1) Um processo pode usar o mesmo soquete para enviar e receber mensagens? Porque?

Sim.
Cada computador tem 2 números de portas disponíveis para serem usados pelos processos para envio e recepção de mensagens.

2) Um processo pode compartilhar portas com outros processos no mesmo computador? Porque?

Não.
Qualquer processo pode fazer uso de várias portas para receber mensagens, mas um processo não pode compartilhar portas com outros processos no mesmo computador.

3) Para enviar ou receber mensagens, um processo precisa primeiro criar o que?

Para enviar ou receber mensagens, um processo precisa primeiro criar uma associação entre um soquete com um endereço IP e com uma porta do host local.

4) Um servidor associa seu soquete a uma porta de serviço que ele torna conhecida dos clientes para que estes possam enviar mensagens a ela. Qual a regra para associação deste soquete a uma porta?

Um cliente vincula seu soquete a qualquer porta local livre.
O método receive retorna, além da mensagem, o endereço IP e a porta da origem permitindo que o destinatário envie uma resposta a este.

5) Qual o problema relacionado à comunicação por datagrama, quanto ao tamanho da mensagem?

Se a mensagem for grande demais para esse vetor, ela será truncada na chegada. O protocolo IP permite datagramas de até 2 elevado a 16 bytes (64 KB), incluindo seu cabeçalho e a área de dados. Entretanto, a maioria dos ambientes impõe uma restrição de tamanho de 8 kbytes. Qualquer aplicativo que exija mensagens maiores do que o tamanho máximo deve fragmentá-las em porções desse tamanho.

6) Qual o problema relacionado à comunicação por datagrama, quanto ao Bloqueio?

Normalmente, os soquetes fornecem operações send não bloqueantes e receive bloqueantes para comunicação por datagrama. A operação send retorna quando tiver repassado a mensagem para as camadas UDP e IP subjacentes, que são responsáveis por transmitila para seu destino.

7) Se o processo que invoca o método receive tiver outra tarefa para fazer enquanto espera pela mensagem, qual o método utilizado?

Ele deve tomar providências para usar threads separadas.

8) Não é adequado que um processo espere indefinidamente para receber algo, pois o processo remetente pode ter falhado ou a mensagem esperada pode ter se perdido. O que é feito neste caso para corrigir isso?

Para atender a tais requisitos, limites temporais (timeouts) podem ser configurados nos soquetes. É difícil escolher um timeout apropriado, porém ele deve ser grande, em comparação com o tempo exigido para transmitir uma mensagem.

9) Em que situações as mensagens podem ser descartadas?

Ocasionalmente, mensagens podem ser descartadas devido a erros de soma de verificação ou porque não há espaço disponível no buffer, na origem ou no destino.

10) Nos serviços de: Domain Name Service DNS, o Voice Over IP (VoIP), são interessantes usar os datagramas UDP , são uma escolha atraente, porque?

os datagramas UDP são uma escolha atraente, pois não sofrem as sobrecargas necessárias à entrega de mensagens garantida. Existem três fontes de sobrecarga principais:
• a necessidade de armazenar informações de estado na origem e no destino;
• a transmissão de mensagens extras;
• a latência do remetente.