model that can account for the findings of Pashler and Johnston (1989) and Pashler (1994b).

The central capacity sharing model begins with the assumption that there are stages of processing that are not capacity limited and others that are. As in most bottleneck models, the present model assumes that capacity-limited stages occur centrally (McCann & Johnston, 1992). We refer to capacity-limited stages as *central stages* from this point forward. Like previous capacity sharing models, the present model assumes that the capacity limitations of the central stages are not all or none. Unlike bottleneck models, the central stages in the present model can process multiple stimuli simultaneously, but when this occurs, processing capacity is shared and processing in both tasks slows down. Initially, we restrict our analysis to the case in which capacity is fixed and fully allocated to the task or tasks being performed; relaxation of the fixed capacity and fully allocated capacity assumptions are explored in a later section.

When total capacity is fixed and fully allocated, we set the sum of the proportions of capacity allocated to Tasks 1 and 2 to equal 1, and, when only one task requires capacity, it receives full capacity. It is important to highlight that this model assumes that capacity sharing is restricted to central stages of processing, hence

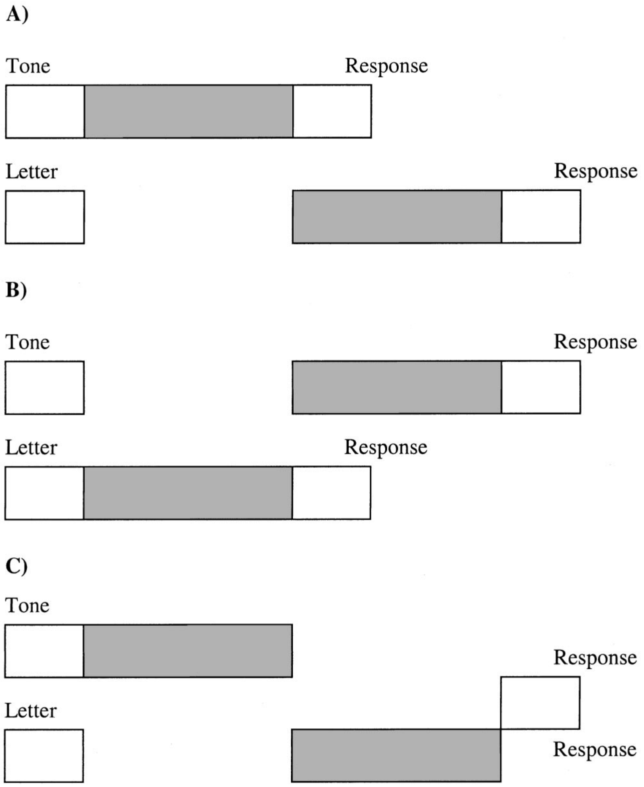

*Figure 4.* Three possible response patterns for the Pashler (1994b) task according to a bottleneck account. Shaded regions represent bottleneck processing, whereas unshaded regions represent stages that can process information from different tasks in parallel. (A) The tone task gets access to bottleneck processing first and a response is emitted immediately. Once the tone has completed bottleneck processing, the letter begins bottleneck processing and emits its response. (B) This is the same as A except that the letter task gains access to the bottleneck first, then the tone. (C) One or the other task gains access to bottleneck processing first, but instead of immediately emitting a response, the response is delayed until the response for the second task has been prepared, at which point both responses are made. Combinations of A and B produce double-ridged responding, whereas C produces spiked responding. From "Graded Capacity-Sharing in Dual-Task Interference?" by H. Pashler, 1994, *Journal of Experimental Psychology: Human Perception and Performance, 20,* p. 334. Copyright 1994 by the American Psychological Association. Adapted with permission of the author.

its name. It is also important to note that a central bottleneck model is a special case of the central capacity sharing model in which all capacity is allocated to Task 1 when both tasks require central processing.

In most PRP studies, there are two cases that summarize how Task 1 and Task 2 central processing overlap. Although other patterns of central processing overlap between Task 1 and Task 2 are possible, these two cases are sufficient to cover what occurs in most PRP studies. All possible cases of central processing overlap are depicted graphically in Figure 5, with Cases A and B being those that apply to most PRP studies. A full presentation of each possible case is included in the Appendix for completeness. In Case A, which occurs at long SOAs in most PRP studies, Task 1 starts and finishes central processing before Task 2 begins central processing (see Figure 5, Case A). In the second case, which occurs at short SOAs, central processing of Task 1 and Task 2 overlap, Task 1 begins central processing before Task 2 begins central processing, and Task 1 finishes central processing before Task 2 (see Figure 5, Case B).

We assume that Case A describes long SOA trials and that Case B describes short SOA trials in the PRP paradigm for the following reasons. In many PRP studies, the mean RT for Task 1 is substantially less than the longest SOA. For example, in Experiment 1 in McCann and Johnston (1992), the mean RT in the first task was 300 ms less than the longest SOA. That means that, on average, participants had executed their first response 300 ms before the presentation of the second stimulus. As a result, on the vast majority of trials, Task 1 and Task 2 central processing could not possibly overlap. Other studies show a similar pattern (Van Selst & Jolicœur, 1997). Therefore, it is reasonable to suppose that long SOA trials are well represented by Case A.

One indication that Case B describes short SOA behavior is that, at short SOAs, when the responses to the first and second tasks are both manual responses, the response for the first task is, on average, elicited well before the response to the second task (DeJong, 1993; McCann & Johnston, 1992; Pashler & Johnston, 1989). If we assume that postcentral processing of the two tasks have roughly the same duration (because both are executed by the same system), the fact that the response to the first task precedes the response to the second task indicates that Task 1 central processing also finished before Task 2 central processing. This establishes that Task 1 central processing finishes before Task 2 central processing.

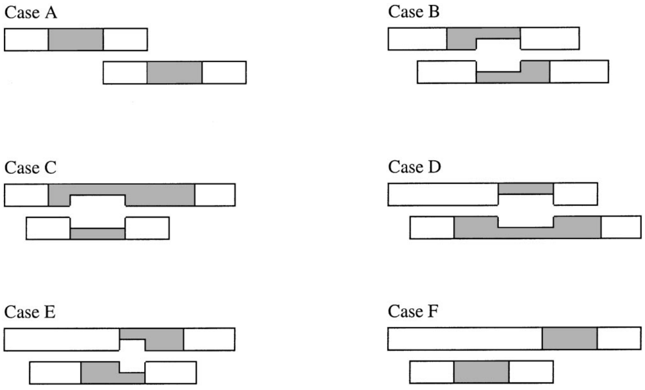

*Figure 5.* The four possible combinations of Task 1 and Task 2 central processing overlap (Cases B, C, D, and E) and the two possible combinations in which central processing of Task 1 and 2 does not overlap (Cases A and F). Of critical importance to the psychological refractory period literature are Cases A and B. Cases A and B are discussed in the main body of the article, whereas a complete exploration of all cases is included in the Appendix.

Finally, how do we know that Task 2 central processing overlaps with Task 1 central processing? It is conceivable that Task 1 central processing has finished by the time Task 2 begins central processing. However, dual-task slowing is observed in Task 2, implying that this is not the case. If Task 2 had full access to central resources, it would not have been delayed. These considerations support the assumptions that, at long SOAs, most trials fall under Case A, whereas at short SOAs, most trials fall under Case B.

Nonetheless, it is possible to imagine pairs of tasks that are not well described by Case B at short SOAs and Case A at long SOAs. Consider a Task 1 that requires a very long period of central processing and a Task 2 that only requires a minimal amount. At short SOAs, Task 1 would begin central processing first. After some time Task 2 would also begin central processing. It is possible that Task 2 could finish central processing before Task 1, in which case the assumption that Case B characterizes central processing at short SOAs would be violated. Fortunately, the response for Task 2 would likely precede the response for Task 1, which would act as an indicator that Case B no longer applied. When testing the predictions of the central capacity sharing model, one must compare empirical results with the case that most appropriately describes the pattern of Task 1 and Task 2 central processing overlap. In general, PRP studies have used tasks that are well described by Case B at short SOAs and Case A at long SOAs.

What predictions does the central capacity sharing model make under these conditions? It is possible to derive equations for the RTs for both Task 1 and Task 2 for both cases. We express the total time taken to perform a task in isolation as the sum of precentral, central, and postcentral stage durations. We refer to these as A, B, and C respectively, where each letter represents the total amount of time needed to complete a stage under full capacity (this distinction is only important to central stages, B, which are capacity limited). We define the *sharing proportion* (SP) as a number from 0 to 1 that represents the proportion of capacity allocated to Task 1 when both tasks require central processes. To be clear, capacity sharing only occurs when the B stages of Task 1 and Task 2 overlap.

## *Case A: No Central Overlap Between Task 1 and Task 2*

In Case A, Task 1 central processing precedes Task 2 central processing, and there is no overlap between central processing of Task 1 and Task 2 (see Figure 5, Case A). This case occurs at long SOAs in most PRP experiments. Task 1 and Task 2 do not require access to central stages simultaneously over the full course of processing, and Task 1 gains access to central stages first. Task 1 RTs can be expressed by the formula A1 + B1 + C1, where A1 refers to the duration of A processing required for Task 1. Likewise Task 2 RTs can be expressed by the formula A2 + B2 + C2.

In summary, RT1 and RT2 are as follows:

$$\mathrm{RT1} = \mathrm{A1} + \mathrm{B1} + \mathrm{C1}, \tag{1}$$

$$\mathrm{RT2} = \mathrm{A2} + \mathrm{B2} + \mathrm{C2}. \tag{2}$$

These equations are identical to those derived from a central bottleneck model at long SOAs.

### *Case B: Central Processing for Task 1 and Task 2 Overlap*

In Case B, central processing for Task 1 and Task 2 overlaps, Task 1 starts central processing before Task 2 central processing starts, and it finishes before Task 2 central processing finishes (see Figure 5, Case B). This case occurs at short SOAs in most PRP experiments. Initially, Task 1 has full access to central stages and uses all available capacity. The period of time for which Task 1 has full capacity can be expressed by the expression SOA + A2 − A1. When Task 2 gains access to some of the central capacity, Task 1 and Task 2 must share the processing capacity (unless all capacity is allocated to either Task 1 or Task 2) until such a time as Task 1 has completed central processing. At that moment, Task 1 has completed SOA + A2 − A1 units of central processing, leaving B1 − (SOA + A2 − A1) units of central processing remaining to be performed. Because Task 1 only has a proportion of central capacity available to it (SP), it will take [B1 − (SOA + A2 − A1)] / SP units of time to perform this central processing. The derivation of this term is illustrated in Figure 6. The total amount of time to complete Task 1 processing can be expressed as the sum of its precentral processing (A1), central processing (SOA + A2 − A1 + [B1 − (SOA + A2 − A1)] / SP), and postcentral processing (C1), which is summarized in the following equation:

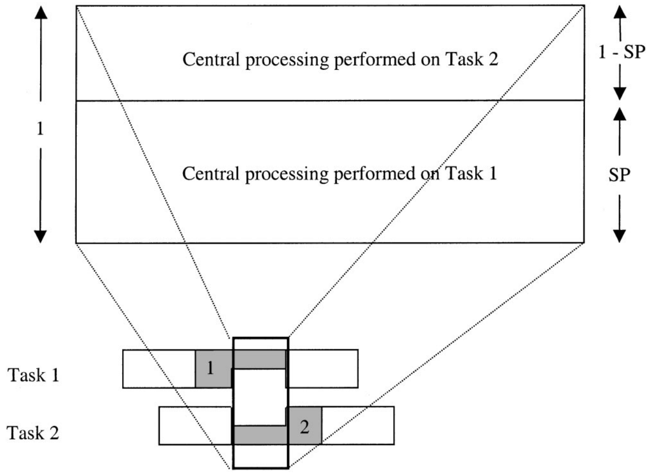
Central processing performed on Task 2
Central processing performed on Task 1
1
1 - SP
SP
Task 1
Task 2
1
2

*Figure 6.* Central processing when central capacity is shared between Task 1 and Task 2. At this point, Task 1 has performed A2 + SOA − A1 units of central processing (in the area marked 1), leaving B1 − (A2 + SOA − A1) units to be performed (to be performed while sharing capacity with Task 2, which is shown in the blowup box). This will take [B1 − (A2 + SOA − A1)] / SP units of time to perform because it has a reduced proportion of capacity allocated to it (SP). In addition, (1 − SP) × [B1 − (A2 + SOA − A1)] / SP units of central processing are performed on Task 2 while Task 1 and Task 2 share central processing. This leaves B2 − (1 − SP) × [B1 − (A2 + SOA − A1)] / SP units of Task 2 central processing to be performed once Task 2 gains full access to central processing (which is performed in the area marked 2). SOA = stimulus onset asynchrony; SP = sharing proportion.

$$\mathrm{RT1} = \mathrm{A1} + \mathrm{SOA} + \mathrm{A2} - \mathrm{A1} + [\mathrm{B1} - (\mathrm{SOA} + \mathrm{A2} - \mathrm{A1})] / \mathrm{SP} + \mathrm{C1}.$$

This equation can be simplified to the following:

$$\mathrm{RT1} = \mathrm{SOA} + \mathrm{A2} + (\mathrm{B1} - \mathrm{SOA} - \mathrm{A2} + \mathrm{A1}) / \mathrm{SP} + \mathrm{C1}. \tag{3}$$

Equation 4 reexpresses the equation in a way that allows us to examine the individual contributions of each processing stage and of SOA:

$$\mathrm{RT1} = [(\mathrm{SP} - 1) / \mathrm{SP}] \times \mathrm{SOA} + [(\mathrm{SP} - 1) / \mathrm{SP}] \times \mathrm{A2} + \mathrm{A1} / \mathrm{SP} + \mathrm{B1} / \mathrm{SP} + \mathrm{C1}. \tag{4}$$

Because SP is always less than (or equal to) 1, (SP −1) / SP will be negative as long as Task 2 receives even a minimal amount of capacity under sharing. This means that decreasing the SOA will increase the Task 1 RT. Similarly, increasing the duration of Task 2 precentral processing (A2) will decrease the Task 1 RT. Increasing the duration of any Task 1 stage will increase the Task 1 RT.

It is also interesting to note that, according to the central capacity sharing model, A1 and B1 manipulations should have larger effects at short SOAs relative to long SOAs (Case A, Equation 1). A 1-ms increase in either A1 or B1 at short SOAs results in a 1 / SP-ms increase in RT1 (Equation 4). However, at long SOAs, a 1-ms increase in either A1 or B1 results in a 1-ms increase in RT1 (Equation 1). It is also worthwhile pointing out that if SP equals one, the central capacity sharing model makes the same predictions as the central bottleneck model for RT1.

Next we derive the expression for RT2. Task 2 begins central processing while capacity must be shared with Task 1; sharing continues until Task 1 completes central processing, at which point Task 2 processing continues with all available central capacity. From the derivation of RT1, we know that the time Task 2 spends sharing is equal to [B1 − (SOA + A2 − A1)] / SP. During this period of time, Task 2 accomplishes (1 − SP) × [B1 − (SOA + A2 − A1)] / SP units of central processing, at which point it gains access to full central processing capacity. As a result, it takes B2 − (1 − SP) × [B1 − (SOA + A2 − A1)] / SP additional units of time for Task 2 to finish central processing (see Figure 6). So, the total amount of time to complete Task 2 processing can be expressed as the sum of precentral processing (A2), central processing {[B1 − (SOA + A2 − A1)] / SP + B2 − (1 − SP) × [B1 − (SOA + A2 − A1)] / SP}, and postcentral processing (C2), which yields the following equation:

$$\mathrm{RT2} = \mathrm{A2} + [\mathrm{B1} - (\mathrm{SOA} + \mathrm{A2} - \mathrm{A1})] / \mathrm{SP} + \{\mathrm{B2} - (1 - \mathrm{SP}) \times [\mathrm{B1} - (\mathrm{SOA} + \mathrm{A2} - \mathrm{A1})] / \mathrm{SP}\} + \mathrm{C2}.$$

Combining the two terms containing B1 − (SOA + A2 − A1) / SP yields

$$\mathrm{RT2} = \mathrm{A2} + \mathrm{B2} + \mathrm{C2} + [1 - (1 - \mathrm{SP})] \times [\mathrm{B1} - (\mathrm{SOA} + \mathrm{A2} - \mathrm{A1})] / \mathrm{SP}.$$

This equation can be simplified to the following:

$$RT2 = A1 + B1 + B2 + C2 - SOA. \tag{5}$$

Of particular interest, RT2 is not dependent on the SP or the duration of precentral stages of Task 2.

It is important to note that these equations only hold as long as Task 1 starts central processing before Task 2 starts central processing, Task 1 finishes central processing before Task 2 finishes central processing, and central processing of Task 1 and Task 2 overlaps. If changing parameters in the equations causes any of these conditions to be violated, a different set of equations become applicable for calculating RT1 and RT2. Equations for all possible cases of central processing overlap and conditions determining which set is applicable are included in the Appendix. As mentioned previously, Cases A and B should be sufficient for most scenarios that occur in typical PRP experiments, hence the focus placed on them in this presentation.

## PRP Phenomena Explained by the Central Capacity Sharing Model

### *Underadditivity of a Precentral Task 2 Manipulation*

The central capacity sharing model can account for the underadditive pattern observed when an early, precentral stage of processing is manipulated in Task 2. As can be seen from Equation 5, RT2 at short SOAs (Case B) does not depend on A2. Therefore, a precentral, Task 2 manipulation has no effect on RT2; whereas at long SOAs (Case A), RT2 is affected by precentral Task 2 manipulations because RT2 now depends on A2 (see Equation 2). This is diagrammed in Figure 7.

### *Slope of −1 for the PRP Effect*

In Task 2, effects of SOA are predicted to have a slope of −1 at short SOAs, as can be seen from Equation 5. At long SOAs there will be no effect of SOA, as can be seen from Equation 2. So, as the SOA is reduced from long SOAs to short SOAs, the slope of the RT2 function will increase (in absolute value) until at the shortest SOA, it will have a slope of −1. Thus the central capacity sharing model predicts a PRP effect with a flat SOA effect for long SOAs and with a slope of −1 for short SOAs, just as a bottleneck model does, and as is often observed experimentally.

### *Effects of SOA on RT1*

The central capacity sharing model predicts that as SOA decreases, RTs to Task 1 will increase unless the SP equals one, in which case RT1 is constant at all SOAs. This prediction can be demonstrated by comparing Equation 4 with Equation 1:

$$[(SP - 1) / SP] \times SOA + [(SP - 1) / SP] \times A2 + A1 / SP + B1 / SP + C1 \geq A1 + B1 + C1.$$

Subtracting C1 from both sides and multiplying both sides by SP yields

$$SP \times SOA - SOA + SP \times A2 - A2 + A1 + B1 \geq SP \times A1 + SP \times B1.$$

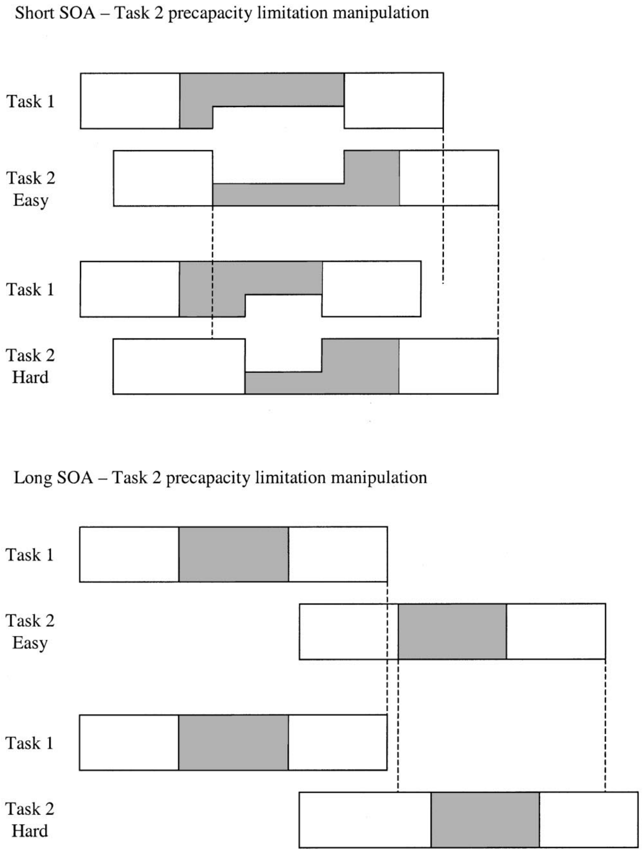
Short SOA – Task 2 precapacity limitation manipulation
Task 1
Task 2
Easy
Task 1
Task 2
Hard
Long SOA – Task 2 precapacity limitation manipulation
Task 1
Task 2
Easy
Task 1
Task 2
Hard

*Figure 7.* The effect of a precentral stage Task 2 manipulation. At short stimulus onset asynchronies (SOAs), this manipulation has no effect on Task 2, but increasing the Task 2 difficulty does result in decreasing Task 1 response times. At long SOAs, making Task 2 more difficult precentrally increases the amount of time it takes to perform Task 2 and has no effect on Task 1.

Grouping all the terms containing SP on the right side of the equation yields

$$A1 + B1 - SOA - A2 \geq SP \times (A1 + B1 - SOA - A2).$$

Dividing both sides by A1 + B1 − SOA − A2 yields

$$1 \geq SP,$$

which is true by definition. Therefore, the central capacity sharing model predicts that RT1 at short SOAs will be larger than RT1 at long SOAs when the SP is less than 1. The size of the increase in RT1 as SOA decreases will vary depending on the SP. The larger the proportion of central processing allocated to Task 1 (while sharing), the smaller the SOA effect. To account for the results in most previous PRP studies, we assume that Task 1 received a large proportion of central processing when sharing. However, the frequent report of experiments with sizable SOA effects in RT1 suggests that SP is often less than 1 (e.g., Carrier & Pashler, 1995; Duncan, 1979; Kahneman, 1973).

On the other hand, there are many reported experiments in which Task 1 SOA effects were not observed. From a central capacity sharing framework, this can be explained by setting SP equal to one. When SP equals one, no RT1 SOA effect is predicted. The central capacity sharing model can predict both the absence and presence of SOA effects on RT1 depending on the SP, which is likely to be influenced by task demands. For example, if task order is fixed, this is likely to drive SP toward one (all capacity allocated to Task 1). Task order is fixed in many PRP studies, so these studies are unlikely to demonstrate large Task 1 SOA effects (unless some other factor drives SP away from one; see below). In addition, most PRP studies require the response to the first task to precede the response to the second task. It seems reasonable to assume that this too would drive SP toward one. The duration of central processing is also likely to affect SP. If the first task is complex and displayed for only a short duration, some central processing of Task 2 may be required to stop the trace of the second stimulus from decaying. Other factors are also likely to influence SP, and determining what they are is an empirical issue.

To test for RT1 SOA effects, it is necessary to find a set of tasks that are likely to encourage participants to share capacity. Many existing PRP studies have factors that likely drive SP toward one (e.g., fixed task order with relatively simple tasks). Therefore, the fact that many studies show small or nil RT1 SOA effect is not surprising. Driving SP away from one by using a variable, within-block presentation order, more centrally demanding tasks, or placing emphasis on the second task might create conditions in which larger Task 1 SOA effects may be expected.

### *Effects of a Task 2 Precentral Manipulation on RT1*

The central capacity sharing model predicts that at short SOAs, RT1 will be affected by a Task 2 precentral manipulation. Specifically, when Task 2 takes longer at a precentral stage, RTs to Task 1 will decrease relative to RT1 in the easy condition of Task 2. This decrease in Task 1 RTs should be small unless the SP is set to allocate a large amount of processing to Task 2 (while capacity is being shared) and the effect size of the Task 2 manipulation is large. This prediction is a consequence of Equation 4. The A2 term in Equation 4 is $[(SP - 1) / SP] \times A2$. Because SP is always less than or equal to 1 and greater than 0, the A2 term will be negative as long as some capacity is allocated to Task 2 while sharing ($SP < 1$). Therefore, increasing A2 will decrease RT1. The smaller the value of SP, the larger the impact of the A2 term on RT1. The predicted size of the A2 effect on RT1 is $(A2 / SP) - A2$. Although the equations derived in this work and those derived by Navon and Miller (2002) appear to be the same, Navon and Miller did not predict any effect of a Task 2 manipulation on Task 1 RTs (see Navon & Miller, 2002, p. 219, Table 2). This prediction seems very important given that the central bottleneck model clearly does not make this prediction. These divergent predictions offer a method by which to distinguish between the two models. In fact, a pattern of results consistent with this prediction has been reported in a study by Pashler (1991, Experiment 4).

### *A Reinterpretation of the Findings of Pashler and Johnston (1989) and Pashler (1994b)*

As shown in the foregoing section, the central capacity sharing model predicts the underadditivity that Pashler and Johnston (1989) observed. Thus the evidence and arguments provided by Pashler and Johnston (1989) do not reject the central capacity sharing model. We now turn to the results of Pashler (1994b). Recall that on each trial a tone and a letter were presented and participants were required to make speeded responses. Five SOAs were used in the tone presentation: −1,000, −500, 0, +500, and +1,000 ms, with equal probability. Participants were free to respond in any order. We are specifically interested in looking at the predictions Pashler made concerning how IRIs would be distributed according to a capacity sharing model in the 0-ms condition. Pashler's (1994b) conclusions hinge on the assumption that capacity sharing models would predict that, in the 0-ms SOA condition, both tasks would receive an equal proportion of the available capacity (SP of .5). This would result in a broad distribution of IRIs centered around 0 ms. However, there is strong evidence that the response order of a previous trial influences the response order of the next trial (DeJong, 1995; Tombu & Jolicœur, 2000). The results suggest a bias to repeat the response order of the previous trial. One way to produce this type of bias in a capacity sharing model is to assume that more capacity is allocated to the task that emitted the first response in the previous trial.

Given this interpretation, one would not expect that both tasks would receive equal priority (or capacity) in the 0-ms condition. Perhaps *on average* this would be the case, but on a trial-to-trial basis, one task could receive more capacity than the other. Indeed, any trial-to-trial fluctuations in allocation policy, be they due to momentary bias in response order or to other causes, would tend to produce bimodal distributions, centered on either side of 0 ms, with few observed IRIs at 0 ms. Thus, the observation that response distributions are often double-ridged (bimodal distribution of IRIs centered on opposite sides of 0 ms) does not seem like a strong diagnostic test to distinguish capacity sharing from bottleneck models.

How might a capacity sharing model account for the results produced by the six spiked participants (tight unimodal distribution of IRIs centered at 0 ms)? In the same manner that a bottleneck model does: If participants hold onto their first response until such a time as they also have the second response prepared, a spiked distribution of IRIs will be produced.

There is an additional interesting finding in Pashler's (1994b) data set. According to a bottleneck model, RTs to the first task should be the same regardless of the SOA between the first and second stimulus. To test this hypothesis, Pashler looked for differences in the RTs to the tone when the tone was responded to first when it preceded the letter by 1,000 ms and when the tone and letter were presented simultaneously. This analysis was also performed on trials in which the letter was responded to first when the presentation of the letter preceded the presentation of the tone by 1,000 ms and when they occurred simultaneously. The results were clearly not in line with the predictions of the bottleneck model: When the tone was responded to first, RTs were 104 ms faster at the 1,000-ms SOA than at the 0-ms SOA (654 ms vs. 758 ms). When the letter was responded to first, the same pattern emerged. RTs were 77 ms faster in the 1,000-ms SOA condition than in the 0-ms SOA condition (697 ms vs. 774 ms).

To reconcile these results with the predictions of the bottleneck model, Pashler (1994b) argued that RTs to Task 1 were slower at the short SOA because the presentation order was not known. He argued that participants are likely to delay processing the first stimulus until such a time as the presentation order has been determined on a small number of trials. This is more likely to have larger effects at the 0-ms SOA because temporal order discrimination will be more difficult than at the 500-ms or 1,000-ms SOAs. In fact, this pattern of results has been found under circumstances in which task order was variable (Pashler, 1990). However, there is a potential flaw in this logic. In Pashler's 1990 study, participants were instructed to respond in the presentation order. In the 1994b study, participants were free to respond in whichever order they wished. If presentation order is not important to responses, as in the 1994b study, why would participants delay processing until such a time as presentation order could be determined? The works of DeJong (1995) and Tombu and Jolicœur (2000) suggest that they do not.

DeJong (1995, Experiment 2) presented participants with one of two tones to which they had to make a speeded two-alternative pitch discrimination, and with one of four shapes to which they had to make a four- to two-alternative identity discrimination (i.e., two of the shapes mapped to one response key, and the other two shapes mapped to the other response key). The SOA between the two stimuli was either 100 or 400 ms, and the presentation order was randomized (i.e., on a random half of the trials the tone was presented first, whereas on the other half the shape was presented first). For some blocks, participants were required to respond to the stimuli in the same order as the stimulus presentation order, whereas for other blocks they were free to respond in any order. A significant interaction between SOA and response order demands was observed. When participants tried to respond to the stimuli in the order in which they were presented, their responses to the first stimulus were slower at the 100-ms SOA than at the 400-ms SOA. Participants were 9 ms slower at the short SOA when the shape was presented first and 13 ms slower at the short SOA when the tone was presented first. However, when participants were free to respond in any order, a different pattern was observed: Responses to the first stimulus were faster at the 100-ms SOA than at the 400-ms SOA. When the tone was presented first they were 18 ms faster at the short SOA, and when the shape was presented first they were 22 ms faster at the short SOA.

Tombu and Jolicœur (2000) performed a study that also addresses the effects of constraining response orders. In their study participants were presented with a letter and a tone on every trial, and they were required to make speeded responses (both were two-alternative discrimination judgments). Both tasks occurred first with equal probability, and the stimulus for the second task followed the first at varying SOAs (50, 150, 350, 750, or 1,550 ms with equal probability). Participants performed two sessions under different instructions. In one session participants were free to respond to the stimuli in whichever order they wished, whereas in the other session they were instructed to respond to the stimuli in the same order as the presentation order. All participated in both conditions, and the order in which they performed them was counterbalanced across participants. Of critical importance to the present discussion was the pattern of Task 1 RTs with decreasing SOA in each condition. Experiments 1 and 3 gave the clearest results, so we concentrate on them. These results have been reproduced in Figure 8. When responses were constrained to match the stimulus presentation order, the results were much the same as those reported by Pashler (1990). As SOA decreased, RT1 increased. On average, RT1 increased 104 ms from the longest to the shortest SOA. However, when participants were free to respond to the stimuli in any order, there was only a slight effect of SOA, and it was in the opposite direction; as SOA decreased, RTs decreased slightly, as found by DeJong (1995). On average, RT1 decreased 32 ms from the longest to the shortest SOA. In summary, when participants are constrained to match response order to stimulus order, RTs to Task 1 are slower at short SOAs than at long ones. However, when response order is not constrained, this effect disappeared and RTs to Task 1 increased slightly with increasing SOA.

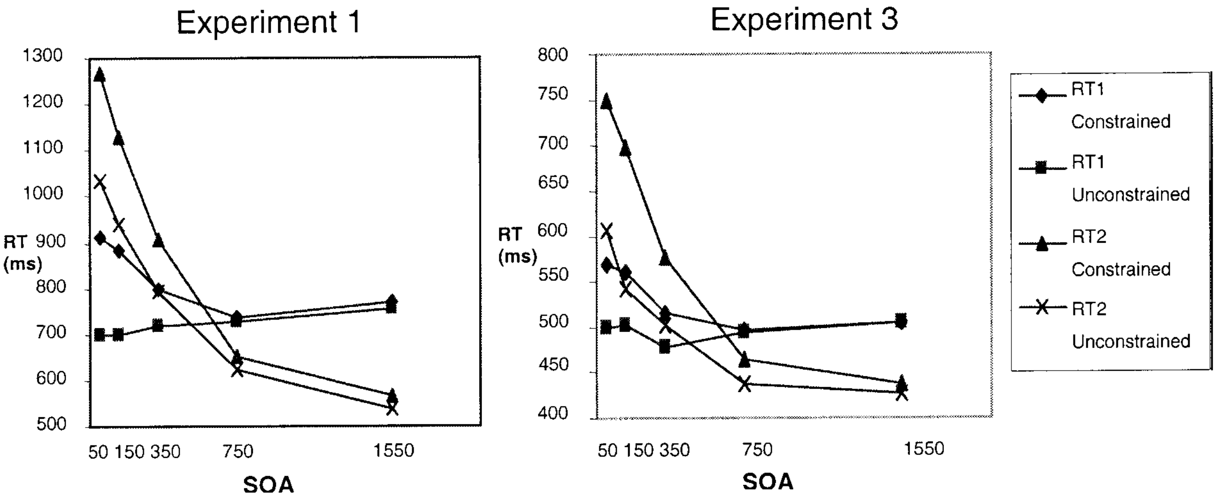

*Figure 8.* Results of all trials from Experiments 1 and 3 from Tombu and Jolicœur (2000) in which response order matched presentation order. When response order was constrained, and participants were required to respond to the stimuli in the same order in which they were presented, Task 1 response times (RTs) increased with decreasing stimulus onset asynchrony (SOA). However, when participants were free to respond to the stimuli in whatever order they wanted, RTs to Task 1 were generally unaffected by changes in SOA (there was a slight decrease in RT to Task 1 with decreasing SOA).

Participants in Pashler's (1994b) study were not constrained to respond to the stimuli in the presentation order. On the basis of the results of DeJong (1995) and Tombu and Jolicœur (2000), and assuming that a bottleneck characterizes human information processing, there should have been no increase in RT1 with decreasing SOA. However, RT1 did increase with decreasing SOA, and therefore this result is problematic for bottleneck models.

In contrast, the central capacity sharing model predicts that RTs to Task 1 will increase with decreasing SOA as long as the SP between Task 1 and Task 2 is less than one. The greater the proportion of central capacity allocated to Task 2, the greater the effect of SOA on RT1. Given that stimulus order was uncertain and the use of a zero SOA, it is likely that capacity will be more evenly split between Task 1 and Task 2 than in situations in which stimulus order is known in advance. When presentation order is fixed, participants can prepare for the first task, which likely results in a greater proportion of capacity (perhaps close to all) allocated to the first task.

One might have expected that in the unconstrained conditions for both DeJong (1995) and Tombu and Jolicœur (2000), SP would have been less than one given that presentation order was random. However, in both of these experiments incentives were used to encourage the participants to respond as quickly as possible. Given that capacity sharing results in increased RT1s, incentives to respond very quickly might discourage capacity sharing and drive SP to one.

It is worth pointing out that the Task 1 slowing at short SOAs when participants are constrained to respond in the presentation order is not only a result of capacity sharing but is likely also caused by participants having to make a judgment of stimulus order. Similar RT1 effects have been reported by Herman and McCauley (1969) even when the second stimulus does not require a response (see Herman & Kantowitz, 1970, for an overview of this sort of RT1 effect).

We can estimate an upper bound on the SP in Pashler's (1994b) study by comparing the RT1s at short and long SOAs for each response order. If we examine Equation 4, we can see that as SOA is varied RT1 varies. So, the difference in RT1 between two SOAs is equal to the difference in SOA terms from Equation 4: $\text{RT1(short)} - \text{RT1(long)} = [(1 - \text{SP}) / \text{SP}] \times$ difference in SOA. However, at some SOA we switch cases from Case B to Case A, and Equation 1 applies. The transition from Equation 4 to Equation 1 is a smooth transition. To switch cases from Case B to Case A, central overlap between Task 1 and Task 2 needs to stop. If Case B applies, $\text{A2} + \text{SOA}$ must be less than or equal to $\text{A1} + \text{B1}$. At the boundary between Case A and Case B, $\text{A2} + \text{SOA}$ equals $\text{A1} + \text{B1}$. Expressed differently, A2 equals $\text{A1} + \text{B1} - \text{SOA}$. If we substitute this value of A2 into Equation 4, it becomes Equation 1. To get an accurate estimate of SP, the difference between two RT1s from Case B should be used. However, if one of the RT1s being examined is from Case A, it is still possible to get an upper bound on SP.

By examining trials in which participants responded in the same order at both long and short SOAs, we can estimate the SP value for Pashler's (1994b) study. The RT was 758 ms when the SOA was zero and participants responded to the tone first. When the letter was presented 1,000 ms after the tone and the tone was responded to first, participants took 654 ms to respond to the tone. Using these values, we can estimate the upper bound on SP when the tone was responded to first to be .906. The RT was 774 ms when the SOA was zero and the letter was responded to first, whereas when the tone was presented 1,000 ms after the letter and the letter was responded to first, participants took 697 ms to respond to the letter. Using these values, we can estimate the upper bound of SP when the letter was responded to first to be .929. Given that the long SOA is longer than the RT1 means, it is likely that these values of SP are significantly overestimated. However, it is still interesting to note that a significant amount of capacity was allocated to Task 2.

## Possible Methods to Differentiate Between the Central Capacity Sharing and Bottleneck Models

How might we choose between the central capacity sharing model and bottleneck models? A condition in which SP is driven down is required (i.e., some sharing must take place). The simplest approach would be to instruct participants to share capacity between the two tasks, although other manipulations might also encourage capacity to be shared between the two tasks. Determining which manipulations meet this criterion remains an empirical issue that requires further investigation. However, if such a situation could be developed, the central capacity sharing model and bottleneck models would make different predictions. Bottleneck models predict no effect of SOA on Task 1 in all cases, whereas the central capacity sharing model predicts that as SOA decreases RTs to Task 1 should increase (if $\text{SP} < 1$). It would seem that we have a method with which to distinguish between the two models. The results of Pashler (1994b) speak to this and suggest that the central capacity sharing model may be better at predicting how participants perform in some dual-task situations. In Pashler's experiment, participants' responses to Task 1 increased as SOA was decreased, as predicted by the central capacity sharing model but not bottleneck models. Further investigations along this line are nonetheless required (see Tombu & Jolicœur, 2002, for one such investigation).

The central capacity sharing model predicts that a precentral Task 2 manipulation will have an inverse effect on RT1 at short SOAs. However, such an effect is likely to be rather small and hard to detect. Nonetheless, this is a clear prediction of the central capacity sharing model, and it would be useful to attempt to verify this prediction.

Other alternatives that might be useful to differentiate between the two models would be to set up situations in which other patterns of central processing overlap (i.e., cases other than Case A and Case B) occur and test the predictions of the central capacity sharing model. If central capacity is shared as postulated by the central capacity sharing model, then several additional predictions can be derived from the equations provided in the Appendix.

## Memory Retrieval and the PRP Paradigm

Carrier and Pashler (1995) performed two experiments in which a memory retrieval task was the second task of a PRP paradigm. The purpose of these experiments was to determine if memory retrieval occurs in parallel with other cognitively demanding operations. In both experiments a tone was presented first, and it required a speeded two-alternative pitch discrimination response. In the first experiment, participants started by learning seven paired associates. Following the tone at varying SOAs, a cue was presented and the task was to name the paired associate as quickly as possible. In this experiment participants were instructed to place equal emphasis on the two tasks. Carrier and Pashler manipulated the difficulty of the memory retrieval task by presenting each cue twice over the course of the experiment. It was hypothesized that participants would be faster to name the paired associate on the second presentation of the cue. From a bottleneck framework, if memory retrieval can occur in parallel with other cognitively demanding operations, the effect of this manipulation on RT2 should be underadditive with decreasing SOA. If memory retrieval cannot occur in parallel with other cognitively demanding operations, the effect of this manipulation on RT2 should be additive with SOA. In addition, the bottleneck model predicts that there should be no effect of SOA on RT1. The effect of the manipulation on RT2 was clearly additive with SOA, and Carrier and Pashler concluded that memory retrieval cannot occur in parallel with other cognitively demanding operations. However, a significant effect of SOA was observed for RT1: As SOA decreased, RT1 increased.

Experiment 2 replicated Experiment 1 with a different memory retrieval task. In this case, a recognition task was used as Task 2. Before each group of experimental trials began, participants learned a list of words. Words in the list could be presented one or five times. It was hypothesized that presenting a word five times would lead to faster recognition at test time. In the experimental trials the first task required a speeded pitch discrimination to a tone. At varying SOAs a word that required a yes-no judgment was presented (was this a word from the study list?). Participants were instructed to make all responses as quickly and accurately as possible and to do as well as possible on the recognition task. If recognition memory retrieval can occur in parallel with other cognitively demanding operations, then the effect of number of presentations in the learning phase should be underadditive with decreasing SOA. If recognition memory retrieval cannot occur in parallel with other cognitively demanding operations, then the effect of this manipulation should be additive with SOA. The results were clearly additive, leading Carrier and Pashler to conclude that memory retrieval cannot occur in parallel with other cognitively demanding operations. However, a significant effect of SOA was again observed for RT1: As SOA decreased, RT1 increased. This effect is a hallmark of capacity sharing and suggests that participants were sharing capacity between the tone and word tasks. In addition, there was a small main effect of the Task 2 memory retrieval manipulation on RT1: Participants were faster on the tone task if they had seen the Task 2 word five times in the memory set than if they had only seen it once. It is unclear, both from a central capacity sharing or a bottleneck model framework, what might have caused this effect.

Carrier and Pashler (1995) argued that one possible explanation for the RT1 SOA effect could be that participants grouped their responses at short SOAs. Capacity models are briefly considered as an alternative account, but Carrier and Pashler rejected this idea on the basis of past results. Two pieces of evidence against capacity sharing models were discussed. The first piece of evidence was from Pashler's 1994b article examining IRIs that we have discussed earlier. As was demonstrated, the central capacity sharing model can account for this result. The second piece of evidence is the correlation of RT1 and RT2 at short SOAs. Carrier and Pashler claimed that this correlation is predicted by bottleneck models. However, the central capacity sharing model presented herein also predicts that, at short SOAs, fast RT1s lead to fast RT2s. When RT1 is fast, Task 1 will finish central processing sooner, releasing it for Task 2 to access sooner, which means that Task 2 will also finish sooner. This piece of evidence provides as much support for the central capacity sharing model as for bottleneck models. Thus it is also insufficient to rule out the central capacity sharing model.

An alternative explanation of these results comes to a conclusion that is opposite to that drawn by Carrier and Pashler (1995). The RT1 SOA effect observed in both experiments is easily explained by the central capacity sharing model. In this view, both the tone and the memory retrieval task were processed in parallel, with most of the capacity allocated to the tone task. However, some capacity was also allocated to the memory task. As a result, the RT1 SOA effect that was observed in both experiments would be expected. This explanation is more parsimonious than postulating two causes of the observed data (bottleneck postponement as well as conjoined responding [grouping], which only happens at short SOAs). Instead, the RT1 SOA effects and the PRP results can all be explained by the same mechanism: a limited pool of central resources that is shared between the two tasks.

Reinterpreting these results as a demonstration of capacity sharing allows us to reconcile Carrier and Pashler's (1995) result with other results in the domain of memory retrieval. Other researchers have concluded that memory retrieval can be carried out in parallel with other cognitively demanding operations (Hommel, 1998; Logan & Delheimer, 2001; Logan & Schulkind, 2000). For example, Hommel (1998) demonstrated crosstalk between two memory tasks. Participants were presented with a red or green H or S and were required to make speeded responses to the color of the letter, the identity of the letter, or both. In Experiment 1 participants responded to the identity of the letter by saying left or right and made a manual left or right key press to the color of the letter. When a Task 2 response and a Task 1 response were compatible, RT1 was faster than when they were incompatible. In Experiment 2, participants were again presented with a red or green H or S, they again made manual left or right key presses in response to the color of the letter, but this time they responded to the letter identity by saying red or green. In Experiment 2, when a Task 2 response and a Task 1 stimulus were compatible, RT1 was faster than when they were incompatible. When the response for the second task was compatible with either the response or the stimulus for the first task, RT1 was faster than when they were incompatible. Therefore, there appeared to be crosstalk between these two concurrent tasks. From this and similar demonstrations of crosstalk, Hommel concluded that parallel memory retrieval was possible. Similar results have been found by Logan and colleagues (Logan & Delheimer, 2001; Logan & Schulkind, 2000), but only when the two tasks share a common task set. It seems difficult to explain these results if memory retrieval cannot occur in parallel with other cognitively demanding tasks. Within a central capacity sharing framework, processing of both tasks can be carried out simultaneously at rates that depend on the SP. Because central processing of both tasks is occurring in parallel, crosstalk between them is possible. This allows for a reconciliation of the apparently contradictory results of Carrier and Pashler (1995) and those of Hommel (1998) and Logan and Schulkind (2000). The results of Carrier and Pashler

(1995) demonstrate that memory retrieval requires a central stage of processing but that capacity is shared between the two tasks. Given that this set of tasks results in capacity sharing, when the two tasks require similar processing to be performed, crosstalk might be observed. This is precisely what was observed in Logan and Schulkind (2000) and Hommel (1998). Although capacity sharing could account for some cases of crosstalk (such as those discussed here), the observation that crosstalk takes place even after participants are instructed to stop performing a secondary task (Hommel & Eglau, in press) suggests that it is not the only source of crosstalk.

## Relaxation of the Assumption of Fixed Capacity

Up until this point we have assumed that central capacity is fixed. There is good evidence that available capacity increases when participants put more effort into the to-be-performed tasks (Kahneman, 1973). When a pair of to-be-performed tasks becomes more difficult, participants put more effort into the tasks and, as a result, marshal more capacity. However, this ability to increase capacity is subject to diminishing returns. Eventually, participants reach a point at which they are already exerting maximum effort and a further increase in task difficulty will not increase effort. There are several ways in which a pair of to-be-performed tasks may become more difficult. For example, the tasks themselves can be made more difficult by manipulating stimulus–response mappings or the signal-to-noise ratio, or the pair of to-be-performed tasks may become more difficult by decreasing the SOA. In addition, performing two tasks at once is more difficult than performing either task in isolation. Under the right set of circumstances (i.e., with tasks that are sufficiently easy when performed alone), it is conceivable that the increased capacity marshaled by increases in effort in the dual-task situation would be sufficient to counteract slowing caused by performing two tasks concurrently. As a result, single-task and dual-task performance could be equal because more effort was exerted in the dual-task condition.

The concept of increases in effort leading to increases in available capacity may be sufficient to explain results such as those found by Schumacher et al. (2001). Schumacher et al. had participants make a vocal response to the pitch of a tone and a manual response to the location of a disk that was presented at one of three locations. There were three trial types: dual task (with a 0-ms SOA), heterogeneous single task (single task trials that were intermixed with the dual-task trials), and homogeneous single task (separate blocks of pitch trials or disk trials). After five sessions of practice, performance in all three conditions was statistically equivalent. Given that the two tasks were relatively easy, making the pair of to-be-performed tasks more difficult by requiring participants to perform both tasks concurrently could lead to an increase in effort on the part of the participants, which could in turn increase processing capacity. At first glance it would seem that, to counteract dual-task costs, effort would have to increase available capacity by a factor of two. However, this would only be true for the case in which Task 1 and Task 2 central processing overlap completely. This can be seen graphically in Figure 9A. If one task gains access to central processing before the other task, the factor by which effort needs to increase available capacity will be less than two. The factor by which effort needs to increase available capacity to counteract dual-task costs is directly related to the amount of central processing overlap. As can be seen by comparing Figure 9B with 9A, as central processing overlap decreases, so does the factor by which effort needs to increase available capacity. In Experiment 1 of Schumacher et al., the mean dual-task RTs were 283 ms for the visual-manual (disk) task and 456 ms for the auditory-vocal (tone) task. Given this discrepancy between mean RTs for the two tasks, it is likely that the visual-manual task gained access to central processing well before the auditory-vocal task required central resources. As a result, the factor by which effort would have to increase available capacity could be relatively modest (i.e., close to 1.0). Therefore it seems possible that the improvement in performance caused by an increase in available capacity caused by increased effort may have been sufficient to counteract the decrement in performance caused by having to share capacity between the two tasks. Given the potentially confounding effects of effort across the conditions in Schumacher et al.'s study, the claim that they have demonstrated "virtually perfect" time sharing may be false.

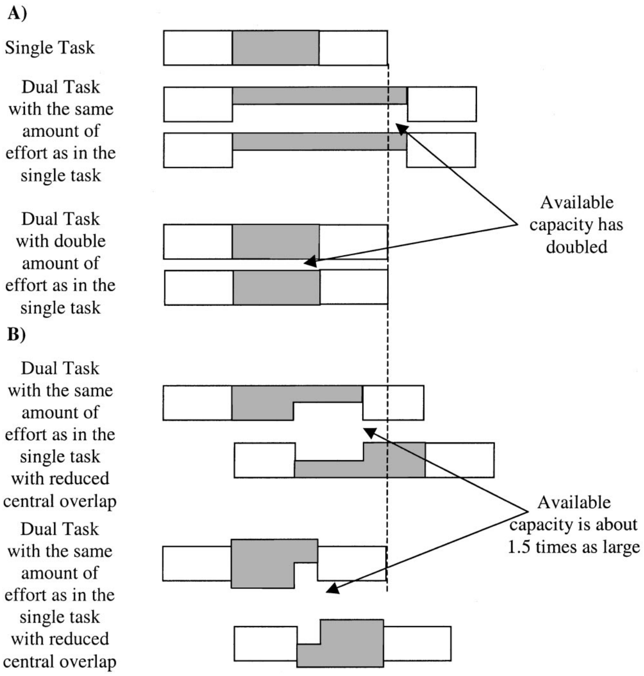
A)
Single Task
Dual Task with the same amount of effort as in the single task
Dual Task with double amount of effort as in the single task
Available capacity has doubled
B)
Dual Task with the same amount of effort as in the single task with reduced central overlap
Dual Task with the same amount of effort as in the single task with reduced central overlap
Available capacity is about 1.5 times as large

*Figure 9.* (A) When central processing of two tasks overlap completely, available capacity must be doubled to counteract dual-task interference. (B) However, as the degree of central processing overlap decreases, the additional available capacity required to counteract dual-task interference decreases (1.5 times the amount of available capacity in single-task situations is required to counteract slowing caused by dual-task interference). The dashed line shows that by increasing the available capacity, dual-task performance can equal single-task performance.

## Relaxation of the Assumption That Capacity Allocated to Task 1 and Task 2 Sums to Full Capacity

The previous discussion of effort has complicated what we might mean by "full capacity." For present purposes, we define full capacity as the available capacity at a fixed level of effort. We have also assumed that the capacity allocated to Task 1 plus the capacity allocated to Task 2 equals full capacity. It is possible, if not probable, that central capacity is subject to “overhead costs,” or concurrence costs (Navon & Gopher, 1979). Overhead costs could be envisioned as capacity allocated to keeping response mappings actively in mind, being prepared for two tasks, or remaining on the lookout for additional stimuli (see also Logan, 1978). For example, after the presentation of the first stimulus, some capacity may be held in reserve and used to keep Task 1 (and Task 2) mappings and stages actively in mind. As a result, some capacity would be withheld from Task 1 processing (and Task 2 processing at sufficiently short SOAs). Once the first task has been completed, however, active maintenance of mappings and stages is no longer needed, and this capacity can be added to the capacity allocated to Tasks 1 and 2. Similarly, some capacity may be allocated to remaining on the lookout for stimuli. Once the second stimulus has occurred, no more stimuli will occur, so capacity allocated for this purpose can be reallocated to processing the tasks. Consider what happens at long SOAs (Case B). Central processing of Task 1 and Task 2 does not overlap. However, a task performed as Task 1 will have less available capacity because some capacity is allocated to keeping mappings and stages in mind, and additional capacity is allocated to remaining on the lookout for stimuli. However, when the same task is performed as Task 2, there will be more available capacity, because the mappings and stages for only one task (Task 2) needs to be kept in mind, and no capacity needs to be allocated to remaining on the lookout for additional stimuli. As a result, a task performed as Task 1 will take longer to be performed than the same task at long SOAs as Task 2, in which overhead costs are diminished. The observation that, for the same task, RT2 at long SOAs is faster than RT1 at long SOAs supports this hypothesis (Logan & Schulkind, 2000).

It is possible to derive new equations for RT1 and RT2 taking overhead costs into consideration. We are assuming that overhead costs decrease as a function of time. As more time has elapsed from the onset of the first stimulus, Task 1 is more likely to be done, freeing up overhead required to keep Task 1 mappings and stages properly organized. If we let *x*(t) represent the proportion of capacity that is unavailable to the tasks (due to overhead costs), where t is the amount of time that has passed since the onset of Task 1, and SP represent the proportion of (full) capacity allocated to Task 1, new equations for RT1 and RT2 at short and long SOAs can be derived. Equations 1, 2, 4, and 5 can be reexpressed as Equations 6, 7, 8, and 9, respectively:

$$\mathrm{RT1(t)(long)} = \mathrm{A1} + \mathrm{B1} / [1 - x(\mathrm{t})] + \mathrm{C1}, \tag{6}$$

$$\mathrm{RT2(t)(long)} = \mathrm{A2} + \mathrm{B2} / [1 - x(\mathrm{t})] + \mathrm{C2}, \tag{7}$$

$$\begin{aligned}\mathrm{RT1(t)(short)} = {} & \{[1 - x(\mathrm{t})] / \mathrm{SP}\} \times \mathrm{A1} \\ & + \{[\mathrm{SP} - 1 + x(\mathrm{t})] / \mathrm{SP}\} \times \mathrm{A2} + \mathrm{B1} / \mathrm{SP} + \mathrm{C1} \\ & + \{[\mathrm{SP} - 1 + x(\mathrm{t})] / \mathrm{SP}\} \times \mathrm{SOA},\end{aligned} \tag{8}$$

$$\begin{aligned}\mathrm{RT2(t)(short)} = {} & \mathrm{A1} + \mathrm{B1} / [1 - x(\mathrm{t})] \\ & + \mathrm{B2} / [1 - x(\mathrm{t})] + \mathrm{C2} - \mathrm{SOA}.\end{aligned} \tag{9}$$

At long SOAs, a task performed as Task 1 will take longer to be performed than the same task as Task 2. This follows because *x*(t) decreases with time, and more time has elapsed from the onset of the first stimulus when a task is performed second. Determining the exact shape of the *x*(t) function is an empirical issue. At this point we assume only that it decreases as a function of time because Task 1 is increasingly likely to be finished. As a result, central capacity overhead associated with holding Task 1 ready is freed up and available for Task 2 processing.

The concept of overhead costs that decrease over time may explain why a PRP effect with a slope greater than −1 is sometimes observed. At short SOAs, RT2 is affected by two temporal parameters: SOA and *x*(t). The SOA component provides a −1 slope to the PRP function, but the *x*(t) also increases with decreasing SOA. As a result, this component would increase the slope of the PRP effect. In theory, this may explain why slopes of greater than −1 are sometimes observed. Assuming that there are individual differences in overhead costs, this may explain why some participants are more likely to exhibit this effect.

Equations 6–9 can also be applied to the previous discussion on effort. Instead of capacity being unavailable because of overhead costs, they can be conceptualized as being unavailable because less than full effort was marshaled to perform the task. It is interesting to note that the core predictions (see earlier section, PRP Phenomena Explained by the Central Capacity Sharing Model) of the central capacity sharing model hold even with relaxation of these two assumptions. However, it is unclear what form an effort function would take. Future research on both overhead costs and effort would be required to determine the shape of these functions.

## Rapid Switching Versus Capacity Sharing

An alternative to the central capacity sharing model is a bottleneck model in which bottleneck processing can switch rapidly between tasks. If this central bottleneck could be switched between tasks instantaneously and without cost, this model could make all of the predictions made by the central capacity sharing model. Miller and Bonnel (1994) explored the predictions for capacity sharing models and various task-switching models in a divided attention paradigm.

The observed results supported capacity models and deterministic switching models. In order for deterministic switching models to provide as good a fit as capacity sharing models, a switch from the left to the right side, or vice versa, would have to have taken 50 ms or less. In addition, participants would need to be switching at a rate faster than one switch every 20 ms. Miller and Bonnel (1994) argued that rapid switching models such as the one required to fit the observed data as well as a capacity model are neurologically implausible given the required rate of switching and average interspike time of 10–20 ms in cortical neurons. These results fail to support rapid switching and instead provide support for capacity sharing.

## Conclusion

We have introduced a capacity sharing model that can account for all of the phenomena that bottleneck models can account for. In addition, the central capacity sharing model can also account for Task 1 slowing with decreasing SOA and slower Task 1 dual-task performance compared with single-task performance. Further investigations of the specific predictions made by the central capacity sharing model are required to test its suitability as a model of human information processing.

## References

Bundesen, C. (1990). A theory of visual attention. *Psychological Review, 97,* 523–547.

Carrier, L. M., & Pashler, H. (1995). Attentional limits in memory retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 21,* 1339–1348.

DeJong, R. (1993). Multiple bottlenecks in overlapping-task performance. *Journal of Experimental Psychology: Human Perception and Performance, 19,* 965–989.

DeJong, R. (1995). The role of preparation in overlapping-task performance. *Quarterly Journal of Experimental Psychology, 48A,* 2–25.

Duncan, J. (1979). Divided attention: The whole is more than the sum of its parts. *Journal of Experimental Psychology: Human Perception and Performance, 5,* 216–228.

Herman, L., & Kantowitz, B. (1970). The psychological refractory period effect: Only half the double-stimulus story? *Psychological Bulletin, 73,* 74–88.

Herman, L., & McCauley, M. (1969). Delay in responding to the first stimulus in the "psychological refractory period" experiment: Comparisons with delay produced by a second stimulus not requiring a response. *Journal of Experimental Psychology, 81,* 344–350.

Hommel, B. (1998). Automatic stimulus–response translation in dual-task performance. *Journal of Experimental Psychology: Human Perception and Performance, 24,* 1368–1384.

Hommel, B., & Eglau, B. (in press). Control of stimulus–response translation in dual-task performance. *Psychological Research.*

Kahneman, D. (1973). *Attention and effort.* Englewood Cliffs, NJ: Prentice Hall.

Logan, G. (1978). Attention in character-classification tasks: Evidence for the automaticity of component stages. *Journal of Experimental Psychology: General, 107,* 32–63.

Logan, G., & Delheimer, J. (2001). Parallel memory retrieval in dual-task situations: II. Episodic memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 27,* 668 –685.

Logan, G., & Gordon, R. (2001). Executive control of visual attention in dual-task situations. *Psychological Review, 108,* 393–434.

Logan, G., & Schulkind, M. (2000). Parallel memory retrieval in dual-task situations: I. Semantic memory. *Journal of Experimental Psychology: Human Perception and Performance, 26,* 1072–1090.

McCann, R., & Johnston, J. (1992). Locus of the single-channel bottleneck in dual-task interference. *Journal of Experimental Psychology: Human Perception and Performance, 18,* 471–484.

McLeod, P. (1977). Parallel processing and the psychological refractory period. *Acta Psychologica, 41,* 381–391.

Meyer, D., & Kieras, D. (1997a). A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms. *Psychological Review, 104,* 3–65.

Meyer, D., & Kieras, D. (1997b). A computational theory of executive cognitive processes and multiple-task performance: Part 2. Accounts of a psychological refractory-period phenomena. *Psychological Review, 104,* 749–791.

Miller, J., & Bonnel, A. M. (1994). Switching or sharing in dual-task line-length discrimination? *Perception & Psychophysics, 56,* 431–446.

Navon, D., & Gopher, D. (1979). On the economy of the human-processing system. *Psychological Review, 86,* 214–255.

Navon, D., & Miller, J. (2002). Queuing or sharing? A critical evaluation of the single-bottleneck notion. *Cognitive Psychology, 44,* 193–251.

Nosofsky, R., & Palmeri, T. (1997). An exemplar-based random walk model of speeded classification. *Psychological Review, 104,* 266–300.

Pashler, H. (1990). Do response modality effects support multiprocessor models of divided attention? *Journal of Experimental Psychology: Human Perception and Performance, 16,* 826–842.

Pashler, H. (1991). Shifting visual attention and selecting motor responses: Distinct attentional mechanisms. *Journal of Experimental Psychology: Human Perception and Performance, 17,* 1023–1040.

Pashler, H. (1994a). Dual-task interference in simple tasks: Data and theory. *Psychological Bulletin, 116,* 220–244.

Pashler, H. (1994b). Graded capacity-sharing in dual-task interference? *Journal of Experimental Psychology: Human Perception and Performance, 20,* 330–342.

Pashler, H., & Johnston, J. (1989). Chronometric evidence for central postponement in temporally overlapping tasks. *Quarterly Journal of Experimental Psychology, 41A,* 19–45.

Posner, M., & Boies, S. (1971). Components of attention. *Psychological Review, 78,* 391–408.

Schumacher, E., Seymour, T., Glass, J., Fencsik, D., Lauber, E., Kieras, D., & Meyer, D. (2001). Virtually perfect time sharing in dual-task performance: Uncorking the central cognitive bottleneck. *Psychological Science, 12,* 101–108.

Telford, C. (1931). The refractory phase of voluntary and associative response. *Journal of Experimental Psychology, 14,* 1–35.

Tombu, M., & Jolicœur, P. (2000, November). *Is the PRP effect due to a strategic or structural bottleneck?* Poster presented at the annual meeting of the Psychonomic Society, New Orleans, LA.

Tombu, M., & Jolicœur, P. (2002). All-or-none bottleneck versus capacity sharing accounts of the psychological refractory period phenomenon. *Psychological Research, 66,* 274–286.

Van Selst, M., & Jolicœur, P. (1997). Decision and response in dual-task interference. *Cognitive Psychology, 33,* 266–307.

Welford, A. (1952). The "psychological refractory period" and the timing of high speed performance: A review and a theory. *British Journal of Psychology, 43,* 2–19.

# Appendix

## The Central Capacity Sharing Model

We now introduce the additional set of predictions made by the central capacity sharing model for the remaining possible combinations of central processing overlap between Task 1 and Task 2, as well as another possible case in which central processing of Task 1 and Task 2 does not overlap. We also introduce four conditions that can be used to determine which case is appropriate under a given set of circumstances.

The total time taken to perform a task in isolation can be expressed as the sum of precentral, central, and postcentral stages. We refer to these as A, B, and C, respectively, where each letter represents the total amount of time needed to complete a stage under full capacity (this distinction is only important to central stages, which are capacity limited). Depending on how Task 1 and Task 2 central stages overlap (see Figure 5), a different set of equations for determining response times (RTs) to Task 1 and Task 2 are needed. Determining which case is applicable in a given situation requires knowing whether each of the following four conditions is true or false:

Condition 1: Task 1 starts central processing before Task 2 central processing starts.

Condition 2: Task 1 finishes central processing before Task 2 central processing finishes.

Condition 3: Task 2 starts central processing before Task 1 finishes central processing.

Condition 4: Task 2 finishes central processing before Task 1 starts central processing.

These conditions can be expressed mathematically as follows.

C1 can be expressed as:
If $A1 \leq A2 + SOA$, then C1 is true, else C1 is false.

C2 is dependent on C1.

If C1 is true, then C2 can be expressed as:
If $A1 - SOA - A2 + B1 + B2 - B2 / (1 - SP) \leq 0$, then C2 is true, else C2 is false.

If C1 is false, then C2 can be expressed as:
If $A1 - SOA - A2 - B1 - B2 + B1 / SP \leq 0$, then C2 is true, else C2 is false.

C3 can be expressed as:
If $A2 + SOA \leq A1 + B1$, then C3 is true, else C3 is false.

Finally, C4 can be expressed as:
If $SOA + A2 + B2 \leq A1$, then C4 is true, else C4 is false.

A different set of equations is needed for every possible combination of violations and confirmations of these conditions. In total, there are six combinations of conditions that could naturally occur. The patterns of temporal overlap (or lack thereof) of central processing of Task 1 and Task 2 are shown in Figure 5. Next, go through the equations that are associated with each combination and the conditions for which they hold, in turn (with the exception of the two cases already discussed).

### *Case C: Central Processing for Task 1 and Task 2 Overlap: Task 1 Starts Central Processing Before Task 2 Starts Central Processing and Finishes Central Processing After Task 2 Finishes Central Processing*

Figure 5, Case C depicts the central processing time course for Case C. Central processing of Task 1 begins before central processing of Task 2, but Task 2 finishes central processing before Task 1 finishes central processing. In terms of the conditions outlined earlier, Conditions 2 and 4 are false and Conditions 1 and 3 are true. In this case it is simplest to begin with the Task 2 RT. All of Task 2's central processing is performed while sharing capacity with Task 1. Therefore, RT2 can be expressed as

$$RT2 = A2 + B2/(1 - SP) + C2. \tag{A1}$$

Task 1 has full access to central processing initially. As earlier, this time period is equal to $SOA + A2 - A1$. Following this, Task 1 must share resources with Task 2 for the next $B2 / (1 - SP)$ units of time, after which Task 1 regains full access to processing capacity. To determine how much longer Task 1 central processing will take, we need to determine the amount of processing that has been done to date and subtract it from the total amount of central processing that Task 1 requires (B1). This amount can be expressed as $B1 - [SOA + A2 - A1 + B2 \times SP/(1 - SP)]$. Adding up all the components of central processing and pre- and postcentral processing gives

$$RT1 = A1 + SOA + A2 - A1 + B2 / (1 - SP) + B1 - [SOA + A2 - A1 + B2 \times SP / (1 - SP)] + C1.$$

This can be simplified to

$$RT1 + A1 + B1 + C1 + B2. \tag{A2}$$

Under Case C, all of Task 2 central processing takes place while central processing in Task 1 is also taking place concurrently. Hence all of Task 2 central processing is slowed because of capacity sharing. If the sharing proportion is skewed toward Task 1, as would usually be expected, Task 2 RTs will be particularly sensitive to manipulations affecting B2, because B2 effects are magnified owing to the chronic sharing in this case.

Comparing this case with Case A (expected at long stimulus onset asynchronies [SOAs]), Task 2 manipulations affecting B2 should be overadditive with decreasing SOA. As SOA is decreased, one would first expect Case A to prevail, with RT2 described by Equation 2. For some range of SOAs, Case B should prevail, with no change in the size of the manipulations affecting B2 (Equation 5). As SOA is shortened further, Case C would then exist, at which point B2 effects would be magnified by $1 / (1 - SP)$. Also RT1 should be strongly affected by manipulations of B2 (see Equation 7 above).

### *Case D: Central Processing for Task 1 and Task 2 Overlap: Task 1 Starts Central Processing After Task 2 Starts Central Processing and Finishes Central Processing Before Task 2 Finishes Central Processing*

Figure 5, Case D shows the signature time course for this scenario. Case D is symmetric with Case C. Now Task 2 starts central processing before Task 1 does, but Task 1 finishes central processing before Task 2 does. In terms of the conditions introduced earlier, Conditions 1 and 4 are false, whereas Conditions 2 and 3 are true. Similar to Task 2 in the previous case, Task 1 RTs can be expressed as

$$RT1 = A1 + B1 / SP + C1. \tag{A3}$$

The derivation of RT2 is best understood by breaking Task 2 central processing into three parts. The first part is the amount of time for which Task 2 initially has full central processing capacity: $A1 - SOA - A2$. The second component of Task 2 central processing describes the period during which Task 1 and Task 2 share central processing. This is the total amount of time it takes for Task 1 to complete central processing: $B1 / SP$. Finally,

(*Appendix continues*)

we need to compute the amount of time it takes for Task 2 to finish central processing after Task 1 has completed central processing and Task 2 regains full central processing capacity. This will equal the total amount of central processing done to date, subtracted from the total amount of central processing required: B2 − [A1 − SOA − A2 + (1 − SP) × B1 / SP]. Putting all these components together along with pre- and postcentral processing gives the Task 2 RT:

$$\mathrm{RT2} = \mathrm{A2} + \mathrm{A1} - \mathrm{SOA} - \mathrm{A2} + \mathrm{B1} / \mathrm{SP} + \mathrm{B2} - [\mathrm{A1} - \mathrm{SOA} - \mathrm{A2} + (1 - \mathrm{SP}) \times \mathrm{B1} / \mathrm{SP}] + \mathrm{C2}.$$

Simplifying the above equation yields

$$\mathrm{RT2} = \mathrm{A2} + \mathrm{B2} + \mathrm{C2} + [1 - (1 - \mathrm{SP})] \times \mathrm{B1} / \mathrm{SP},$$

which can be simplified further to

$$\mathrm{RT2} = \mathrm{A2} + \mathrm{B2} + \mathrm{C2} + \mathrm{B1}. \quad \text{(A4)}$$

Case D, like Case C, requires a significant imbalance in the amount of central processing required for Task 1 and Task 2. Furthermore, for Case D, a large imbalance in the duration of precentral processing across tasks would also be needed to instantiate conditions leading to Case D. Should these conditions prevail, however, one would expect RT1 to reflect manipulations affecting B1 in an overadditive fashion, relative to a long SOA baseline described by Case A. If the SP was biased toward Task 1, however, only moderate overadditivity would be expected (1 / SP). RT2 would be expected to depend on B1 and thus also to be affected by manipulations affecting B1.

### *Case E: Central Processing for Task 1 and Task 2 Overlap: Task 1 Starts Central Processing After Task 2 Starts Central Processing and Finishes Central Processing After Task 2 Finishes Central Processing*

Figure 5, Case E depicts the time course of central processing for Task 1 and 2 under this set of circumstances. Task 2 starts central processing before Task 1 starts, and Task 2 finishes central processing before Task 1 finishes. There is central processing overlap as Task 1 starts central processing before Task 2 finishes. Initially, Task 2 has the entire central processing capacity. However, this only lasts for the first A1 − SOA − A2 time units of central processing, at which time Task 1 also uses some of the available capacity. Task 2 then performs the remainder of its central processing under sharing conditions. The amount of time this takes can be expressed as [B2 − (A1 − SOA − A2)] / (1 − SP). Adding up all of the components of the Task 2 RT yields

$$\mathrm{RT2} = \mathrm{A2} + \mathrm{A1} - \mathrm{SOA} - \mathrm{A2} + [\mathrm{B2} - (\mathrm{A1} - \mathrm{SOA} - \mathrm{A2})] / (1 - \mathrm{SP}) + \mathrm{C2},$$

yielding the following equation after some simplification:

$$\mathrm{RT2} = [1 - 1 / (1 - \mathrm{SP})] \times \mathrm{A1} - [1 - 1 / (1 - \mathrm{SP})] \times \mathrm{SOA} + \mathrm{A2} / (1 - \mathrm{SP}) + \mathrm{B2} / (1 - \mathrm{SP}) + \mathrm{C2}. \quad \text{(A5)}$$

Task 1 central processing time can be expressed as the amount of time that Task 1 spent sharing capacity with Task 2 ([B2 − (A1 − SOA − A2)] / (1 − SP)), plus the amount of time it takes for central processing to be finished. This second term can be expressed as the total amount of central processing required minus the amount of processing already performed: B1 − SP × [B2 − (A1 − SOA − A2)] / (1 − SP). Putting precentral, central, and postcentral components together yields

$$\mathrm{RT1} = \mathrm{A1} + [\mathrm{B2} - (\mathrm{A1} - \mathrm{SOA} - \mathrm{A2})] / (1 - \mathrm{SP}) + \mathrm{B1} - \mathrm{SP} \times [\mathrm{B2} - (\mathrm{A1} - \mathrm{SOA} - \mathrm{A2})] / (1 - \mathrm{SP}) + \mathrm{C1},$$

and this equation simplifies to

$$\mathrm{RT1} = \mathrm{B1} + \mathrm{C1} + \mathrm{B2} + \mathrm{SOA} + \mathrm{A2}. \quad \text{(A6)}$$

RT1 is not dependent on the duration of Task 1 precentral processing, but it is dependent on Task 2 precentral and central processing, as well as SOA.

Should conditions in Case E prevail, RT1 would depend on A2, with slower response as the duration of precentral processing in Task 2 was increased. RT1 would also depend on SOA, with a positive slope of +1. Both of these effects would occur because increasing SOA or the duration of A2 has for effect to increase overlap of central processing, leading to a larger period of capacity sharing, and thus to longer processing times.

Similarly, RT2 would be expected to decrease as SOA was shortened, because reducing SOA would lead to a shorter duration of capacity sharing. Increasing the duration of A1 would also lead to faster RT2s. Also, relative to a long SOA baseline (Case A), manipulations affecting A2 and B2 would be overadditive.

### *Case F: Central Processing of Task 1 and Task 2 Do Not Overlap: Task 2 Starts and Finishes Central Processing Before Task 1 Starts Central Processing*

Figure 5, Case F depicts the time course of central processing for this scenario. Task 1 and 2 central processing does not overlap, and Task 2 central processing occurs before Task 1 central processing. In terms of the conditions outlined above, Conditions 1 and 2 are false and Conditions 3 and 4 are true. In this case Task 1 and 2 RTs can be expressed as

$$\mathrm{RT1} = \mathrm{A1} + \mathrm{B1} + \mathrm{C1}, \quad \text{(A7)}$$

$$\mathrm{RT2} = \mathrm{A2} + \mathrm{B2} + \mathrm{C2}. \quad \text{(A8)}$$

The central processing timing scenario in Case F could perhaps be achieved at very short SOAs if the duration of A1 was very long while the duration of A2 was very short. Under these conditions, even at very short SOAs, factor manipulations affecting any stage of processing would only affect the task in which the factor was manipulated, and the observed effects should be additive with SOA relative to a single long SOA baseline in which conditions for Case A prevailed.

Received March 16, 2001
Revision received July 19, 2002
Accepted August 9, 2002 ■